Introduction to R -- Workshop

Some define statistics as the field that focuses on turning information into knowledge. The first step in that process is to summarise and describe the raw information - the data. In this lab we explore flights, specifically a random sample of domestic flights that departed from the three major New York City airport in 2013. We will generate simple graphical and numerical summaries of data on these flights and explore delay times. As this is a large data set, along the way you’ll also learn skills of data processing and factoring.

**Step 1:**

Load some packages:

* library(statsr)
* library(dplyr)
* library(ggplot2)

Now some data too ☺

The Bureau of Transportation Statistics (http://www.rita.dot.gov/bts/about/) (BTS) is a statistical agency that is a part of the Research and Innovative Technology Administration (RITA). As its name implies, BTS collects and makes available transportation data, such as the flights data we will be working with in this lab. We begin by loading the nycflights data frame. Type the following in your console to load the data:

* data(nycflights)

The data frame containing 32735 flights that shows up in your workspace is a data matrix, with each row representing an observation and each column representing a variable. **R** calls this data format a **data frame**, which is a term that will be used throughout the labs.

We can view the **names of the variables** and obtain a **summary** by the commands**:**

* names(nycflights)
* summary(nycflights)

A very useful function for taking a quick peek at your data frame, and viewing its dimensions and data types is **str** , which stands for structure.

* str(nycflights)

**Questions**

* We might want to find out how delayed flights headed to a particular destination tend to be.
* We might want to evaluate how departure delays vary over months.
* We might want to determine which of the three major NYC airports has a better on time percentage for departing flights.

**Why?**

When a flight is late then it costs the airline a fee for delaying other airplanes.

**Basis Data Manipulation**

Here are some basic data manipulation commands in R with the dplyr package. For each of these commands, have a play and type them into R. Firstly, you might want to type **?filter** to see what the functions does!

* filter()
* arrange()
* select()
* distinct()
* mutate()
* summarise()
* sample\_n()

**Step 2: Analysis of the data**

We can examine the distribution of departure delays of all flights with a histogram.

* ggplot(data = nycflights, aes(x = dep\_delay)) + geom\_histogram()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`. That is,

* ggplot(data = nycflights, aes(x = dep\_delay)) + geom\_histogram(binwidth=30)

This function says to plot the dep\_delay variable from the nycflights data frame on the x-axis. It also defines a **geom** (short for geometric object), which describes the type of plot you will produce.

Histograms are generally a very good way to see the shape of a single distribution, but that shape can change depending on how the data is split between the different bins. You can use the basic R function, **hist()** as well to plot histograms if you like. You can easily define the **binwidth** you want to use:

* ggplot(data = nycflights, aes(x = dep\_delay)) + geom\_histogram(binwidth = 15)

Try plotting a couple more histograms of the data with different binwdiths.

**Exercise: How do these histograms with the various binwidths compare?**

If we want to focus on departure delays of flights headed to RDU only, we need to first filter the data for

flights headed to LAX ( dest == "LAX" ) and then make a histogram of only departure delays of only those flights.

* lax\_flights <- nycflights %>% filter(dest == "LAX")
* ggplot(data = lax\_flights, aes(x = dep\_delay)) + geom\_histogram(binwidth=30)

Let’s break this code up:

Line 1: Take the nycflights data frame, filter for flights headed to RDU, and save the result as a new data frame called rdu\_flights .

* == means “if it’s equal to”.
* LAX is in quotation marks since it is a character string.

Line 2: Basically the same ggplot call from earlier for making a histogram, except that it uses the data

frame for flights headed to LAX instead of all flights.

Logical operators: As a recap, here are some of the logical operators in R:

* == means “equal to”
* != means “not equal to”
* or < means “greater than” or “less than”
* >= or <= means “greater than or equal to” or “less than or equal to”

**Numerical Summaries**

Type **summary** or **summarise** to obtain

* lax\_flights %>% summarise(mean\_dd = mean(dep\_delay), sd\_dd = sd(dep\_delay), n = n())

Note that in the summarise function we created a list of two elements. The names of these elements are user defined, like mean\_dd , sd\_dd , n , and you could customize these names as you like (just don’t use spaces in your names). Calculating these summary statistics also require that you know the function calls. Note that n() reports the sample size.

**Summary statistics:** Some useful function calls for summary statistics for a single numerical variable are as

follows:

mean

* median
* sd
* var
* IQR
* range
* min
* max

We can also filter based on multiple criteria. Suppose we are interested in flights headed to San Francisco

(SFO) in February:

* sfo\_feb\_flights <- nycflights %>% filter(dest == "SFO", month == 2)

Note that we can separate the conditions using commas if we want flights that are both headed to SFO and in February. If we are interested in either flights headed to SFO or in February we can use the | instead of the comma.

* dim(sfo\_feb\_flights)

**Question: Create a new data frame that includes flights headed to SFO in February, and save this data frame as sfo\_feb\_flights . How many flights meet these criteria?**

**Question: Make a histogram and calculate appropriate summary statistics for arrival delays of sfo\_feb\_flights.**

We can modify the above command using the group\_by function to get the same

summary stats for each origin airport:

* lax\_flights %>% group\_by(origin) %>% summarise(mean\_dd = mean(dep\_delay), sd\_dd = sd(dep\_delay), n = n())

Here, we first grouped the data by origin, and then calculated the summary statistics.

**Question: Calculate the median and interquartile range for arr\_delay s of flights in the sfo\_feb\_flights data frame, grouped by carrier. Which carrier is the has the highest IQR of arrival delays?**

* sfo\_feb\_flights %>% group\_by(carrier) %>% summarize(med\_ad = median(arr\_delay), iqr\_ad = IQR(arr\_delay)) %>% arrange(desc(iqr\_ad))

**Departure delays over months**

Which month would you expect to have the highest average delay departing from an NYC airport?

Let’s think about how we would answer this question:

First, calculate monthly averages for departure delays. With the new language we are learning, we need to group\_by months, then summarise mean departure delays. Then, we need to arrange these average delays in descending order

nycflights %>% group\_by(month) %>% summarise(mean\_dd = mean(dep\_delay)) %>% arrange(desc(mean\_dd))

**Question: Which month has the highest average departure delay from an NYC airport?**

Hint:

* nycflights %>% group\_by(month) %>% summarise(mean\_dd = mean(dep\_delay)) %>% arrange(desc(mean\_dd))

**Question: Which month has the highest median departure delay from Question 5 an NYC airport?**

**Question: Is the mean or the median a more reliable measure for deciding which month(s) to avoid flying if you really dislike delayed flights, and why?**

We can also visualize the distributions of departure delays across months using side-by-side box plots:

* ggplot(nycflights, aes(x = factor(month), y = dep\_delay)) + geom\_boxplot()

**On time departure rate for NYC airports**

Suppose you will be flying out of NYC and want to know which of the three major NYC airports has the best on time departure rate of departing flights. Suppose also that for you a flight that is delayed for less than 5 minutes is basically “on time”. You consider any flight delayed for 5 minutes of more to be “delayed”.

In order to determine which airport has the best on time departure rate, we need to first classify each flight as “on time” or “delayed”, then group flights by origin airport, then calculate on time departure rates for each origin airport, and finally arrange the airports in descending order for on time departure percentage. Let’s start with classifying each flight as “on time” or “delayed” by creating a new variable with the mutate function.

* nycflights <- nycflights %>% mutate(dep\_type = ifelse(dep\_delay < 5, "on time", "delayed"))

The first argument in the mutate function is the name of the new variable we want to create, in this case dep\_type . Then if dep\_delay < 5 we classify the flight as "on time" and "delayed" if not, i.e. if the flight is delayed for 5 or more minutes.

Note that we are also overwriting the nycflights data frame with the new version of this data frame that includes the new dep\_type variable.

We can handle all the remaining steps in one code chunk:

* nycflights %>% group\_by(origin) %>% summarise(ot\_dep\_rate = sum(dep\_type == "on time") / n()) %>% arrange(desc(ot\_dep\_rate))

**Question: If you were selecting an airport simply based on time departure percentage, which NYC airport would you choose to fly out of?**

Hint:

* nycflights %>% group\_by(origin) %>% summarise(ot\_dep\_rate = sum(dep\_type == "on time") / n()) %>% arrange(desc(ot\_dep\_rate))

We can also visualize the distribution of on time departure rate across the three airports using a segmented bar plot.

* ggplot(data = nycflights, aes(x = origin, fill = dep\_type)) + geom\_bar()

**Question: Mutate the data frame so that it includes a new variable that contains the average speed, avg\_speed traveled by the plane for each flight (in mph). What is the tail number of the plane with the fastest avg\_speed?**

**Hint: Average speed can be calculated as distance divided by number of hours of travel, and note that air\_time is given in minutes. If you just want to show the avg\_speed and tailnum and none of the other variables, use the select function at the end of your pipe to select just these two variables with select(avg\_speed, tailnum) . You can Google this tail number to find out more about the aircraft.**

Solution (hint):

* nycflights <- nycflights %>% mutate(avg\_speed = distance / (air\_time/60)) nycflights %>% select(avg\_speed, tailnum) %>% arrange(desc(avg\_speed))

**Question: Make a scatterplot of avg\_speed vs. distance . What is the relationship between average speed and distance.**

**Question: Suppose you define a flight to be “on time” if it gets to the destination on time or earlier than expected, regardless of any departure delays. Mutate the data frame to create a new variable called arr\_type with levels "on time" and "delayed" based on this definition. Also mutate to create a new variable called del\_type with levels "on time" and "delayed" depending on whether there was only departure delay. Then, determine the on time arrival percentage based on whether the flight departed on time or not. What percent of flights that were "delayed" departing arrive "on time" ? *(numerical output =0.27)***

* nycflights %>% mutate(arr\_type = ifelse(arr\_delay <= 0, "on time", "delayed")) %>% mutate(dep\_type = ifelse(dep\_delay <= 0, "on time", "delayed")) %>% select(arr\_type, dep\_type) %>% table()