STA404/504 HW6

Ethan Gutknecht

10/09/2022

Learning Objectives:

- Read in data using **"readxl"** package.
- Work with time formatting variables and strings.
- Documenting a fully reproducible data cleaning/visualization process in a scripted language.

Assignment Description:

The dataset "WCRFPUS2w.xls" 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 last week of February 2022. The goal of this assignment is to read in this data and 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.

When creating the plots, make sure you add **at least one** additional aesthetic feature, that has not been used in class, to the line plot. For example, you may change the shape of the line, color, background, font, theme, etc. You may add different additional aesthetic features for different plots, but it's ok if you only add one additional aesthetic feature in one of the plots.

The dataset is obtained by clicking "Download Data (XLS File)" from U.S. Energy Information Administration: https://www.eia.gov/dnav/pet/hist/LeafHandler.ashx?n=PET&s=WCRFPUS2&f=W Please use this data (either download it directly from the website, or from canvas) to answer the following two questions:

```
#import libraries
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.1.3
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## Warning: package 'tidyr' was built under R version 4.1.3
```

```
## Warning: package 'readr' was built under R version 4.1.3
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

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

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.1.3
```

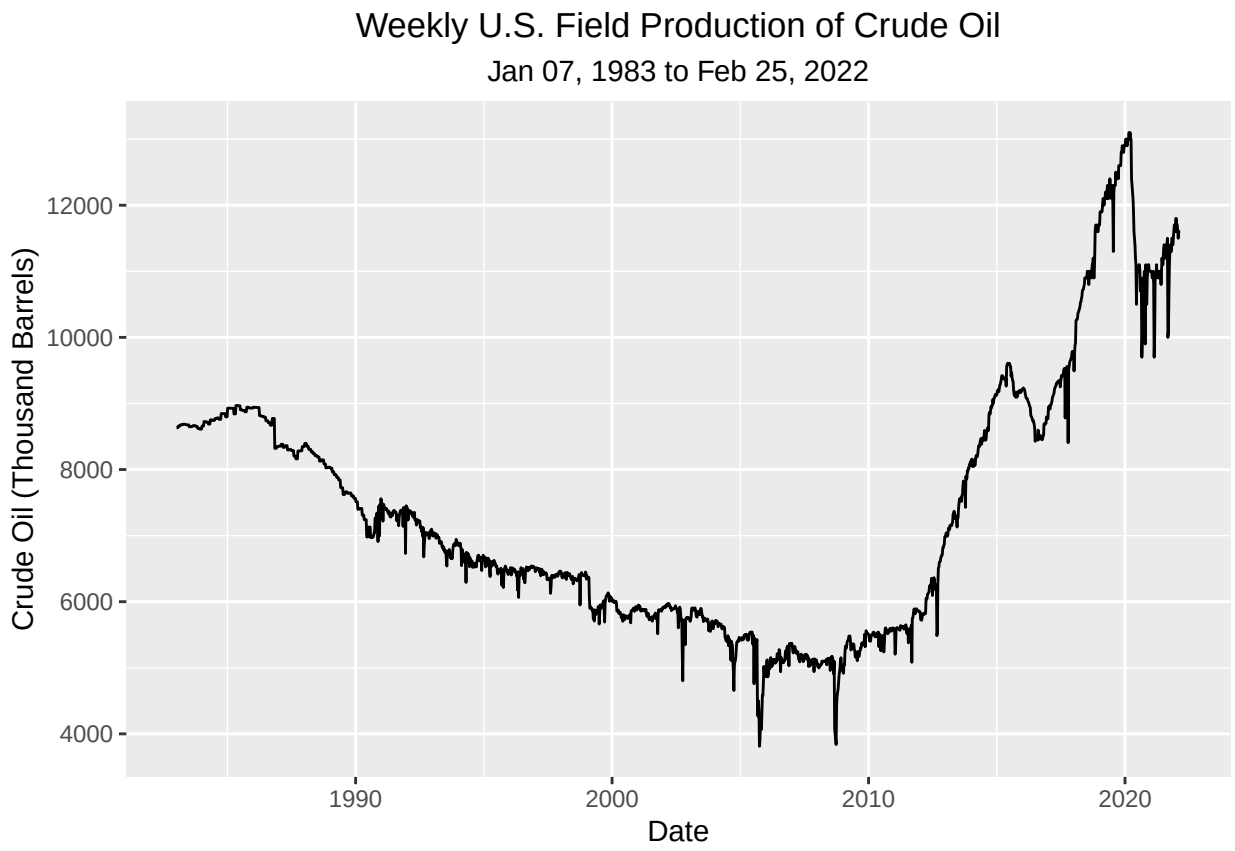
```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
hw6data <- read_xls("WCRFPUS2w.xls", sheet = 2, skip = 2)
hw6data$Date = as_date(hw6data$Date)
```

Question 1

Create a weekly U.S. field production of crude oil line plot which looks similar like the one in this website https://www.eia.gov/dnav/pet/hist/LeafHandler.ashx?n=PET&s=WCRFPUS2&f=W (The plot in the website is interactive, you can just generate a static plot. Make sure the beginning date and ending date and other necessary information is clearly displayed either in the label, or in titles/footnotes. You don't need to make the color, font, other format exactly the way it is in that website. It's fine if the tick marks on the x axis is different from the website. But if you managed to make the tick marks exactly the same as the website, this is considered as the "additional aesthetic feature").

```
# Create graph of crude oil
ggplot(data=hw6data) +
  geom_line(aes(x=`Date`, y=`Weekly U.S. Field Production of Crude Oil  (Thousand Barrels per Day)`)) +
  labs(x="Date", y= "Crude Oil (Thousand Barrels)") +
  ggtitle("Weekly U.S. Field Production of Crude Oil", subtitle="Jan 07, 1983 to Feb 25, 2022") +
  theme(plot.title = element_text(hjust = 0.5), plot.subtitle = element_text(hjust = 0.5))
```



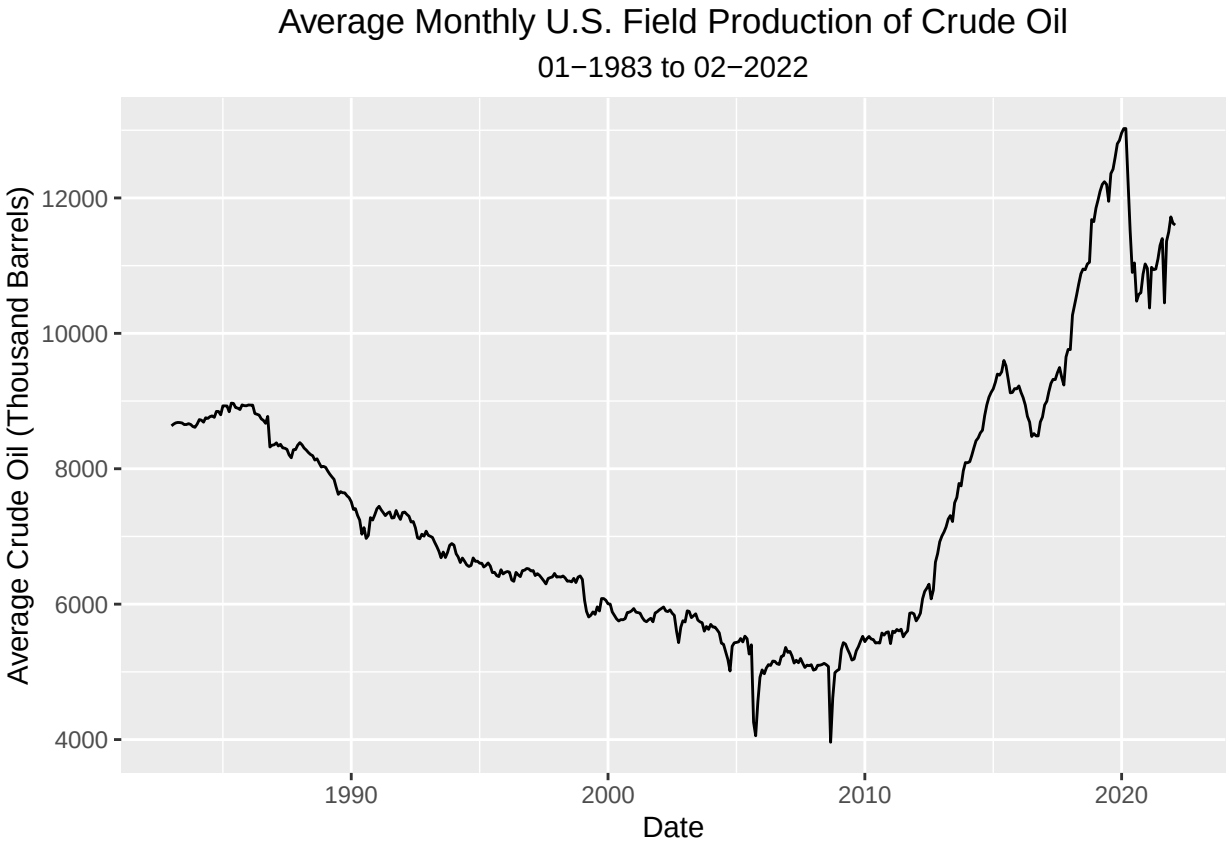
Question 2

Create a monthly U.S. field production of crude oil line plot. You can calculate the average crude oil production for the same month to get the average value. (e.g. if there're 4 or 5 data points of the weekly oil production for a month, you can just average them to get the monthly average value). When creating the plot, you may use the first day (or any specific day) of each month to store the monthly average.

```
# Seperate the data by year and month
# Summarize the data by the mean oil production
# and create a date variable to reference
q2hw6Data <- hw6data %>%
  mutate(month = month(Date), year = year(Date)) %>%
  group_by(year, month) %>%
  summarize(averageForMonth = mean(`Weekly U.S. Field Production of Crude Oil  (Thousand Barrels per Da
            newDate = as.Date(paste(year, month, "01", sep = "-")))
```

```
## `summarise()` has grouped output by 'year', 'month'. You can override using the
## `.groups` argument.
```

```
# Create graph of average oil
ggplot(data=q2hw6Data) +
  geom_line(aes(x=newDate, y=averageForMonth)) +
  labs(x="Date", y= "Average Crude Oil (Thousand Barrels)") +
  ggtitle("Average Monthly U.S. Field Production of Crude Oil", subtitle="01-1983 to 02-2022") +
  theme(plot.title = element_text(hjust = 0.5), plot.subtitle = element_text(hjust = 0.5))
```



Some Hints:
1. To read in the excel file, load the package **readxl**, and use the function **read_excel()** or **read_xls()**. You may want to specify what spreadsheet to read the data, and skip the first few lines of observations.
2. To get the y-axis in a format such as 10,000, try the option "**labels=scales::comma**" in **scale_y_continuous()**.