MSCS 264: Homework #6

Due Friday March 3

YOUR NAME HERE

Save this file as HW6_YOUR_NAME.Rmd and change the author above.

For this assignment, you will continue your work with the flights data.

First filter so you are only examining flights that were not cancelled using the code chunk below (after removing `eval = FALSE`).

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

Part 1: Debugging practice

Finding and fixing problems in code is known as "debugging". Before attempting these problems, try out Chunk 1 and Chunk 2 of ch5_6_debugging_practice.Rmd in the Class Code folder. I also put in the key for these two!

As you think about debugging, here are some general tips. * First, read the entire error message. Even if you cannot tell what exactly the problem is, the message might give you a hint about where the problem is (which line of code).

- Closely examine the code and envision (or sketch!) what you THINK the code should be doing.
- Run just one line at a time, and compare to your sketch of what you think should be happening.

In this homework, we will tackle Chunk 3 and 4 of the debugging worksheet. (They are copied here.)

1. Make a copy of this chunk. Fix it so it works. Give a short explanation about the problem and/or why your solution works.

```
# Chunk 3 - will get error in ggplot
not_cancelled %>%
  group_by(carrier) %>%
  summarise(num_flights = n(),
            total_dist = sum(distance)) %>%
  mutate(avg_dist = total_dist / num_flights)
```

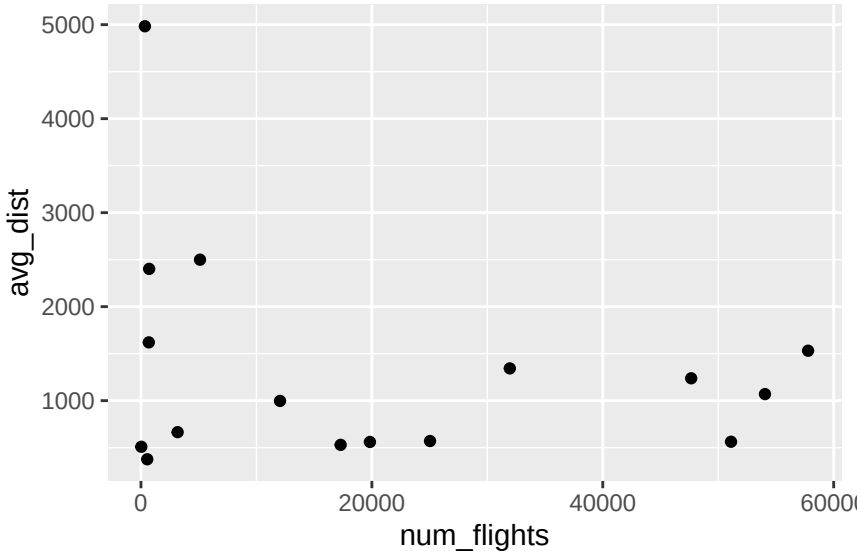
```
## # A tibble: 16 x 4
##    carrier num_flights total_dist avg_dist
##    <chr>         <int>      <dbl>    <dbl>
##  1 9E            17294    9163911     530.
##  2 AA            31947   42913762    1343.
##  3 AS              709    1703018    2402
##  4 B6            54049   57815654    1070.
##  5 DL            47658   58999610    1238.
##  6 EV            51108   28766906     563.
##  7 F9              681    1103220    1620
```

```
##  8 FL             3175    2110700     665.
##  9 HA              342    1704186    4983
## 10 MQ            25037   14280468     570.
## 11 OO               29      14769     509.
## 12 UA            57782   88482811    1531.
## 13 US            19831   11121739     561.
## 14 VX             5116   12787097    2499.
## 15 WN            12044   12007523     997.
## 16 YV              544     204782     376.
```

```
ggplot(data = not_cancelled, aes(x = num_flights, y = avg_dist)) +
  geom_point()
```

```
## Error in FUN(X[[i]], ...): object 'num_flights' not found
```

```
# Chunk 3 - will get error in ggplot
not_cancelled %>%
  group_by(carrier) %>%
  summarise(num_flights = n(),
            total_dist = sum(distance)) %>%
  mutate(avg_dist = total_dist / num_flights)%>%
  ggplot(aes(x = num_flights, y = avg_dist)) +
  geom_point()
```

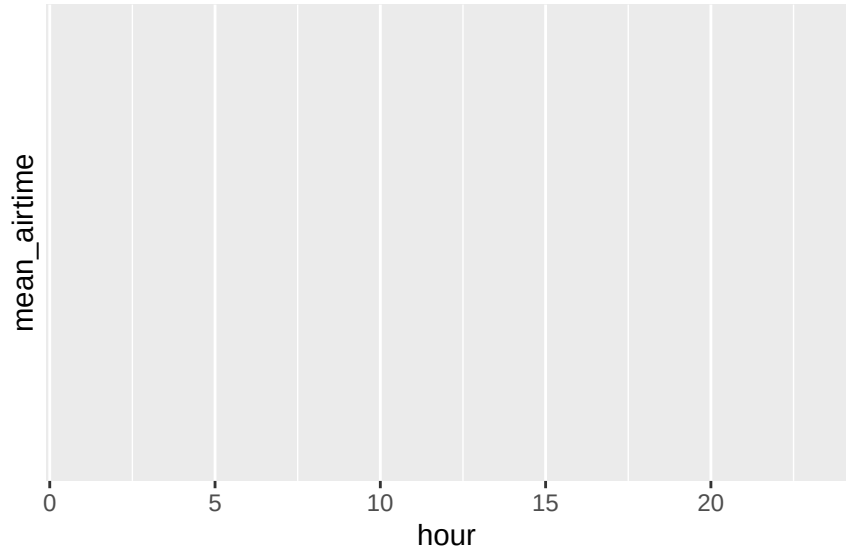


Ans: The mutated dataframe was not passed onto the ggplot() function so it could not find the variable num_flights in the no_canclled dataframe>

2. Make a copy of this chunk. Fix it so it works. Give a short explanation about the problem and/or why your solution works.

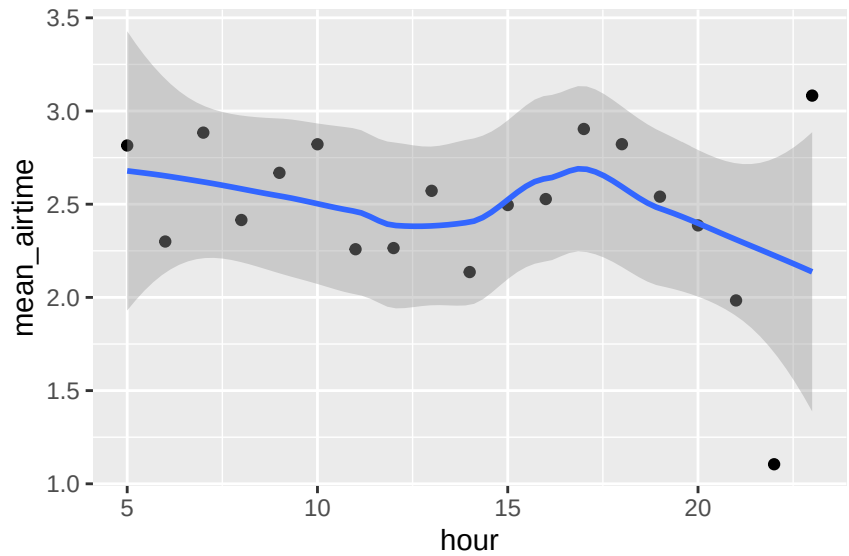
```
# Chunk 4 - will get blank ggplot
flights %>%
  select(air_time, hour) %>%
  mutate(air_time_hours = air_time / 60) %>%
  group_by(hour) %>%
  summarize(mean_airtime = mean(air_time_hours)) %>%
  ggplot(aes(x = hour, y = mean_airtime)) +
    geom_point() +
    geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 20 rows containing non-finite values (stat_smooth).

## Warning: Removed 20 rows containing missing values (geom_point).
```




```
# Chunk 4 - will get blank ggplot
flights %>%
  select(air_time, hour) %>%
  filter(!is.na(air_time))%>%
  mutate(air_time_hours = air_time / 60) %>%
  group_by(hour) %>%
  summarize(mean_airtime = mean(air_time_hours)) %>%
  ggplot(aes(x = hour, y = mean_airtime)) +
    geom_point() +
    geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Ans: The dataset had values in airtime that were NA and that was making the mean NA because any arithmatic done with NA is NA.

Part 2: group_by and summarize

3. Find the furthest distance for each origin airport.

```
not_cancelled|>
  select(origin, distance)|>
  group_by(origin)|>
  summarise(max_distance = mean(distance))
```

```
## # A tibble: 3 x 2
##   origin max_distance
##   <chr>         <dbl>
## 1 EWR           1065.
## 2 JFK           1275.
## 3 LGA            785.
```

4. Find the average arrival delay at MSP compared to ORD (Chicago-O'Hare). (hint: use filter, group_by, and summarize!)

```
not_cancelled|>
  select(dest, arr_delay)|>
  filter(dest %in% c("MSP", "ORD"))|>
  group_by(dest)|>
  summarise(avg_arr_delay = mean(arr_delay))
```

```
## # A tibble: 2 x 2
##   dest  avg_arr_delay
##   <chr>         <dbl>
## 1 MSP            7.27
## 2 ORD            5.88
```

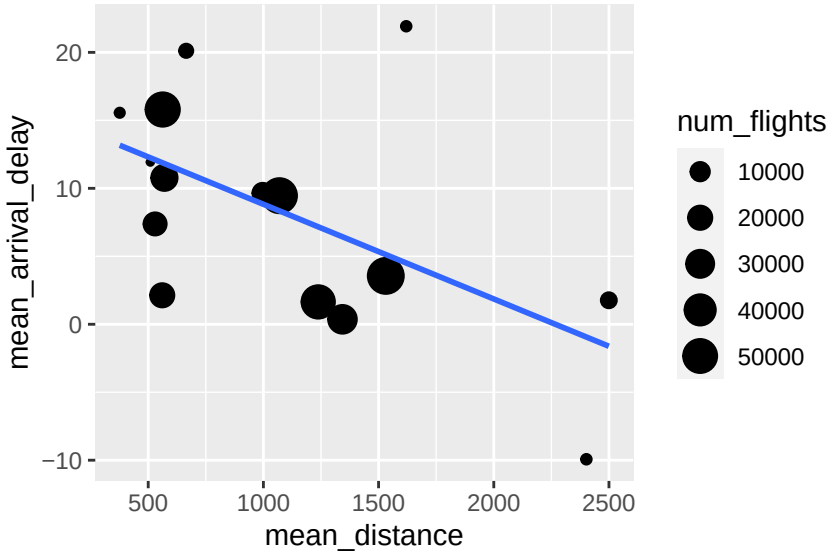
5. We are going to create a plot that looks like the one in the Homework/images folder called "carrier_distance_number.png". To do so, do each of the following steps. Check each step as you go, then connect it to the next with a pipe.

- use group_by and summarize to create the necessary variables *for each carrier*: mean distance, mean arrival delay, and number of flights.
- make a plot using geom_point and the appropriate aesthetics. Doe you see any outliers?
- insert a "filter" before your ggplot to remove carriers with unusually high mean distances.

```
tbl_5 <- not_cancelled|>
  select(carrier, distance, arr_delay)|>
  group_by(carrier)|>
  summarise(mean_distance = mean(distance), mean_arrival_delay = mean(arr_delay), num_flights = n())|>
  filter(mean_distance < 3000)|>
  ggplot(aes(mean_distance, mean_arrival_delay))+
  geom_point(aes(size = num_flights))+
  geom_smooth(se = FALSE, method = "lm")
```

```
tbl_5
```

```
## `geom_smooth()` using formula 'y ~ x'
```



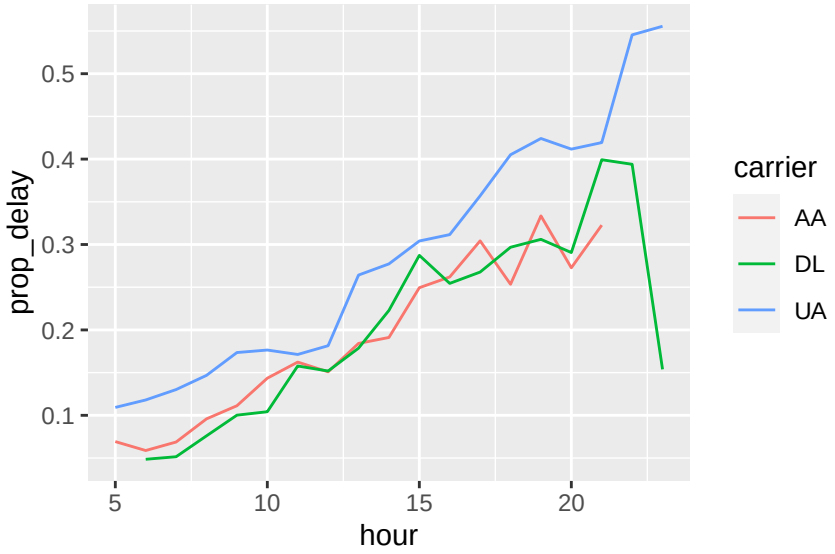
6. Comment on what you learn from the above plot.

Ans: A linear model be a better fit when we get rid of the outliers. And we need to get rid of the outliers before we plot the dtat.

7. We are going to make the plot in homework/images/prop_delay_by_hour_carrier.png. The x axis is the scheduled departure hour (`hour`). The y axis is the proportion of flights in that hour that have a departure delay greater than 10 minutes. Each line represents one of three carriers: AA, DL, UA. Hint: you'll need filter, group_by and summarize! Try sketching the data on paper before you write your code, and be sure to check your data before you pipe it into your ggplot()!

```
not_cancelled|>
  select(carrier, dep_delay, hour)|>
  filter(carrier %in% c("AA", "DL", "UA"))|>
  group_by(carrier, hour)|>
  summarise(prop_delay = (sum(dep_delay > 10))/length(dep_delay))|>
  ggplot(aes(hour, prop_delay))+
  geom_line(aes(color = carrier))
```

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



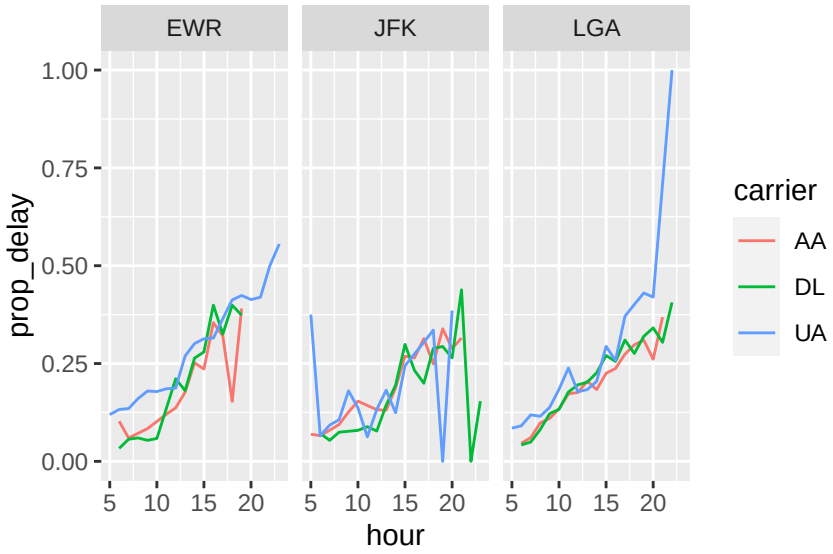
8. Comment on what trends you see in the graph.

Ans: As the hour increases the proportion of flights that has a departure delay greater than 10 minutes increases and hits the peak around midnight for all three arilines.

9. Copy and modify your code from 7 so that there are three separate plots, one for each origin airport (hint: facet_wrap)

```
not_cancelled|>
  select(carrier, dep_delay, hour, origin)|>
  filter(carrier %in% c("AA", "DL", "UA"))|>
  group_by(carrier, hour, origin)|>
  summarise(prop_delay = (sum(dep_delay > 10))/length(dep_delay))|>
  ggplot(aes(hour, prop_delay))+
  geom_line(aes(color = carrier))+
  facet_wrap(~origin)
```

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


EWR
JFK
LGA
1.00
0.75
0.50
0.25
0.00
prop_delay
5
10
15
20
hour
carrier
AA
DL
UA