Assignment 6

Jack Scherer

2024-11-04

Exercises

Many of the exercises below build and solidify our data wrangling skills that rely on the use of operations taught in Chapter 4 and Chapter 7. This is a great time to review those chapters if you have not been using the syntax frequently. The exercises below have been written in a way that YOU must do all the data wrangling yourself - the time for providing lots of code to help you over the hump is over - your data science wrangling skills start now!

Exercise 1

A common task is to take a set of data that has multiple categorical variables and create a table of the number of cases for each combination. An introductory statistics textbook contains a data set summarizing student surveys from several sections of an intro class. The two variables of interest are `Gender` and `Year` which are the students gender and year in college. *Note: you will need to refer to Chapter 4 and Chapter 7 for some of the operations needed below - this is a great time to review chapter 4!*

a) Download the data set using the following:

```
Survey <- read.csv('https://www.lock5stat.com/datasets2e/StudentSurvey.csv', na.strings=c('',' '))
```

b) Select the specific columns of interest **Year** and **Gender**

```
Survey.2 <- Survey %>% select( c(Year, Gender) )
```

c) Convert the **Year** column to factors and properly order the factors based on common US progression (FirstYear - Sophomore - Junior - Senior)

```
Survey.2$Year <- factor( Survey.2$Year ) #Change Year column to factors

Survey.2 <- Survey.2 %>%
  mutate( Year = fct_relevel(Year, 'FirstYear', 'Sophomore', 'Junior', 'Senior') ) #Put in correct orde
```

d) Convert the **Gender** column to factors and rename them Male/Female.

```
Survey.2$Gender <- factor( Survey.2$Gender )

Survey.2 <- Survey.2 %>%
  mutate( Gender = fct_recode(Gender, 'Male' = 'M'),
          Gender = fct_recode(Gender, 'Female' = 'F') )
```

e) Produce a data set with eight rows and three columns that contains the number of responses for each gender:year combination. *You might want to look at the following functions:* ***`dplyr::count`*** *and* ***`dplyr::drop_na`****.*

```
Sum.Survey <- drop_na( Survey.2 ) %>%
  count( Gender, Year )

Sum.Survey
```

```
##   Gender      Year  n
## 1 Female FirstYear 43
## 2 Female Sophomore 96
## 3 Female    Junior 18
## 4 Female    Senior 10
## 5   Male FirstYear 51
## 6   Male Sophomore 99
## 7   Male    Junior 17
## 8   Male    Senior 26
```

f) Pivot the table in part (e) to produce a table of the number of responses in the following form:

Gender	First Year	Sophomore	Junior	Senior
Female				
Male				

```
Wide.Sum.Survey <- Sum.Survey %>%
  pivot_wider( names_from = Year, values_from = n )

Wide.Sum.Survey
```

```
## # A tibble: 2 x 5
##   Gender FirstYear Sophomore Junior Senior
##   <fct>      <int>     <int>  <int>  <int>
## 1 Female        43        96     18     10
## 2 Male          51        99     17     26
```

Exercise 2

From this book's GitHub there is a .csv file of the daily maximum temperature in Flagstaff at the Pulliam Airport. The link is: https://raw.githubusercontent.com/BuscagliaR/STA_444_v2/master/data-raw/FlagMaxTemp.csv

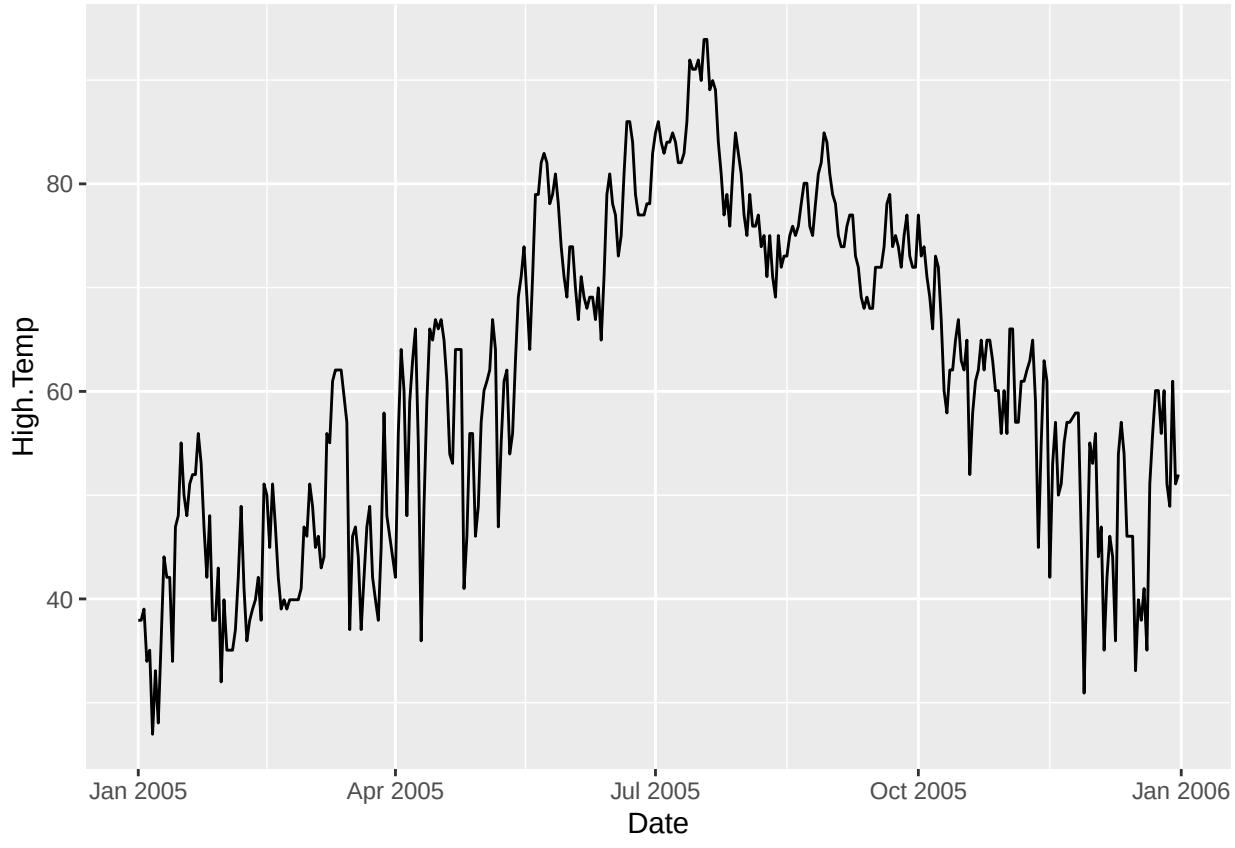
a) Create a line graph that gives the daily maximum temperature for 2005. *Make sure the x-axis is a date and covers the whole year.*

```
temps <- read.csv('https://raw.githubusercontent.com/BuscagliaR/STA_444_v2/master/data-raw/FlagMaxTemp.

long.temps <- temps %>% pivot_longer(
  X1:X31,
  names_to = 'Day',
  values_to = 'High.Temp' )
```

```
long.temps$Day <- long.temps$Day %>%  #Remove the 'X' at the start of the day string
  str_sub( start=2, end=3 )

long.temps$Day <- as.integer(long.temps$Day) #convert the Day column into an integer.

long.temps <- long.temps %>%
  mutate( Date = make_date(year=Year, month=Month, day=Day) )

temps.2005 <- long.temps %>% #make seperate data frame for 2005
  filter( Year == 2005 )

temps.2005$High.Temp <- as.numeric( temps.2005$High.Temp )
```

```
## Warning: NAs introduced by coercion
```

```
temps.2005 <- temps.2005 %>%
  filter( !is.na(High.Temp) ) %>%
  filter( !is.na(Date) )

ggplot( temps.2005, aes(x=Date, y=High.Temp) ) +
  geom_line()
```

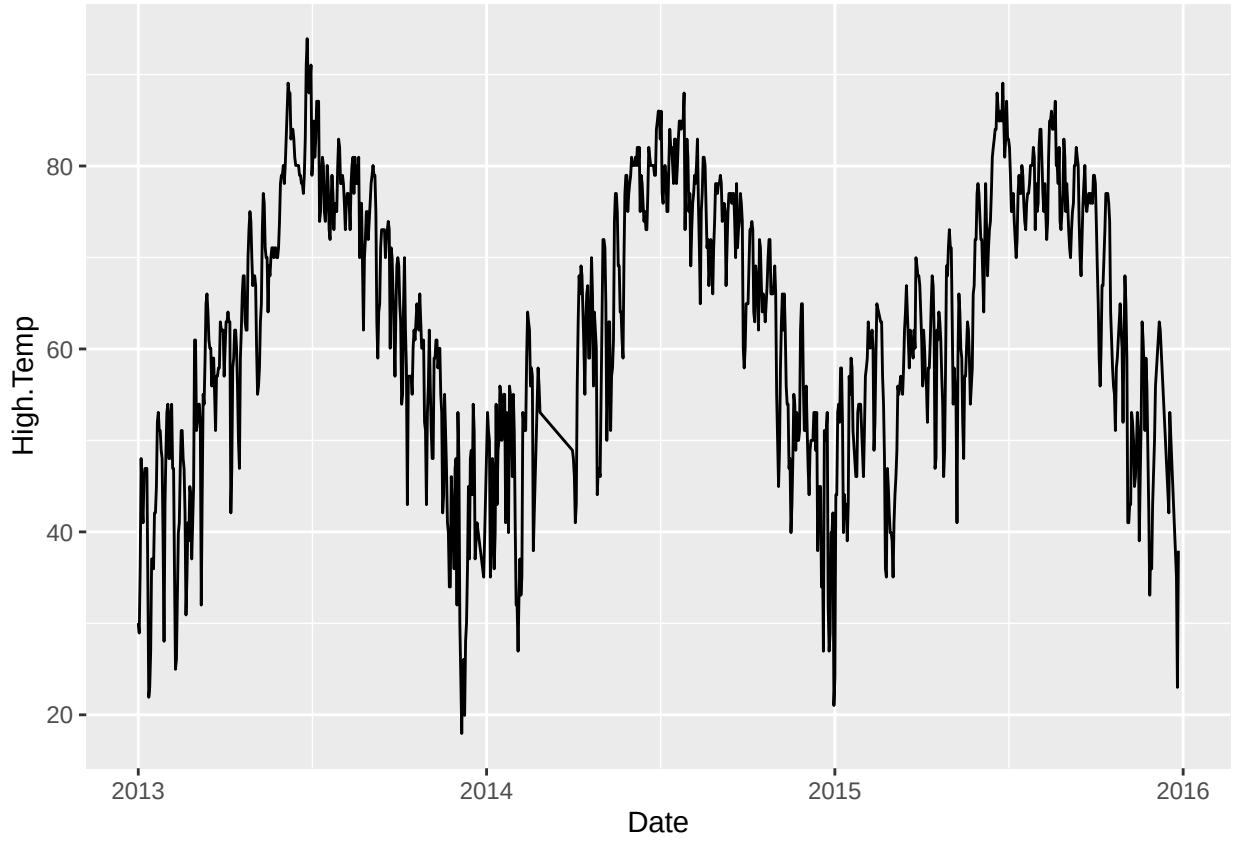


b) Create a line graph that gives the monthly average maximum temperature for 2013 - 2015. *Again the x-axis should be the date and span 3 years.*

```
temps.2013.2015 <- long.temps %>%
  filter( Year >= 2013 & Year <= 2015)

temps.2013.2015$High.Temp <- as.numeric( temps.2013.2015$High.Temp )
```

```
## Warning: NAs introduced by coercion
```

```
temps.2013.2015 <- temps.2013.2015 %>%
  filter( !is.na(High.Temp) ) %>%
  filter( !is.na(Date) )

ggplot( temps.2013.2015, aes(x=Date, y=High.Temp) ) +
  geom_line()
```



Exercise 3

For this problem we will consider two simple data sets.

```
A <- tribble(
  ~Name, ~Car,
  'Alice', 'Ford F150',
  'Bob',   'Tesla Model III',
  'Charlie', 'VW Bug')
```

```
B <- tribble(
  ~First.Name, ~Pet,
  'Bob',  'Cat',
  'Charlie', 'Dog',
  'Alice', 'Rabbit')
```

a) Combine the data frames together to generate a data set with three rows and three columns using `join` commands.

```
C <- full_join( A, B, by=c("Name" = "First.Name") )
C
```

```
## # A tibble: 3 x 3
##   Name    Car             Pet
##   <chr>   <chr>           <chr>
## 1 Alice   Ford F150       Rabbit
## 2 Bob     Tesla Model III Cat
## 3 Charlie VW Bug          Dog
```

b) It turns out that Alice also has a pet guinea pig. Add another row to the `B` data set. Do this using either the base function `rbind`, or either of the `dplyr` functions `add_row` or `bind_rows`.

```
B <- B %>%
  add_row( First.Name='Alice', Pet='Guinea pig' )
```

c) Combine again the `A` and `B` data sets together to generate a data set with four rows and three columns using `join` commands.

```
D <- full_join( A, B, by=c("Name" = "First.Name") )
D
```

```
## # A tibble: 4 x 3
##   Name    Car             Pet
##   <chr>   <chr>           <chr>
## 1 Alice   Ford F150       Rabbit
## 2 Alice   Ford F150       Guinea pig
## 3 Bob     Tesla Model III Cat
## 4 Charlie VW Bug          Dog
```

Note: You may want to also try using ***`cbind`*** *to address questions (a) and (c). Leave this as a challenge question and focus on the easier to use* ***`join`*** *functions introduced in this chapter.*

Exercise 4

The package `nycflights13` contains information about all the flights that arrived in or left from New York City in 2013. This package contains five data tables, but there are three data tables we will work with. The data table `flights` gives information about a particular flight, `airports` gives information about a particular airport, and `airlines` gives information about each airline. Create a table of all the flights on February 14th by Virgin America that has columns for the carrier, destination, departure time, and flight duration. Join this table with the airports information for the destination. Notice that because the

column for the destination airport code doesn't match up between `flights` and `airports`, you'll have to use the `by=c("TableA.Col"="TableB.Col")` argument where you insert the correct names for `TableA.Col` and `TableB.Col`.

```
library(nycflights13)
```

```
## Warning: package 'nycflights13' was built under R version 4.3.3
```

```
flights <- flights
airports <- airports
airlines <- airlines

flights.feb.14 <- flights %>%
  filter( month == 2 & day == 14 ) %>%
  select( 'carrier', 'dest', 'dep_time', 'air_time')

flights.com <-
  inner_join( flights.feb.14, airports, join_by("dest" == "faa") )
flights.com
```

```
## # A tibble: 933 x 11
##    carrier dest  dep_time air_time name         lat   lon   alt    tz dst   tzone
##    <chr>   <chr>    <int>    <dbl> <chr>      <dbl> <dbl> <dbl> <dbl> <chr> <chr>
##  1 UA      ORD         59      108 Chicago ~  42.0 -87.9   668    -6 A     Amer~
##  2 US      CLT        454       88 Charlott~  35.2 -80.9   748    -5 A     Amer~
##  3 UA      IAH        510      205 George B~  30.0 -95.3    97    -6 A     Amer~
##  4 UA      IAH        531      216 George B~  30.0 -95.3    97    -6 A     Amer~
##  5 AA      MIA        541      160 Miami In~  25.8 -80.3     8    -5 A     Amer~
##  6 UA      DFW        551      202 Dallas F~  32.9 -97.0   607    -6 A     Amer~
##  7 B6      BOS        552       35 General ~  42.4 -71.0    19    -5 A     Amer~
##  8 B6      FLL        553      164 Fort Lau~  26.1 -80.2     9    -5 A     Amer~
##  9 B6      BUF        553       61 Buffalo ~  42.9 -78.7   724    -5 A     Amer~
## 10 US      DCA        553       38 Ronald R~  38.9 -77.0    15    -5 A     Amer~
## # i 923 more rows
```