CSI 2300: Intro to Data Science

In-Class Exercise 08: Exploratory Data Analysis

1. The Texas Water Development Board (TWDB) is a Texas state agency that provides a variety of water data resources. In particular, its Water Science and Conservation group maintains Water Data for Texas, which provides data on water reservoirs across Texas. This problem involves their reservoir dataset. We will need to read in this data directly to RStudio from "https://www.waterdatafortexas.org/reservoirs/statewide.csv". There are 29 rows of non-data related header at the top of the file. Make sure to disregard this when inputing the data. Show the code needed to accomplish this.

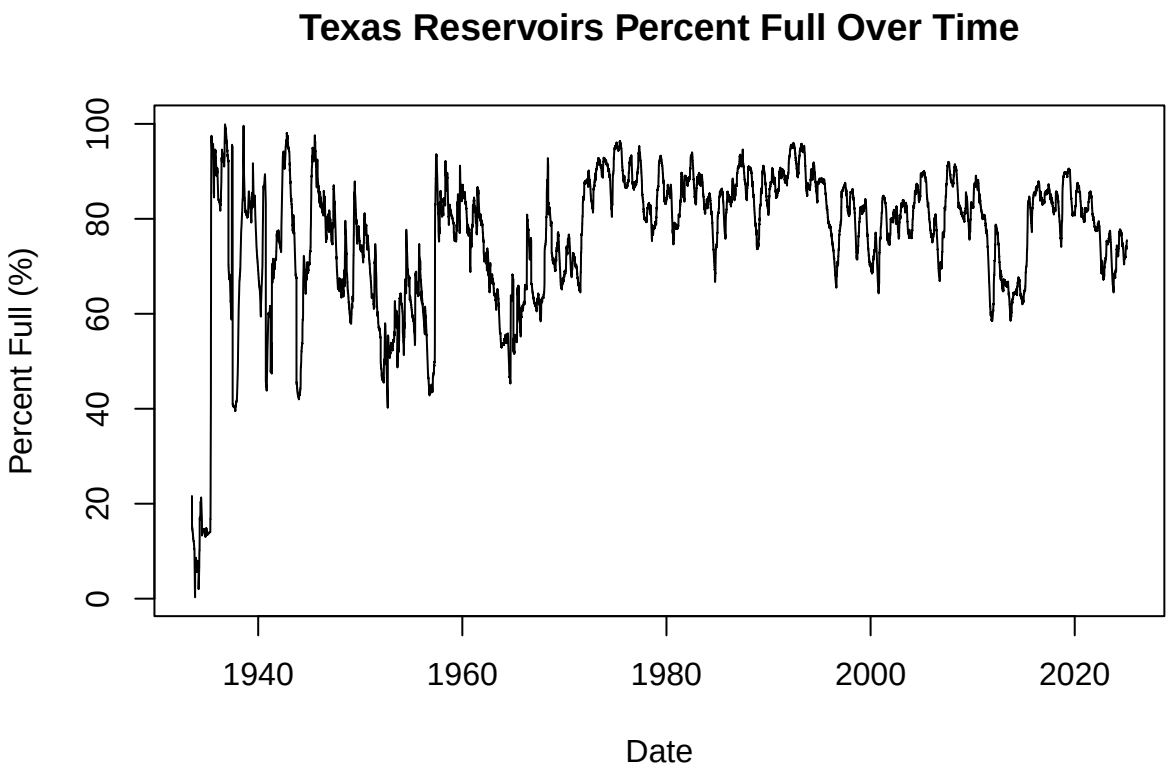
```
www <-"https://www.waterdatafortexas.org/reservoirs/statewide.csv"
water <- read.csv(www, header = TRUE, skip = 29)
```

2. How many observations are there in the dataset? How many variables are there, and what are their default names? Please include the R commands used (if any) to find out this information. If no specific commands were used, then note that.

```
#How many observations are there in the dataset? How many variables are there, and what are their defau

# Number of observations and variables
dim(water)
# [1] 33465     5

# Default variable names
names(water)
# [1] "date"                  "reservoir_storage"     "conservation_storage"
# [4] "percent_full"          "conservation_capacity"
```

3. The `percent_full` variable is the ratio of conservation storage to conservation capacity expressed as a percentage. Visualize how full Texas reservoirs have been since the data began to be recorded. In other words, plot the percent full over time. What are your observations about this data?

```
# Convert date to Date format
water$date <- as.Date(water$date)

# Plot percent_full over time
plot(water$date, water$percent_full, type = "l", xlab = "Date", ylab = "Percent Full (%)", main = "Texas
```



There might be seasonal patterns, with levels rising and falling at certain times of the year. Over the long term, there could be a trend of decreasing or increasing reservoir levels, depending on the data.

Over the following four questions, we will be incrementally creating a plot of the yearly percent full from 2016 - 2020, overlaying each new year on top of the others.

4. First, filter the data to just contain the specified years.

```
water2020 <- water[water$date >= "2016-01-01" & water$date <= "2016-12-31", ]

#do the same as above but loop between the years of 2016 to 2019
water2019 <- water[water$date >= "2019-01-01" & water$date <= "2019-12-31", ]
water2018 <- water[water$date >= "2018-01-01" & water$date <= "2018-12-31", ]
water2017 <- water[water$date >= "2017-01-01" & water$date <= "2017-12-31", ]
water2016 <- water[water$date >= "2016-01-01" & water$date <= "2016-12-31", ]
```

5. Run the following command:

```
#plot(1:365, seq(50, 100, len=365), type = "n", xlab = "", ylab = "")
```

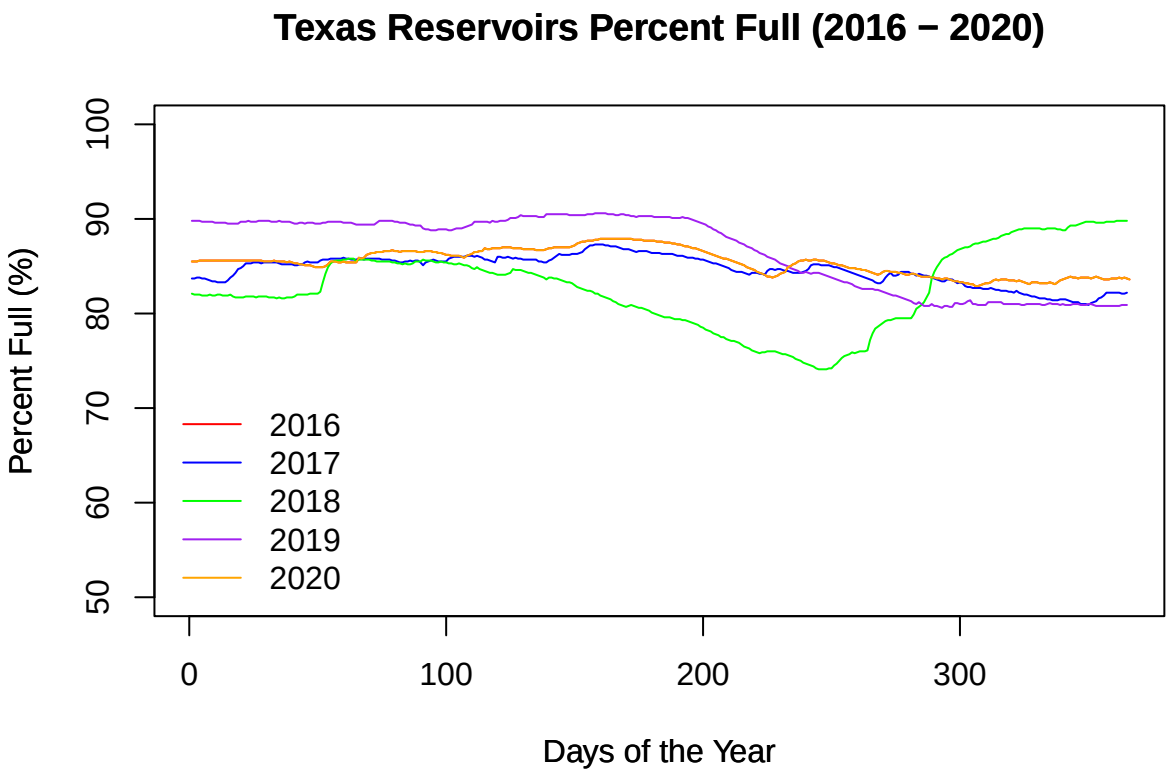
This will create a blank plot where the year in days is on the x-axis, and the y-axis will contain the range of percentages that should be present in the data.

6. Now, using the `line()` command, overlay the percent full for each of the years (2016 - 2020) with each year being a different line and different color. *Note: `for` loops could be useful here but are not required because we have not covered these in class yet.

```r
# Create a blank plot
plot(1:365, seq(50, 100, len=365), type = "n", xlab = "Days of the Year", ylab = "Percent Full (%)")

# Define colors for each year
my_colors <- c("red", "blue", "green", "purple", "orange")

# Overlay lines for each year
lines(1:nrow(water2016), water2016$percent_full, col = my_colors[1])
lines(1:nrow(water2017), water2017$percent_full, col = my_colors[2])
lines(1:nrow(water2018), water2018$percent_full, col = my_colors[3])
lines(1:nrow(water2019), water2019$percent_full, col = my_colors[4])
lines(1:nrow(water2020), water2020$percent_full, col = my_colors[5])
```

7. Now this data can be difficult to understand without some form of label information. Add label information and a legend to this plot in order to help the reader understand what is being displayed.

```r
# Create a blank plot with the correct dimensions
plot(1:365, seq(50, 100, len = 365), type = "n", xlab = "Days of the Year", ylab = "Percent Full (%)", m

# Define colors for each year
my_colors <- c("red", "blue", "green", "purple", "orange")

# Overlay lines for each year
lines(1:nrow(water2016), water2016$percent_full, col = my_colors[1])
lines(1:nrow(water2017), water2017$percent_full, col = my_colors[2])
lines(1:nrow(water2018), water2018$percent_full, col = my_colors[3])
lines(1:nrow(water2019), water2019$percent_full, col = my_colors[4])
lines(1:nrow(water2020), water2020$percent_full, col = my_colors[5])

# Add a legend
legend("bottomleft", legend = c("2016", "2017", "2018", "2019", "2020"), col = my_colors, lty = 1, bty =

# Add a title
title("Texas Reservoirs Percent Full (2016 - 2020)", xlab = "Days of the Year", ylab = "Percent Full (%
```



Each year has a unique pattern, with some years showing higher reservoir levels than others. Comparing the years, you can see if reservoir levels are generally increasing, decreasing, or staying the same over time.

8. One of the ways you can help identify patterns in time series data is to create a "smoother," such as the lowess smoother. The lowess smoother will fit a line to noisy, sparse, or weak data to improve your ability to see patterns. The following code will create a plot and add a few lines to represent various smoothers:

```
# Filter data for the last 20 years
water20 <- water[water$date > "2000-01-01", ]
date20 <- as.Date(water20$date)

# Plot percent_full over time
plot(date20, water20$percent_full, type = "l", xlab = "Year", ylab = "Percent Full (%)")

# Add lowess smoothers with different f values
smooth1 <- lowess(water20$percent_full, f = 0.1)
lines(date20, smooth1$y, lwd = 2, col = 4)  # col = 4 is blue

smooth2 <- lowess(water20$percent_full, f = 0.5)
lines(date20, smooth2$y, lwd = 2, col = 2)  # col = 2 is red

smooth3 <- lowess(water20$percent_full, f = 0.05)
lines(date20, smooth3$y, lwd = 2, col = "green")
```

```r
# Add a legend
legend("bottomleft", legend = c("f = 0.1", "f = 0.5", "f = 0.05"), col = c(4, 2, "green"), lty = 1, lwd

# Add a title
title("Texas Reservoirs Percent Full (2000 - 2020)", xlab = "Year", ylab = "Percent Full (%)")
```

Given this visualization, do learn anything about the most recent 20 years of observations? What effect does increasing the `f` argument in the `lowess` call have? Which value of `f` would you choose to use to display the data?

The plot shows reservoir levels over the last 20 years, with smoothed lines added to help identify trends.