Dhir Thacker | 17070122019 | CSE-1

SD lab assignment 8 (R-assignment 4)

Linear regression in R

```
Loading ggplot
&
Importing dataset
```{r}

library(ggplot2)

trainingSet = read.csv('train.csv')
View(trainingSet)
```
```

trainingSet | r_linear_regression.R | Untitled

Filter

	x	y
1	24	21.5494520
2	50	47.4644630
3	15	17.2186563
4	38	36.5863980
5	87	87.2889839
6	36	32.4638749
7	12	10.7808968
8	81	80.7633986
9	25	24.6121515
10	5	6.9633191
11	16	11.2375734
12	16	13.5329021
13	24	24.6032390
14	39	39.4004998
15	54	48.4375384
16	60	61.6990032
17	26	26.9283242
18	73	70.4052055

Showing 1 to 19 of 699 entries, 2 total columns
```
```

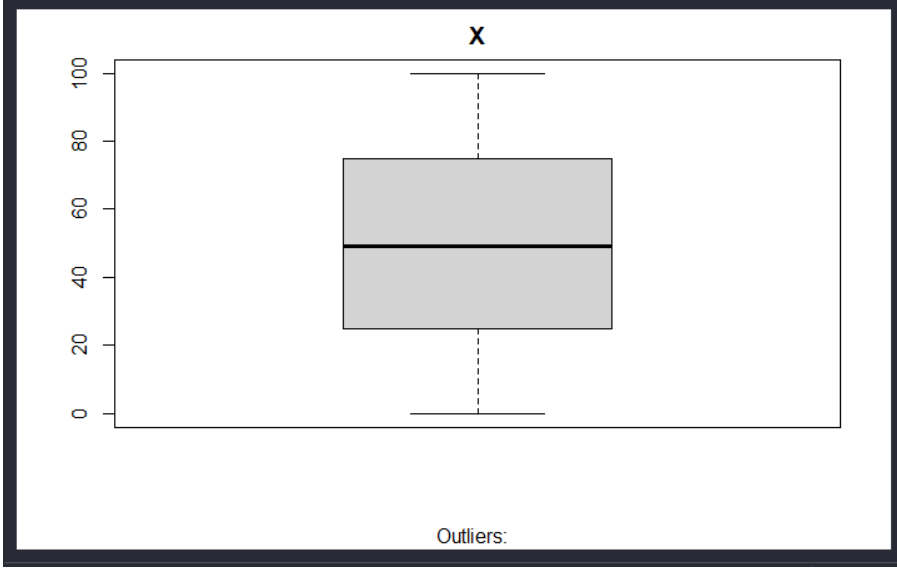
Checking for NA and missing values

```{r}
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...
```

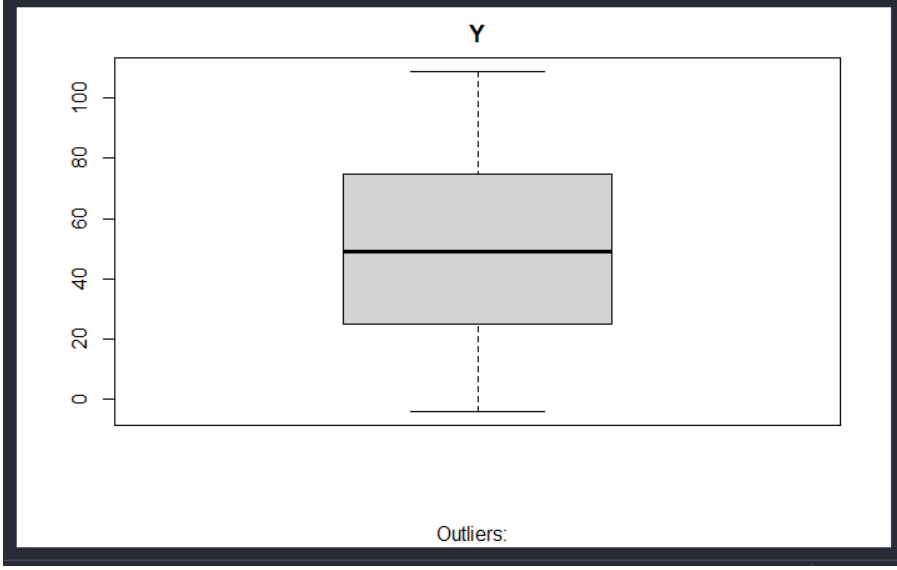
Boxplot for x

```{r}
boxplot(trainingSet$x, main='X', sub=paste('Outliers: ',
                                           boxplot.stats(trainingSet$x)$out))
```



Boxplot for y

```{r}
boxplot(trainingSet$y, main='Y', sub=paste('Outliers: ',
                                           boxplot.stats(trainingSet$y)$out))
```



Finding correlation

```{r}
cor(trainingSet$x, trainingSet$y)
```

```
[1] 0.9953399
```

0.99 shows a very strong correlation

Fitting simple linear regression
. is used to fit predictor using all independent variables

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

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
```

In Linear Regression, the Null Hypothesis is that the coefficients associated with the variables is equal to zero.

The alternate hypothesis is that the coefficients are not equal to zero
(i.e. there exists a relationship between the independent variable in question and the dependent variable).

P value has 3 stars which means x is of very high statistical significance.

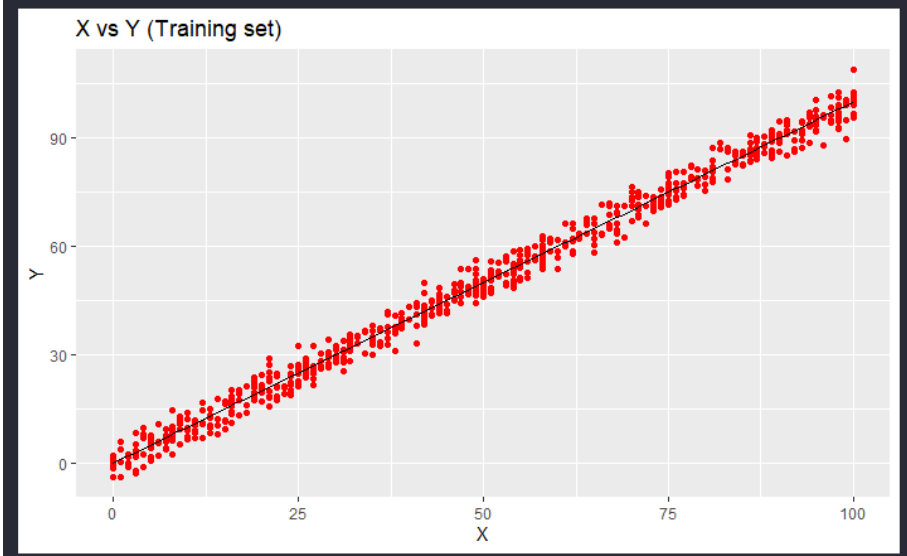
P value is less than 0. Genraaly below 0.05 is considered good.

R-Squared tells us is the proportion of variation in the dependent (response) variable that has been explained by this model.

R square is 0.99 which shows very good variation between dependent variable(y) and independent variable(x).

Visualizing training set results

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



No outliers present and there is a linear relationship

Importing test data

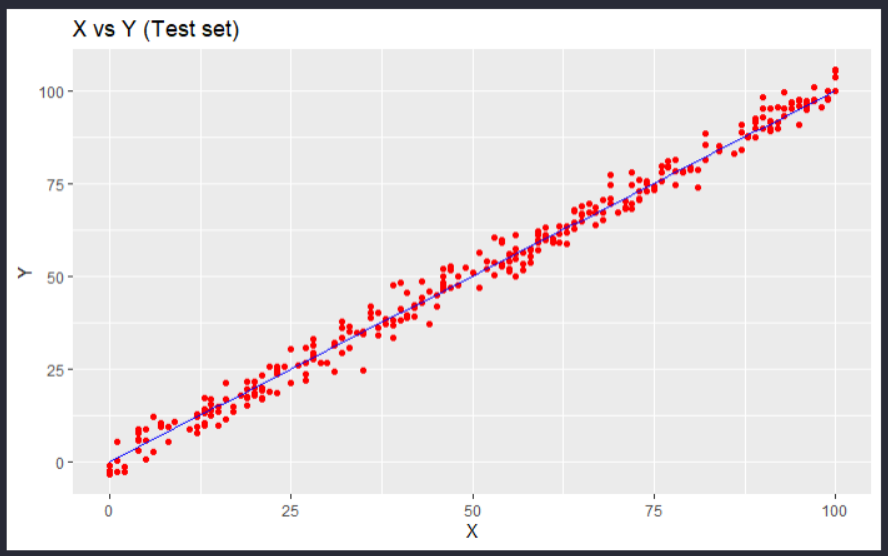
```{r}

testSet = read.csv('test.csv')
```

Predicting test results

```{r}

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

Visualizing the test set results

```{r}

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

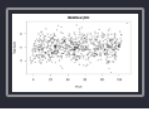
Finding accuracy

```{r}
compare <- cbind(actual=testSet$x, y_pred)
mean(apply(compare, 1, min)/apply(compare, 1, max))
mean(0.9, 0.9, 0.9, 0.9)
```

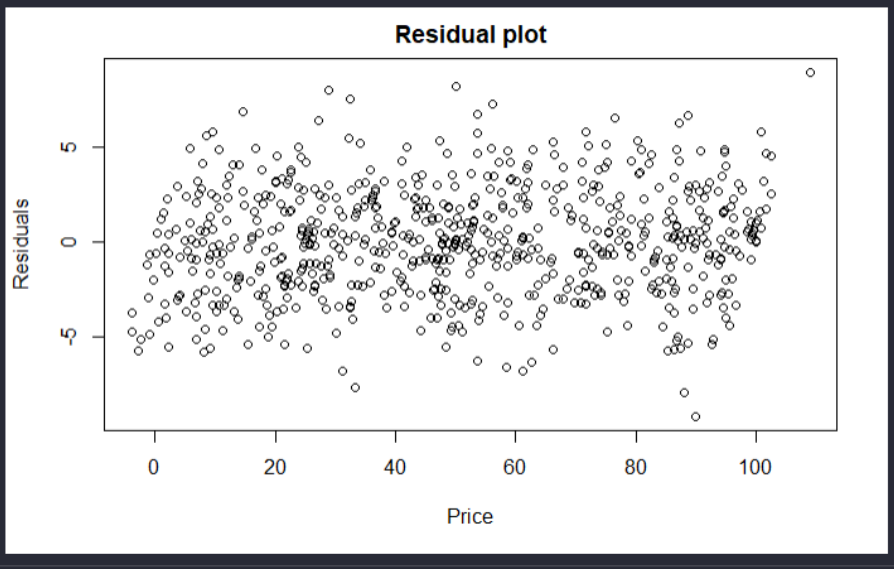
```
[1] -Inf
[1] 0.9
```

Check for residual mean and distribution

```{r}
plot(trainingSet$y, resid(regressor),
     ylab="Residuals", xlab="Price",
     main="Residual plot")
mean(regressor$residuals)
```

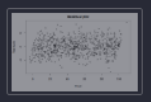


R Console



Check for residual mean and distribution

```{r}
plot(trainingSet$y, resid(regressor),
     ylab="Residuals", xlab="Price",
     main="Residual plot")
mean(regressor$residuals)
```



R Console

```
[1] -2.207445e-17
```