

Package ‘ez’

October 7, 2009

Type Package

Title Easy analysis and visualization of factorial experiments.

Version 1.4.2

Date 2009-10-07

Author Michael A. Lawrence

Maintainer Michael A. Lawrence <Mike.Lawrence@dal.ca>

Depends R (>= 2.9.0), car (>= 1.2-15), reshape (>= 0.8.3), plyr (>= 0.1.9), ggplot2 (>= 0.8.3)

Description This package facilitates easy analysis of factorial experiments, including purely within-Ss designs (a.k.a. “repeated measures”), purely between-Ss designs, and mixed within-and-between-Ss designs. The functions in this package provide easy access to descriptive statistics, ANOVA, permutation tests, and visualization of results.

License GPL-3

LazyLoad yes

Repository CRAN

Date/Publication 2009-10-07 15:32:58

R topics documented:

ez-package	2
ANT	2
ezANOVA	3
ezPerm	7
ezPlot	9
ezStats	13

Index	16
--------------	-----------

 ez-package

Easy analysis and visualization of factorial experiments.

Description

This package facilitates easy analysis of factorial experiments, including purely within-Ss designs (a.k.a. "repeated measures"), purely between-Ss designs, and mixed within-and-between-Ss designs. The functions in this package provide easy access to descriptive statistics ([ezStats](#)), ANOVA ([ezANOVA](#)), non-parametric permutation tests ([ezPerm](#)), and visualization of results ([ezPlot](#)).

Details

Package:	ez
Type:	Package
Version:	1.3
Date:	2009-05-19
License:	GPL-3
LazyLoad:	yes

This package contains several useful functions:

[ezANOVA](#) Provides simple interface to ANOVA, including assumption checks.

[ezPerm](#) Provides simple interface to the Permutation test.

[ezPlot](#) Uses the `ggplot2` graphing package to generate plots for any given user-requested effect, by default producing error bars that facilitate visual post-hoc multiple comparisons.

[ezStats](#) Provides between-Ss descriptive statistics for any given user-requested effect.

Author(s)

Author & Maintainer: Michael A. Lawrence <Mike.Lawrence@dal.ca>

See Also

[ezANOVA](#), [ezPerm](#), [ezPlot](#), [ezStats](#)

 ANT

ANT data

Description

Simulated data from then Attention Network Test (see reference below), consisting of 2 within-Ss variables ("cue" and "flanker"), 1 between-Ss variable ("group") and 2 dependent variables (response time, "rt", and accuracy, "acc")

Usage

```
data (ANT)
```

Format

A data frame with 5760 observations on the following 10 variables.

sid a factor with levels 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

group a factor with levels Control Treatment

block a numeric vector

trial a numeric vector

cue a factor with levels No Cue Center Cue Double Cue Spatial Cue

flanker a factor with levels Neutral Flanker Congruent Flanker Incongruent
Flanker

location a factor with levels down up

direction a factor with levels left right

rt a numeric vector

acc a numeric vector

References

J Fan, BD McCandliss, T Sommer, A Raz, MI Posner (2002). Testing the efficiency and independence of attentional networks. *Journal of Cognitive Neuroscience*, **14**, 340-347.

See Also

[ezANOVA](#), [ezPerm](#), [ezPlot](#), [ezStats](#)

Examples

```
data (ANT)
head (ANT)
str (ANT)
summary (ANT)
```

ezANOVA

Function to perform a factorial ANOVA

Description

This function provides easy analysis of data from factorial experiments, including purely within-Ss designs (a.k.a. "repeated measures"), purely between-Ss designs, and mixed within-and-between-Ss designs, yielding ANOVA results and assumption checks.

Usage

```
ezANOVA (
  data
  , dv
  , sid
  , within = NULL
  , between = NULL
)
```

Arguments

<code>data</code>	Data frame containing the data to be analyzed.
<code>dv</code>	.() object specifying the column in <code>data</code> that contains the dependent variable. Values in this column must be numeric.
<code>sid</code>	.() object specifying the column in <code>data</code> that contains the variable specifying the case/Ss identifier.
<code>within</code>	Optional .() object specifying one or more columns in <code>data</code> that contain independent variables that are manipulated within-Ss.
<code>between</code>	Optional .() object specifying one or more columns in <code>data</code> that contain independent variables that are manipulated between-Ss.

Details

While `within` and `between` are both optional, at least one column of `data` must be provided to either `within` or `between`. Any numeric or character variables in `data` that are specified as either `sid`, `within` or `between` will be converted to a factor with a warning. Prior to running, `dv` is collapsed to a mean for each cell defined by the combination of `sid`, `within` or `between`.

Value

A list containing one or more of the following components:

<code>ANOVA</code>	A data frame containing the ANOVA results.
<code>Mauchly's Test for Sphericity</code>	If any within-Ss variables with >2 levels are present, a data frame containing the results of Mauchly's test for Sphericity. Only reported for effects >2 levels because sphericity necessarily holds for effects with only 2 levels.
<code>Sphericity Corrections</code>	If any within-Ss variables are present, a data frame containing the Greenhouse-Geisser & Huynh-Feldt epsilon values, and corresponding corrected p-values.
<code>Levene's Test for Homogeneity</code>	If the design is purely between-Ss, a data frame containing the results of Levene's test for Homogeneity of variance.

Warning

The statistical computing in `ezAnova()` is driven by the `Anova` function from the `car` package, using the univariate Type-II test. If there are too few Ss in the data set for `Anova` to compute its MANOVA, `ezAnova()` will revert to using `aoV` for computing the ANOVA, in which case no assumption tests are provided. When assumption tests are provided via `Anova`, Huynh-Feldt corrected p-values where the Huynh-Feldt epsilon >1 will use 1 as the correction epsilon.

Note

Some column names in the output data frames are abbreviated to conserve space:

DFn Degrees of Freedom in the numerator (a.k.a. DFeffect).

DFd Degrees of Freedom in the denominator (a.k.a. DFerror).

SSn Sum of Squares in the numerator (a.k.a. SSeffect).

SSd Sum of Squares in the denominator (a.k.a. SSerror).

F F-value.

p p-value (probability of the null hypothesis given the data).

p<.05 Highlights p-values less than the traditional alpha level of .05.

pes Partial Eta-Squared ($SSn/(SSn+SSd)$).

GGe Greenhouse-Geisser epsilon.

p[GGe] p-value after correction using Greenhouse-Geisser epsilon.

p[GGe <.05] Highlights p-values (after correction using Greenhouse-Geisser epsilon) less than the traditional alpha level of .05.

HFe Huynh-Feldt epsilon.

p[HFe] p-value after correction using Huynh-Feldt epsilon.

p[HFe <.05] Highlights p-values (after correction using Huynh-Feldt epsilon) less than the traditional alpha level of .05.

W Mauchly's W statistic

Author(s)

Michael A. Lawrence (Mike.Lawrence@dal.ca)

See Also

[ezPerm](#), [ezPlot](#), [ezStats](#)

Examples

```
#Read in the ANT data (see ?ANT).
data(ANT)
```

```
#Show summaries of the ANT data.
head(ANT)
str(ANT)
```

```

summary(ANT)

#Compute some useful statistics per cell.
cell_stats = ddply(
  .data = ANT
  , .variables = .( sid , group , cue , flanker )
  , .fun <- function(x){
    #Compute error rate as percent.
    error_rate = (1-mean(x$acc))*100
    #Compute mean RT (only accurate trials).
    mean_rt = mean(x$rt[x$acc==1])
    #Compute SD RT (only accurate trials).
    sd_rt = sd(x$rt[x$acc==1])
    return(c(error_rate=error_rate,mean_rt=mean_rt,sd_rt=sd_rt))
  }
)

#Run an ANOVA on the mean_rt data.
mean_rt_anova = ezANOVA(
  data = cell_stats
  , dv = .(mean_rt)
  , sid = .(sid)
  , within = .(cue,flanker)
  , between = .(group)
)

#Show the ANOVA & assumption tests.
print(mean_rt_anova)

#Run an ANOVA on the mean_rt data, ignoring group.
mean_rt_anova2 = ezANOVA(
  data = cell_stats
  , dv = .(mean_rt)
  , sid = .(sid)
  , within = .(cue,flanker)
)

#Show the ANOVA & assumption tests.
print(mean_rt_anova2)

#Run a purely between-Ss ANOVA on the mean_rt data.
##Note how ezANOVA automatically collapses the unspecified within-Ss data.
mean_rt_anova3 = ezANOVA(
  data = cell_stats
  , dv = .(mean_rt)
  , sid = .(sid)
  , between = .(group)
)

#Show the ANOVA & assumption tests.
print(mean_rt_anova3)

```

`ezPerm`*Function to perform a factorial permutation test*

Description

This function provides easy non-parametric permutation test analysis of data from factorial experiments, including purely within-Ss designs (a.k.a. "repeated measures"), purely between-Ss designs, and mixed within-and-between-Ss designs.

Usage

```
ezPerm(  
  data  
  , dv  
  , sid  
  , within = NULL  
  , between = NULL  
  , perms  
)
```

Arguments

<code>data</code>	Data frame containing the data to be analyzed.
<code>dv</code>	.() object specifying the column in <code>data</code> that contains the dependent variable. Values in this column must be numeric.
<code>sid</code>	.() object specifying the column in <code>data</code> that contains the variable specifying the case/Ss identifier.
<code>within</code>	Optional .() object specifying one or more columns in <code>data</code> that contain independent variables that are manipulated within-Ss.
<code>between</code>	Optional .() object specifying one or more columns in <code>data</code> that contain independent variables that are manipulated between-Ss.
<code>perms</code>	An integer > 0 specifying the number of permutations to compute.

Details

While `within` and `between` are both optional, at least one column of `data` must be provided to either `within` or `between`. Any numeric or character variables in `data` that are specified as either `sid`, `within` or `between` will be converted to a factor with a warning. The expected standard deviation of p-values is approximately $\sqrt{\text{true_p} * (1 - \text{true_p}) / \text{perms}}$; significance tests using an alpha of .05 should therefore employ at least 1e3 permutations. As the permutation test is computationally intensive, it is advisable to pre-test smaller values of `perms` and extrapolate to estimate the total test duration before attempting a full run. To facilitate such extrapolation, test duration is provided in the output after running a permutation test.

Value

A list containing one or more of the following components:

Permutation Test

A data frame containing the permutation test results.

Test Duration

An estimate of the test duration in seconds.

Warning

`ezPerm()` is a work in progress. Under the current implementation, only main effects may be trusted.

Author(s)

Michael A. Lawrence (Mike.Lawrence@dal.ca)

See Also

[ezANOVA](#), [ezPlot](#), [ezStats](#)

Examples

```
#Read in the ANT data (see ?ANT).
data(ANT)

#Show summaries of the ANT data.
head(ANT)
str(ANT)
summary(ANT)

#Compute some useful statistics per cell.
cell_stats = ddply(
  .data = ANT
  , .variables = .( sid , group , cue , flanker )
  , .fun <- function(x){
    #Compute error rate as percent.
    error_rate = (1-mean(x$acc))*100
    #Compute mean RT (only accurate trials).
    mean_rt = mean(x$rt[x$acc==1])
    #Compute SD RT (only accurate trials).
    sd_rt = sd(x$rt[x$acc==1])
    return(c(error_rate=error_rate,mean_rt=mean_rt,sd_rt=sd_rt))
  }
)

#Compute the grand mean RT per Ss.
gmrt = ddply(
  .data = cell_stats
  , .variables = .( sid , group )
  , .fun <- function(x){
```

```

        y = mean(x$mean_rt)
        return(c(y=y))
    }
)

#Run a purely between-Ss ANOVA on the mean_rt data.
# (Completes after ~30s on a 2.4GHz processor).
mean_rt_perm = ezPerm(
  data = gmrt
  , dv = .(y)
  , sid = .(sid)
  , between = .(group)
  , perms = 1e3
)

#Show the Permutation test.
print(mean_rt_perm)

```

 ezPlot

Function to plot data from a factorial experiment

Description

This function provides easy visualization of any given user-requested effect from factorial experiments, including purely within-Ss designs (a.k.a. "repeated measures"), purely between-Ss designs, and mixed within-and-between-Ss designs. By default, Fisher's Least Significant Difference is computed to provide error bars that facilitate visual post-hoc multiple comparisons (see Warning section below).

Usage

```

ezPlot(
  data
  , dv
  , sid
  , within = NULL
  , between = NULL
  , x
  , do_lines = TRUE
  , do_bars = TRUE
  , bar_width = NULL
  , bar_size = NULL
  , split = NULL
  , row = NULL
  , col = NULL
  , to_numeric = NULL
  , x_lab = NULL
  , y_lab = NULL
)

```

```

, split_lab = NULL
, levels = NULL
)

```

Arguments

<code>data</code>	Data frame containing the data to be analyzed.
<code>dv</code>	.() object specifying the column in <code>data</code> that contains the dependent variable. Values in this column should be of the numeric class.
<code>sid</code>	.() object specifying the column in <code>data</code> that contains the variable specifying the case/Ss identifier. Values in this column will be converted to factor class if necessary.
<code>within</code>	Optional .() object specifying the column(s) in <code>data</code> that contains independent variables that are manipulated within-Ss. Values in this column will be converted to factor class if necessary.
<code>between</code>	Optional .() object specifying the column(s) in <code>data</code> that contains independent variables that are manipulated between-Ss. Values in this column will be converted to factor class if necessary.
<code>x</code>	.() object specifying the variable to plot on the x-axis.
<code>do_lines</code>	Logical. If TRUE, lines will be plotted connecting groups of points.
<code>do_bars</code>	Logical. If TRUE, error bars will be plotted.
<code>bar_width</code>	Optional numeric value specifying custom widths for the error bar hat.
<code>bar_size</code>	Optional numeric value or vector specifying custom size of the error bars.
<code>split</code>	Optional .() object specifying a variable by which to split the data into different shapes/colors (and line types, if <code>do_lines==TRUE</code>).
<code>row</code>	Optional .() object specifying a variable by which to split the data into rows.
<code>col</code>	Optional .() object specifying a variable by which to split the data into columns.
<code>to_numeric</code>	Optional .() object specifying any variables that need to be converted to the numeric class before plotting.
<code>x_lab</code>	Optional character string specifying the x-axis label.
<code>y_lab</code>	Optional character string specifying the y-axis label.
<code>split_lab</code>	Optional character string specifying the key label.
<code>levels</code>	Optional named list where each item name matches a factored column in <code>data</code> that needs either reordering of levels, renaming of levels, or both. Each item should be a list containing named elements <code>new_order</code> or <code>new_names</code> or both.

Details

While `within` and `between` are both optional, at least one column of `data` must be provided to either `within` or `between`. Any numeric or character variables in `data` that are specified as either `sid`, `within` or `between` will be converted to a factor with a warning. Fisher's Least Significant Difference is computed as $\sqrt{2} * qt(.975, DFd) * \sqrt{MSd/N}$, where N is taken as the mean N per group in cases of unbalanced designs.

Value

A printable/modifiable ggplot2 object.

Warning

The default error bars are Fisher's Least Significant Difference for the plotted effect, facilitating visual post-hoc multiple comparisons. Note however that in the context of mixed within-and-between-Ss designs, these bars can only be used for within-Ss comparisons.

Author(s)

Michael A. Lawrence <Mike.Lawrence@dal.ca>

See Also

[ezANOVA](#), [ezPerm](#), [ezStats](#)

Examples

```
#Read in the ANT data (see ?ANT).
data(ANT)

#Show summaries of the ANT data.
head(ANT)
str(ANT)
summary(ANT)

#Compute some useful statistics per cell.
cell_stats = ddply(
  .data = ANT
  , .variables = .( sid , group , cue , flanker )
  , .fun <- function(x){
    #Compute error rate as percent.
    error_rate = (1-mean(x$acc))*100
    #Compute mean RT (only accurate trials).
    mean_rt = mean(x$rt[x$acc==1])
    #Compute SD RT (only accurate trials).
    sd_rt = sd(x$rt[x$acc==1])
    return(c(error_rate=error_rate,mean_rt=mean_rt,sd_rt=sd_rt))
  }
)

#Run an ANOVA on the mean_rt data.
mean_rt_anova = ezANOVA(
  data = cell_stats
  , dv = .(mean_rt)
  , sid = .(sid)
  , within = .(cue,flanker)
  , between = .(group)
)

#Show the ANOVA & assumption tests.
```

```
print(mean_rt_anova)

#Plot the main effect of group.
group_plot = ezPlot(
  data = cell_stats
  , dv = .(mean_rt)
  , sid = .(sid)
  , between = .(group)
  , x = .(group)
  , do_lines = FALSE
  , x_lab = 'Group'
  , y_lab = 'RT (ms)'
)

#Show the plot.
print(group_plot)

#Plot the cue*flanker interaction.
cue_by_flanker_plot = ezPlot(
  data = cell_stats
  , dv = .(mean_rt)
  , sid = .(sid)
  , within = .(cue,flanker)
  , x = .(flanker)
  , split = .(cue)
  , x_lab = 'Flanker'
  , y_lab = 'RT (ms)'
  , split_lab = 'Cue'
)

#Show the plot.
print(cue_by_flanker_plot)

#Plot the group*cue*flanker interaction.
group_by_cue_by_flanker_plot = ezPlot(
  data = cell_stats
  , dv = .(mean_rt)
  , sid = .(sid)
  , within = .(cue,flanker)
  , between = .(group)
  , x = .(flanker)
  , split = .(cue)
  , col = .(group)
  , x_lab = 'Flanker'
  , y_lab = 'RT (ms)'
  , split_lab = 'Cue'
)

#Show the plot.
print(group_by_cue_by_flanker_plot)

#Re-plot the main effect of group, using the levels
##argument to re-arrange/rename levels of group
```

```

group_plot = ezPlot(
  data = cell_stats
  , dv = .(mean_rt)
  , sid = .(sid)
  , between = .(group)
  , x = .(group)
  , do_lines = FALSE
  , x_lab = 'Group'
  , y_lab = 'RT (ms)'
  , levels = list(
    group = list(
      new_order = c('Treatment', 'Control')
      , new_names = c('Treatment\nGroup', 'Control\nGroup')
    )
  )
)

#Show the plot.
print(group_plot)

```

 ezStats

Function to obtain descriptive statistics from a factorial experiment

Description

This function provides easy computation of descriptive statistics (between-Ss means, between-Ss SD, Fisher's Least Significant Difference) for data from factorial experiments, including purely within-Ss designs (a.k.a. "repeated measures"), purely between-Ss designs, and mixed within-and-between-Ss designs.

Usage

```

ezStats(
  data
  , dv
  , sid
  , within = NULL
  , between = NULL
)

```

Arguments

<code>data</code>	Data frame containing the data to be analyzed.
<code>dv</code>	.() object specifying the column in <code>data</code> that contains the dependent variable. Values in this column should be of the numeric class.
<code>sid</code>	.() object specifying the column in <code>data</code> that contains the variable specifying the case/Ss identifier. Values in this column will be converted to factor class if necessary.

<code>within</code>	Optional <code>.</code> (<code>.</code>) object specifying the column(s) in <code>data</code> that contains independent variables that are manipulated within-Ss. Values in this column will be converted to factor class if necessary.
<code>between</code>	Optional <code>.</code> (<code>.</code>) object specifying the column(s) in <code>data</code> that contains independent variables that are manipulated between-Ss. Values in this column will be converted to factor class if necessary.

Details

While `within` and `between` are both optional, at least one column of `data` must be provided to either `within` or `between`. Any numeric or character variables in `data` that are specified as either `sid`, `within` or `between` will be converted to a factor with a warning. Fisher's Least Significant Difference is computed as $\sqrt{2} * qt(.975, DFd) * \sqrt{MSd/N}$, where `N` is taken as the mean `N` per group in cases of unbalanced designs.

Value

A data frame containing the descriptive statistics for the requested effect. `N` = number of Ss per cell. `Mean` = between-Ss mean. `SD` = between-Ss SD. `FLSD` = Fisher's Least Significant Difference.

Warning

The descriptives include Fisher's Least Significant Difference for the requested effect. In the context of purely within-Ss or purely between-Ss this value may be used for post-hoc multiple comparisons. Note however that in the context of mixed within-and-between-Ss designs, this value can only be used for within-Ss comparisons.

Author(s)

Michael A. Lawrence (Mike.Lawrence@dal.ca)

See Also

[ezANOVA](#), [ezPerm](#), [ezPlot](#)

Examples

```
#Read in the ANT data (see ?ANT).
data(ANT)

#Show summaries of the ANT data.
head(ANT)
str(ANT)
summary(ANT)

#Compute some useful statistics per cell.
cell_stats = ddply(
  .data = ANT
  , .variables = .( sid , group , cue , flanker )
  , .fun <- function(x){
    #Compute error rate as percent.
```

```
        error_rate = (1-mean(x$acc))*100
        #Compute mean RT (only accurate trials).
        mean_rt = mean(x$rt[x$acc==1])
        #Compute SD RT (only accurate trials).
        sd_rt = sd(x$rt[x$acc==1])
        return(c(error_rate=error_rate,mean_rt=mean_rt,sd_rt=sd_rt))
    }
)

#Run an ANOVA on the mean_rt data.
mean_rt_anova = ezANOVA(
  data = cell_stats
  , dv = .(mean_rt)
  , sid = .(sid)
  , within = .(cue,flanker)
  , between = .(group)
)

#Show the ANOVA & assumption tests.
print(mean_rt_anova)

#Compute descriptives for the main effect of group.
group_descriptives = ezStats(
  data = cell_stats
  , dv = .(mean_rt)
  , sid = .(sid)
  , between = .(group)
)

#Show the descriptives.
print(group_descriptives)
```

Index

*Topic **datasets**

ANT, [2](#)

*Topic **package**

ez-package, [2](#)

Anova, [5](#)

ANT, [2](#)

aov, [5](#)

ez (*ez-package*), [2](#)

ez-package, [2](#)

ezANOVA, [2](#), [3](#), [3](#), [8](#), [11](#), [14](#)

ezPerm, [2](#), [3](#), [5](#), [7](#), [11](#), [14](#)

ezPlot, [2](#), [3](#), [5](#), [8](#), [9](#), [14](#)

ezStats, [2](#), [3](#), [5](#), [8](#), [11](#), [13](#)