Package 'api2lm'

October 12, 2022

Type Package

Title Functions and Data Sets for the Book ``A Progressive Introduction to Linear Models''

Version 0.1.4

Author Joshua P. French

Maintainer Joshua P. French <joshua.french@ucdenver.edu>

Description Simplifies aspects of linear regression analysis, particularly simultaneous inference. Additionally, supports ``A Progressive Introduction to Linear Models'' by Joshua French (<https://jfrench.github.io/LinearRegression/>).

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

Suggests testthat (>= 3.0.0), ggplot2, knitr

VignetteBuilder knitr

Depends R (>= 3.5.0)

Config/testthat/edition 3

NeedsCompilation no

Repository CRAN

Date/Publication 2022-09-30 07:40:09 UTC

R topics documented:

`autoplot.confint_adjust` *Plot* `confint_adjust` *object*

Description

Plot a `confint_adjust` object produced by the `confint_adjust` function. The plotting function internally calls the `autoplot` function. Note: the `ggplot2` package must be loaded (i.e., `library(ggplot2)` or `ggplot2::autoplot` must be specifically called for this function to work. See Examples.

Usage

```
autoplot.confint_adjust(object, parm, ...)
```

Arguments

`object`	An `confint_adjust` object produced by the `confint_adjust` function.
`parm`	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
`...`	Not used

Value

None.

Author(s)

Joshua French

Examples

```
fit <- lm(100/mpg ~ disp + hp + wt + am, data = mtcars)
# standard intervals
cia <- confint_adjust(fit)
# if ggplot2 package is available
if (require(ggplot2)) {
autoplot(cia)
# select subset of plots
autoplot(cia, parm = c("hp", "disp"))
}
```

coef_matrix *Return coefficient matrix*

Description

coef_matrix returns the coefficients element of the summary function, which is a matrix with columns for the estimated coefficients, their standard error, t-statistic and corresponding (two-sided) p-values.

Usage

```
coef_matrix(object)
```

Arguments

object	an object of class "lm", usually, a result of a call to lm.

Value

A $p \times 4$ matrix with columns for the estimated coefficient, its standard error, t-statistic and corresponding (two-sided) p-value. Aliased coefficients are omitted. The additional class coef_matrix is added for custom printing.

Author(s)

Joshua P. French

Examples

```
## a fitted model
fit <- lm(100/mpg ~ disp + hp + wt + am, data = mtcars)
coef_matrix(fit)
print(coef_matrix(fit), digits = 3)
```

confint_adjust *Adjust confidence intervals for multiple comparisons*

Description

A function to produce adjusted confidence intervals with a family-wise confidence level of at least level for lm objects (not applicable if no adjustment is used). Internally, the function is a slight revision of the code used in the confint.lm function.

Usage

```
confint_adjust(object, parm, level = 0.95, method = "none")
```

Arguments

object	a fitted model object.
parm	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
level	the confidence level required.
method	A character string indicating the type of adjustment to make. The default choice is "none". The other option is "bonferroni".

Details

Let a = 1 - level. Let p be the number of estimated coefficients in the fitted model. All intervals are computed using the formula estimate +/- m * ese, where m is a multiplier and ese is the estimated standard error of the estimate.

method = "none" (no correction) produces the standard t-based confidence intervals with multiplier qt(1 - a/2, df = object$df.residual).

method = "bonferroni" produces Bonferroni-adjusted intervals that use the multiplier m = qt(1 - a/(2 * k), df = object$df.residual), where k is the number of intervals being produced.

method = "wh" produces Working-Hotelling-adjusted intervals that are valid for all linear combinations of the regression coefficients, which uses the multiplier m = sqrt(p * qf(level, df1 = p, df2 = object$df.residual)).

Value

A confint_adjust object, which is simply a a data.frame with columns term, lwr (the lower confidence limit), and upr (the upper confidence limit).

References

Bonferroni, C. (1936). Teoria statistica delle classi e calcolo delle probabilita. Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze, 8, 3-62.

Working, H., & Hotelling, H. (1929). Applications of the theory of error to the interpretation of trends. Journal of the American Statistical Association, 24(165A), 73-85. doi:10.1080/01621459.1929.10506274

Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2004). Applied Linear Statistical Models, 5th edition. New York: McGraw-Hill/Irwin. (p. 230)

See Also

confint.lm

Examples

```
## an extension of the documentation for confint.lm
fit <- lm(100/mpg ~ disp + hp + wt + am, data = mtcars)
# standard intervals
confint_adjust(fit)
# bonferroni-adjusted intervals
```

```
(cib <- confint_adjust(fit, method = "b"))
# plot results
plot(cib)
plot(cib, parm = c("hp", "disp"))
if (require(ggplot2)) {
  autoplot(cib)
  autoplot(cib, parm = c("hp", "disp"))
}
#' working-hotelling-adjusted intervals
(ciwh <- confint_adjust(fit, method = "wh"))
```

`dwaine` *Dwaine Studios data*

Description

Data from the Dwaine Studios data in Applied Linear Statistical Models, 5th edition, p. 237. From the book:

Dwaine Studios, Inc., operates portrait studios in 21 cities of medium size. These studios specialize in portraits of children. The company is considering an expansion into other cities of medium size and wishes to investigates where sales (`sales`) in a community can be predicted from the number of persons aged 16 or younger in the community (`targetpop`) and the per capita disposable personal income in the community (`dpi`). Data on these variables for the most recent year for the 21 cities in which Dwaine Studios is now operating are included in the data set.

Usage

```
data(dwaine)
```

Format

A data frame with 21 observations and 3 variables:

`targetpop` The number of persons aged 16 or younger in thousands of persons.

`dpi` Per capita disposable personal income in thousands of dollars.

`sales` Sales in thousands of dollars

Author(s)

Joshua P. French

References

Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2004). Applied Linear Statistical Models, 5th edition. New York: McGraw-Hill/Irwin.

plot.confint_adjust *Plot* confint_adjust *x*

Description

Plot a confint_adjust x produced by the confint_adjust function. See Examples.

Usage

```
## S3 method for class 'confint_adjust'
plot(x, parm, mar = c(5.1, 7.1, 4.1, 2.1), line = mar[2] - 1, ...)
```

Arguments

x	An confint_adjust x produced by the confint_adjust function.
parm	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
mar	A numerical vector of the form c(bottom, left, top, right) which gives the number of lines of margin to be specified on the four sides of the plot. The default is c(5, 7, 4, 2) + 0.1.
line	The MARgin line, starting at 0 counting outwards, to draw the y-axis label. The default is 1 unit less than mar[2].
...	Additional arguments passed to plot.

Details

The plot function doesn't automatically adjust the margins to account for the label names. If you need more space for your labels, then increase the second element of mar from 7.1 upward and line upward. Alternatively, if you need less space, then you can decrease both of these values. Or you could use the autoplot function that automatically controls the spacing.

Value

None.

Author(s)

Joshua P. French

Examples

```
fit <- lm(100/mpg ~ disp + hp + wt + am, data = mtcars)
# standard intervals
cia <- confint_adjust(fit)
plot(cia)
# plot subset of intervals
```

```
plot(cia, parm = c("hp", "disp"))
# adjust margin and line for better formatting
plot(cia, parm = 2:3, mar = c(5.1, 4.1, 4.1, 2.1))
```

`predict_adjust` *Adjust prediction intervals for multiple comparisons*

Description

A function to produce adjusted confidence/prediction intervals for predicted mean/new responses with a family-wise confidence level of at least `level` for `lm` objects (not applicable if no adjustment is used). Internally, the function is a slight revision of the code used in the `predict.lm` function.

Usage

```
predict_adjust(
  object,
  newdata,
  se.fit = FALSE,
  scale = NULL,
  df = Inf,
  interval = c("none", "confidence", "prediction"),
  level = 0.95,
  type = c("response", "terms"),
  method = "none",
  terms = NULL,
  na.action = stats::na.pass,
  pred.var = res.var/weights,
  weights = 1,
  ...
)
```

Arguments

`object`	Object of class inheriting from `"lm"`
`newdata`	An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.
`se.fit`	A switch indicating if standard errors are required.
`scale`	Scale parameter for std.err. calculation.
`df`	Degrees of freedom for scale.
`interval`	Type of interval calculation. Can be abbreviated.
`level`	Tolerance/confidence level.
`type`	Type of prediction (response or model term). Can be abbreviated.
`method`	A character string indicating the type of adjustment to make. The default choice is `"none"`. The other available options are `"bonferroni"`, `"wh"` (Working-Hotelling), and `"scheffe"`.

terms	If type = "terms", which terms (default is all terms), a character vector.
na.action	function determining what should be done with missing values in newdata. The default is to predict NA.
pred.var	the variance(s) for future observations to be assumed for prediction intervals. See 'Details'.
weights	variance weights for prediction. This can be a numeric vector or a one-sided model formula. In the latter case, it is interpreted as an expression evaluated in newdata.
...	further arguments passed to or from other methods.

Details

Let a = 1 - level. All intervals are computed using the formula prediction +/- m * epesd, where m is a multiplier and epesd is the estimated standard deviation of the prediction error of the estimate.

method = "none" (no correction) produces the standard t-based confidence intervals with multiplier stats::qt(1 - a/2, df = object$df.residual).

method = "bonferroni" produces Bonferroni-adjusted intervals that use the multiplier m = stats::qt(1 -a/(2 * k), df = object$df.residual), where k is the number of intervals being produced.

The Working-Hotelling and Scheffe adjustments are distinct; the Working-Hotelling typically is related to a multiple comparisons adjustment for confidence intervals of the response mean while the Scheffe adjustment is typically related to a multiple comparisons adjustment for prediction intervals for a new response. However, references often uses these names interchangeably, so we use them equivalently in this function.

method = "wh" (Working-Hotelling) or method = "scheffe" and interval = "confidence" produces Working-Hotelling-adjusted intervals that use the multiplier m = sqrt(p * stats::qf(level,df1 = p, df2 = object$df.residual)), where p is the number of estimated coefficients in the model.

method = "wh" (Working-Hotelling) or method = "scheffe" and interval = "prediction" produces Scheffe-adjusted intervals that use the multiplier m = sqrt(k * stats::qf(level,df1 = k, df2 = object$df.residual)), where k is the number of intervals being produced.

Value

predict_adjust produces:

A vector of predictions if interval = "none".

A matrix of predictions and bounds with column names fit, lwr, and upr if interval is set. For type = "terms" this is a matrix with a column per term and may have an attribute "constant".

If se.fit is TRUE, a list with the following components is returned:

- fit: vector or matrix as above
- se.fit: standard error of predicted means
- residual.scale: residual standard deviations
- df: degrees of freedom for residual

References

Bonferroni, C. (1936). Teoria statistica delle classi e calcolo delle probabilita. Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze, 8, 3-62.

Working, H., & Hotelling, H. (1929). Applications of the theory of error to the interpretation of trends. Journal of the American Statistical Association, 24(165A), 73-85. doi:10.1080/01621459.1929.10506274

Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2004). Applied Linear Statistical Models, 5th edition. New York: McGraw-Hill/Irwin.

See Also

predict.lm

Examples

```
fit <- lm(100/mpg ~ disp + hp + wt + am, data = mtcars)
newdata <- as.data.frame(rbind(
               apply(mtcars, 2, mean),
               apply(mtcars, 2, median)))
predict_adjust(fit, newdata = newdata,
               interval = "confidence",
               method = "none")
predict_adjust(fit, newdata = newdata,
               interval = "confidence",
               method = "bonferroni")
predict_adjust(fit, newdata = newdata,
               interval = "confidence",
               method = "wh")
predict_adjust(fit, newdata = newdata,
               interval = "prediction",
               method = "scheffe")
```

---

print.coef_matrix *Print an object of class* coef_matrix *produced by the* coef_matrix *function.*

Description

Print an object of class coef_matrix produced by the coef_matrix function.

Usage

```
## S3 method for class 'coef_matrix'
print(x, digits = 2, ...)
```

Arguments

x	An coef_matrix object produced by the coef_matrix function.
digits	the minimum number of significant digits to be used: see print.default.
...	Additional arguments to the print.data.frame function, such as digits.

Value

A $p \times 4$ matrix with columns for the estimated coefficient, its standard error, t-statistic and corresponding (two-sided) p-value.

Author(s)

Joshua French

Examples

```
fit <- lm(100/mpg ~ disp + hp + wt + am, data = mtcars)
(coefm <- coef_matrix(fit))
# print more digits
print(coefm, digits = 8)
```

---

print.confint_adjust *Print* confint_adjust *object*

Description

Print an object of class confint_adjust produced by the confint_adjust function.

Usage

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

Arguments

x	An confint_adjust object produced by the confint_adjust function.
...	Additional arguments to the print.data.frame function, such as digits.

Value

A data.frame with columns term, lwr, and upr, which are the coefficients for which inference is being made, and the lower and upper bounds of the confidence intervals for each coefficient, respectively.

Author(s)

Joshua French

Examples

```
fit <- lm(100/mpg ~ disp + hp + wt + am, data = mtcars)
(cia <- confint_adjust(fit))
print(cia, digits = 3)
```

print.predict_adjust *Print* predict_adjust *object*

Description

Print an object of class predict_adjust produced by the predict_adjust function.

Usage

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

Arguments

x	An predict_adjust object produced by the predict_adjust function.
...	Additional arguments to the print.default function, such as digits.

Value

Depending on the interval argument of predict_adjust:

A vector of predictions if interval = "none".

A matrix of predictions and bounds with column names fit, lwr, and upr if interval is set. For type = "terms" this is a matrix with a column per term and may have an attribute "constant".

If se.fit is TRUE, a list with the following components is returned:

- fit: vector or matrix as above
- se.fit: standard error of predicted means
- residual.scale: residual standard deviations
- df: degrees of freedom for residual

Author(s)

Joshua French

Examples

```
fit <- lm(100/mpg ~ disp + hp + wt + am, data = mtcars)
(cia <- predict_adjust(fit))
print(cia, digits = 3)
```

`toluca` *Toluca Company data*

Description

Toluca Company data in Applied Linear Statistical Models, 5th edition, p. 19. From the book:

The Toluca Company manufactures refrigeration equipment as well as many replacement parts. In the past, one of the replacement parts has been produced periodically in lots of varying sizes. When a cost improvement program was undertaken, company officials wished to determine the optimum lot size for producing this part. The production of this part involves setting up the production process (which must be done no matter what is the lot size) and machining and assembly operations. One key input for the model to ascertain the optimum lot size was the relationship between lot size and labor hours required to produce the lot. To determine this relationship, data on lot size (`lot_size`) and work hours (`work_hours`) for 25 recent production runs were utilized.

Usage

```
data(toluca)
```

Format

A data frame with 25 observations and 2 variables:

`lot_size` Number of replacement parts produced in the lot.

`work_hours` Number of hours of work required to produce the lot.

Author(s)

Joshua P. French

References

Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2004). Applied Linear Statistical Models, 5th edition. New York: McGraw-Hill/Irwin.

Index