LinearReggresion

Code

Loading ggplot

Hide

```
library(ggplot2)
```

Print the head of the dataset

Hide

```
path <- "/Users/pulkitbatra/Downloads/archive-2/train.csv"
trainingSet = read.csv(path)
```

Check for NA and missing values is.na return a vector with value TT for missing values.

Hide

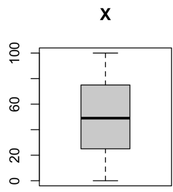
```
numberOfNA = length(which(is.na(trainingSet)==T))
if(numberOfNA > 0) {
  cat('Number of missing values found: ', numberOfNA)
  cat('\nRemoving missing values...')
  trainingSet = trainingSet[complete.cases(trainingSet), ]
}
```

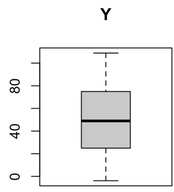
```
Number of missing values found:  1
Removing missing values...
```

Check for outliers Divide the graph area in 2 columns

Hide

```
par(mfrow = c(1, 2))
# Boxplot for X
boxplot(trainingSet$x, main='X', sub=paste('Outliers: ', boxplot.stats(trainingSet$x)$out))
# Boxplot for Y
boxplot(trainingSet$y, main='Y', sub=paste('Outliers: ', boxplot.stats(trainingSet$y)$out))
```





Outliers:

Outliers:

Hide

```
cor(trainingSet$x, trainingSet$y)
```

```
[1] 0.9953399
```

0.99 shows a very strong relation.

Hide

```
regressor = lm(formula = y ~.,
               data = trainingSet)
```

Hide

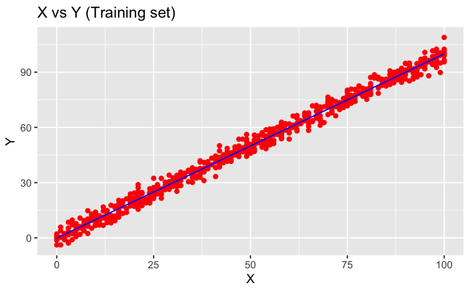
```
summary(regressor)
```

```

Call:
lm(formula = y ~ ., data = trainingSet)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.1523 -2.0179  0.0325  1.8573  8.9132 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.107265   0.212170  -0.506    0.613    
x            1.000656   0.003672 272.510   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.809 on 697 degrees of freedom
Multiple R-squared:  0.9907,    Adjusted R-squared:  0.9907 
F-statistic: 7.426e+04 on 1 and 697 DF,  p-value: < 2.2e-16
```

plot

Hide

```
ggplot() +
  geom_point(aes(x = trainingSet$x, y = trainingSet$y),
             colour = 'red') +
  geom_line(aes(x = trainingSet$x, y = predict(regressor, newdata = trainingSet)),
            colour = 'blue') +
  ggtitle('X vs Y (Training set)') +
  xlab('X') +
  ylab('Y')
```



Test

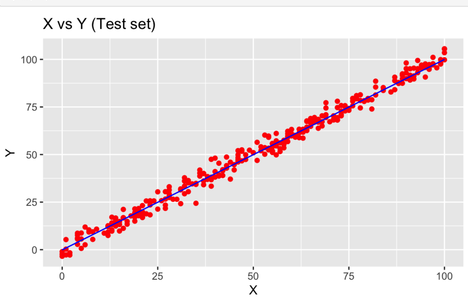
Hide

```
testPath <- "/Users/pulkitbatra/Downloads/archive-2/test.csv"
testSet = read.csv(testPath)

y_pred = predict(regressor, newdata = testSet)
```

Visualsing the result

Hide

```
ggplot() +
  geom_point(aes(x = testSet$x, y = testSet$y),
             colour = 'red') +
  geom_line(aes(x = trainingSet$x, y = predict(regressor, newdata = trainingSet)),
            colour = 'blue') +
  ggtitle('X vs Y (Test set)') +
  xlab('X') +
  ylab('Y')
```



Plot shows model was a good fit.

Hide

```
compare <- cbind (actual=testSet$x, y_pred)  # combine actual and predicted
mean (apply(compare, 1, min)/apply(compare, 1, max))
```

```
[1] -Inf
```

Hide

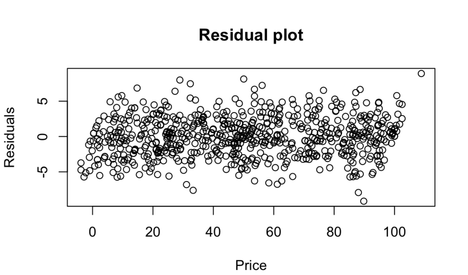
```
mean(0.9,0.9,0.9,0.9,0.9)
```

```
[1] 0.9
```

Check for residual mean and distribution

Hide

```
plot(trainingSet$y, resid(regressor),
     ylab="Residuals", xlab="Price",
     main="Residual plot")
```



Hide

```
mean(regressor$residuals)
```

```
[1] -1.353233e-16
```