Lab2Q1

Installing packages

```r
library(nycflights13)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Loading data

```r
data("flights")
head(flights)
```

```
## # A tibble: 6 × 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
## # ... 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>
```

Function 1 - Select

```r
temp1 = select(flights, year:day, dep_delay, arr_delay)
dim(temp1)
```

```
## [1] 336776      5
```

```r
head(temp1)
```

```
## # A tibble: 6 × 5
##    year month   day dep_delay arr_delay
##   <int> <int> <int>     <dbl>     <dbl>
## 1  2013     1     1         2        11
## 2  2013     1     1         4        20
## 3  2013     1     1         2        33
## 4  2013     1     1        -1       -18
## 5  2013     1     1        -6       -25
## 6  2013     1     1        -4        12
```

Function 2 - Filter

```
temp2 = filter(flights, dep_delay > 240)
dim(temp2)
```

```
## [1] 1524   19
```

```
head(temp2)
```

```
## # A tibble: 6 × 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>
## 1  2013     1     1      848           1835       853     1001
## 2  2013     1     1     1815           1325       290     2120
## 3  2013     1     1     1842           1422       260     1958
## 4  2013     1     1     2115           1700       255     2330
## 5  2013     1     1     2205           1720       285       46
## 6  2013     1     1     2343           1724       379      314
## # ... 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>
```

Function 3 - Arrange

```
temp3 = arrange(flights, year, month, day)
dim(temp3)
```

```
## [1] 336776     19
```

```
head(temp3)
```

```
## # A tibble: 6 × 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
## # ... 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>
```

This arranges the dataset in increasing order of Year,Month and Day

Function 4 - Mutate

```r
temp4 = mutate(flights, speed = air_time / distance)
dim(temp4)
```

```
## [1] 336776     20
```

```r
head(temp4)
```

```
## # A tibble: 6 × 20
##    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
## # ... with 13 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>, speed <dbl>
```

```r
head(temp4$speed)
```

```
## [1] 0.1621429 0.1603107 0.1469238 0.1161168 0.1522310 0.2086231
```

This function creates a new coloumn based on some give formula

Function 5 - Summarise

```r
temp5 = summarise(flights, delay = mean(dep_time,na.rm = T))
temp5
```

```
## # A tibble: 1 × 1
##     delay
##     <dbl>
## 1 1349.11
```

Function 6 - Group_By

group_by function is usually used with combination of other functions to first split the data according to some factor and then apply operations on each split.One example can be found below.

```r
by_tailnum =  group_by(flights, tailnum)
delay = summarise(by_tailnum,
    count = n(),
    dist = mean(distance),
    delay = mean(arr_delay)
    )
delay = filter(delay, count > 20, dist < 2000)
```

The advantage of dplyr is that the expressions in select(), filter(), arrange(), mutate(), and summarise() are translated into SQL so they can be run on the database.