Stat 405: Coding Clinic 2

Ajjit and Alex

2019-02-13

Reviewing dataframe operations

- First some setup
- type `flights` into the console to view the dataframe
- familiarize yourself with the columns and their meanings

Reviewing dataframe operations

- filtering rows and selectng columns

```
flights[flights$month>=12, c("month", "day","carrier")]
```

- creating new columns

```
flights$air_time_hrs = flights$air_time/60
```

One more operation: Arranging/sorting

- Say you want to see the most delayed flights

```
#How to sort dataframes by 1 column
flights[order(flights$dep_delay),]
```

To make your lives easier, use dplyr

- First install it

```r
install.packages("dplyr")
library(dplyr)
```

To make your lives easier, use dplyr

- First install it

```r
install.packages("dplyr")
library(dplyr)
```

- Essentially a set of 'verbs' that allow you to solve vast majority of data manipulation problems
- All verbs work similarly:
    - The first argument is a data frame.
    - The subsequent arguments describe what to do with the data frame, using the variable names (without quotes).
    - The result is a new data frame.

dplyr basics

- Filter observations by their values (`filter()`).
- Select columns by their names (`select()`).
- Create new variables with functions of existing variables (`mutate()`)
- Reorder the rows (`arrange()`).

Filtering dataframes (again)

- Previously:

```
dec_flights = flights[flights$month == 12,]
```

Filtering dataframes (again)

- Previously:

```
dec_flights = flights[flights$month == 12,]
```

- With dplyr:

```
dec_flights = mutate(flights, month == 12)
```

Filtering dataframes (again)

- Previously:

```
dec_flights = flights[flights$month == 12,]
```

- With dplyr:

```
dec_flights = mutate(flights, month == 12)
```

- Or if you want to filter to flights on December 1st

Filtering dataframes (again)

- Previously:

```
dec_flights = flights[flights$month == 12,]
```

- With dplyr:

```
dec_flights = mutate(flights, month == 12)
```

- Or if you want to filter to flights on December 1st

```
dec_1st_flights = mutate(flights, month == 12, day == 1)
```

- Isn't that pretty!

Selecting columns (again)

- Previously:

```
flights_red = flights[,c("dep_time","arr_time")]
```

Selecting columns (again)

- Previously:

```
flights_red = flights[,c("dep_time","arr_time")]
```

- With dplyr:

```
flights_red = select(flights, dep_time, arr_time)
```

- Lets make it even prettier with the pipe (%>%)

```
flights_red = flights %>%
                select(dep_time, arr_time)
```

Creating new columns (again)

- Previously:

```
flights$air_time_hrs = flights$air_time/60
```

Creating new columns (again)

- Previously:

```
flights$air_time_hrs = flights$air_time/60
```

- With dplyr:

```
flights = flights %>%
            mutate(air_time_hrs = air_time/60)
```

Exercise:

Make a new dataframe called flights_changed with the following changes.

- Create a new column called cancelled that is TRUE if dep_time has an NA value, and FALSE otherwise
- Filter the dataframe to flights which started later than scheduled (ie departure delay is greater than 0)
- select the following columns: dep_time, arr_time, carrier, origin, dest, air_time, distance

Exercise:

Make a new dataframe called flights_changed with the following changes.

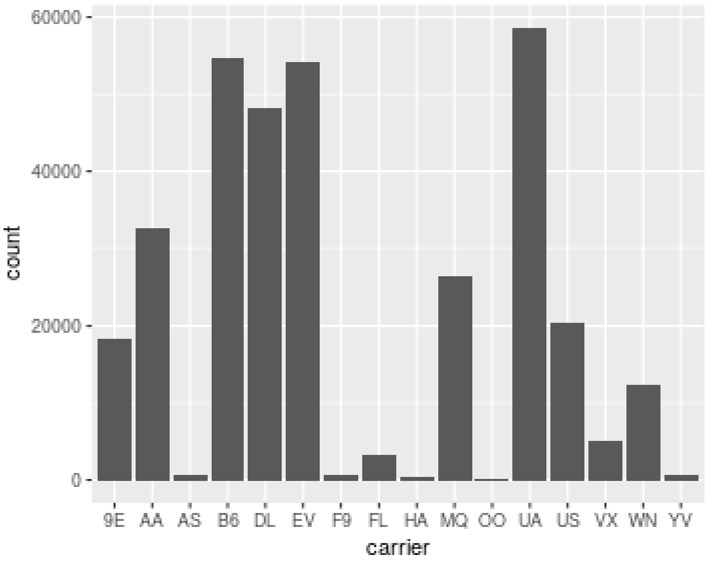
- Create a new column called cancelled that is TRUE if dep_time has an NA value, and FALSE otherwise
- Filter the dataframe to flights which started later than scheduled (ie departure delay is greater than 0)
- select the following columns: dep_time, arr_time, carrier, origin, dest, air_time, distance
- In this new dataset, order the dataframe by departure delay, with the most delayed flights appearing at the top.

Data viz: Intro to ggplot()

- “The simple graph has brought more information to the data analyst’s mind than any other device.” — John Tukey

- How to create a plot with ggplot:

```
ggplot(data = flights) +
     geom_bar(aes(x=carrier))
```



Data viz: Intro to ggplot()

- Generic formula for plots:

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

- Aesthetics (aes): visual property of the objects in your plot. Aesthetics include things like the size, the shape, or the color of your points

```
ggplot(data = flights) +
      geom_bar(aes(x=carrier, fill = origin))
```

Data viz: Exercise

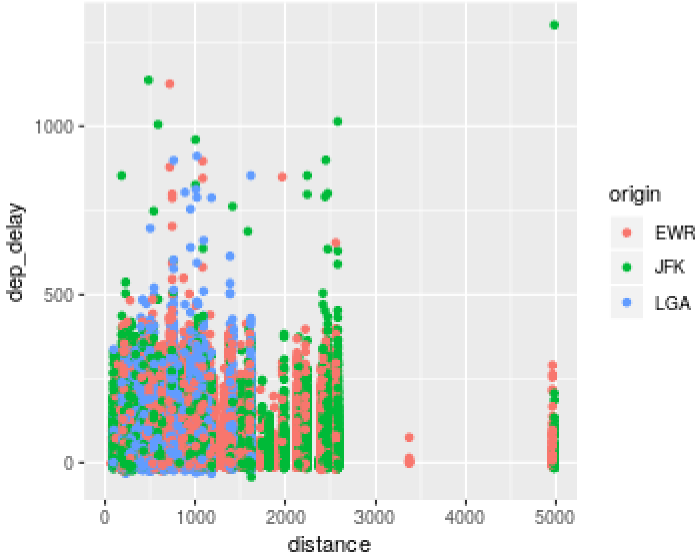
- Create a scatterplot of distance vs departure delay colored in by origin airport. (hint look at `?geom_point` and the `color` aesthetic)

Data viz: Exercise

- Create a scatterplot of distance vs departure delay colored in by origin airport. (hint look at `?geom_point` and the `color` aesthetic)

```
ggplot(data = flights) +
      geom_point(aes(x=distance, y=dep_delay, col = origin))
```

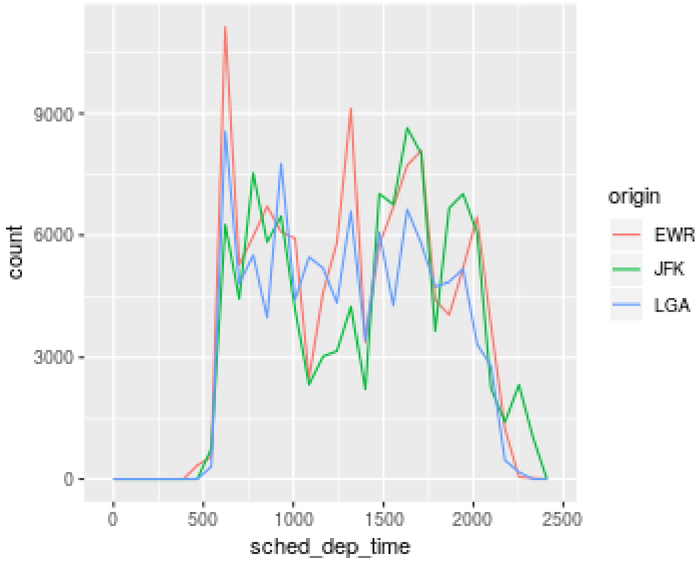
```
## Warning: Removed 8255 rows containing missing values (geom_point).
```



More cool ggplot examples

```
ggplot(data = flights) +
      geom_freqpoly(aes(x=sched_dep_time, col = origin))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


count
sched_dep_time
origin
EWR
JFK
LGA