Data 412 HW 4

Ashley Totten

February 20, 2025

Star Wars

```r
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
data(starwars)
```

1. The only difference in the output is that the group_by function put the droid species first in the tibble whereas the .by function put the human species first.

```r
starwars %>%
  filter(homeworld == "Tatooine") %>%
  group_by(species) %>%
  summarize(height_mean = mean(height), height_sd = sd(height), mass_mean = mean(mass), mass_
# A tibble: 2 x 6
  species height_mean height_sd mass_mean mass_sd   num
  <chr>         <dbl>     <dbl>     <dbl>   <dbl> <int>
1 Droid          132       49.5      53.5    30.4     2
2 Human          179.      12.8      NA      NA       8
```

```
starwars %>%
  filter(homeworld == "Tatooine") %>%
  summarize(.by = species, height_mean = mean(height), height_sd = sd(height), mass_mean = me
```

```
# A tibble: 2 x 6
  species height_mean height_sd mass_mean mass_sd   num
  <chr>         <dbl>     <dbl>     <dbl>   <dbl> <int>
1 Human          179.      12.8      NA      NA       8
2 Droid          132       49.5      53.5    30.4     2
```

2. The tallest droid is C-3PO. The tallest human is Darth Vader.

```
starwars %>%
  filter(homeworld == "Tatooine") %>%
  select(name, species, height) %>%
  arrange(species, desc(height))
```

```
# A tibble: 10 x 3
   name               species height
   <chr>              <chr>    <int>
 1 C-3PO              Droid      167
 2 R5-D4              Droid       97
 3 Darth Vader        Human      202
 4 Anakin Skywalker   Human      188
 5 Biggs Darklighter  Human      183
 6 Cliegg Lars        Human      183
 7 Owen Lars          Human      178
 8 Luke Skywalker     Human      172
 9 Beru Whitesun Lars Human      165
10 Shmi Skywalker     Human      163
```

3. The smallest droid is R5-D4, he appeared in a New Hope. The smallest human is Beru Whitesun Lars who appeared in A New Hope, Attack of the Clones, and Revenge of the Sith.

```
starwars %>%
  filter(homeworld == "Tatooine") %>%
  select(name, species, mass, films) %>%
  arrange(species, mass)
```

```
# A tibble: 10 x 4
   name               species  mass films
   <chr>              <chr>   <dbl> <list>
 1 R5-D4              Droid      32 <chr [1]>
 2 C-3PO              Droid      75 <chr [6]>
 3 Beru Whitesun Lars Human      75 <chr [3]>
 4 Luke Skywalker     Human      77 <chr [5]>
 5 Biggs Darklighter  Human      84 <chr [1]>
 6 Anakin Skywalker   Human      84 <chr [3]>
 7 Owen Lars          Human     120 <chr [3]>
 8 Darth Vader        Human     136 <chr [4]>
 9 Shmi Skywalker     Human      NA <chr [2]>
10 Cliegg Lars        Human      NA <chr [1]>
```

NYC Flights

```
library(nycflights13)
data(flights)
```

1. There are 10200 flights that had an arrival delay of 2 hours or more. There are 139504 flights that were operated by United, American, or Delta.

```
flights %>%
  filter(arr_delay >= 120) -> delay_2
head(delay_2, 10)
```

```
# A tibble: 10 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
 1  2013     1     1      811            630       101     1047            830
 2  2013     1     1      848           1835       853     1001           1950
 3  2013     1     1      957            733       144     1056            853
 4  2013     1     1     1114            900       134     1447           1222
 5  2013     1     1     1505           1310       115     1638           1431
 6  2013     1     1     1525           1340       105     1831           1626
 7  2013     1     1     1549           1445        64     1912           1656
 8  2013     1     1     1558           1359       119     1718           1515
 9  2013     1     1     1732           1630        62     2028           1825
10  2013     1     1     1803           1620       103     2008           1750
# i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
count(delay_2)
```

```
# A tibble: 1 x 1
      n
  <int>
1 10200
```

```
flights %>%
  filter( carrier == "DL"| carrier == "AA"|carrier == "UA") -> carriers
head(carriers, 10)
```

```
# A tibble: 10 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
 1  2013     1     1      517            515         2      830            819
 2  2013     1     1      533            529         4      850            830
 3  2013     1     1      542            540         2      923            850
 4  2013     1     1      554            600        -6      812            837
 5  2013     1     1      554            558        -4      740            728
 6  2013     1     1      558            600        -2      753            745
 7  2013     1     1      558            600        -2      924            917
 8  2013     1     1      558            600        -2      923            937
 9  2013     1     1      559            600        -1      941            910
10  2013     1     1      559            600        -1      854            902
# i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
count(carriers)
```

```
# A tibble: 1 x 1
```

```
       n
   <int>
1 139504
```

2. I could not get the 'and' syntax to work so I was not able to complete this in one pipe.

```
flights %>%
  filter(arr_delay >= 120, dep_delay <= 0, between(month, 7, 8)) -> temp_flights
flights %>%
  filter(arr_delay >= 120, dep_delay <= 0,month == 6, between(day, 21, 30)) -> temp_flights2
flights %>%
  filter(arr_delay >= 120, dep_delay <= 0, month == 9, between(day, 1, 22)) -> temp_flights3

aflights <- rbind(temp_flights, temp_flights2)
flights2 <- rbind(aflights, temp_flights3)
flights2 %>%
  arrange(month)
```

```
# A tibble: 16 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
 1  2013     6    24     1602           1605        -3     2134           1916
 2  2013     6    27     2052           2100        -8       13           2210
 3  2013     6    30     1423           1425        -2     1816           1554
 4  2013     7     1      905            905         0     1443           1223
 5  2013     7     7     1659           1700        -1     2050           1823
 6  2013     7     7     1727           1730        -3     2203           1951
 7  2013     7     7     1746           1755        -9     2133           1921
 8  2013     7     7     1823           1830        -7     2201           1955
 9  2013     7    22     1555           1600        -5     2139           1938
10  2013     7    22     1606           1615        -9     2056           1831
11  2013     7    22     1628           1630        -2     2151           1939
12  2013     7    28     1710           1711        -1     2248           2039
13  2013     8     8     1457           1500        -3     1828           1624
14  2013     8    13      657            659        -2     1015            814
15  2013     8    28     1157           1200        -3     1520           1316
16  2013     9    19      656            700        -4     1037            833
# i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>
```

```r
head(flights2, 10)
```

```
# A tibble: 10 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
 1  2013     7     1      905            905         0     1443           1223
 2  2013     7     7     1659           1700        -1     2050           1823
 3  2013     7     7     1727           1730        -3     2203           1951
 4  2013     7     7     1746           1755        -9     2133           1921
 5  2013     7     7     1823           1830        -7     2201           1955
 6  2013     7    22     1555           1600        -5     2139           1938
 7  2013     7    22     1606           1615        -9     2056           1831
 8  2013     7    22     1628           1630        -2     2151           1939
 9  2013     7    28     1710           1711        -1     2248           2039
10  2013     8     8     1457           1500        -3     1828           1624
# i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>
```

3. The destinations that are the shortest distance are La Guardia (LGA) and Philidelphia International (PHL).

```r
flights %>%
  select(distance, dest) %>%
  arrange(distance) %>%
  head(2)
```

```
# A tibble: 2 x 2
  distance dest
     <dbl> <chr>
1       17 LGA
2       80 PHL
```

```r
data(airports)
```

4. There are some flights where their scheduled time and delay times are not equal to their departure times. These could be due to inaccuracies when inputting data or a complication with combining a negative delay time with the scheduled departure time.

```
flights %>%
  mutate(dep_time_true = dep_time == dep_delay + sched_dep_time) %>%
  filter(dep_time_true == FALSE)
```

```
# A tibble: 99,777 x 20
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
 1  2013     1     1      554            600        -6      812            837
 2  2013     1     1      555            600        -5      913            854
 3  2013     1     1      557            600        -3      709            723
 4  2013     1     1      557            600        -3      838            846
 5  2013     1     1      558            600        -2      753            745
 6  2013     1     1      558            600        -2      849            851
 7  2013     1     1      558            600        -2      853            856
 8  2013     1     1      558            600        -2      924            917
 9  2013     1     1      558            600        -2      923            937
10  2013     1     1      559            600        -1      941            910
# i 99,767 more rows
# i 12 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>, dep_time_true <lgl>
```

5. Frontier Airlines has the worst average departure time with a mean of over 20 minutes.

```
flights[!is.na(flights$dep_delay),] %>%
  summarize(.by = carrier, avg_delay = mean(dep_delay)) %>%
  arrange(desc(avg_delay))
```

```
# A tibble: 16 x 2
   carrier avg_delay
   <chr>       <dbl>
 1 F9           20.2
 2 EV           20.0
 3 YV           19.0
 4 FL           18.7
 5 WN           17.7
 6 9E           16.7
 7 B6           13.0
 8 VX           12.9
 9 OO           12.6
10 UA           12.1
```

```
11 MQ           10.6 
12 DL            9.26
13 AA            8.59
14 AS            5.80
15 HA            4.90
16 US            3.78
```

```
data(airlines)
```

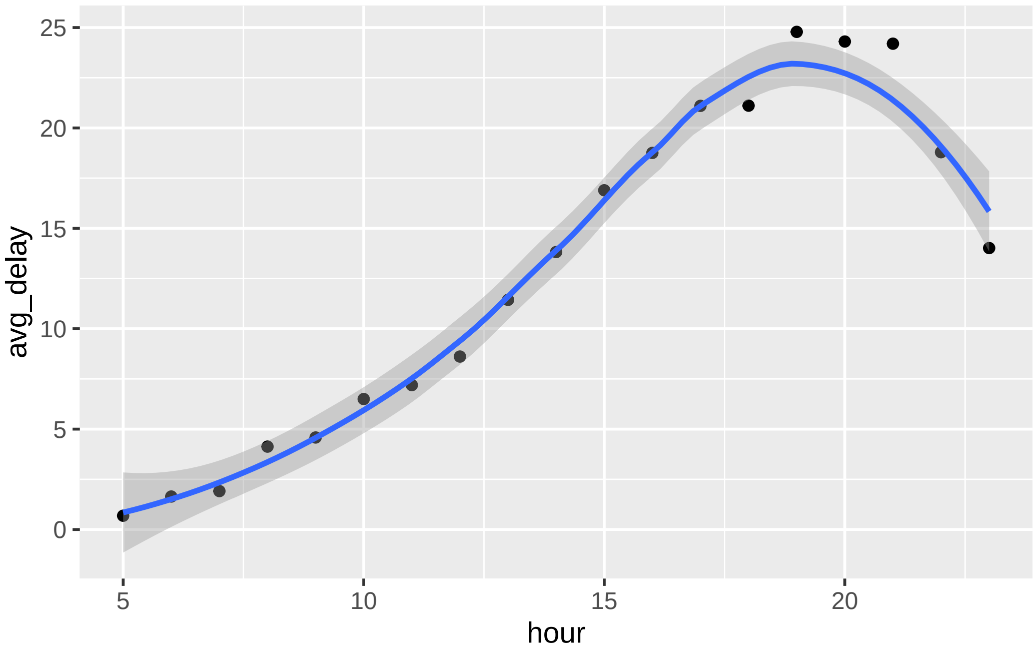
6. c. You should schedule your flights for 5 am to minimize the expected delay time.

```
library(ggplot2)

flights[!is.na(flights$dep_delay),] %>%
  summarize(.by = hour, avg_delay = mean(dep_delay)) %>%
  arrange(desc(avg_delay)) -> flights_delay

ggplot(flights_delay, aes(hour, avg_delay))+
  geom_point()+
  geom_smooth()
```

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



7. The result surprises me because I would have thought that the select function, like the rest of R, was case sensitive, but it is not. The default setting is for the helper functions is to be case sensitive, you can override this by using ignore.case.

```
flights |> select(contains("TIME"))
```

```
# A tibble: 336,776 x 6
   dep_time sched_dep_time arr_time sched_arr_time air_time time_hour
      <int>          <int>    <int>          <int>    <dbl> <dttm>
 1      517            515      830            819      227 2013-01-01 05:00:00
 2      533            529      850            830      227 2013-01-01 05:00:00
 3      542            540      923            850      160 2013-01-01 05:00:00
 4      544            545     1004           1022      183 2013-01-01 05:00:00
 5      554            600      812            837      116 2013-01-01 06:00:00
 6      554            558      740            728      150 2013-01-01 05:00:00
 7      555            600      913            854      158 2013-01-01 06:00:00
 8      557            600      709            723       53 2013-01-01 06:00:00
 9      557            600      838            846      140 2013-01-01 06:00:00
10      558            600      753            745      138 2013-01-01 06:00:00
# i 336,766 more rows
```