Practice 2_DA5030

Shivam Verma

Question 1. 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.

```
data <- USArrests

# z-score calculation
data$z_score <- abs(((mean(data$Murder)) - data$Murder) / (sd(data$Murder)))

# filter outliers
outliers <- data[which(data$z_score > 1.50), 0:4 ]
outliers
```

```
##                Murder Assault UrbanPop Rape
## Florida          15.4     335       80 31.9
## Georgia          17.4     211       60 25.8
## Louisiana        15.4     249       66 22.2
## Mississippi      16.1     259       44 17.1
## North Dakota      0.8      45       44  7.3
## South Carolina   14.4     279       48 22.5
```

Question 2. 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.

```
cor.test(x = data$Murder, y = data$UrbanPop, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$Murder and data$UrbanPop
## t = 0.48318, df = 48, p-value = 0.6312
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2128979  0.3413107
## sample estimates:
##        cor
## 0.06957262
```

As value for pearson correlation estimates to be around 0.07 this shows a weak linear relationship between urban population and murder. As one increases, the other typically doesn't increase.

Question 3. 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.

```r
# Weighted Avg Forecast
df<- read.csv("mobile.csv")
n<- nrow(df)
last2 <- df[n:(n-1), 2]
w <- c(5,2)
sw <- w*last2
Ft_wavg <- round(sum(sw)/sum(w))
Ft_wavg
```

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

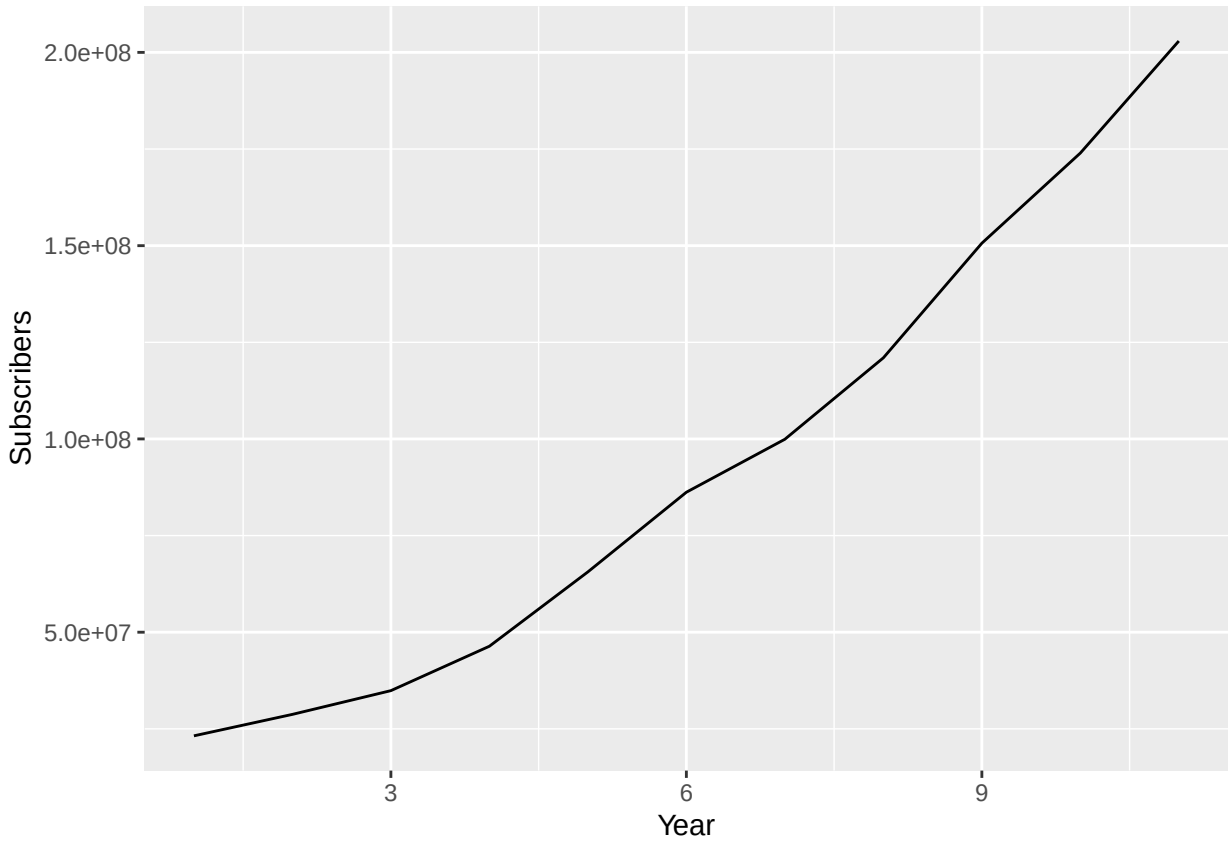
```r
# Exponential Smoothing Forecast
a <- 0.4
df$Ft <- 0
df$E <- 0

df$Ft[1] <- df$Subscribers[1]
for(i in 2:n)
{
  df$Ft[i] <- df$Ft[i-1] + a * df$E[i-1]
  df$E[i] = df$Subscribers[i] - df$Ft[i]

}
Ft_esp <- round(df$Ft[n] + a * df$E[n])
Ft_esp
```

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

```r
# Liner Regression Forecast
ggplot(df, aes(x=Year, y = Subscribers)) + geom_line()
```




```
model <- lm(df$Subscribers ~ df$Year)
summary(model)
```

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

```
Ft_lm <- -15710760 + 18276748 *(n+1)
Ft_lm
```

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

Question 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)?

```
# Weighted Avg
df<- read.csv("mobile.csv")
df$Ft <- 0
df$SqErr <- 0
df$Ft[1] <- df$Subscribers[1]
df$Ft[2] <- df$Subscribers[2]
w <- c(5,2)
for(i in 3:n)
{
  last2 <- df[(i-1): (i-2), 2]
  sw <- w*last2
  df$Ft[i]<- (sum(sw)/sum(w))
  df$SqErr[i] <- (df$Subscribers[i] - df$Ft[i])^2
}
MSE_wavg <- mean(df$SqErr)
MSE_wavg
```

```
## [1] 5.441439e+14
```

```
# Exponential Smoothing model
df<- read.csv("mobile.csv")
a <- 0.4
df$Ft <- 0
df$E <- 0

df$Ft[1] <- df$Subscribers[1]
for(i in 2:n)
{
  df$Ft[i] <- df$Ft[i-1] + a * df$E[i-1]
  df$E[i] = df$Subscribers[i] - df$Ft[i]

}
df$SqErr <- 0
for(i in 1:n)
{
  df$SqErr[i] <- (df$E[i])^2
```

```
}
MSE_esp <- mean(df$SqErr)
MSE_esp
```

```
## [1] 1.473838e+15
```

```
#Linear Regression model
df<- read.csv("mobile.csv")
df$Ft <- 0
df$SqErr <- 0
for(i in 1:n)
{
  df$Ft[i] <- (-15710760) + 18276748 *(i)
  df$SqErr[i] <- (df$Subscribers[i] - df$Ft[i])^2

}
MSE_lm <- mean(df$SqErr)
MSE_lm
```

```
## [1] 1.265347e+14
```

Linear Regression Trendline model has the smallest Mean squared error(MSE). Thus, it is better than other prediction models.

Question 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.

```
df <- c(Ft_wavg, Ft_esp, Ft_lm)
new_w <- c(1, 2, 4)
new_sw <- new_w*df
avg_Forecast <- sum(new_sw)/sum(new_w)
avg_Forecast
```

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