TSF TASK 1

Hrishav Mahajan

10/2/2021

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring:

- HTML
- PDF
- MS Word documents.

For more details on using R Markdown click here (http://rmarkdown.rstudio.com).

For the data frame click here (https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/student_scores%20-%20student_scores.csv).

Here we will be using the fabulous four packages included in tidyverse of the total available eight. That is tidyr, ggplot2, readrand dplyr. The remaining four are as follows: purrr, tibble, stringr and forcats.

Importing packages.

```
install.packages("tidyr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages("ggplot2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages("readr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages("dplyr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

Loading packages.

```
library(tidyr)
library(ggplot2)
library(readr)
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
```

Importing the data frame and then displaying the first 6 rows. Since it is a very small data frame which doesn’t have NaN values we are skipping on the data cleaning part.

```
df<-read_csv("http://bit.ly/w-data")
```

```
## `curl` package not installed, falling back to using `url()`
```

```
## Rows: 25 Columns: 2
```

```
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (2): Hours, Scores
```

```
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(df)
```

```
## # A tibble: 6 × 2
##   Hours Scores
##   <dbl>  <dbl>
## 1   2.5     21
## 2   5.1     47
## 3   3.2     27
## 4   8.5     75
## 5   3.5     30
## 6   1.5     20
```

Now applying simple linear regression in R.

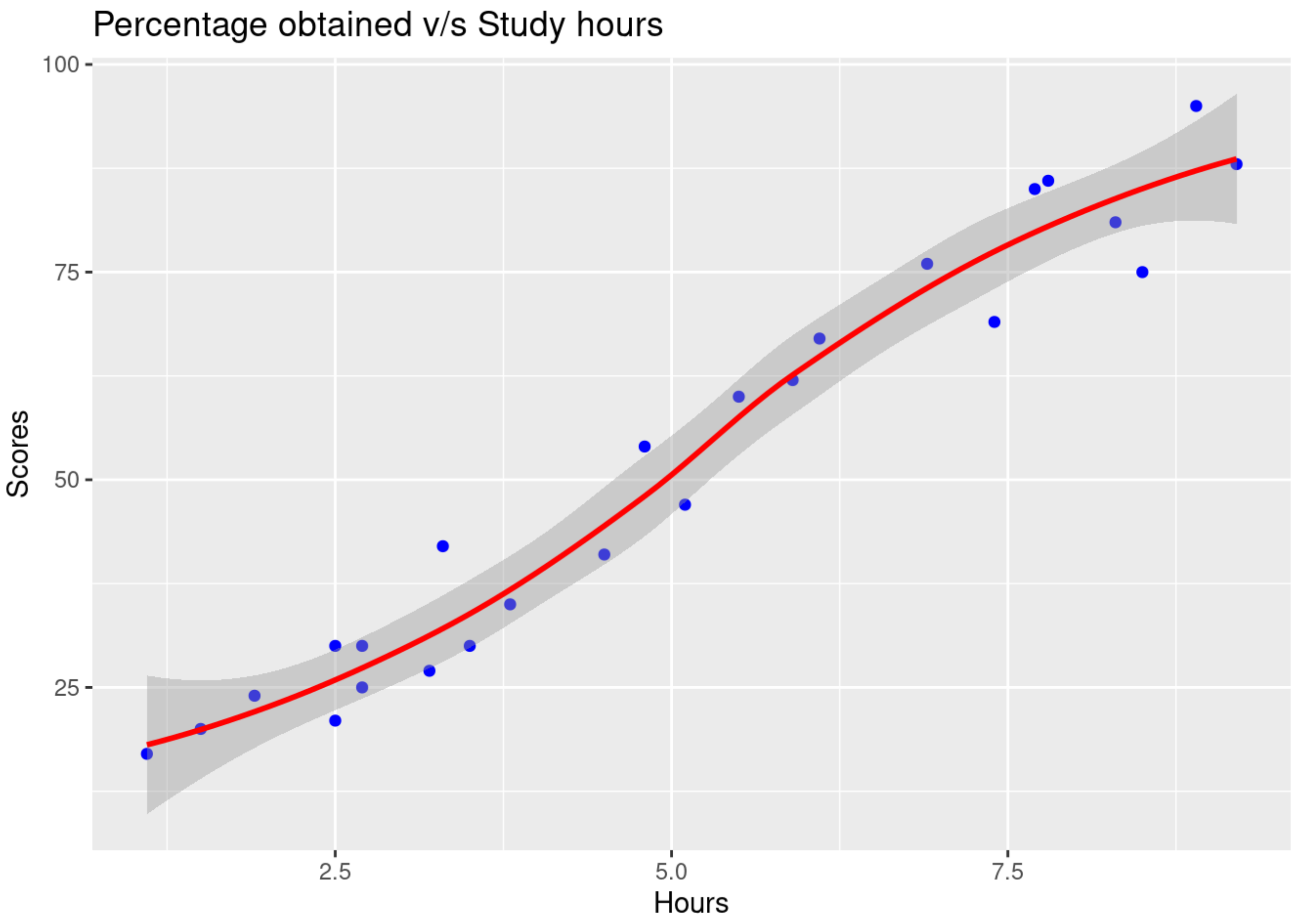
```
model<- lm(Scores~Hours,df)
summary(model)
```

```
## 
## Call:
## lm(formula = Scores ~ Hours, data = df)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -10.578  -5.340   1.839   4.593   7.265 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   2.4837     2.5317   0.981    0.337    
## Hours         9.7758     0.4529  21.583   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 5.603 on 23 degrees of freedom
## Multiple R-squared:  0.9529, Adjusted R-squared:  0.9509 
## F-statistic: 465.8 on 1 and 23 DF,  p-value: < 2.2e-16
```

In this chunk we are using the ggplot function and adding two layers to it and playing a bit with the aesthetics.

```
ggplot(data=df)+
  geom_point(mapping=aes(x=Hours, y=Scores),color="blue")+
  geom_smooth(mapping=aes(x=Hours, y=Scores),color="red")+
  labs(title="Percentage obtained v/s Study hours",)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```
n<-data.frame(Hours=9.25)
result<-predict(model,n)
print(result)
```

```
##        1
## 92.90985
```

Thus we get the predicted value as 92.90985% for a student studying for 9.25 hours.