In-class Assignment 9

Andrew Shao (NetID: as13381)

For Questions 1-5, we consider the relational data from the R package `nycflights13`.

Question 1 (1 pt): Add the locations (i.e. the `lat` and `lon`) of the origin and destination to `flights`.

```
flights %>%
  left_join(select(airports, faa, lat, lon), by = c('origin' = 'faa')) %>%
  rename_with(~paste0('origin_', .x), all_of(c('lat', 'lon'))) %>%
  left_join(select(airports, faa, lat, lon), by = c('dest' = 'faa')) %>%
  rename_with(~paste0('dest_', .x), all_of(c('lat', 'lon')))
```

Answer:

```
## # A tibble: 336,776 x 23
##     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      544            545        -1     1004           1022
##  5  2013     1     1      554            600        -6      812            837
##  6  2013     1     1      554            558        -4      740            728
##  7  2013     1     1      555            600        -5      913            854
##  8  2013     1     1      557            600        -3      709            723
##  9  2013     1     1      557            600        -3      838            846
## 10  2013     1     1      558            600        -2      753            745
## # i 336,766 more rows
## # i 15 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>, origin_lat <dbl>,
## #   origin_lon <dbl>, dest_lat <dbl>, dest_lon <dbl>
```

Question 2 (1 pt): Filter `flights` to only show flights with planes that have flown at least 100 flights.

```
flights %>% semi_join(flights %>%
                        group_by(tailnum) %>%
                        summarise(nflights = n()) %>%
```

```
                        filter(nflights >= 100),
                      by = 'tailnum')
```

Answer:

```
## # A tibble: 230,902 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      544            545        -1     1004           1022
##  4  2013     1     1      554            558        -4      740            728
##  5  2013     1     1      555            600        -5      913            854
##  6  2013     1     1      557            600        -3      709            723
##  7  2013     1     1      557            600        -3      838            846
##  8  2013     1     1      558            600        -2      849            851
##  9  2013     1     1      558            600        -2      853            856
## 10  2013     1     1      558            600        -2      923            937
## # i 230,892 more rows
## # 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>
```

Question 3 (1 pt): What does `anti_join(flights, airports, by = c("dest" = "faa"))` tell you? What does `anti_join(airports, flights, by = c("faa" = "dest"))` tell you?

Answer: `anti_join(flights, airports, by = c("dest" = "faa"))` shows the flights whose destination is at an airport that is not included in the `airports` table.
`anti_join(airports, flights, by = c("faa" = "dest"))` shows the airports in `airports` which there are no flights in the `flights` table that have them as destinations.

```
anti_join(flights, airports, by = c("dest" = "faa"))
```

```
## # A tibble: 7,602 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      544            545        -1     1004           1022
##  2  2013     1     1      615            615         0     1039           1100
##  3  2013     1     1      628            630        -2     1137           1140
##  4  2013     1     1      701            700         1     1123           1154
##  5  2013     1     1      711            715        -4     1151           1206
##  6  2013     1     1      820            820         0     1254           1310
##  7  2013     1     1      820            820         0     1249           1329
##  8  2013     1     1      840            845        -5     1311           1350
##  9  2013     1     1      909            810        59     1331           1315
## 10  2013     1     1      913            918        -5     1346           1416
## # i 7,592 more rows
## # 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
anti_join(airports, flights, by = c("faa" = "dest"))
```

```
## # A tibble: 1,357 x 8
##    faa   name                            lat    lon   alt    tz dst   tzone
##    <chr> <chr>                         <dbl>  <dbl> <dbl> <dbl> <chr> <chr>
##  1 04G   Lansdowne Airport              41.1  -80.6  1044    -5 A     America/~
##  2 06A   Moton Field Municipal Airport  32.5  -85.7   264    -6 A     America/~
##  3 06C   Schaumburg Regional            42.0  -88.1   801    -6 A     America/~
##  4 06N   Randall Airport                41.4  -74.4   523    -5 A     America/~
##  5 09J   Jekyll Island Airport          31.1  -81.4    11    -5 A     America/~
##  6 0A9   Elizabethton Municipal Airport 36.4  -82.2  1593    -5 A     America/~
##  7 0G6   Williams County Airport        41.5  -84.5   730    -5 A     America/~
##  8 0G7   Finger Lakes Regional Airport  42.9  -76.8   492    -5 A     America/~
##  9 0P2   Shoestring Aviation Airfield   39.8  -76.6  1000    -5 U     America/~
## 10 0S9   Jefferson County Intl          48.1 -123.    108    -8 A     America/~
## # i 1,347 more rows
```

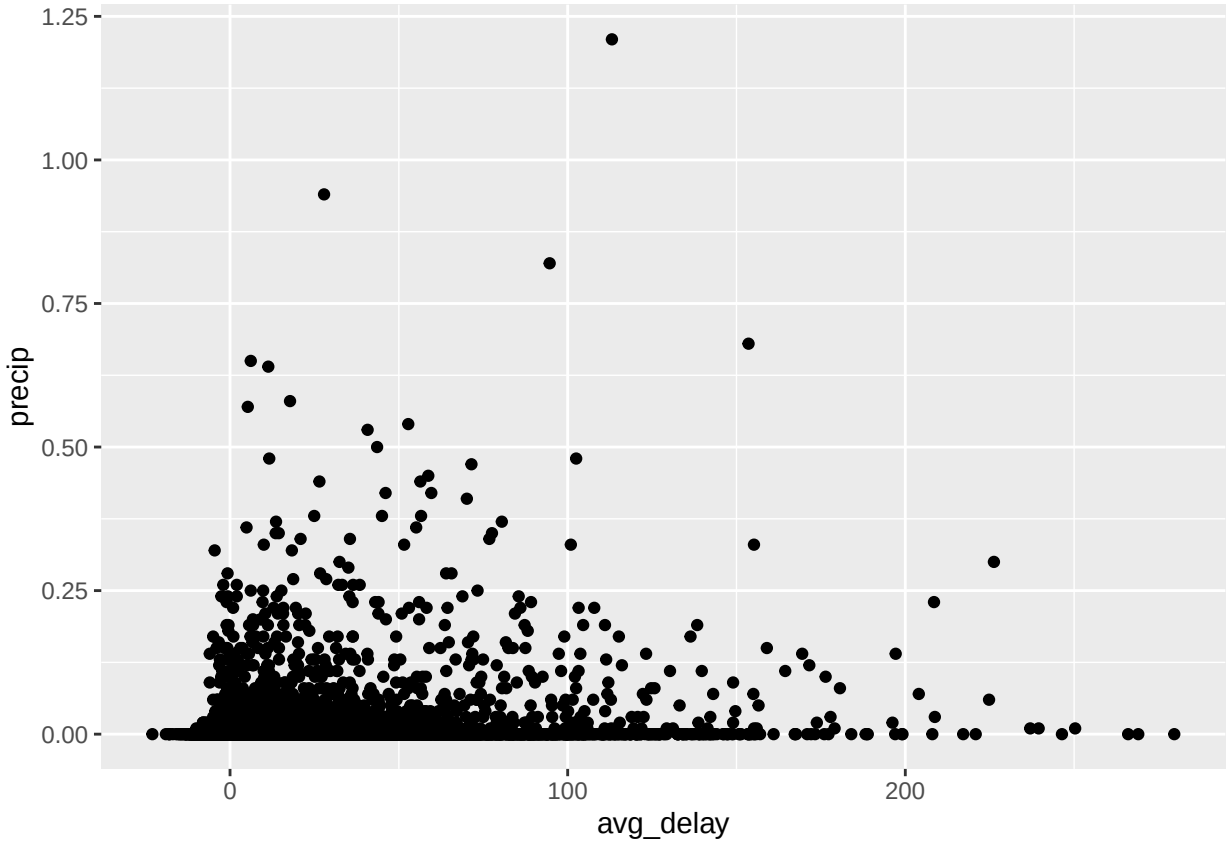
Question 4 (1 pt): Draw a plot for the average of departure-delay hours (`flights$dep_delay`) vs. the precipitation amount (`weather$precip`).

```r
flights %>%
  group_by(year, month, day, hour, origin) %>%
  summarise(avg_delay = mean(dep_delay, na.rm = T)) %>%
  inner_join(weather, by = c('year', 'month', 'day', 'hour', 'origin')) %>%
  ggplot() +
  geom_point(aes(x = avg_delay, y = precip))
```

Answer:

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

```
## Warning: Removed 52 rows containing missing values (`geom_point()`).
```



Question 5 (1 pt): Create data frames `flight1` and `flight2` using the following code. How many unique observations do these two data frames contain in total?

```
flights_daily <- flights%>%select(year:day, origin, dest)
flight1 <- flights_daily[1:1e5,]
flight2 <- flights_daily[5e4:2e5,]
```

```
nrow(union(flight1, flight2))
```

Answer:

```
## [1] 37995
```