Practice 2

Smit Patil

1. The built-in dataset USArrests contains statistics about violent crime rates in the US States. Determine which states are outliers in terms of murders. Outliers, for the sake of this question, are defined as values that are more than 1.5 standard deviations from the mean.

```
#Problem 1

#Importing libraries
library(tidyr)
library(knitr)
library(tinytex)
library(data.table)

#Retriving data from R
state_names <- data.table("State" = state.name)
USA_Arrests <- cbind(state_names, USArrests)

#Calculating Mean, Standard Deviation and Z-Score for Murder
mean_murder <- mean(USA_Arrests$Murder)
stdev_murder <- sd(USA_Arrests$Murder)
zscore <- abs((USA_Arrests$Murder - mean_murder)/stdev_murder)

#Joining column states_zscore to the table
states_zscore <- cbind(state_names, zscore)

#Calculating outlier states with condition (Z-Score > 1.5)
outlier_states <- states_zscore[zscore>1.5]
outlier_states
```

```
##             State   zscore
## 1:        Florida 1.747671
## 2:        Georgia 2.206860
## 3:      Louisiana 1.747671
## 4:    Mississippi 1.908387
## 5:   North Dakota 1.604405
## 6: South Carolina 1.518077
```

2. For the same dataset as in (1), is there a correlation between urban population and murder, i.e., as one goes up, does the other statistic as well? Comment on the strength of the correlation. Calculate the Pearson coefficient of correlation in R.

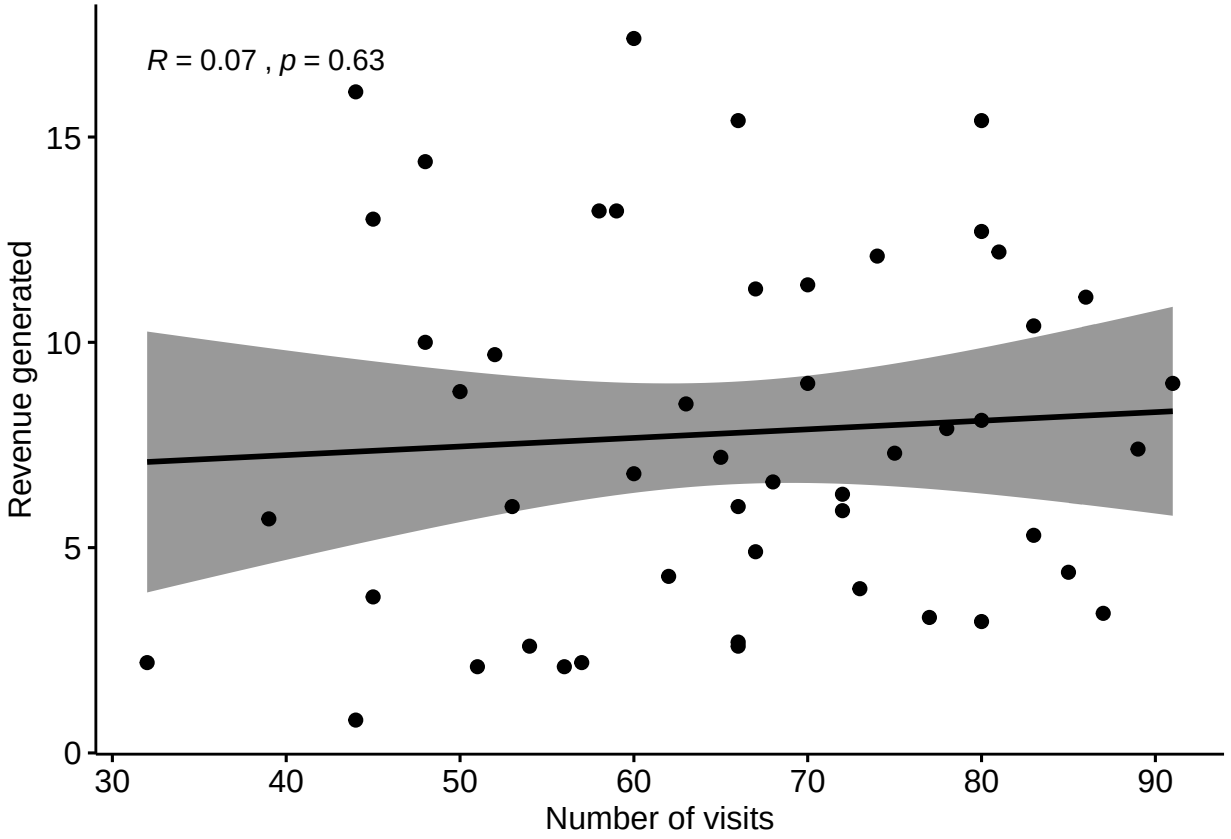
```
#Problem 2

#Plotting the Pearson Correlation using library(ggpubr)
library(ggpubr)
```

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

```
ggscatter(USA_Arrests, x = "UrbanPop", y = "Murder",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "Number of visits", ylab = "Revenue generated")
```

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



3. Based on the data on the growth of mobile phone use in Brazil (you'll need to copy the data and create a CSV that you can load into R or use the gsheet2tbl() function from the gsheet package), forecast phone use for the next time period using a 2-year weighted moving average (with weights of 5 for the most recent year, and 2 for other), exponential smoothing (alpha of 0.4), and linear regression trendline.

```
#Problem 3

#Importing Brazil mobile phone growth dataset
brazil_mobile <- read.csv("brazil_mobile_phone_growth.csv")

#Calculating last 2 year from the dataset
n <- nrow(brazil_mobile)
last_2 <- brazil_mobile[c(n,n-1),2]
```

```r
#Assigning weights to the last 2 year
weight <- c(5,2)
sw <- weight*last_2

#Forecasting using the weighted average
forecast_WA <- sum(sw)/sum(weight)
forecast_WA
```

```
## [1] 194662700
```

```r
#Creating a new data set to forecat using Exponential Smoothing and Linear Regression
mobile_growth <- brazil_mobile

#Setting alpha equals to 0.4
a<- 0.4

#Creating new columns Ft and E
mobile_growth$Ft <- 0
mobile_growth$E <- 0

#Calculating values for Ft and E
mobile_growth$Ft[1] <- mobile_growth[1,2]

for (i in 2:n)
  {
  mobile_growth$Ft[i] <- mobile_growth$Ft[i-1] + a*mobile_growth$E[i-1]
  mobile_growth$E[i] <- mobile_growth[i,2] - mobile_growth$Ft[i]
  }

#Forecast using exponential smoothing
forecast_ES <- mobile_growth$Ft[n] + a*mobile_growth$E[n]
forecast_ES
```

```
## [1] 165168214
```

```r
#Creating model for linear regression
model <- lm(mobile_growth$Subscribers ~ mobile_growth$Year)

summary(model)
```

```
##
## Call:
## lm(formula = mobile_growth$Subscribers ~ mobile_growth$Year)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -12307858  -9795553  -4238521   7402838  20622182
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -15710760    8041972  -1.954   0.0825 .
## mobile_growth$Year  18276748    1185724  15.414  8.9e-08 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12440000 on 9 degrees of freedom
## Multiple R-squared:  0.9635, Adjusted R-squared:  0.9594
## F-statistic: 237.6 on 1 and 9 DF,  p-value: 8.903e-08
```

```
print(model)
```

```
##
## Call:
## lm(formula = mobile_growth$Subscribers ~ mobile_growth$Year)
##
## Coefficients:
##        (Intercept)  mobile_growth$Year
##          -15710760            18276748
```

```
#Forecast using linear rigression
forecast_LM <- -15710760 + 18276748*12
forecast_LM
```

```
## [1] 203610216
```

4. Calculate the squared error for each model, i.e., use the model to calculate a forecast for each given time period and then the squared error. Finally, calculate the average (mean) squared error for each model. Which model has the smallest mean squared error (MSE)?

```
#Probelm 4

##### LINEAR REGRESSION #####

#Creating data for linear regression
data_LM <- brazil_mobile

data_LM$F <- 0
data_LM$absError <- 0
data_LM$sqrdError <- 0

#Calculating Sqaured Error for linear regression
for (i in 1:nrow(data_LM))
  {
  data_LM$F[i] <- -15710760 + 18276748 * data_LM$Year[i]
  data_LM$absError[i] <- abs(data_LM$Subscribers[i] - data_LM$F[i])
  data_LM$sqrdError[i] <- data_LM$absError[i] ^ 2
  }

#Mean Sqarred Error for linear regression
MSE_LM <- mean(data_LM$sqrdError)

##### EXPONENTIAL SMOOTHING #####

#Creating data for exponential smoothing
```

```r
data_ES <- brazil_mobile

data_ES$Ft <- 0
data_ES$E <- 0
data_ES$sqrdError <- 0

data_ES$Ft[1] <- data_ES[1,2]

#Calculating Sqaured Error for exponential smoothing
for (i in 2:nrow(data_ES))
  {
  data_ES$Ft[i] <- data_ES$Ft[i-1] + a*data_ES$E[i-1]
  data_ES$E[i] <- data_ES$Subscribers[i] - data_ES$Ft[i]
  data_ES$sqrdError[i] <- data_ES$E[i] ^ 2
  }

#Mean Sqarred Error for exponential smoothing
MSE_ES <- mean(data_ES$sqrdError)

##### WEIGHTED AVERAGE #####

#Creating data for weighted average
data_WA <- brazil_mobile

data_WA$Ft <- 0
data_WA$Error <- 0
data_WA$sqrdError <- 0

data_WA$Ft[1] <- data_WA$Subscribers[1]
data_WA$Ft[2] <- data_WA$Subscribers[2]

#Calculating Sqaured Error for weighted average
for (i in 3:nrow(data_WA))
  {
  last2 <- data_WA$Subscribers[c(i-1,i-2)]
  weight <- c(5,2)
  sw <- weight*last2
  data_WA$Ft[i] <- sum(sw)/sum(weight)
  data_WA$Error[i] <- abs(data_WA$Subscribers[i]-data_WA$Ft[i])
  data_WA$sqrdError[i] <- data_WA$Error[i] ^ 2
  }

#Mean Sqarred Error for weighted average
MSE_WA <- mean(data_WA$sqrdError)

#Creating a datatable for all three models
Model <- c("2-year Weighted Average", "Exponential Smoothing", "Linear Regression")
Mean_Square_Error <- c(MSE_WA,MSE_ES,MSE_LM)

MSE_Table <- data.table(Model, Mean_Square_Error)
MSE_Table[order(Mean_Square_Error)]
```

```
##                       Model Mean_Square_Error
```

```
## 1:       Linear Regression      1.265347e+14
## 2: 2-year Weighted Average      5.441439e+14
## 3:   Exponential Smoothing      1.473838e+15
```

5. Calculate a weighted average forecast by averaging out the three forecasts calculated in (3) with the following weights: 4 for trend line, 2 for exponential smoothing, 1 for weighted moving average. Remember to divide by the sum of the weights in a weighted average.

```
#Problem 5

#Calculating weighted average for all three forecasts
values <- c(forecast_LM, forecast_ES, forecast_WA)
weights <- c(4,2,1)

weight_values <- values*weights

forecast <- sum(weight_values)/sum(weights)
forecast
```

```
## [1] 191348570
```