Assignment2

Mengyue Sun

9/29/2018

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

```
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------------------------------- tidyv
```

```
## v ggplot2 3.0.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.6
## v tidyr   0.8.1     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

```
## -- Conflicts -------------------------------------------------------------------------- tidyverse_c
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
str(USArrests)
```

```
## 'data.frame':    50 obs. of  4 variables:
##  $ Murder  : num  13.2 10 8.1 8.8 9 7.9 3.3 5.9 15.4 17.4 ...
##  $ Assault : int  236 263 294 190 276 204 110 238 335 211 ...
##  $ UrbanPop: int  58 48 80 50 91 78 77 72 80 60 ...
##  $ Rape    : num  21.2 44.5 31 19.5 40.6 38.7 11.1 15.8 31.9 25.8 ...
```

```
USArrests2 <- mutate(USArrests, stateNames = row.names(USArrests) )
#Find outliers in assaults
average <- mean(USArrests2$Assault)
std <- sd(USArrests2$Assault)
outliers <- filter(USArrests2, Assault > 1.5 * std + average)
outliers$stateNames
```

```
## [1] "Florida"        "Maryland"       "North Carolina"
```

We can see that Florida, Maryland, and North Carolina are outliers in terms of assaults.

2.For the same dataset as in (1), is there a correlation between murder and assault, 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(USArrests2$Murder, USArrests2$Assault, method = "spearman")
```

```
## Warning in cor.test.default(USArrests2$Murder, USArrests2$Assault, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  USArrests2$Murder and USArrests2$Assault
## S = 3805.3, p-value = 4.485e-13
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
```

```
## 0.8172735
```

There is a very strong positive correlation between murder and assaults.

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), forecast phone use for the next time period using a simple moving average, a 3-year weighted moving average (with weights of 4 for the most recent year, and 1 for the others), exponential smoothing (alpha of 0.2), and linear regression trendline.

```
library(readxl)
phoneUse <- read_excel("/Users/sunmengyue/Dropbox/2. Northeastern Unviersity/Fall2018/DA 5030 DataMining
#Use a 3-year moving average to predict the next month
n <- nrow(phoneUse)
last3 <- phoneUse[n : (n - 2), 2]
mean(last3$Subscribers)
```

```
## [1] 175848268
```

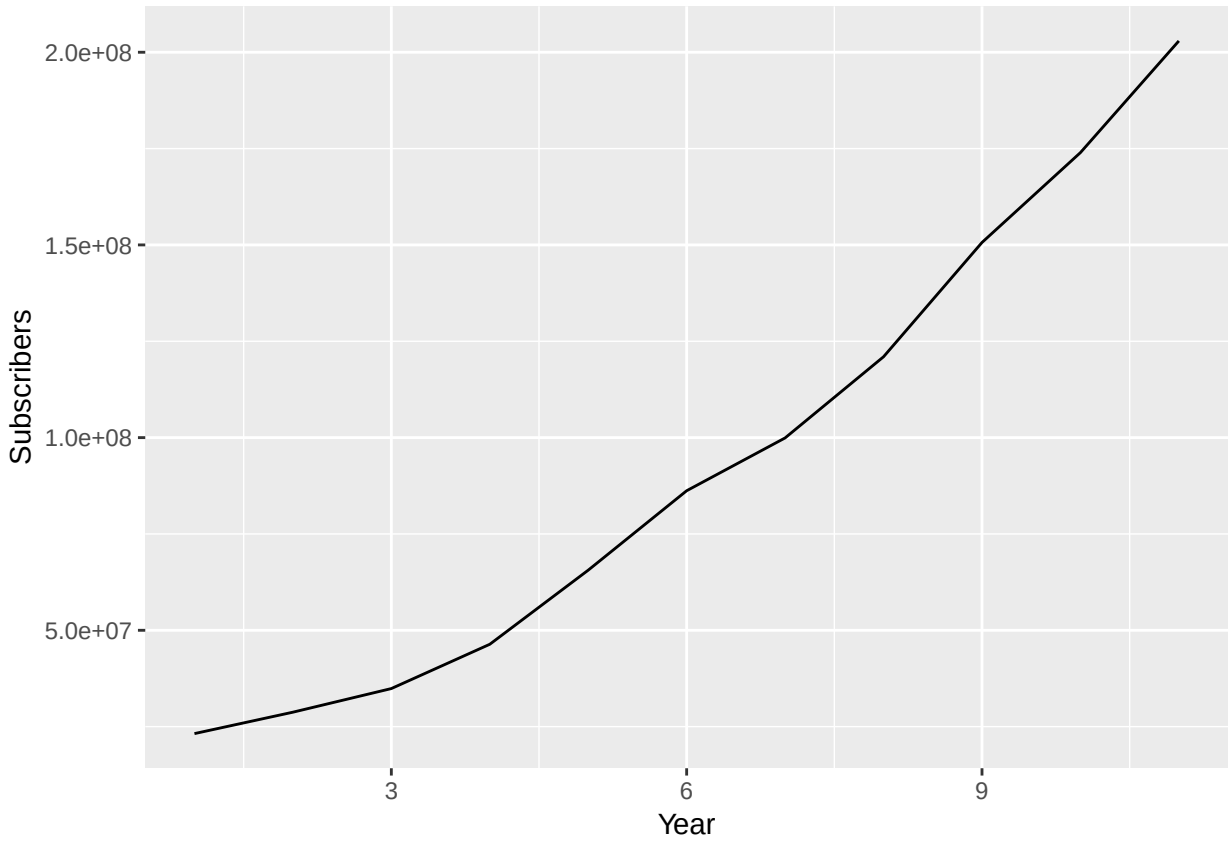
```
# 3-year weighted moving average
w <- c(4, 1, 1)
sw <- last3 * w
F <- sum(sw$Subscribers) / sum(w)
F
```

```
## [1] 189396150
```

```
#exponential smoothing
phoneUse2 <- mutate(phoneUse, Ft = 0, E = 0)
phoneUse2$Ft[1] <- phoneUse2$Subscribers[1]
for (i in 2 : nrow(phoneUse)) {
  phoneUse2$Ft[i] = phoneUse2$Ft[i - 1] + 0.2 * phoneUse2$E[i - 1]
  phoneUse2$E[i] = phoneUse2$Subscribers[i] - phoneUse2$Ft[i]
}
subscriber12 <- phoneUse2$Ft[11] + 0.2 * phoneUse2$E[11]
subscriber12
```

```
## [1] 123744469
```

```
#linear regression trendline
ggplot(phoneUse, aes(x = Year, y = Subscribers)) + geom_line()
```




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

```
##
## Call:
## lm(formula = phoneUse$Subscribers ~ phoneUse$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 .
## phoneUse$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
```

```
# subscriber = -15710760 + 18276748 * year, so at the 12th year:
subscriberYear12 <- -15710760 + 18276748 * 12
subscriberYear12
```

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

4. Calculate the average mean squared error for each model, i.e., use the model to calculate a forecast for

each given time period and then the error.

```
#MSE in the moving average model, mean(last3$Subscribers) is the last 3 year moving average
#add a squared erro column using the mutate() function in dplyr, and save it to the phoneUseMA_SE table
phoneUseMA_SE <- mutate(phoneUse, sqErr = (Subscribers - mean(last3$Subscribers))^2)
phoneUseMA_SE
```

```
## # A tibble: 11 x 3
##     Year Subscribers   sqErr
##    <dbl>       <dbl>   <dbl>
##  1     1    23188171 2.33e16
##  2     2    28745769 2.16e16
##  3     3    34880964 1.99e16
##  4     4    46373266 1.68e16
##  5     5    65605000 1.22e16
##  6     6    86210336 8.03e15
##  7     7    99918621 5.77e15
##  8     8   120980103 3.01e15
##  9     9   150641403 6.35e14
## 10    10   173959368 3.57e12
## 11    11   202944033 7.34e14
```

```
#create a new variable named phoneUseMA_MSE to save the mean square error
phoneUseMA_MSE <-mean(phoneUseMA_SE$sqErr)
phoneUseMA_MSE
```

```
## [1] 1.01743e+16
```

```
#MSE in the 3-year weighted moving average, F is the 3 year weighted moving average
phoneUseWMA_SE <- mutate(phoneUse, sqErr = (Subscribers - F)^2)
phoneUseWMA_SE
```

```
## # A tibble: 11 x 3
##     Year Subscribers   sqErr
##    <dbl>       <dbl>   <dbl>
##  1     1    23188171 2.76e16
##  2     2    28745769 2.58e16
##  3     3    34880964 2.39e16
##  4     4    46373266 2.05e16
##  5     5    65605000 1.53e16
##  6     6    86210336 1.06e16
##  7     7    99918621 8.01e15
##  8     8   120980103 4.68e15
##  9     9   150641403 1.50e15
## 10    10   173959368 2.38e14
## 11    11   202944033 1.84e14
```

```
phoneUseWMA_MSE <- mean(phoneUseWMA_SE$sqErr)
phoneUseWMA_MSE
```

```
## [1] 1.257695e+16
```

```
#MSE in the exponential smoothing method, subscriber12 is the subscribers predicted in the 12th year us
phoneUseES_SE <- mutate(phoneUse, sqErr = (Subscribers - subscriber12)^2)
phoneUseES_SE
```

```
## # A tibble: 11 x 3
##     Year Subscribers   sqErr
```

```
##    <dbl>       <dbl>   <dbl>
##  1     1    23188171 1.01e16
##  2     2    28745769 9.02e15
##  3     3    34880964 7.90e15
##  4     4    46373266 5.99e15
##  5     5    65605000 3.38e15
##  6     6    86210336 1.41e15
##  7     7    99918621 5.68e14
##  8     8   120980103 7.64e12
##  9     9   150641403 7.23e14
## 10    10   173959368 2.52e15
## 11    11   202944033 6.27e15
```

```
phoneUseES_MSE <- mean(phoneUseES_SE$sqErr)
phoneUseES_MSE
```

```
## [1] 4.354656e+15
```

```
#MSE using linear regression trendline, subscriberYear12 is the subscribers predicted in the 12th year
phoneUselr_SE <- mutate(phoneUse, sqErr = (Subscribers - subscriberYear12)^2)
phoneUselr_SE
```

```
## # A tibble: 11 x 3
##     Year Subscribers   sqErr
##    <dbl>       <dbl>   <dbl>
##  1     1    23188171 3.26e16
##  2     2    28745769 3.06e16
##  3     3    34880964 2.85e16
##  4     4    46373266 2.47e16
##  5     5    65605000 1.90e16
##  6     6    86210336 1.38e16
##  7     7    99918621 1.08e16
##  8     8   120980103 6.83e15
##  9     9   150641403 2.81e15
## 10    10   173959368 8.79e14
## 11    11   202944033 4.44e11
```

```
phoneUselr_MSE <- mean(phoneUselr_SE$sqErr)
phoneUselr_MSE
```

```
## [1] 1.549235e+16
```

5. Which model has the smallest mean squared error (MSE)? By comparing the MSE within each model, we know that the exponential smoothing model has the least MSE (4.354656e+15).

6.Calculate a weighted average forecast by averaging out the three forecasts calculated in (3) with the following weights: 3 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.

```
#subscriberYear12 is the subscribers predicted in the 12th year using linear regression,
#subscriber12 is the subscribers predicted in the 12th year using exponential smoothing,
#F is the 3 year weighted moving average
weightSum <- 3 + 2 + 1
weightedAve <- (3 * subscriberYear12 + 2*subscriber12 + F)/weightSum
weightedAve
```

```
## [1] 174619289
```