dplyr Part 2

install package "nycflights13"

```
install.packages("nycflights13")
```

load packages

```
library(tidyverse)
```
remember dplyr is part of the tidyverse pkg

```
library(nycflights13)
```

pull the documenation for flights, type:

```
?flights
```
you will see some details on the right bottom pane under help

```
head(flights)
```

```
tail(flights)
```

if you want to read it and see it on the upper right enviroment pane, you could do this:

```
flights <- flights
```

Recap the major functions:

filter: Subset observations based on their values.

summarize: Create summary measures of variables over the entire data frame.

group_by: Create summary over groups of observations on variables.

mutate: Create a new variable in the data frame by mutating existing ones.

arrange: Arrange/sort the rows based on one or more variables.

Give me only flights for january 1st

```
filter(flights, month==1, day==1)
```

You can save the results

```
Jan1 <- filter(flights, month==1, day==1)
```

Practice 1:

- Create a filtered data frame called **"Dec252013"** that only includes data from Dec 25, 2013.
- Create a data frame called **"NotJune"** that includes every month except June
- Create a data frame called **"Sept*or*Oct"** that includes flights for September or October
- Filter flights to only include flights operated by United, American or Delta

New functions:

between: Subset observations that happen between two values.

is.na: Identify null values.

desc: Sort values in descending order.

select: Choose only the variables/values you are interested in.

starts_with: Select all the variables that start with a certain value.

ends_with: Select all the variables that end with a certain value.

contains: Select all the variables that contain a certain value.

one_of: Select any variable which matches one of the stings in a vector.

rename: Change the name of a column in your data.

between()

Filter flights between July and September

```
filter (flights, month >= 7, month <=9)
```

or

```
filter (flights, between(month,7,9))
```

is.na()

How many flights have a missing dep_time?

```
filter (flights, is.na(dep_time))
```

desc()

Arrange flights by departure time.

`arrange(flights, dep_time)` default ascending order

`arrange(flights, desc(dep_time))` descending order

*If there are missing values they are always sorted at the end.

Practice2:

- Sort flight to find the ones that left the earliest?
- Sort flights to find the fastest flights?
- Sort flight to only have the top 3 fastest flights

  *When you arrange, missing values are sorted at the end of the data set.

  ### Tip!

How could you arrange your data to list NA values first?

```
arrange(flights, !is.na(dep_time))
```

select()

select columns by name, we just want to look at year, month, day

```
select (flights, year, month, day)
```

select all columns between year and day

```
select (flights, year:day)
```

select all columns except those between year and day

```
select (flights, -(year:day))
```

Helper functions that are useful to use with select:

starts_with()

select all columns/variables that starts with "dep" , "arr"

```
select(flights,starts_with("dep"), starts_with("arr"))
```

ends_with()

select all columns/variables that end with "delay"

```
select (flights, ends_with("delay"))
```

contains()

select all columns/variables that contain "delay"

```
select(flights, contains("delay"))
```

one_of()

so you have a list (vector) of variables called vars

```
vars <- c("year", "month", "day", "dep_delay", "arr_delay")
```

select any variable which matches one of the strings in the vector

```
select(flights, one_of(vars))
```

rename()

rename the "tailnum" column to something more readable, "tail_num"

```
rename(flights, tail_num = tailnum)
```
new name = old name

summarize()

You can use all kinds of functions to summarize the data,

There are many other summary statistics you could consider such as:

- **sd()** standard deviation
- **min()** minimum value
- **max()** maximum value
- **median()** median value
- **sum()** sum of values
- **n()** returns the length of vector
- **first()** returns first value in vector
- **last()** returns last value in vector
- **n_distinct()** number of distinct values in vector

Determine the mean of dep_delay (remove missing values with na.rm)

```
summarize(flights, delay=mean(dep_delay, na.rm = TRUE))
```
gave the result a name "delay"

The result is not useful as it simply gave you the mean of all the delays.

group_by()

Used to get summary information aobut groups of data, instead of the complete dataset.

Group by year, month, day, store it in by_day

```
by_day <- group_by(flights, year, month, day)
summarize(by_day, delay=mean(dep_delay, na.rm = TRUE))
```

This now gives you the mean delay for each day.

Practice 3

Instead of creating new variables to store results, use the %>% operator to chain all the functions together

- Group flights by distination.
- Summarize to compute the distance, average delay, and number of flights
- Filter to return only distination that had over 20 delays and not Honolulu

```
group_by(dest) %>%
summarize (
  count=n(), dist=mean(distance, na.rm = TRUE),
  delay = mean(arr_delay, na.rm = TRUE)) %>%
filter(count >20, dest !="HNL")
```

Which carrier has the worst delays? Return just the top 5 offenders.