# Week Three - AE04

Brandon Leslie

## AE-04: NYC Flights + Data Wrangling

### Exercise 1

The *flights* data frame has 336,776 rows, and each row represents a flight.

### Exercise 2

The names of the variables in *flights* are...

```
 [1] "year"           "month"          "day"            "dep_time"      
 [5] "sched_dep_time" "dep_delay"      "arr_time"       "sched_arr_time"
 [9] "arr_delay"      "carrier"        "flight"         "tailnum"       
[13] "origin"         "dest"           "air_time"       "distance"      
[17] "hour"           "minute"         "time_hour"     
```

### Exercise 3

#### 3A

Here is a dataframe with the variables *dep_delay* & *arr_delay*.

```
# A tibble: 336,776 x 2
   arr_delay dep_delay
       <dbl>     <dbl>
 1        11         2
 2        20         4
 3        33         2
 4       -18        -1
 5       -25       -6
 6        12       -4
 7        19       -5
 8       -14       -3
 9        -8       -3
10         8       -2
# i 336,766 more rows
```

### 3B

Here is a data frame that keeps every variable excepot *dep_delay*

```
# A tibble: 336,776 x 18
    year month   day dep_time sched_dep_time arr_time sched_arr_time arr_delay
   <int> <int> <int>    <int>          <int>    <int>          <int>     <dbl>
 1  2013     1     1      517            515      830            819        11
 2  2013     1     1      533            529      850            830        20
 3  2013     1     1      542            540      923            850        33
 4  2013     1     1      544            545     1004           1022       -18
 5  2013     1     1      554            600      812            837       -25
 6  2013     1     1      554            558      740            728        12
 7  2013     1     1      555            600      913            854        19
 8  2013     1     1      557            600      709            723       -14
 9  2013     1     1      557            600      838            846        -8
10  2013     1     1      558            600      753            745         8
# i 336,766 more rows
# i 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>
```

### 3C

Make a data frame that includes all the variables between *year* & *dep_delay* inclusively.

```
# A tibble: 336,776 x 6
    year month   day dep_time sched_dep_time dep_delay
   <int> <int> <int>    <int>          <int>     <dbl>
 1  2013     1     1      517            515         2
 2  2013     1     1      533            529         4
 3  2013     1     1      542            540         2
 4  2013     1     1      544            545        -1
 5  2013     1     1      554            600        -6
 6  2013     1     1      554            558        -4
 7  2013     1     1      555            600        -5
 8  2013     1     1      557            600        -3
 9  2013     1     1      557            600        -3
10  2013     1     1      558            600        -2
# i 336,766 more rows
```

### 3D

Using the *select* & *contains()* to make a data frame that includes variables associated with the arrival.

```
# A tibble: 336,776 x 3
   arr_time sched_arr_time arr_delay
      <int>          <int>     <dbl>
 1      830            819        11
 2      850            830        20
 3      923            850        33
 4     1004           1022       -18
 5      812            837       -25
 6      740            728        12
 7      913            854        19
 8      709            723       -14
 9      838            846        -8
10      753            745         8
# i 336,766 more rows
```

## Exercise 4

### 4A

Display the first five moves of the *flights* data frame.

```
# A tibble: 5 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      544            545        -1     1004           1022
```

```
5  2013     1     1      554            600        -6      812            837
# 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>
```

### 4B

Display the last two rows of the *flights* data frame.

```
# A tibble: 2 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     9    30       NA            840        NA       NA           1020
2  2013     9    30       NA           1159        NA       NA           1344
# 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>
```

## Exercise 5

### 5A

Flights with the shortest dep_delay at the top

```
# A tibble: 336,776 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    12     7     2040           2123       -43       40           2352
 2  2013     2     3     2022           2055       -33     2240           2338
 3  2013    11    10     1408           1440       -32     1549           1559
 4  2013     1    11     1900           1930       -30     2233           2243
 5  2013     1    29     1703           1730       -27     1947           1957
 6  2013     8     9      729            755       -26     1002            955
 7  2013    10    23     1907           1932       -25     2143           2143
 8  2013     3    30     2030           2055       -25     2213           2250
 9  2013     3     2     1431           1455       -24     1601           1631
10  2013     5     5      934            958       -24     1225           1309
# i 336,766 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>
```

When *dep_delay* has a negative value, that means the flight(s) have arrived earlier then expected.

### 5B

Flights with the longest *dep_delay* at the top.

```
# A tibble: 336,776 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     9      641            900      1301     1242           1530
 2  2013     6    15     1432           1935      1137     1607           2120
 3  2013     1    10     1121           1635      1126     1239           1810
 4  2013     9    20     1139           1845      1014     1457           2210
 5  2013     7    22      845           1600      1005     1044           1815
 6  2013     4    10     1100           1900       960     1342           2211
 7  2013     3    17     2321            810       911      135           1020
 8  2013     6    27      959           1900       899     1236           2226
 9  2013     7    22     2257            759       898      121           1026
10  2013    12     5      756           1700       896     1058           2020
# i 336,766 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>
```

### 5C

Data frame that only lncludes the variables *tailnum*, *carrier*, and *departure delay* for the flight with the longest deature delays at the top.

```
# A tibble: 1 x 3
  carrier tailnum dep_delay
  <chr>   <chr>       <dbl>
1 B6      N592JB        -43
```

## Exercise 6

### 6A

Data frame where all rows have the destination RDU.

```
# A tibble: 8,163 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      800            810       -10      949            955
 2  2013     1     1      832            840        -8     1006           1030
 3  2013     1     1      851            851         0     1032           1036
 4  2013     1     1      917            920        -3     1052           1108
 5  2013     1     1     1024           1030        -6     1204           1215
 6  2013     1     1     1127           1129        -2     1303           1309
 7  2013     1     1     1157           1205        -8     1342           1345
 8  2013     1     1     1240           1235         5     1415           1415
 9  2013     1     1     1317           1325        -8     1454           1505
10  2013     1     1     1449           1450        -1     1651           1640
# i 8,153 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>
```

### 6B

All rows where destination is RDU and arrival time is less than 0.

```
# A tibble: 4,232 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      800            810       -10      949            955
 2  2013     1     1      832            840        -8     1006           1030
 3  2013     1     1      851            851         0     1032           1036
 4  2013     1     1      917            920        -3     1052           1108
 5  2013     1     1     1024           1030        -6     1204           1215
 6  2013     1     1     1127           1129        -2     1303           1309
 7  2013     1     1     1157           1205        -8     1342           1345
 8  2013     1     1     1317           1325        -8     1454           1505
 9  2013     1     1     1505           1510        -5     1654           1655
10  2013     1     1     1800           1800         0     1945           1951
# i 4,222 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>
```

The code I have written to produce this data frame are using logical operators to filter out rows that are deemed as “false.” I’m telling the filter() function to use the data from the *nycflights13* package with the pipe ( %>% ) operator. Then from that data, I'm setting logical conditions with the equal to and less then operator so that the output will only display rows without a FALSE Boolean value.

## Exercise 7

### 7A

Frequency table of the destination locations for flights from New York.

```
# A tibble: 105 x 2
   dest      n
   <chr> <int>
 1 ABQ     254
 2 ACK     265
 3 ALB     439
 4 ANC       8
 5 ATL   17215
 6 AUS    2439
 7 AVL     275
 8 BDL     443
 9 BGR     375
10 BHM     297
# i 95 more rows
```

### 7B

```
# A tibble: 1 x 2
  month     n
  <int> <int>
1     2 24951
```

There was *24951* flights that month.

### 7C

```
# A tibble: 1 x 3
  month   day     n
  <int> <int> <int>
1    11    27  1014
```

There was *1014* flights that day.

## Exercise 8

### 8A

*air_time* from minutes -> hours, and creation of new variable> (*mph*)

```
# A tibble: 336,776 x 4
   air_time distance hours   mph
      <dbl>    <dbl> <dbl> <dbl>
 1      227     1400 3.78   370.
 2      227     1416 3.78   374.
 3      160     1089 2.67   408.
 4      183     1576 3.05   517.
 5      116      762 1.93   394.
 6      150      719 2.5    288.
 7      158     1065 2.63   404.
 8       53      229 0.883  259.
 9      140      944 2.33   405.
10      138      733 2.3    319.
# i 336,766 more rows
```

### 8B

```
# A tibble: 12 x 3
   month     n  prop
   <int> <int> <dbl>
 1     1 27004  8.02
 2     2 24951  7.41
 3     3 28834  8.56
 4     4 28330  8.41
 5     5 28796  8.55
 6     6 28243  8.39
 7     7 29425  8.74
 8     8 29327  8.71
 9     9 27574  8.19
10    10 28889  8.58
11    11 27268  8.10
12    12 28135  8.35
```

The proportion of flights that take place is around 8.7 ### 8C Creation of new variable, *rdu_bound*, and calculation of what proportion of flights are from RDU.

```
# A tibble: 3 x 4
# Groups:   origin [3]
  origin rdu_bound     n  prop
  <chr>  <chr>     <int> <dbl>
1 EWR    Yes        1482  1.23
2 JFK    Yes        3100  2.79
3 LGA    Yes        3581  3.42
```

## Exercise 9

### 9A

Mean arrival delay for all flights.

```
# A tibble: 1 x 1
  mean_arr_delay
           <dbl>
1           12.6
```

## Exercise 10

### 10A

Mean arrival delay for each month.

```
# A tibble: 12 x 2
   month mean_arr_delay
   <int>          <dbl>
 1     1           6.13
 2     2           5.61
 3     3           5.81
 4     4          11.2
 5     5           3.52
 6     6          16.5
 7     7          16.7
 8     8           6.04
 9     9          -4.02
```

```
10    10          -0.167
11    11           0.461
12    12          14.9
```

### 10B

The airport with the shortest median departure delay is **LGA**.

```
# A tibble: 3 x 2
  origin med_dep_delay
  <chr>          <dbl>
1 EWR               -1
2 JFK               -1
3 LGA               -3
```