SARFRAZ_HW4

Hussain Sarfraz

9/27/2021

Question 1

Introduction/Pre-work

To start off I wanted to understand the two datasets used in the anti_join() function. This was 'flights' and 'airports'

I typed "?flights" and "?airports" in R to get a description of each data set and the columns used in each one. This is what I got:

flights: This data set contains data for all the flights that departed NYC. It also includes the data for the arrival airport.

airports: This data set gives the latitude and longitude for each airport. This data set does not have a focus on flights that left from a particular region (the flights data set has a focus on a particular region which is New York since it only holds records for flights that departed NYC)

Part 1: anti_join(flights, airports, by = c("dest" = "faa") So given this context the code **anti_join(flights, airports, by = c("dest" = "faa"))** would use the **dest** column in the flights dataset and compare it to the **faa** column values in the airports dataset. After the comparison of the column values R is going to see which values in the **dest** column do not have a match with the values in the **faa** column.

Since the **anti_join()** function was used R is going to display the **dest** column values that do not have a match or appear in the **faa** column.

So to conclude, the code **anti_join(flights, airports, by = c("dest" = "faa"))** displays the destinations (in the flights dataset) that are not present in the airports dataset.

1. One explanation about why this might be the case is that the destinations in the flights dataset might be international destinations. Maybe the airports dataset only includes locations of airports in the U.S only.
2. Another reason is that there might have been a error in the data entry for destinations which is why there was no match between the **dest** and **faa** datasets.

Part 2: anti_join(airports, flights, by = c("faa" = "dest")) The code **anti_join(airports, flights, by = c("faa" = "dest"))** works in a similar manner. It would use the **faa** column in the airports dataset and compare it to the **dest** column values in the flights dataset. After the comparison of the column values R is going to see which values in the **faa** column do not have a match with the values in the **dest** column.

Since the **anti_join()** function was used R is going to display the **faa** column values that do not have a match or appear in the **dest** column.

So to conclude, the code **anti_join(airports, flights, by = c("faa" = "dest"))** displays the destinations (in the airports dataset) that are not present in the flights dataset.

1. As mentioned earlier, the flights dataset only includes the information of flights that left NYC. This means that in the flights dataset the departure airport would be one based in New York. Given this, it can be concluded that the destinations displayed in the **anti_join(airports, flights, by = c("faa" = "dest"))** statement are showing airport destinations that airplanes (from NYC airports) have not traveled too.
2. Another reason is that there might have been a error in the data entry for destinations which is why there was no match between the **faa** and **dest** datasets.

Question 2

Introduction/Pre-work

I created a object **non_cancelled** which filters out the cancelled flights in the **flights** dataset. I did this so I do not see empty data values in my final datasets

```
not_cancelled <- flights %>%
  filter(!is.na(dep_delay), !is.na(arr_delay))
```

Part 1: Finding day in 2013 that has longest average delay I start off with answering this problem I had to first find the day in 2013 that had the longest average delay. I found this out by first filtering my data with the **filter()** function to only include flights that occurred in 2013.

I then used the **group_by()** function to calculate the average delays per day.

The **mutate()** function was then used to create a new column in the **not_cancelled** dataset that would display the total delay for every flight.

The **summarize()** function was used to calculate the average delay per day. Then, **arrange()** was used to order the average delays from greatest to least.

head() was not required, but I just used it in my code to only see the day that had the greatest delay (this was the first row in the new dataset that was made and the output is shown below)

```
not_cancelled %>%
  filter(year == 2013) %>%
  group_by(.,year, month, day) %>%
  mutate(total_delay = dep_delay + arr_delay) %>%
  summarize(ave_delay = mean(total_delay, na.rm=T),) %>%
  arrange(desc(ave_delay)) %>%
  head(1)
```

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

```
## # A tibble: 1 x 4
## # Groups:   year, month [1]
##    year month   day ave_delay
##   <int> <int> <int>     <dbl>
## 1  2013     3     8      170.
```

Part 2: Looking at weather data to see what happened Now I had to investigate further and understand why there were the highest delays on March 8th.

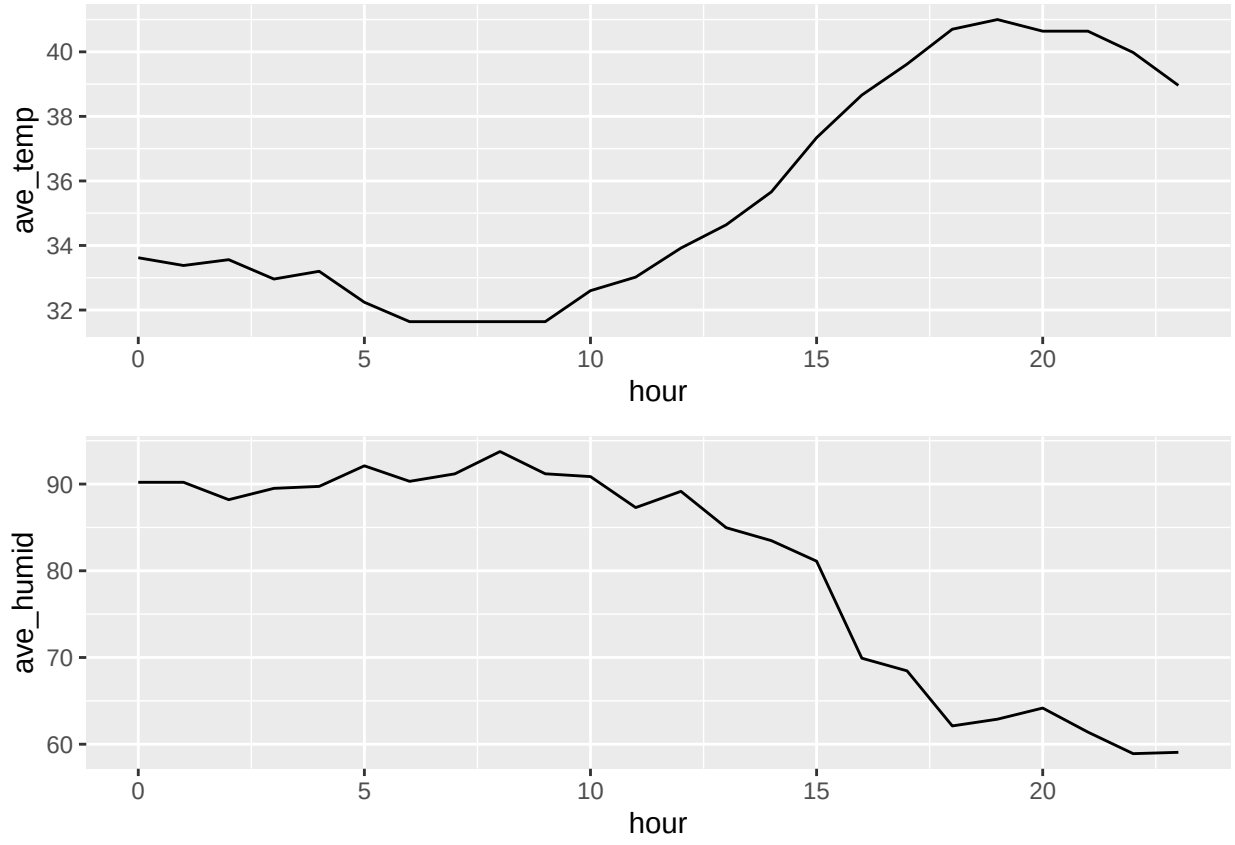
To do this I am creating two objects (**hour_temp** and **hour_humid**). **hour_temp** would display a line graph of the average temperature in each hour while the object **hour_humid** would display a line graph of the average humidity in each hour of the day.

To display the two line graphs together I used the function **grid_arrange**. I did this for easier analysis of the 2 graphs.

```
hour_temp <- weather %>%
  filter(year == 2013, month == 3, day == 8) %>%
  group_by(hour) %>%
  summarize(ave_temp = mean(temp), ave_humid = mean(humid)) %>%
  ggplot() +
  geom_line( mapping = aes(x=hour,y=ave_temp))
```

```
hour_humid <- weather %>%
  filter(year == 2013, month == 3, day == 8) %>%
  group_by(hour) %>%
  summarize(ave_temp = mean(temp), ave_humid = mean(humid)) %>%
  ggplot() +
  geom_line( mapping = aes(x=hour,y=ave_humid))
```

```
grid.arrange(hour_temp, hour_humid)
```



As you can see in hours 0-13(approximately) of both graphs the temperature was below 36 degrees and the humidity was above 85 degrees.

The low temperature suggests that the weather in March 8th was cold and it was probably snowing which is why there were a lot of delays on March 8th.

The high humidity is also another explanation as to why there were many delays. In fact, the humidity decreased to 75 degrees at around 16 hours which means that the humidity was high for most of the day.

Question 3

I now want to find which airplane models have the highest average speed. I am then going to see if my results were suprising based on what I know/have learned about the planes.

I used started off by joining the **not_cancelled** dataset to the **planes** dataset. I did this because I did not want all the values in the **planes** dataset and just wanted the values that had a match (through the **tailnum** column). I am joining the datasets since I want the model number and use that to compare the average speed.

I had to first use the **mutate** function to calculate the **avg_trip_speed** since there was no exsisting column in the **flights** dataset that gave this value. The average trip speed is by hour since I convert the **air_time** variable from a minutes to hour format by dividing by 60.

After I get **avg_trip_speed** for each plane I then need to group by model and manufacturer so I can calculate the average speed for each airplane model. The variable **ave_model_speed** makes this calculation. I then use the **arrange** function to sort the **ave_model_speed** from greatest to lowest order. This way I know which airplane models have the highest average speed.

```
not_cancelled %>%
  left_join(planes, by = "tailnum") %>%
  mutate(avg_trip_speed = distance/(air_time/60)) %>%
  group_by(model, manufacturer) %>%
  summarize(ave_model_speed = mean(avg_trip_speed,na.rm=T)) %>%
  arrange(desc(ave_model_speed)) %>%
  head(10)
```

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

```
## # A tibble: 10 x 3
## # Groups:   model [10]
##    model     manufacturer     ave_model_speed
##    <chr>     <chr>                      <dbl>
##  1 777-222   BOEING                      483.
##  2 A330-243  AIRBUS                      480.
##  3 767-424ER BOEING                      467.
##  4 A321-231  AIRBUS INDUSTRIE            460.
##  5 A330-223  AIRBUS INDUSTRIE            458.
##  6 757-212   BOEING                      456.
##  7 A319-115  AIRBUS                      455.
##  8 767-223   BOEING                      452.
##  9 A330-323  AIRBUS                      450.
## 10 A319-112  AIRBUS                      450.
```

Now from what I have seen in the dataset. The 3 planes that have the highest average speed are Boeing, Air Bus, and Air Bus Industine. I am not surprised with the results since a search online has informed me that

Boeing and Air Buses are well known planes used by many airline companies. All three plane models can travel super fast and are used for long haul flights. The aircraft models have been in the market for many years and are bought by major airline companies such as American Airlines, Delta, United Airlines, etc.

Question 4

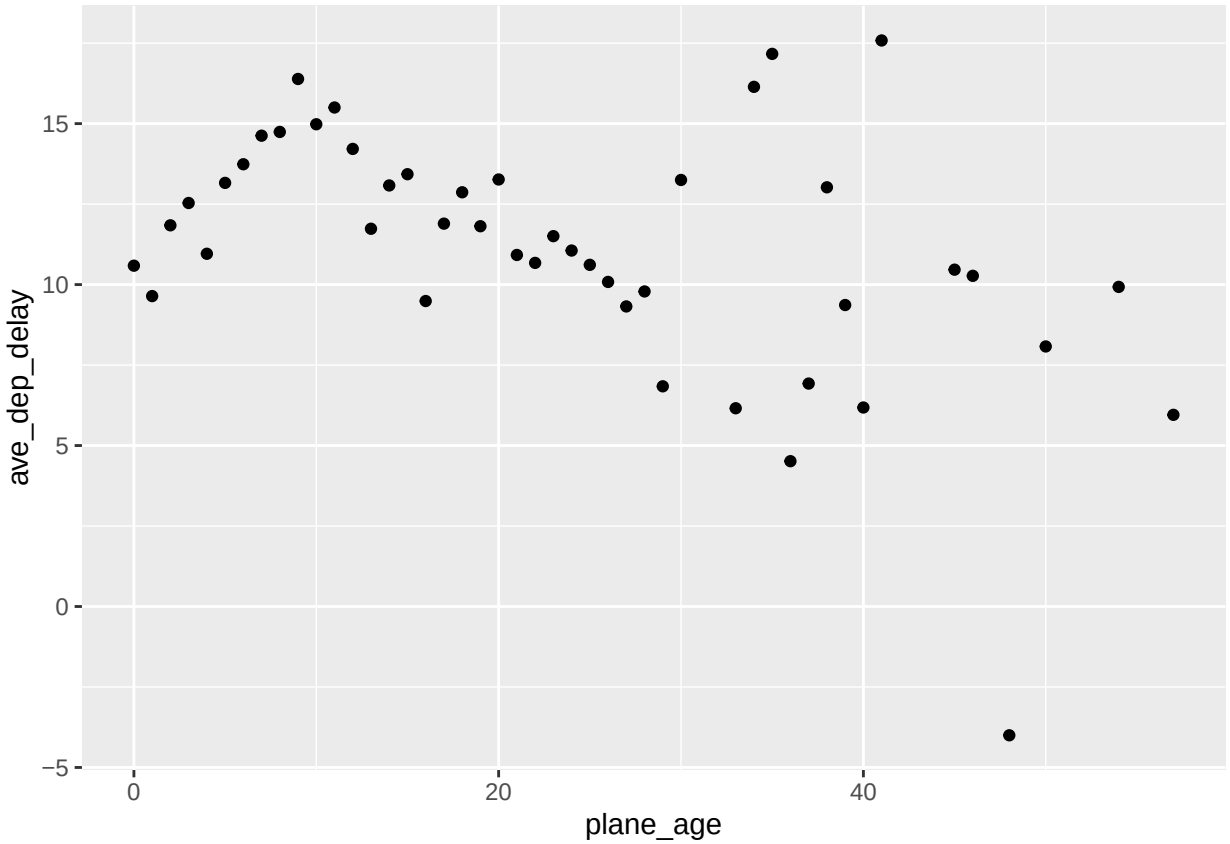
Now I am going to find out if there is a relationship between a planes age and its delays. To start I have to join the flights dataset with the planes dataset.

I got the planes age by subtracting the x year (year in flight dataset) with the y year (year in planes dataset). I then used **group_by** to get the average departure and arrival delays for each plane.

Here is my scatter plot graph of a planes age and the average departure delay as the plane age increases.

```
flights %>%
  left_join(planes, by= "tailnum") %>%
  mutate(plane_age = year.x - year.y) %>%
  group_by(plane_age) %>%
  summarize(ave_dep_delay = mean(dep_delay, na.rm=T),
            ave_arr_delay = mean(arr_delay, na.rm=T)) %>%
   ggplot() +
   geom_point(mapping = aes(x=plane_age, y=ave_dep_delay))
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



The scatterplot shows that there is not a relationship between **plane age** and **departure delays** because throughout the planes lifecycle most of the points with a plane age of 0-30 is in the 10-20 range and it does not decrease. There are a few points after the plane age of 30 that have a departure delay below 10.

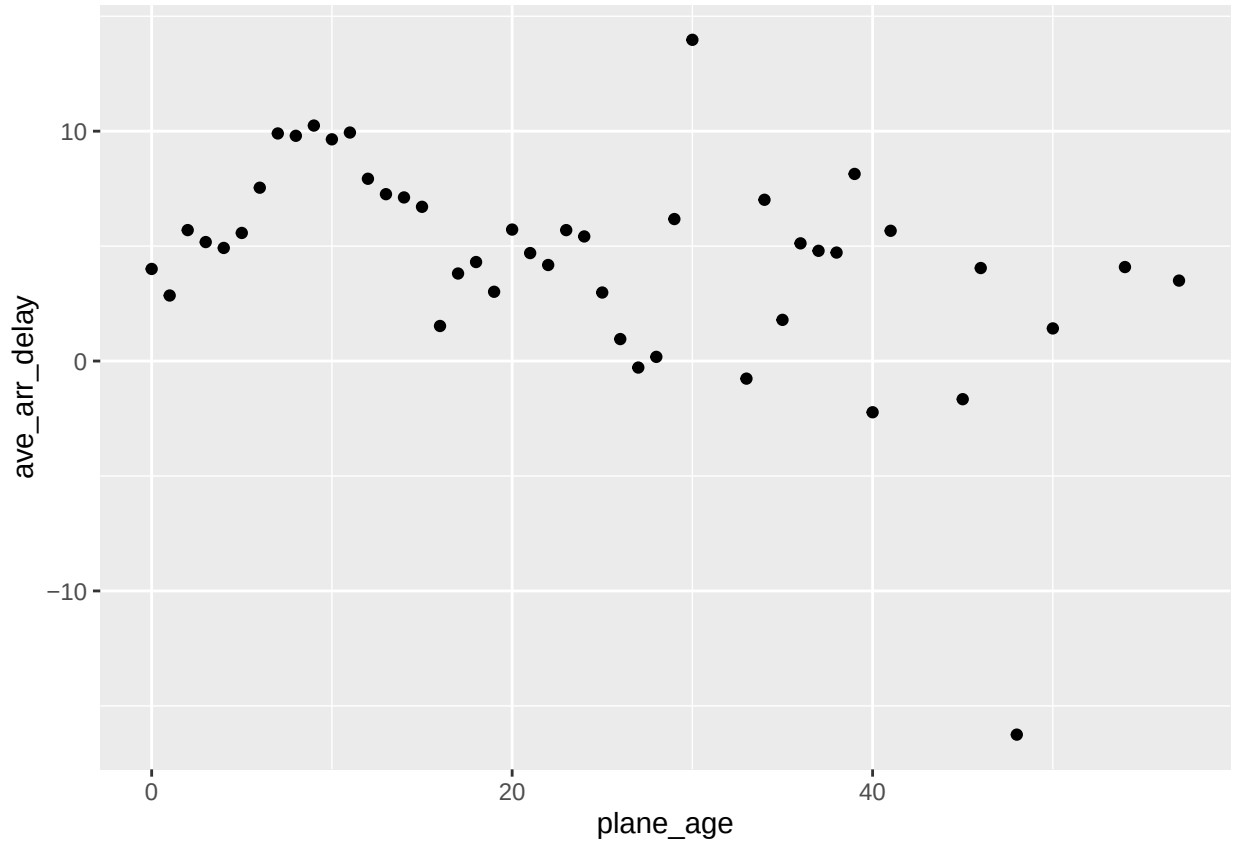
This means that as the plane gets older (beyond the age of 30) the departure delay does slightly decrease but there are no average departure delays that are below zero.

This graph could possibly suggest that as a plane gets older it experiences less departure delays. But ultimately, no matter what the planes age it would most likely experience a departure delay. There is one outlier point that does have a negative departure delay (this outlier point has a plane age close to 50). But this outlier is only one point and it does not acculturatly represent the data.

Here is my scatter plot graph of a planes age and the average arrival delay as the plane ages.

```
flights %>%
   left_join(planes, by= "tailnum") %>%
   mutate(plane_age = year.x - year.y) %>%
   group_by(plane_age) %>%
   summarize(ave_dep_delay = mean(dep_delay, na.rm=T),
             ave_arr_delay = mean(arr_delay, na.rm=T)) %>%
   ggplot() +
   geom_point(mapping = aes(x=plane_age, y=ave_arr_delay))
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



As shown in the scatterplot the relationship between **plane age** and **arrival delays** does not exist, but it is stronger than the relationship between **plane age** and **departure delays**.

This is because most of the points in the scattersplot have a delay between 0-10. Majority of the points in the **plane age** and **departure delays** scatterplot had a delay which was greater than 10 which suggests that the average arrival delay time would most likely be lower (by a few minutes) than the average departure delay time.

Conclusion

So I have concluded that there is no relationship between airplane age and delays. This was surprising since I assumed that airplanes with a low age would have less average delays since the airplane was new/young. However, I did notice that the planes in the **plane age** and **departure delays** scatter plot had higher average delay times then the planes listed in the **plane age** and **arrival delays** scatterplot (but do note that this difference was not big and was only a few minutes – 10 minutes approximately)

A possible explanation for this difference between departure and arrival delays could be that before a plane departures it needs to go through aircraft cleaning and loading the luggage off the plane. When a plane arrives the moment the plane lands on a airport is counted as arriving and the other factors are not taken into consideration (such as aircraft cleaning and loading the luggage off the plane).

NOTE: But as I said before this observation might not hold true because the difference between average departure and arrival delays was only by a few minutes (10 minutes approximately) and nothing too big.