Package 'linearReg'

November 15, 2024

Type Package

Title Linear Regression Analysis Tools

Version 0.1.0

Author Cindy Li

Maintainer Cindy Li <xinjinli@umich.edu>

Description This package offers a tool for conducting linear regression analysis, enabling users to fit models, summarize results, and compute key statistics including confidence intervals for coefficients, R-squared, and F-statistics. Designed to be user-friendly for both educational environments and practical data analysis tasks, linearReg provides an accessible way to perform and interpret linear regression with an emphasis on clarity and ease of use.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

Imports stats

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

fitLinearModel *Linear Regression Function*

Description

Performs linear regression on the provided dataset using the ordinary least squares method. The function calculates coefficients, residuals, various statistics including R-squared, adjusted R-squared, and F-statistics, and returns a comprehensive summary of the model.

Usage

```
fitLinearModel(formula, data)
```

Arguments

formula	An object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	A data.frame that contains the variables in the model.

Value

A list containing model coefficients, standard errors, t-statistics, p-values, sigma squared, R-squared, adjusted R-squared, F-statistic, p-value of the F-statistic, number of observations, and number of predictors.

Examples

```
data(iris)
model = fitLinearModel(Petal.Length ~ Petal.Width + Sepal.Length, iris)
print(model)
```

getAdjustedRSquared *Return the adjusted R Squared value for the model*

Description

Return the adjusted R Squared value for the model

Usage

```
getAdjustedRSquared(model)
```

Arguments

model	The model list object returned by ‘fitLinearModel‘.

Value

Returns adjusted_R_squared Squared for model

Examples

```
data(iris)
model = fitLinearModel(Petal.Length ~ Petal.Width + Sepal.Length, iris)
adjusted_R_squared = getAdjustedRSquared(model)
```

`getConfidenceInterval` *Confidence Interval for Model Coefficients*

Description

Calculates the confidence intervals for the regression coefficients at the specified confidence level.

Usage

```
getConfidenceInterval(model, level = 0.95)
```

Arguments

`model`	The model list object returned by 'fitLinearModel'.
`level`	The confidence level for the interval (default is 0.95).

Value

Returns a data frame with estimates of the coefficients and their lower and upper bounds.

Examples

```
data(iris)
model = fitLinearModel(Petal.Length ~ Petal.Width + Sepal.Length, iris)
ci = getConfidenceInterval(model)
print(ci)
```

`getFStatistic` *Return the F statistics value for the model*

Description

Return the F statistics value for the model

Usage

```
getFStatistic(model)
```

Arguments

`model`	The model list object returned by 'fitLinearModel'.

Value

Returns F statistics for model

Examples

```
data(iris)
model = fitLinearModel(Petal.Length ~ Petal.Width + Sepal.Length, iris)
F_statistic = getFStatistic(model)
print(F_statistic)
```

`getRSquared` *Return the R Squared value for the model*

Description

Return the R Squared value for the model

Usage

```
getRSquared(model)
```

Arguments

`model`	The model list object returned by ‘fitLinearModel‘.

Value

Returns R Squared for model

Examples

```
data(iris)
model = fitLinearModel(Petal.Length ~ Petal.Width + Sepal.Length, iris)
R_squared = getRSquared(model)
print(R_squared)
```

`model_summary` *Model Summary*

Description

Displays a summary of the linear regression model including estimates of the coefficients, standard errors, t-values, p-values for coefficients, significance codes, residual standard error, R-squared, adjusted R-squared, and F-statistic.

Usage

```
model_summary(model)
```

Arguments

model	The model list object returned by ‘fitLinearModel()‘.

Value

Prints the summary table

Examples

```
data(iris)
model = fitLinearModel(Petal.Length ~ Petal.Width + Sepal.Length, iris)
model_summary(model)
```