Stat. 450 Section 1 or 2: Homework 4

Prof. Eric A. Suess

So how should you complete your homework for this class?

- First thing to do is type all of your information about the problems you do in the text part of your R Notebook.
- Second thing to do is type all of your R code into R chunks that can be run.
- If you load the tidyverse in an R Notebook chunk, be sure to include the "message = FALSE" in the {r}, so {r message = FALSE}.
- Last thing is to spell check your R Notebook. Edit > Check Spelling... or hit the F7 key.

Homework 4:

```
Read: Chapter 5
Do 5.4.1 Exercise 4
Do 5.5.2 Exericise 1, 4
Do 5.6.7 Exercise 1
```

```r
library(tidyverse)
```

5.4.1

4.

Yes. The contains() helper function picks out all of the variables in the dataset that contains the word TIME. The function is also not case sensitive.

```r
library(nycflights13)

flights
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   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
flights %>% select(contains("TIME"))
```

```
## # A tibble: 336,776 x 6
```

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

The select() helpers are not case sensitive, when R is case sensitive.

To change the default. Don't know why it does not show the columns like above.

```r
flights %>% select(contains("TIME", ignore.case = FALSE))
```

```
## # A tibble: 336,776 x 0
```

5.5.2

1.

Minutes since midnight.

```
flights
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

Covert dep_time and sechedule_dep_time to minutes since midnight.

dep_time %/% 100 * 60 This give the minutes since midnight.

dep_time %% 100 This gives the reminder in minutes.

```
flights %>% mutate(dep_time_mins = ( ( (dep_time %/% 100) * 60 ) + (dep_time %% 100)),
                   sched_dep_time_mins = ( ( (sched_dep_time %/% 100) * 60 ) + (sched_dep_time %% 100))
```

```
## # A tibble: 336,776 x 21
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 336,766 more rows, and 14 more variables: sched_arr_time <int>,
## #   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_mins <dbl>,
## #   sched_dep_time_mins <dbl>
```

4.

Ten most delayed flights. There are no ties in these 10.

```
flights %>% arrange(desc(dep_delay)) %>%
  head(10)
```

```
## # A tibble: 10 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     9      641            900      1301     1242
##  2  2013     6    15     1432           1935      1137     1607
##  3  2013     1    10     1121           1635      1126     1239
##  4  2013     9    20     1139           1845      1014     1457
##  5  2013     7    22      845           1600      1005     1044
##  6  2013     4    10     1100           1900       960     1342
##  7  2013     3    17     2321            810       911      135
##  8  2013     6    27      959           1900       899     1236
##  9  2013     7    22     2257            759       898      121
## 10  2013    12     5      756           1700       896     1058
## # ... with 12 more variables: sched_arr_time <int>, 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>
```

5.6.7

1.

Brainstorm at least 5 different ways to assess the typical delay characteristics of a group of flights.

1. median and mean of dep_delay time in minutes.
2. sd of dep_delay time in minites
3. median and mean of arr_delay time in minutes.
4. sd of dep_delay time in minutes
5. is the distribution of arr_delay symmetric or skewed? Same questions for dep_delay?

Which is more important: arrival delay or departure delay?

Arrival delay is more important.

```
flights %>% select(dep_delay, arr_delay)  %>%
  summarize( n=n(), dep_delay_median = median(dep_delay, na.rm = TRUE),
                    dep_delay_mean = mean(dep_delay, na.rm = TRUE),
                    dep_delay_sd = sd(dep_delay, na.rm = TRUE),
                    arr_delay_median = median(arr_delay, na.rm = TRUE),
                    arr_delay_mean = mean(arr_delay, na.rm = TRUE),
                    arr_delay_sd = sd(arr_delay, na.rm = TRUE) )
```

```
## # A tibble: 1 x 7
##        n dep_delay_median dep_delay_mean dep_delay_sd arr_delay_median
##    <int>            <dbl>          <dbl>        <dbl>            <dbl>
## 1 336776               -2           12.6         40.2               -5
## # ... with 2 more variables: arr_delay_mean <dbl>, arr_delay_sd <dbl>
```

What proportion of flights are on time or arrive early? Approximtely 60% of all flights are on time.

```
flights %>% summarize(flt_ontime = mean(arr_delay <= 0, na.rm = TRUE) )
```

```
## # A tibble: 1 x 1
##   flt_ontime
##        <dbl>
## 1      0.594
```

Which arrier/airline has the best ontime rate?

```
flights %>% group_by(carrier) %>%
  summarize(flt_ontime = mean(arr_delay <= 0, na.rm = TRUE) ) %>%
  arrange(flt_ontime)
```

```
## # A tibble: 16 x 2
##    carrier flt_ontime
##    <chr>        <dbl>
##  1 FL           0.403
##  2 F9           0.424
##  3 EV           0.521
##  4 YV           0.526
##  5 MQ           0.533
##  6 WN           0.560
##  7 B6           0.563
##  8 UA           0.615
##  9 9E           0.616
## 10 US           0.629
```

```
## 11 OO           0.655
## 12 DL           0.656
## 13 VX           0.659
## 14 AA           0.665
## 15 HA           0.716
## 16 AS           0.733
```

What proportion of flight are 10 mins or more late?

```
flights %>% summarize(flt_late10 = mean(arr_delay >= 10, na.rm = TRUE) )
```

```
## # A tibble: 1 x 1
##   flt_late10
##        <dbl>
## 1      0.290
```

```
flights %>% group_by(carrier) %>%
  summarize(flt_late10 = mean(arr_delay >= 10, na.rm = TRUE) ) %>%
    arrange(flt_late10)
```

```
## # A tibble: 16 x 2
##    carrier flt_late10
##    <chr>        <dbl>
##  1 HA           0.190
##  2 AS           0.190
##  3 VX           0.229
##  4 DL           0.232
##  5 AA           0.234
##  6 US           0.235
##  7 OO           0.241
##  8 UA           0.271
##  9 9E           0.291
## 10 WN           0.307
## 11 B6           0.316
## 12 MQ           0.335
## 13 EV           0.366
## 14 YV           0.373
## 15 FL           0.426
## 16 F9           0.449
```

What proportion of flight are 30 mins or more late?

```
flights %>% summarize(flt_late30 = mean(arr_delay >= 30, na.rm = TRUE) )
```

```
## # A tibble: 1 x 1
##   flt_late30
##        <dbl>
## 1      0.161
```

```
flights %>% group_by(carrier) %>%
  summarize(flt_late30 = mean(arr_delay >= 30, na.rm = TRUE) ) %>%
  arrange(flt_late30)
```

```
## # A tibble: 16 x 2
##    carrier flt_late30
##    <chr>        <dbl>
##  1 HA          0.0585
##  2 AS          0.0931
```

```
##  3 US          0.107
##  4 DL          0.119
##  5 VX          0.122
##  6 AA          0.123
##  7 UA          0.141
##  8 WN          0.164
##  9 B6          0.178
## 10 MQ          0.181
## 11 9E          0.183
## 12 OO          0.207
## 13 FL          0.216
## 14 YV          0.232
## 15 EV          0.233
## 16 F9          0.254
```