SimpLin Vignette

Karissa Palmer

2024-01-28

Package Description

`SimpLin` utilizes Rcpp/RcppArmadillo to perform simple linear regression (SLR) on two input vectors, **x** and **y**.

Installation

The package is available for installation via the author's Github account. Users may utilize the below code to install the package.

```
require(devtools)
# install_github()
library(SimpLin)
```

Using the Package

What follows is a short tutorial on how to use the package `SimpLin`. An important note for users to remember is that the package will only take $n \times 1$ vectors as input and the vectors must be of the same length, as well as be numeric. The model that we are trying to fit with this package is always

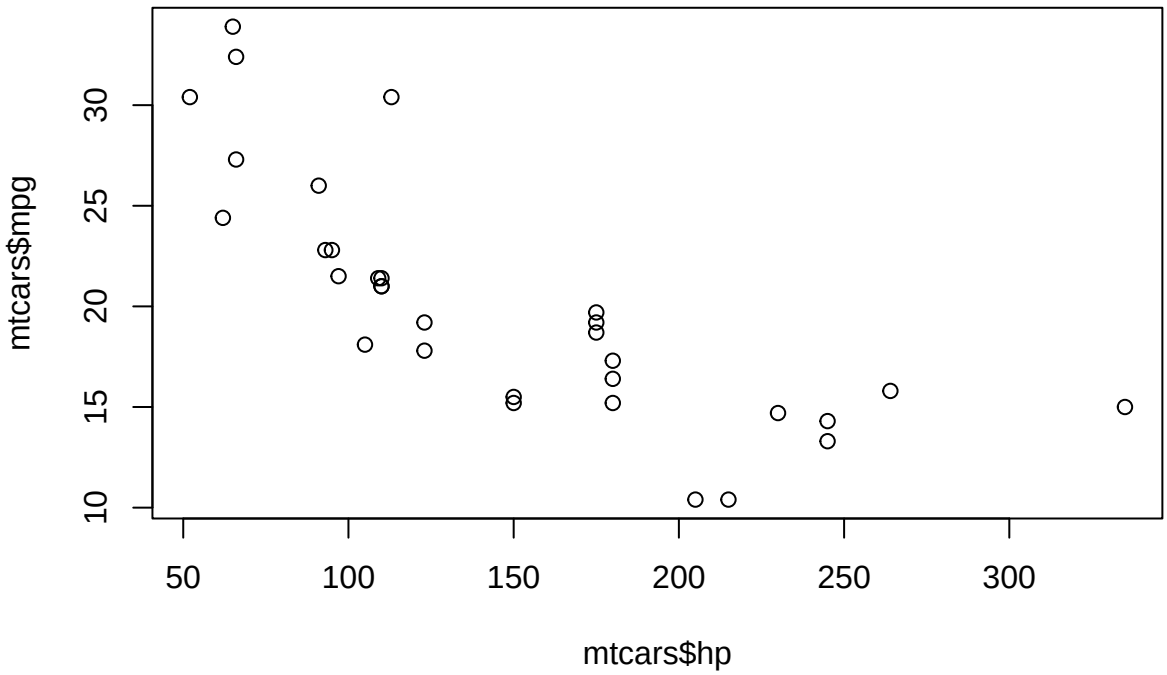
$$y = \beta_0 + \beta_1 x + \epsilon$$

Moving to the example, we use the data set `mtcars` and will create a SLR model between horsepower and miles per gallon (mpg).

```
data(mtcars)
head(mtcars, n = 3)
#>                mpg cyl disp  hp drat    wt  qsec vs am gear carb
#> Mazda RX4     21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
#> Mazda RX4 Wag 21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
#> Datsun 710    22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
```

We may visualize the relationship between horsepower and mpg below.

```
plot(mtcars$hp, mtcars$mpg)
```



Now, to use `SimpLin`, we use the function within the package `SimpLinR`. It will output the estimated regression coefficients for β_0 and β_1, as well as their standard errors and 95% confidence intervals, residuals, and predicted values all as a list. We have

```
mod_output<- SimpLinR(x = mtcars$hp, y = mtcars$mpg)
```

We may access each element of the list to create plots, tables, or do other analysis. For example, the confidence intervals and standard errors of β_0 and β_1 can be accessed as follows.

```
mod_output$Conf_Ints
#>           Lower      Upper
#> b0 26.76194879 33.4357723
#> b1 -0.08889465 -0.0475619
mod_output$SEs
#>          [,1]
#> b0 1.6339210
#> b1 0.0101193
```

We have that the estimates for β_0 and β_1 themselves are

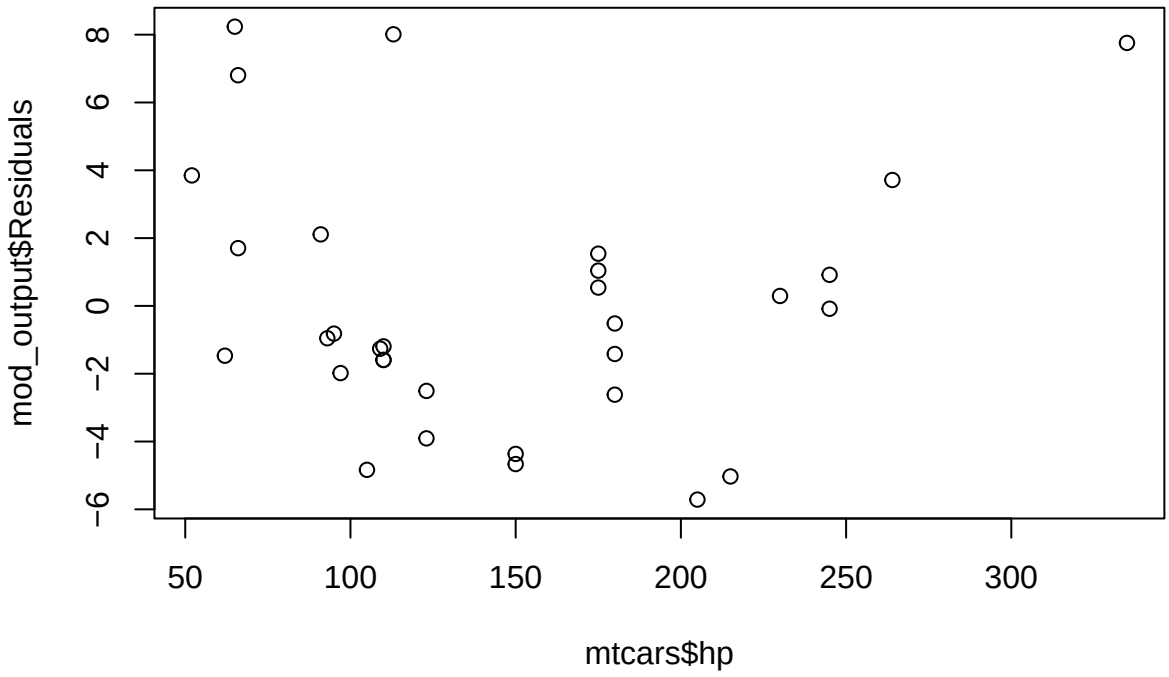
```
mod_output$Coefficients
#>            [,1]
#> b0 30.09886054
#> b1 -0.06822828
```

which indicates that the line of best fit (LOBF) is

$$\hat{y} = 30.0989 + (-0.0682)x$$

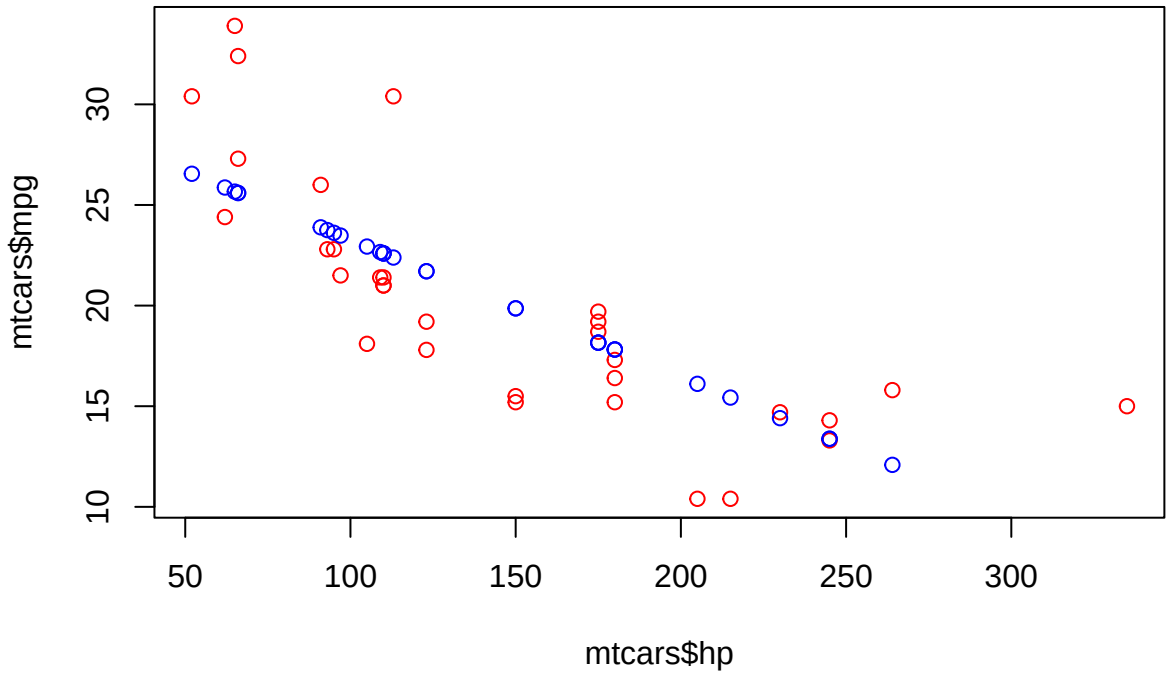
Plotting the residuals, below.

```
plot(mtcars$hp, mod_output$Residuals)
```



The predicted values along with the true values.

```
plot(mtcars$hp, mtcars$mpg, col = 'red')
points(mtcars$hp, mod_output$Pred_Vals, col = 'blue')
```



Troubleshooting

As mentioned earlier in the document, users must make sure that the vectors input in the function `SimpLinR` are of the same length and are numeric. To demonstrate the errors that users may see if either of these are violated by the input, we have the following.

```r
#Vectors of different lengths
SimpLinR(x = c(1,1), y = c(1))
#> Error in SimpLinR(x = c(1, 1), y = c(1)): The vectors are not the same length.

#Non-numeric entries in a vector
SimpLinR(x = c(1,1), y = c(1, 'b'))
#> Error in SimpLinR(x = c(1, 1), y = c(1, "b")): The vectors must both be numeric.
```