homework4.2

Andrew Estes

11/14/2021

Question 1.

Consider again the built-in dataset "nycflight13" – library(nycflight13)

In our earlier assignment, we only used the table named "flights," but there are actually several tables (or tibbles), each containing a SQL-style database; "flights", "airlines", "airports", "planes", and "weather". Explore each and discuss how they capture different kinds of information.

Use appropriate tidyverse packages to join the existing tables to create a new tibble, NewarkSept29, that includes the month, day, hour, and the full name of the carrier for all flights that originated at EWR (Newark) on 29 September. Describe (in your RMarkdown file) how your new tibble was made from existing ones, and why having multiple tables that follow SQL structure may make more sense for this kind of data than a single "rectangle."

When you have finished, please upload a pdf of your RMarkdown file.

```
#installing the necessary libraries
suppressPackageStartupMessages(library(nycflights13))
suppressPackageStartupMessages(library(tidyverse))
```

`{r, results='hide'} #viewing the five tables flights airlines airports planes weather` The "flights" tibble has 336,776 observations of 19 variables. The "airports" tibble has 1,458 observations of 8 variables. The "airlines" tibble has 16 observations of 2 variables. The "planes" tibble has 3,322 observations of 9 variables. The "weather" tibble has 26,115 observations of 15 variables.

Flights captures all of the flight information with respect to time, airlines, and airports. Airports gives us the geographic location of all the destination airports. Airlines gives us both the abbreviated and full names of each airline. Planes tells us the technical specs of each flight's plane. And weather is data captured hourly and contains expected variables such as temperature, humidity, pressure, etc.

```
#creating the first tibble
flights.September <- flights %>%
  filter(month == 9) %>%
  filter(day == 29) %>%
  filter(origin == "EWR")
```

While all of the information can likely be done as one giant block of code, for readability purposes I have broken it into several stages. For the first step, we created a new tibble for the flight data that matches the requested limitaitons of flights originated from EWR on September 29. This reduced it from 336 thousand observations down to 315 observations of the same 19 variables.

```
#join time
tib1 <- inner_join(airlines, flights.September)
```

The above function creates the tibble that includes the full name of the airline. The flights tibble only had the abbreviated name so we had to use the airlines dataset to get this requested information. This simply adds one variable to the 19 variable flights.September tibble.

```
#cleaning up
NewarkSept29 <- tib1 %>%
  select(month, day, hour, name)
NewarkSept29
```

This was fairly simple. We were requested to present the month, day, hour, and full airline name. Using the select command made this a simple task, resulting in 315 observations of 4 variables.

Finally, we were asked to address the potential need for separate tibbles. I think there are three main reasons for the separation with the first being processing speed. The largest tibble had 336 thousand observations and this only covered one year. Most analysis would cover decades of data and processing millions of unnecessary datapoints just slows things down. The second step is data uniformity. As shown in the beginning paragraph, each of these tibbles has wildly different observation and variable numbers. The final reason is a not necessarily a technical reason, but a business/practical issue. The collectors/inputs of the data systems work for different entities. Weather is handled by a different division than flights. Airports is likely an open-sourced dataset that was adapted for the NYCFlights13 library. Different people, departments, divisions will have different reasons of looking at the data and will have different methods of inputting the data. Things like weather can be automated from digital guages (provided they don't break) but they tap into a different database than the flights dataset which likely relates to the Air Traffic Controller's system. The Airlines dataset could be created by hand, and could be open-sourced just like the Airports dataset.