### ASSIGNMENT 2 ###  
  
## Please find all the answers summarized in the word file "Answers to AS2 - Daniele Melotti" ##

### 1) Import and Preprocess Data  
  
## a) Import the datasets.  
#install.packages("data.table", dependencies = T)  
#install.packages('plyr', repos = "http://cran.us.r-project.org")  
options(repos = list(CRAN="http://cran.rstudio.com/"))  
require(plyr)

## Loading required package: plyr

## Warning: package 'plyr' was built under R version 4.0.5

require(data.table)

## Loading required package: data.table

## Warning: package 'data.table' was built under R version 4.0.5

require(dplyr)

## Loading required package: dplyr

## Warning: package 'dplyr' was built under R version 4.0.5

## Registered S3 methods overwritten by 'tibble':  
## method from   
## format.tbl pillar  
## print.tbl pillar

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':  
##   
## between, first, last

## The following objects are masked from 'package:plyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

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

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

d1 <- fread("https://bit.ly/3c4AHbL", colClasses = list(character = 1:5, numeric = 6:12))  
d2 <- fread("https://bit.ly/3nZicL2", colClasses = list(character = 1:5, numeric = 6:12))  
  
## b) Take a look at the 1999 data.  
# Print the dimensions:  
dim(d1)

## [1] 117421 12

# Print the first 3 rows.  
head(d1, 3)

## X..RD Action.Code State.Code County.Code Site.ID Parameter POC  
## 1: RD I 01 027 0001 88101 1  
## 2: RD I 01 027 0001 88101 1  
## 3: RD I 01 027 0001 88101 1  
## Sample.Duration Unit Method Date Sample.Value  
## 1: 7 105 120 19990103 NA  
## 2: 7 105 120 19990106 NA  
## 3: 7 105 120 19990109 NA

## c) Using the 1999 data, print the summary statistics of the variable with summary().  
summary(d1$Sample.Value)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.00 7.20 11.50 13.74 17.90 157.10 13217

# (i) Compute the number of NAs using table() and is.na(), then divide the numbers by the total number of observations.  
table(is.na(d1$Sample.Value))

##   
## FALSE TRUE   
## 104204 13217

table(is.na(d1$Sample.Value))/sum(table(is.na(d1$Sample.Value)))

##   
## FALSE TRUE   
## 0.8874392 0.1125608

# (ii) What is the percentage of the PM2.5 observations that are missing (round up to 3 decimal places)?  
paste(round(sum(is.na(d1$Sample.Value) == T)/sum(table(is.na(d1$Sample.Value)))\*100, 3), "%")

## [1] "11.256 %"

## d) Bind the 1999 data and 2012 data and assign the aggregated data to an object called ‘pm’. Then, subset the years from the Date variable and convert it into a factor variable called ‘year’.  
pm <- rbind(d1, d2)  
  
# I did not just model the Date variable and rename it because it will be useful later (around 2j)), so I simply created a new variable year:  
pm$year <- substr(pm$Date, 1, 4)  
pm$year <- as.factor(pm$year)  
  
## e) Rename the Sample.Value variable to PM.  
pm <- pm %>%  
 rename(PM = Sample.Value)

### 2) Data Exploration with Visualization using ggplot2  
  
## a) Set the seed at 2021 and draw 1,000 randomly selected samples from the data (i.e., pm) using the sampling function in dplyr package.  
set.seed(2021)  
sample <- sample\_n(pm, 1000,na.rm = T)  
  
## b) Create boxplots of all monitor values in 1999 and 2012 using the randomly selected sampled data.  
# Taking the log of the PM values (with base 2):  
sample$PM <- log(sample$PM, 2)

## Warning: NaNs produced

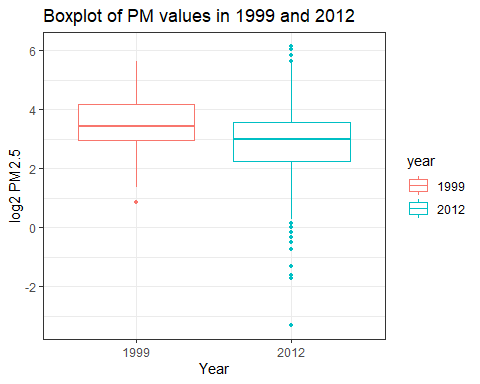
# Creating the box plots:  
require(ggplot2)

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.0.5

box <- ggplot(subset(sample), aes(x = year, y = PM, col = year))  
  
box +  
 geom\_boxplot(outlier.size = 1) +  
 labs(title = "Boxplot of PM values in 1999 and 2012",  
 x = "Year",  
 y = "log2 PM2.5") +  
 theme\_bw()

## Warning: Removed 89 rows containing non-finite values (stat\_boxplot).



## c) Describe what you observe in terms of the means and variances of the observations in 1999 and 2012?  
# If the distribution was normal, we would be able to say that the median shown on the boxplots is equal to the mean (and the mode as well). However, the PM data is probably not normally distributed, so we may only discuss about the median. If we look at the median value of the boxplots, we can see that PM air pollution in the US was a little higher in 1999 than in 2012, on average. The difference is more visible when we compare the two upper quartiles, as we see that the one for the 1999 data is well above the one for 2012 data. The same is true in the case of the lower quartile. However, we can also see that there are way more outliers for the 2012 data; moreover, the whiskers are a little longer when compared to those of the 1999 data. These features can be taken as a sign that the variance is higher in the case of 2012 data.

## d) Identify a monitor in New York State that has data in 1999 and 2012.  
# Subsetting the data to include only the observations from New York and only including the County.Code and the Site.ID using filter(), select() and unique():  
ny <- pm %>%  
 filter(State.Code == 36) %>%  
 select(County.Code, Site.ID, year) %>%  
 unique()  
  
## e) Create a new variable called Site.Code that combines the county code and the site ID into a single string by using paste() with "." as the separator.  
ny$Site.Code <- paste(ny$County.Code, ny$Site.ID, sep = ".")  
  
## f) Find the intersection of the sites in between 1999 and 2012 which gives us the list of monitors in NY operated both in 1999 and 2012 using split() and intersect().  
monitor <- split(ny$Site.Code, ny$year)  
inter <- intersect(monitor$"1999", monitor$"2012")  
  
## g) Write a block of code to identify the monitor in NY state that had the most data using mutate(), filter(), group\_by(), summarize() and arrange().  
pm %>%  
 mutate(Site.Code = paste(County.Code, Site.ID, sep = "." )) %>%  
 filter(Site.Code %in% inter) %>%  
 group\_by(Site.Code) %>%  
 summarize(n = n()) %>%  
 arrange(desc(n))

## Warning: `...` is not empty.  
##   
## We detected these problematic arguments:  
## \* `needs\_dots`  
##   
## These dots only exist to allow future extensions and should be empty.  
## Did you misspecify an argument?

## # A tibble: 10 x 2  
## Site.Code n  
## <chr> <int>  
## 1 101.0003 527  
## 2 013.0011 213  
## 3 031.0003 198  
## 4 001.0005 186  
## 5 067.1015 153  
## 6 063.2008 152  
## 7 029.0005 94  
## 8 001.0012 92  
## 9 005.0080 92  
## 10 085.0055 38

## h) Subset the data (i.e. pm) that contains observations from the monitor we just identified (State.Code = 36 & County.Code = 101 & Site.ID = 0003) and assign the subset data to an object called pmsub.  
pmsub <- subset(pm, State.Code == "36" & County.Code == "101" & Site.ID == "0003")  
  
## i) Using the lubridate package, convert the Date variable into a date object and create a variable called yday containing info on day of the year using yday().  
require(lubridate)

## Loading required package: lubridate

## Warning: package 'lubridate' was built under R version 4.0.5

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':  
##   
## hour, isoweek, mday, minute, month, quarter, second, wday, week,  
## yday, year

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

pmsub$Date <- as.character(pmsub$Date)  
pmsub$Date <- as\_date(pmsub$Date, format = "%Y%m%d")  
  
pmsub$yday <- yday(pmsub$Date)  
  
## j) Draw a scatter plot by mapping the year-day variable on the x-axis, PM2.5 level on the y-axis separately for 1999 and 2012. Make sure to label the x-axis, separate the plots using the facet function and use the base white theme.  
sct <- ggplot(pmsub, aes(x = yday, y = PM))  
  
sct +  
 geom\_point() +  
 labs(x = "Day of the Year", y = "PM") +  
 facet\_wrap(~ year, ncol = 2) +  
 theme\_bw()

## Warning: Removed 45 rows containing missing values (geom\_point).

