# Package ‘prediction’

## April 19, 2017

**Type** Package  
**Title** Tidy, Type-Safe ‘prediction()’ Methods  
**Description** A one-function package containing ‘prediction()’, a type-safe alternative to ‘predict()’ that always returns a data frame. The package currently supports common model types (e.g., `"lm", "glm") from the 'stats' package, as well as numerous other model classes from other add-on packages. See the README or main package documentation page for a complete listing.

**License** MIT + file LICENSE  
**Version** 0.2.0  
**Date** 2017-04-19  
**URL** [https://github.com/leeper/prediction](https://github.com/leeper/prediction)  
**BugReports** [https://github.com/leeper/prediction/issues](https://github.com/leeper/prediction/issues)  
**Imports** utils, stats  
**Suggests** datasets, methods, testthat  
**Enhances** AER, betareg, brglm, crch, e1071, gam, gee, glmx, lme4, MASS, mclgit, mlogit, mnlogit, MNP, nlme, nnet, ordinal, plm, pscl, quantreg, sampleSelection, survey (>= 3.31-5), survival, VGAM

**RoxygenNote** 6.0.1  
**NeedsCompilation** no  
**Author** Thomas J. Leeper [aut, cre], Carl Ganz [ctb]  
**Maintainer** Thomas J. Leeper <thosjleeper@gmail.com>  
**Repository** CRAN  
**Date/Publication** 2017-04-19 10:30:58 UTC

## R topics documented:

- prediction-package .......................................................... 2  
- build_datalist ............................................................... 7  
- find_data ................................................................. 8  
- mean_or_mode ............................................................. 10  
- seq_range ............................................................... 11

---

1
Extract Predictions from a Model Object

Description

Extract predicted values via `predict` from a model object, conditional on data, and return a data frame.

Usage

```
prediction(model, ...)
```

```
## Default S3 method:
prediction(model, data = find_data(model, parent.frame()),
  at = NULL, type = "response", ...)
```

```
## S3 method for class 'Arima'
prediction(model, ...)
```

```
## S3 method for class 'ar'
prediction(model, data, at = NULL, ...)
```

```
## S3 method for class 'arima0'
prediction(model, data, at = NULL, ...)
```

```
## S3 method for class 'betareg'
prediction(model, data = find_data(model, parent.frame()),
  at = NULL, type = c("response", "link", "precision", "variance",
  "quantile"), ...)
```

```
## S3 method for class 'clm'
prediction(model, data = find_data(model, parent.frame()),
  at = NULL, type = NULL, category, ...)
```

```
## S3 method for class 'coxph'
prediction(model, data = find_data(model, parent.frame()),
  at = NULL, type = c("risk", "expected", "lp"), ...)
```

```
## S3 method for class 'crch'
prediction(model, data = find_data(model), at = NULL,
  type = c("response", "location", "scale", "quantile"), ...)
```

```
## S3 method for class 'gam'
prediction(model, data = find_data(model, parent.frame()),
  at = NULL, type = c("response", "link", "terms"), ...)
```
## S3 method for class 'gee'
prediction(model, ...)

## S3 method for class 'glm'
prediction(model, data = find_data(model, parent.frame()),
          at = NULL, type = c("response", "link"), ...)

## S3 method for class 'glmx'
prediction(model, data = find_data(model, parent.frame()),
          at = NULL, type = c("response", "link", "scale"), ...)

## S3 method for class 'gls'
prediction(model, data = find_data(model), at = NULL, ...)

## S3 method for class 'hetglm'
prediction(model, data = find_data(model, parent.frame()),
          at = NULL, type = c("response", "link", "scale"), ...)

## S3 method for class 'hurdle'
prediction(model, data = find_data(model, parent.frame()),
          at = NULL, type = c("response", "count", "prob", "zero"), ...)

## S3 method for class 'hoxlr'
prediction(model, data = find_data(model), at = NULL,
          type = c("class", "probability", "cumprob", "location", "scale"), ...)

## S3 method for class 'ivreg'
prediction(model, data = find_data(model, parent.frame()),
          at = NULL, ...)

## S3 method for class 'lm'
prediction(model, data = find_data(model, parent.frame()),
          at = NULL, type = "response", ...)

## S3 method for class 'lme'
prediction(model, data = find_data(model), at = NULL, ...)

## S3 method for class 'loess'
prediction(model, data = find_data(model, parent.frame()),
          at = NULL, type = "response", ...)

## S3 method for class 'lqs'
prediction(model, data = find_data(model), at = NULL, ...)

## S3 method for class 'mclogit'
prediction(model, data = find_data(model), at = NULL, ...)

## S3 method for class 'mca'
prediction(model, data = find_data(model), at = NULL, ...)
prediction(model, data = find_data(model), parent.frame()),
at = NULL, type = c("response", "link"), ...)

## S3 method for class 'merMod'
prediction(model, data = find_data(model), at = NULL,
type = c("response", "link"), ...)

## S3 method for class 'mnlogit'
prediction(model, data = find_data(model), parent.frame()),
at = NULL, category, ...)

## S3 method for class 'mnp'
prediction(model, data = find_data(model)),
at = NULL, type = NULL, category, ...

## S3 method for class 'multinom'
prediction(model, data = find_data(model), parent.frame()),
at = NULL, type = NULL, category, ...

## S3 method for class 'nls'
prediction(model, data = find_data(model), parent.frame()),
at = NULL, ...)

## S3 method for class 'nnet'
prediction(model, data = find_data(model), parent.frame()),
at = NULL, type = NULL, category, ...

## S3 method for class 'plm'
prediction(model, data = find_data(model), parent.frame()),
at = NULL, ...

## S3 method for class 'polr'
prediction(model, data = find_data(model), parent.frame()),
at = NULL, type = NULL, category, ...

## S3 method for class 'ppr'
prediction(model, data = find_data(model), parent.frame()),
at = NULL, ...

## S3 method for class 'princomp'
prediction(model, data = find_data(model), parent.frame()),
at = NULL, ...

## S3 method for class 'rlm'
prediction(model, data = find_data(model), parent.frame()),
at = NULL, type = "response", ...)
prediction-package

    prediction(model, data = find_data(model, parent.frame()),
              at = NULL, ...)

    ## S3 method for class 'selection'
    prediction(model, data = find_data(model, parent.frame()),
              at = NULL, type = "response", ...)  
    
    ## S3 method for class 'survreg'
    prediction(model, data = find_data(model, parent.frame()),
              at = NULL, type = c("response", "lp", "quantile", "uquantile"), ...)  
    
    ## S3 method for class 'svm'
    prediction(model, data = NULL, at = NULL, category, ...)

    ## S3 method for class 'svyglm'
    prediction(model, data = find_data(model, parent.frame()),
              at = NULL, type = c("response", "link"), ...)

    ## S3 method for class 'zeroinfl'
    prediction(model, data = find_data(model, parent.frame()),
              at = NULL, type = c("response", "count", "prob", "zero"), ...)

Arguments

model A model object, perhaps returned by lm or glm.

... Additional arguments passed to predict methods.

data A data.frame over which to calculate marginal effects. If missing, find_data is used to specify the data frame.

at A list of one or more named vectors, specifically values at which to calculate the predictions. These are used to modify the value of data (see build_datalist for details on use).

type A character string indicating the type of marginal effects to estimate. Mostly relevant for non-linear models, where the reasonable options are "response" (the default) or "link" (i.e., on the scale of the linear predictor in a GLM). For models of class "polr" (from polr), possible values are "class" or "probs"; both are returned.

category For multi-level or multi-category outcome models (e.g., ordered probit, multinomial logit, etc.), a value specifying which of the outcome levels should be used for the "fitted" column. If missing, some default is chosen automatically.

Details

This function is simply a wrapper around predict that returns a data frame containing the value of data and the predicted values with respect to all variables specified in data.

Methods are currently implemented for the following object classes:

* "lm", see lm
• “glm”, see glm, glm.nb, glmx, hetglm, brglm
• “ar”, see ar
• “Arima”, see arima
• “arima0”, see arima0
• “betareg”, see betareg
• “clm”, see clm
• “coxph”, see coxph
• “crch”, see crch
• “gam”, see gam
• “gee”, see gee
• “gls”, see gls
• “hurdle”, see hurdle
• “hxlr”, see hxlr
• “ivreg”, see ivreg
• “lda”, see lda
• “lme”, see lme
• “loess”, see loess
• “lqs”, see lqs
• “mca”, see mca
• “mclogit”, see mclogit
• “merMod”, see lmer, glmer
• “mnlogit”, see mnlogit
• “mmp”, see mnp
• “naiveBayes”, see naiveBayes
• “nlme”, see nlme
• “nls”, see nls
• “nnet”, see nnet
• “plm”, see plm
• “polr”, see polr
• “ppr”, see ppr
• “princomp”, see princomp
• “qda”, see qda
• “rlm”, see rlm
• “rq”, see rq
• “selection”, see selection
• “survreg”, see survreg
• “svm”, see svm
• “svyglm”, see svyglm
• “zeroinfl”, see zeroinfl
Value

A data frame with class "prediction" that has a number of rows equal to number of rows in data, or a multiple thereof, if !is.null(at). The return value contains data (possibly modified by at using build_datalist), plus a column containing fitted/predicted values ("fitted") and a column containing the standard errors thereof ("se.fitted"). Additional columns may be reported depending on the object class.

See Also

find_data, build_datalist, mean_or_mode, seq_range

Examples

require("datasets")
x <- lm(Petal.Width ~ Sepal.Length * Sepal.Width * Species, data = iris)
# prediction for every case
prediction(x)

# prediction for first case
prediction(x, iris[1,])

# basic use of 'at' argument
prediction(x, at = list(Species = c("setosa", "virginica")))

# prediction at means/modes of input variables
prediction(x, at = lapply(iris, mean_or_mode))

# prediction with multi-category outcome
## Not run:
library("mlogit")
data("Fishing", package = "mlogit")
Fish <- mlogit.data(Fishing, varying = c(2:9), shape = "wide", choice = "mode")
mod <- mlogit(mode ~ price + catch, data = Fish)
prediction(mod)
prediction(mod, category = 3)

## End(Not run)
find_data

Arguments

  data  A data.frame containing the original data.
  at    A list of one or more named vectors of values, which will be used to specify
        values of variables in data. See examples.
  ...   Ignored.

Value

  A list of data.frames.

Author(s)

  Thomas J. Leeper

See Also

  find_data, mean_or_mode, seq_range

Examples

  # basic examples
  require("datasets")
  build_datalist(head(mtcars), at = list(cyl = c(4, 6)))
  str(build_datalist(head(mtcars, at = list(cyl = c(4,6), wt = c(1,2,3)))))

find_data Extract data from a model object

Description

  Attempt to reconstruct the data used to create a model object

Usage

  find_data(model, ...)

  ## Default S3 method:
  find_data(model, env = parent.frame(), ...)

  ## S3 method for class 'data.frame'
  find_data(model, ...)

  ## S3 method for class 'lm'
  find_data(model, env = parent.frame(), ...)
### find_data

```r
find_data(model, env = parent.frame(), ...)
```

### Arguments

- `model` The model object.
- `...` Additional arguments passed to methods.
- `env` An environment in which to look for the data argument to the modelling call.

### Details

This is a convenience function and, as such, carries no guarantees. To behave well, it typically requires that a model object be specified using a formula interface and an explicit `data` argument. Models that can be specified using variables from the `.GlobalEnv` or with a non-formula interface (e.g., a matrix of data) will tend to generate errors. `find_data` is an S3 generic so it is possible to expand it with new methods.

### Value

A data frame containing the original data used in a modelling call, modified according to the original model’s `subset` and `na.action` arguments, if appropriate.

### See Also

- `prediction`, `build_datalist`, `mean_or_mode`, `seq_range`
Examples

```r
require("datasets")
x <- lm(mpg ~ cyl * hp + wt, data = head(mtcars))
find_data(x)
```

mean_or_mode

Class-dependent variable aggregation

Description

Summarize a vector/variable into a single number, either a mean (median) for numeric vectors or the mode for categorical (character, factor, ordered, or logical) vectors. Useful for aggregation.

Usage

```r
mean_or_mode(x)
```

Arguments

- `x` A vector.

Value

A numeric or factor vector of length 1.
seq_range

See Also

prediction, build_datalist, seq_range

Examples

```r
require("datasets")
# mean for numerics
mean_or_mode(iris)
mean_or_mode(iris["Sepal.Length"])
mean_or_mode(iris["Species"])

# median for numerics
median_or_mode(iris)
```

seq_range  Create a sequence over the range of a vector

Description

Define a sequence of evenly spaced values from the minimum to the maximum of a vector

Usage

```r
seq_range(x, n = 2)
```

Arguments

- `x`  
  A numeric vector

- `n`  
  An integer specifying the length of sequence (i.e., number of points across the range of `x`)

Value

A vector of length `n`.

See Also

mean_or_mode, build_datalist

Examples

```r
identical(range(1:5), seq_range(1:5, n = 2))
seq_range(1:5, n = 3)
```
Index

*Topic data
  builddatalist, 7

*Topic manip
  builddatalist, 7

*Topic models
  predictionpackage, 2

ar, 6
arima, 6
arima0, 6

betareg, 6
brglm, 6
builddatalist, 5, 7, 9, 11

clm, 6
coxph, 6
crch, 6

find_data, 5, 7, 8, 8

gam, 6
glm, 5, 6
glm.nb, 6
glmer, 6
glmx, 6
gls, 6

hetglm, 6
hurdle, 6
hxlr, 6

ivreg, 6

lda, 6
lm, 5
lmer, 6
loess, 6
lqs, 6

mca, 6
mclogit, 6
mean_or_mode, 7–9, 10, 11
median_or_mode(mean_or_mode), 10
mnlogit, 6
mnp, 6

naiveBayes, 6
nlme, 6
nls, 6
nnet, 6

plm, 6
polr, 5, 6
ppr, 6
predict, 2, 5
prediction, 9, 11
prediction(predictionpackage), 2
predictionpackage, 2
prediction.ar(predictionpackage), 2
prediction.Arima(predictionpackage), 2
prediction.arima0(predictionpackage), 2
prediction.betareg(predictionpackage), 2
prediction.clm(predictionpackage), 2
prediction.coxph(predictionpackage), 2
prediction.crch(predictionpackage), 2
prediction.default(predictionpackage), 2
prediction.gam(predictionpackage), 2
prediction.gee(predictionpackage), 2
prediction.glm(predictionpackage), 2
prediction.glmx(predictionpackage), 2
prediction.gls(predictionpackage), 2
prediction.hetglm(predictionpackage), 2
prediction.hurdle(predictionpackage), 2
prediction.hxlr(predictionpackage), 2

12
INDEX

prediction.ivreg (prediction-package), 2
prediction.lm (prediction-package), 2
prediction.lme (prediction-package), 2
prediction.loess (prediction-package), 2
prediction.lqs (prediction-package), 2
prediction.mca (prediction-package), 2
prediction.mclogit
  (prediction-package), 2
prediction.merMod (prediction-package), 2
prediction.mnlogit
  (prediction-package), 2
prediction.mnp (prediction-package), 2
prediction.multinom
  (prediction-package), 2
prediction.nls (prediction-package), 2
prediction.nnet (prediction-package), 2
prediction.plm (prediction-package), 2
prediction.polr (prediction-package), 2
prediction.ppr (prediction-package), 2
prediction.princomp
  (prediction-package), 2
prediction.rlm (prediction-package), 2
prediction.rq (prediction-package), 2
prediction.selection
  (prediction-package), 2
prediction.survreg
  (prediction-package), 2
prediction.svm (prediction-package), 2
prediction.svyglm (prediction-package), 2
prediction.zeroinfl
  (prediction-package), 2
princomp, 6
qda, 6
rlm, 6
rq, 6
selection, 6
seq_range, 7–9, 11, 11
survreg, 6
svm, 6
svyglm, 6
zeroinfl, 6