Data Analysis Lab

Assignment Instructions Complete all questions below. After completing the assignment, knit your document, and download both your .Rmd and knitted output. Upload your files for peer review.

For each response, include comments detailing your response and what each line does.

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Question 1. Using the nycflights13 dataset, find all flights that departed in July, August, or September using the helper function between().

```
flights %>%
  filter(!is.na(dep_time)) %>%
  filter(between(month, 7, 9))
```

```
## # A tibble: 84,448 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     7     1        1       2029     212     236    2359     157 B6
##  2  2013     7     1        2       2359       3     344     344       0 B6
##  3  2013     7     1       29       2245     104     151       1     110 B6
##  4  2013     7     1       43       2130     193     322      14     188 B6
##  5  2013     7     1       44       2150     174     300     100     120 AA
##  6  2013     7     1       46       2051     235     304    2358     186 B6
##  7  2013     7     1       48       2001     287     308    2305     243 VX
##  8  2013     7     1       58       2155     183     335      43     172 B6
##  9  2013     7     1      100       2146     194     327      30     177 B6
## 10  2013     7     1      100       2245     135     337     135     122 B6
## # ... with 84,438 more rows, 9 more variables: flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, and abbreviated variable names
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
## # i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

Question 2. Using the nycflights13 dataset sort flights to find the 10 flights that flew the furthest. Put them in order of fastest to slowest.

```
flights %>%
  filter(!is.na(dep_time)) %>%
  arrange(desc(distance), desc(distance/(arr_time-dep_time))) %>%
  slice_head(n = 10)
```

```
## # A tibble: 10 x 19
##     year month   day dep_time sched_de~1 dep_d~2 arr_t~3 sched~4 arr_d~5 carrier
##    <int> <int> <int>    <int>      <int>   <dbl>   <int>   <int>   <dbl> <chr>
##  1  2013     6     6     1044       1000      44    1441    1435       6 HA
##  2  2013     6     8      951       1000      -9    1352    1435     -43 HA
##  3  2013     5     6      956       1000      -4    1358    1500     -62 HA
##  4  2013     6     7      952       1000      -8    1354    1435     -41 HA
##  5  2013     9     6      955       1000      -5    1359    1445     -46 HA
##  6  2013     7     4      950       1000     -10    1359    1430     -31 HA
##  7  2013    10     5     1002       1000       2    1418    1450     -32 HA
##  8  2013     8    27     1000       1000       0    1419    1440     -21 HA
##  9  2013     5     3     1001       1000       1    1424    1500     -36 HA
## 10  2013     5    31     1013       1000      13    1436    1500     -24 HA
## # ... with 9 more variables: flight <int>, tailnum <chr>, origin <chr>,
## #   dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour <dttm>, and abbreviated variable names 1: sched_dep_time,
## #   2: dep_delay, 3: arr_time, 4: sched_arr_time, 5: arr_delay
## # i Use `colnames()` to see all variable names
```

Question 3. Using the nycflights13 dataset, calculate a new variable called "hr_delay" and arrange the flights dataset in order of the arrival delays in hours (longest delays at the top). Put the new variable you created just before the departure time.Hint: use the experimental argument .before.

```
flights %>%
  mutate(hr_delay = arr_delay / 60 + dep_delay / 60 ) %>%
  arrange(desc(arr_delay)) %>%
  relocate(hr_delay, .before = dep_time)
```

```
## # A tibble: 336,776 x 20
##     year month   day hr_delay dep_time sched_d~1 dep_d~2 arr_t~3 sched~4 arr_d~5
##    <int> <int> <int>    <dbl>    <int>     <int>   <dbl>   <int>   <int>   <dbl>
##  1  2013     1     9     42.9      641       900    1301    1242    1530    1272
##  2  2013     6    15     37.7     1432      1935    1137    1607    2120    1127
##  3  2013     1    10     37.2     1121      1635    1126    1239    1810    1109
##  4  2013     9    20     33.7     1139      1845    1014    1457    2210    1007
##  5  2013     7    22     33.2      845      1600    1005    1044    1815     989
##  6  2013     4    10     31.5     1100      1900     960    1342    2211     931
##  7  2013     3    17     30.4     2321       810     911     135    1020     915
##  8  2013     7    22     29.9     2257       759     898     121    1026     895
##  9  2013    12     5     29.6      756      1700     896    1058    2020     878
## 10  2013     5     3     29.2     1133      2055     878    1250    2215     875
## # ... with 336,766 more rows, 10 more variables: carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>, and abbreviated variable names
```

```
## #   1: sched_dep_time, 2: dep_delay, 3: arr_time, 4: sched_arr_time,
## #   5: arr_delay
## # i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

Question 4. Using the nycflights13 dataset, find the most popular destinations (those with more than 2000 flights) and show the destination, the date info, the carrier. Then show just the number of flights for each popular destination.

```
#  find the most popular destinations (those with more than 2000 flights)
dest_2000 <- flights %>%
  group_by(dest) %>%
  summarise(total_flights=n()) %>%
  filter(total_flights > 2000)

# get the rest information
flights %>%
  filter(dest %in% dest_2000$dest) %>%
  select(dest, year, month, day, carrier)
```

```
## # A tibble: 302,969 x 5
##    dest   year month   day carrier
##    <chr> <int> <int> <int> <chr>
##  1 IAH    2013     1     1 UA
##  2 IAH    2013     1     1 UA
##  3 MIA    2013     1     1 AA
##  4 ATL    2013     1     1 DL
##  5 ORD    2013     1     1 UA
##  6 FLL    2013     1     1 B6
##  7 IAD    2013     1     1 EV
##  8 MCO    2013     1     1 B6
##  9 ORD    2013     1     1 AA
## 10 PBI    2013     1     1 B6
## # ... with 302,959 more rows
## # i Use `print(n = ...)` to see more rows
```

```
# Then show just the number of flights for each popular destination.
dest_2000
```

```
## # A tibble: 46 x 2
##    dest  total_flights
##    <chr>         <int>
##  1 ATL           17215
##  2 AUS            2439
##  3 BNA            6333
##  4 BOS           15508
##  5 BTV            2589
##  6 BUF            4681
##  7 CHS            2884
##  8 CLE            4573
##  9 CLT           14064
## 10 CMH            3524
## # ... with 36 more rows
## # i Use `print(n = ...)` to see more rows
```

Question 5. Using the nycflights13 dataset, find the flight information (flight number, origin, destination, carrier, number of flights in the year, and percent late) for the flight numbers with the highest percentage of arrival delays. Only include the flight numbers that have over 100 flights in the year.

```
# !!! the dataset contains the same flight with different origin:destination:carrier which is quite uns
df1 <- flights %>%
      group_by(flight) %>%
      summarise(
        total_flights = n(),
        percent_late = mean(arr_delay > 0, na.rm = TRUE)
        ) %>%
      filter(total_flights > 100) %>%
      slice_max(percent_late)

df2 <- flights %>%
  select(flight, origin, dest, carrier) %>%
  filter(flight %in% df1$flight) %>%
  distinct()

inner_join(df1,df2)
```

```
## Joining, by = "flight"
```

```
## # A tibble: 1 x 6
##   flight total_flights percent_late origin dest  carrier
##    <int>         <int>        <dbl> <chr>  <chr> <chr>
## 1   3075           162        0.728 JFK    CVG   MQ
```