Package 'DrewDayRFunctions'

August 15, 2024

Title R Functions for Epidemiologic Applications by Drew Day

Version 1.0.1

Author Drew Day [aut, cre]

Maintainer Drew Day <dday612@gmail.com>

Description Includes a variety of useful functions for rapidly performing regressions, performing sample size calculations, visualizing results, and other epidemiology-related applications collated over the career of Drew Day, an environmental epidemiologist.

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports mgcv, mgcViz, dplyr, sandwich, lmtest, ggplot2, caret, splines, emmeans, logistf, mice, SimMultiCorrData, DHARMa, pbapply, cowplot, scales, msm, fastDummies, rcompanion, DT, kableExtra, htmltools

Suggests rmarkdown, knitr, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Contents

GAMResults *Run multiple GAMs and return results*

Description

GAMResults runs GAM models with smooths on a series of predictor variables of interest for all combinations of predictor and outcome variables, allowing for categorical and continuous interaction terms. A blog post with some nice visualizations and explanations of what GAMs are can be found here.

Usage

```
GAMResults(
  prednames,
  outnames,
  covnames = NULL,
  Data,
  logout = F,
  logpred = F,
  Outtitle = "Outcome",
  Predtitle = "Exposure",
  ixterm = NULL,
  smooth.select = F,
  bs_s = "'tp'",
  K_s = -1,
  bs_ti = "'cr'",
  K_ti = NA,
  na.action = na.exclude,
  plotresid = F,
  quantiletrimcheck = F,
  trimdim = c("x", "y"),
  trimperc = 5
)
```

Arguments

prednames	A character vector of the predictor variables.
outnames	A character vector of the outcome variables. The function currently allows for numeric or binary ("factor" or "character" class) outcome variables.
covnames	A character vector of all covariates or a list of character vectors of covariates of the same length as `length(prednames) * length(outnames)` so that each unique combination of prednames and outnames has a defined set of covariates. Note that in ordering this list the function loops through the `prednames` and then the `outnames` (e.g., Outcome 1 - Predictor 1, Outcome 1 - Predictor 2, Outcome 1 - Predictor 3, Outcome 2 - Predictor 1, etc.), and so you should order the list of covariates accordingly. These covariates will be included as linear

(i.e., not smooth) terms in the model. These can include spline terms, such as ns natural splines or bs b-splines as defined by the 'splines' package (e.g., "ns(Year, df=3)"). These can also include interaction terms denoted by the ":" separator , though you should make sure to include the main effects too (e.g., c("Income","HouseholdSize", "Income:HouseholdSize")).

Data — A data frame containing all prednames, outnames, and covnames as columns.

logout — A TRUE/FALSE value for whether to natural log-transform each outcome variable. Defaults to FALSE.

logpred — A TRUE/FALSE value for whether to natural log-transform each predictor variable. Defaults to FALSE.

Outtitle — A single character value for the name of the column containing outcome variables in the data frame of results to be returned. Defaults to "Outcome".

Predtitle — A single character value for the name of the column containing predictor variables in the data frame of results to be returned. Defaults to "Exposure".

ixterm — A single character value for the name of an interacting variable. This variable can be either categorical, in which case the column should be of a character or factor class, or it can be continuous.

smooth.select — A TRUE/FALSE value for whether to add an extra penalty to each smooth term so that it can be penalized to zero. Defaults to FALSE, and this corresponds to the 'select' parameter in the gam function.

bs_s — A character value wrapped in single quotes reflecting the basis spline to be used in the s smooths, which will be used as the smooth terms in all models except if a continuous * continous interaction is included (i.e., if ixterm is a continuous variable, in which case ti smooths are used. Defaults to "'tp'", indicating thin-plate regression splines (TPRS) will be used for the basis spline (see ?tprs). This is also the default basis spline for s.

K_s — An integer reflecting the k parameter for s smooth terms. The meaning of this term is that the maximum effective degrees of freedom (EDF) of each smooth is this value minus one (see ?choose.k). This defaults to -1, which is the default value for s. Setting K_s to -1 is the equivalent of not specifying a value, in which case the 'k' value default value for s is 10 (i.e., maximum EDF=9) when there is a single term in the smooth (e.g., s(X1)) and bs="tp". For your information, the default k when using two-term smooths (e.g., s(X1, X2)) for s when bs="tp" is 30 and for >=3-term smooths this is 110 (see ?tprs). Note that increasing this value increases the subspace of functions (can be thought of as a ceiling on possible EDF values), and therefore higher K values can lead to higher EDF (see ?choose.k).

bs_ti — A character value wrapped in single quotes reflecting the basis spline to be used in the ti tensor product interaction smooths, which will be used only if a continuous * continous interaction is included (i.e., if ixterm is a continuous variable). Defaults to "'cr'", indicating cubic regression splines will be used for the basis spline (see ?smooth.construct.cr.smooth.spec).

K_ti — An integer reflecting the k parameter for ti. This defaults to NA, which is also the default k for ti, and which means default k values will be used. The documentation shown with ?smooth.construct.cr.smooth.spec suggests that the

default K is 10, but in practice it appears that the default k is actually 5 based on the bsdim values output in my own experiments. This means 4 knots in the univariate ti terms and 4 each for the bivariate ti(var1, var2) terms.

na.action — This is the na.action that will be passed to gam. Defaults to na.exclude.

plotresid — A TRUE/FALSE value indicating whether the GAM plots showing the smooth fit and CIs should also include points showing the model residuals on the plot (see l_points). Defaults to FALSE.

quantiletrimcheck — A TRUE/FALSE value indicating whether a certain quantile of one or both of the predictor and/or outcome variables should be trimmed as a sensitivity analysis. Defaults to FALSE. If TRUE, the data will be trimmed at the top trimperc percentiles. The purpose of this check is to make sure that the observed association curves don't change radically when removing edge points considering that splines can be at times sensitive to data at the edges of the distribution.

trimdim — A character value or vector of "x", "y", or c("x", "y") indicating whether to trim the predictor (x) or outcome (y) values when quantiletrimcheck is TRUE. Defaults to c("x", "y").

trimperc — An integer or numeric value indicating total percentage of the data to be trimmed from the high and low ends of the data distribution if quantiletrimcheck is TRUE. Defaults to 5, meaning that everything below the 2.5th percentile and above the 97.5th percentile will be removed before running the GAM models.

Details

GAMResults bases its models on the assumption that there will be only one univariate smooth term in each model for the predictor of interest and that utilizes the smooth function s with a user-defined basis spline and "k" parameter. This is akin to gam(y ~ s(x, bs = bs_s, k = K_s) + z, ...) where y is an outcome variable, x is a predictor variable of interest, and z is a covariate. If an outcome variable is binary, a logistic GAM is run.

In the case of a continuous interacting variable (i.e., ixterm), this function utilizes ti for univariate tensor product interaction terms for both the predictor of interest and interaction term (similar to main effects in a linear model with a multiplicative interaction term), as well as a bivariate tensor product interaction for those two variables. This is the same as gam(y ~ ti(x, bs = bs_ti, k = K_ti) + ti(iota, bs = bs_ti,k = K_ti) + ti(x, iota, bs = bs_ti, k = K_ti) + z, ...) where iota is the interacting variable. This is implemented as it is in the ModelArray R package and as succinctly described on this linguistics professor's blog post.

In the case of a categorical interacting variable (i.e., ixterm), two separate models are run to obtain all the relevant parameter estimates. The first model is structured as gam(y ~ s(x, bs = bs_s, k = K_s, by = iota) + iota + z), while the second is gam(y ~ s(x, bs = bs_s, k = K_s, by = ordered(iota)) + s(x, bs = bs_s, k = K_s) + iota + z). From the first model we are able to obtain the factor level-specific curves, and from the second model we can obtain curves of the differences between those level-specific curves. Note that the curve for the referent level produced by the second term "s(x)" in the second model should theoretically be equivalent to the referent level curve produced by the first term in the first model, but in practice they tend to have minuscule differences in EDF, exact fitted curve values, etc., but it's close enough I find it convenient to grab terms from these two models as if they were equivalent. Further explanation of both models can be

found at this CrossValidated blog post. Examples of the second model are shown in this blog post and in example papers like Zhernakova et al. 2022.

Value

GAMResults returns a list of:

Matrix	A data frame of important results from the GAM models.
Plots	A list of ggplot plots of class c("plotSmooth", "gg") showing the predictor of interest curves plotted against predicted outcome values. These plots are based on the mgcViz package plotting functions l_ciPoly, l_fitLine, and l_rug.
GAMlist	A list of all GAM models stored as objects of class "gam".
IxGAMlist	If ixterm is not NULL and is a categorical variable, the additional GAM models that were run to calculate the interaction term coefficients are stored here as a list of "gam" class objects.

Examples

```
# The first example involves analyzing several GAMs with no interaction terms.

set.seed(100)
gamdat1 <- data.frame(X1 = rnorm(500), X2 = rnorm(500), Z = rnorm(500,0,2))

gamdat1$yhat1 <- exp(sin(gamdat1$X1)^0.5) + gamdat1$Z * 0.75 - 2
gamdat1$yhat2 <- exp(cos(gamdat1$X2)^2) + gamdat1$Z * 2 + 3

set.seed(102)
gamdat1$y1 <- gamdat1$yhat1 + rnorm(500)
set.seed(103)
gamdat1$y2 <- gamdat1$yhat2 + rnorm(500)

gamres1 <- GAMResults(prednames = c("X1", "X2"), outnames = c("y1", "y2"),
                      covnames = "Z", Data = gamdat1)

mgcViz::gridPrint(grobs = gamres1$Plots, ncol = 2)

# This next example involves a factor-smooth interaction

set.seed(100)
gamdat2 <- data.frame(X = rnorm(500), Z = rnorm(500,0,2),
                      S = as.factor(sample(c("M","F"), 500, replace = T)))
gamdat2$yhat <- ifelse(gamdat2$S == "M", sin(gamdat2$X) * 3,
                       sin(gamdat2$X) * 0.5)
gamdat2$yhat <- gamdat2$yhat + gamdat2$Z * 2
set.seed(101)
gamdat2$y <- gamdat2$yhat + rnorm(500)

gamres2 <- GAMResults("X", "y", "Z", gamdat2, ixterm = "S")

mgcViz::gridPrint(grobs = gamres2$Plots$`y~X`)
```

GenericDataSummary *Generate an html R Markdown document briefly summarizing a dataset*

Description

GenericDataSummary creates a Markdown document for any dataset that separately summarizes continuous variables in that dataset with a summary table and boxplots and provides frequency counts for all categorical data. There is also an option to include a data dictionary table in this simple summary document. Uses for this function include quickly inspecting a dataset for aberrant values and getting a sense for which variables are included.

Usage

```
GenericDataSummary(filename = NULL, directory = getwd(), Data, Dict = NULL)
```

Arguments

filename	A file name and path for the .html R Markdown document to be generated. Defaults to "DataSummary_X.html" where X is the system date and time when the document is saved. You can specify the ".html" suffix here or omit it, in which case it will be added for you.
directory	The folder in which you'd like to save the html document. This defaults to your current working directory.
Data	A data frame to be summarized.
Dict	An optional data frame for the data dictionary to be included for reference as a table in the Markdown document. This defaults to NULL.

Details

This Markdown document makes use of datatable from the DT package for the data dictionary and continuous summary statistics table, which can be sorted and searched. Categorical variables are printed out in a scrollable html table made using the kable function from the kableExtra package.

Value

GenericDataSummary saves an html document to the specified filename and directory.

Examples

```
# Make a data dictionary for the built-in iris dataset
irisdd <- data.frame(Variable = names(iris),
Description = gsub("\\.", " ", names(iris)), Levels = "")
irisdd[irisdd$Variable == "Species", "Levels"] <-
paste(unique(iris$Species), collapse = ", ")

# create the summary document
```

```
GenericDataSummary("NewIrisSummary", getwd(), iris, irisdd)
```

GLMResults *Run multiple linear regressions and return results*

Description

GLMResults is a complicated helper function I've piecemeal added to throughout the years to handle all sorts of repeated linear regression analyses. It outputs a table of key regression output including optional diagnostic test values, a list of the model objects, and a coefficient forest plot. Supported regression types include linear regression (lm) and binomial GLMs (glm with binomial family and link "logit" for logistic regressions or Firth's logistic regression using logistf, "log" for binomial regressions, or "probit" for probit regressions, in addition to robust Poisson regressions for estimating risk ratios for binary outcomes where the family is poisson, the link function is "log", and robust errors are used).

Usage

```
GLMResults(
  prednames,
  outnames,
  covnames = NULL,
  Data,
  logout = F,
  logpred = F,
  logbasepred = 10,
  logbaseout = exp(1),
  Outtitle = "Outcome",
  Predtitle = "Exposure",
  MyMult = 1,
  ixterm = NULL,
  Firth = F,
  altprednames = NULL,
  altoutnames = NULL,
  altixname = NULL,
  binomlink = "logit",
  binfunc = binomial,
  robust = T,
  robustdf = F,
  HCtype = "HC0",
  predspline = F,
  predsplinedf = 3,
  plotExpBeta = F,
  plotPercChange = F,
  LOOCV = F,
  facetcol = NULL,
```

```
  covix = NULL,
  ixpred = T,
  extradiag = T,
  leverage.test = F,
  leverage.cutoff = 0.2,
  leverage.meancutoff = NULL,
  post.power = F,
  effect.size = 0.5,
  nsim = 1000,
  mice = F,
  micevars = NULL,
  miceiter = 10
)
```

Arguments

prednames	A character vector of the column names of predictors of interest you'd like to evaluate in separate regressions. These column names can refer to character, factor, or numeric columns in the data frame.
outnames	A character vector of the column names of outcome variables you'd like to evaluate in separate regressions. These column names can refer to character, factor, or numeric columns in the data frame. For each outnames, lm() linear regressions will be run if the referenced column is numeric, or a glm() GLM with a binomial() family will be run with the link function specified in the input binomlink. This does not yet support characters or factors with >2 unique values (i.e., it can only accommodate continuous or binary outcome variables).
covnames	Either a character vector of column names for all the covariates you'd like to include in each regression or a list of character vectors equal in length to the length of the unique combination of prednames and outnames. Defaults to NULL, meaning no covariates are included. This can be a list in order to control for separate covariates for each unique combination of prednames and outnames, in which case the list must be of length length(prednames) * length(outnames). Note that in ordering this list the function loops through the prednames and then the outnames (e.g., Outcome 1 - Predictor 1, Outcome 1 - Predictor 2, Outcome 1 - Predictor 3, Outcome 2 - Predictor 1, etc.), and so you should order the list of covariates accordingly. These can include spline terms, such as ns natural splines or bs b-splines as defined by the 'splines' package (e.g., "ns(Year, df=3)"). These can also include interaction terms denoted by the ":" separator , though you should make sure to include the main effects too (e.g., c("Income","HouseholdSize", "Income:HouseholdSize")).
Data	A data frame object of class "data.frame".
logout	If TRUE, this will log-transform each outcome variable (the log base will be logbaseout) prior to running the regression and then return both the raw coefficients as well as converted coefficients to percent changes using the formula percent change = (exp(raw coefficient)-1) x 100. Can be either a single value or a logical vector of length equal to the length of outnames. Defaults to FALSE.
logpred	If TRUE, this will log-transform each predictor variable (the log base will be logbasepred) prior to running the regression and then return the raw coeffi-

cients. Can be either a single value or a logical vector of length equal to the length of prednames. Defaults to FALSE.

logbasepred — The base for the log-transformation of one or more of the predictor variables. Defaults to 10.

logbaseout — The base for the log transformation of the outcome variables. Defaults to exp(1) (i.e., natural log transformation).

Outtitle — Defines the title used for the column of outcome variables in the table and on the plot. Defaults to "Outcome".

Predtitle — Defines the title used for the column of predictor variables in the table and on the plot. Defaults to "Exposure".

MyMult — The table will output raw and transformed coefficients and CIs. This multiplier is an optional scalar to multiply the raw coefficients for an estimate of the coefficient for something other than a 1-unit increase. Defaults to 1, and can be either a number or the character "IQR", in which case the raw coefficient will be multiplied by the interquartile range of the predictor variable.

ixterm — A column name for an optional interaction term for the prednames predictors of interest (I abbreviate interaction as 'ix'). Can refer to either a character, factor, or numeric-class column in the Data data frame. All relevant interaction and main coefficients will be output. Defaults to NULL.

Firth — If TRUE and if a given outnames variable is binary, this will perform a Firth's logistic regression (logistf) instead of a logistic regression (glm).Defaults to FALSE.

altprednames — This is a vector of character values for alternate names for the predictor variable column names specified in prednames to replace those column names in the output tables and plots. Defaults to NULL.

altoutnames — This is a vector of character values for alternate names for the outcome variable column names specified in outnames to replace those column names in the output tables and plots. Defaults to NULL.

altixname — This is a character value for an alternate name for the interaction variable column name specified in ixterm to replace that column name in the output tables and plots. Defaults to NULL.

binomlink — The link function to use for the family specified in binfunc for GLMs of binary outnames. Can be "logit", "log", or "probit". Defaults to "logit".

binfunc — The family to use when outnames is binary. Can be binomial, poisson, or quasibinomial. Defaults to binomial. If the Poisson family is used, one should make sure to set binomlink to "log" and robust to TRUE.

robust — If TRUE, this calculates and returns robust CIs and p-values based on the vcovHC function instead of the default CIs and p-values. Defaults to TRUE.

robustdf — If FALSE, this sets coefci to calculate z-score-based robust confidence intervals if robust is TRUE. If TRUE, t test-based robust confidence intervals are instead calculated based on the residual degrees of freedom in each regression if robust is TRUE. Defaults to FALSE, which is the default for the df parameter in coefci.

HCtype — This is the formula used for the vcovHC function calculation of robust CIs and p-values. Defaults to "HC0".

predsplinedf This specifies the number of degrees of freedom for the predictor variable natural splines if predspline is TRUE. Defaults to 3.

plotExpBeta If TRUE, this will plot exp(coefficient) values) in the coefficient forest plot. Defaults to FALSE.

plotPercChange If TRUE, this will plot percent changes (i.e., it will plot (exp(coefficient)-1) x 100 values) in the coefficient forest plot. Defaults to FALSE. If logistic regressions were run and both plotExpBeta and plotPercChange are FALSE, raw coefficients will be plotted.

LOOCV If TRUE, this performs leave one out cross validation using the 'caret' package. If a given model is a linear regression (lm), the RMSE will be reported in the tables. If a model is a GLM (glm), the accuracy will be reported in the tables. The train function input method is trainControl(method = "repeatedcv", number = 10, repeats = 50). Defaults to FALSE.

facetcol This defines the number of facet_wrap columns for the coefficient plots (facetted by outcome variable). If NULL, this will equal the number of unique outnames values. Defaults to NULL.

covix This character vector of covariate column names allows for covariates to also have interaction terms with the ixterm interaction variable. If one wishes to have all covariates have interaction terms with ixterm, the character vector for covix should be the same as that for covnames. Defaults to NULL, meaning that there are no interactions with any of the covariates.

ixpred If TRUE, an interaction term is included between each of the prednames predictors of interest and ixterm. This can be set to FALSE to allow for interactions only with the covariates defined in covix. If covix is not NULL and ixpred is TRUE, there will be interaction terms with both the predictor of interest and the covariates. Defaults to TRUE.

extradiag If TRUE, this will include extra diagnostic information in the tables. For linear regressions (lm), this will include a p-value for the heteroskedasticity of model residuals (bptest), the number of Cook's distances > 0.5, and the maximum Cook's distance, the latter two of which define high leverage points. For GLMs (glm), this will include output from DHARMa package tests, including a heteroskedasticity of residuals p-value from testQuantiles, the minimum p-value from testQuantiles, the number of Cook's distances > 0.5, the maximum Cook's distance, a p-value for a test of uniformity (testUniformity), a p-value for outliers (testOutliers), and a p-value for dispersion (testDispersion). Defaults to TRUE.

leverage.test If TRUE, this runs all regressions while excluding "high leverage" observations as defined by the user. Defaults to FALSE.

leverage.cutoff
This is the Cook's distance cutoff above which observations will be excluded if leverage.test is TRUE. Defaults to 0.2.

leverage.meancutoff
If leverage.cutoff is set to NULL, this is used as a multiplier by the mean Cook's distance to get a new cutoff above which observations will be excluded if leverage.test is TRUE. For example, if set to 2, the cutoff will be set to the mean Cook's distance times 2. If both leverage.cutoff and leverage.meancutoff

are set to NULL, the cutoff will be set to the mean Cook's distance times 4. Defaults to NULL.

post.power — If TRUE, this repeatedly simulates data with similar structure to the actual data using rcorrvar and then uses a user-specified predictor of interest coefficient to calculate a simulated outcome variable (all covariate coefficients are drawn from the original model estimates), runs regressions on these simulated data frames, collects the iterated predictor of interest coefficient p-values, and calculates how many of those p-values are <0.05. This is a form of posterior power test for different effect sizes of the predictor of interest association with the outcome variable. Defaults to FALSE.

effect.size — This is the user-specified effect size for which the post-hoc power is tested if post.power is set to TRUE. Defaults to 0.5.

nsim — This is the number of simulations for the post-hoc power test if post.power is set to TRUE. Defaults to 1E3.

mice — If TRUE, this performs multiple imputation (MICE using mice) on the user-specified variables and pools model estimates across the imputed datasets. Defaults to FALSE.

micevars — A character vector of the names of variables to be imputed using MICE. Defaults to NULL, in which case it will include all predictor, outcome, and covariate variables that have missing values.

miceiter — The number of multiple imputations to perform if mice is set to TRUE. Defaults to 10.

predspline: — If TRUE, this transforms each predictor of interest into a natural spline term using ns from the ' splines' package. A coefficient for each degree of freedom is returned. Defaults to FALSE.

Details

GLMResults was built over several years and so has incorporated all sorts of functionality related to linear regressions that I've found useful. This is my main workhorse function for most analyses calling for regressions.

Value

GLMResults returns a list of:

Matrix — A table of all the key variables. Beta, LCI, and UCI refer to the coefficient for the predictor of interest and the lower and upper 95% confidence intervals, respectively. Beta.Transform, etc. are those estimates transformed in some way as specified by the user, for instance by being multiplied by the IQR, exponentiated or transformed into percent changes (done by default when the outcome is binary), or some other multiplication specified by the user.

GGplot — A forest ggplot of all predictor of interest coefficients and CIs.

LMlist — A list of all the lm, glm, or logistf objects run for each unique combination of predictor and outcome variable names (i.e., prednames and outnames).

Examples

```
# OLS linear regression with a simple set of outcomes, predictors, and
#  covariates without using robust errors

data("mtcars")
carpreds <- c("disp", "cyl", "drat", "wt")
carouts <- c("mpg", "hp")
carcovars <- c("am", "vs", "gear")

lmres <- GLMResults(prednames = carpreds, outnames = carouts,
                    covnames = carcovars, Data = mtcars, robust = F)

# Probit regression with an interaction term on the predictor of interest and
#  on the covariate "gear" while also using robust errors (the default)

glmres <- GLMResults(carpreds, c("vs", "am"), c("gear", "carb"), mtcars,
                     ixterm = "qsec", covix = "gear", binomlink = "probit",
                     altprednames = c("Displacement", "Cylinders", "Axle Ratio",
                     "Weight"), altoutnames = c("Engine Type", "Transmission Type"),
                     altixname = "Quarter Mile Time")

# Robust Poisson regression
glmres <- GLMResults(carpreds[-1], c("vs", "am"), c("gear", "carb"), mtcars,
                     binomlink = "log", binfunc = poisson,
                     altprednames = c("Displacement", "Cylinders", "Axle Ratio",
                     "Weight"), altoutnames = c("Engine Type", "Transmission Type"))

# MICE example
# note that robust errors are not attainable when using MICE, so I use robust = F

mtcars_miss <- mtcars
set.seed(101)
mtcars_miss[sample(1:nrow(mtcars_miss), 3), "cyl"] <- NA
set.seed(104)
mtcars_miss[sample(1:nrow(mtcars_miss), 4), "wt"] <- NA

lmres <- GLMResults(prednames = carpreds, outnames = carouts,
                    covnames = carcovars, Data = mtcars, mice = T,
                    micevars = c("cyl", "wt"), robust = F)
```

`InteractionCoefPlot` *Visualize continuous by continuous interactions*

Description

`InteractionCoefPlot` produces a plot of how the coefficient between one variable in a bivariate continuous interaction and the outcome changes over levels of the other continuous variable. This is a useful visualization of bivariate interactions between two continuous variables.

Usage

```
InteractionCoefPlot(
  model,
  data,
  pred,
  ixterm,
  multiplier = 1,
  coeftransform = NULL,
  predname = NULL,
  ixname = NULL,
  outname = NULL,
  title = NULL,
  fillcolor = NULL,
  autotitle = F,
  addpvallab = F,
  labsize = 4,
  lengthout = 50,
  shadebysig = F,
  otherix = F,
  robust = T,
  logistic = F,
  xlablog10 = F
)
```

Arguments

model	A model of class lm or glm.
data	The data frame used to generate model.
ixterm	The column name of the interacting variable in the data frame data.
multiplier	A multiplier for the coefficients of interest (e.g., an IQR of the predictor of interest). Defaults to 1.
coeftransform	An optional transformation function for the coefficients (e.g., tenfoldperc if the predictor of interest is on a log10 scale where tenfoldperc <- function(x) ((10^x) - 1) * 100). Defaults to NULL.
predname	An optional new name for the variable pred. Defaults to NULL.
ixname	An optional new name for ixterm. Defaults to NULL.
outname	An optional new name for the outcome variable. Defaults to NULL.
title	An optional plot title. Defaults to NULL.
fillcolor	The color for the shading of the 95% CI region. Defaults to "grey50".
autotitle	An option to automatically generate a plot title. Defaults to FALSE.
addpvallab	An option to add a label to the plot that shows the interaction term p-value. Defaults to FALSE.
labsize	The label text size if addpvallab is TRUE. Defaults to 4.
lengthout	The number of levels of ixterm generated with the length.out argument for the seq function, defaults to 50.

shadebysig	An option to shade the CI region areas that don't overlap the null a darker tint of fillcolor. Defaults to FALSE.
otherix	An option to include a second plot in which pred and ixterm are flipped. Defaults to FALSE.
robust	This is an option to use HC0 robust sandwich errors for the CIs (see vcovHC for details on HC estimators). Defaults to TRUE. This should be set to TRUE if model is a robust Poisson model for RR estimation.
logistic	An option for if model is a logistic glm. Defaults to FALSE. This option sets the null line to 1, exponentiates all the coefficients to bring them to an odds ratio scale, and changes the y label to "OR" instead of "Coefficient".
xlablog10	An option to change the x-axis labels to be of the form "10^x". This is useful if your predictor of interest (or also the ixterm variable if otherix is TRUE) is on a logarithmic scale.

Details

Plot axis labels, fonts, etc. can be changed with standard ggplot options. For example, if the model is a robust Poisson GLM, you can set logistic = TRUE when running this function and saving it into an object that in this example I will call plotlist and then change the "OR" label to "RR" by running plotlist$MainGGplot + ylab("Main Variable RR") or something similar.

Value

InteractionCoefPlot returns a list of:

MainGGplot	The interaction plot
OtherGGplot	The other interaction plot if otherix = TRUE
MainMatrix	The matrix of values used to generate the main plot
OtherMatrix	The matrix of values used to generate the other plot if otherix = TRUE

Examples

```
set.seed(1)
exdat <- as.data.frame(mvtnorm::rmvnorm(500, mean=rep(0,5)))
names(exdat) <- paste0("X", 1:ncol(exdat))
exdat$X1X2 <- exdat$X1 * exdat$X2
exdat$yhat <- c(-1 + as.matrix(exdat) %*% c(0.5, 1.5, 0, -2, 2, 1))
set.seed(2)
exdat$y <- exdat$yhat + rnorm(500,0,5)
exlm <- lm(y ~ X1 * X2 + X3 + X4 + X5, exdat)
explot <- InteractionCoefPlot(exlm, exdat, "X1", "X2", addpvallab=T,
shadebysig=T, robust=F, otherix=T)
cowplot::plot_grid(explot$MainGGplot, explot$OtherGGplot, ncol=1)
```

Index