Simple Linear Regression

A simple linear regression model that describes the relationship between two variables x and y can be expressed by the following equation. The numbers α and β are called parameters, and ϵ is the error term.

$$y = \alpha + \beta x + \epsilon$$

For example, in the data set faithful, it contains sample data of two random variables named waiting and eruptions.

The waiting variable denotes the waiting time until the next eruptions, and eruptions denotes the duration. Its linear regression model can be expressed as:

$$Eruption = \alpha + \beta\, Waiting + \epsilon$$

Estimated Simple Regression Equation

If we choose the parameters α and β in the simple linear regression model so as to minimize the sum of squares of the error term ϵ, we will have the so called estimated simple regression equation.

It allows us to compute fitted values of y based on values of x.

$$\hat{y} = a + b\, x$$

Problem

Apply the simple linear regression model for the data set faithful, and estimate the next eruption duration if the waiting time since the last eruption has been 80 minutes.

Solution

We apply the lm function to a formula that describes the variable eruptions by the variable waiting, and save the linear regression model in a new variable eruption.lm.

```
> eruption.lm = lm(eruptions ~ waiting, data=faithful)
```

Then we extract the parameters of the estimated regression equation with the coefficients function.

```
> coeffs = coefficients(eruption.lm); coeffs
(Intercept)     waiting
  -1.874016    0.075628
```

We now fit the eruption duration using the estimated regression equation.

```
> waiting = 80                                  # the waiting time
> duration = coeffs[1] + coeffs[2]*waiting
> duration
(Intercept)
     4.1762
```

Answer

Based on the simple linear regression model, if the waiting time since the last eruption has been 80 minutes, we expect the next one to last 4.1762 minutes.

Alternative Solution

We wrap the waiting parameter value inside a new data frame named newdata.

```
> newdata = data.frame(waiting=80) # wrap the parameter
```

Then we apply the predict function to eruption.lm along with newdata.

```
> predict(eruption.lm, newdata)       # apply predict
     1
4.1762
```

Coefficient of Determination

The coefficient of determination of a linear regression model is the quotient of the variances of the fitted values and observed values of the dependent variable.

If we denote y_i as the observed values of the dependent variable, $\overline{y}$ as its mean, and $\widehat{y}_i$ as the fitted value, then the coefficient of determination is:

$$r^2 = \frac{\sum (\widehat{y}_i - \overline{y})^2}{\sum (y_i - \overline{y})^2}$$

Problem

Find the coefficient of determination for the simple linear regression model of the data set faithful.

Solution

We apply the lm function to a formula that describes the variable eruptions by the variable waiting, and save the linear regression model in a new variable eruption.lm.

```
> eruption.lm = lm(eruptions ~ waiting, data=faithful)
```

Then we extract the coefficient of determination from the r.squared attribute of its summary.

```
> summary(eruption.lm)$r.squared
[1] 0.81146
```

Answer

The coefficient of determination of the simple linear regression model for the data set faithful is 0.81146.

Note

Further detail of the r.squared attribute can be found in the R documentation.

```
> help(summary.lm)
```

Significance Test for Linear Regression

Assume that the error term ϵ in the linear regression model is independent of x, and is normally distributed, with zero mean and constant variance.

We can decide whether there is any significant relationship between x and y by testing the null hypothesis that $\beta = 0$.

Problem

Decide whether there is a significant relationship between the variables in the linear regression model of the data set faithful at .05 significance level.

Solution

We apply the lm function to a formula that describes the variable eruptions by the variable waiting, and save the linear regression model in a new variable eruption.lm.

```
> eruption.lm = lm(eruptions ~ waiting, data=faithful)
```

Then we print out the F-statistics of the significance test with the summary function.

```
> summary(eruption.lm)

Call:
lm(formula = eruptions ~ waiting, data = faithful)

Residuals:
    Min      1Q  Median      3Q     Max
-1.2992 -0.3769  0.0351  0.3491  1.1933

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.87402    0.16014   -11.7   <2e-16 ***
waiting      0.07563    0.00222    34.1   <2e-16 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 0.497 on 270 degrees of freedom
Multiple R-squared: 0.811,      Adjusted R-squared: 0.811
F-statistic: 1.16e+03 on 1 and 270 DF,  p-value: <2e-16
```

Answer

As the p-value is much less than 0.05, we reject the null hypothesis that $\beta = 0$. Hence there is a significant relationship between the variables in the linear regression model of the data set faithful.

Note

Further detail of the summary function for linear regression model can be found in the R documentation.

```
> help(summary.lm)
```

Confidence Interval for Linear Regression

Assume that the error term ϵ in the linear regression model is independent of x, and is normally distributed, with zero mean and constant variance.

For a given value of x, the interval estimate for the mean of the dependent variable, $\bar{y}$, is called the confidence interval.

Problem

In the data set faithful, develop a 95% confidence interval of the mean eruption duration for the waiting time of 80 minutes.

Solution

We apply the lm function to a formula that describes the variable eruptions by the variable waiting, and save the linear regression model in a new variable eruption.lm.

```
> attach(faithful)     # attach the data frame

> eruption.lm = lm(eruptions ~ waiting)
```

Then we create a new data frame that set the waiting time value.

```
> newdata = data.frame(waiting=80)
```

We now apply the predict function and set the predictor variable in the newdata argument.

We also set the interval type as "confidence", and use the default 0.95 confidence level.

```
> predict(eruption.lm, newdata, interval="confidence")
    fit    lwr    upr
1 4.1762 4.1048 4.2476

> detach(faithful)     # clean up
```

Answer

The 95% confidence interval of the mean eruption duration for the waiting time of 80 minutes is between 4.1048 and 4.2476 minutes.

Note

Further detail of the predict function for linear regression model can be found in the R documentation.

```
> help(predict.lm)
```

Alternative Solution

We create a function

```
> f <- function(x) {
+ newdata = data.frame(waiting=x)
+ predict.lm = predict(eruption.lm, newdata, interval="confidence")
+ return(predict.lm)
+ }
```

Now only give the value to predict

```
> f(10)
       fit       lwr       upr
1 -1.117737 -1.390249 -0.845224
```

Create one seq. and evaluated it

```
> x = 10:100
> f(x)
> predict = f(x)

> plot(eruptions~waiting, data=faithful)
> lines(x,predict[,1],col="red")
> lines(x,predict[,2],col="blue")
> lines(x,predict[,3],col="blue")
```

Prediction Interval for Linear Regression

Assume that the error term ϵ in the simple linear regression model is independent of x, and is normally distributed, with zero mean and constant variance.

For a given value of x, the interval estimate of the dependent variable y is called the prediction interval.

Problem

In the data set faithful, develop a 95% prediction interval of the eruption duration for the waiting time of 80 minutes.

Solution

We apply the lm function to a formula that describes the variable eruptions by the variable waiting, and save the linear regression model in a new variable eruption.lm.

```
> attach(faithful)     # attach the data frame

> eruption.lm = lm(eruptions ~ waiting)
```

Then we create a new data frame that set the waiting time value.

```
> newdata = data.frame(waiting=80)
```

We now apply the predict function and set the predictor variable in the newdata argument.

We also set the interval type as "predict", and use the default 0.95 confidence level.

```
> predict(eruption.lm, newdata, interval="predict")
    fit    lwr    upr
1 4.1762 3.1961 5.1564

> detach(faithful)     # clean up
```

Answer

The 95% prediction interval of the eruption duration for the waiting time of 80 minutes is between 3.1961 and 5.1564 minutes.

Note

Further detail of the predict function for linear regression model can be found in the R documentation.

```
> help(predict.lm)
```

Alternative Solution

We create a function

```
> f1 <- function(x) {
+ newdata = data.frame(waiting=x)
+ predict.lm = predict(eruption.lm, newdata, interval="predict")
+ return(predict.lm)
+ }
```

Now only give the value to predict

```
> f1(10)
       fit      lwr       upr
1 -1.117737 -2.13254 -0.1029329
```

Create one seq. and evaluated it

```
> x = 10:100
> f1(x)
> predict1 = f1(x)

> plot(eruptions~waiting, data=faithful)
> lines(x,predict1[,1],col="red")
> lines(x,predict1[,2],col="blue")
> lines(x,predict1[,3],col="blue")
```

Residual Plot

The residual data of the simple linear regression model is the difference between the observed data of the dependent variable y and the fitted values $\hat{y}$.

$$Residual = y - \hat{y}$$

Problem

Plot the residual of the simple linear regression model of the data set faithful against the independent variable waiting.

Solution

We apply the lm function to a formula that describes the variable eruptions by the variable waiting, and save the linear regression model in a new variable eruption.lm.

Then we compute the residual with the resid function.

```
> eruption.lm = lm(eruptions ~ waiting, data=faithful)

> eruption.res = resid(eruption.lm)
```

We now plot the residual against the observed values of the variable waiting.

```
> plot(faithful$waiting, eruption.res,
+     ylab="Residuals", xlab="Waiting Time",
+     main="Old Faithful Eruptions")

> abline(0, 0)                                    # the horizon
```

Note

Further detail of the resid function can be found in the R documentation.

```
> help(resid)
```

Standardized Residual

The standardized residual is the residual divided by its standard deviation.

$$Standardized\ Residual\ i = \frac{Residual\ i}{Standard\ Deviation\ of\ Residual\ i}$$

Problem

Plot the standardized residual of the simple linear regression model of the data set faithful against the independent variable waiting.

Solution

We apply the lm function to a formula that describes the variable eruptions by the variable waiting, and save the linear regression model in a new variable eruption.lm.

Then we compute the standardized residual with the rstandard function.

```
> eruption.lm = lm(eruptions ~ waiting, data=faithful)
```

```
> eruption.stdres = rstandard(eruption.lm)
```

We now plot the standardized residual against the observed values of the variable waiting.

```
> plot(faithful$waiting, eruption.stres,
+     ylab="Standardized Residuals",
+     xlab="Waiting Time",
+     main="Old Faithful Eruptions")
> abline(0, 0)                          # the horizon
```

Note

Further detail of the rstandard function can be found in the R documentation.

```
> help(rstandard)
```

Normal Probability Plot of Residuals

The normal probability plot is a graphical tool for comparing a data set with the normal distribution.

We can use it with the standardized residual of the linear regression model and see if the error term ϵ is actually normally distributed.

Problem

Create the normal probability plot for the standardized residual of the data set faithful.

Solution

We apply the lm function to a formula that describes the variable eruptions by the variable waiting, and save the linear regression model in a new variable eruption.lm.

Then we compute the standardized residual with the rstandard function.

```
> eruption.lm = lm(eruptions ~ waiting, data=faithful)

> eruption.stdres = rstandard(eruption.lm)
```

We now create the normal probability plot with the qqnorm function, and add the qqline for further comparison.

```
> qqnorm(eruption.stdres,
+     ylab="Standardized Residuals",
+     xlab="Normal Scores",
+     main="Old Faithful Eruptions")

> qqline(eruption.stdres)
```

Note

Further detail of the qqnorm and qqline functions can be found in the R documentation.

```
> help(qqnorm)
```

Exercise

This function computes model II simple linear regression using the following methods: ordinary least squares (OLS), major axis (MA), standard major axis (SMA), and ranged major axis (RMA). The model only accepts one response and one explanatory variable.

```
> install.packages("lmodel2")


> library(lmodel2)


> help(lmodel2)
```

```
## Example 1 (surgical unit data)
> data(mod2ex1)
> Ex1.res <- lmodel2(Predicted_by_model ~ Survival, data=mod2ex1, nperm=99)
> Ex1.res
> plot(Ex1.res)
```

```
## Example 2 (eagle rays and Macomona)
> data(mod2ex2)
> Ex2.res <- lmodel2(Prey ~ Predators, data=mod2ex2, "relative", "relative", 99)
> Ex2.res
> op <- par(mfrow = c(1,2))
> plot(Ex2.res, "SMA")
> plot(Ex2.res, "RMA")
> par(op)
```

```
## Example 3 (cabezon spawning)
> op <- par(mfrow = c(1,2))
> data(mod2ex3)
> Ex3.res <- lmodel2(No_eggs ~ Mass, data=mod2ex3, "relative", "relative", 99 )
> Ex3.res
> plot(Ex3.res, "SMA")
> plot(Ex3.res, "RMA")
> par(op)
```

```
## Example 4 (highly correlated random variables)
> op <- par(mfrow=c(1,2))
> data(mod2ex4)
> Ex4.res <- lmodel2(y ~ x, data=mod2ex4, "interval", "interval", 99)
> Ex4.res
> plot(Ex4.res, "OLS")
> plot(Ex4.res, "MA")
> par(op)



# Example 5 (uncorrelated random variables)
> data(mod2ex5)
> Ex5.res <- lmodel2(random_y ~ random_x, data=mod2ex5, "interval", "interval", 99)
> Ex5.res
> op <- par(mfrow = c(2,2))
> plot(Ex5.res, "OLS")
> plot(Ex5.res, "MA")
> plot(Ex5.res, "SMA")
> plot(Ex5.res, "RMA")
> par(op)



## Example 6 where cor(y,x) = 0 by construct (square grid of points)
> y0 = rep(c(1,2,3,4,5),5)
> x0 = c(rep(1,5),rep(2,5),rep(3,5),rep(4,5),rep(5,5))
> plot(x0, y0)
> Ex6 = as.data.frame(cbind(x0,y0))
> zero.res = lmodel2(y0 ~ x0, data=Ex6, "relative", "relative")
> print(zero.res)
> op <- par(mfrow = c(1,2))
> plot(zero.res, "OLS")
> plot(zero.res, "MA")
> par(op)
```