Package ‘riskyr’

January 3, 2019

Type Package
Title Rendering Risk Literacy more Transparent
Version 0.2.0
Date 2018-12-20
Author Hansjoerg Neth [aut, cre],
       Felix Gaisbauer [aut],
       Nico Gradwohl [aut],
       Wolfgang Gaissmaier [aut]
Maintainer Hansjoerg Neth <h.neth@uni.kn>
Description Risk-related information (like the prevalence of conditions and the sensitivity and specificity of diagnostic tests or treatment decisions) can be expressed in terms of probabilities or frequencies. By providing a toolbox of methods and metrics, 'riskyr' computes, translates, and visualizes risk-related information in a variety of ways. Offering multiple complementary perspectives on the interplay between key parameters renders teaching and training of risk literacy more transparent.
Depends R (>= 3.4.0)
Imports utils (>= 3.4.0)
Suggests devtools, rmarkdown, knitr, roxygen2, pkgdown, spelling
Collate 'comp_util.R' 'init_txt.R' 'init_pal.R' 'init_prob.R'
       'comp_prob_prob.R' 'init_freq.R' 'comp_min_N.R' 'init_num.R'
       'init_prob_num.R' 'init_freq_num.R' 'comp_freq_freq.R'
       'comp_prob_freq.R' 'comp xxxx_prob.R' 'comp_popu.R'
       'comp_accu.R' 'plot_util.R' 'plot_area.R' 'plot_tab.R'
       'plot_prism.R' 'plot_bar.R' 'plot_icons.R' 'plot_curve.R'
       'plot_plane.R' 'plot_fnet.R' 'plot_tree.R' 'plot_mosaic.R'
       'data.R' 'read_data.R' 'riskyr_class.R' 'start_riskyr.R'
Encoding UTF-8
LazyData true
License GPL-2 | GPL-3
BugReports https://github.com/hneth/riskyr/issues
R topics documented:

VignetteBuilder knitr
RoxygenNote 6.1.1
Language en-US
NeedsCompilation no
Repository CRAN
Date/Publication 2019-01-03 01:00:03 UTC

R topics documented:

acc .................................................. 4
accu .................................................. 5
as_pb ................................................ 7
as_pc ................................................ 8
comp_acc .......................................... 9
comp_accu_freq .................................. 11
comp_accu_prob ................................. 13
comp_complement ............................... 16
comp_complete_prob_set ...................... 17
comp_comp_pair ................................. 19
comp_err ......................................... 20
comp_fart ....................................... 21
comp_FDR ....................................... 22
comp_FOR ....................................... 23
comp_freq ....................................... 24
comp_freq_freq ................................. 27
comp_freq_prob ................................. 28
comp_min_N ..................................... 31
comp_mirt ....................................... 33
comp_NPV ....................................... 34
comp_popu ..................................... 35
comp_ppod ..................................... 36
comp_PPV ....................................... 38
comp_prev ....................................... 39
comp_prob ....................................... 40
comp_prob_freq ................................. 43
comp_prob_prob ................................. 44
comp_sens ...................................... 47
comp_spec ...................................... 48
cond_false ...................................... 49
cond_true ...................................... 50
cr .................................................. 51
dec_cor ......................................... 52
dec_err .......................................... 53
dec_neg .......................................... 55
dec_pos .......................................... 56
df_scenarios .................................... 57
### R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>err</td>
<td>58</td>
</tr>
<tr>
<td>fa</td>
<td>59</td>
</tr>
<tr>
<td>fart</td>
<td>60</td>
</tr>
<tr>
<td>FDR</td>
<td>61</td>
</tr>
<tr>
<td>FOR</td>
<td>63</td>
</tr>
<tr>
<td>freq</td>
<td>64</td>
</tr>
<tr>
<td>hi</td>
<td>65</td>
</tr>
<tr>
<td>init_num</td>
<td>66</td>
</tr>
<tr>
<td>init_pal</td>
<td>68</td>
</tr>
<tr>
<td>init_txt</td>
<td>69</td>
</tr>
<tr>
<td>is_complement</td>
<td>71</td>
</tr>
<tr>
<td>is_extreme_prob_set</td>
<td>73</td>
</tr>
<tr>
<td>is_freq</td>
<td>75</td>
</tr>
<tr>
<td>is_perc</td>
<td>76</td>
</tr>
<tr>
<td>is_prob</td>
<td>77</td>
</tr>
<tr>
<td>is_suff_prob_set</td>
<td>78</td>
</tr>
<tr>
<td>is_valid_prob_pair</td>
<td>79</td>
</tr>
<tr>
<td>is_valid_prob_set</td>
<td>80</td>
</tr>
<tr>
<td>is_valid_prob_triple</td>
<td>82</td>
</tr>
<tr>
<td>mi</td>
<td>84</td>
</tr>
<tr>
<td>mirt</td>
<td>85</td>
</tr>
<tr>
<td>N</td>
<td>86</td>
</tr>
<tr>
<td>NPV</td>
<td>87</td>
</tr>
<tr>
<td>num</td>
<td>88</td>
</tr>
<tr>
<td>pal</td>
<td>89</td>
</tr>
<tr>
<td>pal_bw</td>
<td>90</td>
</tr>
<tr>
<td>pal_kn</td>
<td>91</td>
</tr>
<tr>
<td>pal_mbw</td>
<td>92</td>
</tr>
<tr>
<td>pal_mod</td>
<td>93</td>
</tr>
<tr>
<td>pal_org</td>
<td>93</td>
</tr>
<tr>
<td>pal_rgb</td>
<td>94</td>
</tr>
<tr>
<td>pal_vir</td>
<td>95</td>
</tr>
<tr>
<td>plot.box</td>
<td>96</td>
</tr>
<tr>
<td>plot.risky</td>
<td>96</td>
</tr>
<tr>
<td>plot_area</td>
<td>98</td>
</tr>
<tr>
<td>plot_bar</td>
<td>103</td>
</tr>
<tr>
<td>plot_curve</td>
<td>105</td>
</tr>
<tr>
<td>plot_fnet</td>
<td>108</td>
</tr>
<tr>
<td>plot_icons</td>
<td>111</td>
</tr>
<tr>
<td>plot_mosaic</td>
<td>114</td>
</tr>
<tr>
<td>plot_plane</td>
<td>115</td>
</tr>
<tr>
<td>plot_prism</td>
<td>117</td>
</tr>
<tr>
<td>plot_tab</td>
<td>123</td>
</tr>
<tr>
<td>plot_tree</td>
<td>127</td>
</tr>
<tr>
<td>popu</td>
<td>130</td>
</tr>
<tr>
<td>ppod</td>
<td>131</td>
</tr>
<tr>
<td>PPV</td>
<td>132</td>
</tr>
<tr>
<td>prev</td>
<td>133</td>
</tr>
</tbody>
</table>
Accuracy (acc) is the probability of a correct decision.

Description

acc defines overall accuracy as the probability of correspondence between a positive decision and true condition (i.e., the proportion of correct classification decisions or of \texttt{dec.cor} cases).

Usage

acc

Format

An object of class \texttt{numeric} of length 1.

Details

Importantly, correct decisions \texttt{dec.cor} are not necessarily positive decisions \texttt{dec.pos}.

Understanding or obtaining the accuracy metric acc:

- Definition: acc is the (non-conditional) probability:
  \[
  \text{acc} = \frac{\text{dec.cor}}{\text{N}}
  \]
  or the base rate (or baseline probability) of a decision being correct, but not necessarily positive.
  acc values range from 0 (no correct decision/prediction) to 1 (perfect decision/prediction).
- Computation: acc can be computed in several ways:
  (a) from \texttt{prob}: \texttt{acc} = (\texttt{prev} \times \texttt{sens}) + [(1 - \texttt{prev}) \times \texttt{spec}]
  (b) from \texttt{freq}: \texttt{acc} = \texttt{dec.cor}/\texttt{N} = \frac{\texttt{hi} + \texttt{cr}}{\texttt{hi} + \texttt{mi} + \texttt{fa} + \texttt{cr}}
  (c) as complement of the error rate \texttt{err}: \texttt{acc} = 1 - \texttt{err}

When frequencies in \texttt{freq} are not rounded, (b) coincides with (a) and (c).
• Perspective: acc classifies a population of N individuals by accuracy/correspondence (acc = dec_cor / N). acc is the "by accuracy" or "by correspondence" counterpart to prev (which adopts a "by condition" perspective) and to ppod (which adopts a "by decision" perspective).

• Alternative names: base rate of correct decisions, non-erroneous cases

• In terms of frequencies, acc is the ratio of dec_cor (i.e., hi + cr) divided by N (i.e., hi + mi + fa + cr):
  acc = dec_cor / N = (hi + cr) / (hi + mi + fa + cr)

• Dependencies: acc is a feature of both the environment (true condition) and of the decision process or diagnostic procedure. It reflects the correspondence of decisions to conditions.

See accu for other accuracy metrics and several possible interpretations of accuracy.

References
Consult Wikipedia:Accuracy_and_precision for additional information.

See Also
comp_acc computes accuracy from probabilities; accu lists all accuracy metrics; comp_accu_prob computes exact accuracy metrics from probabilities; comp_accu_freq computes accuracy metrics from frequencies; comp_sens and comp_PPV compute related probabilities; is_extreme_prob_set verifies extreme cases; comp_complement computes a probability’s complement; is_complement verifies probability complements; comp_prob computes current probability information; prob contains current probability information; is_prob verifies probabilities.

Other probabilities: FDR, FOR, NPV, PPV, err, fart, mirt, ppod, prev, sens, spec
Other metrics: accu, comp_accu_freq, comp_accu_prob, comp_acc, comp_err, err

Examples
acc <- .50       # sets a rate of correct decisions of 50%
acc <- 50/100    # (dec_cor) for 50 out of 100 individuals
is_prob(acc)     # TRUE

---

accu

A list containing current accuracy information.

Description
accu contains current accuracy information returned by the corresponding generating function comp_accu_prob.

Usage
accu
Format

An object of class `list` of length 5.

Details

Current metrics include:

1. **acc**: Overall accuracy as the probability (or proportion) of correctly classifying cases or of `dec_cor` cases:
   - See `acc` for definition and explanations.
   - `acc` values range from 0 (no correct prediction) to 1 (perfect prediction).

2. **wacc**: Weighted accuracy, as a weighted average of the sensitivity `sens` (aka. hit rate `HR`, `TPR`, `power` or `recall`) and the the specificity `spec` (aka. `TNR`) in which `sens` is multiplied by a weighting parameter `w` (ranging from 0 to 1) and `spec` is multiplied by `w`'s complement (`1 - w`):
   - `wacc = (w * sens) + ((1 - w) * spec)`
   - If `w = .50`, `wacc` becomes balanced accuracy `bacc`.

3. **mcc**: The Matthews correlation coefficient (with values ranging from -1 to +1):
   - `mcc = ((hi * cr) - (fa * mi)) / sqrt((hi + fa) * (hi + mi) * (cr + fa) * (cr + mi))`
   - A value of `mcc = 0` implies random performance; `mcc = 1` implies perfect performance.
   - See Wikipedia: Matthews correlation coefficient for additional information.

4. **f1s**: The harmonic mean of the positive predictive value `PPV` (aka. `precision`) and the sensitivity `sens` (aka. hit rate `HR`, `TPR`, `power` or `recall`):
   - `f1s = 2 * (PPV * sens) / (PPV + sens)`
   - See Wikipedia: F1 score for additional information.

Notes:

- Accuracy metrics describe the correspondence of decisions (or predictions) to actual conditions (or truth).
- There are several possible interpretations of accuracy:
  1. as *probabilities* (i.e., `acc` being the probability or proportion of correct classifications, or the ratio `dec_cor/N`),
  2. as *frequencies* (e.g., as classifying a population of `N` individuals into cases of `dec_cor` vs. `dec_err`),
  3. as *correlations* (e.g., see `mcc` in `accu`).

- Computing exact accuracy values based on probabilities (by `comp_accu_prob`) may differ from accuracy values computed from (possibly rounded) frequencies (by `comp_accu_freq`).
- When frequencies are rounded to integers (see the default of `round = TRUE` in `comp_freq` and `comp_freq_prob`) the accuracy metrics computed by `comp_accu_freq` correspond to these rounded values. Use `comp_accu_prob` to obtain exact accuracy metrics from probabilities.
as_pb

See Also

The corresponding generating function `comp_accu_prob` computes exact accuracy metrics from probabilities; `acc` defines accuracy as a probability; `comp_accu_freq` computes accuracy metrics from frequencies; `num` for basic numeric parameters; `freq` for current frequency information; `prob` for current probability information; `txt` for current text settings.

Other lists containing current scenario information: `freq`, `num`, `pal_bw`, `pal_kn`, `pal_mbw`, `pal_mod`, `pal_org`, `pal_rgb`, `pal_vir`, `pal`, `prob`, `txt_TF`, `txt_org`, `txt`

Other metrics: `acc`, `comp_accu_freq`, `comp_accu_prob`, `comp_acc`, `comp_err`, `err`

Examples

```r
accu <- comp_accu_prob()  # => compute exact accuracy metrics (from probabilities)
accu  # => current accuracy information

## Contrasting comp_accu_freq and comp_accu_prob:
# (a) comp_accu_freq (based on rounded frequencies):
freq1 <- comp_freq(N = 10, prev = 1/3, sens = 2/3, spec = 3/4)  # => rounded frequencies!
accu1 <- comp_accu_freq(freq1$h, freq1$m, freq1$f, freq1$cr)  # => accu1 (based on rounded freq).
accu1

#  # (b) comp_accu_prob (based on probabilities):
accu2 <- comp_accu_prob(prev = 1/3, sens = 2/3, spec = 3/4)  # => exact accu (based on prob).
accu2
all.equal(accu1, accu2)  # => 4 differences!

# (c) comp_accu_freq (exact values, i.e., without rounding):
freq3 <- comp_freq(N = 10, prev = 1/3, sens = 2/3, spec = 3/4, round = FALSE)
accu3 <- comp_accu_freq(freq3$h, freq3$m, freq3$f, freq3$cr)  # => accu3 (based on EXACT freq).
accu3
all.equal(accu2, accu3)  # => TRUE (qed).
```

---

as_pb

Display a percentage as a (numeric and rounded) probability.

Description

`as_pb` is a function that displays a percentage `perc` as a probability (rounded to `n_digits` decimals).

Usage

```r
as_pb(perc, n_digits = 4)
```

Arguments

- `perc` A percentage (as a scalar or vector of numeric values from 0 to 100).
- `n_digits` Number of decimal places to which percentage is rounded. Default: `n_digits = 4`. 

Details

as_pb and its complement function as_pc allow toggling the display of numeric values between percentages and probabilities.

Value

A probability (as a numeric value).

See Also

is_perc verifies a percentage; is_prob verifies a probability; is_valid_prob_set verifies the validity of probability inputs; num contains basic numeric variables; init_num initializes basic numeric variables; prob contains current probability information; comp_prob computes current probability information; freq contains current frequency information; comp_freq computes current frequency information; comp_complement computes a probability’s complement; comp_comp_pair computes pairs of complements.

Other utility functions: as_pc, plot.box

Other display functions: as_pc

Examples

as_pb(1/3) # => 0.3333
as_pb(as_pc(2/3)) # => 0.6667 (rounded to 4 decimals)

as_pc

Display a probability as a (numeric and rounded) percentage.

Description

as_pc is a function that displays a probability prob as a percentage (rounded to n_digits decimals).

Usage

as_pc(prob, n_digits = 2)

Arguments

prob A probability (as a scalar or vector of numeric values from 0 to 1).

n_digits Number of decimal places to which percentage is rounded. Default: n_digits = 2.

Details

as_pc and its complement function as_pb allow toggling the display of numeric values between percentages and probabilities.
comp_acc

Value

A percentage (as a numeric value).

See Also

is_prob verifies a probability; is_perc verifies a percentage; is_valid_prob_set verifies the validity of probability inputs; num contains basic numeric variables; init_num initializes basic numeric variables; prob contains current probability information; comp_prob computes current probability information; freq contains current frequency information; comp_freq computes current frequency information; comp_complement computes a probability’s complement; comp_comp_pair computes pairs of complements.

Other utility functions: as_pb, plot.box

Other display functions: as_pb

Examples

as_pc(.50)   #  50
as_pc(1/3)   # 33.33
as_pc(1/3, n_digits = 0) # 33
as_pc(as_pb(12.3)) # 12.3

comp_acc

Compute overall accuracy (acc) from probabilities.

Description

comp_acc computes overall accuracy acc from 3 essential probabilities prev, sens, and spec.

Usage

comp_acc(prev, sens, spec)

Arguments

prev    The condition’s prevalence prev (i.e., the probability of condition being TRUE).
sens    The decision’s sensitivity sens (i.e., the conditional probability of a positive decision provided that the condition is TRUE).
spec    The decision’s specificity value spec (i.e., the conditional probability of a negative decision provided that the condition is FALSE).
Details

comp_acc uses probabilities (not frequencies) as inputs and returns an exact probability (proportion) without rounding.

Understanding the probability acc:

• Definition: acc is the (non-conditional) probability:
  \[ \text{acc} = p(\text{dec\_cor}) = \frac{\text{dec\_cor}}{N} \]
  or the base rate (or baseline probability) of a decision being correct, but not necessarily positive.
  acc values range from 0 (no correct decision/prediction) to 1 (perfect decision/prediction).

• Computation: acc can be computed in 2 ways:
  (a) from prob: \[ \text{acc} = (\text{prev} \times \text{sens}) + [(1 - \text{prev}) \times \text{spec}] \]
  (b) from freq: \[ \text{acc} = \text{dec\_cor}/N = (\text{hi} + \text{cr})/(\text{hi} + \text{mi} + \text{fa} + \text{cr}) \]
  When frequencies in freq are not rounded, (b) coincides with (a).

• Perspective: acc classifies a population of N individuals by accuracy/correspondence (acc = dec\_cor/N).
  acc is the "by accuracy" or "by correspondence" counterpart to prev (which adopts a "by condition" perspective) and to ppod (which adopts a "by decision" perspective).

• Alternative names of acc: base rate of correct decisions, non-erroneous cases

• In terms of frequencies, acc is the ratio of dec\_cor (i.e., hi + cr) divided by N (i.e., hi + mi + fa + cr):
  \[ \text{acc} = \frac{\text{dec\_cor}}{N} = \frac{(\text{hi} + \text{cr})}{(\text{hi} + \text{mi} + \text{fa} + \text{cr})} \]

• Dependencies: acc is a feature of both the environment (true condition) and of the decision process or diagnostic procedure. It reflects the correspondence of decisions to conditions.

See accu for other accuracy metrics and several possible interpretations of accuracy.

Value

Overall accuracy acc as a probability (proportion). A warning is provided for NaN values.

See acc for definition and accu for other accuracy metrics. comp_accu_freq and comp_accu_prob compute accuracy metrics from frequencies and probabilities.

See Also

acc defines accuracy as a probability; accu lists all accuracy metrics; comp_accu_prob computes exact accuracy metrics from probabilities; comp_accu_freq computes accuracy metrics from frequencies; comp_sens and comp_PPV compute related probabilities; is_extreme_prob_set verifies extreme cases; comp_complement computes a probability’s complement; is_complement verifies probability complements; comp_prob computes current probability information; prob contains current probability information; is_prob verifies probabilities.

Other functions computing probabilities: comp_FDR, comp_FOR, comp_NPV, comp_PPV, comp_accu_freq, comp_accu_prob, comp_comp_pair, comp_complement, comp_complete_prob_set, comp_err, comp_fart, comp_mirt, comp_ppod, comp_prob_freq, comp_prob, comp_sens, comp_spec

Other metrics: accu, acc, comp_accu_freq, comp_accu_prob, comp_err, err
Examples

# ways to work:
comp_acc(.10, .200, .300) # => acc = 0.29
comp_acc(.50, .333, .666) # => acc = 0.4995

# watch out for vectors:
prev.range <- seq(0, 1, by = .1)
comp_acc(prev.range, .5, .5) # => 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5

# watch out for extreme values:
comp_acc(1, 1, 1) # => 1
comp_acc(1, 1, 0) # => 1
comp_acc(1, 0, 1) # => 0
comp_acc(1, 0, 0) # => 0
comp_acc(0, 1, 1) # => 1
comp_acc(0, 1, 0) # => 0
comp_acc(0, 0, 1) # => 1
comp_acc(0, 0, 0) # => 0

comp_accu_freq

Compute accuracy metrics of current classification results.

Description

comp_accu_freq computes a list of current accuracy metrics from the 4 essential frequencies (hi, mi, fa, cr) that constitute the current confusion matrix and are contained in freq.

Usage

comp_accu_freq(hi = freq$hi, mi = freq$mi, fa = freq$fa,
    cr = freq$cr, w = 0.5)

Arguments

  hi               The number of hits hi (or true positives).
  mi               The number of misses mi (or false negatives).
  fa               The number of false alarms fa (or false positives).
  cr               The number of correct rejections cr (or true negatives).
  w                The weighting parameter w (from 0 to 1) for computing weighted accuracy wacc.
                  Default: w = .50 (i.e., yielding balanced accuracy bacc).
Details

Currently computed accuracy metrics include:

1. **acc**: Overall accuracy as the proportion (or probability) of correctly classifying cases or of **dec_cor** cases:
   \[ \text{acc} = \frac{\text{dec_cor}}{N} = \frac{\text{hi} + \text{cr}}{\text{hi} + \text{mi} + \text{fa} + \text{cr}} \]
   Values range from 0 (no correct prediction) to 1 (perfect prediction).

2. **wacc**: Weighted accuracy, as a weighted average of the sensitivity **sens** (aka. hit rate **HR**, **TPR**, **power** or **recall**) and the specificity **spec** (aka. **TNR**) in which **sens** is multiplied by a weighting parameter \( w \) (ranging from 0 to 1) and **spec** is multiplied by \( w \)'s complement (1 - \( w \)):
   \[ \text{wacc} = (w \times \text{sens}) + ((1 - w) \times \text{spec}) \]
   If \( w = 0.5 \), **wacc** becomes balanced accuracy **bacc**.

3. **mcc**: The Matthews correlation coefficient (with values ranging from -1 to +1):
   \[ \text{mcc} = \frac{(\text{hi} \times \text{cr}) - (\text{fa} \times \text{mi})}{\sqrt{((\text{hi} + \text{fa}) \times (\text{hi} + \text{mi}) \times (\text{cr} + \text{fa}) \times (\text{cr} + \text{mi}))}} \]
   A value of \( \text{mcc} = 0 \) implies random performance; \( \text{mcc} = 1 \) implies perfect performance.
   See Wikipedia: Matthews correlation coefficient for additional information.

4. **f1s**: The harmonic mean of the positive predictive value **PPV** (aka. **precision**) and the sensitivity **sens** (aka. hit rate **HR**, **TPR**, **power** or **recall**):
   \[ \text{f1s} = 2 \times \frac{(\text{PPV} \times \text{sens})}{(\text{PPV} + \text{sens})} \]
   See Wikipedia: F1 score for additional information.

Notes:

- Accuracy metrics describe the correspondence of decisions (or predictions) to actual conditions (or truth).

There are several possible interpretations of accuracy:

1. as probabilities (i.e., acc being the proportion of correct classifications, or the ratio **dec_cor**/**N**),
2. as frequencies (e.g., as classifying a population of **N** individuals into cases of **dec_cor** vs. **dec_err**),
3. as correlations (e.g., see **mcc** in **accu**).

- Computing exact accuracy values based on probabilities (by **comp_accu_prob**) may differ from accuracy values computed from (possibly rounded) frequencies (by **comp_accu_freq**). When frequencies are rounded to integers (see the default of **round = TRUE** in **comp_freq** and **comp_freq_prob**) the accuracy metrics computed by **comp_accu_freq** correspond to these rounded values. Use **comp_accu_prob** to obtain exact accuracy metrics from probabilities.

Value

A list **accu** containing current accuracy metrics.

References

Consult Wikipedia: Confusion matrix for additional information.
comp_accu_prob

See Also

accu for all accuracy metrics; comp_accu_prob computes exact accuracy metrics from probabilities; num for basic numeric parameters; freq for current frequency information; txt for current text settings; pal for current color settings; popu for a table of the current population.

Other metrics: accu, acc, comp_accu_prob, comp_acc, comp_err, err

Other functions computing probabilities: comp_FDR, comp_FOR, comp_NPV, comp_PPV, comp_accu_prob, comp_acc, comp_comp_pair, comp_complement, comp_complete_prob_set, comp_err, comp_fart, comp_mirt, comp_ppod, comp_prob_freq, comp_prob, comp_sens, comp_spec

Examples

comp_accu_freq()  # => accuracy metrics for freq of current scenario
comp_accu_freq(hi = 1, mi = 2, fa = 3, cr = 4)  # medium accuracy, but cr > hi

# Extreme cases:
comp_accu_freq(hi = 1, mi = 1, fa = 1, cr = 1)  # random performance
comp_accu_freq(hi = 0, mi = 0, fa = 1, cr = 1)  # random performance: wacc and f1s are NaN
comp_accu_freq(hi = 1, mi = 0, fa = 0, cr = 1)  # perfect accuracy/optimal performance
comp_accu_freq(hi = 0, mi = 1, fa = 1, cr = 0)  # zero accuracy/worst performance, but see f1s
comp_accu_freq(hi = 1, mi = 0, fa = 0, cr = 0)  # perfect accuracy, but see wacc and mcc

# Effects of w:
comp_accu_freq(hi = 3, mi = 2, fa = 1, cr = 4, w = 1/2)  # equal weights to sens and spec
comp_accu_freq(hi = 3, mi = 2, fa = 1, cr = 4, w = 2/3)  # more weight to sens
comp_accu_freq(hi = 3, mi = 2, fa = 1, cr = 4, w = 1/3)  # more weight to spec

## Contrasting comp_accu_freq and comp_accu_prob:
# (a) comp_accu_freq (based on rounded frequencies):
freq1 <- comp_freq(N = 10, prev = 1/3, sens = 2/3, spec = 3/4)  # => hi = 2, mi = 1, fa = 2, cr = 5
accu1 <- comp_accu_freq(freq1$hi, freq1$mi, freq1$fa, freq1$cr)  # => accu1 (based on rounded freq).
# accu1
#
# (b) comp_accu_prob (based on probabilities):
accu2 <- comp_accu_prob(prev = 1/3, sens = 2/3, spec = 3/4)  # => exact accu (based on prob).
# accu2
all.equal(accu1, accu2)  # => 4 differences!
#
# (c) comp_accu_freq (exact values, i.e., without rounding):
freq3 <- comp_freq(N = 10, prev = 1/3, sens = 2/3, spec = 3/4, round = FALSE)
accu3 <- comp_accu_freq(freq3$hi, freq3$mi, freq3$fa, freq3$cr)  # => accu3 (based on EXACT freq).
# accu3
all.equal(accu2, accu3)  # => TRUE (qed).

comp_accu_prob  Compute exact accuracy metrics based on probabilities.
**Description**

`comp_accu_prob` computes a list of exact accuracy metrics from a sufficient and valid set of 3 essential probabilities (`prev`, and `sens` or its complement `mirt`, and `spec` or its complement `fart`).

**Usage**

```r
comp_accu_prob(prev = prob$prev, sens = prob$sens, mirt = NA, spec = prob$spec, fart = NA, tol = 0.01, w = 0.5)
```

**Arguments**

- `prev`: The condition's prevalence `prev` (i.e., the probability of condition being TRUE).
- `sens`: The decision's sensitivity `sens` (i.e., the conditional probability of a positive decision provided that the condition is TRUE). `sens` is optional when its complement `mirt` is provided.
- `mirt`: The decision's miss rate `mirt` (i.e., the conditional probability of a negative decision provided that the condition is TRUE). `mirt` is optional when its complement `sens` is provided.
- `spec`: The decision's specificity value `spec` (i.e., the conditional probability of a negative decision provided that the condition is FALSE). `spec` is optional when its complement `fart` is provided.
- `fart`: The decision's false alarm rate `fart` (i.e., the conditional probability of a positive decision provided that the condition is FALSE). `fart` is optional when its complement `spec` is provided.
- `tol`: A numeric tolerance value for `is_complement`. Default: `tol = .01`.
- `w`: The weighting parameter `w` (from 0 to 1) for computing weighted accuracy `wacc`. Default: `w = .50` (i.e., yielding balanced accuracy `bacc`).

**Notes:**

- Accuracy metrics describe the *correspondence* of decisions (or predictions) to actual conditions (or truth).
  - There are several possible interpretations of accuracy:
    1. as *probabilities* (i.e., `acc` being the proportion of correct classifications, or the ratio `dec.cor/N`),
    2. as *frequencies* (e.g., as classifying a population of `N` individuals into cases of `dec.cor` vs. `dec.err`),
    3. as *correlations* (e.g., see `mcc` in `accu`).
- Computing exact accuracy values based on probabilities (by `comp_accu_prob`) may differ from accuracy values computed from (possibly rounded) frequencies (by `comp_accu_freq`). When frequencies are rounded to integers (see the default of `round = TRUE` in `comp_freq` and `comp_freq_prob`) the accuracy metrics computed by `comp_accu_freq` correspond to these rounded values. Use `comp_accu_prob` to obtain exact accuracy metrics from probabilities.
Details

Currently computed accuracy metrics include:

1. acc: Overall accuracy as the proportion (or probability) of correctly classifying cases or of dec_cor cases:
   (a) from prob: \( \text{acc} = (\text{prev} \times \text{sens}) + [(1 - \text{prev}) \times \text{spec}] \)
   (b) from freq: \( \text{acc} = \frac{\text{dec_cor}}{\text{N}} = \frac{\text{hi} + \text{cr}}{(\text{hi} + \text{mi} + \text{fa} + \text{cr})} \)
   When frequencies in freq are not rounded, (b) coincides with (a).
   Values range from 0 (no correct prediction) to 1 (perfect prediction).

2. wacc: Weighted accuracy, as a weighted average of the sensitivity sens (aka. hit rate HR, TPR, power or recall) and the the specificity spec (aka. TNR) in which sens is multiplied by a weighting parameter \( w \) (ranging from 0 to 1) and spec is multiplied by \( w \)'s complement \( (1 - w) \):
   \( \text{wacc} = (w \times \text{sens}) + ((1 - w) \times \text{spec}) \)
   If \( w = .50 \), wacc becomes balanced accuracy bacc.

3. mcc: The Matthews correlation coefficient (with values ranging from -1 to +1):
   \( \text{mcc} = \frac{(\text{hi} \times \text{cr}) - (\text{fa} \times \text{mi})}{\sqrt{(\text{hi} + \text{fa}) \times (\text{hi} + \text{mi}) \times (\text{cr} + \text{fa}) \times (\text{cr} + \text{mi})}} \)
   A value of \( \text{mcc} = 0 \) implies random performance; \( \text{mcc} = 1 \) implies perfect performance.
   See Wikipedia: Matthews correlation coefficient for additional information.

4. f1s: The harmonic mean of the positive predictive value PPV (aka. precision) and the sensitivity sens (aka. hit rate HR, TPR, power or recall):
   \( \text{f1s} = \frac{2 \times (\text{PPV} \times \text{sens})}{(\text{PPV} + \text{sens})} \)
   See Wikipedia: F1 score for additional information.

Note that some accuracy metrics can be interpreted as probabilities (e.g., acc) and some as correlations (e.g., mcc).

Also, accuracy can be viewed as a probability (e.g., the ratio of or link between dec_cor and N) or as a frequency type (containing dec_cor and dec_err).

comp_accu_prob computes exact accuracy metrics from probabilities. When input frequencies were rounded (see the default of round = TRUE in comp_freq and comp_freq_prob) the accuracy metrics computed by comp_accu correspond these rounded values.

Value

A list accu containing current accuracy metrics.

References

Consult Wikipedia: Confusion matrix for additional information.

See Also

accu for all accuracy metrics; comp_accu_freq computes accuracy metrics from frequencies; num for basic numeric parameters; freq for current frequency information; txt for current text settings; pal for current color settings; popu for a table of the current population.
Other metrics: accu, acc, comp_accu_freq, comp_acc, comp_err, err

Other functions computing probabilities: comp_FDR, comp_FOR, comp_NPV, comp_PPV, comp_accu_freq, comp_acc, comp_comp_pair, comp_complement, comp_complete_prob_set, comp_err, comp_fart, comp_mirt, comp_ppod, comp_prob_freq, comp_prob, comp_sens, comp_spec

Examples

comp_accu_prob() # => accuracy metrics for prob of current scenario
comp_accu_prob(prev = .2, sens = .5, spec = .5) # medium accuracy, but cr > hi.

# Extreme cases:
comp_accu_prob(prev = NaN, sens = NaN, spec = NaN) # returns list of NA values
comp_accu_prob(prev = 0, sens = NaN, spec = 1) # returns list of NA values
comp_accu_prob(prev = .5, sens = .5, spec = .5) # perfect accuracy
comp_accu_prob(prev = .5, sens = 1, spec = 1) # returns list of NA values
comp_accu_prob(prev = .5, sens = 0, spec = 0) # zero accuracy, but f1s is NaN
comp_accu_prob(prev = 1, sens = 1, spec = 0) # perfect, but see wacc (0.5) and mcc (0)

# Effects of w:
comp_accu_prob(prev = .5, sens = .6, spec = .4, w = 1/2) # equal weights to sens and spec
comp_accu_prob(prev = .5, sens = .6, spec = .4, w = 2/3) # more weight on sens: wacc up
comp_accu_prob(prev = .5, sens = .6, spec = .4, w = 1/3) # more weight on spec: wacc down

# Contrasting comp_accu_freq and comp_accu_prob:
# (a) comp_accu_freq (based on rounded frequencies):
freq1 <- comp_freq(N = 10, prev = 1/3, sens = 2/3, spec = 3/4) # => rounded frequencies!
accu1 <- comp_accu_freq(freq1$hi, freq1$mi, freq1$fa, freq1$cr) # => accu1 (based on rounded freq).
# accu1

# (b) comp_accu_prob (based on probabilities):
accu2 <- comp_accu_prob(prev = 1/3, sens = 2/3, spec = 3/4) # => exact accu (based on prob).
# accu2
all.equal(accu1, accu2) # => 4 differences!
#
# (c) comp_accu_freq (exact values, i.e., without rounding):
freq3 <- comp_freq(N = 10, prev = 1/3, sens = 2/3, spec = 3/4, round = FALSE)
accu3 <- comp_accu_freq(freq3$hi, freq3$mi, freq3$fa, freq3$cr) # => accu3 (based on EXACT freq).
# accu3
all.equal(accu2, accu3) # => TRUE (qed).

---

**comp_complement**

Compute a probability's complement probability.

**Description**

comp_complement computes the probability complement of a given probability prob.
Usage

comp_complement(prob)

Arguments

prob A numeric probability value (in range from 0 to 1).

Details

The type and range of prob is verified with is_prob.

Value

A numeric probability value (in range from 0 to 1).

See Also

is_complement verifies numeric complements; comp_comp_pair returns a probability and its complement; is_prob verifies probabilities.

Other functions computing probabilities: comp_FDR, comp_FOR, comp_NPV, comp_PPV, comp_accu_freq, comp_accu_prob, comp_acc, comp_comp_pair, comp_complete_prob_set, comp_err, comp_fart, comp_mirt, comp_ppod, comp_prob_freq, comp_prob, comp_sens, comp_spec

Examples

    comp_complement(0)  # => 1
    comp_complement(1)  # => 0
    comp_complement(2)  # => NA + warning (beyond range)
    comp_complement("p")  # => NA + warning (non-numeric)

comp_complete_prob_set

Compute a complete set of probabilities from valid probability inputs.

Description

comp_complete_prob_set is a function takes a valid set of (3 to 5) probabilities as inputs (as a vector) and returns the complete set of (3 essential and 2 optional) probabilities.

Usage

comp_complete_prob_set(prev, sens = NA, mirt = NA, spec = NA, fart = NA)
Arguments

prev
The condition’s prevalence \( \text{prev} \) (i.e., the probability of condition being TRUE).

sens
The decision’s sensitivity \( \text{sens} \) (i.e., the conditional probability of a positive decision provided that the condition is TRUE). \( \text{sens} \) is optional when its complement \( \text{mirt} \) is provided.

mirt
The decision’s miss rate \( \text{mirt} \) (i.e., the conditional probability of a negative decision provided that the condition is TRUE). \( \text{mirt} \) is optional when its complement \( \text{sens} \) is provided.

spec
The decision’s specificity value \( \text{spec} \) (i.e., the conditional probability of a negative decision provided that the condition is FALSE). \( \text{spec} \) is optional when its complement \( \text{fart} \) is provided.

fart
The decision’s false alarm rate \( \text{fart} \) (i.e., the conditional probability of a positive decision provided that the condition is FALSE). \( \text{fart} \) is optional when its complement \( \text{spec} \) is provided.

Details

Assuming that \( \text{is\_valid\_prob\_set} = \text{TRUE} \) this function uses \text{comp\_comp\_pair} on the two optional pairs (i.e., \( \text{sens} \) and \( \text{mirt} \), and \( \text{spec} \) and \( \text{fart} \)) and returns the complete set of 5 probabilities.

Value

A vector of 5 probabilities: \( c(\text{prev}, \text{sens}, \text{mirt}, \text{spec}, \text{fart}) \).

See Also

\text{is\_valid\_prob\_set} verifies a set of probability inputs; \text{is\_extreme\_prob\_set} verifies extreme cases; \text{comp\_comp\_pair} computes pairs of complements; \text{is\_complement} verifies numeric complements; \text{is\_prob} verifies probabilities; \text{comp\_prob} computes current probability information; \text{prob} contains current probability information; \text{init\_num} initializes basic numeric variables; \text{num} contains basic numeric variables.

Other functions computing probabilities: \text{comp\_FDR, comp\_FOR, comp\_NPV, comp\_PPV, comp\_accu\_freq, comp\_accu\_prob, comp\_acc, comp\_comp\_pair, comp\_complement, comp\_err, comp\_fart, comp\_mirt, comp\_ppod, comp\_prob\_freq, comp\_prob, comp\_sens, comp\_spec}

Examples

# ways to work:
\[
\text{comp\_complete\_prob\_set}(1, .8, \text{NA}, .7, \text{NA}) \# => 1.0 0.8 0.2 0.7 0.3
\text{comp\_complete\_prob\_set}(1, \text{NA}, .8, \text{NA}, .4) \# => 1.0 0.2 0.8 0.6 0.4
\]

# watch out for:
\[
\text{comp\_complete\_prob\_set}(8) \quad \# => 8 \text{ NA NA NA NA} + \text{warnings}
\text{comp\_complete\_prob\_set}(8, 7, 6, 5, 4) \quad \# => 8 7 6 5 4 + \text{no warning (valid set assumed)}
\text{comp\_complete\_prob\_set}(8, .8, \text{NA}, .7, \text{NA}) \# => 8.0 0.8 0.2 0.7 0.3 + \text{no warning (sic)}
\text{comp\_complete\_prob\_set}(8, 2, \text{NA}, 3, \text{NA}) \# => 8 2 \text{ NA NA} + \text{no warning (sic)}
\]
`comp_comp_pair`  

Compute a probability’s (missing) complement and return both.

**Description**

`comp_comp_pair` is a function that takes 0, 1, or 2 probabilities (p1 and p2) as inputs. If either of them is missing (NA), it computes the complement of the other one and returns both probabilities.

**Usage**

```r
comp_comp_pair(p1 = NA, p2 = NA)
```

**Arguments**

- **p1**  
  A numeric probability value (in range from 0 to 1). p1 is optional when p2 is provided.

- **p2**  
  A numeric probability value (in range from 0 to 1). p2 is optional when p1 is provided.

**Details**

`comp_comp_pair` does nothing when both arguments are provided (i.e., `!is.na(p1) & !is.na(p2)`) and only issues a warning if both arguments are missing (i.e., `is.na(p1) & is.na(p2)`).

Inputs are not verified: Use `is_prob` to verify that an input is a probability and `is_complement` to verify that two provided values actually are complements.

**Value**

A vector v containing 2 numeric probability values (in range from 0 to 1): `v = c(p1, p2)`.

**See Also**

- `is_complement` verifies numeric complements; `is_valid_prob_set` verifies sets of probabilities; `comp_complete_prob_set` completes valid sets of probabilities; `is_extreme_prob_set` verifies extreme cases; `comp_prob` computes current probability information; `prob` contains current probability information; `is_prob` verifies probabilities.

Other functions computing probabilities: `comp_FDR`, `comp_FOR`, `comp_NPV`, `comp_PPV`, `comp_accu_freq`, `comp_acceu_prob`, `comp_acc`, `comp_complement`, `comp_complete_prob_set`, `comp_err`, `comp_fart`, `comp_mirt`, `comp_ppod`, `comp_prob_freq`, `comp_prob`, `comp_sens`, `comp_spec`.

**Examples**

```r
# ways to work:
comp_comp_pair(1, 0)  # => 1 0
comp_comp_pair(0, 1)  # => 0 1
comp_comp_pair(1, NA) # => 1 0
comp_comp_pair(NA, 1) # => 0 1
```
comp_err

Compute overall error rate (err) from probabilities.

Description

comp_err computes overall error rate \( err \) from 3 essential probabilities \( \text{prev}, \text{sens}, \text{and spec} \).

Usage

\[
\text{comp_err} \left( \text{prev}, \text{sens}, \text{spec} \right)
\]

Arguments

\begin{itemize}
  \item \texttt{prev} The condition’s prevalence \( \text{prev} \) (i.e., the probability of condition being TRUE).
  \item \texttt{sens} The decision’s sensitivity \( \text{sens} \) (i.e., the conditional probability of a positive decision provided that the condition is TRUE).
  \item \texttt{spec} The decision’s specificity value \( \text{spec} \) (i.e., the conditional probability of a negative decision provided that the condition is FALSE).
\end{itemize}

Details

\[
\text{comp_err} \text{ uses } \text{comp_acc} \text{ to compute } \text{err} \text{ as the complement of } \text{acc}:
\]

\[
\text{err} = 1 - \text{acc}
\]

See \texttt{comp_acc} and \texttt{acc} for further details and \texttt{accu} for other accuracy metrics and several possible interpretations of accuracy.

Value

Overall error rate \( \text{err} \) as a probability (proportion). A warning is provided for NaN values.

See Also

\texttt{comp_acc} computes overall accuracy \( \text{acc} \) from probabilities; \texttt{accu} lists all accuracy metrics; \texttt{comp_accu_prob} computes exact accuracy metrics from probabilities; \texttt{comp_accu_freq} computes accuracy metrics from frequencies; \texttt{comp_sens} and \texttt{comp_PPV} compute related probabilities; \texttt{is_extreme_prob_set} verifies extreme cases; \texttt{comp_complement} computes a probability’s complement; \texttt{is_complement} verifies probability complements; \texttt{comp_prob} computes current probability information; \texttt{prob} contains current probability information; \texttt{is_prob} verifies probabilities.
Other functions computing probabilities: `comp_FDR`, `comp_FOR`, `comp_NPV`, `comp_PPV`, `comp_accu_freq`, `comp_accu_prob`, `comp_acc`, `comp_comp_pair`, `comp_complement`, `comp_complete_prob_set`, `comp_fart`, `comp_mirt`, `comp_ppod`, `comp_prob_freq`, `comp_prob`, `comp_sens`, `comp_spec`

Other metrics: `accu`, `acc`, `comp_accu_freq`, `comp_accu_prob`, `comp_acc`, `err`

Examples

# ways to work:
comp_err(.10, .200, .300) # => err = 0.71
comp_err(.50, .333, .666) # => err = 0.5005

# watch out for vectors:
prev.range <- seq(0, 1, by = .1)
comp_err(prev.range, .5, .5) # => 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5

# watch out for extreme values:
comp_err(1, 1, 1) # => 0
comp_err(1, 1, 0) # => 0

comp_err(1, 0, 1) # => 1
comp_err(1, 0, 0) # => 1

comp_err(0, 1, 1) # => 0
comp_err(0, 1, 0) # => 1

comp_err(0, 0, 1) # => 0
comp_err(0, 0, 0) # => 1

---

`comp_fart`  
*Compute a decision’s false alarm rate from its specificity.*

Description

`comp_fart` is a conversion function that takes a specificity `spec` – given as a probability (i.e., a numeric value in the range from 0 to 1) – as its input, and returns the corresponding false alarm rate `fart` – also as a probability – as its output.

Usage

`comp_fart(spec)`

Arguments

`spec`  
The decision’s specificity value `spec` as a probability.
Details
The false alarm rate \( \text{fart} \) and specificity \( \text{spec} \) are complements (\( \text{fart} = (1 - \text{spec}) \)) and both features of the decision process (e.g., a diagnostic test).

The function \( \text{comp}_\text{fart} \) is complementary to the conversion function \( \text{comp}_\text{spec} \) and uses the generic function \( \text{comp}_\text{complement} \).

Value
The decision’s false alarm rate \( \text{fart} \) as a probability.

See Also
\( \text{comp}_\text{complement} \) computes a probability’s complement; \( \text{is}_\text{complement} \) verifies probability complements; \( \text{comp}_\text{prob} \) computes current probability information; \( \text{prob} \) contains current probability information; \( \text{is}_\text{prob} \) verifies probabilities.

Other functions computing probabilities: \( \text{comp}_\text{FDR}, \text{comp}_\text{FOR}, \text{comp}_\text{NPV}, \text{comp}_\text{PPV}, \text{comp}_\text{accu_freq}, \text{comp}_\text{accu_prob}, \text{comp}_\text{acc}, \text{comp}_\text{comp_pair}, \text{comp}_\text{complement}, \text{comp}_\text{complete_prob_set}, \text{comp}_\text{err}, \text{comp}_\text{mirt}, \text{comp}_\text{ppod}, \text{comp}_\text{prob_freq}, \text{comp}_\text{prob}, \text{comp}_\text{sens}, \text{comp}_\text{spec} \)

Examples
\[
\begin{align*}
\text{comp}_\text{fart}(2) & \quad \# \Rightarrow \text{NA + warning (beyond range)} \\
\text{comp}_\text{fart}(1/3) & \quad \# \Rightarrow 0.6666667 \\
\text{comp}_\text{fart}(\text{comp}_\text{complement}(0.123)) & \quad \# \Rightarrow 0.123 \\
\end{align*}
\]

\( \text{comp}_\text{FDR} \)
Compute a decision’s false detection rate (FDR) from probabilities.

Description
\( \text{comp}_\text{FDR} \) computes the false detection rate \( \text{FDR} \) from 3 essential probabilities \( \text{prev}, \text{sens}, \) and \( \text{spec} \).

Usage
\( \text{comp}_\text{FDR}(\text{prev}, \text{sens}, \text{spec}) \)

Arguments
\begin{itemize}
\item \texttt{prev} \quad The condition’s prevalence \( \text{prev} \) (i.e., the probability of condition being TRUE).
\item \texttt{sens} \quad The decision’s sensitivity \( \text{sens} \) (i.e., the conditional probability of a positive decision provided that the condition is TRUE).
\item \texttt{spec} \quad The decision’s specificity value \( \text{spec} \) (i.e., the conditional probability of a negative decision provided that the condition is FALSE).
\end{itemize}
**comp_FOR**

Details

comp_FDR uses probabilities (not frequencies) and does not round results.

Value

The false detection rate FDR as a probability. A warning is provided for NaN values.

See Also

comp_sens and comp_PPV compute related probabilities; is_extreme_prob_set verifies extreme cases; comp_complement computes a probability's complement; is_complement verifies probability complements; comp_prob computes current probability information; prob contains current probability information; is_prob verifies probabilities.

Other functions computing probabilities: comp_FOR, comp_NPV, comp_PPV, comp_accu_freq, comp_accu_prob, comp_acc, comp_comp_pair, comp_complement, comp_complete_prob_set, comp_err, comp_fart, comp_mirt, comp_ppod, comp_prob_freq, comp_prob, comp_sens, comp_spec

Examples

```r
# (1) Ways to work:
comp_FDR(.50, .500, .500) # => FDR = 0.5 = (1 - PPV)
comp_FDR(.50, .333, .666) # => FDR = 0.5007 = (1 - PPV)
```

---

**Description**

comp_FOR computes the false omission rate FOR from 3 essential probabilities prev, sens, and spec.

**Usage**

comp_FOR(prev, sens, spec)

**Arguments**

- **prev**: The condition’s prevalence prev (i.e., the probability of condition being TRUE).
- **sens**: The decision’s sensitivity sens (i.e., the conditional probability of a positive decision provided that the condition is TRUE).
- **spec**: The decision’s specificity value spec (i.e., the conditional probability of a negative decision provided that the condition is FALSE).

**Details**

comp_FOR uses probabilities (not frequencies) and does not round results.
Value

The false omission rate FOR as a probability. A warning is provided for NaN values.

See Also

comp_spec and comp_NPV compute related probabilities; is_extreme_prob_set verifies extreme cases; comp_complement computes a probability’s complement; is_complement verifies probability complements; comp_prob computes current probability information; prob contains current probability information; is_prob verifies probabilities.

Other functions computing probabilities: comp_FDR, comp_NPV, comp_PPV, comp_accu_freq, comp_accu_prob, comp_acc, comp_comp_pair, comp_complement, comp_complete_prob_set, comp_err, comp_fart, comp_mirt, comp_ppod, comp_prob_freq, comp_prob, comp_sens, comp_spec

Examples

C H1I ways to work:
comp_FORWARD NSSSL NVVVI C ]^ for ] PNUPPT ] H1 M npvI
comp_freq

Compute frequencies from (3 essential) probabilities.

Description

comp_freq computes frequencies (typically as rounded integers) given 3 basic probabilities – prev, sens, and spec – for a population of N individuals. It returns a list of 11 frequencies freq as its output.

Usage

comp_freq(prev = num$prev, sens = num$sens, spec = num$spec, N = num$N, round = TRUE)

Arguments

prev The condition’s prevalence prev (i.e., the probability of condition being TRUE).
sens The decision’s sensitivity sens (i.e., the conditional probability of a positive decision provided that the condition is TRUE).
spec The decision’s specificity value spec (i.e., the conditional probability of a negative decision provided that the condition is FALSE).
N The number of individuals in the population. If N is unknown (NA), a suitable minimum value is computed by comp_min_N.
round Boolean value that determines whether frequencies are rounded to the nearest integer. Default: round = TRUE.
Note: Removed n_digits parameter: Number of digits to which frequency values are to be rounded when round = FALSE. Default: n_digits = 5.
Details

In addition to `prev`, both `sens` and `spec` are necessary arguments. If only their complements `mirt` or `fart` are known, use the wrapper function `comp_freq_prob` which also accepts `mirt` and `fart` as inputs (but requires that the entire set of provided probabilities is sufficient and consistent). Alternatively, use `comp_complement`, `comp_comp_pair`, or `comp_complete_prob_set` to obtain the 3 essential probabilities.

`comp_freq` is the frequency counterpart to the probability function `comp_prob`.

By default, `comp_freq` and its wrapper function `comp_freq_prob` round frequencies to nearest integers to avoid decimal values in `freq` (i.e., `round = TRUE` by default). When frequencies are rounded, probabilities computed from `freq` may differ from exact probabilities. Using the option `round = FALSE` turns off rounding.

Key relationships between probabilities and frequencies:

- Three perspectives on a population:
  A population of `N` individuals can be split into 2 subsets of frequencies in 3 different ways:
  1. by condition:
     \[ N = \text{cond_true} + \text{cond_false} \]
     The frequency `cond_true` depends on the prevalence `prev` and the frequency `cond_false` depends on the prevalence’s complement `1 - prev`.
  2. by decision:
     \[ N = \text{dec_pos} + \text{dec_neg} \]
     The frequency `dec_pos` depends on the proportion of positive decisions `ppod` and the frequency `dec_neg` depends on the proportion of negative decisions `1 - ppod`.
  3. by accuracy (i.e., correspondence of decision to condition):
     \[ N = \text{dec_cor} + \text{dec_err} \]
     Each perspective combines 2 pairs of the 4 essential probabilities (hi, mi, fa, cr).

When providing probabilities, the population size `N` is a free parameter (independent of the essential probabilities `prev`, `sens`, and `spec`).

If `N` is unknown (NA), a suitable minimum value can be computed by `comp_min_N`.

- Defining probabilities in terms of frequencies:
  Probabilities are – determine, describe, or are defined as – the relationships between frequencies. Thus, they can be computed as ratios between frequencies:
  1. prevalence `prev`:
     \[ \text{prev} = \text{cond_true}/N = (\text{hi} + \text{mi}) / (\text{hi} + \text{mi} + \text{fa} + \text{cr}) \]
  2. sensitivity `sens`:
     \[ \text{sens} = \text{hi}/\text{cond_true} = \text{hi} / (\text{hi} + \text{mi}) = (1 - \text{mirt}) \]
  3. miss rate `mirt`:
     \[ \text{mirt} = \text{mi}/\text{cond_true} = \text{mi} / (\text{hi} + \text{mi}) = (1 - \text{sens}) \]
  4. specificity `spec`:
     \[ \text{spec} = \text{cr}/\text{cond_false} = \text{cr} / (\text{fa} + \text{cr}) = (1 - \text{fart}) \]
  5. false alarm rate `fart`:
     \[ \text{fart} = \text{fa}/\text{cond_false} = \text{fa} / (\text{fa} + \text{cr}) = (1 - \text{spec}) \]
  6. proportion of positive decisions `ppod`:
     \[ \text{ppod} = \text{dec_pos}/N = (\text{hi} + \text{fa}) / (\text{hi} + \text{mi} + \text{fa} + \text{cr}) \]
7. positive predictive value PPV:
   \[ PPV = \frac{hi}{dec_pos} = \frac{hi}{hi + fa} = (1 - FDR) \]
8. negative predictive value NPV:
   \[ NPV = \frac{cr}{dec_neg} = \frac{cr}{mi + cr} = (1 - FOR) \]
9. false detection rate FDR:
   \[ FDR = \frac{fa}{dec_pos} = \frac{fa}{hi + fa} = (1 - PPV) \]
10. false omission rate FOR:
    \[ FOR = \frac{mi}{dec_neg} = \frac{mi}{mi + cr} = (1 - NPV) \]
11. accuracy acc:
    \[ acc = \frac{dec_cor}{N} = \frac{hi + cr}{hi + mi + fa + cr} \]

Note: When frequencies are rounded (by round = TRUE in comp_freq), probabilities computed from freq may differ from exact probabilities.

Functions translating between representational formats: comp_prob_prob, comp_prob_freq, comp_freq_prob, comp_freq_freq (see documentation of comp_prob_prob for details).

Value
A list freq containing 11 frequency values.

See Also

comp_freq_prob corresponding wrapper function; num contains basic numeric variables; init_num initializes basic numeric variables; freq contains current frequency information; prob contains current probability information; comp_prob computes current probability information; comp_complement computes a probability's complement; comp_comp_pair computes pairs of complements; comp_complete_prob_set completes valid sets of probabilities; comp_min_N computes a suitable population size N (if missing).

Other functions computing frequencies: comp_freq_freq, comp_freq_prob, comp_min_N, comp_popu, comp_prob_prob

Examples

```r
comp_freq() # => ok, using current defaults
length(comp_freq()) # => 11

# Rounding effects:
comp_freq(prev = .5, sens = .5, spec = .5, N = 1) # => yields fa = 1 (see ?round for reason)
comp_freq(prev = .1, sens = .9, spec = .8, N = 10) # => 1 hit (TP, rounded)
comp_freq(prev = .1, sens = .9, spec = .8, N = 10, round = FALSE) # => hi = .9
comp_freq(prev = 1/3, sens = 6/7, spec = 2/3, N = 1, round = FALSE) # => hi = 0.2857143

# Extreme cases:
comp_freq(prev = 1, sens = 1, spec = 1, N = 100) # => ok, N hits (TP)
comp_freq(prev = 1, sens = 1, spec = 0, N = 100) # => ok, N hits
comp_freq(prev = 1, sens = 0, spec = 1, N = 100) # => ok, N misses (FN)
comp_freq(prev = 1, sens = 0, spec = 0, N = 100) # => ok, N misses
comp_freq(prev = 0, sens = 1, spec = 1, N = 100) # => ok, N correct rejections (TN)
comp_freq(prev = 0, sens = 1, spec = 0, N = 100) # => ok, N false alarms (FP)
```
# Watch out for:
comp_freq(prev = 1, sens = 1, spec = 1, N = NA) # => ok, but warning that N = 1 was computed
comp_freq(prev = 1, sens = 1, spec = 1, N = 0) # => ok, but all 0 + warning (extreme case: N hits)
comp_freq(prev = .5, sens = .5, spec = .5, N = 10, round = TRUE) # => ok, rounded (see mi and fa)
comp_freq(prev = .5, sens = .5, spec = .5, N = 10, round = FALSE) # => ok, not rounded

# Ways to fail:
comp_freq(prev = NA, sens = 1, spec = 1, 100) # => NAs + warning (prev NA)
comp_freq(prev = 1, sens = NA, spec = 1, 100) # => NAs + warning (sens NA)
comp_freq(prev = 1, sens = 1, spec = NA, 100) # => NAs + warning (spec NA)
comp_freq(prev = 8, sens = 1, spec = 1, 100) # => NAs + warning (prev beyond range)
comp_freq(prev = 1, sens = 8, spec = 1, 100) # => NAs + warning (sens beyond range)

---

**comp_freq_freq**  
Compute frequencies from (4 essential) frequencies.

### Description

*comp_freq_freq* computes current frequency information from 4 essential frequencies (hi, mi, fa, cr). It returns a list of 11 frequencies freq for a population of N individuals as its output.

### Usage

```
comp_freq_freq(hi = freq$hi, mi = freq$mi, fa = freq$fa,
               cr = freq$cr)
```

### Arguments

- **hi**: The number of hits hi (or true positives).
- **mi**: The number of misses mi (or false negatives).
- **fa**: The number of false alarms fa (or false positives).
- **cr**: The number of correct rejections cr (or true negatives).

### Details

Key relationships between frequencies and probabilities (see documentation of *comp_freq* or *comp_prob* for details):

- Three perspectives on a population:
  - by condition / by decision / by accuracy.
- Defining probabilities in terms of frequencies:
  - Probabilities can be computed as ratios between frequencies, but beware of rounding issues.

Functions translating between representational formats: *comp_prob_prob*, *comp_prob_freq*, *comp_freq_prob*, *comp_freq_freq* (see documentation of *comp_prob_prob* for details).
**See Also**

*comp_freq_prob* computes current frequency information from (3 essential) probabilities; *comp_prob_freq* computes current probability information from (4 essential) frequencies; *comp_prob_prob* computes current probability information from (3 essential) probabilities; *num* contains basic numeric parameters; *init_num* initializes basic numeric parameters; *prob* contains current probability information; *comp_prob* computes current probability information; *freq* contains current frequency information; *comp_freq* computes current frequency information; *is_prob* verifies probability inputs; *is_freq* verifies frequency inputs.

Other functions computing frequencies: *comp_freq_prob, comp_freq, comp_min_N, comp_popu, comp_prob_prob*

Other format conversion functions: *comp_freq_prob, comp_prob_freq, comp_prob_prob*

**Examples**

```r
# Basics:
comp_freq_freqHI
all.equal(freq, comp_freq_freqHII) # => should be TRUE

# Circular chain:
# 1. Current numeric parameters:
um

# 2. Compute all 10 probabilities in prob (from essential probabilities):
prob <- comp_prob()
prob

# 3. Compute 9 frequencies in freq from probabilities:
freq <- comp_freq(round = FALSE) # no rounding (to obtain same probabilities later)
freq

# 4. Compute 9 frequencies AGAIN (but now from frequencies):
freq_freq <- comp_freq_freq()

# 5. Check equality of results (steps 2. and 4.):
all.equal(freq, freq_freq) # => should be TRUE!
```

---

**comp_freq_prob**  
Compute frequencies from (3 essential) probabilities.

**Description**

*comp_freq_prob* computes current frequency information from a sufficient and valid set of 3 essential probabilities (*prev*, and *sens* or its complement *mirt*, and *spec* or its complement *fart*). It returns a list of 11 frequencies (*freq*) as its output.
Usage

```r
comp_freq_prob(prev = prob$prev, sens = prob$sens, mirt = NA,
    spec = prob$spec, fart = NA, tol = 0.01, N = freq$N,
    round = TRUE)
```

Arguments

- **prev**: The condition’s prevalence `prev` (i.e., the probability of condition being `TRUE`).
- **sens**: The decision’s sensitivity `sens` (i.e., the conditional probability of a positive decision provided that the condition is `TRUE`). `sens` is optional when its complement `mirt` is provided.
- **mirt**: The decision’s miss rate `mirt` (i.e., the conditional probability of a negative decision provided that the condition is `TRUE`). `mirt` is optional when its complement `sens` is provided.
- **spec**: The decision’s specificity value `spec` (i.e., the conditional probability of a negative decision provided that the condition is `FALSE`). `spec` is optional when its complement `fart` is provided.
- **fart**: The decision’s false alarm rate `fart` (i.e., the conditional probability of a positive decision provided that the condition is `FALSE`). `fart` is optional when its complement `spec` is provided.
- **tol**: A numeric tolerance value for `is_complement`. Default: `tol = .01`.
- **N**: The number of individuals in the population. If `N` is unknown (NA), a suitable minimum value is computed by `comp_min_N`.
- **round**: A Boolean value that determines whether frequencies are rounded to the nearest integer. Default: `round = TRUE`.

Details

`comp_freq_prob` is a wrapper function for the more basic function `comp_freq`, which only accepts 3 essential probabilities (i.e., `prev`, `sens`, and `spec`) as inputs.

Defaults and constraints:

- **Initial values:**
  By default, the values of `prev`, `sens`, and `spec` are initialized to the probability information currently contained in `prob`.
  Similarly, the population size `N` uses the frequency information currently contained in `freq` as its default. If `N` is unknown (NA), a suitable minimum value is computed by `comp_min_N`.
- **Constraints:**
  When using `comp_freq_prob` with the arguments `mirt` and `fart`, their complements `sens` and `spec` must either be valid complements (as in `is_complement`) or set to NA.
  In addition to `prev`, both `sens` and `spec` are necessary arguments. If only their complements `mirt` or `fart` are known, first use `comp_complement`, `comp_comp_pair`, or `comp_complete_prob_set` to compute the 3 essential probabilities.
• Rounding:
  By default, \texttt{comp_freq_prob} and its basic function \texttt{comp_freq} round frequencies to nearest integers to avoid decimal values in \texttt{freq} (i.e., \texttt{round = TRUE} by default).
  When frequencies are rounded, probabilities computed from \texttt{freq} may differ from exact probabilities.
  Using the option \texttt{round = FALSE} turns off rounding.

Key relationships between frequencies and probabilities (see documentation of \texttt{comp_freq} or \texttt{comp_prob} for details):

• Three perspectives on a population:
  by condition / by decision / by accuracy.
• Defining probabilities in terms of frequencies:
  Probabilities can be computed as ratios between frequencies, but beware of rounding issues.

Functions translating between representational formats: \texttt{comp_prob_prob}, \texttt{comp_prob_freq}, \texttt{comp_freq_prob}, \texttt{comp_freq_freq} (see documentation of \texttt{comp_prob_prob} for details).

Value

A list \texttt{freq} containing 11 frequency values.

See Also

\texttt{comp_freq_freq} computes current frequency information from (4 essential) frequencies; \texttt{comp_prob_freq} computes current probability information from (4 essential) frequencies; \texttt{comp_prob_prob} computes current probability information from (3 essential) probabilities; \texttt{num} contains basic numeric variables; \texttt{init_num} initializes basic numeric variables; \texttt{freq} contains current frequency information; \texttt{comp_freq} computes current frequency information; \texttt{prob} contains current probability information; \texttt{comp_prob} computes current probability information; \texttt{comp_complement} computes a probability’s complement; \texttt{comp_comp_pair} computes pairs of complements; \texttt{comp_complete_prob_set} completes valid sets of probabilities; \texttt{comp_min_N} computes a suitable population size \texttt{N} (if missing).

Other functions computing frequencies: \texttt{comp_freq_freq}, \texttt{comp_freq}, \texttt{comp_min_N}, \texttt{comp_popu}, \texttt{comp_prob_prob}

Other format conversion functions: \texttt{comp_freq_freq}, \texttt{comp_prob_freq}, \texttt{comp_prob_prob}

Examples

\begin{verbatim}
# Basics:
comp_freq_prob(prev = .1, sens = .9, spec = .8, N = 100) # => ok: hi = 9, ... cr = 72.
# Same case with complements (using NAs to prevent defaults):
comp_freq_prob(prev = .1, sens = NA, mirt = .1, spec = NA, fart = .2, N = 100) # => same result

comp_freq_prob() # => ok, using probability info currently contained in prob
length(comp_freq_prob()) # => a list containing 9 frequencies
all.equal(freq, comp_freq_prob()) # => TRUE, unless prob has been changed after computing freq
freq <- comp_freq_prob() # => computes frequencies and stores them in freq
\end{verbatim}
# Ways to work:
comp_freq_prob(prev = 1, sens = 1, spec = 1, N = 101) # => ok + warning: N hits (TP)

# Same case with complements (using NAs to prevent defaults):
comp_freq_prob(prev = 1, sens = NA, mirt = 0, spec = NA, fart = 0, N = 101)

comp_freq_prob(prev = 1, sens = 1, spec = 0, N = 102) # => ok + warning: N hits (TP)
comp_freq_prob(prev = 1, sens = 0, spec = 1, N = 103) # => ok + warning: N misses (FN)
comp_freq_prob(prev = 1, sens = 0, spec = 0, N = 104) # => ok + warning: N misses (FN)
comp_freq_prob(prev = 0, sens = 1, spec = 1, N = 105) # => ok + warning: N correct rejections (TN)
comp_freq_prob(prev = 0, sens = 1, spec = 0, N = 106) # => ok + warning: N false alarms (FP)

# Same case with complements (using NAs to prevent defaults):
comp_freq_prob(prev = 0, sens = NA, mirt = 0,
              spec = NA, fart = 1, N = 106) # => ok + warning: N false alarms (FP)

# Watch out for:
comp_freq_prob(prev = 1, sens = 1, spec = 1, N = NA) # => ok + warning: N = 1 computed
comp_freq_prob(prev = 1, sens = 1, spec = 1, N = 0) # => ok, but all 0 + warning (NPV = NaN)
comp_freq_prob(prev = .5, sens = .5, spec = .5, N = 10, round = TRUE) # => ok, but all rounded
comp_freq_prob(prev = .5, sens = .5, spec = .5, N = 10, round = FALSE) # => ok, but not rounded

# Ways to fail:
comp_freq_prob(prev = NA, sens = 1, spec = 1, 100) # => NAs + no warning (prev NA)
comp_freq_prob(prev = 1, sens = NA, spec = 1, 100) # => NAs + no warning (sens NA)
comp_freq_prob(prev = 1, sens = 1, spec = NA, 100) # => NAs + no warning (spec NA)
comp_freq_prob(prev = 8, sens = 1, spec = 1, 100) # => NAs + warning (prev beyond range)
comp_freq_prob(prev = 1, sens = 8, spec = 1, 100) # => NAs + warning (sens & spec beyond range)

---

**comp_min_n**

*Compute a suitable minimum population size value N.*

**Description**

`comp_min_n` computes a population size value N (an integer as a power of 10) so that the frequencies of the 4 combinations of conditions and decisions (i.e., the cells of the confusion table, or center row of boxes in the frequency prism) reach or exceed a minimum value `min_freq` given the basic parameters `prev`, `sens`, and `spec` (`spec = 1 - fart`).

**Usage**

`comp_min_n(prev, sens, spec, min_freq = 1)`

**Arguments**

`prev` The condition’s prevalence value `prev` (i.e., the probability of condition being TRUE).
sens  The decision’s sensitivity value sens (i.e., the conditional probability of a positive decision provided that the condition is TRUE).

spec  The specificity value spec (i.e., the conditional probability of a negative decision provided that the condition is FALSE).

min_freq  The minimum frequency of each combination of a condition and a decision (i.e., hits, misses, false alarms, and correct rejections). Default: min_freq = 1.

Details

Using this function helps avoiding excessively small decimal values in categories – especially hi, mi, fa, cr – when expressing combinations of conditions and decisions as natural frequencies. As values of zero (0) are tolerable, the function only increases \( N \) (in powers of 10) while the current value of any frequency (cell in confusion table or leaf of a frequency tree) is positive but below min_freq.

By default, `comp_freq_prob` and `comp_freq` round frequencies to nearest integers to avoid decimal values in freq (i.e., round = TRUE by default). Using the option round = FALSE turns off rounding.

Value

An integer value \( N \) (as a power of 10).

See Also

population size \( N \); `num` contains basic numeric parameters; `freq` contains current frequency information; `comp_freq` computes frequencies from probabilities; `prob` contains current probability information; `comp_prob` computes probabilities from probabilities; `comp_freq_freq` computes current frequency information from (4 essential) frequencies; `comp_freq_prob` computes current frequency information from (3 essential) probabilities; `comp_prob_freq` computes current probability information from (4 essential) frequencies; `comp_prob_prob` computes current probability information from (3 essential) probabilities.

Other functions computing frequencies: `comp_freq_freq`, `comp_freq_prob`, `comp_freq`, `comp_popu`, `comp_prob_prob`

Examples

```r
comp_min_N(0, 0, 0) # => 1
comp_min_N(1, 1, 1) # => 1

comp_min_N(1, 1, 1, min_freq = 10) # => 10
comp_min_N(1, 1, 1, min_freq = 99) # => 100

comp_min_N(.1, .1, .1) # => 100 = 10^2
comp_min_N(.001, .1, .1) # => 10 000 = 10^4
comp_min_N(.001, .001, .1) # => 1 000 000 = 10^6
comp_min_N(.001, .001, .001) # => 1 000 000 = 10^6
```
**comp_mirt**

*Compute a decision’s miss rate from its sensitivity.*

**Description**

`comp_mirt` is a conversion function that takes a sensitivity `sens` – given as a probability (i.e., a numeric value in the range from 0 to 1) – as its input, and returns the corresponding miss rate `mirt` – also as a probability – as its output.

**Usage**

`comp_mirt(sens)`

**Arguments**

`sens`  
The decision’s sensitivity `sens` as a probability.

**Details**

The miss rate `mirt` and sensitivity `sens` are complements (`mirt = 1 - sens`) and both features of the decision process (e.g., a diagnostic test).

The function `comp_mirt` is complementary to the conversion function `comp_sens` and uses the generic function `comp_complement`.

**Value**

The decision’s miss rate `mirt` as a probability.

**See Also**

- `comp_complement` computes a probability’s complement; `is_complement` verifies probability complements; `comp_prob` computes current probability information; `prob` contains current probability information; `is_prob` verifies probabilities.

Other functions computing probabilities: `comp_FDR, comp_FOR, comp_NPV, comp_PPV, comp_accu_freq, comp_accu_prob, comp_acc, comp_comp_pair, comp_complement, comp_complete_prob_set, comp_err, comp_fart, comp_ppod, comp_prob_freq, comp_prob, comp_sens, comp_spec`

**Examples**

```r
comp_mirt(2)  # => NA + warning (beyond range)
comp_mirt(1/3) # => 0.6666667
comp_mirt(comp_complement(0.123)) # => 0.123
```
**comp_NPV**

*Compute a decision’s negative predictive value (NPV) from probabilities.*

**Description**

`comp_NPV` computes the negative predictive value `NPV` from 3 essential probabilities `prev`, `sens`, and `spec`.

**Usage**

```r
comp_NPV(prev, sens, spec)
```

**Arguments**

- `prev` The condition’s prevalence `prev` (i.e., the probability of condition being TRUE).
- `sens` The decision’s sensitivity `sens` (i.e., the conditional probability of a positive decision provided that the condition is TRUE).
- `spec` The decision’s specificity value `spec` (i.e., the conditional probability of a negative decision provided that the condition is FALSE).

**Details**

`comp_NPV` uses probabilities (not frequencies) and does not round results.

**Value**

The negative predictive value `NPV` as a probability. A warning is provided for NaN values.

**See Also**

- `comp_spec` and `comp_PPV` compute related probabilities;
- `is_extreme_prob_set` verifies extreme cases;
- `comp_complement` computes a probability’s complement;
- `is_complement` verifies probability complements;
- `comp_prob` computes current probability information;
- `prob` contains current probability information;
- `is_prob` verifies probabilities.

Other functions computing probabilities: `comp_FDR`, `comp_FOR`, `comp_PPV`, `comp_accu_freq`, `comp_accu_prob`, `comp_acc`, `comp_com_pair`, `comp_complement`, `comp_complete_prob_set`, `comp_err`, `comp_fart`, `comp_mirt`, `comp_ppod`, `comp_prob_freq`, `comp_prob`, `comp_sens`, `comp_spec`

**Examples**

```r
# (1) Ways to work:
comp_NPV(.50, .500, .500) # => NPV = 0.5
comp_NPV(.50, .333, .666) # => NPV = 0.4996

# (2) Watch out for vectors:
prev <- seq(0, 1, .1)
comp_NPV(prev, .5, .5) # => without NaN values
```
comp_popu

Compute a population table from frequencies.

**Description**

`comp_popu` is a function that computes a table `popu` (as an R data frame) from the current frequency information (contained in `freq`).

**Usage**

```r
comp_popu(hi = freq$hi, mi = freq$mi, fa = freq$fa, cr = freq$cr,
           cond_lbl = txt$condLbl, cond_true_lbl = txt$cond_true_lbl,
           cond_false_lbl = txt$cond_false_lbl, dec_lbl = txt$dec_lbl,
           dec_pos_lbl = txt$dec_pos_lbl, dec_neg_lbl = txt$dec_neg_lbl,
           sdt_lbl = txt$sdt_lbl, hi_lbl = txt$hi_lbl, mi_lbl = txt$mi_lbl,
           fa_lbl = txt$fa_lbl, cr_lbl = txt$cr_lbl)
```

**Arguments**

- **hi**: The number of hits `hi` (or true positives).
- **mi**: The number of misses `mi` (or false negatives).
- **fa**: The number of false alarms `fa` (or false positives).
- **cr**: The number of correct rejections `cr` (or true negatives).
- **cond_lbl**: Text label for condition dimension ("by cd" perspective).
- **cond_true_lbl**: Text label for `cond_true` cases.
- **cond_false_lbl**: Text label for `cond_false` cases.
- **dec_lbl**: Text label for decision dimension ("by dc" perspective).
- **dec_pos_lbl**: Text label for `dec_pos` cases.
- **dec_neg_lbl**: Text label for `dec_neg` cases.
- **sdt_lbl**: Text label for 4 cases/combinations (SDT classifications).
- **hi_lbl**: Text label for `hi` cases.
- **mi_lbl**: Text label for `mi` cases.
- **fa_lbl**: Text label for `fa` cases.
- **cr_lbl**: Text label for `cr` cases.
**Format**

An object of class `data.frame` with \( N \) rows and 3 columns ("Truth", "Decision", "SDT").

**Details**

`comp_popu` also uses the current text settings contained in `txt`. A visualization of the current population contained in `popu` is provided by `plot_icon`.

**Value**

A data frame `popu` containing \( N \) rows (individual cases) and 3 columns ("Truth", "Decision", "SDT") encoded as ordered factors (with 2, 2, and 4 levels, respectively).

**See Also**

the corresponding data frame `popu`; `read_popu` interprets a data frame as a `riskyr` scenario; `num` for basic numeric parameters; `freq` for current frequency information; `txt` for current text settings; `pal` for current color settings.

Other functions computing frequencies: `comp_freq_freq`, `comp_freq_prob`, `comp_freq`, `comp_min_N`, `comp_prob_prob`

**Examples**

```r
popu <- comp_popu()  # => initializes popu (with current values of freq and txt)
dim(popu)             # => N x 3
head(popu)

# (A) Diagnostic/screening scenario (using default labels):
comp_popu(hi = 4, mi = 1, fa = 2, cr = 3)  # => computes a table of \( N = 10 \) cases.

# (B) Intervention/treatment scenario:
comp_popu(hi = 3, mi = 2, fa = 1, cr = 4,
          cond_lbl = "Treatment", cond_true_lbl = "pill", cond_false_lbl = "placebo",
          dec_lbl = "Health status", dec_pos_lbl = "healthy", dec_neg_lbl = "sick")

# (C) Prevention scenario (e.g., vaccination):
comp_popu(hi = 3, mi = 2, fa = 1, cr = 4,
          cond_lbl = "Vaccination", cond_true_lbl = "yes", cond_false_lbl = "no",
          dec_lbl = "Disease", dec_pos_lbl = "no flu", dec_neg_lbl = "flu")
```

---

**comp_ppod**

Compute the proportion of positive decisions (\( ppod \)) from probabilities.

**Description**

`comp_ppod` computes the proportion of positive decisions `ppod` from 3 essential probabilities `prev`, `sens`, and `spec`.
**comp_ppod**

**Usage**

```r
comp_ppod(prev, sens, spec)
```

**Arguments**

- `prev` The condition’s prevalence `prev` (i.e., the probability of condition being TRUE).
- `sens` The decision’s sensitivity `sens` (i.e., the conditional probability of a positive decision provided that the condition is TRUE).
- `spec` The decision’s specificity value `spec` (i.e., the conditional probability of a negative decision provided that the condition is FALSE).

**Details**

`comp_ppod` uses probabilities (not frequencies) as inputs and returns a proportion (probability) without rounding.

**Definition:** `ppod` is proportion (or probability) of positive decisions:

```latex
ppod = \frac{\text{dec_pos}}{N} = \frac{(hi + fa)}{(hi + mi + fa + cr)}
```

Values range from 0 (only negative decisions) to 1 (only positive decisions).

Importantly, positive decisions `dec_pos` are not necessarily correct decisions `dec_cor`.

**Value**

The proportion of positive decisions `ppod` as a probability. A warning is provided for NaN values.

**See Also**

- `comp_sens` and `comp_NPV` compute related probabilities; `is_extreme_prob_set` verifies extreme cases; `comp_complement` computes a probability’s complement; `is_complement` verifies probability complements; `comp_prob` computes current probability information; `prob` contains current probability information; `is_prob` verifies probabilities.

Other functions computing probabilities: `comp_FDR`, `comp_FOR`, `comp_NPV`, `comp_PPV`, `comp_accur_freq`, `comp_accu_prob`, `comp_acc`, `comp_comp_pair`, `comp_complement`, `comp_complete_prob_set`, `comp_err`, `comp_fart`, `comp_mirt`, `comp_prob_freq`, `comp_prob`, `comp_sens`, `comp_spec`

**Examples**

# (1) ways to work:
```r
comp_ppod(.10, .200, .300) # => ppod = 0.65
comp_ppod(.50, .333, .666) # => ppod = 0.3335
```

# (2) watch out for vectors:
```r
prev <- seq(.00, .1)
comp_ppod(prev, .50, .5)  # => 0.50 0.53 0.56 0.59 0.62 0.65 0.68 0.71 0.74 0.77 0.80
comp_ppod(prev, , 1.0)  # => 0 0 0 0 0 0 0 0 0 1 1
```

# (3) watch out for extreme values:
```r
comp_ppod(1, 1, 1)  # => 1
comp_ppod(1, 1, 0)  # => 1
```
comp_PPV

Compute a decision's positive predictive value (PPV) from probabilities.

Description

comp_PPV computes the positive predictive value PPV from 3 essential probabilities prev, sens, and spec.

Usage

comp_PPV(prev, sens, spec)

Arguments

prev The condition’s prevalence prev (i.e., the probability of condition being TRUE).
sens The decision’s sensitivity sens (i.e., the conditional probability of a positive decision provided that the condition is TRUE).
spec The decision’s specificity value spec (i.e., the conditional probability of a negative decision provided that the condition is FALSE).

Details

comp_PPV uses probabilities (not frequencies) and does not round results.

Value

The positive predictive value PPV as a probability. A warning is provided for NaN values.

See Also

comp_sens and comp_NPV compute related probabilities; is_extreme_prob_set verifies extreme cases; comp_complement computes a probability’s complement; is_complement verifies probability complements; comp_prob computes current probability information; prob contains current probability information; is_prob verifies probabilities.

Other functions computing probabilities: comp_FDR, comp_FOR, comp_NPV, comp_accu_freq, comp_accu_prob, comp_acc, comp_comp_pair, comp_complement, comp_complete_prob_set, comp_err, comp_fart, comp_mirt, comp_ppod, comp_prob_freq, comp_prob, comp_sens, comp_spec
Examples

# (1) Ways to work:
comp_PPV(.50, .500, .500)  # => PPV = 0.5
comp_PPV(.50, .333, .666)  # => PPV = 0.499

# (2) Watch out for vectors:
prev <- seq(0, 1, .1)
comp_PPV(prev, .5, .5)  # => without NaN values
comp_PPV(prev, 0, 1)  # => with NaN values

# (3) Watch out for extreme values:
comp_PPV(prev = 1, sens = 0, spec = .5)  # => NaN, only mi: hi = 0 and fa = 0: PPV = 0/0 = NaN
is_extreme_prob_set(prev = 1, sens = 0, spec = .5)  # => verifies extreme cases

comp_PPV(prev = 0, sens = .5, spec = 1)  # => NaN, only cr: hi = 0 and fa = 0: PPV = 0/0 = NaN
is_extreme_prob_set(prev = 0, sens = .5, spec = 1)  # => verifies extreme cases

comp_PPV(prev = .5, sens = 0, spec = 1)  # => NaN, only cr: hi = 0 and fa = 0: PPV = 0/0 = NaN
is_extreme_prob_set(prev = .5, sens = 0, spec = 1)  # => verifies extreme cases

---

**comp_prev**

Compute the condition's prevalence (baseline probability) from frequencies.

**Description**

*comp_prev* computes a condition’s prevalence value *prev* (or baseline probability) from 4 essential frequencies (*hi*, *mi*, *fa*, *cr*).

**Usage**

```
comp_prev(hi = freq$hi, mi = freq$mi, fa = freq$fa, cr = freq$cr)
```

**Arguments**

- **hi**: The number of hits *hi* (or true positives).
- **mi**: The number of misses *mi* (or false negatives).
- **fa**: The number of false alarms *fa* (or false positives).
- **cr**: The number of correct rejections *cr* (or true negatives).

**Details**

A condition’s prevalence value *prev* is the probability of the condition being TRUE.

The probability *prev* can be computed from frequencies as the the ratio of *cond_true* (i.e., *hi* + *mi*) divided by *N* (i.e., *hi* + *mi* + *fa* + *cr*):

\[
prev = \frac{\text{cond_true}}{N} = \frac{(hi + mi)}{(hi + mi + fa + cr)}
\]
See Also

num contains basic numeric parameters; init_num initializes basic numeric parameters; prob contains current probability information; comp_prob computes current probability information; freq contains current frequency information; comp_freq computes current frequency information; is_prob verifies probability inputs; is_freq verifies frequency inputs.

---

**comp_prob**

*Compute probabilities from (3 essential) probabilities.*

**Description**

comp_prob computes current probability information from 3 essential probabilities (prev, sens or mirt, spec or fart). It returns a list of 13 probabilities prob as its output.

**Usage**

```r
comp_prob(prev = num$prev, sens = num$sens, mirt = NA,
spec = num$spec, fart = NA, tol = .01)
```

**Arguments**

- `prev` The condition’s prevalence value prev (i.e., the probability of the condition being TRUE).
- `sens` The decision’s sensitivity value sens (i.e., the conditional probability of a positive decision provided that the condition is TRUE). sens is optional when its complement mirt is provided.
- `mirt` The decision’s miss rate value mirt (i.e., the conditional probability of a negative decision provided that the condition is TRUE). mirt is optional when its complement sens is provided.
- `spec` The decision’s specificity value spec (i.e., the conditional probability of a negative decision provided that the condition is FALSE). spec is optional when its complement fart is provided.
- `fart` The decision’s false alarm rate fart (i.e., the conditional probability of a positive decision provided that the condition is FALSE). fart is optional when its complement spec is provided.
- `tol` A numeric tolerance value for is_complement. Default: tol = .01.

**Details**

comp_prob assumes that a sufficient and consistent set of essential probabilities (i.e., prev and either sens or its complement mirt, and either spec or its complement fart) is provided.

comp_prob computes and returns a full set of basic and various derived probabilities (e.g., the probability of a positive decision ppod, the probability of a correct decision acc, the predictive values PPV and NPV, as well as their complements FDR and FOR) in its output of a list prob.
Extreme probabilities (sets containing two or more probabilities of 0 or 1) may yield unexpected values (e.g., predictive values PPV or NPV turning NaN when is_extreme_prob_set evaluates to TRUE).

comp_prob is the probability counterpart to the frequency function comp_freq.

Key relationships between probabilities and frequencies:

- Three perspectives on a population:
  A population of \( N \) individuals can be split into 2 subsets of frequencies in 3 different ways:

  1. by condition:
     \[ N = \text{cond_true} + \text{cond_false} \]
     The frequency \( \text{cond_true} \) depends on the prevalence \( \text{prev} \) and the frequency \( \text{cond_false} \)
     depends on the prevalence’s complement \( 1 - \text{prev} \).

  2. by decision:
     \[ N = \text{dec_pos} + \text{dec_neg} \]
     The frequency \( \text{dec_pos} \) depends on the proportion of positive decisions \( \text{ppod} \) and the
     frequency \( \text{dec_neg} \) depends on the proportion of negative decisions \( 1 - \text{ppod} \).

  3. by accuracy (i.e., correspondence of decision to condition):
     \[ N = \text{dec_cor} + \text{dec_err} \]

Each perspective combines 2 pairs of the 4 essential probabilities (hi, mi, fa, cr).

When providing probabilities, the population size \( N \) is a free parameter (independent of the
essential probabilities \( \text{prev, sens, and spec} \)).

If \( N \) is unknown (NA), a suitable minimum value can be computed by \( \text{comp_min_N} \).

- Defining probabilities in terms of frequencies:

Probabilities are – determine, describe, or are defined as – the relationships between frequen-
cies. Thus, they can be computed as ratios between frequencies:

  1. prevalence \( \text{prev} \):
     \[ \text{prev} = \text{cond_true}/N = (\text{hi} + \text{mi}) / (\text{hi} + \text{mi} + \text{fa} + \text{cr}) \]

  2. sensitivity \( \text{sens} \):
     \[ \text{sens} = \text{hi}/\text{cond_true} = \text{hi} / (\text{hi} + \text{mi}) = (1 - \text{mirt}) \]

  3. miss rate \( \text{mirt} \):
     \[ \text{mirt} = \text{mi}/\text{cond_true} = \text{mi} / (\text{hi} + \text{mi}) = (1 - \text{sens}) \]

  4. specificity \( \text{spec} \):
     \[ \text{spec} = \text{cr}/\text{cond_false} = \text{cr} / (\text{fa} + \text{cr}) = (1 - \text{fart}) \]

  5. false alarm rate \( \text{fart} \):
     \[ \text{fart} = \text{fa}/\text{cond_false} = \text{fa} / (\text{fa} + \text{cr}) = (1 - \text{spec}) \]

  6. proportion of positive decisions \( \text{ppod} \):
     \[ \text{ppod} = \text{dec_pos}/N = (\text{hi} + \text{fa}) / (\text{hi} + \text{mi} + \text{fa} + \text{cr}) \]

  7. positive predictive value \( \text{PPV} \):
     \[ \text{PPV} = \text{hi}/\text{dec_pos} = \text{hi} / (\text{hi} + \text{fa}) = (1 - \text{FDR}) \]

  8. negative predictive value \( \text{NPV} \):
     \[ \text{NPV} = \text{cr}/\text{dec_neg} = \text{cr} / (\text{mi} + \text{cr}) = (1 - \text{FOR}) \]

  9. false detection rate \( \text{FDR} \):
     \[ \text{FDR} = \text{fa}/\text{dec_pos} = \text{fa} / (\text{hi} + \text{fa}) = (1 - \text{PPV}) \]

  10. false omission rate \( \text{FOR} \):
      \[ \text{FOR} = \text{mi}/\text{dec_neg} = \text{mi} / (\text{mi} + \text{cr}) = (1 - \text{NPV}) \]
11. accuracy acc:
   acc = \text{dec.cor}/N = (hi + cr) / (hi + mi + fa + cr)

   Note: When frequencies are rounded (by round = TRUE in \text{comp.freq}), probabilities computed from \text{freq} may differ from exact probabilities.

   Functions translating between representational formats: \text{comp.prob.prob, comp.prob.freq, comp.freq.prob, comp.freq.freq} (see documentation of \text{comp.prob.prob} for details).

Value

A list \text{prob} containing 13 probability values.

See Also

\text{prob} contains current probability information; \text{accu} contains current accuracy information; \text{num} contains basic numeric parameters; \text{init.num} initializes basic numeric parameters; \text{pal} contains current color information; \text{txt} contains current text information; \text{freq} contains current frequency information; \text{comp.freq} computes frequencies from probabilities; \text{is_valid_prob_set} verifies sets of probability inputs; \text{is_extreme_prob_set} verifies sets of extreme probabilities; \text{comp_min_N} computes a suitable minimum population size \(N\); \text{comp.freq.freq} computes current frequency information from (4 essential) frequencies; \text{comp.freq.prob} computes current frequency information from (3 essential) probabilities; \text{comp.prob.freq} computes current probability information from (4 essential) frequencies; \text{comp.prob.prob} computes current probability information from (3 essential) probabilities.


Examples

# Basics:
comp.prob(prev = .11, sens = .88, spec = .77)  # => ok: PPV = 0.3210614
comp.prob(prev = .11, sens = NA, mirt = .12, spec = NA, fart = .23)  # => ok: PPV = 0.3210614
comp.prob()  # => ok, using current defaults
length(comp.prob())  # => 13 probabilities

# Ways to work:
comp.prob(.99, sens = .99, spec = .99)  # => ok: PPV = 0.999898
comp.prob(.99, sens = .99, spec = NA, fart = .10)  # => ok: PPV = 0.9988789

# Watch out for extreme cases:
comp.prob(1, sens = 0, spec = 1)  # => ok, but with warnings (as PPV & FDR are NaN)
comp.prob(1, sens = 0, spec = 0)  # => ok, but with warnings (as PPV & FDR are NaN)
comp.prob(1, sens = 0, spec = NA, fart = 0)  # => ok, but with warnings (as PPV & FDR are NaN)
comp.prob(1, sens = 0, spec = NA, fart = 1)  # => ok, but with warnings (as PPV & FDR are NaN)

# Watch out for extreme cases:
comp.prob(1, sens = 0, spec = 1)  # => ok, but with warnings (as PPV & FDR are NaN)
comp.prob(1, sens = 0, spec = 0)  # => ok, but with warnings (as PPV & FDR are NaN)
comp.prob(1, sens = 0, spec = NA, fart = 0)  # => ok, but with warnings (as PPV & FDR are NaN)
**comp_prob_freq**

comp_prob(1, sens = 0, spec = NA, fart = 1) # => ok, but with warnings (as PPV & FDR are NaN)

comp_prob(1, sens = 1, spec = 0) # => ok, but with warnings (as NPV & FDR are NaN)

comp_prob(1, sens = 1, spec = 1) # => ok, but with warnings (as NPV & FDR are NaN)

comp_prob(1, sens = 1, spec = NA, fart = 0) # => ok, but with warnings (as NPV & FDR are NaN)

comp_prob(1, sens = 1, spec = NA, fart = 1) # => ok, but with warnings (as NPV & FDR are NaN)

# Ways to fail:
comp_prob(NA, 1, 1, NA) # => only warning: invalid set (prev not numeric)
comp_prob(8, 1, 1, NA) # => only warning: prev no probability
comp_prob(1, 8, 1, NA) # => only warning: sens no probability
comp_prob(1, 1, 1, 1) # => only warning: is_complement not in tolerated range

---

**comp_prob_freq**  
_Compute probabilities from (4 essential) frequencies._

---

**Description**

comp_prob_freq computes current probability information from 4 essential frequencies (hi, mi, fa, cr). It returns a list of 11 frequencies freq for a population of N individuals as its output.

**Usage**

```
comp_prob_freq(hi = freq$hi, mi = freq$mi, fa = freq$fa,  
                cr = freq$cr)
```

**Arguments**

- **hi**  
The number of hits hi (or true positives).

- **mi**  
The number of misses mi (or false negatives).

- **fa**  
The number of false alarms fa (or false positives).

- **cr**  
The number of correct rejections cr (or true negatives).

**Details**

Key relationships between frequencies and probabilities (see documentation of comp_freq or comp_prob for details):

- Three perspectives on a population:
  - by condition / by decision / by accuracy.
- Defining probabilities in terms of frequencies:
  - Probabilities can be computed as ratios between frequencies, but beware of rounding issues.

Functions translating between representational formats: comp_prob_prob, comp_prob_freq, comp_freq_prob, comp_freq_freq (see documentation of comp_prob_prob for details).
See Also

`comp_freq_freq` computes current frequency information from (4 essential) frequencies; `comp_freq_prob` computes current frequency information from (3 essential) probabilities; `comp_prob_prob` computes current probability information from (3 essential) probabilities; `num` contains basic numeric parameters; `init_num` initializes basic numeric parameters; `prob` contains current probability information; `comp_prob` computes current probability information; `freq` contains current frequency information; `comp_freq` computes current frequency information; `is_prob` verifies probability inputs; `is_freq` verifies frequency inputs.

Other functions computing probabilities: `comp_FDR`, `comp_FOR`, `comp_NPV`, `comp_PPV`, `comp_accu_freq`, `comp_accu_prob`, `comp_acc`, `comp_comp_pair`, `comp_complement`, `comp_complete_prob_set`, `comp_err`, `comp_fart`, `comp_mirt`, `comp_ppod`, `comp_prob`, `comp_sens`, `comp_spec`

Other format conversion functions: `comp_freq_freq`, `comp_freq_prob`, `comp_prob_prob`

Examples

```r
## Basics:
comp_freq_freq()  # => computes prob from current freq

## Beware of rounding:
all.equal(prob, comp_freq_freq())  # => would be TRUE (IF freq were NOT rounded!)
fe <- comp_freq(round = FALSE)  # compute exact freq (not rounded)
all.equal(prob, comp_prob_freq(fe$h, fe$mi, fe$fa, fe$cr))  # is TRUE (qed).

## Explain by circular chain (compute prob 1. from num and 2. from freq)
# 0. inspect current numeric parameters:
num

# 1. Compute currently 11 probabilities in prob (from essential probabilities):
prob <- comp_prob()
prob

# 2. Compute currently 11 frequencies in freq (from essential probabilities):
freq <- comp_freq(round = FALSE)  # no rounding (to obtain same probabilities later)
freq

# 3. Compute currently 11 probabilities again (but now from frequencies):
prob_freq <- comp_prob_freq()
prob_freq

# 4. Check equality of probabilities (in steps 1. and 3.):
all.equal(prob, prob_freq)  # => should be TRUE!
```
Description

`comp_prob_prob` computes current probability information from a sufficient and valid set of 3 essential probabilities (`prev`, and `sens` or its complement `mirt`, and `spec` or its complement `fart`). It returns a list of 11 probabilities (`prob`) as its output.

Usage

```r
comp_prob_prob(prev = prob$prev, sens = prob$sens, mirt = NA,
                 spec = prob$spec, fart = NA, tol = 0.01)
```

Arguments

- `prev` - The condition’s prevalence value `prev` (i.e., the probability of condition being TRUE).
- `sens` - The decision’s sensitivity value `sens` (i.e., the conditional probability of a positive decision provided that the condition is TRUE). `sens` is optional when its complement `mirt` is provided.
- `mirt` - The decision’s miss rate value `mirt` (i.e., the conditional probability of a negative decision provided that the condition is TRUE). `mirt` is optional when its complement `sens` is provided.
- `spec` - The decision’s specificity value `spec` (i.e., the conditional probability of a negative decision provided that the condition is FALSE). `spec` is optional when its complement `fart` is provided.
- `fart` - The decision’s false alarm rate `fart` (i.e., the conditional probability of a positive decision provided that the condition is FALSE). `fart` is optional when its complement `spec` is provided.
- `tol` - A numeric tolerance value for `is_complement`. Default: `tol = 0.01`.

Details

`comp_prob_prob` is a wrapper function for the more basic function `comp_prob`.

Extreme probabilities (sets containing 2 or more probabilities of 0 or 1) may yield unexpected values (e.g., predictive values PPV or NPV turning NaN when `is_extreme_prob_set` evaluates to TRUE).

Key relationships between frequencies and probabilities (see documentation of `comp_freq` or `comp_prob` for details):

- Three perspectives on a population: by condition / by decision / by accuracy.
- Defining probabilities in terms of frequencies: Probabilities can be computed as ratios between frequencies, but beware of rounding issues.

Functions translating between representational formats:

1. `comp_prob_prob` (defined here) is a wrapper function for `comp_prob` and an analog to 3 other format conversion functions:
2. `comp_prob_freq` computes current probability information contained in `prob` from 4 essential frequencies (`hi, mi, fa, cr`).

3. `comp_freq_prob` computes current frequency information contained in `freq` from 3 essential probabilities (`prev, sens, spec`).

4. `comp_freq_freq` computes current frequency information contained in `freq` from 4 essential frequencies (`hi, mi, fa, cr`).

**Value**

A list `prob` containing 11 probability values.

**See Also**

- `comp_freq_prob` computes current frequency information from (3 essential) probabilities;
- `comp_freq_freq` computes current frequency information from (4 essential) frequencies;
- `comp_prob_freq` computes current probability information from (4 essential) frequencies;
- `num` contains basic numeric variables;
- `init_num` initializes basic numeric variables;
- `freq` contains current frequency information;
- `comp_prob` computes current probability information;
- `comp_freq` computes current frequency information;
- `comp_complement` computes a probability’s complement;
- `comp_comp_pair` computes pairs of complements;
- `comp_complete_prob_set` completes valid sets of probabilities;
- `comp_min_N` computes a suitable population size `N` (if missing).

Other functions computing frequencies:

- `comp_freq_freq`, `comp_freq_prob`, `comp_freq`, `comp_min_N`, `comp_popu`

Other format conversion functions:

- `comp_freq_freq`, `comp_freq_prob`, `comp_prob_freq`

**Examples**

```r
# Basics:
comp_prob_prob(prev = .11, sens = .88, spec = .77)  # => ok: PPV = 0.3210614
comp_prob_prob(prev = .11, sens = NA, mirt = .12, spec = NA, fart = .23)  # => ok: PPV = 0.3210614
comp_prob_prob()  # => ok, using current defaults
length(comp_prob_prob())  # => 11 probabilities

# Ways to work:
comp_prob_prob(.99, sens = .99, spec = .99)  # => ok: PPV = 0.999898
comp_prob_prob(.99, sens = .90, spec = NA, fart = .10)  # => ok: PPV = 0.9988789

# Watch out for extreme cases:
comp_prob_prob(1, sens = 0, spec = 1)  # => ok, but with warnings (as PPV & FDR are NaN)
comp_prob_prob(1, sens = 0, spec = 0)  # => ok, but with warnings (as PPV & FDR are NaN)
comp_prob_prob(1, sens = 0, spec = NA, fart = 0)  # => ok, but with warnings (as PPV & FDR are NaN)
comp_prob_prob(1, sens = 0, spec = NA, fart = 1)  # => ok, but with warnings (as PPV & FDR are NaN)

comp_prob_prob(1, sens = 1, spec = 0)  # => ok, but with warnings (as NPV & FOR are NaN)
comp_prob_prob(1, sens = 1, spec = 1)  # => ok, but with warnings (as NPV & FOR are NaN)
comp_prob_prob(1, sens = 1, spec = NA, fart = 0)  # => ok, but with warnings (as NPV & FOR are NaN)
comp_prob_prob(1, sens = 1, spec = NA, fart = 1)  # => ok, but with warnings (as NPV & FOR are NaN)

# Ways to fail:
comp_prob_prob(NA, 1, 1, NA)  # => only warning: invalid set (prev not numeric)
```
**comp_sens**

Compute a decision’s sensitivity from its miss rate.

**Description**

`comp_sens` is a conversion function that takes a miss rate `mirt` – given as a probability (i.e., a numeric value in the range from 0 to 1) – as its input, and returns the corresponding sensitivity `sens` – also as a probability – as its output.

**Usage**

`comp_sens(mirt)`

**Arguments**

- **mirt**
  
  The decision’s miss rate `mirt` as a probability.

**Details**

The sensitivity `sens` and miss rate `mirt` are complements (`sens = (1 - mirt)`) and both features of the decision process (e.g., a diagnostic test).

The function `comp_sens` is complementary to the conversion function `comp_mirt` and uses the generic function `comp_complement`.

**Value**

The decision’s sensitivity `sens` as a probability.

**See Also**

- `comp_complement` computes a probability’s complement; `is_complement` verifies probability complements; `comp_prob` computes current probability information; `prob` contains current probability information; `is_prob` verifies probabilities.

**Examples**

- `comp_sens(2)  # => NA + warning (beyond range)`
- `comp_sens(1/3)  # => 0.6666667`
- `comp_sens(comp_complement(0.123))  # => 0.123`
comp_spec

*Compute a decision’s specificity from its false alarm rate.*

**Description**

comp_spec is a conversion function that takes a false alarm rate `fart` – given as a probability (i.e., a numeric value in the range from 0 to 1) – as its input, and returns the corresponding specificity `spec` – also as a probability – as its output.

**Usage**

`comp_spec(fart)`

**Arguments**

- `fart` The decision’s false alarm rate `fart` as a probability.

**Details**

The specificity `spec` and the false alarm rate `fart` are complements (spec = (1 - fart)) and both features of the decision process (e.g., a diagnostic test).

The function `comp_spec` is complementary to the conversion function `comp_fart` and uses the generic function `comp_complement`.

**Value**

The decision’s specificity `spec` as a probability.

**See Also**

`comp_complement` computes a probability’s complement; `is_complement` verifies probability complements; `comp_prob` computes current probability information; `prob` contains current probability information; `is_prob` verifies probabilities.

Other functions computing probabilities: `comp_FDR`, `comp_FOR`, `comp_NPV`, `comp_PPV`, `comp_accu_freq`, `comp_accu_prob`, `comp_acc`, `comp_comp_pair`, `comp_complement`, `comp_complete_prob_set`, `comp_err`, `comp_fart`, `comp_mirt`, `comp_ppod`, `comp_prob_freq`, `comp_prob`, `comp_sens`

**Examples**

```r
comp_spec(2)  # => NA + warning (beyond range)
comp_spec(1/3) # => 0.6666667
comp_spec(comp_complement(0.123)) # => 0.123
```
**cond_false**

*Number of individuals for which the condition is false.*

**Description**

*cond_false* is a frequency that describes the number of individuals in the current population *N* for which the condition is FALSE (i.e., actually false cases).

**Usage**

`cond_false`

**Format**

An object of class `numeric` of length 1.

**Details**

Key relationships:

1. to probabilities: The frequency of *cond_false* individuals depends on the population size *N* and the complement of the condition's prevalence *1 - prev* and is split further into two subsets of *fa* by the false alarm rate *fart* and *cr* by the specificity *spec*.

   Perspectives:
   
   (a) by condition:
   
   The frequency *cond_false* is determined by the population size *N* times the complement of the prevalence *(1 - prev)*:
   
   \[
   \text{cond_false} = N \times (1 - \text{prev})
   \]

   (b) by decision:
   
   a. The frequency *fa* is determined by *cond_false* times the false alarm rate *fart = (1 - spec)* (aka. FPR):
   
   \[
   \text{fa} = \text{cond_false} \times \text{fart} = \text{cond_false} \times (1 - \text{spec})
   \]

   b. The frequency *cr* is determined by *cond_false* times the specificity *spec = (1 - fart)*:
   
   \[
   \text{cr} = \text{cond_false} \times \text{spec} = \text{cond_false} \times (1 - \text{fart})
   \]

2. to other frequencies: In a population of size *N* the following relationships hold:

   - *N = cond_true + cond_false* (by condition)
   - *N = dec_pos + dec_neg* (by decision)
   - *N = dec_cor + dec_err* (by correspondence of decision to condition)
   - *N = hi + mi + fa + cr* (by condition x decision)

   Current frequency information is computed by `comp_freq` and contained in a list `freq`.

**References**

Consult *Wikipedia: Confusion matrix* for additional information.
cond_true

See Also

is_freq verifies frequencies; num contains basic numeric parameters; init_num initializes basic numeric parameters; freq contains current frequency information; comp_freq computes current frequency information; prob contains current probability information; comp_prob computes current probability information.

Other frequencies: N, cond_true, cr, dec_cor, dec_err, dec_neg, dec_pos, fa, hi, mi

Examples

cond_false <- 1000 * .90  # => sets cond_false to 90% of 1000 = 900 cases.
is_freq(cond_false)        # => TRUE
is_prob(cond_false)        # => FALSE, as cond_false is no probability [but (1 - prev) and spec are]

cond_true

Number of individuals for which the condition is true.

Description

cond_true is a frequency that describes the number of individuals in the current population N for which the condition is TRUE (i.e., actually true cases).

Usage

cond_true

Format

An object of class numeric of length 1.

Details

Key relationships:

1. to probabilities: The frequency of cond_true individuals depends on the population size N and the condition's prevalence prev and is split further into two subsets of hi by the sensitivity sens and mi by the miss rate mirt.

Perspectives:

(a) by condition:
   The frequency cond_true is determined by the population size N times the prevalence prev:
   \[ \text{cond_true} = N \times \text{prev} \]

(b) by decision:
   a. The frequency hi is determined by cond_true times the sensitivity sens (aka. hit rate HR):
      \[ hi = \text{cond_true} \times \text{sens} \]
   b. The frequency mi is determined by cond_true times the miss rate mirt = (1 - sens):
      \[ mi = \text{cond_true} \times \text{mirt} = \text{cond_true} \times (1 - \text{sens}) \]
2. to other frequencies: In a population of size \( N \) the following relationships hold:

- \( N = \text{cond\_true} + \text{cond\_false} \) (by condition)
- \( N = \text{dec\_pos} + \text{dec\_neg} \) (by decision)
- \( N = \text{dec\_cor} + \text{dec\_err} \) (by correspondence of decision to condition)
- \( N = \text{hi} + \text{mi} + \text{fa} + \text{cr} \) (by condition x decision)

Current frequency information is computed by `comp\_freq` and contained in a list `freq`.

**References**


**See Also**

`is\_freq` verifies frequencies; `num` contains basic numeric parameters; `init\_num` initializes basic numeric parameters; `freq` contains current frequency information; `comp\_freq` computes current frequency information; `prob` contains current probability information; `comp\_prob` computes current probability information.

Other frequencies: \( N, \text{cond\_false}, \text{cr}, \text{dec\_cor}, \text{dec\_err}, \text{dec\_neg}, \text{dec\_pos}, \text{fa}, \text{hi}, \text{mi} \)

**Examples**

```r
cond\_true <- 1000 * .10  # => sets cond\_true to 10% of 1000 = 100 cases.
is\_freq(cond\_true)       # => TRUE
is\_prob(cond\_true)      # => FALSE, as cond\_true is no probability (but prev and sens are)
```

---

**cr**

*Frequency of correct rejections or true negatives (TN).*

**Description**

`cr` is the frequency of correct rejections or true negatives (TN) in a population of \( N \) individuals.

**Usage**

`cr`

**Format**

An object of class `numeric` of length 1.
Details

Definition: \( cr \) is the frequency of individuals for which Condition = FALSE and Decision = FALSE (negative).

\( cr \) is a measure of correct classifications, not an individual case.

Relationships:

1. to probabilities: The frequency \( cr \) depends on the specificity \( spec \) (aka. true negative rate, TNR) and is conditional on the prevalence \( prev \).

2. to other frequencies: In a population of size \( N \) the following relationships hold:
   - \( N = \text{cond_true} + \text{cond_false} \) (by condition)
   - \( N = \text{dec_pos} + \text{dec_neg} \) (by decision)
   - \( N = \text{dec_cor} + \text{dec_err} \) (by correspondence of decision to condition)
   - \( N = \text{hi} + \text{mi} + \text{fa} + cr \) (by condition x decision)

See Also

\( spec \) is the specificity or correct rejection rate (aka. true negative rate TNR); \( num \) contains basic numeric parameters; \( init_num \) initializes basic numeric parameters; \( freq \) contains current frequency information; \( comp_freq \) computes current frequency information; \( prob \) contains current probability information; \( comp_prob \) computes current probability information; \( is_freq \) verifies frequencies.

Other essential parameters: \( fa, hi, mi, prev, sens, spec \)

Other frequencies: \( N, \text{cond_false}, \text{cond_true}, \text{dec_cor}, \text{dec_err}, \text{dec_neg}, \text{dec_pos}, \text{fa}, \text{hi}, \text{mi} \)

---

\( \text{dec_cor} \)  \( Number \ of \ individuals \ for \ which \ the \ decision \ is \ correct. \)

Description

\( \text{dec_cor} \) is a frequency that describes the number of individuals in the current population \( N \) for which the decision is correct/accurate (i.e., cases in which the decision corresponds to the condition).

Usage

\( \text{dec_cor} \)

Format

An object of class numeric of length 1.
dec_err

Details

Key relationships:

1. to probabilities: The frequency of dec_cor individuals depends on the population size \( N \) and the accuracy \( acc \).

2. to other frequencies: In a population of size \( N \) the following relationships hold:
   - \( N = \text{cond_true} + \text{cond_false} \) (by condition)
   - \( N = \text{dec_pos} + \text{dec_neg} \) (by decision)
   - \( N = \text{dec_cor} + \text{dec_err} \) (by correspondence of decision to condition)
   - \( \text{dec_cor} = \text{hi} + \text{cr} \)
   - \( \text{dec_err} = \text{mi} + \text{fa} \)
   - \( N = \text{hi} + \text{mi} + \text{fa} + \text{cr} \) (by condition x decision)

3. correspondence: When not rounding the frequencies of \( freq \) then
   - \( \text{dec_cor} = N \times acc = \text{hi} + \text{cr} \)

Current frequency information is computed by \text{comp_freq} and contained in a list \text{freq}.

References

Consult Wikipedia: Confusion matrix for additional information.

See Also

is_freq verifies frequencies; num contains basic numeric parameters; init_num initializes basic numeric parameters; freq contains current frequency information; comp_freq computes current frequency information; prob contains current probability information; comp_prob computes current probability information.

Other frequencies: \( N, \text{cond_false}, \text{cond_true}, \text{cr}, \text{dec_err}, \text{dec_neg}, \text{dec_pos}, \text{fa}, \text{hi}, \text{mi} \)

Examples

dec_cor <- 1000 * .50  # => sets dec_cor to 50% of 1000 = 500 cases.
is_freq(dec_cor)        # => TRUE
is_prob(dec_cor)        # => FALSE, as dec_cor is no probability (but acc, bacc/wacc ARE)

---

dec_err  

*Number of individuals for which the decision is erroneous.*

Description

\( \text{dec_err} \) is a frequency that describes the number of individuals in the current population \( N \) for which the decision is incorrect or erroneous (i.e., cases in which the decision does not correspond to the condition).
Usage

\texttt{dec_err}

Format

An object of class \texttt{numeric} of length 1.

Details

Key relationships:

1. to probabilities: The frequency of \texttt{dec_err} individuals depends on the population size \( N \) and is equal to the sum of false negatives \( mi \) and false positives \( fa \).

2. to other frequencies: In a population of size \( N \) the following relationships hold:

\begin{itemize}
  \item \( N = \text{cond\_true} + \text{cond\_false} \) (by condition)
  \item \( N = \text{dec\_pos} + \text{dec\_neg} \) (by decision)
  \item \( N = \text{dec\_cor} + \text{dec\_err} \) (by correspondence of decision to condition)
  \item \( \text{dec\_cor} = \text{hi} + \text{cr} \)
  \item \( \text{dec\_err} = \text{mi} + \text{fa} \)
  \item \( N = \text{hi} + \text{mi} + \text{fa} + \text{cr} \) (by condition x decision)
\end{itemize}

Current frequency information is computed by \texttt{comp\_freq} and contained in a list \texttt{freq}.

References

Consult Wikipedia: Confusion matrix for additional information.

See Also

\texttt{is\_freq} verifies frequencies; \texttt{num} contains basic numeric parameters; \texttt{init\_num} initializes basic numeric parameters; \texttt{freq} contains current frequency information; \texttt{comp\_freq} computes current frequency information; \texttt{prob} contains current probability information; \texttt{comp\_prob} computes current probability information.

Other frequencies: \( N, \text{cond\_false}, \text{cond\_true}, \text{cr}, \text{dec\_cor}, \text{dec\_neg}, \text{dec\_pos}, \text{fa}, \text{hi}, \text{mi} \)

Examples

\begin{verbatim}
  dec_err <- 1000 * .50  # => sets dec_err to 50% of 1000 = 500 cases.
  is_freq(dec_err)        # => TRUE
  is_prob(dec_err)        # => FALSE, as dec_err is no probability (but acc, bacc/wacc ARE)
\end{verbatim}
\textit{dec\_neg} \hspace{3cm} \textit{Number of individuals for which the decision is negative.}

**Description**

\texttt{dec\_neg} is a frequency that describes the number of individuals in the current population \( N \) for which the decision is negative (i.e., cases not called or not predicted).

**Usage**

\texttt{dec\_neg}

**Format**

An object of class \texttt{numeric} of length 1.

**Details**

Key relationships:

1. to probabilities: The frequency of \texttt{dec\_neg} individuals depends on the population size \( N \) and the decision’s proportion of negative decisions \( (1 - \text{ppod}) \) and is split further into two subsets of \( \text{cr} \) by the negative predictive value \texttt{npv} and \( \text{mi} \) by the false omission rate \texttt{FOR} = 1 - \texttt{NPV}.

   Perspectives:
   
   (a) by condition:
   
   The frequency \texttt{dec\_neg} is determined by the population size \( N \) times the proportion of negative decisions \( (1 - \text{ppod}) \):
   
   \[ \texttt{dec\_neg} = N \times (1 - \text{ppod}) \]

   (b) by decision:
   
   a. The frequency \texttt{cr} is determined by \texttt{dec\_neg} times the negative predictive value \texttt{NPV}:
   
   \[ \texttt{cr} = \texttt{dec\_neg} \times \texttt{NPV} \]

   b. The frequency \texttt{mi} is determined by \texttt{dec\_neg} times the false omission rate \texttt{FOR} = (1 - \texttt{NPV}):
   
   \[ \texttt{mi} = \texttt{dec\_neg} \times \texttt{FOR} = \texttt{dec\_neg} 	imes (1 - \texttt{NPV}) \]

2. to other frequencies: In a population of size \( N \) the following relationships hold:

   - \( \texttt{N} = \texttt{cond\_true} + \texttt{cond\_false} \) (by condition)
   - \( \texttt{N} = \texttt{dec\_pos} + \texttt{dec\_neg} \) (by decision)
   - \( \texttt{N} = \texttt{dec\_cor} + \texttt{dec\_err} \) (by correspondence of decision to condition)
   - \( \texttt{N} = \texttt{hi} + \texttt{mi} + \texttt{fa} + \texttt{cr} \) (by condition x decision)

Current frequency information is computed by \texttt{comp\_freq} and contained in a list \texttt{freq}.

**References**

Consult Wikipedia: Confusion matrix for additional information.
See Also

- `is_freq` verifies frequencies;
- `num` contains basic numeric parameters;
- `init_num` initializes basic numeric parameters;
- `freq` contains current frequency information;
- `comp_freq` computes current frequency information;
- `prob` contains current probability information;
- `comp_prob` computes current probability information.

Other frequencies: `N, cond_false, cond_true, cr, dec_cor, dec_err, dec_pos, fa, hi, mi`

Examples

```r
dec_neg <- 1000 * .67  # => sets dec_neg to 67% of 1000 = 670 cases.
is_freq(dec_neg)      # => TRUE
is_prob(dec_neg)      # => FALSE, as dec_neg is no probability (but ppod, NPV and FOR are)
```

---

**dec_pos**

*Number of individuals for which the decision is positive.*

**Description**

`dec_pos` is a frequency that describes the number of individuals in the current population `N` for which the decision is positive (i.e., called or predicted cases).

**Usage**

`dec_pos`

**Format**

An object of class `numeric` of length 1.

**Details**

Key relationships:

1. **to probabilities:** The frequency of `dec_pos` individuals depends on the population size `N` and the decision’s proportion of positive decisions `ppod` and is split further into two subsets of `hi` by the positive predictive value `PPV` and `fa` by the false detection rate `FDR = 1 - PPV`.

   **Perspectives:**
   
   (a) **by condition:**
   
   The frequency `dec_pos` is determined by the population size `N` times the proportion of positive decisions `ppod`:
   
   \[
   \text{dec_pos} = N \times \text{ppod}
   \]

   (b) **by decision:**
   
   a. The frequency `hi` is determined by `dec_pos` times the positive predictive value `PPV` (aka. `precision`):
   
   \[
   \text{hi} = \text{dec_pos} \times \text{PPV}
   \]
   
   b. The frequency `fa` is determined by `dec_pos` times the false detection rate `FDR = (1 - PPV)`:
   
   \[
   \text{fa} = \text{dec_pos} \times \text{FDR} = \text{dec_pos} \times (1 - \text{PPV})
   \]
2. to other frequencies: In a population of size \(N\) the following relationships hold:

- \(N = \text{cond_true} + \text{cond_false}\) (by condition)
- \(N = \text{dec_pos} + \text{dec_neg}\) (by decision)
- \(N = \text{dec_cor} + \text{dec_err}\) (by correspondence of decision to condition)
- \(N = \text{hi} + \text{mi} + \text{fa} + \text{cr}\) (by condition \times decision)

Current frequency information is computed by \text{comp_freq} and contained in a list \text{freq}.

References
Consult Wikipedia: Confusion matrix for additional information.

See Also
\text{is_freq} verifies frequencies; \text{num} contains basic numeric parameters; \text{init_num} initializes basic numeric parameters; \text{freq} contains current frequency information; \text{comp_freq} computes current frequency information; \text{prob} contains current probability information; \text{comp_prob} computes current probability information.

Other frequencies: \(N, \text{cond_false}, \text{cond_true}, \text{cr}, \text{dec_cor}, \text{dec_err}, \text{dec_neg}, \text{fa}, \text{hi}, \text{mi}\)

Examples
\[
dec_pos \leftarrow 1000 \times 0.33 \quad \# \Rightarrow \text{sets dec_pos to 33\% of 1000 = 330 cases.}
\]
\[
is_freq(dec_pos) \quad \# \Rightarrow \text{TRUE}
\]
\[
is_prob(dec_pos) \quad \# \Rightarrow \text{FALSE, as dec_pos is no probability (but ppod and PPV are)}
\]

\[
\begin{array}{l}
\text{df_scenarios} \\
A \text{collection of riskyr scenarios from various sources (as df).}
\end{array}
\]

Description
\text{df_scenarios} is an R data frame that contains a collection of scenarios from the scientific literature and other sources.

Usage
\text{df_scenarios}

Format
A data frame with currently 25 rows (i.e., scenarios) and 21 columns (variables describing each scenario):

See \text{scenarios} for a list of scenarios and the variables currently contained in \text{df_scenarios}.

Note that names of variables (columns) correspond to a subset of \text{init_txt} (to initialize \text{txt}) and \text{init_num} (to initialize \text{num}).

The variables \text{scen_src} and \text{scen_apa} provide a scenario’s source information.
Details

When loading riskyr, all scenarios contained in df_scenarios are converted into a list of riskyr objects scenarios.

See Also

scenarios contains all scenarios as riskyr objects; riskyr initializes a riskyr scenario; txt contains basic text information; init_txt initializes text information; num contains basic numeric parameters; init_num initializes basic numeric parameters; pal contains current color information; init_pal initializes color information.

| err | Error rate (err) as the probability of an incorrect decision. |

Description

err defines the error rate as the complement of accuracy acc or lack of correspondence of decisions to conditions.

Usage

er

Format

An object of class numeric of length 1.

Details

Definition:

\[ err = (1 - acc) \]

When freq are not rounded (round = FALSE) then

\[ err = \frac{dec_{err}}{N} = \frac{(mi + fa)}{N} \]

err is currently not included in prob, but shown in plots.

See err’s complement of accuracy acc for computation and accu for current accuracy metrics and several possible interpretations of accuracy.

See Also

acc provides overall accuracy; comp_acc computes accuracy from probabilities; accu lists current accuracy metrics; comp_accu_prob computes exact accuracy metrics from probabilities; comp_accu_freq computes accuracy metrics from frequencies; comp_sens and comp_PPV compute related probabilities; is_extreme_prob_set verifies extreme cases; comp_complement computes a probability's complement; is_complement verifies probability complements; comp_prob computes current probability information; prob contains current probability information; is_prob verifies probabilities.
Other probabilities: FDR, FOR, NPV, PPV, acc, fart, mirt, ppod, prev, sens, spec
Other metrics: accu, acc, comp_accu_freq, comp_accu_prob, comp_accu, comp_err

Examples

\[
\begin{align*}
\text{err} &\leftarrow 0.50 \quad \# \text{sets a rate of incorrect decisions of 50}\% \\
\text{err} &\leftarrow 50/100 \quad \# \text{(dec_err) for 50 out of 100 individuals} \\
\text{is_prob(err)} &\leftarrow \# \text{TRUE}
\end{align*}
\]

---

\textit{fa}

\textit{Frequency of false alarms or false positives (FP).}

**Description**

fa is the frequency of false alarms or false positives (FP) in a population of \(N\) individuals.

**Usage**

\texttt{fa}

**Format**

An object of class \texttt{numeric} of length 1.

**Details**

Definition: fa is the frequency of individuals for which Condition = FALSE and Decision = TRUE (positive).

fa is a measure of incorrect classifications (type-I-errors), not an individual case.

Relationships:

1. to probabilities: The frequency \(fa\) depends on the false alarm rate \(fart\) (aka. false positive rate, FPR) and is conditional on the prevalence \(prev\).

2. to other frequencies: In a population of size \(N\) the following relationships hold:

- \(N = \text{cond_true} + \text{cond_false}\) (by condition)
- \(N = \text{dec_pos} + \text{dec_neg}\) (by decision)
- \(N = \text{dec_cor} + \text{dec_err}\) (by correspondence of decision to condition)
- \(N = \text{hi} + \text{mi} + \text{fa} + \text{cr}\) (by condition x decision)
See Also

fart is the probability of false alarms (aka. false positive rate FPR or fallout); num contains basic numeric parameters; init_num initializes basic numeric parameters; freq contains current frequency information; comp_freq computes current frequency information; prob contains current probability information; comp_prob computes current probability information; is_freq verifies frequencies.

Other essential parameters: cr, hi, mi, prev, sens, spec

Other frequencies: N, cond_false, cond_true, cr, dec_cor, dec_err, dec_neg, dec_pos, hi, mi

---

**fart**

The false alarm rate (or false positive rate) of a decision process or diagnostic procedure.

---

Description

fart defines a decision’s false alarm rate (or the rate of false positives): The conditional probability of the decision being positive if the condition is FALSE.

Usage

fart

Format

An object of class numeric of length 1.

Details

Understanding or obtaining the false alarm rate fart:

- Definition: fart is the conditional probability for an incorrect positive decision given that the condition is FALSE:
  \[
  \text{fart} = p(\text{decision} = \text{positive} \mid \text{condition} = \text{FALSE})
  \]
  or the probability of a false alarm.

- Perspective: fart further classifies the subset of cond_false individuals by decision (fart = fa/cond_false).

- Alternative names: false positive rate (FPR), rate of type-I errors (alpha), statistical significance level, fallout

- Relationships:
  a. fart is the complement of the specificity spec:
     \[
     \text{fart} = 1 - \text{spec}
     \]
  b. fart is the opposite conditional probability – but not the complement – of the false discovery rate or false detection rate FDR:
     \[
     \text{FDR} = p(\text{condition} = \text{FALSE} \mid \text{decision} = \text{positive})
     \]
• In terms of frequencies, \( \text{fart} \) is the ratio of \( \text{fa} \) divided by \( \text{cond\_false} \) (i.e., \( \text{fa} + \text{cr} \)):
\[
\text{fart} = \frac{\text{fa}}{\text{cond\_false}} = \frac{\text{fa}}{\text{fa} + \text{cr}}
\]
• Dependencies: \( \text{fart} \) is a feature of a decision process or diagnostic procedure and a measure of incorrect decisions (false positives).
However, due to being a conditional probability, the value of \( \text{fart} \) is not intrinsic to the decision process, but also depends on the condition’s prevalence value \( \text{prev} \).

References
Consult Wikipedia for additional information.

See Also
\text{comp\_fart} computes \( \text{fart} \) as the complement of \text{spec\_prob} contains current probability information; \text{comp\_prob} computes current probability information; \text{num} contains basic numeric parameters; \text{init\_num} initializes basic numeric parameters; \text{comp\_freq} computes current frequency information; \text{is\_prob} verifies probabilities.

Other probabilities: \text{FDR, FOR, NPV, PPV, acc, err, mirt, ppod, prev, sens, spec}

Examples
\[
\text{fart} \leftarrow .25 \quad \# \text{ sets a false alarm rate of 25%}
\text{fart} \leftarrow 25/100 \quad \# (\text{decision} = \text{positive}) \text{ for 25 out of 100 people with} (\text{condition} = \text{FALSE})
\text{is\_prob}(\text{fart}) \quad \# \text{TRUE}
\]

---

\text{FDR} \quad \text{The false detection rate of a decision process or diagnostic procedure.}

Description
\text{FDR} \text{ defines a decision’s false detection (or false discovery) rate (FDR): The conditional probability of the condition being FALSE provided that the decision is positive.}

Usage
\text{FDR}

Format
An object of class \text{numeric} of length 1.
Details

Understanding or obtaining the false detection fate or false discovery rate (FDR):

- Definition: FDR is the conditional probability for the condition being FALSE given a positive decision:
  \[ \text{FDR} = p(\text{condition} = \text{FALSE} \mid \text{decision} = \text{positive}) \]

- Perspective: FDR further classifies the subset of \text{dec\_pos} individuals by condition (FDR = \text{fa/dec\_pos} = \text{fa}(\text{hi} + \text{fa})).

- Alternative names: false discovery rate

- Relationships:
  a. FDR is the complement of the positive predictive value \text{PPV}:
     \[ \text{FDR} = 1 - \text{PPV} \]
  b. FDR is the opposite conditional probability – but not the complement – of the false alarm rate \text{fart}:
     \[ \text{fart} = p(\text{decision} = \text{positive} \mid \text{condition} = \text{FALSE}) \]

- In terms of frequencies, FDR is the ratio of \text{fa} divided by \text{dec\_pos} (i.e., \text{hi} + \text{fa}):
  \[ \text{FDR} = \text{fa/dec\_pos} = \text{fa}(\text{hi} + \text{fa}) \]

- Dependencies: FDR is a feature of a decision process or diagnostic procedure and a measure of incorrect decisions (positive decisions that are actually FALSE). However, due to being a conditional probability, the value of FDR is not intrinsic to the decision process, but also depends on the condition’s prevalence value \text{prev}.

References

Consult Wikipedia for additional information.

See Also

\text{prob} contains current probability information; \text{comp\_prob} computes current probability information; \text{num} contains basic numeric parameters; \text{init\_num} initializes basic numeric parameters; \text{freq} contains current frequency information; \text{comp\_freq} computes current frequency information; \text{is\_prob} verifies probabilities.

Other probabilities: \text{FOR, NPV, PPV, acc, err, fart, mirt, ppod, prev, sens, spec}

Examples

\begin{verbatim}
FDR <<- .45    # sets a false detection rate (FDR) of 45%
FDR <<- 45/100 # (condition = FALSE) for 45 out of 100 people with (decision = positive)
is_prob(FDR)    # TRUE
\end{verbatim}
FOR defines a decision’s false omission rate (FOR): The conditional probability of the condition being TRUE provided that the decision is negative.

Usage

FOR

Format

An object of class numeric of length 1.

Details

Understanding or obtaining the false omission rate FOR:

- Definition: FOR is the so-called false omission rate: The conditional probability for the condition being TRUE given a negative decision:
  \[ \text{FOR} = \frac{\text{mi}}{\text{dec_neg}} = \frac{\text{mi}}{(\text{mi} + \text{cr})} \]
- Perspective: FOR further classifies the subset of dec_neg individuals by condition (FOR = mi/dec_neg = mi/(mi + cr))
- Alternative names: none?
- Relationships:
  a. FOR is the complement of the negative predictive value NPV:
  \[ \text{FOR} = 1 - \text{NPV} \]
  b. FOR is the opposite conditional probability – but not the complement – of the miss rate mirt (aka. false negative rate FDR):
  \[ \text{mirt} = \frac{\text{mi}}{\text{dec_neg}} = \frac{\text{mi}}{(\text{mi} + \text{cr})} \]
- In terms of frequencies, FOR is the ratio of mi divided by dec_neg (i.e., mi + cr):
  \[ \text{NPV} = \frac{\text{mi}}{\text{dec_neg}} = \frac{\text{mi}}{(\text{mi} + \text{cr})} \]
- Dependencies: FOR is a feature of a decision process or diagnostic procedure and a measure of incorrect decisions (negative decisions that are actually FALSE). However, due to being a conditional probability, the value of FOR is not intrinsic to the decision process, but also depends on the condition’s prevalence value prev.

References

Consult Wikipedia for additional information.
See Also

comp_FOR computes FOR as the complement of NPV; prob contains current probability information; comp_prob computes current probability information; num contains basic numeric parameters; init_num initializes basic numeric parameters; comp_freq computes current frequency information; is_prob verifies probabilities.

Other probabilities: FDR, NPV, PPV, acc, err, fart, mirt, ppod, prev, sens, spec

Examples

```r
FOR <- .05 # sets a false omission rate of 5%
FOR <- 5/100 # (condition = TRUE) for 5 out of 100 people with (decision = negative)
is_prob(FOR) # TRUE
```

freq

List current frequency information.

Description

freq is a list of named numeric variables containing 11 frequencies:

Usage

freq

Format

An object of class list of length 11.

Details

1. the population size \( N \)
2. the number of cases for which \( \text{cond\_true} \)
3. the number of cases for which \( \text{cond\_false} \)
4. the number of cases for which \( \text{dec\_pos} \)
5. the number of cases for which \( \text{dec\_neg} \)
6. the number of cases for which \( \text{dec\_cor} \)
7. the number of cases for which \( \text{dec\_err} \)
8. the number of true positives, or hits \( hi \)
9. the number of false negatives, or misses \( mi \)
10. the number of false positives, or false alarms \( fa \)
hi

11. the number of true negatives, or correct rejections cr

These frequencies are computed from basic parameters (contained in num) and computed by using comp_freq.

The list freq is the frequency counterpart to the list containing probability information prob.

Natural frequencies are always expressed in relation to the current population of size N.

Key relationships between frequencies and probabilities (see documentation of comp_freq or comp_prob for details):

- Three perspectives on a population:
  - by condition / by decision / by accuracy.
- Defining probabilities in terms of frequencies:
  - Probabilities can be computed as ratios between frequencies, but beware of rounding issues.

Functions translating between representational formats: comp_prob_prob, comp_prob_freq, comp_freq_prob, comp_freq_freq (see documentation of comp_prob_prob for details).

Visualizations of current frequency information are provided by plot_prism and plot_icons.

See Also

comp_freq computes current frequency information; num contains basic numeric variables; init_num initializes basic numeric variables; prob contains current probability information; num contains basic numeric parameters; init_num initializes basic numeric parameters; txt contains current text information; init_txt initializes text information; pal contains current color information; init_pal initializes color information.

Other lists containing current scenario information: accu, num, pal_bw, pal_kn, pal_mbw, pal_mod, pal_org, pal_rgb, pal_vir, pal, prob, txt_TF, txt_org, txt

Examples

```r
freq <- comp_freq()  # => initialize freq to default parameters
defreq  # => show current values
length(freq)  # => 11 known frequencies
names(freq)  # => show names of known frequencies
```

---

**hi**

*Frequency of hits or true positives (TP).*

Description

hi is the frequency of hits or true positives (TP) in a population of N individuals.

Usage

hi
init_num

**Format**

An object of class numeric of length 1.

**Details**

Definition: hi is the frequency of individuals for which Condition = TRUE and Decision = TRUE (positive).

hi is a measure of correct classifications, not an individual case.

Relationships:

1. to probabilities: The frequency hi depends on the sensitivity sens (aka. hit rate or true positive rate, TPR) and is conditional on the prevalence prev.
2. to other frequencies: In a population of size N the following relationships hold:
   - N = cond_true + cond_false (by condition)
   - N = dec_pos + dec_neg (by decision)
   - N = dec_cor + dec_err (by correspondence of decision to condition)
   - N = hi + mi + fa + cr (by condition x decision)

**See Also**

sens is the probability of hits or hit rate HR; num contains basic numeric parameters; init_num initializes basic numeric parameters; freq contains current frequency information; comp_freq computes current frequency information; prob contains current probability information; comp_prob computes current probability information; is_freq verifies frequencies.

Other frequencies: N, cond_false, cond_true, cr, dec_cor, dec_err, dec_neg, dec_pos, fa, mi

Other essential parameters: cr, fa, mi, prev, sens, spec
Arguments

prev  The condition’s prevalence value prev (i.e., the probability of condition being TRUE).

sens  The decision’s sensitivity value sens (i.e., the conditional probability of a positive decision provided that the condition is TRUE).

spec  The decision’s specificity value spec (i.e., the conditional probability of a negative decision provided that the condition is FALSE). spec is optional when its complement fart is provided.

fart  The decision’s false alarm rate fart (i.e., the conditional probability of a positive decision provided that the condition is FALSE). fart is optional when its complement spec is provided.

N  The population size N.

Details

If spec is provided, its complement fart is optional. If fart is provided, its complement spec is optional. If no N is provided, a suitable minimum value is computed by comp_min_N.

Value

A list containing a valid quadruple of probabilities (prev, sens, spec, and fart) and one frequency (population size N).

See Also

num contains basic numeric parameters; pal contains current color settings; txt contains current text settings; freq contains current frequency information; comp_freq computes frequencies from probabilities; prob contains current probability information; comp_prob computes current probability information; is_valid_prob_set verifies sets of probability inputs; is_extreme_prob_set verifies sets of extreme probabilities; comp_min_N computes a suitable minimum population size N.

Other functions initializing scenario information: init_pal, init_txt, riskyr

Examples

# ways to succeed:
init_num(1, 1, 1, 0, 100)  # => succeeds
init_num(1, 1, 0, 1, 100)  # => succeeds

# watch out for:
init_num(1, 1, 0, 1)      # => succeeds (with N computed)
init_num(1, 1, NA, 1, 100)  # => succeeds (with spec computed)
init_num(1, 1, NA, NA, 100)  # => succeeds (with fart computed)
init_num(1, 1, NA, 1)     # => succeeds (with spec and N computed)
init_num(1, 1, 0, NA)     # => succeeds (with fart and N computed)
init_num(1, 1, .51, .50, 100)  # => succeeds (as spec and fart are within tolарated range)

# ways to fail:
init_num(prev = NA)        # => NAs + warning (NA)
init_num(prev = 88)        # => NAs + warning (beyond range)
### Description

*init_pal* initializes basic color information (i.e., all colors corresponding to functional roles in the current scenario and used throughout the *riskyr* package).

### Usage

```r
init_pal(N_col = pal_def["N"], cond_true_col = pal_def["cond_true"],
         cond_false_col = pal_def["cond_false"],
         dec_pos_col = pal_def["dec_pos"], dec_neg_col = pal_def["dec_neg"],
         dec_cor_col = pal_def["dec_cor"], dec_err_col = pal_def["dec_err"],
         hi_col = pal_def["hi"], mi_col = pal_def["mi"],
         fa_col = pal_def["fa"], cr_col = pal_def["cr"],
         PPV_col = pal_def["ppv"], NPV_col = pal_def["npv"],
         txt_col = pal_def["txt"], brd_col = pal_def["brd"])
```

### Arguments

- **N_col**  
  Color representing the *population* of *N* cases or individuals.

- **cond_true_col**  
  Color representing cases of *cond_true*, for which the current condition is TRUE.

- **cond_false_col**  
  Color representing cases of in *cond_false*, for which the current condition is FALSE.

- **dec_pos_col**  
  Color representing cases of *dec_pos*, for which the current decision is positive.

- **dec_neg_col**  
  Color representing cases in *dec_neg*, for which the current decision is negative.

- **dec_cor_col**  
  Color representing cases of correct decisions *dec_cor*, for which the current decision is accurate.

- **dec_err_col**  
  Color representing cases in erroneous decisions *dec_err*, for which the current decision is inaccurate.

- **hi_col**  
  Color representing *hits* or true positives in *hi* (i.e., correct cases for which the current condition is TRUE and the decision is positive).

- **mi_col**  
  Color representing *misses* or false negatives in *mi* (i.e., incorrect cases for which the current condition is TRUE but the decision is negative).

- **fa_col**  
  Color representing *false alarms* or false positives in *fa* (i.e., incorrect cases for which the current condition is FALSE but the decision is positive).

- **cr_col**  
  Color representing *correct rejections* or true negatives in *cr* (i.e., correct cases for which the current condition is FALSE and the decision is negative).
**PPV_col**  Color representing *positive predictive values* PPV (i.e., the conditional probability that the condition is TRUE, provided that the decision is positive).

**NPV_col**  Color representing *negative predictive values* NPV (i.e., the conditional probability that the condition is FALSE, provided that the decision is negative).

**txt_col** Color used for text labels.

**brd_col** Color used for borders (e.g., around bars or boxes).

**Details**

All color information of the current scenario is stored as named colors in a list `pal`. `init_pal` allows changing colors by assigning new colors to existing names.

**See Also**

`num` contains basic numeric parameters; `init_num` initializes basic numeric parameters; `txt` contains current text information; `init_txt` initializes text information; `pal` contains current color information; `init_pal` initializes color information; `freq` contains current frequency information; `comp_freq` computes current frequency information; `prob` contains current probability information; `comp_prob` computes current probability information.

Other functions initializing scenario information: `init_num`, `init_txt`, `riskyr`

**Examples**

```r
init_pal()  # => define and return a vector of current (default) colors
length(init_pal())  # => 15 named colors
pal <- init_pal(N_col = "steelblue4")  # => change a color (stored in pal)
pal <- init_pal(brd_col = NA)  # => remove a color
```

**init_txt**  *Initialize basic text elements.*

**Description**

`init_txt` initializes basic text elements `txt` (i.e., all titles and labels corresponding to the current scenario) that are used throughout the `riskyr` package.

**Usage**

```r
init_txt(scen_lbl = txt_lbl_def$scen_lbl, 
  scen_txt = txt_lbl_def$scen_txt, scen_src = txt_lbl_def$scen_src, 
  scen_apa = txt_lbl_def$scen_apa, scen_lng = txt_lbl_def$scen_lng, 
  popu_lbl = txt_lbl_def$popu_lbl, N_lbl = txt_lbl_def$N_lbl, 
  cond_lbl = txt_lbl_def$condLbl, 
  cond_true_lbl = txt_lbl_def$cond_true_lbl, 
  cond_false_lbl = txt_lbl_def$cond_falselbl, 
  dec_lbl = txt_lbl_def$dec_lbl, dec_pos_lbl = txt_lbl_def$dec_pos_lbl, 
```
dec_negLbl = txtLbl_def$dec_negLbl, accLbl = txtLbl_def$accLbl,
dec_corLbl = txtLbl_def$dec_corLbl,
dec_errLbl = txtLbl_def$dec_errLbl, sdtLbl = txtLbl_def$sdtLbl,
hiLbl = txtLbl_def$hiLbl, miLbl = txtLbl_def$miLbl,
faLbl = txtLbl_def$faLbl, crLbl = txtLbl_def$crLbl)

Arguments

scenLbl The current scenario title (sometimes in Title Caps).
scen_txt A longer text description of the current scenario (which may extend over several lines).
scen_src The source information for the current scenario.
scen_apa Source information in APA format.
scen_lng Language of the current scenario (as character code). Options: "en": English, "de": German.
popuLbl A general name describing the current population.
NLbl A brief label for the current population popu or sample.
condLbl A general name for the condition dimension currently considered (e.g., some clinical condition).
cond_trueLbl A short label for the presence of the current condition or cond_true cases (the condition's true state of TRUE).
cond_falseLbl A short label for the absence of the current condition or cond_false cases (the condition's true state of FALSE).
decLbl A general name for the decision dimension (e.g., some diagnostic test) currently made.
dec_posLbl A short label for positive decisions or dec_pos cases (e.g., predicting the presence of the condition).
dec_negLbl A short label for negative decisions or dec_neg cases (e.g., predicting the absence of the condition).
accLbl A general name for the accuracy dimension (e.g., correspondence of decision to condition).
dec_corLbl A short label for correct decisions or dec_cor cases (e.g., accurately predicting the condition).
dec_errLbl A short label for erroneous decisions or dec_err cases (e.g., inaccurately predicting the condition).
sdtLbl A name for the case/category/cell dimension in the 2x2 contingency table (SDT: condition x decision).
hiLbl A short label for hits or true positives hi (i.e., correct decisions of the presence of the condition, when the condition is actually present).
miLbl A short label for misses or false negatives mi (i.e., incorrect decisions of the absence of the condition when the condition is actually present).
faLbl A short label for false alarms or false positives fa (i.e., incorrect decisions of the presence of the condition when the condition is actually absent).
crLbl A short label for correct rejections or true negatives cr (i.e., a correct decision of the absence of the condition, when the condition is actually absent).
Details

All textual elements that specify titles and details of the current scenario are stored as named elements (of type character) in a list `txt`. `init_txt` allows changing elements by assigning new character objects to existing names.

However, you can directly specify scenario-specific text elements when defining a scenario with the `riskyr` function.

See Also

txt for current text settings; pal for current color settings; num for basic numeric parameters.

Other functions initializing scenario information: init_num, init_pal, riskyr

Examples

```r
init_txt()
# defines a list of (default) text elements
length(init_txt()) # 21

# Customizing current text elements:
txt <- init_txt(scen_lbl = "My scenario",
                scen_src = "My source",
                N_lbl = "My population")
```

---

is_complement	Verify that two numbers are complements.

Description

is_complement is a function that takes 2 numeric arguments (typically probabilities) as inputs and verifies that they are complements (i.e., add up to 1, within some tolerance range tol).

Usage

```r
is_complement(p1, p2, tol = 0.01)
```

Arguments

- **p1**
  A numeric argument (typically probability in range from 0 to 1).
- **p2**
  A numeric argument (typically probability in range from 0 to 1).
- **tol**
  A numeric tolerance value. Default: `tol = .01`.
Details

Both \( p_1 \) and \( p_2 \) are necessary arguments. If one or both arguments are NA, \( \text{is_complement} \) returns NA (i.e., neither TRUE nor FALSE).

The argument \( \text{tol} \) is optional (with a default value of .01) Numeric near-complements that differ by less than this value are still considered to be complements.

This function does not verify the type, range, or sufficiency of the inputs provided. See \( \text{is_prob} \) and \( \text{is_suff_prob_set} \) for this purpose.

Value

NA or a Boolean value: NA if one or both arguments are NA; TRUE if both arguments are provided and complements (in \( \text{tol} \) range); otherwise FALSE.

See Also

\( \text{comp_complement} \) computes a probability’s complement; \( \text{comp_comp_pair} \) computes pairs of complements; \( \text{num} \) contains basic numeric variables; \( \text{init_num} \) initializes basic numeric variables; \( \text{prob} \) contains current probability information; \( \text{comp_prob} \) computes current probability information; \( \text{freq} \) contains current frequency information; \( \text{comp_freq} \) computes current frequency information; \( \text{is_valid_prob_set} \) verifies the validity of probability inputs; \( \text{as_pc} \) displays a probability as a percentage; \( \text{as_pb} \) displays a percentage as probability.

Other verification functions: \( \text{is_extreme_prob_set}, \text{is_freq}, \text{is_perc}, \text{is_prob}, \text{is_suff_prob_set}, \text{is_valid_prob_pair}, \text{is_valid_prob_set}, \text{is_valid_prob_triple} \)

Examples

# Basics:
\[
\begin{align*}
\text{is_complement}(0, 1) & \quad \# => \text{TRUE} \\
\text{is_complement}(1/3, 2/3) & \quad \# => \text{TRUE} \\
\text{is_complement}(0.33, 0.66) & \quad \# => \text{TRUE} \quad \text{(as within default \text{tol} = .01)} \\
\text{is_complement}(0.33, 0.65) & \quad \# => \text{FALSE} \quad \text{(as beyond default \text{tol} = .01)}
\end{align*}
\]

# watch out for:
\[
\begin{align*}
\text{is_complement}(\text{NA}, \text{NA}) & \quad \# => \text{NA} \quad \text{(but not FALSE)} \\
\text{is_complement}(1, \text{NA}) & \quad \# => \text{NA} \quad \text{(but not FALSE)} \\
\text{is_complement}(2, -1) & \quad \# => \text{TRUE} + \text{warnings} \quad \text{(\( p_1 \) and \( p_2 \) beyond range)} \\
\text{is_complement}(8, -7) & \quad \# => \text{TRUE} + \text{warnings} \quad \text{(\( p_1 \) and \( p_2 \) beyond range)} \\
\text{is_complement}(0.3, 0.6) & \quad \# => \text{FALSE} + \text{warning} \quad \text{(beyond tolerance)} \\
\text{is_complement}(0.3, 0.6, \text{tol} = 0.1) & \quad \# => \text{TRUE} \quad \text{(due to increased tolerance)}
\end{align*}
\]

# ways to fail:
\[
\begin{align*}
\# \text{is_complement}(0, 0) & \quad \# => \text{FALSE} + \text{warning} \quad \text{(beyond tolerance)} \\
\# \text{is_complement}(1, 1) & \quad \# => \text{FALSE} + \text{warning} \quad \text{(beyond tolerance)} \\
\# \text{is_complement}(8, 8) & \quad \# => \text{FALSE} + \text{warning} \quad \text{(beyond tolerance)}
\end{align*}
\]
is_extreme_prob_set

Verify that a set of probabilities describes an extreme case.

Description

is_extreme_prob_set verifies that a set of probabilities (i.e., prev, and sens or mirt, and spec or fart) describe an extreme case.

Usage

is_extreme_prob_set(prev, sens = NA, mirt = NA, spec = NA, fart = NA)

Arguments

prev  The condition’s prevalence value prev (i.e., the probability of condition being TRUE).

sens  The decision’s sensitivity sens (i.e., the conditional probability of a positive decision provided that the condition is TRUE). sens is optional when is complement mirt is provided.

mirt  The decision’s miss rate mirt (i.e., the conditional probability of a negative decision provided that the condition is TRUE). mirt is optional when is complement sens is provided.

spec  The decision’s specificity spec (i.e., the conditional probability of a negative decision provided that the condition is FALSE). spec is optional when is complement fart is provided.

fart  The decision’s false alarm rate fart (i.e., the conditional probability of a positive decision provided that the condition is FALSE). fart is optional when its complement spec is provided.

Details

If TRUE, a warning message describing the nature of the extreme case is printed to allow anticipating peculiar effects (e.g., that PPV or NPV values cannot be computed or are NaN).

This function does not verify the type, range, sufficiency, or consistency of its arguments. See is_prob, is_suff_prob_set, is_complement, is_valid_prob_pair and is_valid_prob_set for these purposes.

Value

A Boolean value: TRUE if an extreme case is identified; otherwise FALSE.
See Also

`is_valid_prob_pair` verifies that a pair of probabilities can be complements; `is_valid_prob_set` verifies the validity of a set of probability inputs; `num` contains basic numeric variables; `init_num` initializes basic numeric variables; `prob` contains current probability information; `comp_prob` computes current probability information; `freq` contains current frequency information; `comp_freq` computes current frequency information; `as_pc` displays a probability as a percentage; `as_pb` displays a percentage as probability

Other verification functions: `is_complement, is_freq, is_perc, is_prob, is_suff_prob_set, is_valid_prob_pair, is_valid_prob_set, is_valid_prob_triple`

Examples

```r
# Identify 6 extreme cases (+ 4 variants):
is_extreme_prob_set(1, 1, NA, 1, NA)  # => TRUE + warning: N true positives
plot_tree(1, 1, NA, 1, NA, N = 100)  # => illustrates this case

is_extreme_prob_set(1, 0, NA, 1, NA)  # => TRUE + warning: N false negatives
plot_tree(1, 0, NA, 1, NA, N = 200)  # => illustrates this case

sens <- .50
is_extreme_prob_set(0, sens, NA, 0, NA)  # => TRUE + warning: N false positives
plot_tree(0, sens, NA, 0, N = 300)  # => illustrates this case
# Variant:
is_extreme_prob_set(0, sens, NA, NA, 1)  # => TRUE + warning: N false positives
plot_tree(0, sens, NA, NA, 1, N = 350)  # => illustrates this case

sens <- .50
is_extreme_prob_set(0, sens, NA, 1)  # => TRUE + warning: N true negatives
plot_tree(0, sens, NA, 1, N = 400)  # => illustrates this case
# Variant:
is_extreme_prob_set(0, sens, NA, NA, 0)  # => TRUE + warning: N true negatives
plot_tree(0, sens, NA, NA, 0, N = 450)  # => illustrates this case

prev <- .50
is_extreme_prob_set(prev, 0, NA, 1, NA)  # => TRUE + warning: 0 hi and 0 fa (0 dec_pos cases)
plot_tree(prev, 0, NA, 1, NA, N = 500)  # => illustrates this case
# # Variant:
is_extreme_prob_set(prev, 0, NA, 0)  # => TRUE + warning: 0 hi and 0 fa (0 dec_pos cases)
plot_tree(prev, 0, NA, 0, N = 550)  # => illustrates this case

prev <- .50
is_extreme_prob_set(prev, 1, NA, 0, NA)  # => TRUE + warning: 0 mi and 0 cr (0 dec_neg cases)
plot_tree(prev, 1, NA, 0, NA, N = 600)  # => illustrates this case
# # Variant:
is_extreme_prob_set(prev, 1, NA, 0)  # => TRUE + warning: 0 mi and 0 cr (0 dec_neg cases)
plot_tree(prev, 1, NA, 0, N = 650)  # => illustrates this case
```
**is_freq**  
Verify that input is a frequency (positive integer value).

### Description

`is_freq` is a function that checks whether its single argument `freq` is a frequency (i.e., a positive numeric integer value).

### Usage

```
is_freq(freq)
```

### Arguments

- **freq**  
  A single (typically numeric) argument.

### Value

A Boolean value: TRUE if `freq` is a frequency (positive integer), otherwise FALSE.

### See Also

- `num` contains basic numeric variables; `init_num` initializes basic numeric variables;  
  `prob` contains current probability information; `comp_prob` computes current probability information;  
  `freq` contains current frequency information; `comp_freq` computes current frequency information;  
  `is_valid_prob_set` verifies the validity of probability inputs; `as_pc` displays a probability as a percentage;  
  `as_pb` displays a percentage as probability.

Other verification functions: `is_complement, is_extreme_prob_set, is_perc, is_prob, is_suff_prob_set,  
`is_valid_prob_pair, is_valid_prob_set, is_valid_prob_triple`

### Examples

```
# ways to succeed:
is_freq(2)  # => TRUE, but does NOT return the frequency 2.
is_freq(0:3)  # => TRUE (for vector)

## ways to fail:
# is_freq(-1)  # => FALSE + warning (negative values)
# is_freq(1:-1)  # => FALSE (for vector) + warning (negative values)
# is_freq(c(1, 1.5, 2))  # => FALSE (for vector) + warning (non-integer values)

## note:
# is.integer(2)  # => FALSE!
```
is_perc is a function that checks whether its single argument `perc` is a percentage (proportion, i.e., a numeric value in the range from 0 to 100).

Usage

`is_perc(perc)`

Arguments

`perc`  
A single (typically numeric) argument.

Value

A Boolean value: `TRUE` if `perc` is a percentage (proportion), otherwise `FALSE`.

See Also

`num` contains basic numeric variables; `init_num` initializes basic numeric variables; `prob` contains current probability information; `comp_prob` computes current probability information; `freq` contains current frequency information; `comp_freq` computes current frequency information; `is_valid_prob_set` verifies the validity of probability inputs; `as_pc` displays a probability as a percentage; `as_pb` displays a percentage as probability.

Other verification functions: `is_complement`, `is_extreme_prob_set`, `is_freq`, `is_prob`, `is_suff_prob_set`, `is_valid_prob_pair`, `is_valid_prob_set`, `is_valid_prob_triple`

Examples

```r
# ways to succeed:
is_perc(2)  # => TRUE, but does NOT return the percentage 2.
is_perc(1/2)  # => TRUE, but does NOT return the percentage 0.5.

## note:  
# pc_sq <- seq(0, 100, by = 10)  
# is_perc(pc_sq)  # => TRUE (for vector)

## ways to fail:  
# is_perc(NA)  # => FALSE + warning (NA values)  
# is_perc(NaN)  # => FALSE + warning (NaN values)  
# is_perc("Bernoulli")  # => FALSE + warning (non-numeric values)  
# is_perc(101)  # => FALSE + warning (beyond range)
```
is_prob

Verify that input is a probability (numeric value from 0 to 1).

Description

is_prob is a function that checks whether its argument prob is a probability (i.e., a numeric value in the range from 0 to 1).

Usage

is_prob(prob, NA_warn = FALSE)

Arguments

prob
A numeric argument (scalar or vector) that is to be checked.

NA_warn
Boolean value determining whether a warning is shown for NA values. Default: NA_warn = FALSE.

Value

A Boolean value: TRUE if prob is a probability, otherwise FALSE.

See Also

num contains basic numeric variables; init_num initializes basic numeric variables; prob contains current probability information; comp_prob computes current probability information; freq contains current frequency information; comp_freq computes current frequency information; is_valid_prob_set verifies the validity of probability inputs; as_pc displays a probability as a percentage; as_pb displays a percentage as probability.

Other verification functions: is_complement, is_extreme_prob_set, is_freq, is_perc, is_suff_prob_set, is_valid_prob_pair, is_valid_prob_set, is_valid_prob_triple

Examples

# ways to succeed:
is_prob(1/2) # => TRUE
p.seq <- seq(0, 1, by = .1) # Vector of probabilities
is_prob(p.seq) # => TRUE (for vector)

## watch out for:
# is_prob(NA) # => FALSE + NO warning!
# is_prob(0/0) # => FALSE + NO warning (NA + NaN values)
# is_prob(0/0, NA_warn = TRUE) # => FALSE + warning (NA values)

## ways to fail:
# is_prob(8, NA_warn = TRUE) # => FALSE + warning (outside range element)
# is_prob(c(.5, 8), NA_warn = TRUE) # => FALSE + warning (outside range vector element)
# is_prob("Laplace", NA_warn = TRUE) # => FALSE + warning (non-numeric values)
is_suff_prob_set

Verify a sufficient set of probability inputs.

Description

is_suff_prob_set is a function that takes 3 to 5 probabilities as inputs and verifies that they are sufficient to compute all derived probabilities and combined frequencies for a population of N individuals.

Usage

is_suff_prob_set(prev, sens = NA, mirt = NA, spec = NA, fart = NA)

Arguments

- **prev**: The condition’s prevalence *prev* (i.e., the probability of condition being TRUE).
- **sens**: The decision’s sensitivity *sens* (i.e., the conditional probability of a positive decision provided that the condition is TRUE). sens is optional when its complement mirt is provided.
- **mirt**: The decision’s miss rate *mirt* (i.e., the conditional probability of a negative decision provided that the condition is TRUE). mirt is optional when its complement sens is provided.
- **spec**: The decision’s specificity value *spec* (i.e., the conditional probability of a negative decision provided that the condition is FALSE). spec is optional when its complement fart is provided.
- **fart**: The decision’s false alarm rate *fart* (i.e., the conditional probability of a positive decision provided that the condition is FALSE). fart is optional when its complement spec is provided.

Details

While no alternative input option for frequencies is provided, specification of the essential probability *prev* is always necessary.

However, for 2 other essential probabilities there is a choice:

1. either *sens* or *mirt* is necessary (as both are complements).
2. either *spec* or *fart* is necessary (as both are complements).

is_suff_prob_set does not verify the type, range, or consistency of its arguments. See *is_prob* and *is_complement* for this purpose.

Value

A Boolean value: TRUE if the probabilities provided are sufficient, otherwise FALSE.
is_valid_prob_pair

See Also

num contains basic numeric variables; init_num initializes basic numeric variables; prob contains current probability information; comp_prob computes current probability information; freq contains current frequency information; comp_freq computes current frequency information; is_valid_prob_set verifies the validity of probability inputs; as_pc displays a probability as a percentage; as_pb displays a percentage as probability.

Other verification functions: is_complement, is_extreme_prob_set, is_freq, is_perc, is_prob, is_valid_prob_pair, is_valid_prob_set, is_valid_prob_triple

Examples

# ways to work:
is_suff_prob_set(prev = 1, sens = 1, spec = 1) # => TRUE
is_suff_prob_set(prev = 1, mirt = 1, spec = 1) # => TRUE
is_suff_prob_set(prev = 1, sens = 1, fart = 1) # => TRUE
is_suff_prob_set(prev = 1, mirt = 1, fart = 1) # => TRUE

# watch out for:
is_suff_prob_set(prev = 1, sens = 2, spec = 3) # => TRUE, but is_prob is FALSE
is_suff_prob_set(prev = 1, mirt = 2, fart = 4) # => TRUE, but is_prob is FALSE
is_suff_prob_set(prev = 1, sens = 2, spec = 3, fart = 4) # => TRUE, but is_prob is FALSE

## ways to fail:
# is_suff_prob_set() # => FALSE + warning (prev missing)
# is_suff_prob_set(prev = 1) # => FALSE + warning (sens or mirt missing)
# is_suff_prob_set(prev = 1, sens = 1) # => FALSE + warning (spec or fart missing)

---

is_valid_prob_pair Verify that a pair of probability inputs can be a pair of complementary probabilities.

Description

is_valid_prob_pair is a function that verifies that a pair of 2 numeric inputs p1 and p2 can be interpreted as a valid pair of probabilities.

Usage

is_valid_prob_pair(p1, p2, tol = 0.01)

Arguments

p1 A numeric argument (typically probability in range from 0 to 1).
p2 A numeric argument (typically probability in range from 0 to 1).
tol A numeric tolerance value.
Details

`is_valid_prob_pair` is a wrapper function that combines `is_prob` and `is_complement` in one function.

Either p1 or p2 must be a probability (verified via `is_prob`). If both arguments are provided they must be probabilities and complements (verified via `is_complement`).

The argument `tol` is optional (with a default value of .01) Numeric near-complements that differ by less than this value are still considered to be complements.

Value

A Boolean value: TRUE if exactly one argument is a probability, if both arguments are probabilities and complements, otherwise FALSE.

See Also

`is_valid_prob_set` uses this function to verify sets of probability inputs; `is_complement` verifies numeric complements; `is_prob` verifies probabilities; `num` contains basic numeric variables; `init_num` initializes basic numeric variables; `prob` contains current probability information; `comp_prob` computes current probability information; `freq` contains current frequency information; `comp_freq` computes current frequency information; `as_pc` displays a probability as a percentage; `as_pb` displays a percentage as probability.

Other verification functions: `is_complement`, `is_extreme_prob_set`, `is_freq`, `is_perc`, `is_prob`, `is_suff_prob_set`, `is_valid_prob_set`, `is_valid_prob_triple`

Examples

```r
# ways to succeed:
is_valid_prob_pair(1, 0)  # => TRUE
is_valid_prob_pair(0, 1)  # => TRUE
is_valid_prob_pair(1, NA) # => TRUE + warning (NA)
is_valid_prob_pair(NA, 1) # => TRUE + warning (NA)
is_valid_prob_pair(.50, .51) # => TRUE (as within tol)

# ways to fail:
is_valid_prob_pair(.50, .52) # => FALSE (as beyond tol)
is_valid_prob_pair(1, 2)    # => FALSE + warning (beyond range)
is_valid_prob_pair(NA, NA)  # => FALSE + warning (NA)
```

---

**is_valid_prob_set**

Verify that a set of probability inputs is valid.

Description

`is_valid_prob_set` is a function that verifies that a set of (3 to 5) numeric inputs can be interpreted as a valid set of (3 essential and 2 optional) probabilities.
is_valid_prob_set

Usage

is_valid_prob_set(prev, sens = NA, mirt = NA, spec = NA, fart = NA, tol = 0.01)

Arguments

prev  The condition’s prevalence prev (i.e., the probability of condition being TRUE).
sens  The decision’s sensitivity sens (i.e., the conditional probability of a positive
decision provided that the condition is TRUE). sens is optional when its comple-
ment mirt is provided.
mirt  The decision’s miss rate mirt (i.e., the conditional probability of a negative deci-
sion provided that the condition is TRUE). mirt is optional when its complement
sens is provided.
spec  The decision’s specificity value spec (i.e., the conditional probability of a nega-
tive decision provided that the condition is FALSE). spec is optional when its com-
plement fart is provided.
fart  The decision’s false alarm rate fart (i.e., the conditional probability of a pos-
tive decision provided that the condition is FALSE). fart is optional when its com-
plement spec is provided.
tol   A numeric tolerance value used by is_complement.

Details

is_valid_prob_set is a wrapper function that combines is_prob, is_suff_prob_set, and is_complement
in one function.

While no alternative input option for frequencies is provided, specification of the essential proba-
bility prev is always necessary. However, for 2 other essential probabilities there is a choice:

1. Either sens or mirt is necessary (as both are complements).
2. Either spec or fart is necessary (as both are complements).

The argument tol is optional (with a default value of .01) and used as the tolerance value of
is_complement.

is_valid_prob_set verifies the validity of inputs, but does not compute or return numeric vari-
ables. Use is_extreme_prob_set to verify sets of probabilities that describe extreme cases and
init_num for initializing basic parameters.

Value

A Boolean value: TRUE if the probabilities provided are valid; otherwise FALSE.

See Also

is_valid_prob_pair verifies that probability pairs are complements; is_prob verifies probabili-
ties; prob contains current probability information; num contains basic numeric variables; init_num
is_valid_prob_triple

initializes basic numeric variables; \texttt{comp\_prob} computes current probability information; \texttt{freq} contains current frequency information; \texttt{comp\_freq} computes current frequency information; \texttt{as\_pc} displays a probability as a percentage; \texttt{as\_pb} displays a percentage as probability.

Other verification functions: \texttt{is\_complement}, \texttt{is\_extreme\_prob\_set}, \texttt{is\_freq}, \texttt{is\_perc}, \texttt{is\_prob}, \texttt{is\_suff\_prob\_set}, \texttt{is\_valid\_prob\_pair}, \texttt{is\_valid\_prob\_triple}

Examples

\# ways to succeed:
is_valid_prob_set(1, 1, 0, 1, 0)  \quad \# \Rightarrow \text{TRUE}
is_valid_prob_set(3, .9, .1, .8, .2)  \quad \# \Rightarrow \text{TRUE}
is_valid_prob_set(3, .9, .1, .8, NA)  \quad \# \Rightarrow \text{TRUE} + \text{warning (NA)}
is_valid_prob_set(3, .9, NA, .8, NA)  \quad \# \Rightarrow \text{TRUE} + \text{warning (NAs)}
is_valid_prob_set(3, .9, NA, NA, .8)  \quad \# \Rightarrow \text{TRUE} + \text{warning (NAs)}
is_valid_prob_set(3, .8, .1, .7, .2, tol = .1)  \quad \# \Rightarrow \text{TRUE} (\text{due to increased tol})

\# watch out for:
is_valid_prob_set(1, 0, 1, 0, 1)  \quad \# \Rightarrow \text{TRUE}, but NO warning about extreme case!
is_valid_prob_set(1, 1, 0, 1, 0)  \quad \# \Rightarrow \text{TRUE}, but NO warning about extreme case!
is_valid_prob_set(1, 1, 0, NA, 1)  \quad \# \Rightarrow \text{TRUE}, but NO warning about extreme case!
is_valid_prob_set(1, 1, 0, NA, 0)  \quad \# \Rightarrow \text{TRUE}, but NO warning about extreme case!

\# ways to fail:
is_valid_prob_set(8, 1, 0, 1, 0)  \quad \# \Rightarrow \text{FALSE} + \text{warning (is\_prob fails)}
is_valid_prob_set(1, 1, 8, 1, 0)  \quad \# \Rightarrow \text{FALSE} + \text{warning (is\_prob fails)}
is_valid_prob_set(2, 1, 3, 1, 4)  \quad \# \Rightarrow \text{FALSE} + \text{warning (is\_prob fails)}
is_valid_prob_set(1, .8, .2, .7, .2)  \quad \# \Rightarrow \text{FALSE} + \text{warning (beyond complement range)}
is_valid_prob_set(1, .8, .3, .7, .3)  \quad \# \Rightarrow \text{FALSE} + \text{warning (beyond complement range)}
is_valid_prob_set(1, 1, 1, 1, 1)  \quad \# \Rightarrow \text{FALSE} + \text{warning (beyond complement range)}
is_valid_prob_set(1, 1, 0, 1, 1)  \quad \# \Rightarrow \text{FALSE} + \text{warning (beyond complement range)}

\begin{Verbatim}

\textbf{is\_valid\_prob\_triple} \quad \textit{Verify that a triple of essential probability inputs is valid.}

\end{Verbatim}

\textbf{Description}

\texttt{is\_valid\_prob\_triple} is a \textbf{deprecated} function that verifies that a set of 3 numeric inputs can be interpreted as a valid set of 3 probabilities.

\textbf{Usage}

\texttt{is\_valid\_prob\_triple(prev, sens, spec)
is_valid_prob_triple

Arguments

prev The condition’s prevalence prev (i.e., the probability of condition being TRUE).
sens The decision’s sensitivity sens (i.e., the conditional probability of a positive decision provided that the condition is TRUE).
spec The decision’s specificity value spec (i.e., the conditional probability of a negative decision provided that the condition is FALSE).

Details

is_valid_prob_triple is a simplified version of is_valid_prob_set. It is a quick wrapper function that only verifies is_prob for all of its 3 arguments.

is_valid_prob_triple does not compute or return numeric variables. Use is_extreme_prob_set to verify extreme cases and comp_complete_prob_set to complete sets of valid probabilities.

Value

A Boolean value: TRUE if the probabilities provided are valid; otherwise FALSE.

See Also

is_extreme_prob_set verifies extreme cases; is_valid_prob_set verifies sets of probability inputs; is_valid_prob_pair verifies that probability pairs are complements; num contains basic numeric variables; init_num initializes basic numeric variables; prob contains current probability information; comp_prob computes current probability information; freq contains current frequency information; comp_freq computes current frequency information; as_pc displays a probability as a percentage; as_pb displays a percentage as probability.

Other verification functions: is_complement, is_extreme_prob_set, is_freq, is_perc, is_prob, is_suff_prob_set, is_valid_prob_pair, is_valid_prob_set

Examples

# ways to work:
is_valid_prob_triple(0, 0, 0)   # => TRUE
is_valid_prob_triple(1, 1, 1)   # => TRUE

## ways to fail:
# is_valid_prob_triple(0, 0)      # => ERROR (as no triple)
# is_valid_prob_triple(0, 0, 7)  # => FALSE + warning (beyond range)
# is_valid_prob_triple(0, NA, 0) # => FALSE + warning (NA)
# is_valid_prob_triple("p", 0, 0) # => FALSE + warning (non-numeric)
mi

Frequency of misses or false negatives (FN).

Description

mi is the frequency of misses or false negatives (FN) in a population of N individuals.

Usage

mi

Format

An object of class numeric of length 1.

Details

Definition: mi is the frequency of individuals for which Condition = TRUE and Decision = FALSE (negative).

mi is a measure of incorrect classifications (type-II errors), not an individual case.

Relationships:

1. to probabilities: The frequency mi depends on the miss rate mirt (aka. false negative rate, FNR) and is conditional on the prevalence prev.

2. to other frequencies: In a population of size N the following relationships hold:
   - N = cond_true + cond_false (by condition)
   - N = dec_pos + dec_neg (by decision)
   - N = dec_cor + dec_err (by correspondence of decision to condition)
   - N = hi + mi + fa + cr (by condition x decision)

See Also

mirt is the probability or rate of misses; num contains basic numeric parameters; init_num initializes basic numeric parameters; freq contains current frequency information; comp_freq computes current frequency information; prob contains current probability information; comp_prob computes current probability information; is_freq verifies frequencies.

Other essential parameters: cr, fa, hi, prev, sens, spec

Other frequencies: N, cond_false, cond_true, cr, dec_cor, dec_err, dec_neg, dec_pos, fa, hi
**mirt**

The miss rate of a decision process or diagnostic procedure.

**Description**

*mirt* defines a decision’s miss rate value: The conditional probability of the decision being negative if the condition is TRUE.

**Usage**

*mirt*

**Format**

An object of class `numeric` of length 1.

**Details**

Understanding or obtaining the miss rate *mirt*:

- **Definition:** *sens* is the conditional probability for an incorrect negative decision given that the condition is TRUE:
  
  \[
  \text{mirt} = p(\text{decision} = \text{negative} \mid \text{condition} = \text{TRUE})
  \]
  
  or the probability of failing to detect true cases (condition = TRUE).

- **Perspective:** *mirt* further classifies the subset of *cond_true* individuals by decision (*mirt = mi/cond_true*).

- **Alternative names:** false negative rate (FNR), rate of type-II errors (beta)

- **Relationships:**
  a. *mirt* is the complement of the sensitivity *sens* (aka. hit rate HR):
     \[
     \text{mirt} = (1 - \text{sens}) = (1 - \text{HR})
     \]
  b. *mirt* is the _opposite_ conditional probability – but not the complement – of the false omission rate *FOR*:
     \[
     \text{FOR} = p(\text{condition} = \text{TRUE} \mid \text{decision} = \text{negative})
     \]

- **In terms of frequencies,** *mirt* is the ratio of *mi* divided by *cond_true* (i.e., *hi + mi*):
  \[
  \text{mirt} = \frac{\text{mi}}{\text{cond_true}} = \frac{\text{mi}}{\text{hi + mi}}
  \]

- **Dependencies:** *mirt* is a feature of a decision process or diagnostic procedure and a measure of incorrect decisions (false negatives).

  However, due to being a conditional probability, the value of *mirt* is not intrinsic to the decision process, but also depends on the condition’s prevalence value *prev*.

**References**

Consult Wikipedia for additional information.
See Also

- `comp_mirt` computes `mirt` as the complement of `sens`; `prob` contains current probability information; `comp_prob` computes current probability information; `num` contains basic numeric parameters; `init_num` initializes basic numeric parameters; `comp_freq` computes current frequency information; `is_prob` verifies probabilities.

Other probabilities: FDR, FOR, NPV, PPV, acc, err, fart, ppod, prev, sens, spec

Examples

```r
mirt <- .15  # => sets a miss rate of 15%
mirt <- 15/100 # => (decision = negative) for 15 out of 100 people with (condition = TRUE)
```

---

N | Number of individuals in the population.
---

Description

N is a frequency that describes the number of individuals in the current population (i.e., the overall number of cases considered).

Usage

N

Format

An object of class numeric of length 1.

Details

Key relationships between frequencies and probabilities (see documentation of `comp_freq` or `comp_prob` for details):

- Three perspectives on a population:
  by condition / by decision / by accuracy.
- Defining probabilities in terms of frequencies:
  Probabilities can be computed as ratios between frequencies, but beware of rounding issues.

Current frequency information is computed by `comp_freq` and contained in a list `freq`.

References

Consult Wikipedia: Statistical population for additional information.
NPV

See Also

- `is_freq` verifies frequencies;
- `num` contains basic numeric parameters;
- `init_num` initializes basic numeric parameters;
- `freq` contains current frequency information;
- `comp_freq` computes current frequency information;
- `prob` contains current probability information;
- `comp_prob` computes current probability information.

Other frequencies:
- `cond_false`,
- `cond_true`,
- `cr`,
- `dec_cor`,
- `dec_err`,
- `dec_neg`,
- `dec_pos`,
- `fa`,
- `hi`,
- `mi`.

Examples

```r
N <- 1000  # => sets a population size of 1000
is_freq(N) # => TRUE
is_prob(N) # => FALSE (as N is no probability)
```

NPV

The negative predictive value of a decision process or diagnostic procedure.

Description

NPV defines some decision’s negative predictive value (NPV): The conditional probability of the condition being FALSE provided that the decision is negative.

Usage

NPV

Format

An object of class `numeric` of length 1.

Details

Understanding or obtaining the negative predictive value NPV:

- Definition: NPV is the conditional probability for the condition being FALSE given a negative decision:
  \[ NPV = p(\text{condition} = \text{FALSE} \mid \text{decision} = \text{negative}) \]
  or the probability of a negative decision being correct.

- Perspective: NPV further classifies the subset of `dec_neg` individuals by condition
  \[ NPV = \text{cr}/\text{dec_neg} = \text{cr}/(\text{mi} + \text{cr}) \]

- Alternative names: true omission rate

- Relationships:
  a. NPV is the complement of the false omission rate `FOR`:
  \[ NPV = 1 - \text{FOR} \]
  b. NPV is the opposite conditional probability – but not the complement – of the specificity `spec`:
  \[ spec = p(\text{decision} = \text{negative} \mid \text{condition} = \text{FALSE}) \]
In terms of frequencies, NPV is the ratio of \( cr \) divided by \( dec\_neg \) (i.e., \( cr + mi \)):
\[
NPV = \frac{cr}{dec\_neg} = \frac{cr}{(cr + mi)}
\]

Dependencies: NPV is a feature of a decision process or diagnostic procedure and – similar to the specificity \( spec \) – a measure of correct decisions (negative decisions that are actually FALSE).
However, due to being a conditional probability, the value of NPV is not intrinsic to the decision process, but also depends on the condition’s prevalence value \( prev \).

References
Consult Wikipedia for additional information.

See Also
\( \text{comp\_NPV} \) computes NPV; \( \text{prob} \) contains current probability information; \( \text{comp\_prob} \) computes current probability information; \( \text{num} \) contains basic numeric parameters; \( \text{init\_num} \) initializes basic numeric parameters; \( \text{comp\_freq} \) computes current frequency information; \( \text{is\_prob} \) verifies probabilities.

Other probabilities: \( \text{FDR, FOR, PPV, acc, err, fart, mirt, ppod, prev, sens, spec} \)

Examples
\[
\begin{align*}
\text{NPV} & \leftarrow .95 & \# \text{ sets a negative predictive value of 95}\% \\
\text{NPV} & \leftarrow 95/100 & \# \text{ (condition = FALSE) for 95 out of 100 people with (decision = negative)} \\
\text{is\_prob}(\text{NPV}) & \leftarrow \text{TRUE}
\end{align*}
\]

\begin{itemize}
  \item \texttt{num}
\end{itemize}

\textit{List current values of basic numeric variables.}

Description
\( \text{num} \) is a list of named numeric variables containing 4 basic probabilities (\( \text{prev, sens, spec, and fart} \)) and 1 frequency parameter (the population size \( N \)).

Usage
\( \text{num} \)

Format
An object of class \texttt{list} of length 5.
See Also

init_num initializes basic numeric parameters; txt contains current text information; init_txt initializes text information; pal contains current color information; init_pal initializes color information; freq contains current frequency information; comp_freq computes current frequency information; prob contains current probability information; comp_prob computes current probability information.

Other lists containing current scenario information: accu, freq, pal_bw, pal_kn, pal_mbw, pal_mod, pal_org, pal_rgb, pal_vir, pal, prob, txt_TF, txt_org, txt

Examples

```r
num <- init_num()  # => initialize num to default parameters
num               # => show defaults
length(num)       # => 5
```

---

pal

*List current values of scenario color palette.*

Description

pal is initialized to a vector of named elements (colors) to define the scenario color scheme that is used throughout the riskyr package.

Usage

pal

Format

An object of class character of length 15.

Details

All color information corresponding to the current scenario is stored as named colors in a vector pal. To change a color, assign a new color to an existing element name.

pal currently contains colors with the following names:

1. N Color representing the population of N cases or individuals.
2. cond_true Color representing cases of cond_true, for which the current condition is TRUE.
3. cond_false Color representing cases of cond_false, for which the current condition is FALSE.
4. dec_pos Color representing cases of dec_pos, for which the current decision is positive.
5. dec_neg Color representing cases in dec_neg, for which the current decision is negative.
6. dec_cor Color representing cases of correct decisions dec_cor, for which the current decision is accurate.
7. **dec_err** Color representing cases of erroneous decisions **dec_err**, for which the current decision is inaccurate.

8. **hi** Color representing **hits** or true positives in **hi** (i.e., correct cases for which the current condition is TRUE and the decision is positive).

9. **mi** Color representing **misses** or false negatives in **mi** (i.e., incorrect cases for which the current condition is TRUE but the decision is negative).

10. **fa** Color representing **false alarms** or false positives in **fa** (i.e., incorrect cases for which the current condition is FALSE but the decision is positive).

11. **cr** Color representing **correct rejections** or true negatives in **cr** (i.e., correct cases for which the current condition is FALSE and the decision is negative).

12. **ppv** Color representing **positive predictive values** PPV (i.e., the conditional probability that the condition is TRUE, provided that the decision is positive).

13. **npv** Color representing **negative predictive values** NPV (i.e., the conditional probability that the condition is FALSE, provided that the decision is negative).

14. **txt** Color used for text labels.

15. **brd** Color used for borders.

Note that color names for frequencies correspond to frequency names, but are different for probabilities (which are written in lowercase and only **ppv** and **npv** have assigned colors).

**See Also**

- `init_pal` initializes color information; `num` contains basic numeric parameters; `init_num` initializes basic numeric parameters; `txt` contains current text information; `init_txt` initializes text information; `freq` contains current frequency information; `comp_freq` computes current frequency information; `prob` contains current probability information; `comp_prob` computes current probability information.

Other lists containing current scenario information: `accu`, `freq`, `num`, `pal_bw`, `pal_kn`, `pal_mbw`, `pal_mod`, `pal_org`, `pal_rgb`, `pal_vir`, `prob`, `txt_TF`, `txt_org`, `txt`.

**Examples**

```r
pal       # shows all current color names and values
pal["hi"] # shows the current color for hits (true positives)
pal["hi"] <- "gold"  # defines a new color for hits (true positives, TP)
```

---

**pal_bw**

*Alternative color palette for black-and-white graphs.*

**Description**

`pal_bw` is initialized to a vector of named elements (colors) to define an alternative (black-and-white, b/w) scenario color scheme.
### Description

`pal_kn` is initialized to a vector of named elements (colors) to define an alternative (uni.kn) scenario color scheme.

### Usage

```r
pal_kn
```

### Format

An object of class `character` of length 15.

### Details

See `pal` for default color information.

Assign `pal <- pal_kn` to use as default color scheme throughout the `riskyr` package.
pal_mbw

See Also

pal contains current color information; init_pal initializes color information.
Other lists containing current scenario information: accu, freq, num, pal_bw, pal_mbw, pal_mod, pal_org, pal_rgb, pal_vir, pal_prob, txt_TF, txt_org, txt

Examples

code

pal_kn # shows all current color names and values
pal_kn["hi"] # shows the current color for hits (true positives)
pal_kn["hi"] <- "grey" # defines a new color for hits (true positives, TP)

date

pal_mbw

Description

pal_mod is initialized to a vector of named colors to define a reduced modern scenario color scheme (in green/blue/bw).

Usage

pal_mbw

Format

An object of class character of length 15.

Details

See pal_org for original color information; pal_mod for a richer modern color palette; and pal_bw for a more reduced black-and-white color palette.
Assign pal <- pal_mbw to use as default color scheme throughout the riskyr package.

See Also

pal contains current color information; init_pal initializes color information; pal_org for original color palette; pal_mod for a richer modern color palette; pal_bw for a more reduced black-and-white color palette.
Other lists containing current scenario information: accu, freq, num, pal_bw, pal_kn, pal_mod, pal_org, pal_rgb, pal_vir, pal_prob, txt_TF, txt_org, txt

Examples

code

pal_mbw # shows all current color names and values
pal_mbw["hi"] # shows the current color for hits (true positives)
pal_mbw["hi"] <- "gold" # defines a new color for hits (true positives, TP)
**pal_mod**

*Modern color palette (in green/blue/orange).*

**Description**

pal_mod is initialized to a vector of named colors to define a modern scenario color scheme (in green/blue/orange).

**Usage**

```r
pal_mod
```

**Format**

An object of class character of length 15.

**Details**

See `pal` for default color information.

Assign `pal <- pal_mod` to use as default color scheme throughout the `riskyr` package.

**See Also**

`pal` contains current color information; `init_pal` initializes color information.

Other lists containing current scenario information: `accu, freq, num, pal_bw, pal_kn, pal_mbw, pal_org, pal_rgb, pal_vir, pal, prob, txt_TF, txt_org, txt`

**Examples**

- `pal_mod` # shows all current color names and values
- `pal_mod["hi"]` # shows the current color for hits (true positives)
- `pal_mod["hi"] <- "gold"` # defines a new color for hits (true positives, TP)

---

**pal_org**

*Original color palette.*

**Description**

pal_org is a copy of `pal` (to retrieve original set of colors in case `pal` is changed).

**Usage**

```r
pal_org
```
Format

An object of class character of length 15.

Details

See `pal` for default color information.
Assign `pal <- pal_org` to re-set default color scheme throughout the `riskyr` package.

See Also

`pal` contains current color information; `init_pal` initializes color information.
Other lists containing current scenario information: `accu`, `freq`, `num`, `pal_bw`, `pal_kn`, `pal_mbw`, `pal_mod`, `pal_rgb`, `pal_vir`, `pal_prob`, `txt_TF`, `txt_org`, `txt`

Examples

```r
pal_org # shows all current color names and values
pal_org["hi"] # shows the current color for hits (true positives)
pal_org["hi"] <- "gold" # defines a new color for hits (true positives, TP)
```

---

`pal_rgb` Alternative color palette for graphs (with RGB colors).

Description

`pal_rgb` is initialized to a vector of named elements (colors) to define an alternative (reduced) scenario color scheme (using red, green, and blue colors).

Usage

`pal_rgb`

Format

An object of class character of length 15.

Details

See `pal` for default color information.
Assign `pal <- pal_rgb` to use as default color scheme throughout the `riskyr` package.

See Also

`pal` contains current color information; `init_pal` initializes color information.
Other lists containing current scenario information: `accu`, `freq`, `num`, `pal_bw`, `pal_kn`, `pal_mbw`, `pal_mod`, `pal_org`, `pal_vir`, `pal_prob`, `txt_TF`, `txt_org`, `txt`
pal_vir

Examples

code
```r
pal_rgb  # shows all current color names and values
pal_rgb["hi"]  # shows the current color for hits (true positives)
pal_rgb["hi"] <- "gold"  # defines a new color for hits (true positives, TP)
```

description

pal_vir is initialized to a vector of named elements (colors) to define a scenario color scheme modeled on the viridis color scale.

Usage

code
```
pal_vir
```

Format

An object of class character of length 15.

Details

These colors are select by the Matplotlib viridis color map created by Stéfan van der Walt and Nathaniel Smith. See the viridislite package (maintained by Simon Garnier) for further information.

Assign `pal <- pal_vir` to use as default color scheme throughout the riskyr package.

See Also

`pal` contains current color information; `init_pal` initializes color information.

Other lists containing current scenario information: `accu, freq, num, pal_bw, pal_kn, pal_mbw, pal_mod, pal_org, pal_rgb, pal_prob, txt_TF, txt_org, txt`

Examples

code
```r
pal_vir  # shows all current color names and values
pal_vir["hi"]  # shows the current color for hits (true positives)
pal_vir["hi"] <- "green3"  # defines a new color for hits (true positives, TP)
```
plot.box

Plot a frequency box object.

Description

plot.box is a utility method that allows to plot low level boxes for riskyr plots.

Usage

```r
## S3 method for class 'box'
plot(x, cur_freq = freq, lbl_txt = txt, col_pal = pal,
     ...)  
```

Arguments

- `x` The box (i.e., an object of class box) to be plotted.
- `cur_freq` Current frequency information (see `freq` for details).
- `lbl_txt` Current text information (see `txt` for details).
- `col_pal` Current color palette (see `pal` for details).
- `...` Additional (graphical) parameters to be passed to the underlying plotting functions.

Details

plot.riskyr also uses the text settings specified in the "riskyr" object.

See Also

Other utility functions: `as_pb`, `as_pc`

plot.riskyr

Plot a riskyr scenario.

Description

plot.riskyr is a method that allows to generate different plot types from a "riskyr" object.

Usage

```r
## S3 method for class 'riskyr'
plot(x = NULL, type = "prism", ...)  
```
Arguments

x

An object of class "riskyr", usually a result of a call to riskyr. Pre-defined scenarios are also of type "riskyr".

type

The type of plot to be generated.

The following plot types are currently available:

1. type = "prism" or type = "net" or type = "tree": Risk information is plotted in a network diagram of frequencies and probabilities (default). See plot_prism for further options.
2. type = "tab" or type = "ftab": Risk information is plotted as a 2-by-2 frequency or contingency table. See plot_tab for further options.
3. type = "area" or type = "mosaic": Risk information is plotted as a mosaic plot (scaled area). See plot_area for further options.
4. type = "bar" or type = "fbar": Risk information is plotted as a bar chart. See plot_bar for further options.
5. type = "icons" or type = "iconarray": The underlying population is plotted as an array of icons. See plot_icons for further options.
6. type = "curve" or type = "curves": Draws curves of selected values (including PPV, NPV). See plot_curve for further options.
7. type = "plane" or type = "planes": Draws a 3D-plane of selected values (e.g., predictive values PPV or NPV). See plot_plane for further options.

... Additional parameters to be passed to the underlying plotting functions.

Details

plot.riskyr also uses the text settings specified in the "riskyr" object.

See Also

riskyr initializes a riskyr scenario.

Other visualization functions: plot_area, plot_bar, plot_curve, plot_fnet, plot_icons, plot_mosaic, plot_plane, plot_prism, plot_tab, plot_tree

Other riskyr scenario functions: read_popu, riskyr, summary.riskyr

Examples

# Select a scenario (from list of scenarios):
s1 <- scenarios$% # select scenario 1 from scenarios
plot(s1) # default plot (type = "prism")

# Plot types currently available:
plot(s1, type = "prism") # prism/network diagram (default)
plot(s1, type = "tree", by = "cd") # tree diagram (only 1 perspective)
plot(s1, type = "area") # area/mosaic plot
plot(s1, type = "tab") # 2x2 frequency/contingency table
plot(s1, type = "bar", dir = 2) # bar plot
plot(s1, type = "icons") # icon array
plot(s1, type = "curve", what = "all") # curves as fn. of prev
plot_area

Plot an area diagram of probabilities or frequencies.

Description

plot_area assigns the total probability or population frequency to an area (square or rectangle) and shows the probability or frequency of 4 classification cases (hi, mi, fa, cr) as relative proportions of this area.

Usage

plot_area(prev = num$prev, sens = num$sens, mirt = NA, spec = num$spec, fart = NA, N = num$N, by = "cddc", p_split = "v", area = "sq", scale = "p", round = TRUE, sum_w = 0.1, gaps = c(NA, NA), f_lbl = "num", f_lbl_sep = NA, f_lbl_sum = "num", f_lbl_hd = "abb", f_lwd = 0, p_lbl1 = NA, arr_c = -3, col_p = c(grey(0.15, 0.99), "yellow", "yellow"), brd_dis = 0.06, lbl_txt = txt, title_lbl = txt$scen_lbl, cex_lbl = 0.9, cex_p_lbl = NA, col_pal = pal, mar_notes = TRUE, ...)

Arguments

prev The condition’s prevalence prev (i.e., the probability of condition being TRUE).
sens The decision’s sensitivity sens (i.e., the conditional probability of a positive decision provided that the condition is TRUE). sens is optional when its complement mirt is provided.
mirt The decision’s miss rate mirt (i.e., the conditional probability of a negative decision provided that the condition is TRUE). mirt is optional when its complement sens is provided.
spec The decision’s specificity value spec (i.e., the conditional probability of a negative decision provided that the condition is FALSE). spec is optional when its complement fart is provided.
fart The decision’s false alarm rate fart (i.e., the conditional probability of a positive decision provided that the condition is FALSE). fart is optional when its complement spec is provided.
N The number of individuals in the population. A suitable value of N is computed, if not provided. Note: N is not represented in the plot, but used for computing frequency information freq from current probabilities prob.
by A character code specifying 2 perspectives that split the population into subsets, with 6 options:
1. "cdcc": by condition (cd) and by decision (dc) (default);
2. "cdac": by condition (cd) and by accuracy (ac);
3. "dcdd": by decision (dc) and by condition (cd);
4. "dcac": by decision (dc) and by accuracy (ac);
5. "accd": by accuracy (ac) and by condition (cd);
6. "acdc": by accuracy (ac) and by decision (dc).

`p_split` Primary perspective for population split, with 2 options:
1. "v": vertical (default);
2. "h": horizontal.

`area` A character code specifying the shape of the main area, with 2 options:
1. "sq": main area is scaled to square (default);
2. "no": no scaling (rectangular area fills plot size).

`scale` Scale probabilities and corresponding area dimensions either by exact probability or by (rounded or non-rounded) frequency, with 2 options:
1. "p": scale main area dimensions by exact probability (default);
2. "f": re-compute probabilities from (rounded or non-rounded) frequencies and scale main area dimensions by their frequency.

Note: scale setting matters for the display of probability values and for area plots with small population sizes \( n \) when \( \text{round} = \text{TRUE} \).

`round` A Boolean option specifying whether computed frequencies are rounded to integers. Default: \( \text{round} = \text{TRUE} \).

`sum_w` Border width of 2 perspective summaries (on top and left borders) of main area as a proportion of area size (i.e., \( 0 <= \text{sum}_w <= 1 \)). Default: \( \text{sum}_w = .10 \). Setting \( \text{sum}_w = 0, \text{NA}, \) or NULL removes summaries; setting \( \text{sum}_w = 1 \) scales summaries to same size as main areas.

`gaps` Size of gaps (as binary numeric vector) specifying the width of vertical and horizontal gaps as proportions of area size. Defaults: \( \text{gaps} = \text{c}(\text{.02}, \text{.00}) \) for \( \text{p_split} = \text{"v"} \) and \( \text{gaps} = \text{c}(\text{.00}, \text{.02}) \) for \( \text{p_split} = \text{"h"} \).

`fLbl` Type of label for showing frequency values in 4 main areas, with 6 options:
1. "def": abbreviated names and frequency values;
2. "abb": abbreviated frequency names only (as specified in code);
3. "nam": names only (as specified in `lbl_txt = txt`);
4. "num": numeric frequency values only (default);
5. "namnum": names (as specified in `lbl_txt = txt`) and numeric values;
6. "no": no frequency labels (same for `fLbl = \text{NA} or \text{NULL}`).

`fLbl_sep` Label separator for main frequencies (used for `fLbl = "def" OR "namnum"`). Use `fLbl_sep = "\" & "\n"` to add a line break between name and numeric value. Default: `fLbl_sep = \text{NA} (set to " or "\n" based on `fLbl`).

`fLbl_sum` Type of label for showing frequency values in summary cells, with same 6 options as `fLbl` (above). Default: `fLbl_sum = "num"`: numeric values only.

`fLbl_hd` Type of label for showing frequency values in header, with same 6 options as `fLbl` (above). Default: `fLbl_hd = "abb"`: abbreviated names only.
plot_area

f_lwd  
Line width of areas. Default: f_lwd = 0.

p_lbl  
Type of label for showing 3 key probability links and values, with 7 options:
1. "def": show links and abbreviated names and probability values;
2. "abb": show links and abbreviated probability names;
3. "nam": show links and probability names (as specified in code);
4. "num": show links and numeric probability values;
5. "namnum": show links with names and numeric probability values;
6. "no": show links with no labels;
7. NA: no link (same for p_lbl = NULL, default).

arr_c  
Arrow code for symbols at ends of probability links (as a numeric value -3 <= arr_c <= +6), with the following options:
• -1 to -3: points at one/other/both end/s;
• 0: no symbols;
• +1 to +3: V-arrow at one/other/both end/s;
• +4 to +6: T-arrow at one/other/both end/s.
Default: arr_c = -3 (points at both ends).

col_p  
Colors of probability links (as vector of 3 colors). Default: col_p = c(grey(.15, .99), "yellow", "yellow", "yellow")
(Also consider: "black", "cornsilk", "whitesmoke").

brd_dis  
Distance of probability links from area border (as proportion of area width).
Default: brd_dis = .06. Note: Adjust to avoid overlapping labels. Negative values show links outside of main area.

lbl_txt  
Default label set for text elements. Default: lbl_txt = txt.

title_lbl  

cex_lbl  
Scaling factor for text labels (frequencies and headers). Default: cex_lbl = .90.

cex_p_lbl  
Scaling factor for text labels (probabilities). Default: cex_p_lbl = cex_lbl - .05.

col_pal  

mar_notes  
Boolean option for showing margin notes. Default: mar_notes = TRUE.

...  
Other (graphical) parameters.

Details

plot_area computes probabilities prob and frequencies freq from a sufficient and valid set of 3 essential probabilities (prev, and sens or its complement mirt, and spec or its complement fart) or existing frequency information freq and a population size of N individuals.

plot_area generalizes and replaces plot_mosaic by removing the dependency on the R packages vcd and grid and providing many additional options.

Value

Nothing (NULL).
See Also

plot_mosaic for older (obsolete) version; plot_tab for plotting table (without scaling area dimensions); pal contains current color settings; txt contains current text settings.

Other visualization functions: plot.risky, plot_bar, plot_curve, plot_fnet, plot_icons, plot_mosaic, plot_plane, plot_prism, plot_tab, plot_tree

Examples

## Basics:
plot_area() # default area plot,
# same as:
# plot_area(by = "cdcd", p_split = "v", area = "sq", scale = "p")

# Local freq and prob values:
plot_area(prev = .5, sens = 4/5, spec = 3/5, N = 10)

# Customizing text and color:
plot_area(prev = .2, sens = 4/5, spec = 3/5, N = 10,
by = "cdcd", p_split = "v", scale = "p",
title_lbl = "Custom text and color:",
lbl_txt = txt_org, f_lbl = "namnum",
f_lwd = 2, col_pal = pal_rgb)
plot_area(prev = .4, sens = 6/7, spec = 4/7, N = 5,
by = "cdac", p_split = "h", scale = "f",
title_lbl = "Custom text and color:",
lbl_txt = txt_org, f_lbl = "namnum", f_lbl_sep = ":\n",
f_lwd = 1, col_pal = pal_kn)

## Versions:
## by x p_split (= [3 x 2 x 2] = 12 versions):
plot_area(by = "cdcd", p_split = "v") # v01 (see v07)
plot_area(by = "cdac", p_split = "v") # v02 (see v11)
plot_area(by = "cdcd", p_split = "h") # v03 (see v05)
plot_area(by = "cdac", p_split = "h") # v04 (see v09)

# plot_area(by = "dcdc", p_split = "v") # v05 (is v03 rotated)
plot_area(by = "dcac", p_split = "v") # v06 (see v12)
# plot_area(by = "dcdc", p_split = "h") # v07 (is v01 rotated)
plot_area(by = "dcac", p_split = "h") # v08 (see v10)

plot_area(by = "accd", p_split = "v") # v09 (is v04 rotated)
# plot_area(by = "acdc", p_split = "v") # v10 (is v08 rotated)
# plot_area(by = "acdc", p_split = "h") # v11 (is v02 rotated)
# plot_area(by = "accd", p_split = "h") # v12 (is v06 rotated)

## Options:
## area:
plot_area(area = "sq") # main area as square (by scaling x-values)
plot_area(area = "no") # rectangular main area (using full plotting region)

# scale (matters for small N):
plot_area(N = 5, prev = .5, sens = .8, spec = .6, 
by = "cdcc", p_split = "v", scale = "p", p_lbl = "def") # scaled by prob (default)
plot_area(N = 5, prev = .5, sens = .8, spec = .6, 
by = "cdcc", p_split = "v", scale = "f", p_lbl = "def") # scaled by freq (for small N)
plot_area(N = 4, prev = .4, sens = .8, spec = .6, 
by = "cdac", p_split = "h", scale = "p", p_lbl = "def") # scaled by prob (default)
plot_area(N = 4, prev = .4, sens = .8, spec = .6, 
by = "cdac", p_split = "h", scale = "f", p_lbl = "def") # scaled by freq (for small N)

# gaps (sensible range: 0--.10):
plot_area(gaps = NA) # use default gaps (based on p_split)
plot_area(gaps = c(0, 0)) # no gaps
plot_area(gaps = c(.05, .01)) # v_gap > h_gap

# freq labels:
plot_area(f_lbl = "def", f_lbl_sep = " = ") # default
plot_area(f_lbl = NA) # NA/NULL: no freq labels (in main area & top/left boxes)
plot_area(f_lbl = "abb") # abbreviated name (i.e., variable name)
plot_area(f_lbl = "num") # only freq number
plot_area(f_lbl = "num") # only freq number
plot_area(f_lbl = "namnum", f_lbl_sep = " :\n", cex_lbl = .75) # explicit & smaller

# prob labels:
plot_area(p_lbl = NA) # no prob labels, no links
plot_area(p_lbl = "no") # show links, but no labels
plot_area(p_lbl = "namnum", cex_lbl = .70) # explicit & smaller labels

# prob arrows:
plot_area(arr_c = +3, f_lbl = NA) # V-shape arrows
plot_area(arr_c = +6, f_lbl = NA) # T-shape arrows
plot_area(arr_c = +6, f_lbl = NA, 
  brd_dis = -.02, col_p = c("black")) # adjust arrow type/position

# f_lwd:
plot_area(f_lwd = 3) # thicker lines
plot_area(f_lwd = .5) # thinner lines
plot_area(f_lwd = 0) # no lines (if f_lwd = 0/NULL/NA: lty = 0)

# sum_w:
plot_area(sum_w = .10) # default (showing top and left freq panels & labels)
plot_area(sum_w = 0) # remove top and left freq panels
plot_area(sum_w = 1, # top and left freq panels scaled to size of main areas 
col_pal = pal_org) # custom colors

## Plain and suggested plot versions:
plot_area(sum_w = 0, f_lbl = "abb", p_lbl = NA) # no compound indicators (on top/left)
plot_area(sum_w = 0, f_lbl = "num", p_lbl = "num") # no gaps, numeric labels
plot_area(f_lbl = "nam", p_lbl = NA, col_pal = pal_mod) # plot with freq labels
plot_area(f_lbl = "num", p_lbl = NA, col_pal = pal_rgb) # no borders around boxes
**plot_bar**

Plot bar charts of population frequencies.

**Description**

plot_bar draws bar charts that represent the proportions of frequencies in the current population `popu` as relatives sizes of rectangular areas.

**Usage**

```r
plot_bar(prev = num$prev, sens = num$sens, mirt = NA,
         spec = num$spec, fart = NA, N = num$N, by = "all", dir = 1,
         scale = "f", round = TRUE, f_lbl = "num", f_lwd = 1, lty = 0,
         lbl_txt = txt, title_lbl = txt$scen_lbl, col_pal = pal,
         mar_notes = TRUE, ...)"
```

**Arguments**

- `prev` The condition’s prevalence `prev` (i.e., the probability of condition being TRUE).
- `sens` The decision’s sensitivity `sens` (i.e., the conditional probability of a positive decision provided that the condition is TRUE). `sens` is optional when its complement `mirt` is provided.
- `mirt` The decision’s miss rate `mirt` (i.e., the conditional probability of a negative decision provided that the condition is TRUE). `mirt` is optional when its complement `sens` is provided.
- `spec` The decision’s specificity value `spec` (i.e., the conditional probability of a negative decision provided that the condition is FALSE). `spec` is optional when its complement `fart` is provided.
- `fart` The decision’s false alarm rate `fart` (i.e., the conditional probability of a positive decision provided that the condition is FALSE). `fart` is optional when its complement `spec` is provided.
- `N` The number of individuals in the population. (This value is not represented in the plot, but used when new frequency information `freq` and a new population table `popu` are computed from scratch from current probabilities.)
- `by` A character code specifying the perspective (or the dimension by which the population is split into 2 subsets) with the following options:
  1. `by = "cd"`: by condition;
  2. `by = "dc"`: by decision;
  3. `by = "ac"`: by accuracy;
  4. `by = "all"` combines perspectives (5 bars, default).
- `dir` Number of directions in which bars are plotted. Options:
  1. `dir = 1`: uni-directional bars (all up, default);
  2. `dir = 2`: bi-directional bars (up vs. down).
scale

Scale the heights of bars either by current frequencies (scale = "f") or by
exact probabilities (scale = "p"). Default: scale = "f". For large population
sizes N and when round = FALSE, both settings yield the same bar heights.

round

Boolean option specifying whether computed frequencies are to be rounded to
integers. Default: round = TRUE.

f_lbl

Type of frequency labels, as character code with the following options:
1. f_lbl = "nam": names;
2. f_lbl = "num": numeric values (default);
3. f_lbl = "abb": abbreviated names;
4. f_lbl = NA/NULL/"no": no labels;
5. f_lbl = "any": abbreviated names and numeric values (abb = num).

f_lwd

Line width of frequency box (border). Values of NA/NULL/0 set lwd to invisible
tiny_lwd <- .001 and lty <- 0 ("blank"). Default: f_lwd = 1.

lty

Line type of frequency box (border). Values of NA/NULL/0 set lty to lty <- 0.
Default: lty = 0 (i.e., no line).

lbl_txt

Current text information (for labels, titles, etc.). Default: lbl_txt = txt
(see init_txt).

title_lbl


col_pal

Current color palette. Default: col_pal = pal (see init_pal).

mar_notes

Boolean option for showing margin notes. Default: mar_notes = TRUE.

...

Other (graphical) parameters (e.g., cex, font, lty, etc.).

Details

If a sufficient and valid set of 3 essential probabilities (prev, and sens or its complement mirt, and
spec or its complement fart) is provided, new frequency information freq and a new population
table popu are computed from scratch. Otherwise, the existing population popu is shown.

By default, plot_bar uses current frequencies (i.e., rounded or not rounded, depending on the value
of round) as bar heights, rather than using exact probabilities to scale bar heights (i.e., default scal-
ing is scale = "f"). Using the option scale = "p" scales bar heights by probabilities (e.g., showing
bars for non-natural frequencies even when frequencies are rounded). When round = FALSE, bar heights for scale = "f" and for scale = "p" are identical.

The distinction between scale = "f" and scale = "p" matters mostly for small populations sizes
N (e.g., when N < 100). For rounded and small frequency values (e.g., freq < 10) switching from
scale = "f" to scale = "p" yields different plots.

plot_bar contrasts compound frequencies along 1 dimension (height). See plot_mosaic for 2-
dimensional visualizations (as areas) and various box) options in plot_tree and plot_fnet for
related functions.

See Also

comp_popu computes the current population; popu contains the current population; comp_freq
computes current frequency information; freq contains current frequency information; num for
basic numeric parameters; txt for current text settings; pal for current color settings

Other visualization functions: plot.risky, plot_area, plot_curve, plot_fnet, plot_icons,
plot_mosaic, plot_plane, plot_prism, plot_tab, plot_tree
Examples

# Basics:
plot_bar(prev = .33, sens = .75, spec = .66, title_lbl = "Test 1")

plot_bar(N = 1000, prev = .33, sens = .75, spec = .60,
         title_lbl = "Test 2")  # by "all" (default)

# Perspectives (by):
plot_bar(N = 1000, prev = .33, sens = .75, spec = .60, by = "cd",
         title_lbl = "Test 3a")  # by condition
plot_bar(N = 1000, prev = .33, sens = .75, spec = .60, by = "cd", dir = 2,
         title_lbl = "Test 3b", f_lbl = "num")  # bi-directional
plot_bar(N = 1000, prev = .33, sens = .75, spec = .60, by = "dc",
         title_lbl = "Test 4a")  # by decision
plot_bar(N = 1000, prev = .33, sens = .75, spec = .60, by = "dc", dir = 2,
         title_lbl = "Test 4b", f_lbl = "num")  # bi-directional
plot_bar(N = 1000, prev = .33, sens = .75, spec = .60, by = "ac",
         title_lbl = "Test 5a")  # by accuracy
plot_bar(N = 1000, prev = .33, sens = .75, spec = .60, by = "ac", dir = 2,
         title_lbl = "Test 5b", f_lbl = "num")  # bi-directional

# Customize colors and text:
plot_bar(dir = 1, f_lbl = "num", col_pal = pal_org)
plot_bar(dir = 2, f_lbl = "nam", col_pal = pal_mod)

# Frequency labels (f_lbl):
plot_bar(f_lbl = "def")  # default labels: name = num
plot_bar(f_lbl = "nam")  # name only
plot_bar(f_lbl = "num")  # numeric value only
plot_bar(f_lbl = "abb")  # abbreviated name
plot_bar(f_lbl = NA)  # no labels (NA/NULL/"no")

# Scaling and rounding effects:
plot_bar(N = 3, prev = .1, sens = .7, spec = .6, dir = 2,
         scale = "f", round = TRUE,
         title_lbl = "Rounding (1)")  # => Scale by freq and round freq.
plot_bar(N = 3, prev = .1, sens = .7, spec = .6, dir = 2,
         scale = "p", round = TRUE,
         title_lbl = "Rounding (2)")  # => Scale by prob and round freq.
plot_bar(N = 3, prev = .1, sens = .7, spec = .6, dir = 2,
         scale = "f", round = FALSE,
         title_lbl = "Rounding (3)")  # => Scale by freq and do NOT round freq.
plot_bar(N = 3, prev = .1, sens = .7, spec = .6, dir = 2,
         scale = "p", round = FALSE,
         title_lbl = "Rounding (4)")  # => Scale by prob and do NOT round freq.
plot_curve  

Plot curves of selected values (e.g., PPV or NPV) as a function of prevalence.

Description

plot_curve draws curves of selected values (including PPV, NPV) as a function of the prevalence (prev) for given values of sensitivity sens (or miss rate mirt) and specificity spec (or false alarm rate fart).

Usage

plot_curve(prev = num$prev, sens = num$sens, mirt = NA,
spec = num$spec, fart = NA, what = c("prev", "PPV", "NPV"),
what_col = pal, uc = 0, show_points = TRUE, log_scale = FALSE,
lbl_txt = txt, title_lbl = NA, p_lbl = "def", cex_lbl = 0.85,
col_pal = pal, mar_notes = TRUE, ...)

Arguments

prev  The condition’s prevalence prev (i.e., the probability of condition being TRUE).
sens  The decision’s sensitivity sens (i.e., the conditional probability of a positive decision provided that the condition is TRUE). sens is optional when its complement mirt is provided.
mirt  The decision’s miss rate mirt (i.e., the conditional probability of a negative decision provided that the condition is TRUE). mirt is optional when its complement sens is provided.
spec  The decision’s specificity spec (i.e., the conditional probability of a negative decision provided that the condition is FALSE). spec is optional when its complement fart is provided.
fart  The decision’s false alarm rate fart (i.e., the conditional probability of a positive decision provided that the condition is FALSE). fart is optional when its complement spec is provided.
what  Vector of character codes that specify the selection of curves to be plotted. Currently available options are c("prev", "PPV", "NPV", "ppod", "acc") (shortcut: what = "all"). Default: what = c("prev", "PPV", "NPV").
uc  Uncertainty range, given as a percentage of the current prev, sens, and spec values (added in both directions). Default: uc = .00 (i.e., no uncertainty). Plausible ranges are 0 < uc < .25.
show_points  Boolean value for showing the point of intersection with the current prevalence prev in all selected curves. Default: show_points = TRUE.
log_scale  Boolean value for switching from a linear to a logarithmic x-axis. Default: log_scale = FALSE.
plot_curve

lbl_txt Labels and text elements. Default: `lbl_txt = txt`.

titleLbl Main plot title. Default: `titleLbl = NA` (using `lbl_txt$scenLbl`).

p_lbl Type of label for shown probability values, with the following options:
1. "abb": show abbreviated probability names;
2. "def": show abbreviated probability names and values (default);
3. "nam": show only probability names (as specified in code);
4. "num": show only numeric probability values;
5. "namnum": show names and numeric probability values;
6. "no": hide labels (same for `p_lbl = NA` or `NULL`).

cex_lbl Scaling factor for the size of text labels (e.g., on axes, legend, margin text). Default: `cex_lbl = .85`.

col_pal Color palette (if `what_col` is unspecified). Default: `col_pal = pal`.

mar_notes Boolean value for showing margin notes. Default: `mar_notes = TRUE`.

... Other (graphical) parameters.

Details

plot_curve is a generalization of plot_PV (see legacy code) that allows for additional dependent values.

See Also

cmp_prob computes current probability information; `prob` contains current probability information; `comp_freq` computes current frequency information; `freq` contains current frequency information; `num` for basic numeric parameters; `txt` for current text settings; `pal` for current color settings.

Other visualization functions: plot_riskyr, plot_area, plot_bar, plot_fnet, plot_icons, plot_mosaic, plot_plane, plot_prism, plot_tab, plot_tree

Examples

# Basics:
# (1) Plot current freq and prob values:
plot_curve() # default curve plot, 
# same as:
# plot_curve(what = c("prev", "PPV", "NPV"))

# hide points and show uncertainty:
plot_curve(show_points = FALSE, uc = .10) # default w/o points, 10% uncertainty range

# (2) Provide local parameters and select curves:
plot_curve(prev = .2, sens = .8, spec = .6, what = c("PPV", "NPV", "acc"), uc = .2)

# All curves: what = ("prev", "PPV", "NPV", "ppod", "acc")
plot_curve(prev = .3, sens = .9, spec = .8, what = "all", col_pal = pal_org) # all curves.

# Selected curves:
plot_curve(what = c("PPV", "NPV"))  # PPV and NPV
plot_curve(what = c("prev", "PPV", "NPV", "acc"))  # prev, PPV, NPV, and acc
plot_curve(what = c("prev", "PPV", "NPV", "ppod"))  # prev, PPV, NPV, and ppod

# Visualizing uncertainty (uc as percentage range):
plot_curve(prev = .3, sens = .9, spec = .8, what = c("prev", "PPV", "NPV"),
uc = .05)  # => prev, PPV and NPV with a 5% uncertainty range
plot_curve(prev = .2, sens = .8, spec = .7, what = "all",
uc = .10)  # => all with a 10% uncertainty range

# X-axis as linear vs. log scale:
plot_curve(prev = .01, sens = .9, spec = .8)  # linear scale
plot_curve(prev = .01, sens = .9, spec = .8, log_scale = TRUE)  # log scale

plot_curve(prev = .0001, sens = .7, spec = .6)  # linear scale
plot_curve(prev = .0001, sens = .7, spec = .6, log_scale = TRUE)  # log scale

# Probability labels:
plot_curve(p_lbl = "abb", what = "all")  # abbreviated names
plot_curve(p_lbl = "nam", what = "all")  # names only
plot_curve(p_lbl = "num", what = "all")  # numeric values only
plot_curve(p_lbl = "namnum", what = "all")  # names and values

# Text and color settings:
plot_curve(title_lbl = "Testing tiny text labels", cex_lbl = .60)
plot_curve(title_lbl = "Testing specific colors", uc = .05,
what = "all", what_col = c("grey", "red3", "green3", "blue3", "gold"))
plot_curve(title_lbl = "Testing color palette", uc = .05,
what = "all", col_pal = pal_org)

plot_fnet  
Plot a network diagram of frequencies and probabilities.

Description
plot_fnet drew a network diagram of frequencies (as nodes) and probabilities (as edges).

Usage
plot_fnet(prev = num(prev), sens = num(sens), mirt = NA,
spec = num(spec), fart = NA, N = freq$N, round = TRUE,
bdd = "cddc", area = "no", p_lbl = "num", show_accu = TRUE,
w_acc = 0.5, title_lbl = txt$scen_lbl, popu_lbl = txt$popu_lbl,
cond_true_lbl = txt$cond_true_lbl,
cond_false_lbl = txt$cond_false_lbl, dec_pos_lbl = txt$dec_pos_lbl,
dec_neg_lbl = txt$dec_neg_lbl, hi_lbl = txt$hi_lbl,
mi_lbl = txt$mi_lbl, fa_lbl = txt$fa_lbl, cr_lbl = txt$cr_lbl,
col_txt = grey(0.01, alpha = 0.99), cex_lbl = 0.85,
Arguments

prev  The condition’s prevalence \( \text{prev} \).
sens  The decision’s sensitivity \( \text{sens} \).
mirt  The decision’s miss rate \( \text{mirt} \).
spec  The decision’s specificity value \( \text{spec} \).
fart  The decision’s false alarm rate \( \text{fart} \).
N    The number of individuals in the population.
round A Boolean option specifying whether computed frequencies are rounded to integers. Default: \( \text{round} = \text{TRUE} \).
by   A character code specifying the perspective (or categories by which the population is split into subsets) with 3 options:
     1. "cddc" ... 1st by condition, 2nd by decision;
     2. "dcdc" ... 1st by decision, 2nd by condition;
     3. "cdac" ... 1st by condition, 2nd by accuracy.
area A character code specifying the area of the boxes (or their relative sizes) with 3 options:
     1. "no" ... all boxes are shown with the same size;
     2. "sq" ... boxes are squares with area sizes scaled proportional to frequencies (default);
     3. "hr" ... boxes are horizontal rectangles with area sizes scaled proportional to frequencies.
p_lbl A character code specifying the type of probability information (on edges) with 4 options:
     1. "nam" ... names of probabilities;
     2. "num" ... numeric values of probabilities (rounded to 3 decimals, default);
     3. "mix" ... names of essential probabilities, values of complements;
     4. "min" ... minimal labels: names of essential probabilities.
show_accu Option for showing current accuracy metrics \( \text{accu} \) on the margin of the plot.
w_acc  Weighting parameter \( w \) used to compute weighted accuracy \( \text{w_acc} \) in \( \text{comp_accu_freq} \).
Various other options allow the customization of text labels and colors:
title_lbl  Text label for current plot title.
popu_lbl  Text label for current population \( \text{popu} \).
cond_true_lbl Text label for current cases of \( \text{cond_true} \).
cond_false_lbl Text label for current cases of \( \text{cond_false} \).
dec_pos_lbl Text label for current cases of \( \text{dec_pos} \).
dec_neg_lbl Text label for current cases of \( \text{dec_neg} \).
Details

plot_fnet is deprecated – please use plot_prism instead.

Value

Nothing (NULL).

See Also

plot_prism is the new version of this function.

Other visualization functions: plot.riskyr, plot.area, plot.bar, plot.curve, plot.icons, plot.mosaic, plot.plane, plot_prism, plot_tab, plot_tree

Examples

plot_fnet() # frequency network with default options (by = "cdcd")

# alternative perspectives:
plot_tree(by = "cdac") # frequency network by condition and accuracy
plot_fnet(by = "dcdc") # frequency network by decision and condition

# See plot_prism for details and additional options.
**plot_icons**  

*Plot an icon array of a population.*

**Description**

`plot_icons` plots a population of which individual's condition has been classified correctly or incorrectly as icons from a sufficient and valid set of 3 essential probabilities (`prev`, and `sens` or its complement `mirt`, and `spec` or its complement `fart`) or existing frequency information `freq` and a population size of `N` individuals.

**Usage**

```r
plot_icons(prev = num$prev, sens = num$sens, mirt = NA,
spec = num$spec, fart = NA, N = freq$N, arr_type = "array",
by = "all", ident_order = c("hi", "mi", "fa", "cr"),
icon_types = 22, icon_size = NULL, icon_brd_lwd = 1.5,
block_d = NULL, border_d = 0.1, block_size_row = 10,
block_size_col = 10, nblocks_row = NULL, nblocks_col = NULL,
fill_array = "left", fill_blocks = "rowwise", lbl_txt = txt,
title_lbl = txt$scen_lbl, cex_lbl = 0.9, col_pal = pal,
transparency = 0.5, mar_notes = TRUE, ...)
```

**Arguments**

- `prev`  
The condition's prevalence `prev` (i.e., the probability of condition being TRUE).
- `sens`  
The decision's sensitivity `sens` (i.e., the conditional probability of a positive decision provided that the condition is TRUE). `sens` is optional when its complement `mirt` is provided.
- `mirt`  
The decision's miss rate `mirt` (i.e., the conditional probability of a negative decision provided that the condition is TRUE). `mirt` is optional when its complement `sens` is provided.
- `spec`  
The decision's specificity value `spec` (i.e., the conditional probability of a negative decision provided that the condition is FALSE). `spec` is optional when its complement `fart` is provided.
- `fart`  
The decision's false alarm rate `fart` (i.e., the conditional probability of a positive decision provided that the condition is FALSE). `fart` is optional when its complement `spec` is provided.
- `N`  
The number of individuals in the population. A suitable value of `N` is computed, if not provided. If `N` is 100,000 or greater it is reduced to 10,000 for the array types if the frequencies allow it.
- `arr_type`  
The icons can be arranged in different ways resulting in different types of displays:

  1. `arr_type = "array"`: Icons are plotted in a classical icon array (default). Icons can be arranged in blocks using `block_d`. The order of filling the array can be customized using `fill_array` and `fill_blocks`.  

```r
plot_icons(prev = num$prev, sens = num$sens, mirt = NA,
...)
```
2. `arr_type = "shuffledarray"`: Icons are plotted in an icon array, but positions are shuffled (randomized). Icons can be arranged in blocks using `block_d`. The order of filling the array can be customized using `fill_array` and `fill_blocks`.

3. `arr_type = "mosaic"`: Icons are ordered like in a mosaic plot. The area size displays the relative proportions of their frequencies.

4. `arr_type = "fillequal"`: Icons are positioned into equally sized blocks. Thus, their density reflects the relative proportions of their frequencies.

5. `arr_type = "fillleft"`: Icons are randomly filled from the left.

6. `arr_type = "filltop"`: Icons are randomly filled from the top.

7. `arr_type = "scatter"`: Icons are randomly scattered into the plot.

by A character code specifying a perspective to split the population into subsets, with 4 options:

1. "all": by condition (cd) and by decision (dc):
   - hi, mi, fa, cr cases (default);
2. "cd": by condition (cd) only:
   - cond_true vs. cond_false cases;
3. "dc": by decision (dc) only:
   - dec_pos vs. dec_neg cases;
4. "ac": by accuracy (ac) only:
   - dec_cor vs. dec_err cases.

ident_order The order in which icon identities (i.e., hi, mi, fa, and cr) are plotted. Default: `ident_order = c("hi", "mi", "fa", "cr")`

icon_types Specifies the appearance of the icons as a vector. Accepts values from 1 to 25 (see `points`).

icon_size Manually specifies the size of the icons via `cex` (calculated dynamically by default).

icon_brd_lwd Specifies the border width of icons (if applicable).

block_d The distance between blocks (does not apply to "fillequal", "fillleft", and "scatter")

border_d The distance of icons to the border.

Additional options for controlling the arrangement of arrays (for `arr_type = "array"` and "shuffledarray"):

- `block_size_row`: specifies how many icons should be in each block row.
- `block_size_col`: specifies how many icons should be in each block column.
- `nblocks_row`: specifies how many blocks there are in each row. Is calculated by default.
- `nblocks_col`: specifies how many blocks are there in each column. Is calculated by default.
- `fill_array`: specifies how the blocks are filled into the array (Options "left" (default) and "top").
- `fill_blocks`: specifies how icons within blocks are filled (Options: `fill_blocks = "rowwise"` (default) and `fill_blocks = "colwise"`)

Generic text and color options:

- `lbl_txt`: Default label set for text elements. Default: `lbl_txt = txt`
plot_icons

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cex_lbl</td>
<td>Scaling factor for text labels. Default: cex_lbl = .90.</td>
</tr>
<tr>
<td>transparency</td>
<td>Specifies the transparency for overlapping icons (not for arr_type = &quot;array&quot; and &quot;shuffledarray&quot;).</td>
</tr>
<tr>
<td>mar_notes</td>
<td>Boolean option for showing margin notes. Default: mar_notes = TRUE.</td>
</tr>
</tbody>
</table>

Other (graphical) parameters.

Details

If probabilities are provided, a new list of natural frequencies freq is computed by comp_freq. By contrast, if no probabilities are provided, the values currently contained in freq are used. By default, comp_freq rounds frequencies to nearest integers to avoid decimal values in freq.

Value

Nothing (NULL).

See Also

Other visualization functions: plot.riskyr, plot_area, plot_bar, plot_curve, plot_fnet, plot_mosaic, plot_plane, plot_prism, plot_tab, plot_tree

Examples

# ways to work:
plot_icons(N = 1000) # icon array with default settings (arr_type = "array")
plot_icons(arr_type = "shuffledarray", N = 1000) # icon array with shuffled IDs

# array types:
plot_icons(arr_type = "mosaic", N = 1000) # areas as in mosaic plot
plot_icons(arr_type = "fillequal", N = 1000) # areas of equal size (probability as density)
plot_icons(arr_type = "fillleft", N = 1000) # icons filled from left to right (in columns)
plot_icons(arr_type = "filltop", N = 1000) # icons filled from top to bottom (in rows)
plot_icons(arr_type = "scatter", N = 1000) # icons randomly scattered

# by argument:
plot_icons(N = 1000, by = "all") # hi, mi, fa, cr (TP, FN, FP, TN) cases
plot_icons(N = 1000, by = "cd") # (hi + mi) vs. (fa + cr) (TP + FN vs. FP + TN) cases
plot_icons(N = 1000, by = "dc") # (hi + fa) vs. (mi + cr) (TP + FP vs. FN + TN) cases
plot_icons(N = 1000, by = "ac") # (hi + cr) vs. (fa + mi) (TP + TN vs. FP + FN) cases

# icon symbols:
plot_icons(N = 100, icon_types = c(21, 23, 24, 23),
block_size_row = 5, block_size_col = 5, nblocks_row = 2, nblocks_col = 2,
block_d = 0.5, border_d = 0.9)

# variants:
plot_icons(N = 800, arr_type = "array", icon_types = c(21, 22, 23, 24),
block_d = 0.5, border_d = 0.5)
plot_mosaic

Plot a mosaic plot of population frequencies.

Description

plot_mosaic drew a mosaic plot that represents the proportions of frequencies in the current population as relatives sizes of rectangular areas.

Usage

```
plot_mosaic(prev = num$prev, sens = num$sens, mirt = NA,
            spec = num$spec, fart = NA, N = num$N, by = "cddc",
            show_accu = TRUE, w_acc = 0.5, title_lbl = txt$scen_lbl,
            col_sdt = c(pal["hi"], pal["mi"], pal["fa"], pal["cr"]))
```

Arguments

- **prev**: The condition’s prevalence `prev`.
- **sens**: The decision’s sensitivity `sens`.
- **mirt**: The decision’s miss rate `mirt`.

Example:

```
plot_icons(N = 1250, sens = 0.9, spec = 0.9, prev = 0.9,
           icon_types = c(21, 23, 24, 23),
           block_size_row = 10, block_size_col = 5,
           nblocks_row = 5, nblocks_col = 5,
           block_d = 0.8,
           border_d = 0.2,
           fill_array = "top")
```

```
plot_icons(N = 800, arr_type = "shuffledarray", icon_types = c(21, 23, 24, 22),
           block_d = 0.5, border_d = 0.5)
```

```
plot_icons(N = 800, arr_type = "shuffledarray", icon_types = c(21, 23, 24, 22),
           icon_brd_col = grey(.33, .99), icon_brd_lwd = 3, cex_lbl = 1.2)
```

```
plot_icons(N = 800, arr_type = "filledequal", icon_types = c(21, 22, 22, 21),
           icon_brd_lwd = .5, cex = 1, cex_lbl = 1.1)
```

# Text and color options:
```
plot Icons(N = 1000, prev = .5, sens = .5, spec = .5, arr_type = "shuffledarray",
           title_lbl = "", lbl_txt = txt_TF, col_pal = pal_vir, mar_notes = FALSE)
```

```
plot_icons(N = 1000, prev = .5, sens = .5, spec = .5, arr_type = "shuffledarray",
           title_lbl = "Green vs. red", col_pal = pal_rgb, transparency = .5)
```

```
plot_icons(N = 1000, prev = .5, sens = .5, spec = .5, arr_type = "shuffledarray",
           title_lbl = "Shades of blue", col_pal = pal_kn, transparency = .3)
```
spec  The decision’s specificity value spec.
fart  The decision’s false alarm rate fart.
N     The number of individuals in the population.
by    A character code specifying the perspective (or categories by which the population is split into subsets) with 3 options:
       1. "cdcd" ... by condition x decision;
       2. "dcdcd" ... by decision x condition;
       3. "cdac" ... by condition x accuracy.
show_accu Option for showing current and exact accuracy metrics accu in the plot.
w_acc  Weighting parameter w used to compute weighted accuracy.
title_lbl Text label for current plot title.
col_sdt  Colors for cases of 4 essential frequencies. Default: col_sdt = c(pal["hi"], pal["mi"], pal["fa"],

Details
plot_mosaic is deprecated – please use plot_area instead.

See Also
plot_area is the new version of this function.
Other visualization functions: plot.risky, plot_area, plot_bar, plot_curve, plot_fnet,
plot_icons, plot_plane, plot_prism, plot_tab, plot_tree

Examples
plot_mosaic()  # plot with default options

plot_plane(prev = num$prev, sens = num$sens, mirt = NA,
           spec = num$spec, fart = NA, what = "PPV", what_col = pal,
           line_col = "grey85", point_col = "yellow", show_point = TRUE,
           step_size = 0.05, theta = -45, phi = 0, lbl_txt = txt,
           title_lbl = NA, p_lbl = "def", cex_lbl = 0.85, col_pal = pal,
           mar_notes = TRUE, ...)

plot_plane
Plot a plane of selected values (e.g., PPV or NPV) as a function of
sensitivity and specificity.

Description
plot_plane draws a 3D-plane of selected values (e.g., predictive values PPV or NPV) as a function of
a decision’s sensitivity sens and specificity value spec for a given prevalence (prev).

Usage
plot_plane(prev = num$prev, sens = num$sens, mirt = NA,
           spec = num$spec, fart = NA, what = "PPV", what_col = pal,
           line_col = "grey85", point_col = "yellow", show_point = TRUE,
           step_size = 0.05, theta = -45, phi = 0, lbl_txt = txt,
           title_lbl = NA, p_lbl = "def", cex_lbl = 0.85, col_pal = pal,
           mar_notes = TRUE, ...)
Arguments

prev The condition’s prevalence prev (i.e., the probability of condition being TRUE).
sens The decision’s sensitivity sens (i.e., the conditional probability of a positive decision provided that the condition is TRUE). sens is optional when its complement mirt is provided.
mirt The decision’s miss rate mirt (i.e., the conditional probability of a negative decision provided that the condition is TRUE). mirt is optional when its complement sens is provided.
spec The decision’s specificity value spec (i.e., the conditional probability of a negative decision provided that the condition is FALSE). spec is optional when its complement fart is provided.
fart The decision’s false alarm rate fart (i.e., the conditional probability of a positive decision provided that the condition is FALSE). fart is optional when its complement spec is provided.
what A character code that specifies one metric to be plotted as a plane. Currently available options are c("PPV", "NPV", "ppod", "acc"). Default: what = "PPV".
line_col Color for lines between surface facets. Default: line_col = "grey85".
point_col Fill color for showing current value on plane. Default: point_col = "yellow".
show_point Boolean option for showing the current value of the selected metric for the current conditions (prev, sens, spec) as a point on the plane. Default: show_point = TRUE.
step_size Sets the granularity of the sens-by-spec grid. (in range 0 <= step_size <= 1). Default: step_size = .05.
theta Horizontal rotation angle (used by persp). Default: theta = -45.
phi Vertical rotation angle (used by persp). Default: phi = 0.
lbl_txt Labels and text elements. Default: lbl_txt = txt.
title_lbl Main plot title. Default: title_lbl = NA (using lbl_txt$scen_lbl).
p_lbl Type of label for shown probability values, with the following options:
1. "abb": show abbreviated probability names;
2. "def": show abbreviated probability names and values (default);
3. "nam": show only probability names (as specified in code);
4. "num": show only numeric probability values;
5. "namnum": show names and numeric probability values;
6. "no": hide labels (same for p_lbl = NA or NULL).
cex_lbl Scaling factor for the size of text labels (e.g., on axes, legend, margin text). Default: cex_lbl = .85.
col_pal Color palette (if what_col is unspecified). Default: col_pal = pal.
mar_notes Boolean value for showing margin notes. Default: mar_notes = TRUE.
... Other (graphical) parameters.
Details

plot_prism is a generalization of plot_PV3d (see legacy code) that allows for additional dependent values.

See Also

comp_popu computes the current population; popu contains the current population; comp_freq computes current frequency information; freq contains current frequency information; num for basic numeric parameters; txt for current text settings; pal for current color settings

Other visualization functions: plot_risky, plot_area, plot_bar, plot_curve, plot_fnet, plot Icons, plot_mosaic, plot_prism, plot_tab, plot_tree

Examples

# Basics:
plot_plane() # => default plot (what = "PPV")
# same as:
# plot_plane(what = "PPV") # => plane of PPV
plot_plane(what = "NPV") # => plane of NPV
plot_plane(what = "ppod") # => plane of ppod
plot_plane(what = "acc") # => plane of acc

# Plot options:
plot_plane(titleLbl = "Testing smaller text labels", cex_lbl = .60) # => no point shown on plane
plot_plane(titlelbl = "Testing plot colors", what_col = "royalblue4", line_col = "sienna2")
plot_plane(titlelbl = "Testing plot in b/w", what_col = "white", line_col = "black")
plot_plane(step_size = .333, what_col = "firebrick") # => coarser granularity + color
plot_plane(step_size = .025, what_col = "chartreuse4") # => finer granularity + color
plot_plane(what_col = "steelblue4", theta = -90, phi = 45) # => rotated, from above

plot_prism

Plot prism diagram of frequencies and probabilities.

Description

plot_prism plots a network diagram of from a sufficient and valid set of 3 essential probabilities (prev, and sens or its complement mirt, and spec or its complement fart) or existing frequency information freq and a population size of N individuals.
Usage

```
plot_prism(prev = num$prev, sens = num$sens, mirt = NA,
            spec = num$spec, fart = NA, N = num$N, by = "cddc",
            area = "no", scale = "p", round = TRUE, f_lbl = "num",
            f_lbl_sep = NA, f_lwd = 0, p_lbd = "mix", arr_c = NA,
            lbl_txt = txt, title_lbl = txt$scen_lbl, cex_lbl = 0.9,
            cex_p_lbd = NA, col_pal = pal, mar_notes = TRUE, ...)
```

Arguments

**prev**

The condition’s prevalence `prev` (i.e., the probability of condition being TRUE).

**sens**

The decision’s sensitivity `sens` (i.e., the conditional probability of a positive decision provided that the condition is TRUE). `sens` is optional when its complement `mirt` is provided.

**mirt**

The decision’s miss rate `mirt` (i.e., the conditional probability of a negative decision provided that the condition is TRUE). `mirt` is optional when its complement `sens` is provided.

**spec**

The decision’s specificity value `spec` (i.e., the conditional probability of a negative decision provided that the condition is FALSE). `spec` is optional when its complement `fart` is provided.

**fart**

The decision’s false alarm rate `fart` (i.e., the conditional probability of a positive decision provided that the condition is FALSE). `fart` is optional when its complement `spec` is provided.

**N**

The number of individuals in the population. A suitable value of `N` is computed, if not provided. Note: `N` is not represented in the plot, but used for computing frequency information `freq` from current probabilities `prob`.

**by**

A character code specifying 1 or 2 perspectives that split the population into 2 subsets. Specifying 1 perspective plots a frequency tree (single tree) with 3 options:

- "cd": by condition only;
- "dc": by decision only;
- "ac": by accuracy only.

Specifying 2 perspectives plots a frequency prism (network, double tree) with 6 options:

- "cdcd": by condition (cd) and by decision (dc) (default);
- "cdac": by condition (cd) and by accuracy (ac);
- "dcdc": by decision (dc) and by condition (cd);
- "dcac": by decision (dc) and by accuracy (ac);
- "accd": by accuracy (ac) and by condition (cd);
- "acdc": by accuracy (ac) and by decision (dc).

**area**

A character code specifying the shapes of the frequency boxes, with 3 options:

- "no": rectangular frequency boxes, not scaled (default);
- "hr": frequency boxes are horizontal rectangles (scaled relative to N).
3. "sq": frequency boxes are squares (scaled relative to N).

**scale**

Scale probabilities and corresponding area dimensions either by exact probability or by (rounded or non-rounded) frequency, with 2 options:

1. "p": scale main area dimensions by exact probability (default);
2. "f": re-compute probabilities from (rounded or non-rounded) frequencies and scale main area dimensions by their frequency.

Note: scale setting matters for the display of probability values and for area plots with small population sizes \(N\) when \(round = TRUE\).

**round**

A Boolean option specifying whether computed frequencies are rounded to integers. Default: \(round = TRUE\).

**f_lbl**

Type of label for showing frequency values in 4 main areas, with 6 options:

1. "def": abbreviated names and frequency values;
2. "abb": abbreviated frequency names only (as specified in code);
3. "nam": names only (as specified in lbl_txt = txt);
4. "num": numeric frequency values only (default);
5. "namnum": names (as specified in lbl_txt = txt) and numeric values;
6. "no": no frequency labels (same for f_lbl = NA or NULL).

**f_lbl_sep**

Label separator for main frequencies (used for f_lbl = "def" OR "namnum"). Use f_lbl_sep = ":\n" to add a line break between name and numeric value. Default: f_lbl_sep = NA (set to ":" or ":\n" based on f_lbl).

**f_lwd**

Line width of areas. Default: f_lwd = 0.

**p_lbl**

Type of label for showing 3 key probability links and values, with many options:

1. "abb": show links and abbreviated probability names;
2. "def": show links and abbreviated probability names and values;
3. "min": show links and minimum (prominent) probability names;
4. "mix": show links and prominent probability names and all values (default);
5. "nam": show links and probability names (as specified in code);
6. "num": show links and numeric probability values;
7. "namnum": show links with names and numeric probability values;
8. "no": show links with no labels (same for p_lbl = NA or NULL).

**arr_c**

Arrow code for symbols at ends of probability links (as a numeric value \(-3 <= arr_c <= +6\)), with the following options:

- \(-1\) to \(-3\): points at one/other/both end/s;
- \(0\): no symbols;
- \(+1\) to \(+3\): V-arrow at one/other/both end/s;
- \(+4\) to \(+6\): T-arrow at one/other/both end/s.

Default: \(arr_c = NA\), but adjusted by area.

**lbl_txt**

Default label set for text elements. Default: lbl_txt = txt.

**title_lbl**


**cex_lbl**

Scaling factor for text labels (frequencies and headers). Default: cex_lbl = .90.
plot_prism

cex_p_lbl  Scaling factor for text labels (probabilities). Default: cex_p_lbl = cex_lbl - .05.
mar_notes  Boolean option for showing margin notes. Default: mar_notes = TRUE.
...        Other (graphical) parameters.

Details

plot_prism generalizes and replaces plot_fnet by removing the dependency on the R package
diagram and providing many additional options.

Value

Nothing (NULL).

See Also

plot_fnet for older (obsolete) version; plot_area for plotting mosaic plot (scaling area dimen-
sions); plot_bar for plotting frequencies as vertical bars; plot_tab for plotting table (without
scaling area dimensions); pal contains current color settings; txt contains current text settings.

Other visualization functions: plot.risky, plot_area, plot_bar, plot_curve, plot_fnet,
plot_icons, plot_mosaic, plot_plane, plot_tab, plot_tree

Examples

## Basics:
# (1) Using global prob and freq values:
plot_prism()  # default prism plot,
# same as:
# plot_prism(by = "ccdc", area = "no", scale = "p",
# f_lbl = "num", f_lwd = 0, cex_lbl = .90,
# p_lbl = "mix", arr_c = -2, cex_p_lbl = NA)

# (2) Providing values:
plot_prism(N = 10, prev = 1/2, sens = 4/5, spec = 3/5)
plot_prism(N = 10, prev = 1/3, sens = 3/5, spec = 4/5, area = "hr")
plot_prism(N = 10, prev = 1/4, sens = 3/5, spec = 2/5, area = "sq", mar_notes = TRUE)

## Custom color and text settings:
plot_prism(col_pal = pal_bw, f_lwd = .5, lwd = .5, lty = 2, # custom fbox color, prob links,
          font = 3, cex_p_lbl = .75)  # and text labels

my_txt <- init_txt(cond_lbl = "The Truth", cond_true_lbl = "so true", cond_false_lbl = "so false",
                   hi_lbl = "TP", mi_lbl = "FN", fa_lbl = "FP", cr_lbl = "TN")
my_col <- init_pal(N_col = rgb(0, 169, 224, max = 255), # seeblau
                   hi_col = "gold", mi_col = "firebrick1", fa_col = "firebrick2", cr_col = "orange")
plot_prism(f_lbl = "nam", lbl_txt = my_txt,
           col_pal = my_col, f_lwd = .5)

## Local values and custom color/txt settings:
plot_prism(N = 7, prev = 1/2, sens = 3/5, spec = 4/5, round = FALSE,
by = "cdac", lbl-txt = txt-org, f-lbl = "namnum", f-lbl-sep = "\n",
  f-lwd = 1, col-pal = pal-rgb) # custom colors

plot_prism(N = 5, prev = 1/2, sens = .8, spec = .5, scale = "p", # note scale!
  by = "cdac", area = "hr", col-pal = pal-bw, f-lwd = 1) # custom colors

plot_prism(N = 3, prev = .50, sens = .50, spec = .50, scale = "p"),
  # note scale!
  area = "sq", lbl-txt = txt-org, f-lbl = "namnum", f-lbl-sep = "\n",
  col-pal = pal-kn, f-lwd = .5) # custom text

## Plot versions:

# (A) tree/single tree (nchar(by) == 2):
# 3 versions:
plot_prism(by = "cd", f-lbl = "def", col-pal = pal-mod) # by condition (freq boxes: hi mi fa cr)
plot_prism(by = "dc", f-lbl = "def", col-pal = pal-mod) # by decision (freq boxes: hi fa mi cr)
plot_prism(by = "ac", f-lbl = "def", col-pal = pal-mod) # by decision (freq boxes: hi cr mi fa)

# (B) prism/double tree (nchar(by) == 4):
# 6 (3 x 2) versions (+ 3 redundant ones):
plot_prism(by = "cdcd") # v01 (default)
plot_prism(by = "cddc") # v02
plot_prism(by = "cdcd") # (+) Message
plot_prism(by = "dcdc") # v03
plot_prism(by = "dcac") # v04
plot_prism(by = "dcdc") # (+) Message
plot_prism(by = "accd") # v05
plot_prism(by = "acdc") # v06
plot_prism(by = "accd") # (+) Message

## Other options:

# area:
plot_prism(area = "no") # rectangular boxes (default): (same if area = NA/NULL)
plot_prism(area = "hr") # horizontal rectangles (widths on each level sum to N)
plot_prism(area = "sq") # squares (areas on each level sum to N)

# scale (matters for scaled areas and small N):
plot_prism(N = 5, prev = .3, sens = .8, spec = .6,
  area = "hr", scale = "p") # widths scaled by prob
plot_prism(N = 5, prev = .3, sens = .8, spec = .6,
  area = "hr", scale = "f") # widths scaled by (rounded or non-rounded) freq
plot_prism(N = 4, prev = .2, sens = .7, spec = .8,
  area = "sq", scale = "p") # areas scaled by prob
plot_prism(N = 4, prev = .2, sens = .7, spec = .8,
  area = "sq", scale = "f") # areas scaled by (rounded or non-rounded) freq

## Frequency boxes:
# f-lbl:
plot_prism(f-lbl = "abb") # abbreviated freq names (variable names)
plot_prism(f-lbl = "nam") # only freq names
plot_prism(f-lbl = "num") # only numeric freq values (default)
plot_prism(f_lbl = "namnum")  # names and numeric freq values
plot_prism(f_lbl = "namnum", cex_lbl = .75)  # smaller freq labels
plot_prism(f_lbl = NA)  # no freq labels
plot_prism(f_lbl = "def")  # informative default: short name and numeric value (abb = num)

# f_lwd:
plot_prism(f_lwd = 0)  # no lines (default), set to tiny_lwd = .001, lty = 0 (same if NA/NULL)
plot_prism(f_lwd = 1)  # basic lines
plot_prism(f_lwd = 3)  # thicker lines
plot_prism(f_lwd = .5)  # thinner lines

## Probability links:
# p_lbl:
plot_prism(p_lbl = "mix")  # abbreviated names with numeric values (abb = num)
plot_prism(p_lbl = NA)  # no prob labels (NA/NULL/"none")
plot_prism(p_lbl = "nam")  # only prob names
plot_prism(p_lbl = "num")  # only numeric prob values
plot_prism(p_lbl = "namnum")  # names and numeric prob values
plot_prism(p_lbl = "namnum", cex_p_lbl = .70)  # smaller prob labels
plot_prism(by = "ccdc", p_lbl = "min")  # minimal labels
plot_prism(by = "ccdc", p_lbl = "min")
plot_prism(by = "ccdc", p_lbl = "mix")  # mix abbreviated names and numeric values
plot_prism(by = "ccdc", p_lbl = "mix")
plot_prism(by = "ccdc", p_lbl = "abb")  # abbreviated names
plot_prism(by = "ccdc", p_lbl = "abb")
plot_prism(p_lbl = "any")  # short name and value (abb = num)

## arr_c:
plot_prism(arr_c = 0)  # acc_c = 0: no arrows
plot_prism(arr_c = -3)  # arr_c = -1 to -3: points at both ends
plot_prism(arr_c = -2)  # point at far end
plot_prism(arr_c = 2)  # crr_c = 1-3: V-shape arrows at far end
plot_prism(arr_c = 3)  # V-shape arrows at both ends
plot_prism(arr_c = 6)  # arr_c = 4-6: T-shape arrows

## Plain plot versions:
plot_prism(area = "no", f_lbl = "def", p_lbl = "num", col_pal = pal_mod, f_lwd = 1, 
            title_lbl = "", mar_notes = FALSE)  # remove titles and margin notes
plot_prism(area = "no", f_lbl = "nam", p_lbl = "min", col_pal = pal_rgb)
plot_prism(area = "no", f_lbl = "abb", p_lbl = "abb", col_pal = pal_bw)
plot_prism(area = "no", f_lbl = "num", p_lbl = "num", col_pal = pal_kn)

plot_prism(area = "hr", f_lbl = "num", f_lwd = .5, p_lbl = NA, arr_c = 0, 
            col_pal = pal_mod, lwd = .5)
plot_prism(area = "hr", f_lbl = "nam", f_lwd = .5, p_lbl = NA, col_pal = pal_bw)
plot_prism(area = "hr", f_lbl = "nam", f_lwd = .5, p_lbl = "num")

plot_prism(area = "sq", f_lbl = "nam", p_lbl = NA, col_pal = pal_rgb)
plot_prism(area = "sq", f_lbl = "num", p_lbl = NA, f_lwd = 1, col_pal = pal_bw, lwd = .5)
plot_prism(area = "sq", f_lbl = "def", f_lbl_sep = ":\n", p_lbl = NA, f_lwd = 1, col_pal = pal_kn)

## Suggested combinations:
plot_prism(f_lbl = "nam", p_lbl = "mix", col_pal = pal_mod)
plot_prism(f_lbl = "namnum", p_lbl = "mix", cex_lbl = .80, cex_p_lbl = .75)
plot_prism(area = "hr", f_lbl = "nam", p_lbl = "num", col_pal = pal_mod)
plot_prism(area = "hr", f_lbl = "abb", p_lbl = "abb", f_lwd = 1, col_pal = pal_bw)
plot_prism(area = "hr", f_lbl = "num", p_lbl = "mix", f_lwd = 1, cex_p_lbl = .75)
plot_prism(area = "sq", f_lbl = "nam", p_lbl = "abb", lbl_txt = txt_tf)
plot_prism(area = "sq", f_lbl = "num", p_lbl = "num", f_lwd = 1, col_pal = pal_rgb)
plot_prism(area = "sq", f_lbl = "namnum", p_lbl = "mix", f_lwd = .5, col_pal = pal_kn)

plot_tab

Plot a 2 x 2 contingency table of population frequencies.

Description

plot_tab plots a 2 x 2 contingency table (aka. confusion table) of 4 classification cases (hi, mi, fa, cr) and corresponding row and column sums.

Usage

plot_tab(prev = num$prev, sens = num$sens, mirt = NA,
spec = num$spec, fart = NA, N = num$N, by = "cddc",
p_split = "v", area = "no", scale = "p", round = TRUE,
f_lbl = "num", f_lbl_sep = NA, f_lbl_sum = f_lbl,
f_lbl_hd = "abb", f_lwd = 0, gaps = c(NA, NA), brd_w = 0.1,
p_lbl = NA, arr_c = -3, col_p = c(grey(0.15, 0.99), "yellow",
"yellow"), brd_dis = 0.3, lbl_txt = txt, title_lbl = txt$scen_lbl,
cex_lbl = 0.9, cex_p_lbl = NA, col_pal = pal, mar_notes = TRUE,
...)

Arguments

prev The condition’s prevalence prev (i.e., the probability of condition being TRUE).
sens The decision’s sensitivity sens (i.e., the conditional probability of a positive decision provided that the condition is TRUE). sens is optional when its complement mirt is provided.
mirt The decision’s miss rate mirt (i.e., the conditional probability of a negative decision provided that the condition is TRUE). mirt is optional when its complement sens is provided.
spec The decision’s specificity value spec (i.e., the conditional probability of a negative decision provided that the condition is FALSE). spec is optional when its complement fart is provided.
fart The decision’s false alarm rate fart (i.e., the conditional probability of a positive decision provided that the condition is FALSE). fart is optional when its complement spec is provided.
N
The number of individuals in the population. A suitable value of N is computed, if not provided. Note: N is not represented in the plot, but used for computing frequency information freq from current probabilities prob.

by
A character code specifying 2 perspectives that split the population into subsets, with 6 options:
1. "cdcd": by condition (cd) and by decision (dc) (default);
2. "cdac": by condition (cd) and by accuracy (ac);
3. "dacc": by decision (dc) and by condition (cd);
4. "dcac": by decision (dc) and by accuracy (ac);
5. "accd": by accuracy (ac) and by condition (cd);
6. "acdc": by accuracy (ac) and by decision (dc).

p_split
Primary perspective for population split, with 2 options:
1. "v": vertical (default);
2. "h": horizontal.
Note: In contrast to plot_area, this setting only determines which 3 probability links are shown (e.g., when p_link = "def").

area
A character code specifying the shape of the main area, with 4 options:
1. "sq": main area is scaled to square;
2. "no": no scaling (rectangular area fills plot size; default).

scale
Scale probabilities (but not table cell dimensions) either by exact probability or by (rounded or non-rounded) frequency, with 2 options:
1. "p": scale main area dimensions by exact probability (default);
2. "f": re-compute probabilities from (rounded or non-rounded) frequencies and scale main area dimensions by their frequency.

Note: scale setting matters for the display of probability values and for area plots with small population sizes N when round = TRUE.

round
A Boolean option specifying whether computed frequencies are rounded to integers. Default: round = TRUE.

f_lbl
Type of label for showing frequency values in 4 main areas, with 6 options:
1. "def": abbreviated names and frequency values (default);
2. "abb": abbreviated frequency names only (as specified in code);
3. "nam": names only (as specified in lb1_txt = txt);
4. "num": numeric frequency values only;
5. "namnum": names (as specified in lb1_txt = txt) and numeric values;
6. "no": no frequency labels (same for f_lbl = NA or NULL).

f_lbl_sep
Label separator for main frequencies (used for f_lbl = "def" OR "namnum"). Use f_lbl_sep = ":\n" to add a line break between name and numeric value. Default: f_lbl_sep = NA (set to "" or ":\n" based on f_lbl).

fLbl_sum
Type of label for showing frequency values in summary cells, with same 6 options as f_lbl (above). Default: fLbl_sum = "def": abbreviated names and numeric values.
plot_tab

f_lbl_hd  Type of label for showing frequency values in header, with same 6 options as f_lbl (above). Default: f_lbl_hd = "abb": abbreviated names only.

f_lwd  Line width of areas. Default: f_lwd = 1.

gaps  Size of gaps (as binary numeric vector) specifying the widths of vertical and horizontal gaps between 2 x 2 table and sums (in bottom row and right column). Default: gaps = c(.05, .06).

brd_w  Border width for showing 2 perspective summaries on top and left borders of main area (as a proportion of area size) in a range 0 <= brd_w <= 1. Default: brd_w = .10.

p_lbl  Type of label for showing 3 key probability links and values, with 7 options:
1. "def": show links and abbreviated names and probability values;
2. "abb": show links and abbreviated probability names;
3. "nam": show links and probability names (as specified in code);
4. "num": show links and numeric probability values;
5. "namnum": show links with names and numeric probability values;
6. "no": show links with no labels;
7. NA: no link (same for p_lbl = NULL, default).

arr_c  Arrow code for symbols at ends of probability links (as a numeric value -3 <= arr_c <= +6), with the following options:
• -1 to -3: points at one/other/both end/s;
• 0: no symbols;
• +1 to +3: V-arrow at one/other/both end/s;
• +4 to +6: T-arrow at one/other/both end/s.
Default: arr_c = -3 (points at both ends).

col_p  Colors of probability links (as vector of 3 colors). Default: col_p = c(grey(.15, .99), "yellow", "yellow2").

brd_dis  Distance of probability links from cell center (as a constant). Default: brd_dis = .30.
Note: Adjust to avoid overlapping labels.

lbl_txt  Default label set for text elements. Default: lbl_txt = txt.


cex_lbl  Scaling factor for text labels (frequencies and headers). Default: cex_lbl = .90.

cex_p_lbl  Scaling factor for text labels (probabilities). Default: cex_p_lbl = cex_lbl - .05.


mar_notes  Boolean option for showing margin notes. Default: mar_notes = TRUE.

Details

plot_tab computes its frequencies freq from a sufficient and valid set of 3 essential probabilities (prev, and sens or its complement mirt, and spec or its complement fart) or existing frequency information freq and a population size of N individuals.

plot_tab is derived from plot_area, but does not scale the dimensions of table cells.
Value

Nothing (NULL).

See Also

plot_area for plotting mosaic plot (scaling area dimensions); pal contains current color settings; txt contains current text settings.

Other visualization functions: plot_riskyr, plot_area, plot_bar, plot_curve, plot_fnet, plot_icons, plot_mosaic, plot_plane, plot_prism, plot_tree

Examples

```r
## Basics:
# (1) Plotting global freq and prob values:
plot_tab()
plot_tab(area = "sq", f_lwd = 3, col_pal = pal_rgb)
plot_tab(f_lbl = "namnum", f_lbl_sep = " ", brd_w = .10, f_lwd = .5)

# (2) Computing local freq and prob values:
plot_tab(prev = .5, sens = 4/5, spec = 3/5, N = 10, f_lwd = 1)

## Plot versions:
# by x p_split [yields (3 x 2) x 2] = 12 versions:
plot_tab(by = "cddc", p_split = "v", p_lbl = "def") # v01 (see v07)
plot_tab(by = "cdac", p_split = "v", p_lbl = "def") # v02 (see v11)
plot_tab(by = "cddc", p_split = "h", p_lbl = "def") # v03 (see v05)
plot_tab(by = "dcac", p_split = "h", p_lbl = "def") # v04 (see v09)

plot_tab(by = "dcdc", p_split = "v", p_lbl = "def") # v05 (is v03 rotated)
plot_tab(by = "dcac", p_split = "v", p_lbl = "def") # v06 (see v12)
plot_tab(by = "dcdc", p_split = "h", p_lbl = "def") # v07 (is v01 rotated)
plot_tab(by = "dcac", p_split = "h", p_lbl = "def") # v08 (see v09)

plot_tab(by = "accd", p_split = "v", p_lbl = "def") # v09 (is v04 rotated)
plot_tab(by = "acdc", p_split = "v", p_lbl = "def") # v10 (is v08 rotated)
plot_tab(by = "accd", p_split = "h", p_lbl = "def") # v11 (is v02 rotated)
plot_tab(by = "acdc", p_split = "h", p_lbl = "def") # v12 (is v06 rotated)

## Explore labels and links:
plot_tab(f_lbl = "abb", p_lbl = NA) # abbreviated labels, no probability links
plot_tab(f_lbl = "num", f_lbl_sum = "num", f_lbl_hd = "num")
plot_tab(f_lbl = "num", f_lbl_sum = "num", f_lbl_hd = "num")
plot_tab(f_lbl = "num", f_lbl_sum = "num", f_lbl_hd = "num")

## Misc. options:
plot_tab(area = "sq") # area: square
plot_tab(title_lbl = "") # no titles
plot_tab(mar_notes = FALSE) # no margin notes

plot_tab(by = "cdcd", gaps = c(.08, .00), area = "sq") # gaps
plot_tree

Plot a tree diagram of frequencies and probabilities.

Description

plot_tree drew a tree diagram of frequencies (as nodes) and probabilities (as edges).

Usage

plot_tree(prev = num$prev, sens = num$sens, mirt = NA,
Spec = num$spec, fart = NA, N = freq$N, round = TRUE,
by = "cd", area = "no", pLbl = "num", show_accur = TRUE,
w_acc = 0.5, titleLbl = txt$scene_lbl, popu_lbl = txt$popu_lbl,
cond_true_lbl = txt$cond_true_lbl,
cond_false_lbl = txt$cond_false_lbl, dec_pos_lbl = txt$dec_pos_lbl,
dec_neg_lbl = txt$dec_neg_lbl, hi_lbl = txt$hi_lbl,
mi_lbl = txt$mi_lbl, fa_lbl = txt$fa_lbl, cr_lbl = txt$cr_lbl,
col_txt = grey(0.01, alpha = 0.99), cex_lbl = 0.85,
col_boxes = pal, col_border = grey(0.33, alpha = 0.99), lwd = 1.5,
box_lwd = 1.5, col_shadow = grey(0.11, alpha = 0.99),
cex_shadow = 0)

Arguments

prev The condition’s prevalence prev.
sens The decision’s sensitivity sens.
mirt The decision’s miss rate mirt.
spec The decision’s specificity value spec.
fart The decision’s false alarm rate fart.
N The number of individuals in the population.
round A Boolean option specifying whether computed frequencies are rounded to
integers. Default: round = TRUE.
by A character code specifying the perspective (or category by which the popula-
tion is split into subsets) with 3 options:
1. "cd" ... by condition;
2. "dc" ... by decision;
3. "ac" ... by accuracy.
area A character code specifying the area of the boxes (or their relative sizes) with 3
options:
1. "no" ... all boxes are shown with the same size;
2. "sq" ... boxes are squares with area sizes scaled proportional to frequencies
   (default);
3. "hr" ... boxes are horizontal rectangles with area sizes scaled proportional
to frequencies.
pLbl A character code specifying the type of probability information (on edges) with
4 options:
1. "nam" ... names of probabilities;
2. "num" ... numeric values of probabilities (rounded to 3 decimals, default);
3. "mix" ... names of essential probabilities, values of complements;
4. "min" ... minimal labels: names of essential probabilities.
show_accur Option for showing current accuracy metrics accu on the margin of the plot.
w_acc Weighting parameter w used to compute weighted accuracy w_acc in comp_accur_freq.
Various other options allow the customization of text labels and colors:
**plot_tree**

- **title_lbl**: Text label for current plot title.
- **popu_lbl**: Text label for current population `popu`.
- **cond_true_lbl**: Text label for current cases of `cond_true`.
- **cond_false_lbl**: Text label for current cases of `cond_false`.
- **dec_pos_lbl**: Text label for current cases of `dec_pos`.
- **dec_neg_lbl**: Text label for current cases of `dec_neg`.
- **hi_lbl**: Text label for hits `hi`.
- **mi_lbl**: Text label for misses `mi`.
- **fa_lbl**: Text label for false alarms `fa`.
- **cr_lbl**: Text label for correct rejections `cr`.
- **col_txt**: Color for text labels (in boxes).
- **cex_lbl**: Scaling factor for text labels (in boxes and on arrows).
- **col_boxes**: Colors of boxes (a single color or a vector with named colors matching the number of current boxes). Default: Current color information contained in `pal`.
- **lwd**: Width of arrows.
- **box_lwd**: Width of boxes.
- **cex_shadow**: Scaling factor of shadows (values > 0 showing shadows). Default: `cex_shadow = 0`.

**Details**

plot_tree is deprecated – please use plot_prism instead.

**Value**

Nothing (NULL).

**See Also**

plot_prism is the new version of this function.

Other visualization functions: plot_riskyr, plot_area, plot_bar, plot_curve, plot_fnet, plot_icons, plot_mosaic, plot_plane, plot_prism, plot_tab

**Examples**

plot_tree()  # frequency tree with current default options (by = "cd")

# alternative perspectives:
plot_tree(by = "dc")  # tree by decision
plot_tree(by = "ac")  # tree by accuracy

# See plot_prism for details and additional options.
A population table based on current frequencies.

Description

`popu` is an R data frame that is computed by `comp_popu` from the current frequency information (contained in `freq`). Each individual is represented as a row; columns represent the individual’s condition (TRUE or FALSE), a corresponding decision (also encoded as TRUE = positive or FALSE = negative), and its classification (in SDT terms) as either true positive (an individual hit `hi`), false negative (an individual miss `mi`), false positive (an individual false alarm `fa`), or true negative (an individual correct rejection `cr`).

Usage

`popu`

Format

An object of class `NULL` of length 0.

Details

`#` `popu` is initialized to `NULL` and needs to be computed by calling `comp_popu` with current parameter settings.

`comp_popu` uses the current text information contained in `txt` to define the labels of conditions, decisions, and SDT classifications.

A visualization of the current population `popu` is provided by `plot_icons`.

Value

A data frame `popu` containing `N` rows (individual cases) and 3 columns ("Truth", "Decision", "SDT") encoded as ordered factors (with 2, 2, and 4 levels, respectively).

See Also

the corresponding generating function `comp_popu`; `read_popu` interprets a data frame as a riskyr scenario; `num` for basic numeric parameters; `freq` for current frequency information; `txt` for current text settings.

Examples

```r
popu <- comp_popu()  # => initializes popu with current values of freq and txt
dim(popu)             # => N x 3
head(popu)            # => shows head of data frame
```
ppod

The proportion (or baseline) of a positive decision.

Description

ppod defines the proportion (baseline probability or rate) of a decision being positive (but not necessarily accurate/correct).

Usage

ppod

Format

An object of class numeric of length 1.

Details

Understanding or obtaining the proportion of positive decisions ppod:

• Definition: ppod is the (non-conditional) probability:
  ppod = p(decision = positive)
  or the base rate (or baseline probability) of a decision being positive (but not necessarily accurate/correct).

• Perspective: ppod classifies a population of N individuals by decision (ppod = dec_pos/N).
  ppod is the "by decision" counterpart to prev (which adopts a "by condition" perspective).

• Alternative names: base rate of positive decisions (PR), proportion predicted or diagnosed, rate of decision cases

• In terms of frequencies, ppod is the ratio of dec_pos (i.e., hi + fa) divided by N (i.e., hi + mi + fa + cr):
  ppod = dec_pos/N = (hi + fa)/(hi + mi + fa + cr)

• Dependencies: ppod is a feature of the decision process or diagnostic procedure.
  However, the conditional probabilities sens, mirt, spec, fart, PPV, and NPV also depend on the condition’s prevalence prev.

References

Consult Wikipedia for additional information.

See Also

prob contains current probability information; comp_prob computes current probability information; num contains basic numeric parameters; init_num initializes basic numeric parameters; freq contains current frequency information; comp_freq computes current frequency information; is_prob verifies probabilities.

Other probabilities: FDR, FOR, NPV, PPV, acc, err, fart, mirt, prev, sens, spec
Examples

```r
ppod <- .50    # sets a rate of positive decisions of 50%
ppod <- 50/100 # (decision = TRUE) for 50 out of 100 individuals
is_prob(ppod)   # TRUE
```

PPV

*The positive predictive value of a decision process or diagnostic procedure.*

Description

PPV defines some decision’s positive predictive value (PPV): The conditional probability of the condition being TRUE provided that the decision is positive.

Usage

PPV

Format

An object of class numeric of length 1.

Details

Understanding or obtaining the positive predictive value PPV:

- **Definition:** PPV is the conditional probability for the condition being TRUE given a positive decision:
  \[ PPV = p(\text{condition} = \text{TRUE} \mid \text{decision} = \text{positive}) \]
  or the probability of a positive decision being correct.
- **Perspective:** PPV further classifies the subset of `dec_pos` individuals by condition (\( PPV = \text{hi}/\text{dec_pos} = \text{hi}/(\text{hi} + \text{fa}) \)).
- **Alternative names:** precision
- **Relationships:**
  a. PPV is the complement of the false discovery or false detection rate FDR:
     \[ PPV = 1 - \text{FDR} \]
  b. PPV is the opposite conditional probability – but not the complement – of the sensitivity sens:
     \[ \text{sens} = p(\text{decision} = \text{positive} \mid \text{condition} = \text{TRUE}) \]
- **In terms of frequencies, PPV is the ratio of hi divided by dec_pos (i.e., hi + fa):**
  \[ PPV = \text{hi}/\text{dec_pos} = \text{hi}/(\text{hi} + \text{fa}) \]
- **Dependencies:** PPV is a feature of a decision process or diagnostic procedure and – similar to the sensitivity sens – a measure of correct decisions (positive decisions that are actually TRUE).

However, due to being a conditional probability, the value of PPV is not intrinsic to the decision process, but also depends on the condition’s prevalence value `prev`. 
References
Consult Wikipedia for additional information.

See Also
comp_PPV computes PPV; prob contains current probability information; comp_prob computes current probability information; num contains basic numeric parameters; init_num initializes basic numeric parameters; comp_freq computes current frequency information; is_prob verifies probabilities.

Other probabilities: FDR, FOR, NPV, acc, err, fart, mirt, ppod, prev, sens, spec

Examples
PPV <- .55  # sets a positive predictive value of 55%
PPV <- 55/100  # (condition = TRUE) for 55 out of 100 people with (decision = positive)
is_prob(PPV)  # TRUE

prev

The prevalence (baseline probability) of a condition.

Description
prev defines a condition’s prevalence value (or baseline probability): The probability of the condition being TRUE.

Usage
prev

Format
An object of class numeric of length 1.

Details
Understanding or obtaining the prevalence value prev:

- Definition: prev is the (non-conditional) probability:
  \[ \text{prev} = p(\text{condition} = \text{TRUE}) \]
  or the base rate (or baseline probability) of the condition’s occurrence or truth.
- In terms of frequencies, prev is the ratio of cond_true (i.e., hi + mi) divided by N (i.e., hi + mi + fa + cr):
  \[ \text{prev} = \frac{\text{cond_true}}{\text{N}} = \frac{(\text{hi} + \text{mi})}{(\text{hi} + \text{mi} + \text{fa} + \text{cr})} \]
- Perspective: prev classifies a population of N individuals by condition (prev = cond_true/N). prev is the "by condition" counterpart to ppod (when adopting a "by decision" perspective) and to acc (when adopting a "by accuracy" perspective).
• Alternative names: base rate of condition, proportion affected, rate of condition = TRUE cases. prev is often distinguished from the incidence rate (i.e., the rate of new cases within a certain time period).

• Dependencies: prev is a feature of the population and of the condition, but independent of the decision process or diagnostic procedure. While the value of prev does not depend on features of the decision process or diagnostic procedure, prev must be taken into account when computing the conditional probabilities sens, mirt, spec, fart, PPV, and NPV (as they depend on prev).

References
Consult Wikipedia for additional information.

See Also
prob contains current probability information; num contains basic numeric variables; init_num initializes basic numeric variables; comp_prob computes derived probabilities; comp_freq computes natural frequencies from probabilities; is_prob verifies probabilities.
Other probabilities: FDR, FOR, NPV, PPV, acc, err, fart, mirt, ppod, sens, spec
Other essential parameters: cr, fa, hi, mi, sens, spec

Examples
prev <- .10      # sets a prevalence value of 10%
prev <- 10/100   # (condition = TRUE) for 10 out of 100 individuals
is_prob(prev)    # TRUE

print.summary.riskyr  Print summary information of a riskyr scenario.

Description
print.summary.riskyr provides a print method for objects of class "summary.riskyr".

Usage
## S3 method for class 'summary.riskyr'
print(x = NULL, ...)

Arguments
x An object of class "summary.riskyr", usually a result of a call to summary.riskyr.
... Additional parameters (to be passed to generic print function).
**prob**

**Format**

Printed output of a "summary.riskyr" object.

**See Also**

riskyr initializes a riskyr scenario.

**Examples**

```r
summary(scenarios$n4)
```

---

**Description**

`prob` is a list of named numeric variables containing 3 essential (1 non-conditional `prev` and 2 conditional `sens` and `spec`) probabilities and 8 derived (`ppod` and `acc`, as well as 6 conditional) probabilities:

**Usage**

```r
prob
```

**Format**

An object of class `list` of length 13.

**Details**

`prob` currently contains the following probabilities:

1. the condition’s prevalence `prev` (i.e., the probability of the condition being TRUE): prev = cond_true/\(N\).
2. the decision’s sensitivity `sens` (i.e., the conditional probability of a positive decision provided that the condition is TRUE).
3. the decision’s miss rate `mirt` (i.e., the conditional probability of a negative decision provided that the condition is TRUE).
4. the decision’s specificity `spec` (i.e., the conditional probability of a negative decision provided that the condition is FALSE).
5. the decision’s false alarm rate `fart` (i.e., the conditional probability of a positive decision provided that the condition is FALSE).
6. the proportion (baseline probability or rate) of the decision being positive `ppod` (but not necessarily true): ppod = dec_pos/\(N\).
7. the decision’s positive predictive value `PPV` (i.e., the conditional probability of the condition being TRUE provided that the decision is positive).
8. the decision’s false detection (or false discovery) rate FDR (i.e., the conditional probability of the condition being FALSE provided that the decision is positive).
9. the decision’s negative predictive value NPV (i.e., the conditional probability of the condition being FALSE provided that the decision is negative).
10. the decision’s false omission rate FOR (i.e., the conditional probability of the condition being TRUE provided that the decision is negative).
11. the accuracy acc (i.e., probability of correct decisions dec_cor or correspondence of decisions to conditions).
12. the conditional probability p_acc_hi (i.e., the probability of hi given that the decision is correct dec_cor).
13. the conditional probability p_err_fa (i.e., the probability of fa given that the decision is erroneous dec_err).

These probabilities are computed from basic probabilities (contained in num) and computed by using comp_prob.

The list prob is the probability counterpart to the list containing frequency information freq.

Note that inputs of extreme probabilities (of 0 or 1) may yield unexpected values (e.g., an NPV value of NaN when is_extreme_prob_set evaluates to TRUE).

Key relationships between frequencies and probabilities (see documentation of comp_freq or comp_prob for details):

- Three perspectives on a population:
  - by condition / by decision / by accuracy.
- Defining probabilities in terms of frequencies:
  - Probabilities can be computed as ratios between frequencies, but beware of rounding issues.

Functions translating between representational formats: comp_prob_prob, comp_prob_freq, comp_freq_prob, comp_freq_freq (see documentation of comp_prob_prob for details).

Visualizations of current probability information are provided by plot_area, plot_prism, and plot_curve.

See Also

num contains basic numeric parameters; init_num initializes basic numeric parameters; txt contains current text information; init_txt initializes text information; pal contains current color information; init_pal initializes color information; freq contains current frequency information; comp_freq computes current frequency information; prob contains current probability information; comp_prob computes current probability information; accu contains current accuracy information.

Other lists containing current scenario information: accu, freq, num, pal_bw, pal_kn, pal_mbw, pal_mod, pal_org, pal_rgb, pal_vir, pal, txt_TF, txt_org, txt

Examples

```r
prob <- comp_prob() # => initialize prob to default parameters
prob # => show current values
length(prob) # => 13
```
**read_popu**

Read a population (given as data frame) into a riskyr scenario.

---

**Description**

`read_popu` interprets a data frame `df` (that contains individual observations of some population) and returns a scenario of class "riskyr".

**Usage**

```r
code
read_popu(df = popu, ix_by_top = 1, ix_by_bot = 2, ix_sdt = 3,
  hi_lbl = txt$hi_lbl, mi_lbl = txt$mi_lbl, fa_lbl = txt$fa_lbl,
  cr_lbl = txt$cr_lbl, ...)  
```

**Arguments**

- `df` A data frame providing a population `popu` of individuals, which are identified on at least 2 binary variables and classified into 4 cases in a 3rd variable. Default: `df = popu` (as data frame).
- `ix_by_top` Index of variable (column) providing the 1st (top) perspective (in `df`). Default: `ix_by_top = 1` (1st column).
- `ix_by_bot` Index of variable (column) providing the 2nd (bot) perspective (in `df`). Default: `ix_by_bot = 2` (2nd column).
- `ix_sdt` Index of variable (column) providing a classification into 4 cases (in `df`). Default: `ix_sdt = 3` (3rd column).
- `hi_lbl` Variable label of cases classified as hi (TP).
- `mi_lbl` Variable label of cases classified as mi (FN).
- `fa_lbl` Variable label of cases classified as fa (FP).
- `cr_lbl` Variable label of cases classified as cr (TN).
- `...` Additional parameters (to be passed to `riskyr` function).

**Details**

Note that `df` needs to be structured according to the `popu` created by `comp_popu`.

**Value**

An object of class "riskyr" describing a risk-related scenario.

**See Also**

the corresponding data frame `popu`; the corresponding generating function `comp_popu`; `riskyr` initializes a riskyr scenario.

Other riskyr scenario functions: `plot.riskyr`, `riskyr`, `summary.riskyr`
Examples

# Generating and interpreting different scenario types:

# (A) Diagnostic/screening scenario (using default labels):
---
popu_diag <- comp_popu(hi = 4, mi = 1, fa = 2, cr = 3)
# popu_diag
scen_diag <- read_popu(popu_diag, scen_lbl = "Diagnostics", popu_lbl = "Population tested")
plot(scen_diag, type = "prism", area = "no", f_lbl = "numnum")

# (B) Intervention/treatment scenario: ---
popu_treat <- comp_popu(hi = 80, mi = 20, fa = 45, cr = 55,
            cond_lbl = "Treatment", cond_true_lbl = "pill", cond_false_lbl = "placebo",
            dec_lbl = "Health status", dec_pos_lbl = "healthy", dec_neg_lbl = "sick")
# popu_treat
scen_treat <- read_popu(popu_treat, scen_lbl = "Treatment", popu_lbl = "Population treated")
plot(scen_treat, type = "prism", area = "sq", f_lbl = "namnum", p_lbl = "num")
plot(scen_treat, type = "icon", lbl_txt = txt_org, col_pal = pal_org)

# (C) Prevention scenario (e.g., vaccination): ---
popu_vacc <- comp_popu(hi = 960, mi = 40, fa = 880, cr = 120,
            cond_lbl = "Vaccination", cond_true_lbl = "yes", cond_false_lbl = "no",
            dec_lbl = "Disease", dec_pos_lbl = "no flu", dec_neg_lbl = "flu")
# popu_vacc
scen_vacc <- read_popu(popu_vacc, scen_lbl = "Prevention", popu_lbl = "Population vaccinated")
plot(scen_vacc, type = "prism", area = "sq", f_lbl = "namnum", col_pal = pal_bw, p_lbl = "num")

---

riskyr

Create a riskyr scenario.

Description

riskyr creates a scenario of class "riskyr", which can be visualized by the plot method plot.riskyr and summarized by the summary method summary.riskyr.

Usage

riskyr(scen_lbl = txt$scen_lbl, popu_lbl = txt$popu_lbl,
        N_lbl = txt$N_lbl, cond_lbl = txt$cond_lbl,
        cond_true_lbl = txt$cond_true_lbl,
        cond_false_lbl = txt$cond_false_lbl, dec_lbl = txt$dec_lbl,
        dec_pos_lbl = txt$dec_pos_lbl, dec_neg_lbl = txt$dec_neg_lbl,
        acc_lbl = txt$acc_lbl, dec_cor_lbl = txt$dec_cor_lbl,
        dec_err_lbl = txt$dec_err_lbl, sdt_lbl = txt$sdt_lbl,
        hi_lbl = txt$hi_lbl, mi_lbl = txt$mi_lbl, fa_lbl = txt$fa_lbl,
        cr_lbl = txt$cr_lbl, prev = NA, sens = NA, spec = NA,
        fart = NA, N = NA, hi = NA, mi = NA, fa = NA, cr = NA,
        scen_lng = txt$scen_lng, scen_txt = txt$scen_txt,
        scen_src = txt$scen_src, scen_apa = txt$scen_apa)
Arguments

scen_lbl The current scenario title (sometimes in Title Caps).

popu_lbl A brief description of the current population or sample.

N_lbl A label for the current population popu or sample.

cond_lbl A label for the condition or feature (e.g., some disease) currently considered.

cond_true_lbl A label for the presence of the current condition or cond_true cases (the condition’s true state of TRUE).

cond_false_lbl A label for the absence of the current condition or cond_false cases (the condition’s true state of FALSE).

dec_lbl A label for the decision or judgment (e.g., some diagnostic test) currently made.

dec_pos_lbl A label for positive decisions or dec_pos cases (e.g., predicting the presence of the condition).

dec_neg_lbl A label for negative decisions or dec_neg cases (e.g., predicting the absence of the condition).

acc_lbl A label for accuracy (i.e., correspondence between condition and decision or judgment).

dec_cor_lbl A label for correct (or accurate) decisions or judgments.

dec_err_lbl A label for incorrect (or erroneous) decisions or judgments.

sdt_lbl A label for the combination of condition and decision currently made.

hi_lbl A label for hits or true positives hi (i.e., correct decisions of the presence of the condition, when the condition is actually present).

mi_lbl A label for misses or false negatives mi (i.e., incorrect decisions of the absence of the condition when the condition is actually present).

fa_lbl A label for false alarms or false positives fa (i.e., incorrect decisions of the presence of the condition when the condition is actually absent).

cr_lbl A label for correct rejections or true negatives cr (i.e., a correct decision of the absence of the condition, when the condition is actually absent).

prev The condition’s prevalence prev (i.e., the probability of condition being TRUE).

sens The decision’s sensitivity sens (i.e., the conditional probability of a positive decision provided that the condition is TRUE). sens is optional when its complement mirt is provided.

spec The decision’s specificity value spec (i.e., the conditional probability of a negative decision provided that the condition is FALSE). spec is optional when its complement fart is provided.

fart The decision’s false alarm rate fart (i.e., the conditional probability of a positive decision provided that the condition is FALSE). fart is optional when its complement spec is provided.

N The number of individuals in the scenario’s population. A suitable value of N is computed, if not provided.
hi  The number of hits hi (or true positives).
mi  The number of misses mi (or false negatives).
fa  The number of false alarms fa (or false positives).
   cr  The number of correct rejections cr (or true negatives).

Details and source information:

scen_lng  Language of the current scenario (as character code). Options: "en" for English, "de" for German.
scen_txt  A longer text description of the current scenario (which may extend over several lines).
scen_src  Source information for the current scenario.
scen_apa  Source information for the current scenario according to the American Psychological Association (APA style).

Format

An object of class "riskyr" with textual and numeric information describing a risk-related scenario.

Details

Beyond basic scenario information (i.e., text elements describing a scenario) only the population size \( N \) and the essential probabilities prev, sens, spec, and fart are used and returned.

Note:

- Basic text information and some numeric parameters (see num and init_num) are integral parts of a riskyr scenario.
- By contrast, basic color information (see pal and init_pal) is not an integral part, but independently defined.
- The names of probabilities (see prob) are currently not an integral part of txt and riskyr scenarios (but defined in prob_lbl_def and label_prob).

Value

An object of class "riskyr" describing a risk-related scenario.

Scenario-specific titles and text labels (see txt):

See Also

init_num and num for basic numeric parameters; init_txt and txt for current text settings; init_pal and pal for current color settings.

Other riskyr scenario functions: plot-riskyr, read_popu, summary.riskyr
Other functions initializing scenario information: init_num, init_pal, init_txt
Examples

# Defining scenarios: -----
# (a) minimal information:
  hustosis <- riskyr(scen_lbl = "Screening for hustosis",
                     N = 1000, prev = .04, sens = .80, spec = .95)

# (2) detailed information:
  scen_reoffend <- riskyr(scen_lbl = "Identify reoffenders",
                          cond_lbl = "being a reoffender",
                          popu_lbl = "Prisoners",
                          cond_true_lbl = "has reoffended",
                          cond_false_lbl = "has not reoffended",
                          dec_lbl = "test result",
                          dec_pos_lbl = "will reoffend",
                          dec_neg_lbl = "will not reoffend",
                          sdt_lbl = "combination",
                          hi_lbl = "reoffender found", mi_lbl = "reoffender missed",
                          fa_lbl = "false accusation", cr_lbl = "correct release",
                          prev = .45, sens = .98, spec = .46, fart = NA,
                          scen_src = "example scenario")

# Using scenarios: -----
summary(hustosis)
plot(hustosis)
  summary(scen_reoffend)
  plot(scen_reoffend)

# 2 ways of defining the same scenario: -----
  s1 <- riskyr(prev = .5, sens = .5, spec = .5, N = 100)  # s1: define by 3 prob & N
  s2 <- riskyr(hi = 25, mi = 25, fa = 25, cr = 25)  # s2: same scenario by 4 freq
  all.equal(s1, s2)  # should be TRUE

# Ways to work: -----
  riskyr(prev = .5, sens = .5, spec = .5, hi = 25, mi = 25, fa = 25, cr = 25)  # works (consistent)
  riskyr(prev = .5, sens = .5, spec = .5, hi = 25, mi = 25, fa = 25)  # works (ignores freq)

## Watch out for:
# riskyr(hi = 25, mi = 25, fa = 25, cr = 25, N = 101)  # warns, uses actual sum of freq
# riskyr(prev = .4, sens = .5, spec = .5, hi = 25, mi = 25, fa = 25, cr = 25)  # warns, uses freq

---

Description

Opens the riskyr package guides
Usage

scenarios

description

scenarios is a list of scenarios of class riskyr collected from the scientific literature and other sources and to be used by visualization and summary functions.

Usage

scenarios

Format

A list with currently 25 scenarios of class riskyr which are each described by 21 variables.

Details

scenarios currently contains the following scenarios (n1 to n12 in English language, n13 to n25 in German language):

1. Bowel cancer screening
2. Cab problem
3. Hemoccult test
4. Mammography screening
5. Mammography (freq)
6. Mammography (prob)
7. Mushrooms
8. Musical town
9. PSA test (baseline)
10. PSA test (patients)
11. Psilocybin screening
12. Sepsis
13. Amniozentese (in German language)
14. HIV-Test 1
15. HIV-Test 2
16. HIV-Test 3
17. HIV-Test 4
18. Mammografie 1
19. Mammografie 2
20. Mammografie 3
21. Mammografie 4
22. Nackenfaltentest (NFT) 1
23. Nackenfaltentest (NFT) 2
24. Sigmoidoskopie 1
25. Sigmoidoskopie 2

Variables describing a scenario:

1. scen_lbl: Text label for current scenario.
2. scen_lng: Language of current scenario (en/de).
3. scen_txt: Description text of current scenario.
4. popu_lbl: Text label for current population.
5. cond_lbl: Text label for current condition.
6. cond_true_lbl: Text label for cond_true cases.
7. cond_false_lbl: Text label for cond_false cases.
8. dec_lbl: Text label for current decision.
9. dec_pos_lbl: Text label for dec_pos cases.
10. dec_neg_lbl: Text label for dec_neg cases.
11. hi_lbl: Text label for cases of hits hi.
12. mi_lbl: Text label for cases of misses mi.
13. fa_lbl: Text label for cases of false alarms fa.
14. cr_lbl: Text label for cases of correct rejections cr.
15. prev: Value of current prevalence prev.
17. spec: Value of current specificity spec.
18. fart: Value of current false alarm rate fart.
19. n: Current population size N.
20. scen_src: Source information for current scenario.
21. scen_apa: Source information in APA format.

Note that names of variables (columns) correspond to a subset of init_txt (to initialize txt) and init_num (to initialize num).

The variables scen_src and scen_apa provide a scenario’s source information.

The information of scenarios is also contained in an R data frame df_scenarios (and generated from the corresponding .rda file in /data/).

See Also

riskyr initializes a riskyr scenario.
sens

**The sensitivity (or hit rate) of a decision process or diagnostic procedure.**

**Description**

sens defines a decision's sensitivity (or hit rate) value: The conditional probability of the decision being positive if the condition is TRUE.

**Usage**

sens

**Format**

An object of class numeric of length 1.

**Details**

Understanding or obtaining the sensitivity sens (or hit rate HR):

- **Definition:** sens is the conditional probability for a (correct) positive decision given that the condition is TRUE:

  \[
  \text{sens} = p(\text{decision} = \text{positive} \mid \text{condition} = \text{TRUE})
  \]

  or the probability of correctly detecting true cases (condition = TRUE).

- **Perspective:** sens further classifies the subset of cond_true individuals by decision (sens = hi/cond_true).

- **Alternative names:** true positive rate (TPR), hit rate (HR), probability of detection, power = 1 - beta, recall

- **Relationships:**
  a. sens is the complement of the miss rate mirt (aka. false negative rate FNR or the rate of Type-II errors):

    \[
    \text{sens} = (1 - \text{miss rate}) = (1 - \text{FNR})
    \]

  b. sens is the opposite conditional probability – but not the complement – of the positive predictive value PPV:

    \[
    \text{PPV} = p(\text{condition} = \text{TRUE} \mid \text{decision} = \text{positive})
    \]

  c. In terms of frequencies, sens is the ratio of hi divided by cond_true (i.e., hi + mi):

    \[
    \text{sens} = \frac{\text{hi}}{\text{cond_true}} = \frac{\text{hi}}{(\text{hi} + \text{mi})}
    \]

- **Dependencies:** sens is a feature of a decision process or diagnostic procedure and a measure of correct decisions (true positives). Due to being a conditional probability, the value of sens is not intrinsic to the decision process, but also depends on the condition’s prevalence value prev.

**References**

Consult Wikipedia for additional information.
spec

See Also

comp_sens computes sens as the complement of mirt; prob contains current probability information; comp_prob computes current probability information; num contains basic numeric parameters; init_num initializes basic numeric parameters; comp_freq computes current frequency information; is_prob verifies probabilities.

Other probabilities: FDR, FOR, NPV, PPV, acc, err, fart, mirt, ppod, prev, spec

Other essential parameters: cr, fa, hi, mi, prev, spec

Examples

sens <- .85  # sets a sensitivity value of 85%
sens <- 85/100  # (decision = positive) for 85 out of 100 people with (condition = TRUE)
is_prob(sens)  # TRUE

**Description**

spec defines a decision’s specificity value (or correct rejection rate): The conditional probability of the decision being negative if the condition is FALSE.

Usage

spec

Format

An object of class numeric of length 1.

Details

Understanding or obtaining the specificity value spec:

- Definition: spec is the conditional probability for a (correct) negative decision given that the condition is FALSE:
  
  \[
  \text{spec} = p(\text{decision} = \text{negative} \mid \text{condition} = \text{FALSE})
  \]
  
  or the probability of correctly detecting false cases (condition = FALSE).

- Perspective: spec further classifies the subset of cond_false individuals by decision (spec = cr/cond_false).

- Alternative names: true negative rate (TNR), correct rejection rate, 1 - alpha

- Relationships:
  
a. spec is the complement of the false alarm rate fart:
  
  \[
  \text{spec} = 1 - \text{fart}
  \]

b. spec is the opposite conditional probability – but not the complement – of the negative predictive value NPV:

\[
\text{NPV} = p(\text{condition} = \text{FALSE} \mid \text{decision} = \text{negative})
\]
• In terms of frequencies, spec is the ratio of cr divided by cond_false (i.e., fa + cr):
  \( \text{spec} = \frac{\text{cr}}{\text{cond}\_false} = \frac{\text{cr}}{\text{fa} + \text{cr}} \)
• Dependencies: spec is a feature of a decision process or diagnostic procedure and a measure
  of correct decisions (true negatives).
  However, due to being a conditional probability, the value of spec is not intrinsic to the deci-
  sion process, but also depends on the condition’s prevalence value prev.

References
Consult Wikipedia for additional information.

See Also
comp_spec computes spec as the complement of fart; prob contains current probability informa-
  tion; comp_prob computes current probability information; num contains basic numeric parameters;
init_num initializes basic numeric parameters; comp_freq computes current frequency informa-
  tion; is_prob verifies probabilities.
Other probabilities: FDR, FOR, NPV, PPV, acc, err, fart, mirt, ppod, prev, sens
Other essential parameters: cr, fa, hi, mi, prev, sens

Examples
spec <- .75    # sets a specificity value of 75%
spec <- 75/100 # (decision = negative) for 75 out of 100 people with (condition = FALSE)
is_prob(spec)  # TRUE

summary.riskyrt

Summarize a riskyr scenario.

Description
summary.riskyrt provides a summary method for objects of class "riskyr".

Usage
## S3 method for class 'riskyr'
summary(object = NULL, summarize = "all", ...)

Arguments
  object          An object of class "riskyr", usually a result of a call to riskyrt. Inbuilt scen- 
                  earios are also of type "riskyr".
  summarize       What is summarized as a vector consisting of c("freq", "prob", "accu") for
                  frequencies, probabilities, and accuracy respectively. The default "all" is an 
                  alias to all three.
  ...             Additional parameters (to be passed to summary functions).
txt

Format
An object of class summary.riskyr with up to 9 entries.

Value
A summary list obj.sum with up to 9 entries, dependent on which information is requested by summarize.
Scenario name, relevant condition, and N are summarized by default.

See Also
riskyr initializes a riskyr scenario.
Other riskyr scenario functions: plot.riskyr, read_popu, riskyr

Examples
summary(scenarios$n4)

Description
txt is initialized to a list of named elements to define basic scenario titles and labels.

Usage
txt

Format
An object of class list of length 21.

Details
All textual elements that specify generic labels and titles of riskyr scenarios are stored as named elements (of type character) in a list txt. To change an element, assign a new character object to an existing name.
The list txt is used throughout the riskyr package unless a scenario defines scenario-specific text labels (when using the riskyr function).

Note:
- Basic text information and some numeric parameters (see num and init_num) are integral parts of a riskyr scenario.
- By contrast, basic color information (see pal and init_pal) is not an integral part, but independently defined.
• The names of probabilities (see `prob`) are currently not an integral part of `txt` and `riskyr` scenarios (but defined in `prob_lbl_def` and `label_prob`).

txt currently contains the following text labels:

1. scen_lbl The current scenario title (sometimes in Title Caps).
2. scen_txt A longer text description of the current scenario (which may extend over several lines).
3. scen_src The source information for the current scenario.
4. scen_apa The source information in APA format.
5. scen_lng The language of the current scenario (as character code). Options: "en": English, "de": German.
6. popu_lbl A general name describing the current population.
7. n_lbl A short label for the current population `popu` or sample.
8. cond_lbl A general name for the condition dimension, or the feature (e.g., some disease) currently considered.
9. cond_true_lbl A short label for the presence of the current condition or `cond_true` cases (the condition's true state of being TRUE).
10. cond_false_lbl A short label for the absence of the current condition or `cond_false` cases (the condition's true state of being FALSE).
11. dec_lbl A general name for the decision dimension, or the judgment (e.g., some diagnostic test) currently made.
12. dec_pos_lbl A short label for positive decisions or `dec_pos` cases (e.g., predicting the presence of the condition).
13. dec_neg_lbl A short label for negative decisions or `dec_neg` cases (e.g., predicting the absence of the condition).
14. acc_lbl A general name for the accuracy dimension, or the correspondence between the condition currently considered and the decision judgment currently made.
15. dec_cor_lbl A short label for correct and accurate decisions or `dec_cor` cases (accurate predictions).
16. dec_err_lbl A short label for incorrect decisions or `dec_err` cases (erroneous predictions).
17. sdt_lbl A general name for all 4 cases/categories/cells of the 2x2 contingency table (e.g., condition x decision, using SDT).
18. hi_lbl A short label for hits or true positives `hi`/TP cases (i.e., correct decisions of the presence of the condition, when the condition is actually present).
19. mi_lbl A short label for misses or false negatives `mi`/FN cases (i.e., incorrect decisions of the absence of the condition when the condition is actually present).
20. fa_lbl A short label for false alarms or false positives `fa`/FP cases (i.e., incorrect decisions of the presence of the condition when the condition is actually absent).
21. cr_lbl A short label for correct rejections or true negatives `cr`/TN cases (i.e., a correct decision of the absence of the condition, when the condition is actually absent).
See Also

init_txt initializes text information; riskyr initializes a riskyr scenario; num contains basic numeric parameters; init_num initializes basic numeric parameters; pal contains current color information; init_pal initializes color information; freq contains current frequency information; comp_freq computes current frequency information; prob contains current probability information; comp_prob computes current probability information.

Other lists containing current scenario information: accu, freq, num, pal_bw, pal_kn, pal_mbw, pal_mod, pal_org, pal_rgb, pal_vir, pal, prob, txt_TF, txt_org

Examples

txt # Show all current names and elements
txt$scen_lbl # Show the current scenario label (e.g., used in plot titles)
txt$scen_lbl <- "My example" # Set a new scenario title

txt_org

List of original values of text elements.

Description

txt_org is a copy of the initial list of text elements to define all scenario titles and labels.

Usage

txt_org

Format

An object of class list of length 21.

Details

See txt for details and default text information.

Assign txt <- txt_org to re-set default text labels.

See Also

txt contains current text information; init_txt initializes text information; pal contains current color information; init_pal initializes color information.

Other lists containing current scenario information: accu, freq, num, pal_bw, pal_kn, pal_mbw, pal_mod, pal_org, pal_rgb, pal_vir, pal, prob, txt_TF, txt

Examples

txt_org # shows original text labels
txt_org["hi"] # shows the original label for hits ("hi")
txt_org["hi"] <- "TP" # defines a new label for hits (true positives, TP)
txt_TF

Alternative text labels (TP, FN, FP, TN).

Description

txt_TF is initialized to alternative text labels to define a frequency naming scheme in which (hi, mi, fa, cr) are called (TP, FN, FP, TN).

Usage

txt_TF

Format

An object of class list of length 21.

Details

See txt for details and default text information.
Assign txt <- txt_TF to use as default text labels.

See Also

txt contains current text information; init_txt initializes text information; pal contains current color information; init_pal initializes color information.

Other lists containing current scenario information: accu, freq, num, pal_bw, pal_kn, pal_mbw, pal_mod, pal_org, pal_rgb, pal_vir, pal, prob, txt_org, txt

Examples

```
txt_TF  # shows text labels of txt_TF
txt_TF["hi"]  # shows the current label for hits ("TP")
txt_TF["hi"] <- "hit"  # defines a new label for hits (true positives, TP)
```
Index

*Topic datasets

acc, 4
accu, 5
cond_false, 49
cond_true, 50
cr, 51
dec_cor, 52
dec_err, 53
dec_neg, 55
dec_pos, 56
df_scenarios, 57
err, 58
fa, 59
fart, 60
FDR, 61
FOR, 63
freq, 64
hi, 65
mi, 84
mirt, 85
N, 86
NPV, 87
num, 88
pal, 89
pal_bw, 90
pal_kn, 91
pal_mbw, 92
pal_mod, 93
pal_org, 93
pal_rgb, 94
pal_vir, 95
popu, 130
ppod, 131
PPV, 132
prev, 133
prob, 135
scenarios, 142
sens, 144
spec, 145
txt, 147
txt_org, 149
txt_TF, 150

acc, 4, 6, 7, 9, 10, 13, 16, 20, 21, 26, 40, 42, 53, 58, 59, 61, 62, 64, 86, 88, 131, 133–136, 145, 146
accu, 5, 6, 10, 12–16, 20, 21, 42, 58, 59, 65, 89–95, 109, 115, 128, 136, 149, 150
accurate (acc), 4
alpha (fart), 60
as_pb, 7, 8, 9, 72, 74–77, 79, 80, 82, 83, 96
as_pc, 8, 8, 72, 74–77, 79, 80, 82, 83, 96
baserate_cond_true (prev), 133
baserate_dec_pos (ppod), 131
beta (mirt), 85

comp_acc, 5, 7, 9, 13, 16–24, 33, 34, 37, 38, 42, 44, 47, 48, 58, 59
comp_accu_freq, 5–7, 10, 11, 12, 14–24, 33, 34, 37, 38, 42, 44, 47, 48, 58, 59, 109, 128
comp_accu_prob, 5–7, 10, 12, 13, 13, 14, 17–24, 33, 34, 37, 38, 42, 44, 47, 48, 58, 59
comp_comp_pair, 8–10, 13, 16–18, 19, 21–26, 29, 30, 33, 34, 37, 38, 42, 44, 46–48, 72
comp_complement, 5, 8–10, 13, 16, 16, 18–26, 29, 30, 33, 34, 37, 38, 42, 44, 46–48, 58, 72
comp_complete_prob_set, 10, 13, 16, 17, 19, 21–26, 29, 30, 33, 34, 37, 38, 42, 44, 46–48, 83
comp_err, 5, 7, 10, 13, 16–19, 20, 22–24, 33, 34, 37, 38, 42, 44, 47, 48, 59
comp_fart, 10, 13, 16–19, 21, 23, 24, 33, 34, 37, 38, 42, 44, 47, 48, 61
plot.prism, 63, 97, 101, 104, 107, 110, 113, 115, 117, 126, 129
plot_tree, 97, 101, 104, 107, 110, 113, 115, 117, 120, 123, 129
plot_table, 97, 101, 104, 107, 110, 113, 115, 117, 120, 123, 129
plot_table_r, 97, 101, 104, 107, 110, 113, 115, 117, 120, 123, 129
plot_tr, 97, 101, 104, 107, 110, 113, 115, 117, 120, 123, 129
plot_tr_r, 97, 101, 104, 107, 110, 113, 115, 117, 120, 123, 129
plot.tr, 97, 101, 104, 107, 110, 113, 115, 117, 120, 123, 129
plot_tr_r, 97, 101, 104, 107, 110, 113, 115, 117, 120, 123, 129
plot_tree, 97, 101, 104, 107, 110, 113, 115, 117, 120, 126, 127
popu, 13, 15, 35, 36, 70, 103, 104, 109, 117, 129, 130, 137, 139, 148
data_popu, 13, 15, 35, 36, 70, 103, 104, 109, 117, 129, 130, 137, 139, 148
power, 6, 12, 15
power(sens), 144
power(pod), 5, 10, 25, 36, 37, 40, 41, 55, 56, 59, 61, 62, 64, 86, 88, 131, 133–135, 145, 146
PPV, 5, 6, 12, 15, 26, 38, 40, 41, 45, 56, 59, 61, 62, 64, 69, 73, 86, 88, 90, 97, 106, 115, 131, 132, 134, 135, 144–146
PR(pod), 131
precision, 6, 12, 15, 56
precision(PPV), 132
precision(pod), 5, 9, 10, 14, 18, 20, 22–25, 28, 29, 31, 34, 36–41, 45, 46, 49, 50, 52, 59–64, 66, 67, 73, 78, 81, 83–86, 88, 98, 100, 103, 104, 106, 109, 111, 114–118, 123, 125, 128, 131–133, 133, 135, 139, 140, 143–146
print.summary.riskyr, 134
read_popu, 36, 97, 130, 137, 140, 147
recall, 6, 12, 15
recall(sens), 144
riskyr, 58, 67, 69, 71, 97, 135, 137, 138, 143, 144, 146, 147, 149
riskyr_guide, 141
scenarios, 57, 58, 97, 142
spec, 5, 6, 9, 12, 14, 15, 18, 20–25, 28, 29, 31, 34, 36–38, 40, 41, 45, 46, 48, 49, 50, 59–62, 64, 66, 67, 73, 78, 81, 83, 84, 86–88, 98, 100, 103, 104, 106, 109, 111, 115–118, 123, 125, 128, 131, 133–135, 139, 140, 143, 145, 146
summary_riskyr, 97, 137, 138, 140, 146
TN (cr), 31
TN, 6, 12, 15, 52
TNR (spec), 145
TP (hi), 65
tpr, 6, 12, 15
tpr(sens), 144
txt-org, 7, 65, 89–95, 136, 149, 149, 150
txt_TF, 7, 65, 89–95, 136, 149, 150
type-I-errors (fa), 59
type-II-errors (mi), 84