STA404/504 HW7

Ethan Gutknecht

10/17/2022

Learning Objectives:

- Forming a data cleaning plan, break it into smaller tasks and complete it step by step.
- Reading in data with poorly structured heading.
- Restructuring specific columns from wide format to tall format.
- Work with time formatting variables and strings.
- Documenting a fully reproducible data cleaning/visualization process in a scripted language.

Assignment Description:

The dataset "Weekly_US_Crude_Oil.xlsx" on canvas contains the weekly U.S. field production of crude oil (Thousand Barrels per day). The data is from the first week of 1983 to the second week of February 2022. The goal of this assignment is to **clean this data to create time series line plots**. The entire data cleaning process must be conducted in R, no cleaning "by hand" may be done in excel before loading the data to R.

Create a line plot that displays both the weekly and quarterly (calculate the average within each quarter) oil production line plots in one picture. You may use different colors for the two lines and the audience should be able to distinguish the two lines and see the overlap clearly.

One Possible Problem-Solving Process:

The data is very messy and improperly structured. The following is one possible way to deal with the problem.

Step 1: Clean and restructure the data in R using the "**dplyr**", "**tidyr**", "**stringr**", and "**lubridate**" packages so that the cleaned data is saved as data frame that has only two columns: date and production.

The date column can be converted from a character to a POSIX format (POSIX dates consist of the year, followed by the month and day, separated by slashes or dashes) using a function from the "**lubridate**" package. You may open the data file in a spreadsheet editor like excel to take a look at the structural issues, but all the work about data cleaning must be done in R. Others will be able to run your code and get the cleaned data, without editing the code, except for changing the working directory.

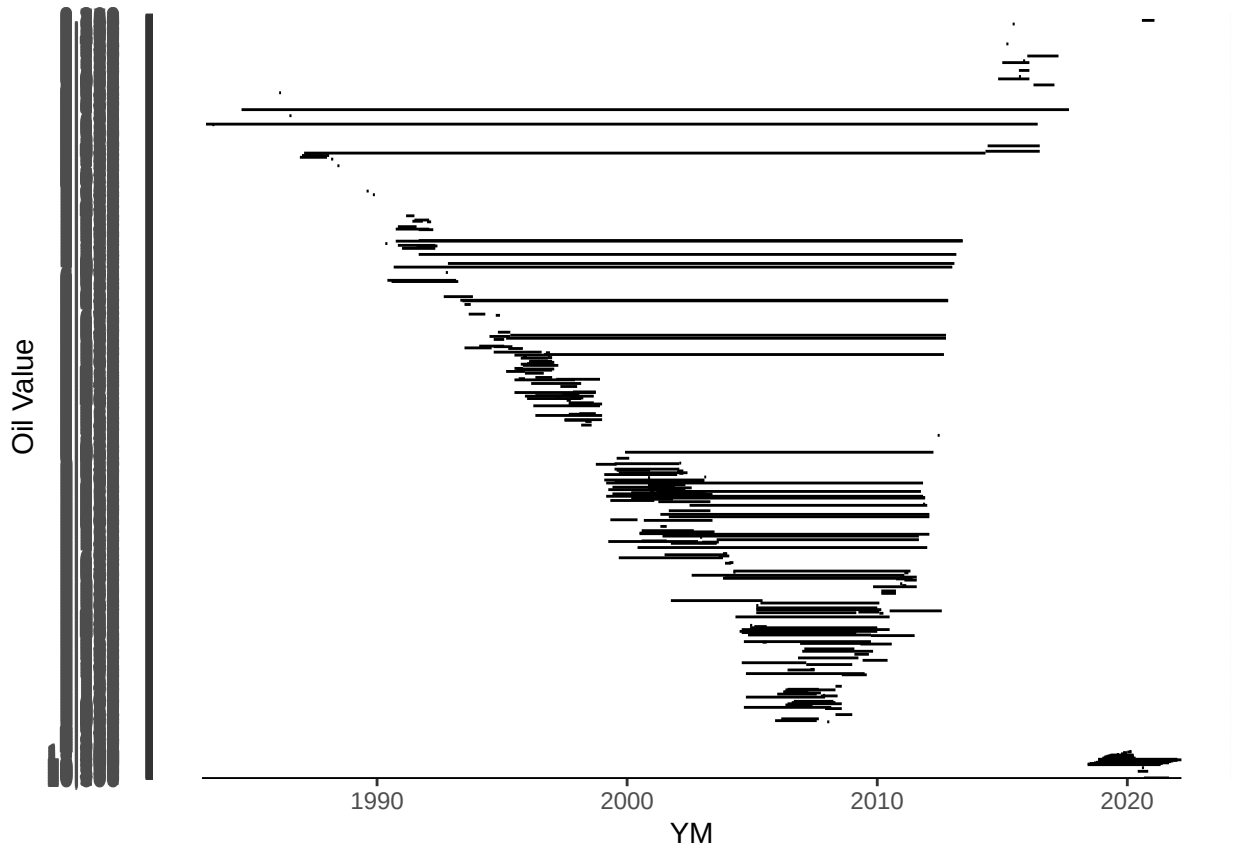
```
# Remove rows that don't matter
oilDataCleaned <- oilData[-c(1,2), ]

# Change column names so it is easier to pivot
colnames(oilDataCleaned) <- c("Year-Month", "Week 1", "Value 1", "Week 2",
                              "Value 2", "Week 3", "Value 3", "Week 4",
```

```
                              "Value 4", "Week 5", "Value 5")

# Pivot to longer
oilDataCleaned2 <- oilDataCleaned %>%
  pivot_longer(cols = c(2,4,6,8,10), values_drop_na = TRUE,
               names_prefix = "Week ", names_to = c("Week"),
               values_to = "Date") %>%
  pivot_longer(cols = c(2,3,4,5,6), values_drop_na = TRUE,
               names_prefix = "Value ", names_to = c("Value"),
               values_to = "Oil Value")%>%
  mutate(YM=ym(`Year-Month`)) %>%
  select(YM, Date, `Oil Value`)


# Show Data
oilDataCleaned2
```

```
## # A tibble: 10,201 x 3
##    YM         Date   `Oil Value`
##    <date>     <chr>  <chr>
##  1 1983-01-01 01/07  8,634
##  2 1983-01-01 01/07  8,634
##  3 1983-01-01 01/07  8,634
##  4 1983-01-01 01/07  8,634
##  5 1983-01-01 01/07
##  6 1983-01-01 01/14  8,634
##  7 1983-01-01 01/14  8,634
##  8 1983-01-01 01/14  8,634
##  9 1983-01-01 01/14  8,634
## 10 1983-01-01 01/14
## # ... with 10,191 more rows
```

Step 2: Use appropriate **ggplot()** code to make plots. You will need to utilize some appropriate **ggplot()** options or other code for data manipulation, that have not been discussed in class.

```
ggplot(data = oilDataCleaned2) +
  geom_line(aes(y = `Oil Value`, x = YM))
```



Some Hints:

1. To read in the excel file, load the package **readxl**, and use the function **read_excel()**. The header of this data is two rows of poorly formatted labels. When you read in the data you may want to use the **"skip="** option to skip the data you don't need.
2. The first column has the year/month combined, followed by five pairs of columns for the dates and productions associated with weeks 1 through 5 of each month. Each of these five column pairs will need to be moved from wide format to tall format using functions in the "**tidyr**" package.
3. Consider separating the data into two data sets (one for all columns related to dates and the other for all columns related with the productions). Then combine the two together after cleanup. You may search for "Join two tbls together" to find a useful way that combines these two datasets.
4. You may need to do some string handling to get things work. Sometimes you may search online for functions that have not been discussed during class, but can satisfy a specific need.
5. To get the y-axis in a format such as $10,000$, you may use the option "**labels=scales::comma**" in **scale_y_continuous()** similar as the homework 6.
6. To drop the missing values, **drop_na()** is one of the functions you can use.