Applied Problem Set # 1

Mia Jiang and Emma Van Lieshout

27/03/2022

Clear Global Environment

Working Directory and Loading Packages

```
# Setting the Working Directory
setwd("~/Desktop/DPPP R/Week 3/applied-problem-set-1-emma-and-mia")
# Load packages
library(tidyverse)
library(ggplot2)
library(dplyr)
library(ggrepel)
```

Front matter This submission is my work alone and complies with the 30535 integrity policy.

Add your initials to indicate your agreement: EKVL, MJ

Late coins used this pset: 0. Late coins left: 9.

Problems

Git Merge Conflicts

1

We had a conflict because we wrote different things (our different names, for example) in the same place/line in the code.

Flight Data: Part 1

Download BTS Data

```
il_flights1 <- read_csv("il_flights_2016_1.csv")
il_flights2 <- read_csv("il_flights_2016_2.csv")
il_flights3 <- read_csv("il_flights_2016_3.csv")
il_flights4 <- read_csv("il_flights_2016_4.csv")
```

```
il_flights5 <- read_csv("il_flights_2016_5.csv")
il_flights6 <- read_csv("il_flights_2016_6.csv")
il_flights7 <- read_csv("il_flights_2016_7.csv")
il_flights8 <- read_csv("il_flights_2016_8.csv")
il_flights9 <- read_csv("il_flights_2016_9.csv")
il_flights10 <- read_csv("il_flights_2016_10.csv")
il_flights11 <- read_csv("il_flights_2016_11.csv")
il_flights12 <- read_csv("il_flights_2016_12.csv")

il_flights <- bind_rows(il_flights1, il_flights2, il_flights3,
                        il_flights4, il_flights5, il_flights6,
                        il_flights7, il_flights8, il_flights9,
                        il_flights10, il_flights11, il_flights12)
il_flights <- select(il_flights, -...31)
```

Data Description

2.2.1

```
names(il_flights)
```

```
##  [1] "YEAR"                  "MONTH"                 "DAY_OF_MONTH"
##  [4] "OP_UNIQUE_CARRIER"     "ORIGIN_AIRPORT_ID"     "ORIGIN_AIRPORT_SEQ_ID"
##  [7] "ORIGIN_CITY_MARKET_ID" "ORIGIN"                "ORIGIN_CITY_NAME"
## [10] "ORIGIN_STATE_ABR"      "DEST_AIRPORT_ID"       "DEST_AIRPORT_SEQ_ID"
## [13] "DEST_CITY_MARKET_ID"   "DEST"                  "DEST_CITY_NAME"
## [16] "DEST_STATE_ABR"        "DEP_TIME"              "DEP_DELAY"
## [19] "DEP_DELAY_NEW"         "ARR_TIME"              "ARR_DELAY"
## [22] "ARR_DELAY_NEW"         "CANCELLED"             "AIR_TIME"
## [25] "DISTANCE"              "CARRIER_DELAY"         "WEATHER_DELAY"
## [28] "NAS_DELAY"             "SECURITY_DELAY"        "LATE_AIRCRAFT_DELAY"
```

```
#View(il_flights)
```

I have also viewed the ReadMe file that came along with the data, and nowhere can I find a unique identifier for each flight. The row numbers are unique to each observation, but they are not an actual column in the data. We could make this into an identifier my using mutate() in combination with the function row_number(). (Function found here: https://bookdown.org/yih_huynh/Guide-to-R-Book/row-number.html)

2.2.2.1

```
print(il_flights)
```

```
## # A tibble: 675,822 x 30
##     YEAR MONTH DAY_OF_MONTH OP_UNIQUE_CARRIER ORIGIN_AIRPORT_ID ORIGIN_AIRPORT_~
##    <dbl> <dbl>        <dbl> <chr>                          <dbl>            <dbl>
##  1  2016    11            1 AA                             13930          1393004
##  2  2016    11            2 AA                             13930          1393004
##  3  2016    11            3 AA                             13930          1393004
##  4  2016    11            4 AA                             13930          1393004
```

```
##  5  2016    11            5 AA                              13930          1393004
##  6  2016    11            6 AA                              13930          1393004
##  7  2016    11            7 AA                              13930          1393004
##  8  2016    11            8 AA                              13930          1393004
##  9  2016    11            9 AA                              13930          1393004
## 10  2016    11           10 AA                              13930          1393004
## # ... with 675,812 more rows, and 24 more variables:
## #   ORIGIN_CITY_MARKET_ID <dbl>, ORIGIN <chr>, ORIGIN_CITY_NAME <chr>,
## #   ORIGIN_STATE_ABR <chr>, DEST_AIRPORT_ID <dbl>, DEST_AIRPORT_SEQ_ID <dbl>,
## #   DEST_CITY_MARKET_ID <dbl>, DEST <chr>, DEST_CITY_NAME <chr>,
## #   DEST_STATE_ABR <chr>, DEP_TIME <chr>, DEP_DELAY <dbl>, DEP_DELAY_NEW <dbl>,
## #   ARR_TIME <chr>, ARR_DELAY <dbl>, ARR_DELAY_NEW <dbl>, CANCELLED <dbl>,
## #   AIR_TIME <dbl>, DISTANCE <dbl>, CARRIER_DELAY <dbl>, ...
```

```
head(il_flights)
```

```
## # A tibble: 6 x 30
##    YEAR MONTH DAY_OF_MONTH OP_UNIQUE_CARRIER ORIGIN_AIRPORT_ID ORIGIN_AIRPORT_S~
##   <dbl> <dbl>        <dbl> <chr>                          <dbl>             <dbl>
## 1  2016    11            1 AA                              13930          1393004
## 2  2016    11            2 AA                              13930          1393004
## 3  2016    11            3 AA                              13930          1393004
## 4  2016    11            4 AA                              13930          1393004
## 5  2016    11            5 AA                              13930          1393004
## 6  2016    11            6 AA                              13930          1393004
## # ... with 24 more variables: ORIGIN_CITY_MARKET_ID <dbl>, ORIGIN <chr>,
## #   ORIGIN_CITY_NAME <chr>, ORIGIN_STATE_ABR <chr>, DEST_AIRPORT_ID <dbl>,
## #   DEST_AIRPORT_SEQ_ID <dbl>, DEST_CITY_MARKET_ID <dbl>, DEST <chr>,
## #   DEST_CITY_NAME <chr>, DEST_STATE_ABR <chr>, DEP_TIME <chr>,
## #   DEP_DELAY <dbl>, DEP_DELAY_NEW <dbl>, ARR_TIME <chr>, ARR_DELAY <dbl>,
## #   ARR_DELAY_NEW <dbl>, CANCELLED <dbl>, AIR_TIME <dbl>, DISTANCE <dbl>,
## #   CARRIER_DELAY <dbl>, WEATHER_DELAY <dbl>, NAS_DELAY <dbl>, ...
```

```
str(il_flights)
```

```
## tibble [675,822 x 30] (S3: tbl_df/tbl/data.frame)
##  $ YEAR                : num [1:675822] 2016 2016 2016 2016 2016 ...
##  $ MONTH               : num [1:675822] 11 11 11 11 11 11 11 11 11 11 ...
##  $ DAY_OF_MONTH        : num [1:675822] 1 2 3 4 5 6 7 8 9 10 ...
##  $ OP_UNIQUE_CARRIER   : chr [1:675822] "AA" "AA" "AA" "AA" ...
##  $ ORIGIN_AIRPORT_ID   : num [1:675822] 13930 13930 13930 13930 13930 ...
##  $ ORIGIN_AIRPORT_SEQ_ID: num [1:675822] 1393004 1393004 1393004 1393004 1393004 ...
##  $ ORIGIN_CITY_MARKET_ID: num [1:675822] 30977 30977 30977 30977 30977 ...
##  $ ORIGIN              : chr [1:675822] "ORD" "ORD" "ORD" "ORD" ...
##  $ ORIGIN_CITY_NAME    : chr [1:675822] "Chicago, IL" "Chicago, IL" "Chicago, IL" "Chicago, IL" ...
##  $ ORIGIN_STATE_ABR    : chr [1:675822] "IL" "IL" "IL" "IL" ...
##  $ DEST_AIRPORT_ID     : num [1:675822] 12889 12889 12889 12889 12889 ...
##  $ DEST_AIRPORT_SEQ_ID : num [1:675822] 1288903 1288903 1288903 1288903 1288903 ...
##  $ DEST_CITY_MARKET_ID : num [1:675822] 32211 32211 32211 32211 32211 ...
##  $ DEST                : chr [1:675822] "LAS" "LAS" "LAS" "LAS" ...
##  $ DEST_CITY_NAME      : chr [1:675822] "Las Vegas, NV" "Las Vegas, NV" "Las Vegas, NV" "Las Vegas,
##  $ DEST_STATE_ABR      : chr [1:675822] "NV" "NV" "NV" "NV" ...
##  $ DEP_TIME            : chr [1:675822] "1702" "1818" "1659" "1706" ...
```

```
##  $ DEP_DELAY            : num [1:675822] -3 73 -6 1 -5 -1 -2 0 30 -1 ...
##  $ DEP_DELAY_NEW        : num [1:675822] 0 73 0 1 0 0 0 0 30 0 ...
##  $ ARR_TIME             : chr [1:675822] "1903" "2105" "1843" "1844" ...
##  $ ARR_DELAY            : num [1:675822] 4 126 -16 -23 -27 -32 -43 -31 -23 -47 ...
##  $ ARR_DELAY_NEW        : num [1:675822] 4 126 0 0 0 0 0 0 0 0 ...
##  $ CANCELLED            : num [1:675822] 0 0 0 0 0 0 0 0 0 0 ...
##  $ AIR_TIME             : num [1:675822] 212 193 200 196 195 192 182 175 169 179 ...
##  $ DISTANCE             : num [1:675822] 1514 1514 1514 1514 1514 ...
##  $ CARRIER_DELAY        : num [1:675822] NA 0 NA NA NA NA NA NA NA NA ...
##  $ WEATHER_DELAY        : num [1:675822] NA 0 NA NA NA NA NA NA NA NA ...
##  $ NAS_DELAY            : num [1:675822] NA 53 NA NA NA NA NA NA NA NA ...
##  $ SECURITY_DELAY       : num [1:675822] NA 0 NA NA NA NA NA NA NA NA ...
##  $ LATE_AIRCRAFT_DELAY  : num [1:675822] NA 73 NA NA NA NA NA NA NA NA ...
```

```
glimpse(il_flights)
```

```
## Rows: 675,822
## Columns: 30
## $ YEAR                  <dbl> 2016, 2016, 2016, 2016, 2016, 2016, 2016, 2016, ~
## $ MONTH                 <dbl> 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, ~
## $ DAY_OF_MONTH          <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1~
## $ OP_UNIQUE_CARRIER     <chr> "AA", "AA", "AA", "AA", "AA", "AA", "AA", "AA", ~
## $ ORIGIN_AIRPORT_ID     <dbl> 13930, 13930, 13930, 13930, 13930, 13930, 13930,~
## $ ORIGIN_AIRPORT_SEQ_ID <dbl> 1393004, 1393004, 1393004, 1393004, 1393004, 139~
## $ ORIGIN_CITY_MARKET_ID <dbl> 30977, 30977, 30977, 30977, 30977, 30977, 30977,~
## $ ORIGIN                <chr> "ORD", "ORD", "ORD", "ORD", "ORD", "ORD", "ORD",~
## $ ORIGIN_CITY_NAME      <chr> "Chicago, IL", "Chicago, IL", "Chicago, IL", "Ch~
## $ ORIGIN_STATE_ABR      <chr> "IL", "IL", "IL", "IL", "IL", "IL", "IL", "IL", ~
## $ DEST_AIRPORT_ID       <dbl> 12889, 12889, 12889, 12889, 12889, 12889, 12889,~
## $ DEST_AIRPORT_SEQ_ID   <dbl> 1288903, 1288903, 1288903, 1288903, 1288903, 128~
## $ DEST_CITY_MARKET_ID   <dbl> 32211, 32211, 32211, 32211, 32211, 32211, 32211,~
## $ DEST                  <chr> "LAS", "LAS", "LAS", "LAS", "LAS", "LAS", "LAS",~
## $ DEST_CITY_NAME        <chr> "Las Vegas, NV", "Las Vegas, NV", "Las Vegas, NV~
## $ DEST_STATE_ABR        <chr> "NV", "NV", "NV", "NV", "NV", "NV", "NV", "NV", ~
## $ DEP_TIME              <chr> "1702", "1818", "1659", "1706", "1820", "1704", ~
## $ DEP_DELAY             <dbl> -3, 73, -6, 1, -5, -1, -2, 0, 30, -1, 4, -1, 0, ~
## $ DEP_DELAY_NEW         <dbl> 0, 73, 0, 1, 0, 0, 0, 0, 30, 0, 4, 0, 0, 9, 0, 0~
## $ ARR_TIME              <chr> "1903", "2105", "1843", "1844", "2000", "1835", ~
## $ ARR_DELAY             <dbl> 4, 126, -16, -23, -27, -32, -43, -31, -23, -47, ~
## $ ARR_DELAY_NEW         <dbl> 4, 126, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4~
## $ CANCELLED             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ AIR_TIME              <dbl> 212, 193, 200, 196, 195, 192, 182, 175, 169, 179~
## $ DISTANCE              <dbl> 1514, 1514, 1514, 1514, 1514, 1514, 1514, 1514, ~
## $ CARRIER_DELAY         <dbl> NA, 0, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ WEATHER_DELAY         <dbl> NA, 0, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ NAS_DELAY             <dbl> NA, 53, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ SECURITY_DELAY        <dbl> NA, 0, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ LATE_AIRCRAFT_DELAY   <dbl> NA, 73, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
```

```
#View(il_flights)
```

```
summary(il_flights)
```

```
##       YEAR          MONTH         DAY_OF_MONTH   OP_UNIQUE_CARRIER
##  Min.   :2016   Min.   : 1.000   Min.   : 1.00   Length:675822
##  1st Qu.:2016   1st Qu.: 4.000   1st Qu.: 8.00   Class :character
##  Median :2016   Median : 7.000   Median :16.00   Mode  :character
##  Mean   :2016   Mean   : 6.575   Mean   :15.76
##  3rd Qu.:2016   3rd Qu.: 9.000   3rd Qu.:23.00
##  Max.   :2016   Max.   :12.000   Max.   :31.00
##
##  ORIGIN_AIRPORT_ID ORIGIN_AIRPORT_SEQ_ID ORIGIN_CITY_MARKET_ID
##  Min.   :10135     Min.   :1013503       Min.   :30135
##  1st Qu.:12889     1st Qu.:1288903       1st Qu.:30977
##  Median :13930     Median :1393004       Median :30977
##  Mean   :13194     Mean   :1319434       Mean   :31427
##  3rd Qu.:13930     3rd Qu.:1393004       3rd Qu.:31650
##  Max.   :15919     Max.   :1591902       Max.   :35497
##
##     ORIGIN          ORIGIN_CITY_NAME   ORIGIN_STATE_ABR   DEST_AIRPORT_ID
##  Length:675822      Length:675822      Length:675822      Min.   :10135
##  Class :character   Class :character   Class :character   1st Qu.:12889
##  Mode  :character   Mode  :character   Mode  :character   Median :13930
##                                                           Mean   :13192
##                                                           3rd Qu.:13930
##                                                           Max.   :15919
##
##  DEST_AIRPORT_SEQ_ID DEST_CITY_MARKET_ID     DEST           DEST_CITY_NAME
##  Min.   :1013503     Min.   :30135       Length:675822      Length:675822
##  1st Qu.:1288903     1st Qu.:30977       Class :character   Class :character
##  Median :1393004     Median :30977       Mode  :character   Mode  :character
##  Mean   :1319240     Mean   :31426
##  3rd Qu.:1393004     3rd Qu.:31650
##  Max.   :1591902     Max.   :35497
##
##  DEST_STATE_ABR       DEP_TIME           DEP_DELAY       DEP_DELAY_NEW
##  Length:675822      Length:675822      Min.   : -58.00   Min.   :   0.00
##  Class :character   Class :character   1st Qu.:  -5.00   1st Qu.:   0.00
##  Mode  :character   Mode  :character   Median :  -2.00   Median :   0.00
##                                        Mean   :  10.73   Mean   :  13.46
##                                        3rd Qu.:   8.00   3rd Qu.:   8.00
##                                        Max.   :1964.00   Max.   :1964.00
##                                        NA's   :11072     NA's   :11072
##    ARR_TIME           ARR_DELAY        ARR_DELAY_NEW       CANCELLED
##  Length:675822      Min.   : -77.000   Min.   :   0.00   Min.   :0.000
##  Class :character   1st Qu.: -15.000   1st Qu.:   0.00   1st Qu.:0.000
##  Mode  :character   Median :  -6.000   Median :   0.00   Median :0.000
##                     Mean   :   4.967   Mean   :  13.41   Mean   :0.017
##                     3rd Qu.:   8.000   3rd Qu.:   8.00   3rd Qu.:0.000
##                     Max.   :1971.000   Max.   :1971.00   Max.   :1.000
##                     NA's   :13272      NA's   :13272
##     AIR_TIME        DISTANCE      CARRIER_DELAY    WEATHER_DELAY
##  Min.   : 13.0   Min.   :  67.0   Min.   :   0.0   Min.   :   0.0
##  1st Qu.: 68.0   1st Qu.: 413.0   1st Qu.:   0.0   1st Qu.:   0.0
```

```
##  Median :101.0   Median : 717.0   Median :   0.0   Median :   0.0
##  Mean   :113.4   Mean   : 814.2   Mean   :  18.8   Mean   :   3.5
##  3rd Qu.:143.0   3rd Qu.:1012.0   3rd Qu.:  18.0   3rd Qu.:   0.0
##  Max.   :574.0   Max.   :4243.0   Max.   :1964.0   Max.   :1157.0
##  NA's   :13272                    NA's   :548614   NA's   :548614
##    NAS_DELAY      SECURITY_DELAY   LATE_AIRCRAFT_DELAY
##  Min.   :   0.0   Min.   :  0      Min.   :   0.0
##  1st Qu.:   0.0   1st Qu.:  0      1st Qu.:   0.0
##  Median :   3.5   Median :  0      Median :   0.0
##  Mean   :  17.5   Mean   :  0      Mean   :  24.6
##  3rd Qu.:  21.0   3rd Qu.:  0      3rd Qu.:  30.0
##  Max.   :1233.0   Max.   :242      Max.   :1420.0
##  NA's   :548614   NA's   :548614   NA's   :548614
```

Pairs: str() and glimpse(), print9) and head() and View() (though these are all slightly different, they basically allow you to look at the raw data. View() has a bit of added functionality it and head() doesn't show the whole data.) Of the str() and glimpse() pair, I think glimpse() gives the better output because it gives the dimensions of the data in a more digestible format. I think View() is the best option of the other group because you can go back to the tab any time without having to call the command again, it doesn't clutter up your markdown document, and has the added functionality I talked about above.

2.2.2.2

print() shows the entire dataset in the R markdown document, similar to just calling the name of the dataset.

head() prints the first 6 rows of the dataset. This is useful to see what kind of values each column takes on.

str() gives me a list of all the variables, the data types, the length of each column vector, and the first few values for each variable.

glimpse() gives me column names, data types, and the first several values. It also reports the dimensions of the data.

View() opens the data in a new window in R, where I can scroll through. I an also search for things, and arrange and filter the data in that view.

summary() tells me some summary statistics for each numerical variable in the data (min/max, quartiles, mean and median, and the number of missing values).

Data Validation

2.3.1

```
library(testthat)
test_that("We have the right number of rows", expect_equal(nrow(il_flights), 675822))
```

```
## Test passed
```

2.3.2

```
test_that("All flights to or from IL airports",
          expect_equal(
            nrow(
              filter(il_flights, DEST_STATE_ABR =="IL"
                     | ORIGIN_STATE_ABR == "IL"
```

```
                    )
                  ),
            675822))
```

```
## Test passed
```

2.3.3

```
not_chicago_flights <-
  filter(il_flights,
         ORIGIN != "MDW" &
         ORIGIN != "ORD" &
         DEST != "MDW" &
         DEST != "ORD"
         )
nrow(not_chicago_flights)
```

```
## [1] 12240
```

12240 flights are left.

2.3.4

```
not_chicago_flights %>%
  group_by(DEST)%>%
  summarise(n_flights = n())%>%
  arrange(desc(n_flights))%>%
  head(5)
```

```
## # A tibble: 5 x 2
##   DEST  n_flights
##   <chr>     <int>
## 1 ATL        2968
## 2 PIA        1999
## 3 MLI        1926
## 4 BMI        1517
## 5 DTW        1228
```

```
not_chicago_flights %>%
  group_by(ORIGIN)%>%
  summarise(n_flights = n())%>%
  arrange(desc(n_flights))%>%
  head(5)
```

```
## # A tibble: 5 x 2
##   ORIGIN n_flights
##   <chr>      <int>
## 1 ATL         2966
## 2 PIA         2001
## 3 MLI         1984
## 4 BMI         1517
## 5 DTW         1233
```

The most common origins and destinations are the same: ATL, PIA, MLI, BMI, and CTW. These are the Atlanta, Peoria, Quad Cities (in Moline), Central Illinois Regional Airport (in Bloomington), and Detroit Airports.

The middle three are Illinois airports (though Quad Cities is kind of also an Iowa airport, since it is just across the border). They are the reason they are included at all in the IL flights dataset, so it makes sense that they would make up a large share of the IL flights. Atlanta is the first largest probably because it generally sees a huge volume of flights, so at each of the middle three airports and other IL airports would be getting a large volume of their traffic from or to Atlanta. Combining all the IL airports' traffic to Atlanta makes it the top of the list. Detroit may be on there for a similar reason: it's the largest close-by airport that isn't in Chicago (which we filtered out) so it stands to reason that a lot of connecting flights to or from smaller Illinois airports go through there.

2.3.5

The FAA reports the average daily capacity was 1,093 on average from 2016-2020 at Midway and 3,365 at O'Hare.

```
MDW_flights_yr_FAA <- 1093*365
MDW_flights_yr_FAA
```

```
## [1] 398945
```

```
ORD_flights_yr_FAA <- 3365*365
ORD_flights_yr_FAA
```

```
## [1] 1228225
```

The FAA data can be found here: https://www.faa.gov/air_traffic/by_the_numbers/media/Air_Traffic_by_the_Numbers_2021.pdf.

The Chicago Department of Aviation says there were 867,635 total flights at O'Hare in 2016. There were 253,046 total flights at Midway in 2016. The data can be found here: https://www.flychicago.com/business/CDA/factsfigures/Pages/airtraffic.aspx.

```
MDW_flights_yr_CDA <- 253046
ORD_flights_yr_CDA <- 867635
```

These estimates differ, perhaps because the volume of flights has grown since 2016, when the CDA data is from, which would make the FAA's 2016-2020 average larger. I am also not quite sure how "capacity" is measured versus the numbers of flights reported in the total from the CDA, so perhaps the CDA excluded certain types of flights. Or, the FAA made a projection of how many flights there might be rather than actually counting how many there were.

Now, We'll filter to only Chicago airports and see how many flights are in the data.

```
chicago_flights <-
  filter(
         il_flights,
         ORIGIN == "MDW" |
         ORIGIN == "ORD" |
         DEST == "MDW" |
         DEST == "ORD"
         )
```

```
MDW_flights_yr_BTS <-
  nrow(chicago_flights %>%
       filter(
              ORIGIN == "MDW" |
              DEST == "MDW"
              )
     )
MDW_flights_yr_BTS
```

```
## [1] 175611
```

```
ORD_flights_yr_BTS <-
  nrow(chicago_flights %>%
       filter(
              ORIGIN == "ORD" |
              DEST == "ORD"
              )
     )
ORD_flights_yr_BTS
```

```
## [1] 487971
```

```
MDW_flights_yr_FAA/ORD_flights_yr_FAA
```

```
## [1] 0.3248143
```

```
MDW_flights_yr_CDA/ORD_flights_yr_CDA
```

```
## [1] 0.2916503
```

```
MDW_flights_yr_BTS/ORD_flights_yr_BTS
```

```
## [1] 0.35988
```

The BTS numbers (175,611 for MDW and 487,971 for ORD) are smaller than both estimates I found on line. I think the reason they don't agree is because the data simply chooses which flights count and which flights don't differently. I think it's plausible that the two datasets I have here, which are coming from an "airport facility" perspective count more than just passenger flights, while the BTS data only counts passenger flights. But, I don't know airline terminology well enough to know for sure what is included in the totals I found online. The amount of data does kind of match what I expect in that the ration between flights at O'Hare and Midway is similar across all data: Midway has about 30% of the number of flights that O'Hare has.

2.3.6

According to an online data source, the highest volume airlines out of O'Hare (in 2018) were United, American, and Delta. I couldn't find data for an earlier year. Southwest, Delta, and Porter airlines had the most flights at Midway. The data is here: https://thepointsguy.com/guide/chicago-ohare-vs-midway-which-airport-should-i-fly-into/#:~:text=United%20operated%20almost%20700%2C000%20flights,distant%20second%20with%201

Here's what the BTS data has to say:

```
midway_carriers <-
  chicago_flights %>%
  filter(
         ORIGIN == "MDW" |
         DEST == "MDW"
         )%>%
  group_by(OP_UNIQUE_CARRIER)%>%
  summarise(n_flights = n())%>%
  arrange(desc(n_flights))
head(midway_carriers, 3)
```

```
## # A tibble: 3 x 2
##   OP_UNIQUE_CARRIER n_flights
##   <chr>                 <int>
## 1 WN                   167239
## 2 DL                     5166
## 3 EV                     1944
```

```
ohare_carriers <-
  chicago_flights %>%
  filter(
         ORIGIN == "ORD" |
         DEST == "ORD"
         )%>%
  group_by(OP_UNIQUE_CARRIER)%>%
  summarise(n_flights = n())%>%
  arrange(desc(n_flights))
head(ohare_carriers, 3)
```

```
## # A tibble: 3 x 2
##   OP_UNIQUE_CARRIER n_flights
##   <chr>                 <int>
## 1 UA                   141500
## 2 AA                   123990
## 3 OO                    83359
```

The top two airlines for each airport (Southwest and Delta for Midway and United and American for O'Hare) are the same as what I found online. However, the third ones don't match. Skywest is the third highest volume for O'Hare and ExpressJet is the third highest volume for Midway. I think they disagree because of the differing years of the data. Among the less-used airlines there may be more competition, causing the amount of flights they have each year to fluctuate more relative to each other than the big airlines. Because of the changes and the slim differences between smaller airlines, different carriers will drop in and out of the third-place spot depending on the year.

3

```
# Filter dataset
chicago_flights_to <- chicago_flights %>%
  filter(DEST %in% c("ORD", "MDW"))
```

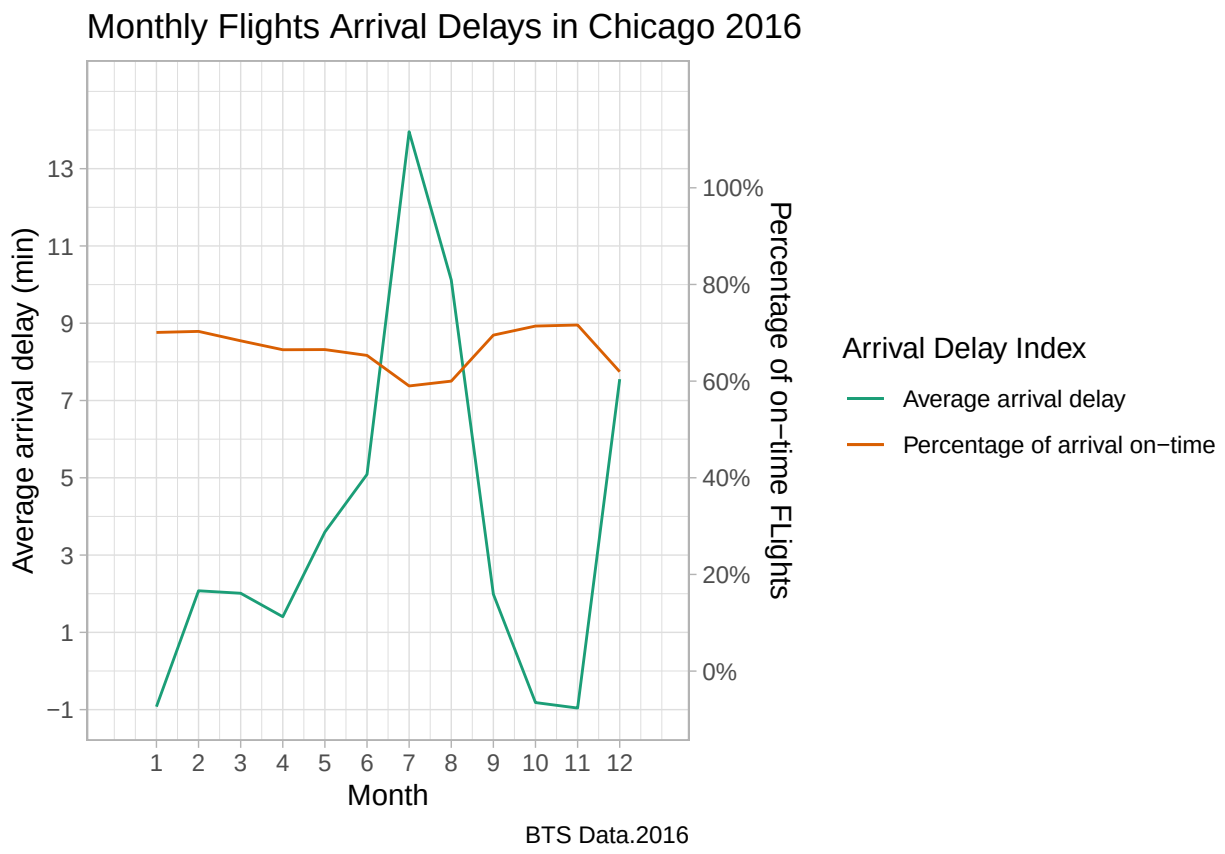
3.1

```
# Query
arr_delay_by_month <- chicago_flights_to %>%
  drop_na(ARR_DELAY) %>%
  mutate(arr_ontime = ifelse(ARR_DELAY <= 0,
                             1,
                             0)) %>%
  group_by(MONTH) %>%
  summarise(mean_arr_delay = mean(ARR_DELAY), # lowest average arrivals delays
            pct_arr_ontime = mean(arr_ontime) # 80% of flights on time
             ) %>%
  arrange(mean_arr_delay,desc(pct_arr_ontime))
arr_delay_by_month
```

```
## # A tibble: 12 x 3
##    MONTH mean_arr_delay pct_arr_ontime
##    <dbl>          <dbl>          <dbl>
##  1    11         -0.959          0.716
##  2     1         -0.924          0.701
##  3    10         -0.817          0.714
##  4     4          1.40           0.665
##  5     9          1.99           0.695
##  6     3          2.01           0.684
##  7     2          2.08           0.703
##  8     5          3.60           0.665
##  9     6          5.09           0.653
## 10    12          7.55           0.620
## 11     8         10.1            0.600
## 12     7         14.0            0.590
```

```
# Plot
ggplot(data = arr_delay_by_month) +
  geom_line(aes(x = MONTH,
                y = mean_arr_delay,
                color = "Average arrival delay")) +
  geom_line(aes(x = MONTH,
                y = pct_arr_ontime/0.08,
                color = "Percentage of arrival on-time")) +
  scale_x_continuous(breaks = seq(1,12,1),
                     limits = c(0,13)) +
  scale_y_continuous(breaks = seq(-1,14,2),
                     limits = c(-1,15),
                     sec.axis = sec_axis(~.*8,
                                         name = "Percentage of on-time FLights",
                                         breaks = seq(0,100,20),
                                         labels = function(x)paste0(x, "%"))) +
  scale_color_manual(values = c("Average arrival delay" = "#1B9E77",
                                "Percentage of arrival on-time" = "#D95F02"),
                     name = "Arrival Delay Index") +
  labs(title = "Monthly Flights Arrival Delays in Chicago 2016",
       caption = "BTS Data.2016",
```

```
       x = "Month",
       y = "Average arrival delay (min)") +
  theme(plot.title = element_text(size = 14, face = "bold"),
        plot.caption = element_text(face = "italic")) +
  theme_light()
```



November has the lowest average arrival delays at -0.9593. However, there is no month when at least 80% flights arrive on time. Consistent with the lowest average arrival delays, November has the highest percentage of flights arrived on time with 71.6%.

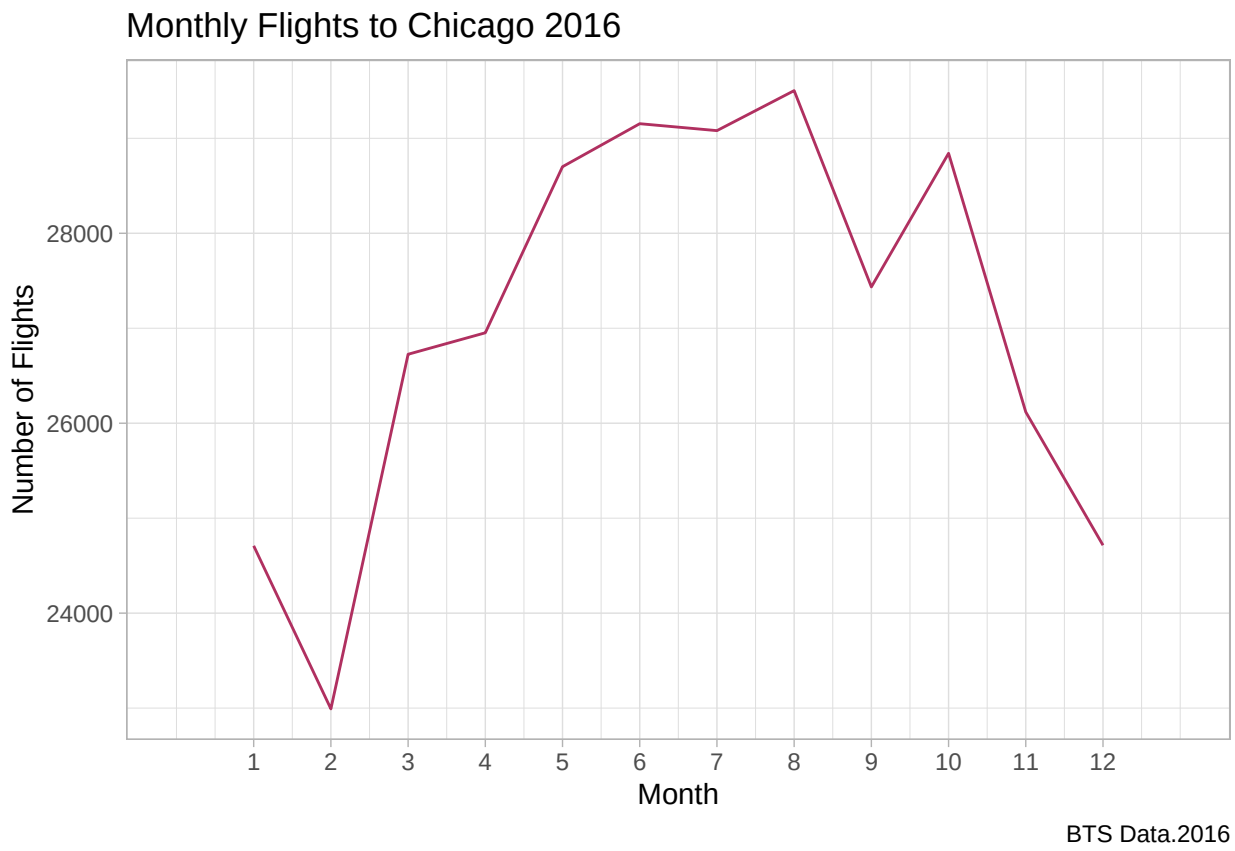
3.1.2

```
# Query
# Clean data
flights_tochi <- chicago_flights_to %>%
  drop_na(ARR_DELAY) %>%
  group_by(MONTH) %>%
  summarise(n_flights = n())%>%
  arrange(desc(n_flights))
flights_tochi
```

```
## # A tibble: 12 x 2
##    MONTH n_flights
```

```
##    <dbl>     <int>
##  1     8     29502
##  2     6     29154
##  3     7     29081
##  4    10     28842
##  5     5     28701
##  6     9     27435
##  7     4     26952
##  8     3     26726
##  9    11     26118
## 10    12     24715
## 11     1     24709
## 12     2     22992
```

```
# Plot
ggplot(flights_tochi) +
  geom_line(aes(x = MONTH,
                y = n_flights),
            color = "maroon") +
  scale_x_continuous(breaks = seq(1,12,1),
                     limits = c(0,13))  +
  labs(title = "Monthly Flights to Chicago 2016",
       caption = "BTS Data.2016",
       x = "Month",
       y = "Number of Flights") +
  theme(plot.title = element_text(size = 14, face = "bold"),
        plot.caption = element_text(face = "italic")) +
  theme_light()
```



The most common flights to Chicago occur in summer (August, June and July), of which August has the highest amount of flights at 29502.

3.1.3

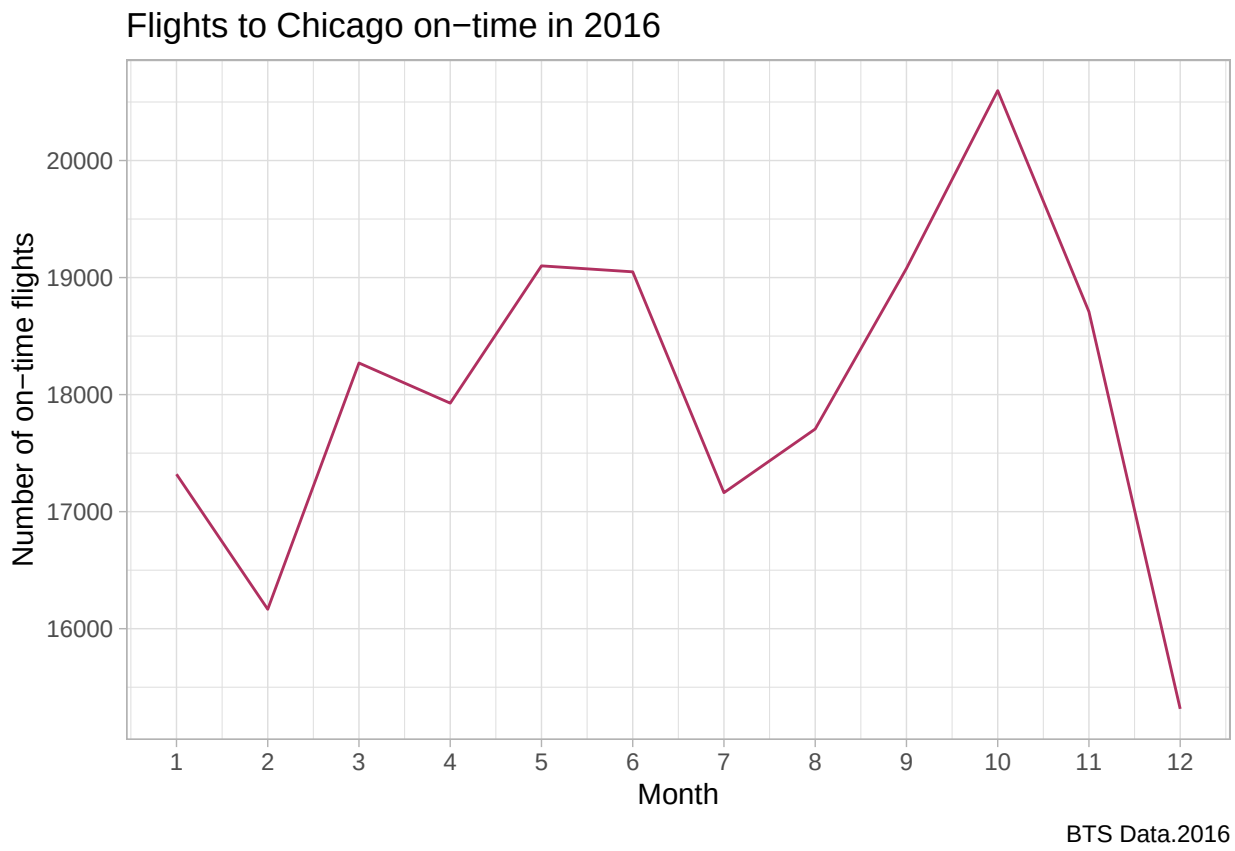
The convention should be held in October since October has the 3rd lowest arrival delays and also has high number of flights(ranks 4th). To be more specific, the average arrival delays is negative, which means flights in October arrives early on average, and the percentage of flights arrival on time is also the 2nd highest. Although it's not the month that has the highest number of flights, the differences is relatively small it won't cause significant issues. October would be convenient for attendees to find appropriate flights and also arrive on time compared to all the other months.

3.1.3a

```
# Merge data
arr_delay_nflights <- merge(arr_delay_by_month,
                            flights_tochi) %>%
  mutate(on_time_flights = pct_arr_ontime * n_flights) %>%
  arrange(desc(on_time_flights)) %>%
  select(MONTH,
         on_time_flights,
         everything())
arr_delay_nflights
```

```
##    MONTH on_time_flights mean_arr_delay pct_arr_ontime n_flights
## 1     10           20597     -0.8173497      0.7141322     28842
## 2      5           19100      3.5966691      0.6654820     28701
## 3      9           19078      1.9874977      0.6953891     27435
## 4      6           19049      5.0885299      0.6533923     29154
## 5     11           18708     -0.9593001      0.7162876     26118
## 6      3           18270      2.0115992      0.6836040     26726
## 7      4           17927      1.4043485      0.6651454     26952
## 8      8           17706     10.1189750      0.6001627     29502
## 9      1           17320     -0.9235501      0.7009592     24709
## 10     7           17162     13.9577731      0.5901448     29081
## 11     2           16166      2.0755915      0.7031141     22992
## 12    12           15315      7.5527008      0.6196642     24715
```

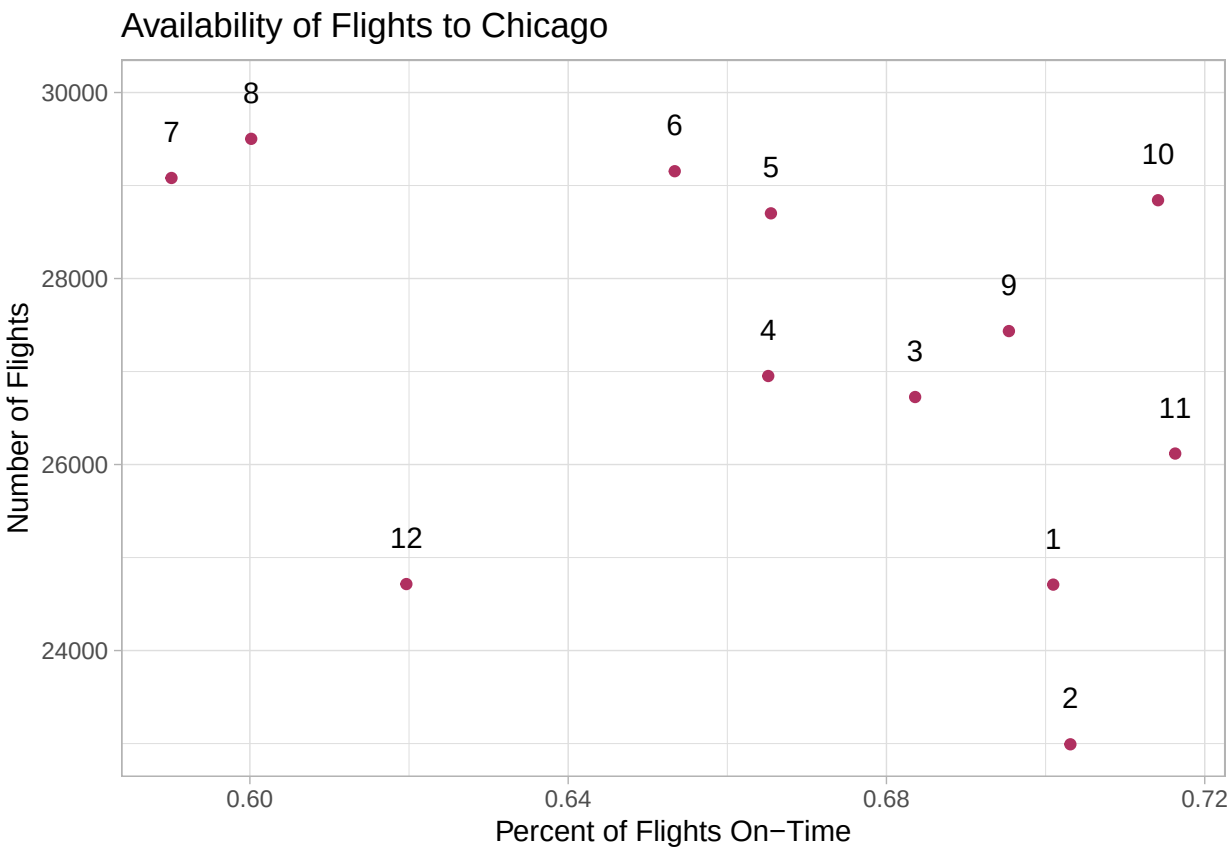
```
# Basic Plot
ggplot(arr_delay_nflights) +
  geom_line(aes(x = MONTH,
                y = on_time_flights),
                color = "maroon") +
  scale_x_continuous("Month", breaks = 1:12) +
  scale_color_manual(name = "Flight Index",
                     labels = c("Number of fligths on-time")) +
  labs(title = "Flights to Chicago on-time in 2016",
       caption = "BTS Data.2016",
       y = "Number of on-time flights") +
  theme(plot.title = element_text(size = 14, face = "bold"),
        plot.caption = element_text(face = "italic")) +
  theme_light()
```



In the plot above, we observe that October has the highest number of flights arrived on-time in 2016, which supports our conclusion made earlier that October is the best choice taking both availability and punctuality into account.Number of on-time flights combined nubmer of flights and percentage of on-time flights together into consideration. Mean arrival delay time is not substantially significant here since it only ranges from -0.96mins to 13.95 mins, therefore we don't include here.

3.1.3b

```
# Sophisticated Plot
arr_delay_nflights  %>%
  mutate(month = as.character(MONTH)) %>%
ggplot(aes(x = pct_arr_ontime,
           y = n_flights)) +
  geom_point(color = "maroon") +
  labs(y = "Number of Flights",
       x = "Percent of Flights On-Time",
       title = "Availability of Flights to Chicago") +
  geom_text(aes(label = month),
            nudge_y = 500,
            label.size = 0.2) +
  theme_light()
```




```
# used this site for code to add the month labels:
# https://stackoverflow.com/questions/15624656/label-points-in-geom-point
```

This sophisticated plot shows that there are some flights that have more overall flights than October. It also shows that October is best for its combination of the two relevant factors. The closer the point is to the upper-right corner, the better it performs in both dimension.
In the basic plot, we "waste" an axis on the month, when we could be using it to compare two dimensions of the problem for each month. We couldn't observe the detailed number of flights and percentage of on-time flights with the basic plot. However, I think the basic plot is more intuitive since we are accustomed to seeing time move from left to right on the horizontal axis. we would rather show the basic plot to the Mayor because it is easier to understand at a glance, which might be all that she has time to give it. It is also more clear that October is a better month than the rest in that graph.

Notes
We don't include the mean arrival delay time since the differences is relatively small.

3.1.3c

We could also analyze the data on taxi trips in Chicago, which can be found here: https://data.cityofchicago.org/Transportation/Taxi-Trips/wrvz-psew. We would look at the number of trips by month. I can tell when a trip occured by looking at the 'Trip Start Timestamp' variable.

Another valuable dataset could be the "calendar" data from AirBnB for Chicago. This could tell us about the pricing of accomodations (with the 'price' and 'adjusted price' variables) for people attending the conference by month (the 'month' variable). We learned about the data here: https://medium.com/@dheknemrunal12/data-analysis-of-airbnb-hotels-at-chicago-84e748f1e054 and it can be downloaded here: http://insideairbnb.com/get-the-data/.

3.2

```
# # Summarise data by carrier for October flights
Oct <- chicago_flights_to %>%
  filter(MONTH == 10) %>%
  drop_na(ARR_DELAY) %>%
  mutate(arr_ontime = ifelse(ARR_DELAY <= 0,
                             1,
                             0)) %>%
  group_by(OP_UNIQUE_CARRIER) %>%
  summarise(mean_arr_delay = mean(ARR_DELAY), # lowest average arrivals delays
            pct_arr_ontime = mean(arr_ontime), # 80% of flights on time
            n_flights = n() # number of flights
            ) %>%
  arrange(mean_arr_delay,
          desc(pct_arr_ontime),
          desc(n_flights))
Oct
```

```
## # A tibble: 11 x 4
##    OP_UNIQUE_CARRIER mean_arr_delay pct_arr_ontime n_flights
##    <chr>                      <dbl>          <dbl>     <int>
##  1 AS                        -8.93           0.783       180
##  2 DL                        -5.93           0.808       889
##  3 WN                        -5.47           0.778      7199
##  4 NK                        -4.59           0.765       826
##  5 F9                        -0.974          0.745       385
##  6 AA                        -0.141          0.676      5218
##  7 B6                         0.338          0.648       210
##  8 UA                         0.650          0.691      6963
##  9 OO                         2.36           0.682      3455
## 10 EV                         4.05           0.680      3373
## 11 VX                         7.98           0.590       144
```

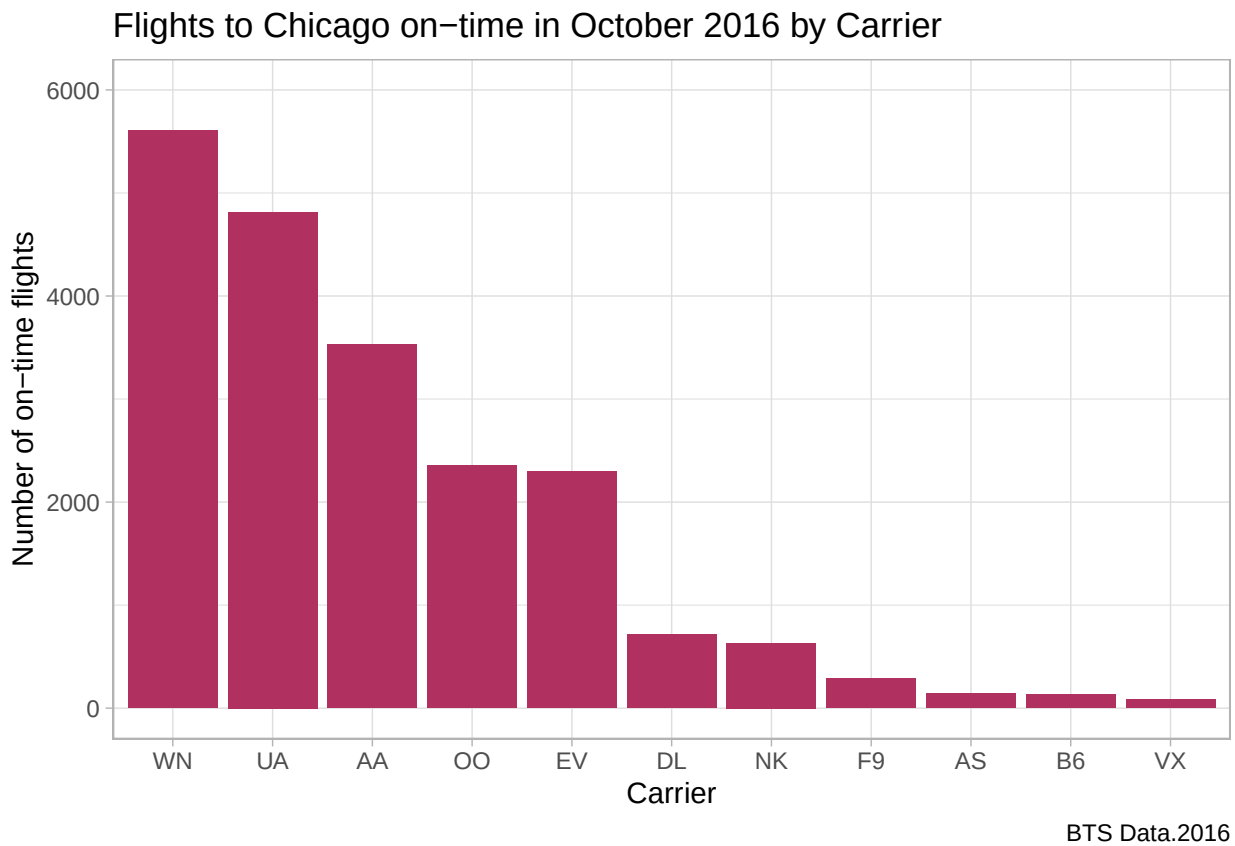
Southwest airline would be the best choice since it has the best overall performance in the average arrival delay time, percentage of flights arrived on-time and the number of flights. Although AS and DL has better performance in flights arriving on time, the number of flights is substantially less than that of Southwest.

```
# Mutate number of on-time flights
Oct <- Oct %>%
  mutate(on_time_flights = pct_arr_ontime * n_flights) %>%
  arrange(desc(on_time_flights)) %>%
  select(OP_UNIQUE_CARRIER,
         on_time_flights,
         everything())
Oct
```

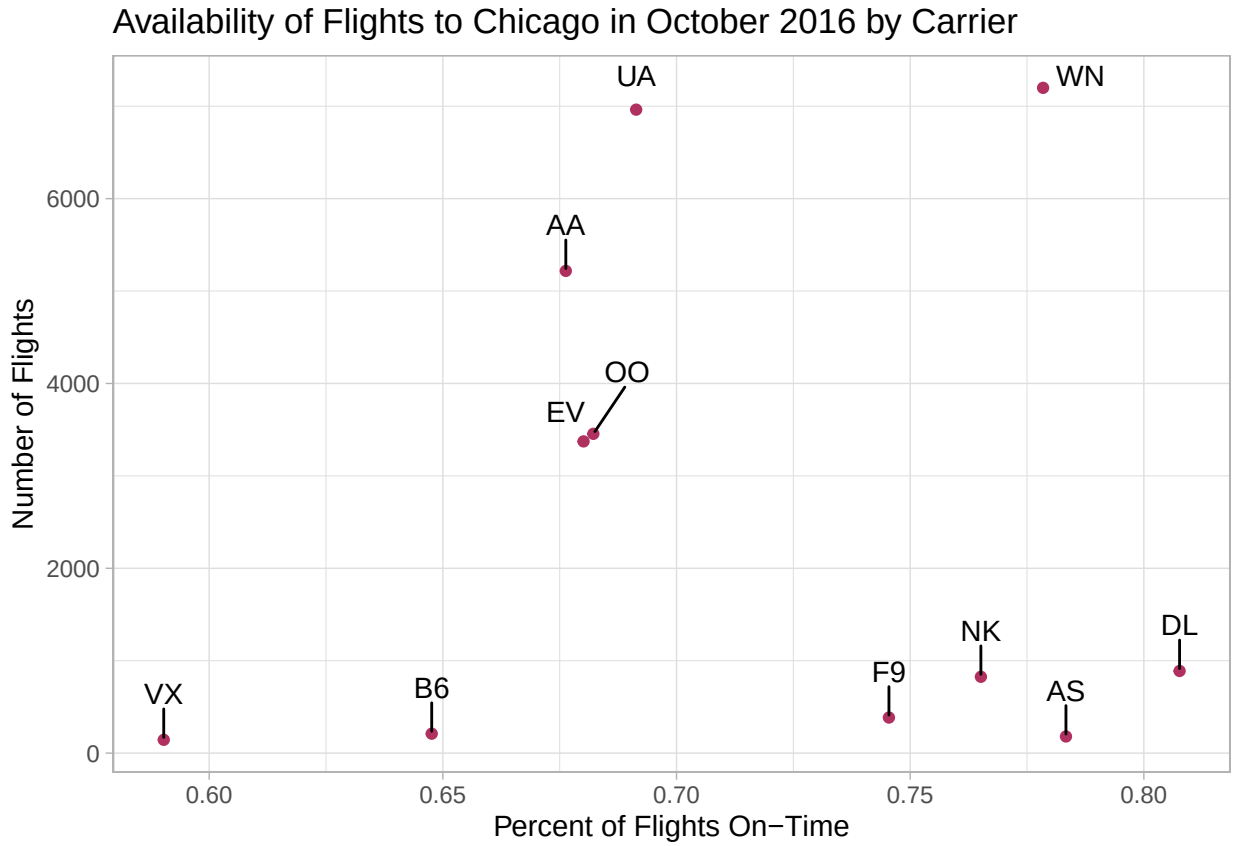
```
## # A tibble: 11 x 5
##    OP_UNIQUE_CARRIER on_time_flights mean_arr_delay pct_arr_ontime n_flights
##    <chr>                       <dbl>          <dbl>          <dbl>     <int>
##  1 WN                           5604          -5.47          0.778      7199
##  2 UA                           4814           0.650         0.691      6963
```

```
##  3 AA                          3529          -0.141          0.676       5218
##  4 OO                          2357           2.36           0.682       3455
##  5 EV                          2294           4.05           0.680       3373
##  6 DL                           718          -5.93           0.808        889
##  7 NK                           632          -4.59           0.765        826
##  8 F9                           287          -0.974          0.745        385
##  9 AS                           141          -8.93           0.783        180
## 10 B6                           136           0.338          0.648        210
## 11 VX                            85           7.98           0.590        144
```

```
# Basic plot
ggplot(Oct) +
  geom_col(aes(x = reorder(OP_UNIQUE_CARRIER,-on_time_flights),
               y = on_time_flights),
           fill = "maroon") +
  scale_y_continuous(limits = c(0,6000)) +
  labs(title = "Flights to Chicago on-time in October 2016 by Carrier",
       caption = "BTS Data.2016",
       x = "Carrier",
       y = "Number of on-time flights") +
  theme(plot.title = element_text(size = 14, face = "bold"),
        plot.caption = element_text(face = "italic")) +
  theme_light()
```



```r
# Sophisticated Plot
Oct %>%
ggplot(aes(x = pct_arr_ontime,
           y = n_flights)) +
  geom_point(color = "maroon") +
  labs(y = "Number of Flights",
       x = "Percent of Flights On-Time",
       title = "Availability of Flights to Chicago in October 2016 by Carrier") +
  geom_text_repel(aes(label = OP_UNIQUE_CARRIER),
            nudge_y = 500,
            label.size = 0.2) +
  theme_light()
```



The basic plot shows that Southwest Airline has the highest number of on-time flights in October 2016, which supports our conclusion made earlier. The sophisticated plot also supports our conclusion since Southwest Airline is the closest to the upper-right corner, which means it has the highest percent of flights on-time compared to all the other carriers which has high number of flights.

The sophisticated plot uncovers information about the number of flights available and also the punctuality, while we only observe one dimension. It could lead to wrong decision since the high number of on-time flights could result from a combination of high volume of flights with relatively low punctuality rate.

We would recommend the sophisticated plot because additional information is necessary to draw the conclusion which Airline to recommend. It won't matter much when we group by month because it's unlikely the number of flights vary substantially between months(which is verified in our dataset). However, the number of flights could vary a lot among different carriers. The plot also confirms that. UA has the 2nd highest number of flights which leads it to be the 2nd preferable Airline in the basic plot, while we observe in the sophisticated plot that it's punctuality rate is not good; Also, DL will be considered much worse than AA

with the basic plot, while it has the highest percentage of flights arriving on-time. The reason is that DL has substantially less number of flights compared to AA. Therefore, we need both dimensions in the plot to make the correct decision.

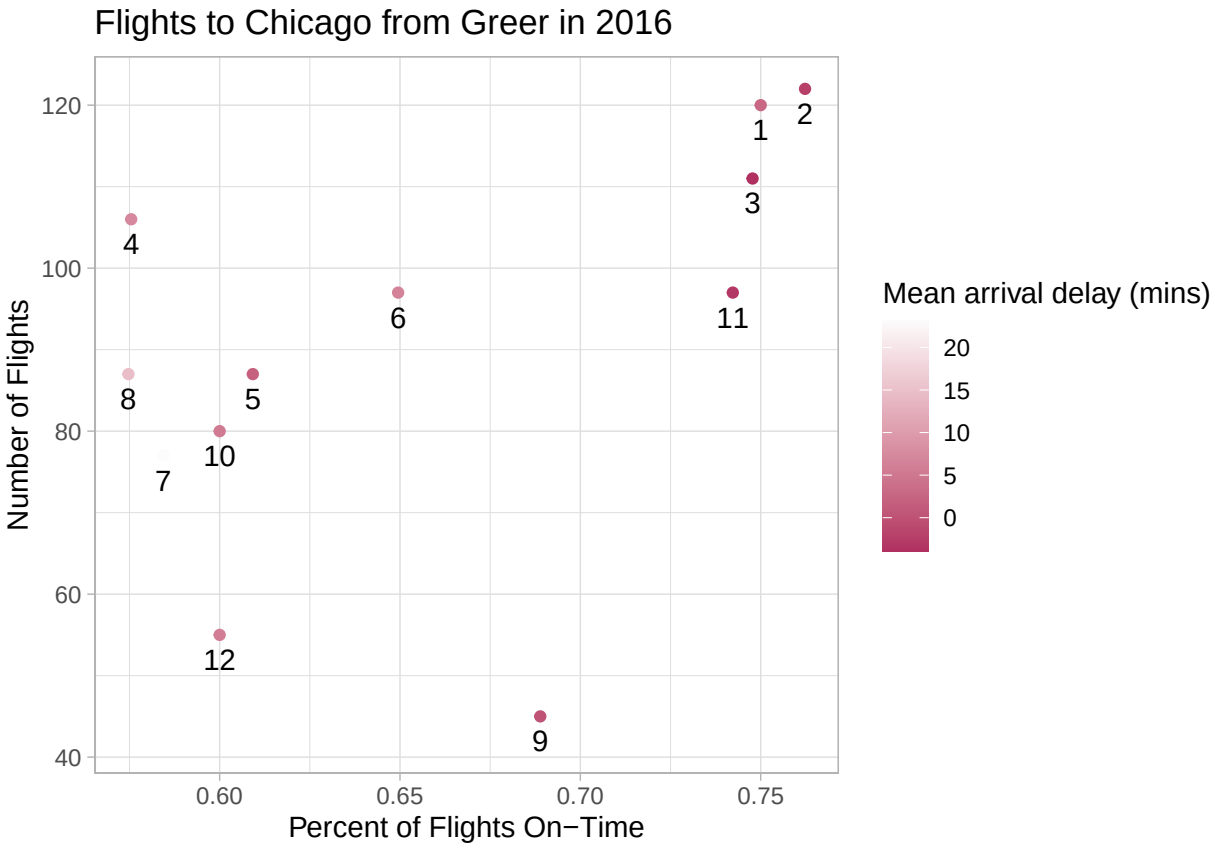
3.3

```
# Subset dataset for ORGIN Greer
greer <- chicago_flights_to %>%
  filter(ORIGIN_CITY_NAME == "Greer, SC")
# Clean data
greer_an <-greer %>%
  drop_na(ARR_DELAY) %>%
  mutate(arr_ontime = ifelse(ARR_DELAY <= 0,1,0)) %>%
  group_by(MONTH) %>%
  summarise(mean_arr_delay = mean(ARR_DELAY), # lowest average arrivals delays
            pct_arr_ontime = mean(arr_ontime), # 80% of flights on time
            n_flights = n() # number of flights
            )%>%
  arrange(mean_arr_delay,
          desc(pct_arr_ontime),
          desc(n_flights))
greer_an
```

```
## # A tibble: 12 x 4
##    MONTH mean_arr_delay pct_arr_ontime n_flights
##    <dbl>          <dbl>          <dbl>     <int>
##  1     3         -3.88           0.748       111
##  2    11         -3.15           0.742        97
##  3     2         -2.11           0.762       122
##  4     9          0.133          0.689        45
##  5     5          1.97           0.609        87
##  6     1          2.87           0.75        120
##  7    10          5.39           0.6          80
##  8    12          6.04           0.6          55
##  9     6          6.40           0.649        97
## 10     4          7.40           0.575       106
## 11     8         14.6            0.575        87
## 12     7         23.1            0.584        77
```

```
# Plot
greer_an %>%
  mutate(month = as.character(MONTH)) %>%
ggplot() +
  geom_point(aes(x = pct_arr_ontime,
           y = n_flights, color = mean_arr_delay)) +
  scale_color_continuous(low ="maroon",
                         high = "grey99",
                       name = "Mean arrival delay (mins)") +
  labs(y = "Number of Flights",
       x = "Percent of Flights On-Time",
       title = "Flights to Chicago from Greer in 2016") +
  geom_text(aes(x = pct_arr_ontime,
```

```
          y = n_flights, label = month),
           nudge_y = -3) +
theme_light()
```



We need to change the decision. If we limit the dataset to flights from Greer since it's the origin where most important members from, February would be better choices compared to October. Flights from Greer in October has relatively limited availability, together with lower on-time percentage and longer delay time compared to several other months. In February, flights are punctuate and highly available, therefore, it'd be a good choice to hold the convention in February. Although January is very close to February in the plot, the mean arrival delay time is obviously greater in January than February. Mean arrival delay time is necessary in this plot since it'd be hard to distinguish January and February without the dimension.
The decision would be superior to the previous one because our major attendees will from Greer, then it's not reasonable to take all the other origins into account and give them the same weight(calculate the overall average and distribution). Flight situations may differ greatly between cities. Therefore, we need to make sure our decision is the best for most attendees.