R Notebook

```
library(readxl)
```

```
data = read.csv("student_scores.csv")
print(data)
```

```
##    Hours Scores
## 1    2.5     21
## 2    5.1     47
## 3    3.2     27
## 4    8.5     75
## 5    3.5     30
## 6    1.5     20
## 7    9.2     88
## 8    5.5     60
## 9    8.3     81
## 10   2.7     25
## 11   7.7     85
## 12   5.9     62
## 13   4.5     41
## 14   3.3     42
## 15   1.1     17
## 16   8.9     95
## 17   2.5     30
## 18   1.9     24
## 19   6.1     67
## 20   7.4     69
## 21   2.7     30
## 22   4.8     54
## 23   3.8     35
## 24   6.9     76
## 25   7.8     86
```

```
View(data)
```

Data Preprocessing

```
head(data)
```

	Hours <dbl>	Scores <int>
1	2.5	21
2	5.1	47
3	3.2	27
4	8.5	75
5	3.5	30

	Hours <dbl>	Scores <int>
6	1.5	20

6 rows

```
tail(data)
```

	Hours <dbl>	Scores <int>
20	7.4	69
21	2.7	30
22	4.8	54
23	3.8	35
24	6.9	76
25	7.8	86

6 rows

```
dim(data)
```

```
## [1] 25  2
```

```
typeof(data$Hours)
```

```
## [1] "double"
```

```
typeof(data$Scores)
```

```
## [1] "integer"
```

Importing the packages required

```
library("caret")
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library("caTools")
library(ggplot2)
library(dplyr)
```

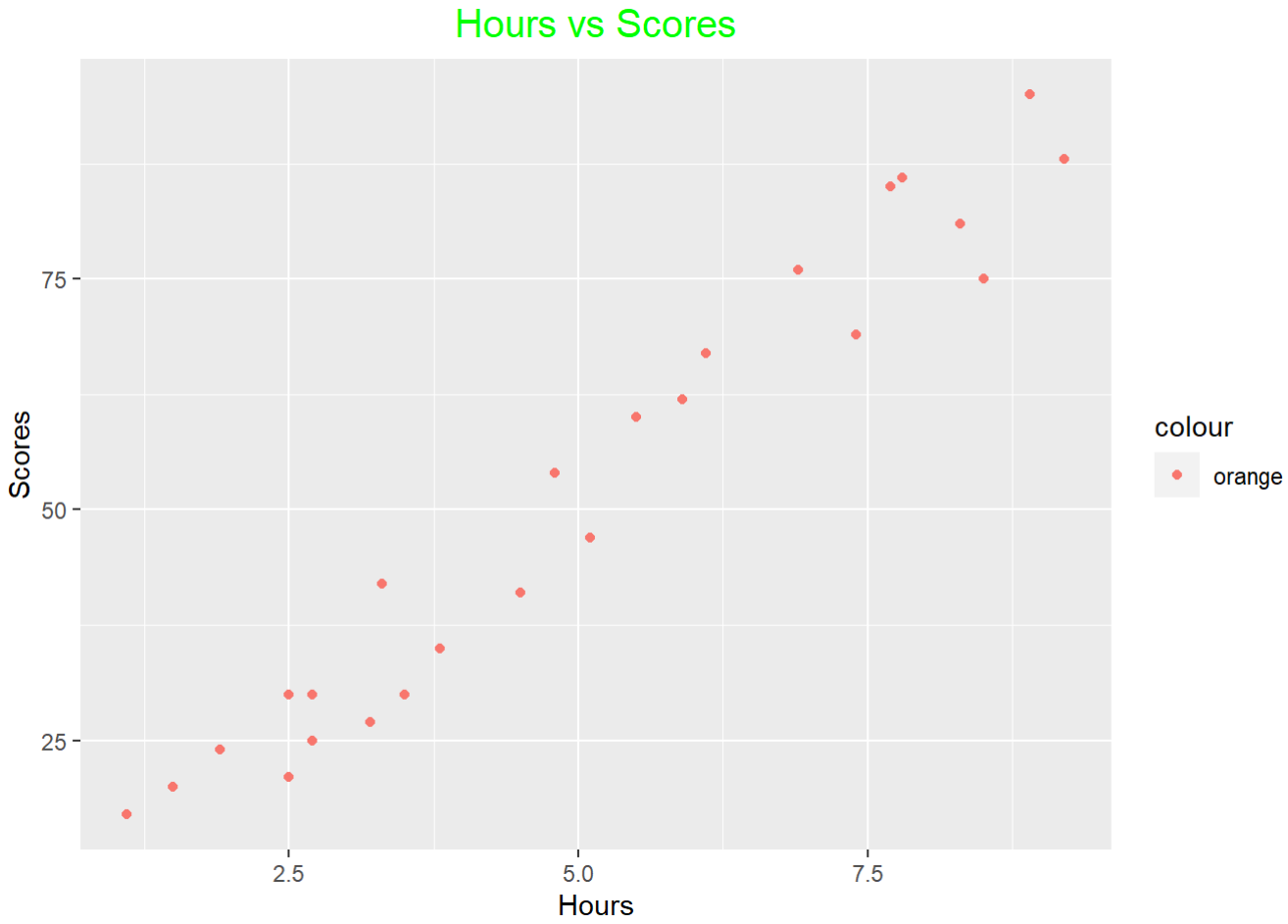
```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Plotting the distribution of scores

```
ggplot(data=data,aes(x=Hours,y=Scores,color="orange"))+
  geom_point() +labs(title = "Hours vs Scores")+
  theme(plot.title = element_text(hjust = 0.5, size = 15, color = "Green"))
```



Preparing the data

```
set.seed(2)
split=sample.split(data$Scores,SplitRatio = 0.8)
train=subset(data,split== TRUE)
test=subset(data,split== FALSE)
train
```

	Hours <dbl>	Scores <int>
1	2.5	21
2	5.1	47
3	3.2	27
4	8.5	75
7	9.2	88
8	5.5	60
9	8.3	81
10	2.7	25
11	7.7	85
12	5.9	62

1-10 of 20 rows Previous **1** 2 Next

```
test
```

	Hours <dbl>	Scores <int>
5	3.5	30
6	1.5	20
16	8.9	95
17	2.5	30
23	3.8	35

5 rows

Training the algorithm

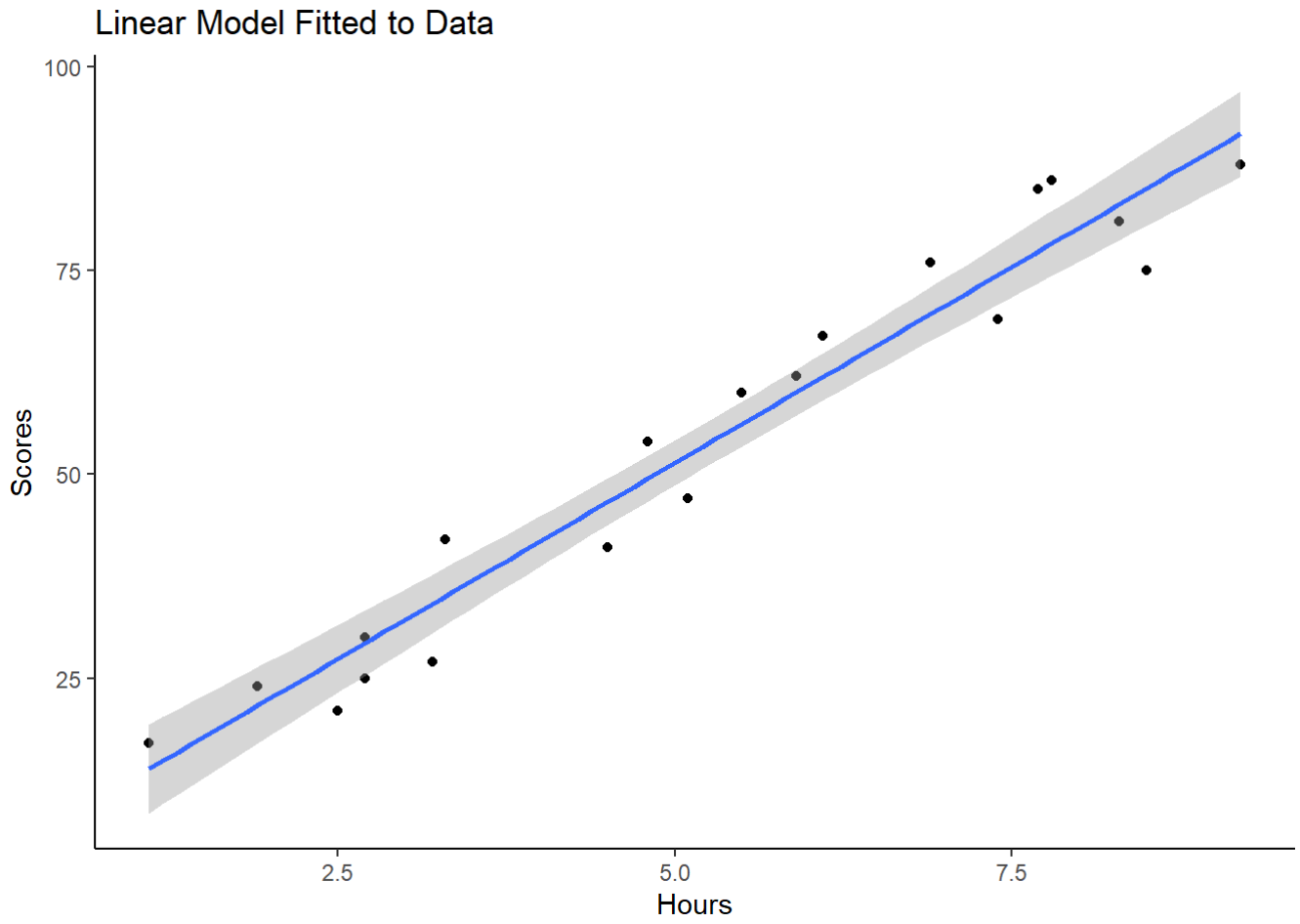
```
model=lm(Scores~ Hours, data = train)
summary(model)
```

```
##
## Call:
## lm(formula = Scores ~ Hours, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.022  -5.385   1.324   4.649   7.702
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.370      3.129   1.077    0.296
## Hours          9.606      0.542  17.725 7.66e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.795 on 18 degrees of freedom
## Multiple R-squared:  0.9458, Adjusted R-squared:  0.9428
## F-statistic: 314.2 on 1 and 18 DF,  p-value: 7.663e-13
```

Plotting the regression line

```
ggplot(data=train,aes(x=Hours, y=Scores)) +
  geom_point()+ geom_smooth(method = "lm")+
  theme(panel.background = element_rect(fill = "white"),
                                                                    axis.line.x=element
_line(),
                                                                    axis.line.y=element
_line()) +
  ggtitle("Linear Model Fitted to Data")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Making Predictions

```
pred=predict(model,test)
pred
```

```
##        5        6       16       17       23
## 36.99123 17.77895 88.86437 27.38509 39.87307
```

Comparing Actual vs Predicted

```
df=data.frame("Actual"=test$Scores,"Predicted"=pred)
df
```

	Actual <int>	Predicted <dbl>
5	30	36.99123
6	20	17.77895
16	95	88.86437
17	30	27.38509
23	35	39.87307

5 rows

You can also test with your own data

```
hours = 9.25
own_pred=predict(model, data.frame(Hours=hours))
paste0("No. of hours= ", hours)
```

```
## [1] "No. of hours= 9.25"
```

```
paste0("Predicted score= ",own_pred)
```

```
## [1] "Predicted score= 92.2265229009136"
```

Evaluating the model

```
error=RMSE(test$Scores,pred)
paste0("Root mean square error= ", error)
```

```
## [1] "Root mean square error= 4.94046106937404"
```