Package: ham (via r-universe)

November 1, 2026

Title Healthcare Analysis Methods

Version 1.4.0.9000

Description Conducts analyses for healthcare program evaluations or intervention studies. Calculates regression analyses for standard ordinary least squares (OLS or linear) or logistic models. Performs regression models used for causal modeling such as differences-in-differences (DID) and interrupted time series (ITS) models. Provides limited interpretations of model results and a ranking of variable importance in models. Performs propensity score models, top-coding of model outcome variables, and can return new data with the newly formed variables. Conducts Bayesian analysis summaries and graphs, decision curve analysis, and produces some Shewhart control charts. Also performs Cronbach's alpha for various scale items (e.g., survey questions). See Github URL for examples in the README file. For more details on the statistical methods, see Allen & Yen (1979, ISBN:0-8185-0283-5), Angrist & Pischke (2009, ISBN:9780691120355), Cohen (1988, ISBN:0-8058-0283-5), Gebski (2012) <doi:10.1017/S0950268812000179>, Gelman & Goodrich (2019) <doi:10.1080/00031305.2018.1549100>, Gelman & Hill (2007, ISBN:978-0-521-68689-1), Harrell (2015, ISBN:978-3-319-19424-0), Imbens & Rubin (2015, ISBN:978-0-521-88588-1), Kline (1999, ISBN:9780415211581), Kruschke (2014, ISBN:9780124058880), Linden (2015) <doi:10.1177/1536867X1501500208>, Merlo (2006) <doi:10.1136/jech.2004.029454>, Muthen & Satorra (1995) <doi:10.2307/271070>, Pitman (1993, ISBN:978-0-387-97974-8), Rabe-Hesketh & Skrondal (2008, ISBN:978-1-59718-040-5), Rosenbaum (2010, ISBN:978-1-4419-1212-1), Ryan (2011, ISBN:978-0-470-59074-4), and Vickers & Elkin (2006) <doi:10.1177/0272989X06295361>.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 2.10)

LazyData true

URL https://github.com/szuniga07/ham

BugReports https://github.com/szuniga07/ham/issues

Imports methods

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Repository https://szuniga07.r-universe.dev

Date/Publication 2026-11-01 21:41:07 UTC

RemoteUrl https://github.com/szuniga07/ham

RemoteRef HEAD

RemoteSha 412cf92b02876c0c1f5211529063af21bcfbea65

Contents

alpha . . . 3
assess . . . 4
Bayes . . . 8
cas . . . 12
co2mcmc . . . 12
co2multi . . . 13
compapt . . . 14
control . . . 15
covshort . . . 16
decide . . . 17
group . . . 18
hosp1 . . . 20
hosprog . . . 21
importance . . . 22
infections . . . 23
interpret . . . 23
itsEffect . . . 24
losmcmc . . . 26
losvary . . . 26
NHSN . . . 27
plot.assess . . . 28
plot.Bayes . . . 32
plot.control . . . 41
plot.decide . . . 44
plot.group . . . 46
plot.importance . . . 49
plot.review . . . 51

print.alpha 54
print.interpret 55
print.review 56
review 56
unemployment 58

Index **59**

`alpha` *Calculates Cronbach's alpha on scale items*

Description

Performs Cronbach's alpha of specified items from a data frame. Cronbach's Alpha is a formula for estimating the internal consistency reliability of a measurement instrument such as survey items (see Allen & Yang, 1979; Kline, 1999). Survey items can have 2 or more categories such as 5-point scales and contain 2 or more items.

Usage

```
alpha(items, data)
```

Arguments

`items`	Vector of item names that form a scale (e.g., 5-point Likert scales)
`data`	Data frame object.

Value

A list object with Cronbach's alpha summary statistics.

References

Allen, M. J., & Yen, W. M. (1979). Introduction to Measurement Theory. Brooks/Cole. ISBN: 0-8185-0283-5. Kline, Paul (1999). Handbook of Psychological Testing (2nd ed). Routledge, New York. ISBN: 9780415211581.

See Also

`print.alpha()`, `interpret()` for a formatted print and interpretation of the 'alpha' class object.

Examples

```
alpha(items=c("i1","i2","i3","i4","i5"), data=cas)

# remove i1 as suggested in the previous example, returns higher alpha
alpha(items=c("i2","i3","i4","i5"), data=cas)
```

`assess` *Assess models with regression*

Description

Fit ordinary least squares (OLS), logistic, and Poisson models. And fit models for causal inference such as differences-in-differences and interrupted time series. Run these models to evaluate program performance or test intervention effects (e.g., healthcare programs). Options are available for top coding the outcome variable as well as propensity scores. New data can optionally be returned that has these additional variables and constructed variables that are used for DID and ITS models.

Usage

```
assess(
  formula = NULL,
  data = NULL,
  regression = "none",
  did = "none",
  its = "none",
  intervention = NULL,
  int.time = NULL,
  treatment = NULL,
  interrupt = NULL,
  subset = NULL,
  stagger = NULL,
  topcode = NULL,
  propensity = NULL,
  trim = NULL,
  weights = NULL,
  offset = NULL,
  newdata = FALSE,
  link = NULL,
  model = NULL
)
```

Arguments

`formula`	a formula object. Use 'Y ~ .' in DID and ITS models to only specify the constructed model variables (e.g., right side of the DID model: Y ~ Post.All + Int.Var + DID). If regression=ols or regression=logistic, 'Y ~ .' will use all variables in the data.frame as is standard in formulas.
`data`	a data.frame in which to interpret the variables named in the formula.
`regression`	Select a regression method for standard regression models (i.e., neither DID nor ITS). Options are regression="ols" (ordinary least squares AKA linear), regression="logistic", or regression='poisson'. Default is regression="none" for no standard regression model.

`did` option for Differences-in-Differences (DID) regression. Select did="two" for models with only 2 time points (e.g., pre/post-test). Select did="many" for >= 3 time points (e.g., monthly time points in 12 months of data). Default is did="none" for no DID.

`its` option for Interrupted Time Series (ITS) regression. Select its="one" for one group (e.g., intervention only). Select its="two" for two groups (intervention and control). Default is did="none" for no ITS.

`intervention` optional intervention variable name selected for DID, ITS, and propensity score models that indicate which cases are in the intervention or not when propensity is a character class (i.e., intervention can be NULL when propensity is a formula class for a non-DID or non-ITS model).

`int.time` optional intervention time variable name selected for DID or ITS models. This indicates the duration of time relative to when the intervention started.

`treatment` optional treatment start period variable name selected for DID models. Select 1 value from 'int.time' to indicate the start of the intervention.

`interrupt` optional interruption (or intervention) period(s) variable name selected for ITS models. Select 1 or more values from 'int.time' to indicate the start and/or key intervention periods. There needs to be at least 2 time points per period, at least 3 is better. For example, interrupt= c(3, 5, 7) will suffice, especially if you want to isolate certain periods but interrupt= c(3, 6, 9) may provide more useful information.

`subset` an expression defining a subset of the observations to use in the regression model. The default is NULL, thereby using all observations. Specify, for example, data$hospital == "NY" or c(1:100,200:300) respectively to use just those observations. This is helpful when doing a submodel for DID or ITS after identifying similar groups. DID and ITS models could be improved by limiting the choice of control groups to only those with similar values on the intervention indicator and baseline trend variable (e.g., 'ITS.Time' and 'ITS.Int') with p-values >= 0.10.

`stagger` optional list to indicate staggered entry into the intervention or treatment group. Relevant model variables are re-coded to appropriate values and can be used for a form of 'stacked' DID or ITS. If a group of cases joins X months after the primary sample, model variables are adjusted X months. This three element list named: 'a' = a character vector for the name of the grouping column; 'b' = specific categories or levels that indicate which cases have a staggered entry; and 'c' = the time point values at staggered entry. Both 'b' and 'c' must have identical lengths. For ITS models, the staggered entry time must be: interrupt 1 < stagger time < interrupt 2. For example, a WHO health policy may have started in the 3rd year of the study period in NY and Toronto but Chicago and LA joined 6 and 12 months later, therefore stagger= list(a= 'city', b=c('Chicago', 'LA')), c=(30, 36) while interrupt= 25. Default is NULL.

`topcode` optional value selected to top code Y (or left-hand side) of the formula. Analyses will be performed using the new top coded variable.

`propensity` optional character vector of variable names or formula class to perform a propensity score model. This requires the 'intervention' option to be selected only when using a character vector. All models will include 'pscore' (propensity

score) in the analysis as a covariate adjustment using the propensity score unless one of the weights options 'ipw' (Inverse Probability Weights for the ATE, Average Treatment Effect, estimator), 'nipw' (Normalized Inverse Probability Weights by Hajek estimator), or 'att' (Average Treatment Effect on the Treated) is selected for the analysis.

`trim` an optional two-element numeric vector that sets limits between 0 and 1 for the propensity score. If NULL, the default values of >= 0 and <= 1 are used (i.e., c(0,1)).

`weights` an optional 1-element character vector of the data frame column name or character vector of either 'ipw', 'nipw', or 'att' for Inverse Probability of Treatment Weighting Using the propensity score (see 'propensity' above) to be used in the fitting process. Or 'nipw' (Normalized Inverse Probability Weights), or 'att' (Average Treatment Effect on the Treated). Should be NULL or a character vector. If non-NULL, weighting is used with weights; otherwise standard regression is used. Weights calculated as ipw: treatment= 1/pscore, control= 1/(1-pscore); nipw: treatment= ipw / sum of treatment ipw, control= ipw / sum of control ipw; att: treatment= 1; control= pscore/(1-pscore).

`offset` an optional 1-element character vector of the data frame column name to be used with the predictor during fitting. An offset term can be included in the formula instead. offset is often used for rates in Poisson models. Can only use the data frame column name, cannot transform the column directly in this argument. If transformations are needed, instead either do so in the formula call (e.g., 'y ~ x1 + offset(log(x2))') or transform the data directly in the data frame.

`newdata` optional logical value that indicates if you want the new data returned. newdata=TRUE will return the data with any new columns created from the DID, ITS, propensity score, or top coding. The default is newdata=FALSE. No new data will be returned if none was created.

`link` a description of the link function to be used in the glm model. This can be a character string naming a link. The 'binomial' family link options are the 'logit' (default), 'probit', 'cauchit', (corresponding to 'logistic', 'normal' and 'Cauchy' CDFs respectively) 'log' and 'cloglog' (complementary log-log); and the 'poisson' family links 'log' (default), 'identity', and 'sqrt'.

`model` a non-assess class regression model that will be converted into an assess class object (e.g., only Cox PH models from the survival package are supported as of yet).

Value

a list of results from selected regression models. Will return new data if selected. And returns relevant model information such as variable names, type of analysis, formula, study information, and summary of ITS effects if analyzed.

References

Angrist, J. D., & Pischke, J. S. (2009). Mostly Harmless Econometrics: An Empiricist's Companion. Princeton University Press. ISBN: 9780691120355.

Gebski, V., et al. (2012). Modelling Interrupted Time Series to Evaluate Prevention and Control of Infection in Healthcare. Epidemiology & Infections, 140, 2131–2141. https://doi.org/10.1017/S0950268812000179

Gelman, A. & Hill, J. (2007). Data Analysis Using Regression and Multilevel/Hierarchical Models. Cambridge University Press. ISBN: 978-0-521-68689-1.

Imbens, G. & Rubin, D. (2015). Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction. Cambridge University Press. ISBN: 978-0-521-88588-1.

Linden, A. (2015). Conducting Interrupted Time-series Analysis for Single- and Multiple-group Comparisons. The Stata Journal, 15, 2, 480-500. https://doi.org/10.1177/1536867X1501500208

Rosenbaum, P. (2010). Design of Observational Studies, Second Edition. New York: Springer. ISBN: 978-1-4419-1212-1.

See Also

`plot.assess()`, `interpret()` for a plot and interpretation of the 'assess' class object.

Examples

```
# ordinary least squares R^2
summary(assess(hp ~ mpg+wt, data=mtcars, regression="ols")$model)

# logistic
summary(assess(formula=vs~mpg+wt+hp, data=mtcars, regression="logistic")$model)

# OLS with a propensity score
summary(assess(formula=los ~ month+program, data=hosprog, intervention = "program",
regression="ols", propensity=c("female","age","risk"))$model)

# OLS: top coding los at 8.27 and propensity score means (top.los and pscore)
summary(assess(formula=los ~ month+program, data=hosprog, intervention = "program",
regression="ols", topcode=8.27, propensity=c("female","age","risk"),
newdata=TRUE)$newdata[, c("los", "top.los", "pscore")])

# differences-in-differences model: using 2 time periods, pre- and post-intervention
summary(assess(formula=los ~ ., data=hosprog, intervention = "program",
int.time="month", treatment = 5, did="two")$DID)

# DID model: using time points
summary(assess(formula=los ~ ., data=hosprog, intervention = "program",
int.time="month", treatment = 5, did="many")$DID)

#interrupted time series model: two groups and 1 interruption (interrupt= 5)
summary(assess(formula=los ~ ., data=hosprog, intervention = "program",
int.time="month", its="two", interrupt = 5)$ITS)

#interrupted time series model: two groups and 2 interruptions (interrupt= c(5,9))
summary(assess(formula=los ~ ., data=hosprog, intervention = "program",
int.time="month", its="two", interrupt = c(5,9))$ITS)
```

Bayes *Summarize Bayesian Markov Chain Monte Carlo (MCMC) object*

Description

Convert a list of Bayesian analysis chains (e.g., coda package mcmc.list objects) into a data frame for analysis and creating plots. For example, models from JAGS or Stan that were converted into coda class objects can be used to create the data frames. Calculates a set of descriptive statistics that summarize MCMC parameters. And values can be calculated for use in descriptive graphs such as values associated with specific percentiles and vice-versa to help set targets, MCMC diagnostics, summaries on hierarchical or multilevel models (up to 3 levels), and the R2 for Bayesian regression models with metric level predictors. It is advisable to restart your R session after using the coda package and prior to using ham.

Usage

```
Bayes(
  x,
  y = "mcmc",
  parameter = NULL,
  mass = 0.95,
  compare = NULL,
  rope = NULL,
  newdata = FALSE,
  type = NULL,
  center = "mode",
  data = NULL,
  dv = NULL,
  iv = NULL,
  expand = NULL,
  targets = NULL
)
```

Arguments

x	list object of multiple MCMC chains (e.g., matrix class list elements or coda mcmc.list).
y	character vector for the type of analysis or output to perform. Select 'post', 'multi', 'target', 'r2', or 'mcmc' for a posterior summary, multilevel/hierarchical model summary (up to 3 levels), target summary, Gelman R-squared statistic, MCMC diagnostics, or list object of MCMC chains converted into a data frame. For MCMC diagnostics, select y='Dx' to get the auto-correlation factor, effective sample size, Monte Carlo standard error, and Gelman-Rubin shrink factor. Default is generic 'mcmc'(no analysis, just MCMC creation).
parameter	single or multiple element character vector name of parameter(s) in MCMC chains to produce summary statistics. When y='target', use the generally 2 to 3

parameters that represent the distribution parameters (e.g., parameter= c('mean', 'sd')). When y='r2', use the regression parameters in order, ending with the residual or level-1 variance (e.g., parameter= c('intercept', 'beta1', 'beta2', 'standard_deviation')). For MCMC diagnostics (i.e., y='Dx'), select only 1 parameter at a time. Default is NULL.

`mass` numeric vector that specifies the credible mass used in the Highest Density Interval (HDI). Default is 0.95.

`compare` numeric vector with one comparison value to determine how much of the distribution is above or below the comparison value. Default is NULL.

`rope` numeric vector with two values that define the Region of Practical Equivalence (ROPE). Test hypotheses by setting low and high values to determine if the Highest Density Interval (HDI) is within or outside of the ROPE. Parameter value declared not credible if the entire ROPE lies outside the HDI of the parameter's posterior (i.e., we reject the null hypothesis). For example, the ROPE of a coin is set to 0.45 to 0.55 but the posterior 95% HDI is 0.61 - 0.69 so we reject the null hypothesis value of 0.50. We can accept the null hypothesis if the entire 95% HDI falls with the ROPE. Default is NULL.

`newdata` optional logical vector that indicates if you want the new MCMC data returned. When newdata=TRUE, it will return the list object of MCMC chains, converted into a data frame. This data is used for analysis and all plots. Please select newdata=TRUE to produce any graphs but not needed when y='multi'. The default is newdata=FALSE.

`type` character vector of length == 1 that indicates the likelihood function used in the model when y='multi' or y='target'. Select 'n', 'ln', 'w', 'g', 't', 'p', 'b', 'bern', and 'bin' for these respective options in Bayesian estimation (multilevel): 'Normal', 'Log-normal', 'Weibull', 'Gamma', 't', 'Poisson', 'Beta', 'Bernoulli', or 'Binomial'. Default is NULL.

`center` character vector that selects the type of central tendency to use when reporting parameter values when y='post', y='target', or y='r2'. Choices include: 'mean', 'median', and 'mode' when y='post', or 'mean' and 'median' when y='r2'. Default is 'mode' when y='post' or 'target' and 'median' when y='r2'.

`data` object name for the observed data when y='multi' or y='r2'. Default is NULL.

`dv` character vector of length == 1 for the dependent variable name in the observed data frame when y='multi'. Default is NULL.

`iv` character vector of length >= 1 for the independent variable name(s) in the observed data frame when y='multi' or y='r2'. When y='multi', enter the lower to higher level clustering or group names (e.g, for health data, iv=c("patient", "hospital"). When type='taov', enter the name of the test group variable. When y='r2', enter the observed data variable names for the hierarchical or multilevel groups. Default is NULL.

`expand` a character vector of length == 1 indicating the variable name to expand aggregated data into non-aggregated data frames when y='multi'. This variable is the denominator that can be used to calculate a rate in the formula numerator/denominator. For example, when the 'numerator' column equals 4 and the 'denominator' column equals 10, then this single row of data is expanded to

10 rows with four values of 1 and six values of 0 when expand='denominator'. Default is NULL.

`targets` allows probabilities and percentiles to be calculated from a list of one, two, or three named elements ('p', 'y', 'e') with numeric values that represent quantile or percentile values between 0 and 1 (p) in the distribution to return associated outcome values (i.e., inverse cumulative distribution function), specific outcome values (y) to retrieve associated probabilities less than Y (i.e., cumulative distribution function), and/or a 2 element list (named 'a' and 'b') of low and high proportions (no particular order) that Cohen's h effect sizes ('e') for proportions are calculated from it (i.e., not Bayesian analysis) to give us an idea of how large differences in targets are (e.g., target 0.50 vs target 0.40 = Cohen's h of 0.201 = 'small effect'–0.2, 0.5, 0.8 as thresholds for small, medium, and large effect sizes). For example, a distribution of harmful hospital readmission rates has an estimated median value of 0.25. Staff are considering 2 types of targets, percentiles (p) of key interest and specific outcome rates (y). They want to know the readmission rate that is at the 40th percentile for a reduced readmission rate (below what is 'average' at the 50th percentile) and the probability greater than a readmission rate of 0.20. They get this information by entering targets=list(p=0.40, y=0.20, e=list(0.50, 0.40)); considering prob(y) from the returned results gives them an idea about the effort needed to meet this target of a reduced readmission rate. And as they are searching for a shift in targets as a 'small effect', but substantial, the Cohen's h effect size of 0.20 indicates that this is met when comparing targets of 0.50 vs 0.40. For list elements 'p' and 'y, selecting 2 values will result in the intervals between the values in the form of the object 'High.Low.Interval'. When there are more than 2 listed values, the interval is between the lowest and highest values. Select type= one of these options: 'n', 'ln', 'w', 'g', 't', 'p', 'b', 'bern', 'bin'. Select parameter= the appropriate center, spread, and possible 3rd shape distribution parameter (e.g., parameter=c('mean', 'sd')). And the options to select center= 'mean', 'median', or 'mode'. When type='bin', the returned 'Mean' value is calculated assuming that the model has an underlying likelihood function using the mode instead of the mean for the beta distribution. If not, the returned 'Mean' will be incorrect. When type='p', linear interpolation is used to calculate returned probabilities and Y values at specific percentiles because ppois() and qpois() work for discrete integers in the Poisson distribution. Default is NULL.

Value

data frame of summary statistics for the MCMC parameter's distribution and/or MCMC data frame. Statistics include highest density interval, effective sample size, proportion of distribution within and outside of a ROPE, distribution compared with a set value, and the parameter's mean, median, and mode. And distribution summaries for multilevel models, target summaries, regression model R2, and MCMC diagnostics.

References

Cohen, J. (1988). Statistical Power Analysis for the Behavioral Sciences, Second Edition. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers. ISBN 0-8058-0283-5

Gelman, A., Goodrich, B., Gabry, J., & Vehtari, A. (2019). R-squared for Bayesian Regression Models. The American Statistician, 73, 3, 307–309. https://doi.org/10.1080/00031305.2018.1549100

Kruschke, J. (2014). Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan, Second Edition. New York: Academic Press. ISBN: 9780124058880

Pitman, J. (1993). Probability. Springer-Verlag. ISBN: 978-0-387-97974-8

See Also

plot.Bayes(), interpret() for a plot and interpretation of the 'Bayes' class object.

Examples

```
# Posterior estimates of length of stay (LOS) #
# What is the 'average' LOS and are we beyond our goal of it being <= 3 days?
blos1 <- Bayes(losmcmc, y="post", parameter="muOfY", newdata=TRUE,
compare=4.5, rope=c(1,3))   #Where are we in relation to 4.5 days?
print(blos1$Posterior.Summary) #looks like LOS is substantially higher than 3 days

# Multilevel or hierarchical model summaries #
# Code below does not run, no 'mcmc_sample' object
# bmulti0 <- Bayes(x=mcmc_sample, parameter=c("theta", "omega","omegaO"),
# y="multi", type="bern", data=mydf, dv="upbin", iv= c("Plant", "Group"))

# Targets for length of stay (LOS) #
# Our administrators ask how far are we from our goals, they ask about targets
# in increments of 5 points of probability or specific days. We answer both.
btarget1 <- Bayes(x=losmcmc, y="target", type="n", parameter=c("muOfY","sigmaOfY"),
newdata=TRUE, target=list(p=c(.35,.4,.45, .5, .55),  y=c(3,4),
e= list(a=c(.35,.4,.45), b= c(.5, .5, .5)) )) # 'newdata' for plots
print(btarget1$Target$Est.Quantile.P)  # 5-point increments
print(btarget1$Target$Est.Prob.GT.Y)   # specific day values, 4 days is plausible

# Gelamn's R^2 #
# the regression model using Base R data, CO2: update ~ conc
bR2 <- Bayes(x=co2mcmc, y='r2', data=CO2, iv="uptake",
parameter=c("b0", "b1", "sigma"), newdata=TRUE)
# bR2 returns various information
print(bR2$R2.Summary$R2)                  # R^2
print(bR2$R2.Summary$Variance.Pred.Y)     # Variance of predicted outcome
print(bR2$R2.Summary$Variance.Residuals)  # Variance of residuals
print(head(bR2$R2.Summary$yPRED))         # Predicted outcomes

# MCMC diagnostics--we'll check chain representativeness and accuracy
bDx <- Bayes(losmcmc, y="Dx", parameter="muOfY")
#average ACF for lags 1-37 across 4 chains
print(rowMeans(bDx$Diagnostics$Auto.Correlation.Factor[-1, -1])) #ACF
print(bDx$Diagnostics$Effective.Sample.Size)          #ESS
print(bDx$Diagnostics$Monte.Carlo.Standard.Error)     #MCSE
#Gelman-Rubin statistic (shrink factor) across various iterations of the chains
print(bDx$Diagnostics$Shrink.Factor)                  #Shrink
```

cas *Patient survey data*

Description

Artificial data of a 5 item hospital satisfaction survey for a Cronbach's alpha scale (cas).

Usage

```
cas
```

Format

`cas`**:**

An artificial data frame with 100 rows and 5 columns:

i1 - i5 5 survey items ...

Source

Artificial dataset created with rbinom for 5 items. For example, rbinom(100, 5, .9) generates 1 item. The prob argument is modified to give more or less consistent ratings per item.

co2mcmc *Markov Chain Monte Carlo linear regression estimates of plant's CO2 uptake regressed on ambient carbon dioxide concentrations*

Description

Markov Chain Monte Carlo linear regression estimates of plant's CO2 uptake regressed on ambient carbon dioxide concentrations

Usage

```
co2mcmc
```

Format

`co2mcmc`**:**

MCMC list of 3 chains with 2000 rows and 4 columns in each of the list elements:

b0 Regression intercept.

b1 Regression coefficient of the carbon dioxide concentration effect.

sigma Square root of model residual variance.

lp__ Total log unnormalized joint posterior density of the model parameters. ...

Source

co2mcmc is a list of 3 MCMC simulations. It estimates a linear regression using the CO2 data frame from base R. A regression model was specified using a normal distribution maximum likelihood and normal and uniform distribution priors for the beta coefficients and sigma parameters.

`co2multi` *Bayes class object with summarized Markov Chain Monte Carlo estimates of plant's CO2 uptake as a binary variable that is above or below the median level*

Description

Bayes class object with summarized Markov Chain Monte Carlo estimates of plant's CO2 uptake as a binary variable that is above or below the median level

Usage

```
co2multi
```

Format

`co2multi`:

Bayes class list of 7 elements with varying rows and columns in each of the list elements:

Posterior.Summary Standard element that is NA in this object because it is not applicable to this analysis.

MCMC Dataframe of 4 combined MCMC chain simulations with 8000 rows and 18 columns. Column 1 is CHAIN indicates the 4 chains, columns 2-13 are 'theta' and indicates each plant's rate of having above the median uptake level, columns 14-17 are 'omega' group rates of 'Type' by 'Treatment', and column 18 is the overall group rate.

Multilevel Posterior estimate summaries in alphabetical order of plant or group names and numerical order plant or group rates.

Target Standard element that is NA in this object because it is not applicable to this analysis.

targets Standard element that is NA in this object because it is not applicable to this analysis.

R2.Summary Standard element that is NA in this object because it is not applicable to this analysis.

parameter 3 parameter names in the multilevel analysis, 'theta', 'omega', and 'omegaO'. ...

Source

co2multi is a list of summarized Markov Chain Monte Carlo estimates of plant's CO2 uptake as a binary variable that is above or below the median level. It summarizes measurements nested within plants nested within the 4 levels from a combination of 'Treatment' by 'Type'. An estimation used the CO2 data frame from base R. A hierarchical estimation model was specified using a gamma distribution maximum likelihood and normal and uniform distribution priors at various group levels.

`compapt` *VA's completed hospital appointments made in the reporting period ending in December, 2020.*

Description

VA's completed hospital appointments made in the reporting period ending in December, 2020.

Usage

```
compapt
```

Format

`compapt`:

A public domain data frame with 1,245 rows and 11 columns:

ID Medical Center Area identification number.

State The two character state code.

Name Mnemonic based on name and location.

Comp07 Appointments completed within 0-7 days of contact.

Comp814 Appointments completed within 8-14 days of contact.

Comp1530 Appointments completed within 15-30 days of contact.

Comp14 Appointments completed within 0-14 days of contact.

Comp30 Appointments completed within 0-30 days of contact.

Total Total appointments completed in which patients made contact in December, 2020.

aTotal The aggregated total count of appoinments for each ID of the Medical Center Area.

Sites The number of unique hospital sites found within each ID. ...

Source

compapt is a public domain data frame from the US Vets Administration on the number of completed appointments for the purpose of demonstrating hierarchical estimates of hospital sites nested within larger medical center areas nested within states. Records for 50 US states, D.C., Puerto Rico, and the Philippines attained from the PDF, "DR158_012021_PublicData_CompletedAppointments".

control *Statistics for Shewhart control charts*

Description

Calculate statistics that can be used to produce X-bar charts, p-charts, and u-charts. This includes producing means for center lines, 3-sigma upper and lower control limits. Users can also calculate values before and after an intervention to see if a change in the control process happened. Values are returned in a data frame.

Usage

```
control(
  x,
  y = NULL,
  time,
  data,
  type = "x",
  subset = NULL,
  n.equal = TRUE,
  intervention = NULL
)
```

Arguments

x	character outcome variable in X bar charts or numerator variable name for u-charts or p-charts (when p-chart data is aggregated).
y	character variable name for u-charts or p-charts (when p-chart data is aggregated). When y is present, it becomes the denominator for a rate calculated as x/y. Default is NULL.
time	time variable name.
data	name of data frame object.
type	indicate what type of control chart is needed. Options for the X-bar, p-, and u-charts should be listed as 'x', 'p', and 'u'. The 'u' chart (and p-chart when y is present, e.g., y='denominator') is a ratio of x:y, therefore both x and y column names are required. Default is the 'x' chart.
subset	an expression defining a subset of the observations to use in the grouping. The default is NULL, thereby using all observations. Specify, for example, data$hospital == "NY" or c(1:100,200:300) respectively to use just those observations.
n.equal	whether there are or we assume equal subgroup (sample) sizes. If n.equal=TRUE, control limits are calculated using the overall mean n value. If n.equal=FALSE, control limits are based on each subgroup's sample size. Default is TRUE.
intervention	a single numeric value for the time when an intervention begins (e.g., intervention=25; intervention begins on the 25th day). Separate means and control limits are calculated pre- and post-intervention. Default is NULL.

Value

data frame of control chart statistics for X-bar charts, p-charts, and u-charts. Includes means, standard deviations, and 3-sigma upper and lower control limit values.

References

Ryan, T. (2011). Statistical Methods for Quality Improvement, Third Edition. New Jersey: John Wiley & Sons, Inc. ISBN: 978-0-470-59074-4.

See Also

`plot.control()` for a plot of the 'control' class object.

Examples

```
## Hospital LOS and readmissions ##
# X-bar chart statistics
spc_x <- control(x="los", time="month", data=hosprog, type="x", n.equal=TRUE)
print(spc_x) # get data frame output

# X-bar chart statistics not assuming equal sample sizes, subsetting for females
spc_x <- control(x="los", time="month", data=hosprog, type="x", n.equal=FALSE,
  subset=hosprog$female==1)
print(spc_x) # get data frame output

# p-chart statistics, using only the numerator (i.e., y=NULL). Specify unequal sample sizes
spc_p <- control(x="rdm30", time="month", data=hosprog, type="p", n.equal=FALSE)
print(spc_p) # get data frame output

# u-chart for infection rates with an intervention at the 22nd month
spc_u <- control(x="HAI", y="PatientDays", time="Month", data=infections,
type="u", n.equal=FALSE, intervention=22)
print(spc_u) # get data frame output
```

---

covshort *The US Department of Health and Human Services data for state reported hospital's critical staff shortage or not for December 14, 2020.*

Description

The US Department of Health and Human Services data for state reported hospital's critical staff shortage or not for December 14, 2020.

Usage

```
covshort
```

Format

`covshort`:

A public domain data frame with 54 rows and 5 columns:

state The two character state code.

shortyes Number of hospitals reporting a critical staffing shortage today (December 14, 2020) in this state.

shortno Number of hospitals reporting as not having a critical staffing shortage today (December 14, 2020) in this state.

total The sum total of shortyes and shortno.

rate The proportion of total that is shortyes. ...

Source

covshort is a public domain data frame from the US Department of Health and Human Services on the number of hospitals from each state and territory with a critical staffing shortage or not for December 14, 2020. This is for the purpose of demonstrating hierarchical estimates of hospitals nested within the US. Records attained from the CSV file, "COVID-19_Reported_Patient_Impact_and_Hospital_Capacity_by_State

---

decide *Statistics for Decision Curve Analysis and logistic regression model classification*

Description

Calculate statistics such as sensitivity, specificity, positive and negative predictive values, and net benefit and interventions saved from a decision curve analysis.

Usage

```
decide(x, threshold)
```

Arguments

`x`	assess regression object, currently logistic regression.
`threshold`	numeric vector of length == 1 that sets the threshold used to calculate sensitivity, specificity, positive and negative predictive values, and net benefit and interventions saved from a decision curve analysis. Select thresholds on appropriate linear scale such as logits for logistic regression models.

Value

summary of model classification based on the selected threshold/cutoff, area under the curve (AUC)/c-statistic, predicted outcomes (transformed if applicable), decision curve analysis values at various percentiles, and sensitivity and specificity related statistics for the regression model at the specified threshold.

References

Vickers, A. & Elkin, E. (2006). Decision Curve Analysis: A Novel Method for Evaluating Prediction Models. Society for Medical Decision Making, 26, 6, 565-574. https://doi.org/10.1177/0272989X06295361

See Also

`plot.decide()` for a plot of the 'decide' class object.

Examples

```
## Predicting car engine shape type, v or straight  ##
# run the model
car_m1 <- assess(formula=vs ~ hp + am, data=mtcars, regression="logistic")
# create a decide object, enter the model name and a threshold on the logit scale
d1 <- decide(x=car_m1, threshold= -0.767)
# View model classification related statistics
print(d1$Model.Summary$Classification)

# View decision curve analysis results like 'net benefit' at various thresholds
print(d1$DCA)
```

---

group *Group level confidence intervals and between-group variation*

Description

Group level confidence intervals and between-group variation

Usage

```
group(
  x,
  y,
  z = NULL,
  data,
  dist = "t",
  conf.int = 0.95,
  increment = 1,
  rolling = NULL,
  quarts = FALSE,
  cluster = FALSE,
  subset = NULL,
  asis = FALSE,
  dataf = NULL
)
```

Arguments

x	group predictor variable name.
y	outcome variable name.
z	time period variable name.
data	name of data frame object.
dist	indicate the distribution used for confidence intervals. Options for the t, binomial, and exact Poisson distributions. Options are 't', 'b', and 'p'. Default is the 't'.
conf.int	select the confidence interval level. Default is 0.95.
increment	specify the increment in time periods. Selecting 3 if data uses the month as the unit of time will give confidence intervals, each based on 3 months. Default is 1.
rolling	indicate the number of time periods for the 'rolling average'. The rolling average consists of >1 time periods but subsequent point estimate increase by a unit of 1. For example, the common 12-month rolling average is based on months 1-12 of data, followed by the next estimate using months 2-13, 3-14, and so on until the last month in the data has been reached. Default is NULL.
quarts	logical TRUE or FALSE that indicates whether to convert continuous x into 4 groups based on quartiles of x. Default is FALSE.
cluster	logical TRUE or FALSE to generate measures of between-group variation such as the Intra-Class Correlation, Median Odds Ratio, or Design Effect. Default is FALSE. Uses binary outcome formula (between-group variance/(between-group variance + (3.14^2/3)) for ICC in Rabe-Hesketh which may be more appropriate for multilevel models. ICC, MOR, DE may be less reliable for binomial and Poisson distributions, use caution.
subset	an expression defining a subset of the observations to use in the grouping. The default is NULL, thereby using all observations. Specify, for example, data$hospital == "NY" or c(1:100,200:300) respectively to use just those observations.
asis	a logical vector that indicates if data will be processed as having only 1 unique observation per 'x' and 'z' combination (i.e., this is intended for use with aggregated data). Default is FALSE. This will allow the plot function to graph single observation data for groups over time. Only the t distribution is used for the overall trend line and confidence band (works in conjunction with 'ocol' and 'oband').
dataf	not currently used, please use 'data' instead.

Value

list of confidence intervals for outcomes by groups, over time, and clustering measures. Some values returned in alphabetical and numerical order based on the group.

References

Merlo, J. (2006). A brief conceptual tutorial of multilevel analysis in social epidemiology: using measures of clustering in multilevel logistic regression to investigate contextual phenomena. Journal of Epidemiological Health, 60, 4, 290-297. https://doi.org/10.1136/jech.2004.029454.

Muthen, B. & Satorra, A. (1995). Complex Sample Data in Structural Equation Modeling. Sociological Methodology, 25, 267-316. https://doi.org/10.2307/271070.

Rabe-Hesketh, S. & Skrondal, A. (2008). Multilevel and Longitudinal Modeling Using Stata, Second Edition. ISBN: 978-1-59718-040-5.

See Also

`plot.group()` for a plot of the 'group' class object.

Examples

```
#default t distribution results
group(x="program", y="los", data=hosprog)
#Rounding LOS to integers
hp2 <- hosprog; hp2$los2 <- round(hp2$los, 0)
#Exact Poisson confidence intervals
group(x="program", y="los2", data=hp2, dist="p")
#Rolling 6-months of data
group(x="program", y="los", z="month", data=hosprog, dist="t", rolling=6)
#Data returned separately for rolling 6-months of data and 3-month increments (e.g., quarters)
group(x="program", y="los", z="month", data=hosprog, dist="t", increment=3, rolling=6)
#Quartile groups for continuous risk score and returned clustering info
group(x="risk", y="los", data=hosprog, quarts=TRUE, cluster=TRUE)
#Binomial distribution with less conservative 90% confidence intervals
group(x="risk", y="rdm30", data=hosprog, quarts=TRUE, dist="b", conf.int=0.90)
```

---

hosp1 *Patient hospital program/intervention data, intervention group only*

Description

Patient hospital program/intervention data, intervention group only

Usage

```
hosp1
```

Format

`hosp1`**:**

An artificial data frame with 352 rows and 10 columns, intervention patients only:

survey Patient satisfaction survey mean score.

los Hospital length of stay (los)

cost Hospital stay cost

rdm30 Patient readmission within 30 days of discharge

death30 Patient death within 30 days of discharge

female Patient sex, 1 indicates female, 0 otherwise

age Patient age

risk Patient health risk score ranging from 0 to 1

month 12 month indicator (1 to 12)

program Indicates patient program participation. 1='yes', 0='no' ...

Source

hosp1 is a subset of the artificial dataset hosprog. It is the intervention group's data used for single group interrupted time series.

`hosprog` *Patient hospital program/intervention data*

Description

Patient hospital program/intervention data

Usage

```
hosprog
```

Format

`hosprog`**:**

An artificial data frame with 720 rows and 10 columns:

survey Patient satisfaction survey mean score.

los Hospital length of stay (los)

cost Hospital stay cost

rdm30 Patient readmission within 30 days of discharge

death30 Patient death within 30 days of discharge

female Patient sex, 1 indicates female, 0 otherwise

age Patient age

risk Patient health risk score ranging from 0 to 1

month 12 month indicator (1 to 12)

program Indicates patient program participation. 1='yes', 0='no' ...

Source

Artificial dataset created by using runif. The strength in the association between each variable is weighted by multiplying each subsequent predictor in increments of 1. For example, Y equals runif(720) multiplied by 1 plus runif(720) multiplied by 2 and so on. This allows some predictors to have stronger correlations with Y.

`importance` *Importance of variables based on partial chi-square statistic*

Description

Calculates partial chi-square (Wald chi-square for individual coefficients) from assess class objects. The importance is the partial chi-square minus its degrees of freedom based on the regression coefficients (Harrell, 2015). A higher chi-square indicates a larger effect by the predictors. Therefore, the rank of the chi-square can indicate which predictors can contribute more in explaining the variation in the outcome variable.

Usage

```
importance(model)
```

Arguments

`model`	an assess class object or models with lm or glm class.

Value

a data.frame object with partial X^2 summary statistics.

References

Harrell, F. E., Jr. (2015). Regression Modeling Strategies, Second Edition. Springer International Publishing. ISBN: 978-3-319-19424-0.

See Also

`plot.importance()` for a plot of the 'importance' class object.

Examples

```
# OLS regression
importance(assess(mpg ~ hp + wt + cyl, data=mtcars, regression= "ols")$model)

# logistic regression
importance(assess(vs~mpg+wt+hp, data=mtcars, regression= "logistic")$model)
```

infections *Hospital acquired infections (HAI) during 41 months.*

Description

Hospital acquired infections (HAI) during 41 months.

Usage

```
infections
```

Format

`infections`**:**

An artificial data frame with 41 rows and 3 columns:

Month Calendar month as an integer, ranging from 1 to 41.

HAI Count of hospital acquired infections (HAI) within 1 month.

PatientDays Count of hospital patient days within 1 month (i.e., the sum of days in the hospital for all patients). ...

Source

infections is an artificial data frame of reasonable hospital acquired infections (HAI) estimates made entirely for the purpose of demonstrating control charts.

interpret *Interpret model output*

Description

Provides simple interpretations of regression coefficients, Cronbach's alpha, and Bayesian Markov Chain Monte Carlo diagnostics from assess, alpha, and Bayes function classes. The interpretations describe coefficients and significance values as well as modifying item scales. The interpretations are text comments associated with specific parameters of the various analyses.

Usage

```
interpret(object, digits = NULL)
```

Arguments

`object`	alpha, assess, and Bayes class objects: alpha, ITS, DID, linear (ols) or logistic models, and Dx and target (Bayes diagnostics and target setting).
`digits`	a non-null value for digits specifies the minimum number of significant digits to be printed in values. The default, NULL, uses 3. Non-integer values will be rounded down, and only values greater than or equal to 1 and no greater than 22 are accepted.

Value

a list with interpretations of Cronbach's alpha scales, regression model results, or MCMC diagnostics.

See Also

print.interpret(), itsEffect() for formatted interpret print classes such as Interrupted Time Series effects calculated for 'Summary 1', 'Summary 2', etc.

Examples

```
# Interpret Cronbach's alpha
interpret(alpha(items=c("i1","i2","i3","i4","i5"), data=cas))

# interpret a standard linear (OLS) regression
hos1 <- assess(formula=survey ~ program + month, data=hosprog, regression= "ols")
interpret(hos1)$model

# interpret a differences-in-differences model
hos2 <- assess(formula=survey ~ ., data=hosprog, intervention = "program",
int.time="month", treatment = 5, did="two", newdata=TRUE)
interpret(hos2)$did  #interpret(hos2) also runs, returns ITS results if present

# interpret an interrupted time series model
hos3 <- assess(formula=survey ~ ., data=hosprog, intervention = "program",
int.time="month", its="two", interrupt = 5)
interpret(hos3)$its

# interpret Bayesian MCMC diagnostics
blos1 <- Bayes(losmcmc, y="Dx", parameter="muOfY")
interpret(blos1$Diagnostics, digits=5)
```

---

itsEffect *Interrupted time series analysis effects*

Description

Calculates effects for intervention and control groups based on interrupted time series models from an assess class object. Within a period (or interruption), the effect that represents the trend during the period is calculated for both groups as well as the difference between the groups. Summary statistics are provided that include the effect sizes, t-statistic, standard errors, p-values, and 95% confidence intervals of the effect sizes. These values are provided for the intervention group, control group (when applicable), and the differences between the two groups (Linden, 2015). These values are automatically generated while running a model in assess and represent changes per unit of time in an intervention period. For example, In "Summary 1", ITS effects are the expected differences for each time point (month, year, etc.) within that time period. The difference between intervention and control groups is per unit of time within the first intervention period and is analogous for subsequent

periods. The summary values are calculated only using the trend coefficients for the intervention and control groups each value represents the change per unit of time for that period. For example, if two interruptions in a 1-group ITS model are at months 5 and 9 of the year with coefficients of 0.25 and 0.23, respectively, then adding the baseline trend -0.29 (ITS.Time) and 0.25 (txp) results in a trend change of -0.04 and would be reported under "Summary 1" for the first intervention period. Subsequently, the same coefficients have 0.23 (txp9) added to calculate the 0.19 (= -0.29 + 0.25 + 0.23) trend change value per unit of time in the second intervention period listed in "Summary 2".

Usage

```
itsEffect(model, type, interruptions)
```

Arguments

`model`	an interrupted time series (ITS) model with the "lm" class,
`type`	analysis type for single or multiple groups and single or multiple time periods. If selected type="sgst", it is single-group single-time; type="sgmt", it is single-group multiple-time; type="mgst", it is multiple-group single-time; and type="mgmt", it is multiple-group multiple-time.
`interruptions`	ITS number of interruptions as numeric vector greater than 0.

Value

a data.frame object of ITS effects and summary statistics. Row names are indicated with first, second, and additional numbers when there are multiple interruptions (e.g., 'Intervention.2'). Generally run within the assess function.

References

Linden, Ariel. (2015). Conducting Interrupted Time-series Analysis for Single- and Multiple-group Comparisons. The Stata Journal, 2015, 15(2), 480-500, https://doi.org/10.1177/1536867X1501500208

See Also

`assess()`, `interpret()` for ITS models and interpretation of the 'itsEffect' class object.

Examples

```
i21 <- assess(formula=survey ~ ., data=hosprog, intervention = "program",topcode =NULL,
int.time="month", regression="none", interrupt=5, its="two", newdata=TRUE, propensity=NULL)
itsEffect(model= i21$ITS, type= "mgst", interruptions= 1)
```

`losmcmc` *Markov Chain Monte Carlo estimates of hospital length of stay from the hosprog data frame*

Description

Markov Chain Monte Carlo estimates of hospital length of stay from the hosprog data frame

Usage

```
losmcmc
```

Format

`losmcmc`**:**

MCMC list of 4 chains with 5000 rows and 2 columns in each of the list elements:

muOfY Estimate of the mean parameter.

sigmaOfY Estimate of the square root of variance parameter. ...

Source

losmcmc is a list of 4 MCMC simulations. It estimates the mean and standard deviation using the artificial hosprog data frame from ham. An estimation was specified using a normal distribution maximum likelihood and normal and uniform distribution priors for the mean and sigma parameters.

`losvary` *Markov Chain Monte Carlo estimates of the differences-in-differences of mean and variation for the hospital length of stay from the hosprog data frame*

Description

Markov Chain Monte Carlo estimates of the differences-in-differences of mean and variation for the hospital length of stay from the hosprog data frame

Usage

```
losvary
```

Format

losvary:

MCMC list of 4 chains with 1500 rows and 13 columns in each of the list elements:

CHAIN Estimate of the mean parameter.

muOfLogY[1] Estimate of the log mean parameter for the control group's pre-intervention period.

muOfLogY[2] Estimate of the log mean parameter for the control group's post-intervention period.

muOfLogY[3] Estimate of the log mean parameter for the treatment group's pre-intervention period.

muOfLogY[4] Estimate of the log mean parameter for the treatment group's post-intervention period.

sigmaOfLogY[1] Estimate of the log standard deviation parameter for the control group's pre-intervention period.

sigmaOfLogY[2] Estimate of the log standard deviation parameter for the control group's post-intervention period.

sigmaOfLogY[3] Estimate of the log standard deviation parameter for the treatment group's pre-intervention period.

sigmaOfLogY[4] Estimate of the log standard deviation parameter for the treatment group's post-intervention period.

B3DIDMu Estimate of the differences-in-differences parameter for the mean.

B3DIDMed Estimate of the differences-in-differences parameter for the median.

B3DIDMode Estimate of the differences-in-differences parameter for the mode.

B3DIDSig Estimate of the differences-in-differences standard deviation (variation) parameter.

...

Source

losvary is a list of 4 MCMC simulations. It estimates the mean and standard deviation using the artificial hosprog data frame from ham. An estimation was specified using a log-normal distribution maximum likelihood and normal and uniform distribution priors for the mean and sigma parameters.

NHSN *NHSN standardized incidence ratios for 4 incidents, 2016-2024.*

Description

NHSN standardized incidence ratios for 4 incidents, 2016-2024.

Usage

```
NHSN
```

Format

NHSN:

The CDC's NHSN Acute Care Hospital data frame with 1,800 rows and 10 columns:

Observed Count of observed hospital CAUTI, CLABSI, SSI, & VAE incidents

Predicted Predicted number of CAUTI, CLABSI, SSI, & VAE incidents

Incident Category of CAUTI, CLABSI, SSI, & VAE incidents

State Indicator of location of the 50 United States

Year Calendar year between 2016 - 2024

Yr Ascending numeric rank of Year, ranging from 1 to 9 (2024)

SIR Standardized Incidence (or Infection) Ratio

Lower Lower range of the SIR's 95% Confidence Interval

Upper Upper range of the SIR's 95% Confidence Interval

logPredicted Log of Predicted, use in offset argument for Poisson regression ...

Source

The 2016-2024 NHSN SIR metrics are publicly available data and are calculated using the 2015 national baseline and risk adjustment methodology. More information about these metrics can be found at: cdc.gov. The incidents are from over 30,000 acute care hospitals across the USA and are for catheter-associated urinary tract infections (CAUTI), central line-associated bloodstream infections (CLABSI), surgical site infections (SSI), and ventilator-associated events (VAE).

---

plot.assess *Prediction plot of treatment and control groups for DID and ITS models and OLS, logistic, Poisson, and Cox models*

Description

Provides partial prediction plots for treatment and control groups from difference-in-difference (DID) and interrupted time series (ITS) models as well as some traditional regression models. The graph will produce lines for treatment/intervention and control groups to gain understanding through a visual representation of the regression coefficients. By default, the treatment/intervention group is represented with a blue line, the control group is represented with a red line, and the counterfactual line, when available, is a dashed line. The regression models ('ols', 'logistic', 'poisson', 'coxph') can have partial prediction plots created, target continuous variables are drawn in the range of the 2.5th to the 97.5th percentiles and are held constant at the median values while the reference level is held constant for categorical variables. There are many options to modify the plot.

Usage

```
## S3 method for class 'assess'
plot(
  x,
  y,
  xlim = NULL,
```

```
  ylim = NULL,
  xlab = NULL,
  ylab = NULL,
  main = NULL,
  lwd = NULL,
  col = NULL,
  tcol = NULL,
  cfact = FALSE,
  conf.int = FALSE,
  adj.alpha = NULL,
  add.means = FALSE,
  add.legend = NULL,
  legend = NULL,
  tgt = NULL,
  tgtcol = "gray",
  cex = 1,
  cex.axis = NULL,
  cex.lab = NULL,
  cex.main = NULL,
  cex.text = NULL,
  cex.legend = NULL,
  name = FALSE,
  coefs = FALSE,
  round.c = NULL,
  pos.text = NULL,
  arrow = FALSE,
  xshift = NULL,
  x.axis = NULL,
  ...
)
```

Arguments

x	assess object. Either difference-in-difference, interrupted time series, or some regression models (OLS, logistic, Poisson, Cox proportional hazards).
y	type of model, specify either 'DID' (difference-in-difference) and 'ITS' (interrupted time series). For regression models ('ols', 'logistic', 'poisson', or 'coxph'), y is a 1 or 2 element character vector with the 1st element being a predictor variable that will have the partial prediction plotted and the 2nd element is the type prediction scale response to be graphed. If only the predictor variable is selected in a 1 element character vector (e.g., y= 'age'), the plot will be graphed using the linear predictor. For 'ols' models, the only option is 'lp' (linear predictor). For 'logistic' models, the options are 'lp', 'exp' for exponentiated values such as exp(lp), and 'plogis' for probabilities (i.e., computes the cumulative distribution function for the logistic distribution). For 'coxph' models from the survival package, the options are 'lp' and 'exp' for hazard ratios. For 'poisson' models, the options are 'lp', 'exp', and 'count' and these can differ depending if there is an offset term. For example, when there is an offset in the

	Poisson model, 'lp' is the log on rate scale, 'exp' is the exp(log on rate scale), and 'count' returns the count on outcome scale, based on the offset variable's median raw value when offset(log(n)). When there is no offset in the Poisson model, 'lp' is the log on the outcome scale and both 'exp' and 'count' return the count on the outcome scale (i.e., they are redundant).
`xlim`	specify plot's x-axis limits with a 2 value vector.
`ylim`	specify plot's y-axis limits with a 2 value vector.
`xlab`	a vector label for the x-axis.
`ylab`	a vector label for the y-axis.
`main`	the main title of the plot.
`lwd`	select the line width.
`col`	specify intervention and control group colors in a vector. Defaults to, if nothing selected, c("blue", "red") or "blue" for single-group Interrupted Time Series models.
`tcol`	specify treatment or interruption line color as a single character vector. Defaults to "gray" if nothing selected.
`cfact`	logical TRUE or FALSE that indicates whether a counterfactual line should be included. Defaults to FALSE.
`conf.int`	logical TRUE or FALSE that indicates whether a 95% confidence interval bands should be included. Defaults to FALSE.
`adj.alpha`	factor modifying the opacity alpha of the confidence interval bands, in the range of 0 to 1. Default is NULL; if conf.int=TRUE, defaults to 0.4.
`add.means`	adds group means by time period based on model data. Default is FALSE
`add.legend`	add a legend by selecting the location as "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right", "center". No legend if nothing selected.
`legend`	a character vector of length >= 1 to appear in the DID and ITS model legends to represent the Intervention and Control groups.
`tgt`	specify 1 or more values on the x-axis of where to add a target line when y='group'. Or 1 or more values on the y-axis of where to add a target line when y='time' or 'roll'. Default is NULL.
`tgtcol`	select a color for the target line. Default is 'gray'.
`cex`	A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default of 1.
`cex.axis`	The magnification to be used for axis annotation relative to the current setting of cex.
`cex.lab`	The magnification to be used for x and y labels relative to the current setting of cex.
`cex.main`	The magnification to be used for main titles relative to the current setting of cex.
`cex.text`	The magnification to be used for the text added into the plot relative to the current setting of 1.

cex.legend	The magnification to be used for the legend added into the plot relative to the current setting of 1.
name	logical TRUE or FALSE that indicates whether coefficient names should be added to the plot. Default is FALSE. It is overridden if coefs = TRUE.
coefs	logical TRUE or FALSE that indicates whether coefficient names, values, and p-value significance symbols (+ p<0.10; * p<0.05; ** p<0.01; *** p<0.001) should be added to the plot. Default is FALSE. coefs = TRUE overrides name = FALSE.
round.c	an integer indicating the number of decimal places to be used for rounding coefficient values. Default is 2.
pos.text	a list of named integer value(s) between 1 to 4 indicating the position of the text added into the plot. List name(s) should use coefficient variable names.
arrow	logical TRUE or FALSE that indicates whether arrows and coefficient names should be added to visualize effects. Default is FALSE.
xshift	shifts one or two of some of the overlapping intervention associated arrows along the x-axis for a better view. Vector values of at least length 1 or 2 can be positive or negative. And xshift should be specified in the order of the coefficients. Only 1 or 2 of the furthest right, vertical lines for the intervention group is shifted (i.e., not left). One line is shifted when there is 1 treatment/interruption period and 2 shifts for 2 periods. (e.g., "DID" before "DID.Trend" for DID models with argument did="many").
x.axis	a vector of unique character or numeric values that makes up x-axis values to replace the intervention time variable values. This will be most helpful if you prefer current calendar years instead of values starting at 1 (e.g., x.axis= sort(unique(data$Year)) for 1900-1999, not 1-100). Must have equal lengths for unique x.axis values and unique replaced values. Default is NULL.
...	additional arguments.

Value

plot of partial predictions for treatment and control groups.

See Also

assess() for the 'assess' class object.

Examples

```
mtcars$cyl <- factor(mtcars$cyl)  # first convert 'cyl' into a factor
mod1 <- assess(mpg ~ hp+ cyl, data=mtcars, regression="ols")  #run model
plot(x=mod1, y="hp")  #select the 'hp' variable, default trend for linear predictor ('lp')
# select 'hp', trend for linear predictor ('lp'), trends for each 'cyl' factor level
plot(x=mod1, y=c("hp","lp", "cyl"), col=c("slategray", "red", "green"),
add.legend="bottom", ylim=c(10, 30))
am2 <- assess(formula= los ~ ., data=hosprog, intervention = "program",
topcode =NULL, int.time="month", regression="none", treatment= 5,
interrupt=c(5,9), did="two", its="two", newdata=TRUE, propensity=NULL)
plot(am2, "DID", add.legend="bottomleft", ylim=c(2, 8))  #DID model, basic plot
```

```
plot(am2, "ITS", add.legend="top", ylim=c(2, 8))         #ITS model, basic plot
plot(am2, "DID", add.legend="topleft", main="DID study", col=c("dodgerblue","magenta"),
ylim=c(2, 8), lwd=6, cex=3, cex.axis=2, cex.lab=1.5, cex.main=3, cex.text=2,
arrow=TRUE, xshift=0.02, coefs=TRUE, round.c=2 )
plot(am2, "ITS", add.legend="top", xlim=c(-.5, 13), ylim=c(2, 8), main="ITS study",
col=c("cyan","hotpink"), tcol="springgreen", lwd=7, cex=2, cex.axis=2, cex.lab=2,
cex.main=3, cex.text=1.2, cex.legend=1.25, name=FALSE, coefs=TRUE, round.c=1,
pos.text= list("txp5"=3, "post9"=4), arrow=TRUE, xshift=c(.5, 1.5),
cfact=TRUE, conf.int=TRUE, adj.alpha=0.2)
# ITS: USA's unemployment rate with a single group, 10 interruption model
key_time <- c(5, 14, 17, 29, 42, 59, 69, 73, 80,92)# key interruption periods
u110 <- assess(formula=rate ~ ., data=unemployment, intervention="usa",# model
               int.time="year", its="one", interrupt= key_time)
# Graph the results, note shift in intercept and slope during 1933's New Deal
plot(u110, "ITS", add.means = TRUE, coefs=TRUE, conf.int=TRUE, adj.alpha = .2,
     lwd=2, col="slategray", tcol= "orange", main="US unemployment rate",
     xlab="Years (1929-2024)", ylab="Proportion of labor market",
     cex.main=2, cex.axis = 1.5, cex.lab = 1.5, cex=2, cex.text = 1.25,
    pos.text=list("ITS.Time"=4, "post42"=1,"txp42"=3,"txp92"=3), x.axis=unemployment$Year)
for(i in 1:length(key_time)) {
  text(key_time[i], .22-(.01*i), cex=1.25, labels =
     paste0(unemployment[ key_time[i], "Year"], ": ", unemployment[ key_time[i], "event"]))
}
# We'll review the partial prediction plot for the 'count', keeping in mind 100,
# for this Poisson regression model.
pois1 <- assess(HAI ~ Month + offset(log(PatientDays)),
data = infections, regression="poisson", link="log")
plot(x=pois1, y=c('Month', 'count'), col='skyblue', tgt=100, tgtcol='magenta', lwd=4)
```

`plot.Bayes` *Bayesian plots for various analyses*

Description

Graph Bayesian diagnostic traceplots, density plots on convergence, autocorrelation factor, calculate Monte Carlo Standard Errors, and effective sample sizes. And plot summaries of posterior distributions, posterior predictive checks, summaries on hierarchical or multilevel models (up to 3 levels), and summary graphs of values associated with specific percentiles (and vice-versa) that can be used to help set targets. Plots are developed from Bayes class objects that converted multiple chains into data frames. It is advisable to restart your R session after using the coda package and prior to using ham.

Usage

```
## S3 method for class 'Bayes'
plot(
  x,
  y = NULL,
  type = "n",
```

```
  parameter = NULL,
  center = "mode",
  mass = 0.95,
  compare = NULL,
  rope = NULL,
  data = NULL,
  dv = NULL,
  iv = NULL,
  add.data = "n",
  group = NULL,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  xlim = NULL,
  ylim = NULL,
  vlim = NULL,
  curve = FALSE,
  lwd = NULL,
  breaks = 15,
  bcol = NULL,
  lcol = NULL,
  pcol = NULL,
  xpt = NULL,
  tgt = NULL,
  tgtcol = "gray",
  tpline = NULL,
  tpcol = NULL,
  pline = 20,
  pct = 95,
  add.legend = NULL,
  legend = NULL,
  cex = 1,
  cex.lab = NULL,
  cex.axis = NULL,
  cex.main = NULL,
  cex.text = NULL,
  cex.legend = NULL,
  HDItext = 0.7,
  math = "n",
  es = "n",
  subset = NULL,
  level = NULL,
  aorder = NULL,
  decreasing = NULL,
  round.c = 2,
  ...
)
```

Arguments

x	Bayes class object.
y	character vector for the type of plot to graph. Select 'post', 'dxa', 'dxd', 'dxg', 'dxt', 'check', 'multi' (specialized version of 'check'), or 'target' for posterior summary, diagnostics (4 'dx' plots produced: autocorrelation factor, density plots on chain convergence, Gelman-Rubin statistic, and traceplot), posterior predictive check, multilevel or hierarchical model summary (up to 3 levels), or target summary plots. When y="vary", it allows multiple posterior predictive check lines for various parameters over observed data. This produces a modified version of y="check" for continuous outcomes when 'type' is one of the following: 'n', 'ln', 'sn', 'w', 'g', 't', 'p', 'b'. This was created for a variation analysis, assessed with a Bayesian DID model of the variance (i.e., 'spread') parameter. 'vary' works for parameters that represent the 'average' (i.e., center) or basically whenever you would like to have multiple posterior predictive check lines in the same graph. Default is 'post'.
type	character vector of length == 1 that indicates the likelihood function used in the model when y='check' or y='multi'. Posterior predictive checks allow us to see how well our estimates match the observed data. These checks are available for Bayesian estimation of outcomes and regression trend lines (with polynomial terms) using various distributions in the likelihood function. Select 'n', 'ln', 'sn', 'w', 'g', 'p', 'b', 't', 'taov', 'taov1', 'ol', 'oq','oc', 'lnl', 'lnq', 'lnc', 'logl', 'logq', 'logc', 'bern', and 'bin' for these respective options in Bayesian estimation (multilevel): 'Normal', 'Log-normal', 'Skew-normal', 'Weibull', 'Gamma', 'Poisson', 'Beta' 't', 't: ANOVA, side view', 't: ANOVA 1 group, side view'; and for regression trend lines: 'OLS: Linear', 'OLS: Quadratic', 'OLS: Cubic', 'Log-normal: Linear', 'Log-normal: Quadratic', 'Log-normal: Cubic', 'Logistic: Linear', 'Logistic: Quadratic', and 'Logistic: Cubic', 'Bernoulli', and 'binomial'. The first 8 selections are for Bayesian estimation of outcomes, the next 9 options were developed to assess regression trend lines from ordinary least squares (OLS), log-normal, and logistic models and also for hierarchical model versions. And the remaining two ('bern' and 'bin') are for when y='multi'. Additional models analogous to 'Generalized Linear Models' can also be graphed on the logit scale using 'OLS' options. For example, plot a logistic model on the logit scale when type='ol' (i.e., view straight trend lines). Or if you prefer viewing results on the probability scale, select type='logl' (i.e., curved lines). And consider using type= 'lnl', 'lnq', 'lnc' for log-normal and Poisson models with lines based on exponentiated values. In general, it is important to note that the observed data may not be on the same scale as the parameter estimates and may not be automatically visible in the graph (i.e., x and y-axis limits may help). When graphing target summary plots that use posterior predictive checks rather than the basic posterior summary graph (plots target values over the parameter estimate of the center such as the mean), enter y='target' and other arguments relevant to y='check'. However, type= 'bern' and 'bin' are not available but type='n', 'ln', 'sn', 'w', 'g', 't', 'p', 'b' are available. When y='vary', select 'n', 'ln', 'sn', 'w', 'g', 't', 'p', 'b'. Default is NULL.
parameter	a character vector of length >= 1 or a 2 element list with the name(s) of parameter in MCMC chains to produce summary statistics. Use a 1 element vector to

get posterior estimates of a single parameter. Use a 2 or more element vector to estimate the average joint effects of multiple parameters (e.g., average infection rate for interventions A and B when parameter= c('IntA', 'IntB')). Use a 2 element list to perform mathematical calculations of multiple parameters (see 'math' below). For example, use parameter=list('hospital_A', 'hospital_Z') if you want to estimate the difference between the hospital's outcomes. Use parameter= list(c('hospital_A','hospital_B'), ('hospital_Y','hospital_Z')) to estimate how different the combined hospitals A and B values are from the combined Hospital Y and Z values. When y='check', use either a multiple element character vector that represents center, spread, and additional distribution parameters in order of 1st, 2nd, and 3rd distribution parameters. For example, mean and sd for a normal distribution; mean log and sd log of a log-normal dist.; xi, omega, and alpha of a skew-normal distribution; shape, scale, and lambda of a Weibull distribution; shape and rate of a Gamma distribution; and mean, SD and nu (i.e., degrees of freedom) of a t-distribution. Similarly, when y='vary' (i.e., modified y='check'), parameter is a list object where each element is a vector of column names for parameter (e.g., list(c('muOfY1', 'sdOfY1'), c('muOfY2', 'sdOfY2')))). Indicate regression parameters in order (e.g., intercept, Beta 1, Beta 2, etc.). When y='multi', use a multiple element character vector to list the parameter names of the hierarchy, in order of the nesting with the lowest level first (e.g., exams nested in patients nested in hospital). When y='multi', for parameters from multiple groups such as various hospitals, only enter the first unit's prefix of each parameter and the remaining groups will be set up for graphing. For example, parameter=c('theta', 'omega') will plot data for `theta[1]` to `theta[8]` and `omega[1]` to `omega[8]` for all 8 hospitals as well.

`center` character vector that selects the type of central tendency to use when reporting parameter values. Choices include: 'mean', 'median', and 'mode'. Default is 'mode'.

`mass` numeric vector that specifies the credible mass used in the Highest Density Interval (HDI). Default is 0.95.

`compare` numeric vector with one comparison value to determine how much of the distribution is above or below the comparison value. Default is NULL.

`rope` numeric vector with two values that define the Region of Practical Equivalence (ROPE). Test hypotheses by setting low and high values to determine if the Highest Density Interval (HDI) is within or outside of the ROPE. Parameter values are declared not credible if the entire ROPE lies outside the HDI of the parameter's posterior (i.e., we reject the null hypothesis). For example, the ROPE of a coin is set to 0.45 to 0.55 but the posterior 95% HDI is 0.61 - 0.69 so we reject the null hypothesis value of the rate of a head is 0.50. We can accept the null hypothesis if the entire 95% HDI falls with the ROPE. Default is NULL.

`data` object name for the observed data when y='check', y='multi' or y='target'.

`dv` character vector of length == 1 for the dependent variable name in the observed data frame when y='check', y='multi' or y='target'. Default is NULL.

`iv` character vector of length >= 1 for the independent variable name(s) in the observed data frame when y='check' or y='multi'. When y='multi', enter the lower to higher level clustering or group names (e.g, for health data, iv=c("patient",

"hospital"). When type='taov', enter the name of the test group variable (e.g., 'intervention'). Default is NULL.

`add.data` character vector of length == 1 to determine the type of observed data added to the plot (to show model fit to data) when y='check' and type= 'ol', 'oq','oc', 'lnl', 'lnq', 'lnc', 'logl', 'logq', or 'logc'. Select 'a', 'u', 'al', 'ul', 'n' for these observed data options: 'All', 'Unit', 'All: Lines', 'Unit: Lines' (unit specific lines are linked), 'none'. Default is 'n' for none or no observed data shown.

`group` character list of length == 2 for 1) the grouping variable name and 2) specific group(s) in the observed data frame. This is primarily used for multilevel or hierarchical models when y='check' or y='multi' that the hierarchies are based on (e.g., hospitals nested within health systems).

`main` the main title of the plot.

`xlab` a character vector label for the x-axis.

`ylab` a character vector label for the y-axis.

`xlim` specify plot's x-axis limits with a 2 element numeric vector.

`ylim` specify plot's y-axis limits with a 2 element numeric vector.

`vlim` two element vector to specify limits for minimum and maximum values used to extrapolate posterior lines along the x-axis. For example, when drawing a log-normal distribution, we may want to have our posterior lines fit within a narrower range while having our graph's x-axis limits extend past those lines. If so, the value limits (vlim) help us keep our posterior predictive check lines within desired limits. When y='check', 'target', or 'vary' and type='b', may need to include vlim argument (e.g., vlim=c(0.0, 1.0)) because the default x-axis might range from 1 to higher, depending on your data, while the beta distribution may have a 0 to 1 range. Default is NULL.

`curve` select a curve to display instead of a histogram when y='post'. Default is FALSE.

`lwd` select the line width.

`breaks` number of breaks in a histogram. Default is 15.

`bcol` a single or multiple element character vector to specify the bar or band color(s). When Bayesian estimates and observed values are present, the first colors are for Bayesian estimates while the last colors are observed values. Defaults to, if nothing selected, 'gray', except when y = 'multi' and then no overall HDI is graphed until a color is selected.

`lcol` a single or multiple element character vector to specify the line color(s). When Bayesian estimates and observed values are present, the first colors are Bayesian estimates while the last colors are observed values. When multiple lines are needed, single item lines precede multiple use lines. For example, a single comparison value line will be assigned the first lcol while both rope lines will be given the same color of the second lcol when y='post'. Defaults to 'gray' if nothing selected.

`pcol` a single or multiple element character vector to specify the point color(s). When Bayesian estimates and observed values are present, the first colors are Bayesian estimates while the last colors are observed values. Defaults to, if nothing selected, 'gray'.

xpt	a numeric vector of single or multiple values that indicate placement of points (+) on the x-axis when y='check'. This is intended for the graphs with predictive checks on Bayesian estimation (i.e., not trend lines). Default is NULL.
tgt	specify 1 or more values on the x- or y-axis of where to add one or more target lines when applicable. Default is NULL.
tgtcol	select one or multiple colors for one or multiple target lines. Default is 'gray'.
tpline	add one or more time point vertical lines using x-axis values. Default is NULL (i.e., no lines).
tpcol	specify a color for the time point line, tpline. Default is NULL.
pline	a numeric vector of length == 1 for the number of random posterior predictive check lines when y='check'. Default is 20.
pct	a numeric integer vector of length == 1 for the percentage of the posterior predictive check heavy tail lines to be drawn when type= 'taov' or 'taov1'. Valid values are 0 < pct < 100. Default is 95 (e.g., 95%).
add.legend	add a legend by selecting the location as "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right", "center". Default is no legend produced if nothing is selected.
legend	a character vector of length >= 1 to appear when y='check', y='multi', and sometimes y='target'. Legends to represent hierarchical estimates and observed values.
cex	A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default of 1.
cex.lab	The magnification to be used for x and y labels relative to the current setting of cex.
cex.axis	The magnification to be used for axis annotation relative to the current setting of cex.
cex.main	The magnification to be used for main titles relative to the current setting of cex.
cex.text	The magnification to be used for the text added into the plot relative to the current setting of 1.
cex.legend	The magnification to be used for the legend added into the plot relative to the current setting of 1.
HDItext	numeric vector of length == 1 that can be a negative or positive value. Identifies placement of HDI text near credible interval when y='post'. Values are relative to the x-axis values. Default is 0.7.
math	mathematics function performed between multiple parameters when y='post'. Available functions are: 'add', 'subtract', 'multiply', 'divide', or 'n' for none (i,e., no functions). Indicate parameters with parameter argument. For example, when math='subtract', use parameter=list('hospital_A', 'hospital_Z') if you want to estimate the difference between the hospital's outcomes. Use parameter=list(c('hospital_A','hospital_B'), ('hospital_Y','hospital_Z')) to estimate how different the combined hospitals A and B values are from the combined hospitals Y and Z. Additionally, compute statistics like the coefficient of variation when math='divide' and parameter= list('Standard_Deviation', 'Mean'). Default is 'n' for no math function.

es	one element vector that indicates which type of likelihood distribution is relevant in calculating Jacob Cohen's effect sizes between 2 parameters when y='post'. Options are 'bern', 'bin', and 'n' for the Bernoulli or binomial distributions for binary outcomes and none (i.e., no distribution, hence no effect size calculated). For example, to get the posterior distribution summary for the difference between the intervention and control groups on 30-day readmissions or not, use es='bern' or 'bin' when y='post', math='subtract', and parameter=list('intMean', 'ctlMean'). Default is 'n' which indicates not to calculate the effect size.
subset	a single or multiple element character or numeric vector of group names that are a subset of the observations to use in the grouping when y='multi'. The default is NULL, thereby using all observations. Specify, for example, enter c('NY', 'Toronto', 'LA', 'Vancouver') to view a graph with only these cities. Default is NULL.
level	a numeric integer of length == 1, either 1, 2, or 3 that indicates the level of the hierarchical/multilevel model when y='multi' and the type of graph to plot. For example, a multilevel model that estimates the proportion of successful exams by patients is considered level=2. And the successful exam rates by patients from various hospitals is level=3. Graphs can be created separately for both level=2 and level=3 when there is a three-level model. The graph when y='multi' can be produced when level=1 for non-hierarchical models if there are estimates for groups. For example, estimating the patient infection rate of hospitals without a hierarchical structure in the model. Default is NULL.
aorder	a logical indicator on whether the ordering of the group levels are in alphabetical order or not when y='multi'. If aorder=TRUE, results are displayed in an increasing alphabetical order based on level name (e.g., 'LA' before 'NY'). If aorder=FALSE, an increasing numeric order based on group parameter values is performed (e.g., 0.65 before 0.70). Default is TRUE.
decreasing	logical TRUE or FALSE that indicates a decreasing or increasing order in displayed results in the y='multi' graph. When decreasing=TRUE, the results go from high to low in alphabetical or numerical order. When decreasing=FALSE, results go from low to high. Default is decreasing=FALSE.
round.c	an integer indicating the number of decimal places when rounding numbers y='multi' and y='target'. Default is 2.
...	additional arguments.

Value

plots assessing model quality, posterior distributions, posterior predictive checks, hierarchical or multilevel model summaries, and target summaries.

References

Cohen, J. (1988). Statistical Power Analysis for the Behavioral Sciences, Second Edition. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers. ISBN 0-8058-0283-5 Gelman, A., Goodrich, B., Gabry, J., & Vehtari, A. (2019). R-squared for Bayesian Regression Models. The American Statistician, 73, 3, 307–309. https://doi.org/10.1080/00031305.2018.1549100

Kruschke, J. (2014). Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan, Second Edition. New York: Academic Press. ISBN: 9780124058880

See Also

Bayes() for the 'Bayes' class object.

Examples

```
# Posterior estimates of length of stay #
# Set up the Bayes object first, convert the chains
blos1 <- Bayes(losmcmc, newdata=TRUE)

#' # Examine Bayesian model diagnostics for estimated length of stay (LOS) #
# Review traceplots, density plots, autocorrelation factor, and Gelman-Rubin statistic
plot(blos1, y="dxt", parameter="muOfY")
plot(blos1, y="dxd", parameter="muOfY")
plot(blos1, y="dxa", parameter="muOfY")
plot(blos1, y="dxg", parameter="muOfY")

# And we can calculate statistics that combine multiple parameters (e.g., calculate
# the mean difference between intervention and control groups). Here we get the
# coefficient of variation by dividing the standard deviation by the mean. We use
# the 'math' argument and arrange our parameters in the proper order.
# We only need the first 4 arguments but we'll add a little more.
plot(x=blos1, y="post", parameter=list("sigmaOfY", "muOfY" ),math="divide",
bcol="cyan", HDItext=.3, main= "Coefficient of Variation")

# Posterior Predictive Checks on how well our model fits the data #
# Estimating center and spread for hospital length of stay
# Generally, we only need the first 6 arguments but we'll modify the plot.
# A model with a gamma likelihood would fit better but this is ok for now.
plot(x=blos1, y="check", type="n", data=hosprog, dv="los",
parameter=c("muOfY", "sigmaOfY"), breaks=30, cex.axis=1.3, lwd=3, xlab=NULL,
pline=20, vlim=c(-2, 20), xlim=c(-2, 20), add.legend="topright",
main="Length of Stay", cex.main=1.5, xpt=5, pcol="red", lcol="orange",
cex.legend=1, bcol="cyan")

# When y='vary', this alternative Posterior Predictive Check involves reviewing lines
# of multiple estimates to assess variation. Key to hospital quality improvement
# goals is reducing variation. For example, viewing both the pre- and post-intervention
# lines can help show that lower length of stay (LOS) hospital visits increased while
# the very high LOS visits decreased. In addition, a Bayesian differences-in-differences
# model assessing the difference in the variation parameter (e.g., standard deviation)
# provides useful information. We show both in this example that uses a log-normal
# likelihood function.

# These make 4 levels for the intervention program * pre/post period
hosprog$prepost <- ifelse(hosprog$month >= 5, 1, 0) #pre/post indicator
# Control group at pre-test
hosprog$Post.Int <- 1
# Control group at post-test
hosprog[, "Post.Int"][hosprog$program == 0 & hosprog$prepost == 1 ] <- 2
```

```
# Intervention group at pre-test
hosprog[, "Post.Int"][hosprog$program == 1 & hosprog$prepost == 0 ] <- 3
# Intervention group at post-test
hosprog[, "Post.Int"][hosprog$program == 1 & hosprog$prepost == 1 ] <- 4
# Create a ham object out of losvary
bvlos <- Bayes(x=losvary, newdata=TRUE)
# list of key center and spread parameters for post check
parls <- list(c("muOfLogY[1]", "sigmaOfLogY[1]"), c("muOfLogY[2]", "sigmaOfLogY[2]"),
              c("muOfLogY[3]", "sigmaOfLogY[3]"), c("muOfLogY[4]", "sigmaOfLogY[4]"))
# We now look at the variation DID parameter which excludes 0 (i.e., significant)
plot(x=bvlos, y="post", parameter="B3DIDSig", bcol="green", compare=0,
HDItext=.3, cex=2, main= "B3DIDSig")

# Let's take a look at the various posterior predictive checks with y='vary'.
# The control group increased standard deviation post-intervention period by
# 1/2 a day while the treatment group decreased it by 1/2 a day.
plot(x=bvlos, y="vary", type="ln", dv="los", breaks=75, pline=3, lwd=1, vlim=c(0, 15),
data=hosprog, lcol= c("pink","red", "cyan", "blue"), xlim=c(0.5, 15), ylim=c(0, .35),
parameter=parls, main="LOS and posterior pred checks", cex.main=2, cex.legend=1.5,
cex.label=1.5, cex.axis=1.5, add.legend="topright",
legend=c("Observed Data", "CTL Pre", "CTL Post", "Treat Pre", "Treat Post"))

# Estimating the regression trend line
# Now lets look at the trend of conc on CO2 uptake from the CO2 data.
# Using a quadratic model with conc^2 would help and an option in ham.
# First, create the Bayes object
bco2 <- Bayes(x=co2mcmc, newdata=TRUE )
# We generally only need the first 7 arguments below.
plot(x=bco2, y="check", type="ol", data=CO2, dv="uptake", iv="conc",
parameter=c("b0","b1"), add.data="al", cex.axis=1.3, lwd=1.5, pline=50,
vlim=c(50, 1100), xlim=c(0, 1100), ylim=c(0, 50), cex=2, cex.lab=2,
pcol="magenta", cex.main=2,cex.legend=1.2,  add.legend="topleft",
lcol="steelblue")              #vlim lets me extrapolate a little

# Hierarchical or Multilevel Model Summary #
# We generally only need the first 3 arguments below. But we'll subset
# on 8 of 12 plants in the level 2 model (observations nested in plants)
# and modify other settings.
plot(x=co2multi, y="multi", level=2, aorder=FALSE, decreasing=FALSE,
subset= c("Qn2","Qn3","Qc3","Qc2","Mn3","Mn2","Mc2","Mc3"),
lcol="blue", pcol= c("red", "skyblue"), round.c=1, bcol="yellow",
xlim=c(-.1, 1), legend=NULL, add.legend="topright", lwd=3, cex.lab=1.2,
cex= 2, cex.main=1.5, cex.axis=.75, cex.legend=1.5, X.Lab=NULL)
# And now the level 3 plot (observation in plants in Treatment by type groups)
plot(x=co2multi, y="multi", level=3, aorder=FALSE, decreasing=FALSE, lcol="blue",
pcol= c("green", "pink"), round.c=1, bcol="lavender", xlim=c(-.1, 1), legend=NULL,
add.legend="right", lwd=3, cex.lab =1.2, cex= 2, cex.main=1.5, cex.axis=.75,
cex.legend=1.5, X.Lab=NULL)

# Targets for length of stay (LOS) #
# Our administrators ask how far are we from our goals, they ask about targets
# in increments of 5 points of probability or specific days. We answer both.
btarget1 <- Bayes(x=losmcmc, y="target", type="n", parameter=c("muOfY","sigmaOfY"),
```

```
newdata=TRUE, target=list(p=c(.35,.4,.45, .5, .55),  y=c(3,4))) # 'newdata' for plots
# Target graph using the standard option, overlaying estimate of mode parameter
plot(x=btarget1, y="target", type="n", lcol="purple", tgtcol="blue", xlim=c(3.5, 5))
# Target graph using a posterior predictive check, more intuitive
plot(x=btarget1, y="target", type="n", data=hosprog, dv="los", breaks=30,
cex.axis=1.3, lwd=3, pline=20, vlim=c(-1, 12), xlim=c(-1, 10),
parameter=c("muOfY","sigmaOfY"), add.legend="topright", main="Length of Stay",
cex.main=1.5, xpt=5, pcol="black", lcol="cyan", tgtcol="blue", bcol="orange",
cex.legend=1.5, cex.text = 2)
```

`plot.control` *Shewhart control charts*

Description

Graph X-bar charts, p-charts, and u-charts. This includes producing means center lines, 3-sigma upper and lower control limits. Users can also calculate values before and after an intervention to see if a change in the control process happened. Values are returned in a data frame.

Usage

```
## S3 method for class 'control'
plot(
  x,
  y = NULL,
  xlim = NULL,
  ylim = NULL,
  xlab = NULL,
  ylab = NULL,
  main = NULL,
  lwd = NULL,
  col = NULL,
  iname = NULL,
  icol = "black",
  trend = FALSE,
  trcol = "gray",
  tgt = NULL,
  tgtcol = "gray",
  tpline = NULL,
  tpcol = NULL,
  cex = 1,
  cex.lab = NULL,
  cex.axis = NULL,
  cex.main = NULL,
  cex.text = NULL,
  x.axis = NULL,
  y.axis = NULL,
  round.c = NULL,
```

```
  ...
)
```

Arguments

x	control object.
y	not currently used, default is NULL.
xlim	specify plot's x-axis limits with a 2 element numeric vector.
ylim	specify plot's y-axis limits with a 2 element numeric vector.
xlab	a character vector label for the x-axis.
ylab	a character vector label for the y-axis.
main	the main title of the plot.
lwd	select the line width.
col	a 2 element character vector to specify the 1) center line and 2) both control limit colors. Defaults to, if nothing selected, c("blue", "red").
iname	intervention name text as a single character vector. Defaults to "Intervention" if nothing is selected.
icol	specify intervention line color as a single character vector. Defaults to "black" if nothing is selected.
trend	a logical vector that adds an ordinary least squares trend line (i.e., simple linear regression line) when trend=TRUE. Default is FALSE.
trcol	select a color for the trend line. Default is 'gray'.
tgt	specify 1 or more values on the y-axis of where to add one or more horizontal target lines. Default is NULL.
tgtcol	select one or multiple colors for one or multiple target lines. Default is 'gray'.
tpline	add one or more time point vertical lines using x-axis values. Default is NULL (i.e., no lines).
tpcol	specify a color for the time point line, tpline. Default is NULL.
cex	A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default of 1.
cex.lab	The magnification to be used for x and y labels relative to the current setting of cex.
cex.axis	The magnification to be used for axis annotation relative to the current setting of cex.
cex.main	The magnification to be used for main titles relative to the current setting of cex.
cex.text	The magnification to be used for the iname text added into the plot relative to the current setting of 1.
x.axis	a vector of unique character or numeric values that makes up x-axis values to replace the time variable values. This will be most helpful if you prefer current calendar months/years instead of values starting at 1 (e.g., x.axis= sort(unique(data$Year)) for 1900-1999, not 1-100). Must have equal lengths for unique x.axis values and replaced values (i.e., nrow(x)). Default is NULL.

y.axis	a vector of unique character or numeric values that makes up y-axis values to replace the outcome variable values. This will be most helpful if your outcome needs to be converted such as rate per 1,000 patient days (e.g., y.axis= seq(min(c(xHAI, xLCL))*1000, max(c(xHAI, xUCL))*1000, length.out=10)). Default is NULL.
round.c	an integer indicating the number of decimal places when rounding numbers such as for y.axis. Default is 2.
...	additional arguments.

Value

plot of Shewhart control charts: X-bar charts, p-charts, and u-charts with 3-sigma control limits.

References

Ryan, T. (2011). Statistical Methods for Quality Improvement, Third Edition. John Wiley & Sons, Inc. ISBN: 978-0-470-59074-4.

See Also

control() for the 'control' class object.

Examples

```
## Hospital LOS and readmissions ##
# X-bar chart
spc_x <- control(x="los", time="month", data=hosprog, type="x", n.equal=TRUE)
# Basic X-bar chart
plot(spc_x)

# p-chart, using only the numerator (i.e., y=NULL). Specify unequal sample sizes
spc_p <- control(x="rdm30", time="month", data=hosprog, type="p", n.equal=FALSE)
# p-chart, adding target and time point lines
plot(spc_p, tgt=c(0,.25), tgtcol="green", ylim=c(0,0.4), tpline=c(4,8),
tpcol= c("yellow","black"))

# u-chart for infection rates with an intervention
spc_u <- control(x="HAI", y="PatientDays", time="Month", data=infections,
type="u", n.equal=FALSE, intervention=22)
# u-chart with trend lines, various graphing options, x.axis start at 2nd year
# and y.axis changed to show HAIs per 1,000 patient days
plot(spc_u, main="u-Chart: HAI per 1,000 Patient Days Pre/Post Intervention",
col=c("green","dodgerblue"), trend=TRUE, trcol="red", x.axis=c((1:41+12)), round.c=1,
y.axis=seq(min(c(spc_u[,2],spc_u$LCL))*1000, max(c(spc_u[,2],spc_u$UCL))*1000, length.out=10),
xlab="Months (starting at year 2)", icol="gray", lwd=2, cex=2,
cex.axis=1.1, cex.main=1.25, cex.text=1.25)
```

`plot.decide` *Decision Curve Analysis plots and regression model classification graphs on sensitivity and specificity*

Description

Graph decide class object's model classification results as well as Decision Curve analysis output for net benefit and interventions saved.

Usage

```
## S3 method for class 'decide'
plot(
  x,
  y = NULL,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  xlim = NULL,
  ylim = NULL,
  lwd = NULL,
  bcol = NULL,
  lcol = NULL,
  add.legend = NULL,
  legend = NULL,
  cex = 1,
  cex.lab = NULL,
  cex.axis = NULL,
  cex.main = NULL,
  cex.legend = NULL,
  round.c = 2,
  ...
)
```

Arguments

`x`	decide object.
`y`	type of plot to display. Select either 'nb', 'is', or 'cl' for a decision curve analysis 'net benefit' and 'interventions saved', or a model classification (e.g., sensitivity and specificity) according to a selected threshold. Net benefit and interventions saved display results for specific percentiles found between the 1st and 99th percentiles. Default is 'nb'.
`main`	the main title of the plot.
`xlab`	a character vector label for the x-axis.
`ylab`	a character vector label for the y-axis.

xlim	specify plot's x-axis limits with a 2 element numeric vector.
ylim	specify plot's y-axis limits with a 2 element numeric vector. When y='is', the y-axis labels are in integers (e.g., 1 to 100) but the actual scale is in decimals (0.0 to 1.0), please note when making y-axis limits.
lwd	select the line width.
bcol	a multiple element character vector of length == 2 to specify the bar, band, or block colors that are the shading of the true and false classification regions of the plot (e.g., true-positive and false-negative). When y='cl', the first color represents 'true' and the second color is for 'negative'. Default is null, if none selected, the colors are c('green', 'red').
lcol	a single or multiple element character vector to specify the line color(s). When y='nb', select up to 3 colors in this order for model, 'net benefit', 'all treated', and 'none treated' line colors. When y='is', select 1 color for the interventions saved line. And when y='cl', select 1 color to represent the selected threshold.
add.legend	add a legend by selecting the location as "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right", "center". Default is no legend produced if nothing is selected.
legend	a character vector of length >= 2 to appear when y='nb' and y='cl' with legend description.
cex	A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default of 1.
cex.lab	The magnification to be used for x and y labels relative to the current setting of cex.
cex.axis	The magnification to be used for axis annotation relative to the current setting of cex.
cex.main	The magnification to be used for main titles relative to the current setting of cex.
cex.legend	The magnification to be used for the legend added into the plot relative to the current setting of 1.
round.c	an integer indicating the number of decimal places when rounding numbers y='multi' and y='target'. Default is 2.
...	additional arguments.

Value

plots of model classification, net benefit, and interventions saved.

References

Vickers, A. & Elkin, E. (2006). Decision Curve Analysis: A Novel Method for Evaluating Prediction Models. Society for Medical Decision Making, 26, 6, 565-574. https://doi.org/10.1177/0272989X06295361

See Also

decide() for the 'decide' class object.

Examples

```
## Predicting car engine shape type, v or straight  ##
# run the model
car_m1 <- assess(formula=vs ~ hp + am, data=mtcars, regression="logistic")
# create a decide object, enter the model name and a threshold on the logit scale
d1 <- decide(x=car_m1, threshold= -0.767)
#Plot the classification results
plot(x=d1, y= "cl")

#Plot the net benefit
plot(x=d1, y= "nb")

#Plot the interventions saved
plot(x=d1, y= "is")
```

`plot.group` *Confidence interval graphs for group class objects*

Description

Confidence interval graphs for group class objects

Usage

```
## S3 method for class 'group'
plot(
  x,
  y = "group",
  order = "alpha",
  decreasing = NULL,
  gcol = "blue",
  gband = FALSE,
  pcol = "red",
  overall = FALSE,
  ocol = "gray",
  oband = FALSE,
  tgt = NULL,
  tcol = "gray",
  tpline = NULL,
  tpcol = "gray",
  xlim = NULL,
  ylim = NULL,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  lwd = 1,
  adj.alpha = 0.4,
```

```
  cex = 1,
  cex.axis = 1,
  cex.lab = 1,
  cex.main = 1,
  cex.text = 1,
  round.c = 2,
  name = FALSE,
  abbrv = 5,
  roll.x = "Stop",
  x.axis = NULL,
  y.axis = NULL,
  ...
)
```

Arguments

x	group object.
y	type of confidence interval object, specify either 'group', 'time', or 'roll'.
order	specify confidence interval object order as 'alpha' or 'numeric' for alphabetical or numerical ordering in the y='group' graph.
decreasing	logical TRUE or FALSE that indicates a decreasing or increasing order in displayed results in the y='group' graph. When decreasing=TRUE, the results go from high to low in alphabetical or numerical order. When decreasing=FALSE, results go from low to high. Default is decreasing=FALSE.
gcol	pick confidence interval line colors for groups in the 'group' graph. Default is 'blue'.
gband	logical TRUE or FALSE that indicates whether group lines have confidence bands for trend over time results. Default is FALSE.
pcol	select point color for 'group' only confidence intervals. Default is 'red'.
overall	logical TRUE or FALSE that indicates whether to include the overall sample confidence intervals (i.e., not each group). Default is FALSE.
ocol	indicate the optional overall line color. Default is 'gray' when overall=TRUE.
oband	logical TRUE or FALSE that indicates whether to add an overall confidence band. Default is FALSE.
tgt	specify 1 or more values on the x-axis of where to add a target line when y='group'. Or 1 or more values on the y-axis of where to add a target line when y='time' or 'roll'. Default is NULL.
tcol	select a color for the target line. Default is 'gray'.
tpline	add one or more time point vertical line(s) using x-axis values when y='time' or y='roll'. Default is NULL.
tpcol	specify a color for the time point line, tpline. Default is NULL.
xlim	specify plot's x-axis limits with a 2 value vector.
ylim	specify plot's y-axis limits with a 2 value vector.
main	the main title of the plot.

xlab	a vector label for the x-axis.
ylab	a vector label for the y-axis.
lwd	select the line width. Default is 1.
adj.alpha	factor modifying the opacity alpha of the confidence interval bands, in the range of 0 to 1. Default is 0.4.
cex	A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default of 1.
cex.axis	The magnification to be used for axis annotation relative to the current setting of cex. Default is 1.
cex.lab	The magnification to be used for x and y labels relative to the current setting of cex. Default is 1.
cex.main	The magnification to be used for main titles relative to the current setting of cex. Default is 1.
cex.text	The magnification to be used for the text added into the plot relative to the current setting of 1.
round.c	an integer indicating the number of decimal places to be used for rounding values. Default is 2.
name	logical TRUE or FALSE that indicates whether group names should be added to the 'time' or 'roll' plots. Default is FALSE.
abbrv	the minimum length of the abbreviations. Default is 5.
roll.x	one of 2 character options to graph rolling average x-axis values as either the 'Start' or 'Stop' of the time period. Selecting 'Start' indicates the time point on the x-axis as the first month (e.g., roll=12 will have a graph that starts at 1 on the x-axis). The default is 'Stop' (e.g., roll=12 will have a graph that starts at 12 on the x-axis).
x.axis	a vector of unique character or numeric values that makes up x-axis values to replace the time variable values. For use only when y="time" or y="roll". This will be most helpful if you prefer current calendar months/years instead of values starting at 1 (e.g., x.axis= sort(unique(data$Year)) for 1900-1999, not 1-100). Must have equal lengths for unique x.axis values and replaced values (i.e., nrow(x)). Default is NULL.
y.axis	a vector of unique character or numeric values that makes up y-axis values to replace the outcome variable values. For use only when y="time" or y="roll". This will be most helpful if your outcome needs to be converted such as rate per 1,000 patient days. The values are spaced equally between the minimum lower confidence interval and maximum upper confidence interval when either y="time" or y="roll", equal to length(y.axis). Review x$Time.CI$Group or x$Roll.CI$Rolling.All to view ranges. Default is NULL.
...	additional arguments.

Value

plot of group level confidence intervals, including estimates over time periods.

See Also

group() for the 'group' class object.

Examples

```
#Simple graph for confidence intervals using the t-distribution
gr1 <- group(x="program", y="los", z="month", data=hosprog, dist="t",
increment=3, rolling=6)
# Group level confidence intervals
plot(x=gr1, y="group", order="numeric", lwd=4, gcol= "blue", pcol="red",
overall=TRUE, oband=TRUE, ocol="gray", tcol="green", tgt=4.5,
cex=1, cex.axis=1, cex.lab=1, cex.text=2,
cex.main=1.25, name=TRUE, adj.alpha=.2)
#Trend plots over time in the 3 month increments (i.e., quarters)
plot(x=gr1, y="time", lwd=4, gcol=c("red", "blue"), gband=TRUE, overall=TRUE,
  oband=TRUE, ocol="gray", tcol="green", tgt=4, tpline=3,
  tpcol="yellow", name=TRUE, cex.axis=1, cex.lab=1, cex.text=2,
  cex.main=1.25, adj.alpha=.3)
#Plot for rolling 6-month averages
plot(x=gr1, y="roll", lwd=4, gcol=c("red", "blue"), gband=TRUE, overall=TRUE,
  oband=TRUE, ocol="gray", tcol="green", tgt=4, tpline=c(4,6),
  tpcol="yellow", name=TRUE, cex.axis=1, cex.lab=1, cex.text=2,
  cex.main=1.25, adj.alpha=.3)

 # Helpful note: You can plot aggregated data if you used the argument asis=TRUE in group(),
```

---

plot.importance *Plot of variable importance ranked by partial chi-square statistic*

Description

Plots an importance class object. Produces a dot chart that places the predictor variable with the highest partial chi-square (Wald chi-square for individual coefficients) at the bottom. It is a metric of the partial chi-square minus its degrees of freedom (Harrell, 2015). Predictor variables with significant p-values at the 0.05 alpha are highlighted red. Consider graphical parameters of mar=c(4.2, 2, 3.5, 3) and oma = c(0, 0, 0, 3).

Usage

```
## S3 method for class 'importance'
plot(
  x,
  y,
  main = NULL,
  cex = NULL,
  pt.cex = NULL,
  pch = NULL,
  color = NULL,
```

```
  lcolor = NULL,
  abbrv = NULL,
  ...
)
```

Arguments

x	importance object.
y	not currently used.
main	overall title for the plot, default is 'Variable Importance'.
cex	the character size to be used. Setting cex to a value smaller than 1 can be a useful way of avoiding label overlap. This sets the actual size, not a multiple of par('cex').
pt.cex	the cex to be applied to plotting symbols, default is 2.
pch	the plotting character or symbol to be used, default is 19.
color	the color to be used for points and labels when there are significant results. Default is 'red'.
lcolor	the color(s) to be used for the horizontal lines. Default is 'gray'.
abbrv	the minimum length of the character length for the coefficient name abbreviations. Default is 7.
...	additional arguments.

Value

plot of variable importance, significant variables highlighted in red.

References

Harrell, F. E., Jr. (2015). Regression Modeling Strategies, Second Edition. Springer International Publishing. ISBN: 978-3-319-19424-0.

See Also

importance() for the 'importance' class object.

Examples

```
# OLS regression
plot(importance(assess(mpg ~ hp + wt + cyl, data=mtcars, regression= "ols")$model))

# logistic regression
plot(importance(assess(vs~mpg+wt+hp, data=mtcars, regression= "logistic")$model))
```

plot.review *Plot of a regression model's coefficient review*

Description

Plots a review class object of a regression model. Plots a chart of the predictor's coefficient estimates with 95% confidence intervals (Harrell, 2015). Excludes the intercept (if present). Horizontal line segments represent the predictor variable coefficients and the vertical line is at 0 or 1 (e.g., odds ratio, incidence rate ratio, hazard ratio and other GLMs at 1) to show if the proper reference is outside of the "normal range" or what represents no significant difference. The review plot can be used on assess objects as well as models created from the base package's lm() and glm() as well as coxph() from the survival package. There are various graphing options and sorting by coefficient names, values, p-value, or the model formula.

Usage

```
## S3 method for class 'review'
plot(
  x,
  y = NULL,
  main = NULL,
  sub = NULL,
  sort = NULL,
  decreasing = NULL,
  abbrv = NULL,
  xlim = NULL,
  ylim = NULL,
  xlab = NULL,
  ylab = NULL,
  lwd = NULL,
  color = NULL,
  pcol = NULL,
  pt.cex = NULL,
  pch = NULL,
  tgt = NULL,
  tcol = NULL,
  cex = NULL,
  cex.axis = NULL,
  cex.lab = NULL,
  cex.main = NULL,
  cex.sub = NULL,
  print = NULL,
  round.c = NULL,
  ...
)
```

Arguments

x	review object from assess() model or lm(), glm(), and coxph() models. For assess objects, use review(my_regression$model) unless using ITS or DID models (e.g., my_did_model$DID). For model objects from lm(), glm(), or coxph(), use review(my_model).
y	not currently used.
main	overall title for the plot, default is NULL which then lists 'review' for OLS regression or the type of summary statistic such as 'Odds Ratio' (logistic), 'Incidence Rate Ratio' (Poisson), or 'Hazard Ratio' (Cox).
sub	a character vector for the subtitle of the plot, default is NULL which then lists the outcome name.
sort	specify how confidence intervals are sorted. Options are 'alpha' (alphabetical coefficient names), 'coef' (coefficient values), 'p' (p-values), or 'enter' (how variables were entered in the model). Default is 'enter'. See decreasing argument to adjust how the selection is then sorted.
decreasing	logical TRUE or FALSE that indicates how confidence intervals are sorted, in decreasing order or not. Defaults to FALSE.
abbrv	the minimum length of the character length for the coefficient name abbreviations. Default is 7.
xlim	specify plot's x-axis limits using a numeric vector with length equal to 2.
ylim	specify plot's y-axis limits using a numeric vector with length equal to 2.
xlab	a vector label for the x-axis. Default is a blank space. Caution: It will overlap the 'sub' argument if sub != "".
ylab	a vector label for the y-axis. Default is a blank space.
lwd	select the line width. Default is 1.
color	the color to be used for the horizontal lines. Default is 'blue'.
pcol	select the point color for the point estimates. Default is 'red'.
pt.cex	the cex to be applied to plotting symbols, default is 1.
pch	the plotting character or symbol to be used, default is 17.
tgt	specify a value on the x-axis of where to add a target line to indicate when confidence intervals fall outside of a normal range or value that indicates no difference.
tcol	select a color for the vertical target line. Default is 'black'.
cex	a numerical value giving the amount by which plotting text and symbols should be magnified relative to the default of 1.
cex.axis	the magnification to be used for axis annotation relative to the current setting of cex. Default is 1.
cex.lab	the magnification to be used for x and y labels relative to the current setting of cex. Default is 1.
cex.main	the magnification to be used for main titles relative to the current setting of cex. Default is 1.

cex.sub	the magnification to be used for subtitles relative to the current setting of cex. Default is 1.
print	logical TRUE or FALSE that indicates whether to print point estimates and confidence intervals. Defaults to FALSE.
round.c	an integer indicating the number of decimal places to be used for rounding values in the printed point estimates and confidence intervals. Default is 4.
...	additional arguments.

Value

plot of the review of regression coefficients, the display can be modified in various ways.

References

Harrell, F. E., Jr. (2015). Regression Modeling Strategies, Second Edition. Springer International Publishing. ISBN: 978-3-319-19424-0.

See Also

review() for the 'review' class object.

Examples

```
# OLS regression--most basic format with Base R lm()
m01 <- lm(mpg ~ wt+hp+am, data=mtcars)
plot(x=review(m01))

# Using the assess function, notice 'm02$model' object below
m02 <- assess(formula=mpg ~ wt+hp+am, data=mtcars, regression="ols")
# let's see the impactful 83.5 hp increase in the 1st to 3rd quartiles
plot(x=review(m02$model, increase=c(hp= 83.5)))

# Logistic model
m03 <- glm(vs ~ wt+hp+am, data=mtcars, family="binomial")
# Display options, sorted plot, and printing the 95% CIs to review
plot(x=review(m03, increase=c(wt=1.1375, hp= 83.5) ), color="aquamarine", lwd=4,
pt.cex= 2, tcol="magenta", sort="coef", cex.axis=2, cex.main=2, xlim=c(-.5, 3),
print=TRUE, round.c=6)

# Poisson model with an offset using assess()
m04 <- assess(formula=HAI ~  Month+ offset(log(PatientDays)),
data=infections, regression="poisson")
# Because 12 months is a meaningful period in program evaluation, Month=12
plot(x=review(m04$model, increase=c(Month=12)), xlim=c(0.6, 1.01),
lwd=7, color="cyan", pcol="salmon", pt.cex=2)

# NOT RUN: Cox Proportional Hazards model
#library(survival)
#m04 <- coxph(Surv(time, status) ~ age+sex+ph.karno, data=cancer)
#plot(x=review(m04, increase=c(age=13, ph.karno=15)))
```

```
# This also works for ITS and DID causal models through ham
im22 <- assess(formula=los ~ ., data=hosprog, intervention = "program",
int.time="month", interrupt= c(5, 9), its="two")
# The intervention group had the biggest change between the baseline and month 5
plot(review(im22$ITS), sort="coef", decreasing=TRUE, color="red", pcol="green")
```

`print.alpha` *Print alpha results*

Description

Formats alpha class results to display summary statistics of scale information. These include the overall alpha, scale mean and standard deviation, item statistics, scale statistics if an item is removed from the scale, and the total sample size.

Usage

```
## S3 method for class 'alpha'
print(x, ...)
```

Arguments

`x`	alpha object from Cronbach's alpha calculation.
`...`	Additional arguments.

Value

formatted alpha results.

See Also

`alpha()` for the 'alpha' class object.

Examples

```
print(alpha(items=c("i1","i2","i3","i4","i5"), data=cas))
```

`print.interpret` *Print interpret object*

Description

Formats interpretations from interpret class objects. Provides simple interpretations of regression coefficients and Cronbach's alpha. Print specific model interpretations (or run all), returned in sentence and paragraph formats.

Usage

```
## S3 method for class 'interpret'
print(x, ...)
```

Arguments

x	interpret object.
...	Additional arguments.

Value

formatted interpret object results.

See Also

`interpret()` for the 'interpret' class object.

Examples

```
#Cronbach's alpha
print(interpret(alpha(items=c("i1","i2","i3","i4","i5"), data=cas)))

#' # interpret a standard linear (OLS) regression
hos1 <- assess(formula=survey ~ program + month, data=hosprog, regression= "ols")
print(interpret(hos1)$model)

# interpret a differences-in-differences model
hos2 <- assess(formula=survey ~ ., data=hosprog, intervention = "program",
int.time="month", treatment = 5, did="two", newdata=TRUE)
interpret(hos2)$did

# interpret an interrupted time series model
hos3 <- assess(formula=survey ~ ., data=hosprog, intervention = "program",
int.time="month", its="two", interrupt = 5)
interpret(hos3)$its

# interpret Bayesian MCMC diagnostics
blos1 <- Bayes(losmcmc, y="Dx", parameter="muOfY")
interpret(blos1$Diagnostics, digits=5)
```

print.review *Print Model Review Results*

Description

Formats 'review' class results to display summary statistics. These include the coefficient's point estimate, lower and upper confidence intervals, and the p-value. Excludes intercept coefficient as does review().

Usage

```
## S3 method for class 'review'
print(x, digits = NULL, ...)
```

Arguments

x	review class object of a regression model.
digits	minimal number of significant digits. Defaults to 3.
...	Additional arguments.

Value

formatted review results in a table.

See Also

review() for the 'review' class object.

Examples

```
# OLS regression
print(review(assess(mpg ~ wt+hp+am, data=mtcars, regression="ols")$model) )
```

review *Creates a review object of regression model results for graphing purposes*

Description

Produces a review object of a regression model's coefficients similar to base R's summary(). However, the purpose is to later pass the object to the plot function to produce a summary of only the predictor variable's coefficients (i.e.,. excludes the intercept). The main difference between the two is that the ham function review can create the effects from increased predictor values (e.g., increase age by 25 years rather than report a 1 year effect).

Usage

```
review(model, increase = NULL, coefs = NULL)
```

Arguments

model	an assess class object or models with 'lm', 'glm', or 'coxph' class. This includes the differences-in-differences and interrupted time series models. For assess objects, make sure to select the correct model list element, for example, model= my_regression$model for ordinary least squares, logistic, or Poisson models. For 'ITS' or 'DID' models, use model= its_model$ITS or model= did_model$DID. If using Base R's lm(), glm() or even coxph(), use the model object name (e.g., model= my_glm_model).
increase	a named numeric vector object of the coefficient name and associated increase in the predictor. For example, in the model y ~ a + x1 + x2, increase= c(x1= 10), will produce point estimates and confidence intervals adjusted to a 10-unit increase in the variable x1. This is most helpful when the continuous predictor has many levels and the interpretation of the standard 1-unit increase is less informative than a larger change in X. For example, discussing a difference in 10 or 20 years may be more helpful in describing the impact on 30-day mortality.
coefs	an expression defining a subset of the predictor coefficient rows (i.e., not the intercept) to view in the review plot. The default is NULL, thereby using all predictor coefficients. Specify, for example, 1:2 to view just the summary of the first 2 coefficients.

Value

a list object with regression model summary statistics such as the point estimates, lower and upper confidence interval limits, p-values, and additional information.

See Also

plot.review(), print.review() for a plot and a formatted print output of the 'review' class object.

Examples

```
# OLS regression
review(assess(mpg ~ hp + wt + cyl, data=mtcars, regression= "ols")$model)

# Creates new classes useful for plots, e.g., plot(review(model))
class(review(assess(mpg ~ hp + wt + cyl, data=mtcars, regression= "ols")$model))

# Works with Base R
review(lm(mpg ~ hp + wt + cyl, data=mtcars))

# Effect after increasing hp and excluding am's coefficient
m03 <- glm(vs ~ wt+hp+am, data=mtcars, family="binomial")
plot(x=review(m03, increase=c(wt=1.1375, hp= 83.5), coefs= c(1, 3) ))
```

`unemployment` *USA's unemployment rate between 1929 and 2024*

Description

USA's unemployment rate between 1929 and 2024

Usage

```
unemployment
```

Format

`unemployment`**:**

An artificial data frame with 96 rows and 5 columns:

Year Calendar year as an integer from 1929 to 2024.

rate Unemployment rate defined as the proportion of the US Labor Market that were unemployed.

event Notable events in the United States that may have impacted the unemployment rate. Other than notable events (or unspecified) listed as 'other'.

year Integer for the number of years for 1929 to 2024, ranging from 1 to 96.

usa Value of 1 to indicate data for the USA. ...

Source

unemployment is an artificial data frame of reasonable estimates made entirely for the purpose of demonstrating interrupted time series with many interruptions. For precise rates, please see a reliable source.

Index

∗ **datasets**
  cas, 12
  co2mcmc, 12
  co2multi, 13
  compapt, 14
  covshort, 16
  hosp1, 20
  hosprog, 21
  infections, 23
  losmcmc, 26
  losvary, 26
  NHSN, 27
  unemployment, 58

alpha, 3
alpha(), *54*
assess, 4
assess(), *25*, *31*

Bayes, 8
Bayes(), *39*

cas, 12
co2mcmc, 12
co2multi, 13
compapt, 14
control, 15
control(), *43*
covshort, 16

decide, 17
decide(), *45*

group, 18
group(), *49*

hosp1, 20
hosprog, 21

importance, 22
importance(), *50*
infections, 23
interpret, 23
interpret(), *3*, *7*, *11*, *25*, *55*
itsEffect, 24
itsEffect(), *24*

losmcmc, 26
losvary, 26

NHSN, 27

plot.assess, 28
plot.assess(), *7*
plot.Bayes, 32
plot.Bayes(), *11*
plot.control, 41
plot.control(), *16*
plot.decide, 44
plot.decide(), *18*
plot.group, 46
plot.group(), *20*
plot.importance, 49
plot.importance(), *22*
plot.review, 51
plot.review(), *57*
print.alpha, 54
print.alpha(), *3*
print.interpret, 55
print.interpret(), *24*
print.review, 56
print.review(), *57*

review, 56
review(), *53*, *56*

unemployment, 58