Assignment_2

Vivekanand Kulkarni 19200231

08/11/2019

1 & 2)

Load in the data as an object called DublinAirport. Notice that you have to skip the first 19 lines contained in the file before beginning to read data.

```
DublinAirport=read.csv("V:\\Study\\R-Programming\\Assignments\\Assignment_2\\mly532.csv", hea
der = TRUE,sep = ",",skip = 19)
str(DublinAirport)
```

```
## 'data.frame':    931 obs. of  12 variables:
##  $ year : int  1941 1941 1942 1942 1942 1942 1942 1942 1942 1942 ...
##  $ month: int  11 12 1 2 3 4 5 6 7 8 ...
##  $ meant: num  6.9 6.5 4.3 2.9 6.3 8.4 10.4 13.1 14.6 14.9 ...
##  $ maxtp: num  14 12.7 11.9 11.6 16.2 16.2 20.9 24.1 22.2 22.3 ...
##  $ mintp: num  -3.1 -3.6 -3.1 -4.3 -6.1 0.8 1.8 1.4 7.2 6.7 ...
##  $ mnmax: num  9.9 9.1 6.9 5.8 9.4 11.9 14.4 18 18.9 18.4 ...
##  $ mnmin: num  3.9 3.9 1.7 0 3.2 4.9 6.3 8.2 10.4 11.4 ...
##  $ rain : num  67.2 41.7 91.9 25.8 76.4 ...
##  $ gmin : num  -5.7 -7.6 -9.5 -10.7 -8.3 -0.4 -0.7 -0.9 2.4 4.6 ...
##  $ wdsp : num  12 12.5 13.1 9 10.7 15.1 12 9.4 13.4 10.8 ...
##  $ maxgt: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ sun  : num  56.1 46.1 72.8 51.4 73.9 ...
```

3) Transform the Column month

Used factor to assign the month columns with month names. And also made use of the built-in constant month.name which is a vector of 12 months.

```
DublinAirport$month<-factor(DublinAirport$month, levels = c('1', "2","3","4","5","6","7","8",
"9","10","11","12"),labels = month.name)
head(DublinAirport)
```

```
##   year    month meant maxtp mintp mnmax mnmin rain  gmin wdsp maxgt   sun
## 1 1941 November   6.9  14.0  -3.1   9.9   3.9 67.2  -5.7 12.0    NA  56.1
## 2 1941 December   6.5  12.7  -3.6   9.1   3.9 41.7  -7.6 12.5    NA  46.1
## 3 1942  January   4.3  11.9  -3.1   6.9   1.7 91.9  -9.5 13.1    NA  72.8
## 4 1942 February   2.9  11.6  -4.3   5.8   0.0 25.8 -10.7  9.0    NA  51.4
## 5 1942    March   6.3  16.2  -6.1   9.4   3.2 76.4  -8.3 10.7    NA  73.9
## 6 1942    April   8.4  16.2   0.8  11.9   4.9 36.9  -0.4 15.1    NA 185.4
```

4)

Use the aggregate function to compute which month has on average the highest and the lowest Precipitation Amount.

```
Mean_rain=aggregate(DublinAirport$rain,list(DublinAirport$month),mean)
names(Mean_rain)[1]<-"Months"
names(Mean_rain)[2]<-"Mean of each month"
Mean_rain_sorted<- order(Mean_rain[,2],decreasing = TRUE)
print(paste0("The month with Highest rain Precipitation Amount is:",month.name[Mean_rain_sort
ed[1]]))
```

```
## [1] "The month with Highest rain Precipitation Amount is:December"
```

```
print(paste0("The month with Lowest rain Precipitation Amount is:",month.name[Mean_rain_sorte
d[12]]))
```

```
## [1] "The month with Lowest rain Precipitation Amount is:February"
```

5)

Creating a new column which contains a factor indicating the season: Winter: December, January, February, Spring: March, April, May, Summer: June, July, August, Autumn: September, October, November

```
DublinAirport$Season<-factor(DublinAirport$month ,levels =c("January", "February","March","Ap
ril","May","June","July","August","September","October","November","December"),labels=c("Wint
er","Winter", "Spring","Spring","Spring", "Summer","Summer","Summer","Autumn","Autumn","Autum
n","Winter"))

head(DublinAirport)
```

```
##   year    month meant maxtp mintp mnmax mnmin rain  gmin wdsp maxgt   sun
## 1 1941 November   6.9  14.0  -3.1   9.9   3.9 67.2  -5.7 12.0    NA  56.1
## 2 1941 December   6.5  12.7  -3.6   9.1   3.9 41.7  -7.6 12.5    NA  46.1
## 3 1942  January   4.3  11.9  -3.1   6.9   1.7 91.9  -9.5 13.1    NA  72.8
## 4 1942 February   2.9  11.6  -4.3   5.8   0.0 25.8 -10.7  9.0    NA  51.4
## 5 1942    March   6.3  16.2  -6.1   9.4   3.2 76.4  -8.3 10.7    NA  73.9
## 6 1942    April   8.4  16.2   0.8  11.9   4.9 36.9  -0.4 15.1    NA 185.4
##   Season
## 1 Autumn
## 2 Winter
## 3 Winter
## 4 Winter
## 5 Spring
## 6 Spring
```

6)

Assign to the DublinAiport object the classes WeatherData and data.frame.

```
class(DublinAirport)<-c('WeatherData','data.frame')
class(DublinAirport)
```

```
## [1] "WeatherData" "data.frame"
```

7)

Write an S3 summary method for an object of class WeatherData which produces the following statistical summaries the rain, maxtp, mintp, maxgt variables split by season: mean, standard deviation, minimum, maximum. Ignore the missing values in the calculations. Test your function on the DublinAirport data set and comment your fndings.

```
summary.WeatherData<-function(x)
{
mean_rain <- aggregate(rain~Season, data = x, mean,na.rm= TRUE)
names(mean_rain)[1]<-"Mean per season"
max_rain <- aggregate(rain~Season, data = x,max,na.rm= TRUE)
names(max_rain)[1]<-"Max  per season"
sd_rain<- aggregate(rain~Season, data=x,sd,na.rm= TRUE)
names(sd_rain)[1]<-"Standard Deviation  per season"
min_rain<- aggregate(rain~Season, data=x,min,na.rm=TRUE)
names(min_rain)[1]<-"Min  per season"

print(mean_rain)
print(max_rain)
print(sd_rain)
print(min_rain)

mean_maxtp <- aggregate(maxtp~Season, data = x, mean,na.rm= TRUE)
names(mean_maxtp)[1]<-"Mean  per season"
max_maxtp <- aggregate(maxtp~Season, data = x,max,na.rm= TRUE)
names(max_maxtp)[1]<-"Max  per season"
sd_maxtp <- aggregate(maxtp~Season, data=x,sd,na.rm= TRUE)
names(sd_maxtp)[1]<-"Standard Deviation  per season"
min_maxtp <- aggregate(maxtp~Season, data=x,min,na.rm=TRUE)
names(min_maxtp)[1]<-"Min  per season"


print(mean_maxtp)
print(max_maxtp)
print(sd_maxtp)
print(min_maxtp)

mean_mintp <- aggregate(mintp~Season, data = x, mean,na.rm= TRUE)
names(mean_mintp)[1]<-"Mean  per season"
max_mintp <- aggregate(mintp~Season, data = x,max,na.rm= TRUE)
names(max_mintp)[1]<-"Max  per season"
sd_mintp <- aggregate(mintp~Season, data=x,sd,na.rm= TRUE)
names(sd_mintp)[1]<-"Standard Deviation per season"
min_mintp <- aggregate(mintp~Season, data=x,min,na.rm=TRUE)
names(min_mintp)[1]<-"Min  per season"


print(mean_mintp)
print(max_mintp)
print(sd_mintp)
print(min_mintp)

mean_maxgt <- aggregate(maxgt~Season, data = x, mean,na.rm= TRUE)
names(mean_maxgt)[1]<-"Mean  per season"
max_maxgt <- aggregate(maxgt~Season, data = x,max,na.rm= TRUE)
names(max_maxgt)[1]<-"Max  per season"
sd_maxgt<- aggregate(maxgt~Season, data=x,sd,na.rm= TRUE)
names(sd_maxgt)[1]<-"Standard Deviation per season"
min_maxgt<- aggregate(maxgt~Season, data=x,min,na.rm=TRUE)
names(min_maxgt)[1]<-"Min  per season"

print(mean_maxgt)
print(max_maxgt)
```

```
print(sd_maxgt)
print(min_maxgt)
}
summary(DublinAirport)
```

```
##   Mean per season     rain
## 1          Winter 63.89017
## 2          Spring 53.54915
## 3          Summer 63.92165
## 4          Autumn 70.20948
##   Max  per season  rain
## 1          Winter 217.0
## 2          Spring 151.8
## 3          Summer 189.9
## 4          Autumn 185.8
##   Standard Deviation  per season     rain
## 1                         Winter 32.87200
## 2                         Spring 27.30880
## 3                         Summer 35.43135
## 4                         Autumn 38.11011
##   Min  per season rain
## 1          Winter  4.7
## 2          Spring  3.6
## 3          Summer  4.0
## 4          Autumn  3.6
##   Mean  per season    maxtp
## 1           Winter 12.73761
## 2           Spring 17.16239
## 3           Summer 23.02511
## 4           Autumn 18.02198
##   Max  per season maxtp
## 1          Winter  17.1
## 2          Spring  23.5
## 3          Summer  28.7
## 4          Autumn  25.1
##   Standard Deviation  per season    maxtp
## 1                         Winter 1.638288
## 2                         Spring 2.798734
## 3                         Summer 1.787048
## 4                         Autumn 3.038054
##   Min  per season maxtp
## 1          Winter   4.8
## 2          Spring   9.9
## 3          Summer  18.4
## 4          Autumn  12.1
##   Mean  per season      mintp
## 1           Winter -3.1931624
## 2           Spring -0.5871795
## 3           Summer  5.9303030
## 4           Autumn  1.2551724
##   Max  per season mintp
## 1          Winter   1.8
## 2          Spring   6.9
## 3          Summer  10.0
## 4          Autumn   7.5
##   Standard Deviation per season    mintp
## 1                         Winter 2.369322
## 2                         Spring 2.401950
## 3                         Summer 1.735196
## 4                         Autumn 2.799624
##   Min  per season mintp
## 1          Winter -12.2
```

```
## 2          Spring  -7.9
## 3          Summer   0.7
## 4          Autumn  -8.4
##   Mean  per season    maxgt
## 1           Winter 53.04933
## 2           Spring 45.57333
## 3           Summer 39.49333
## 4           Autumn 47.23661
##   Max  per season maxgt
## 1          Winter    80
## 2          Spring    66
## 3          Summer    56
## 4          Autumn    73
##   Standard Deviation per season    maxgt
## 1                         Winter 8.767892
## 2                         Spring 7.426136
## 3                         Summer 6.263642
## 4                         Autumn 7.875278
##   Min  per season maxgt
## 1          Winter    35
## 2          Spring    28
## 3          Summer    27
## 4          Autumn    27
```

Created an S3 plot method for the class WeatherData that produces the following plots. (1) Plot of the monthly Air Temperature (C) (maxtp, mintp). (2) Plot of the Precipitation Amount (mm) (rain). (3) Plot of the Highest Gust (knot) (maxgt). The user must be able to decide which years to plot. By default it will use the data from 2015 until 2018. The user must be able to decide which plot to draw (i.e, only one of the three, two of the three, or all three plots). By default the function will create all three plots. The plots must be on a single panel. The plots must have meaningful labels and/or titles, and a legend if needed. Test your function on the DublinAirport data set.

```
DublinAirport<- na.omit(DublinAirport)
library(ggplot2)
library(gridExtra)
library(dplyr)
```

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

```
## The following object is masked from 'package:gridExtra':
##
##     combine
```

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

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

```
plot.WeatherData <- function(Dframe, yearfrom = 2015, yearto = 2018, number_of_plots = 3){

  # Selecting the data based on the range of years inputted to the function

  plot_data <- Dframe[Dframe$year >= yearfrom & Dframe$year <= yearto, ]

  # Plot of the monthly Air Temperature (C) (maxtp, mintp).
  plot1 <- ggplot(plot_data, aes(x = year, y = maxtp, colour = month)) +
  geom_jitter(alpha = 0.50) +
  ggtitle(" Plot of the monthly Air Temperature (C) (maxtp, mintp).") +
  xlab("Years") +
  ylab("Maximum and Minimum Temperatures") +
  theme(axis.text.x = element_text(angle = 90, face = "bold", size = 10)) +
  geom_jitter(aes(x = year, y = mintp, colour = month))

  # Plot of the Precipitation Amount (mm) (rain).
  plot2 <- ggplot(plot_data, aes(x = year, y = rain, colour = month)) +
  geom_jitter(alpha = 0.50) +
  ggtitle(" Plot of the Precipitation Amount (mm) (rain).") +
  xlab("Years") +
  ylab("Precipitation Amount (mm) in terms of Rain") +
  theme(axis.text.x = element_text(angle = 90, face = "bold", size = 10))

  # Plot of the Highest Gust (knot) (maxgt).
  plot3 <- ggplot(plot_data, aes(x = year, y = maxgt, colour = month  )) +
  geom_jitter(alpha = 0.50) +
  ggtitle(" Plot of the Highest Gust (knot) (maxgt).") +
  xlab("Year") +
  ylab("Highest Gust (knot)") +
  theme(axis.text.x = element_text(angle = 90, face = "bold", size = 10))

  # Comparing the number_of_plots variable, and retrieving according to the request.
  if(number_of_plots == 1)
  {
    #Display plot1 if number_of_plots is 1
    plot1
  }
  else if(number_of_plots == 2)
  {
    #Display plot1 & plot2 if number_of_plots is 2
    require(gridExtra)
    grid.arrange(plot1, plot2,nrow =1, ncol = 2)
  }
  else if(number_of_plots == 3)
  {
    # By Default all the 3 plots are to be Displayed.
    # Display plot1, plot2 & plot3 if the number_of_plots is 3
    require(gridExtra)
    grid.arrange(plot1, plot2, plot3, nrow =2, ncol = 2)
  }
}

# The plot function is called and defined for the WeatherData Class
# Arguments to plot() functions are
# Dframe --> The Data Frame, in this case DublinAirport
# yearfrom --> The year to begin with
# yearto --> Till which year the data should be selected.
```

```
# number_of_plots --> Number of plots that are to be  displayed among the 3 available plots.
 By default all 3 plots are displayed.

# For the demonstration purpose, Dframe = DublinAirport,  yearfrom = 2000, yearto = 2018, num
ber_of_plots = 3
plot(DublinAirport, 2000,2018,3)
```