Introduction to data

Tom Buonora

nycflights contains flights that departed from the three major New York City airports in 2013.

Creating a reproducible lab report

The web site for the Bureau of Transportation Statistics has moved ! It is now here

Load data

```
data(nycflights)
# View(nycflights)
```

View column names

```
for ( i in names(nycflights))
{
  cat("\t",i,"\n")
}
```

```
	year
	month
	day
	dep_time
	dep_delay
	arr_time
	arr_delay
	carrier
	tailnum
	flight
	origin
	dest
	air_time
	distance
	hour
	minute
```

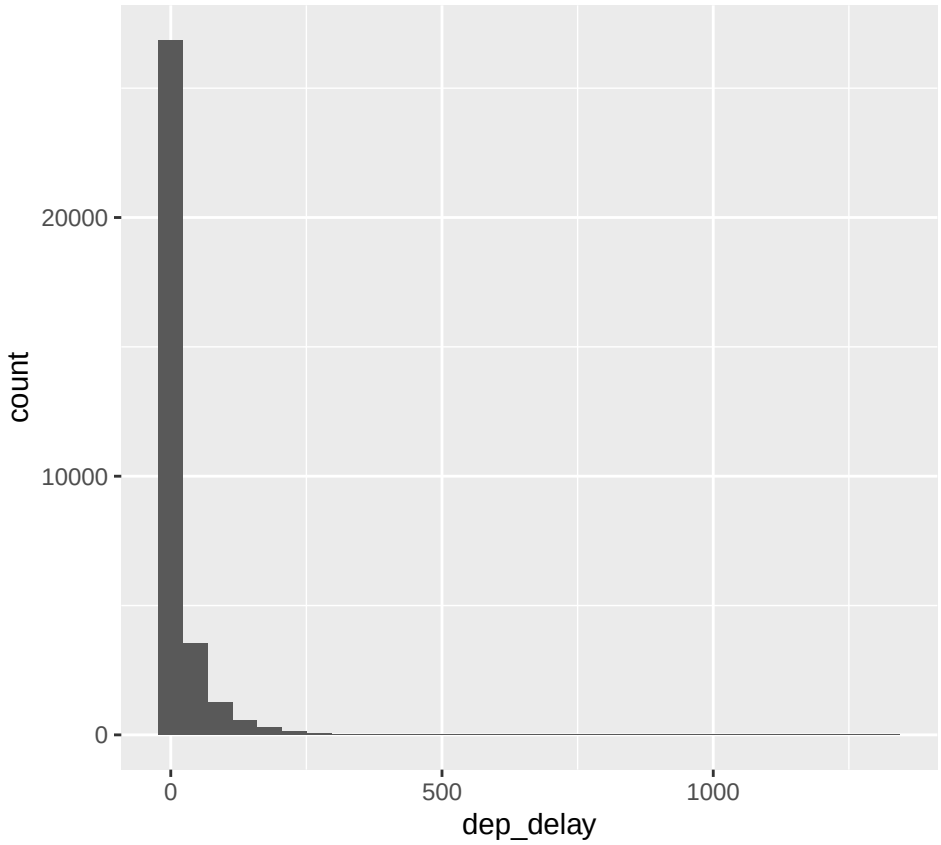
Analysis

Lab report

Departure delays

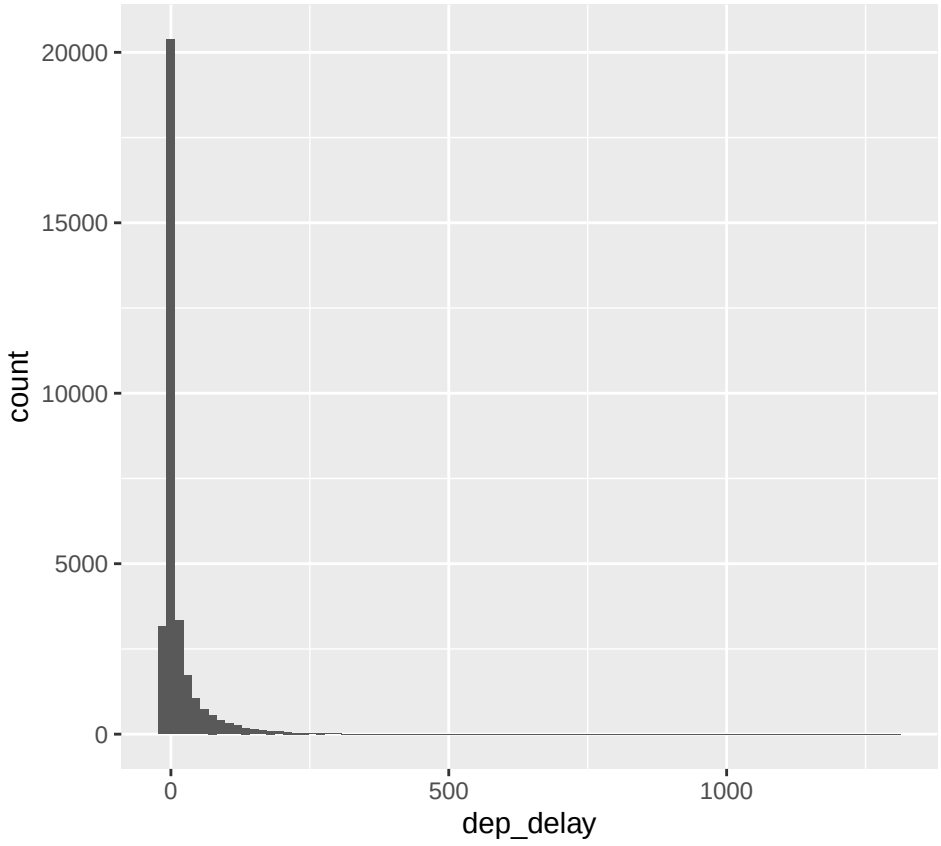
Plot departure delays of all flights with a histogram. The bin width will default to 30

```
ggplot(data = nycflights, aes(x = dep_delay)) +
  geom_histogram()
```



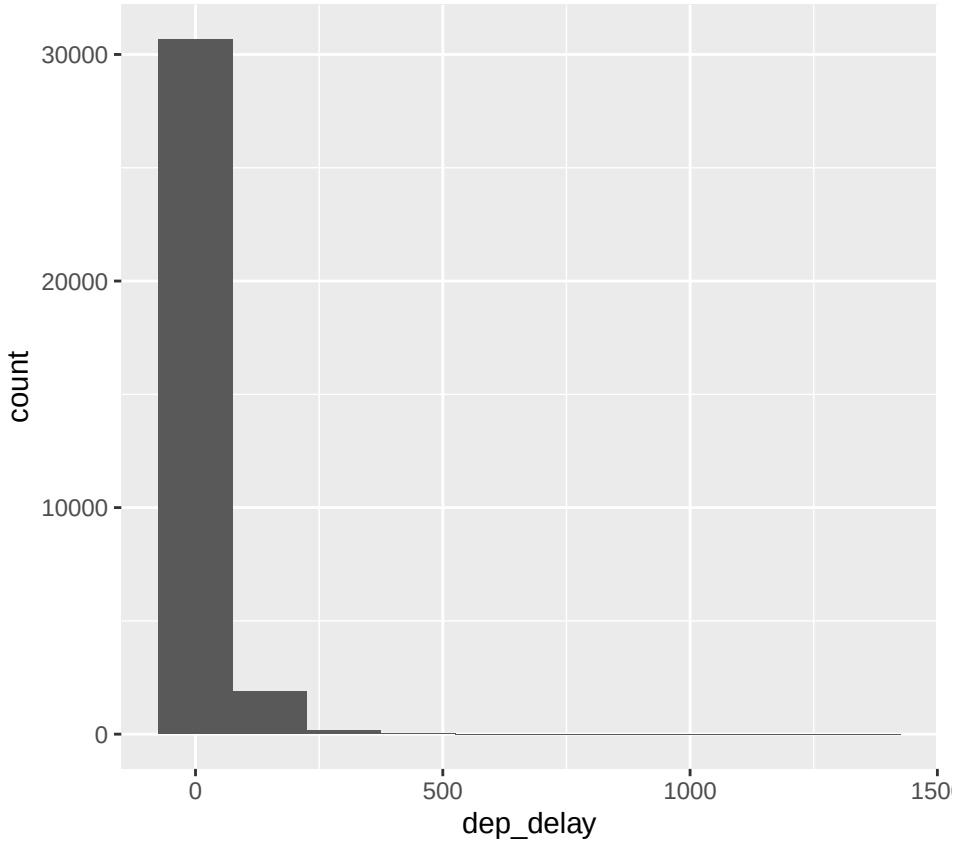
Plot again with binwidth of 15

```
ggplot(data = nycflights, aes(x = dep_delay)) +
  geom_histogram(binwidth = 15)
```



Again with 150

```
ggplot(data = nycflights, aes(x = dep_delay)) +
  geom_histogram(binwidth = 150)
```



1. Look carefully at these three histograms. How do they compare? Are features revealed in one that are obscured in another?

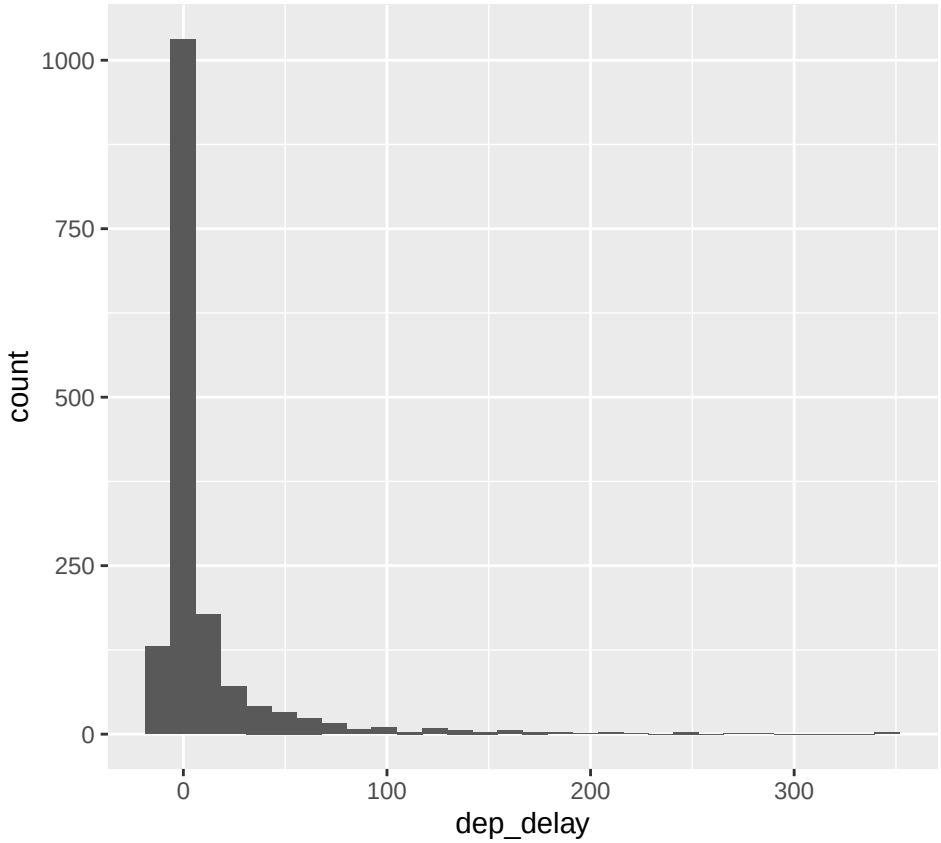
The binwidth of 150 will do a poor job of delineating between on time vs a 30 minute delay.

The binwidth of 15 does a pretty good job of displaying that about 10% of all flights have a small delay.

And 10% of all flights depart a bit early.

Plot the flights with LAX as a destination :

```
lax_flights <- nycflights %>%
  filter(dest == "LAX")
ggplot(data = lax_flights, aes(x = dep_delay)) +
  geom_histogram()
```



Display the mean and median of the LAX flights.

```
  # comment=NA just removes the hash tags
  # noquote() will remove the quotes
  # not sure what removes the [1]
  # knitr::kables removes everything
  # cat prints value only

lax_stats<-lax_flights %>%
  summarise(mean_dd   = mean(dep_delay),
            median_dd = median(dep_delay),
            n         = n())         # same as nrow()

# lax_stats is a tibble, which is a data type that doesnt work on a lot of usual methods

cat(sprintf("Mean=%.2f", lax_stats[1,"mean_dd"][1,1]))
```

```
## Mean=9.78
```

```
cat(sprintf("Median=%.2f", lax_stats[1,"median_dd"][1,1]))
```

```
## Median=-1.00
```

```
cat(sprintf("No. Flights=%s", lax_stats[1,"n"][1,1]))
```

```
## No. Flights=1583
```

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

- mean
- median
- sd
- var
- IQR
- min
- max

Note that each of these functions takes a single vector as an argument and returns a single value.

Select flights headed to San Francisco in February

```
sfo_feb_flights <- nycflights %>%
  filter(dest == "SFO", month == 2)
```

How many flights meet the criteria?

```
knitr::kable(nrow(sfo_feb_flights), col.names = "")
```

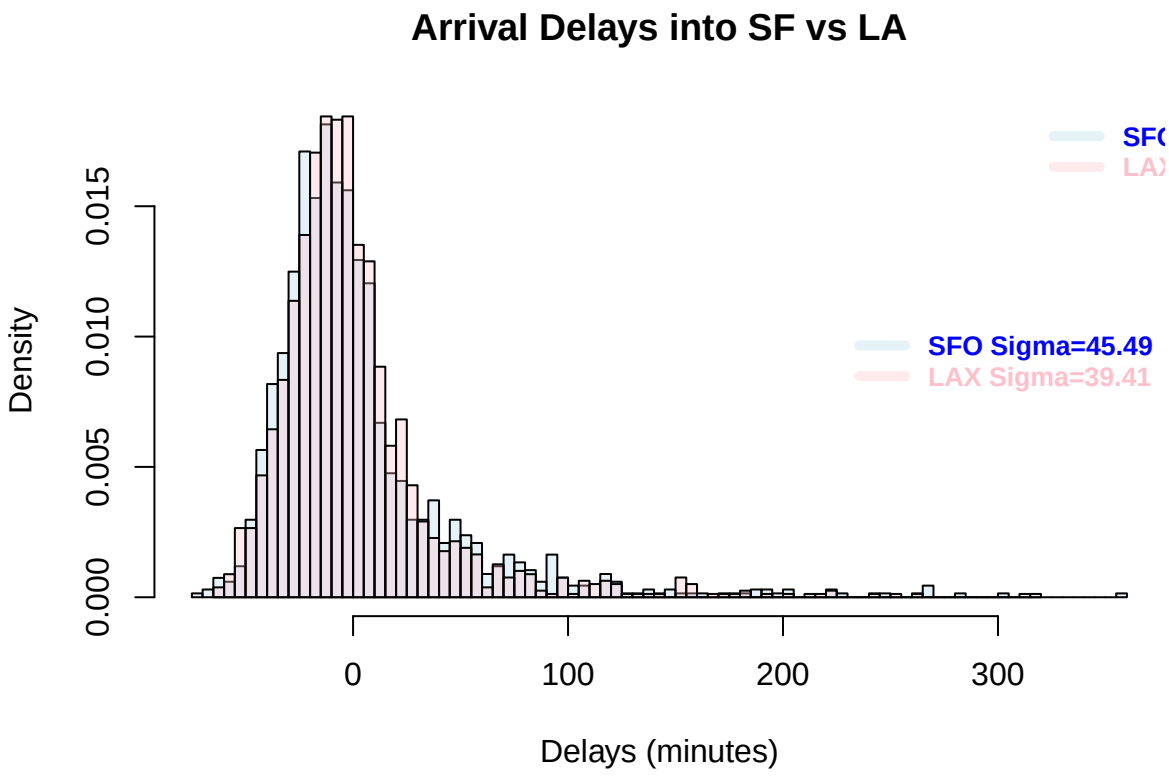
68

1. Describe the distribution of the **arrival** delays of these flights using a histogram and appropriate summary statistics. **Hint:** The summary statistics you use should depend on the shape of the distribution.

I decided to try to compare LAX vs SFO side by side. For me, this evolved into learning something useful.

```
# summarize by destination

# 1) Group the stats by destination

arr_delays_by_dest<-nycflights %>%
  group_by(dest) %>%
  summarise(median_dd = median(arr_delay), earliest=min(arr_delay),
            sd=sd(arr_delay), latest=max(arr_delay),  n_flights = n())


# 2) Create 2 labels that we will post the standard deviation of arrival delays for each city

dest1<-subset(arr_delays_by_dest,dest=='SFO')
dest2<-subset(arr_delays_by_dest,dest=='LAX')


# create labels for the standard deviation of JFK and SFO
```

```r
lg1=sprintf("SFO Sigma=%s",round(dest1$sd,2))
lg2=sprintf("LAX Sigma=%s",round(dest2$sd,2))


sfo_flights <- nycflights %>% filter(dest == "SFO")


# freq is TRUE for counts, FALSE for distribution
hg1<-hist(lax_flights$arr_delay, plot=FALSE, breaks=100)
# hg2<-hist(sfo2$arr_delay, plot=FALSE,  breaks=100)

# 3) Create a color thats nearly transparent
transparent_blue<-rgb(173,216,230,max = 255, alpha = 80, names = "lt.blue")
transparent_pink <- rgb(255,192,203, max = 255, alpha = 80, names = "lt.pink")

# plot the SF delays
hist(sfo_flights$arr_delay,   main="Arrival Delays into SF vs LA",
 xlab="Delays (minutes)",
col=transparent_blue,
freq=FALSE,  breaks=100)

# Now add LA
plot(hg1, col = transparent_pink, add = TRUE, freq=FALSE, breaks=100)


legend("topright", c("SFO", "LAX"), cex=0.8, col=c(transparent_blue,transparent_pink ),
       text.col = c("blue","pink"), text.width = 14, text.font=2, lty=1, lwd=5, bty="n");

legend("right", c(lg1, lg2), cex=0.8, col=c(transparent_blue,transparent_pink ),
       text.col = c("blue","pink"), text.font=2, lty=1, lwd=5, bty="n");
```



1. Calculate the median and interquartile range for `arr_delays` of flights in in the `sfo_feb_flights` data frame, grouped by carrier. Which carrier has the most variable arrival delays?

The IQR is the middle 50% so measures the range of the middle. It wont be effected by the outliers.

The Standard Deviation on the other hand measures how far all values are from the mean and its weighted more by larger differences.

```
arr_delays_by_carrier<-sfo_feb_flights %>%
  group_by(carrier) %>%
  summarise(median_dd = median(arr_delay), iqr= IQR(arr_delay), sd=sd(arr_delay),  n_flights = n())

highest_sd<-arr_delays_by_carrier[which.max(arr_delays_by_carrier$sd),]

highest_iqr<-arr_delays_by_carrier[which.max(arr_delays_by_carrier$iqr),]

sprintf("The carrier with the highest standard deviation is %s",highest_sd$carrier)
```

```
## [1] "The carrier with the highest standard deviation is UA"
```

```
sprintf("The carrier with the highest IQR is %s",highest_iqr$carrier)
```

```
## [1] "The carrier with the highest IQR is DL"
```

Departure delays by month

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

Let's think about how you could answer this question:

- First, calculate monthly averages for departure delays. With the new language you are learning, you could
  - `group_by` months, then
  - `summarise` mean departure delays.
- Then, you could to `arrange` these average delays in `desc`ending order

```
dep_delays_by_month<-nycflights %>%
  group_by(month) %>%
  summarise(mean_dd = mean(dep_delay)) %>%
  arrange(desc(mean_dd))

sprintf("The month with the greatest (mean) delays is %s",
        dep_delays_by_month[which.max(dep_delays_by_month$mean_dd),"month"])
```

```
## [1] "The month with the greatest (mean) delays is 7"
```

I was thinking the New Yorks winter would lead to greater delays. Now I think the increase in flights during the summer lead to delays.

1. Suppose you really dislike departure delays and you want to schedule your travel in a month that minimizes your potential departure delay leaving NYC. One option is to choose the month with the lowest mean departure delay. Another option is to choose the month with the lowest median departure delay. What are the pros and cons of these two choices?

The mean is influenced by outliers, for example if 1 flight was delayed by 30 hours, the median wont be effected Honestly, I might try to isolate the median by month and orgin

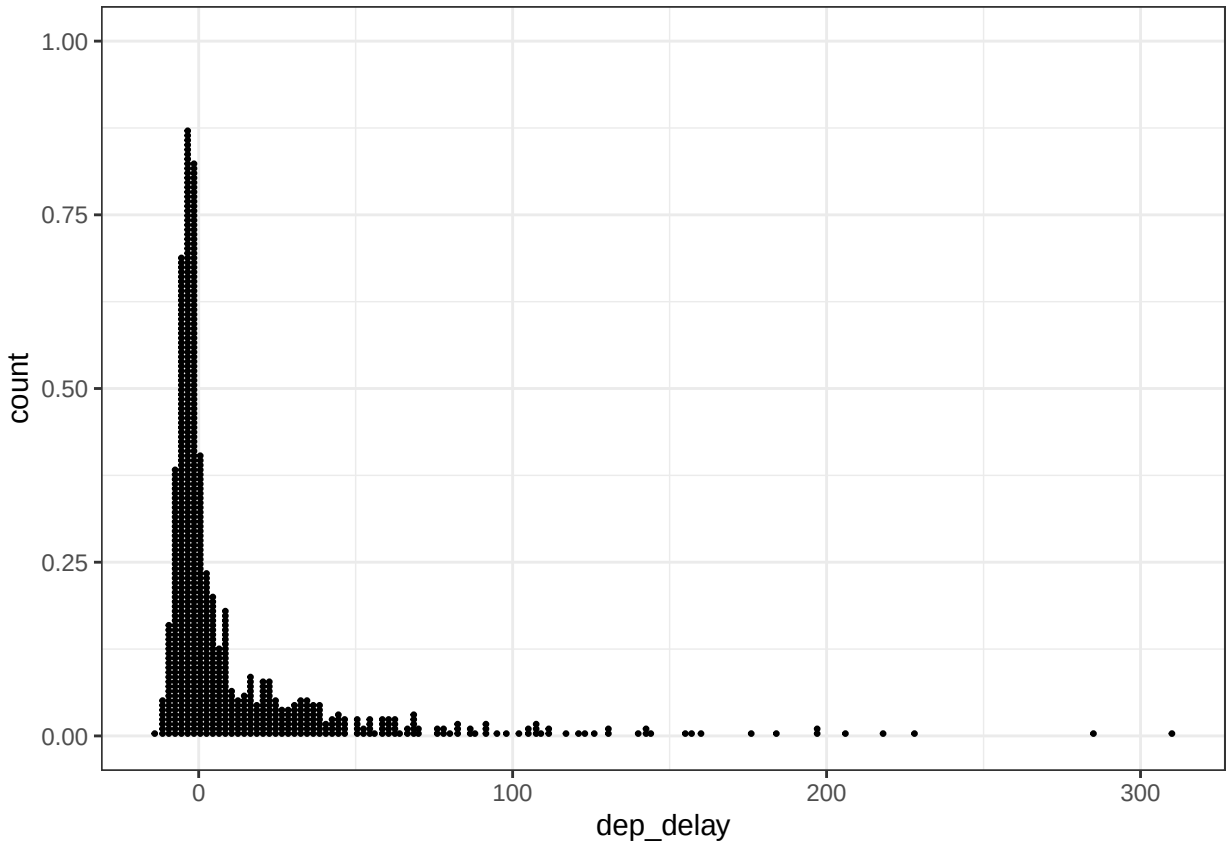
```
dep_delays_by_month_and_origin<-nycflights %>%
  group_by(month, origin) %>%
  summarise(median_dd = median(dep_delay))

dep_delays_by_month_and_origin[which.max(dep_delays_by_month_and_origin$median_dd),]
```

```
## # A tibble: 1 x 3
## # Groups:   month [1]
##   month origin median_dd
##   <int> <chr>      <dbl>
## 1    12 EWR            5
```

```
least_delays<-subset(nycflights,origin=='EWR' & month==2)
```

Then just to get a visual sense I lay out the delays in a stacked dot plot.

```
ggplot(least_delays, aes(x = dep_delay)) + geom_dotplot(binwidth = 1.5) + theme_bw()
```



On time departure rate for NYC airports

Create a new categorical field to describe each flight as either "on time" or "delayed"

```
nycflights <- nycflights %>%
  mutate(dep_type = ifelse(dep_delay < 5, "on time", "delayed"))
```

Create an on_time ratio for each origin. The "ot_dep_rate" is the number of "on time" flights divided by the total flights

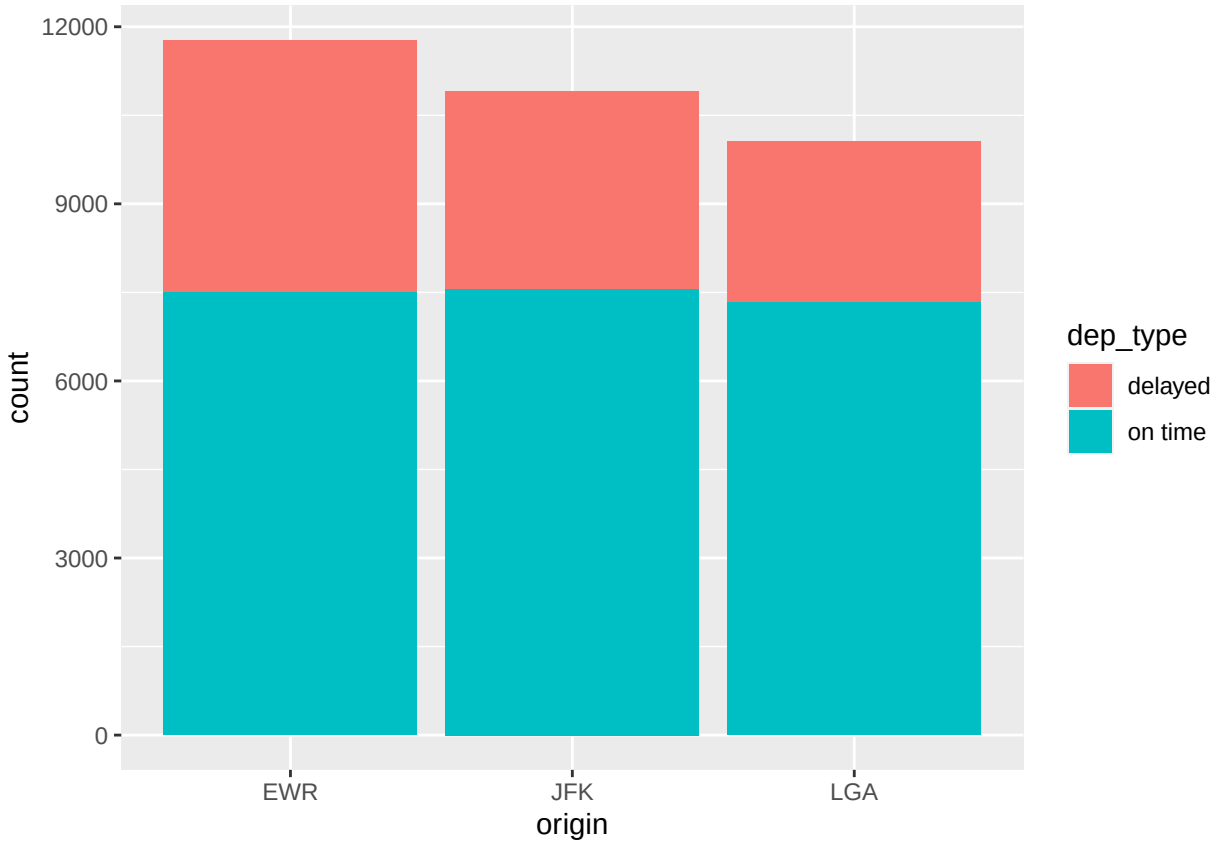
```
nycflights %>%
  group_by(origin) %>%
  summarise(ot_dep_rate = sum(dep_type == "on time") / n()) %>%
  arrange(desc(ot_dep_rate))
```

```
## # A tibble: 3 x 2
##   origin ot_dep_rate
##   <chr>        <dbl>
## 1 LGA          0.728
## 2 JFK          0.694
## 3 EWR          0.637
```

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

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

```
ggplot(data = nycflights, aes(x = origin, fill = dep_type)) +
  geom_bar()
```



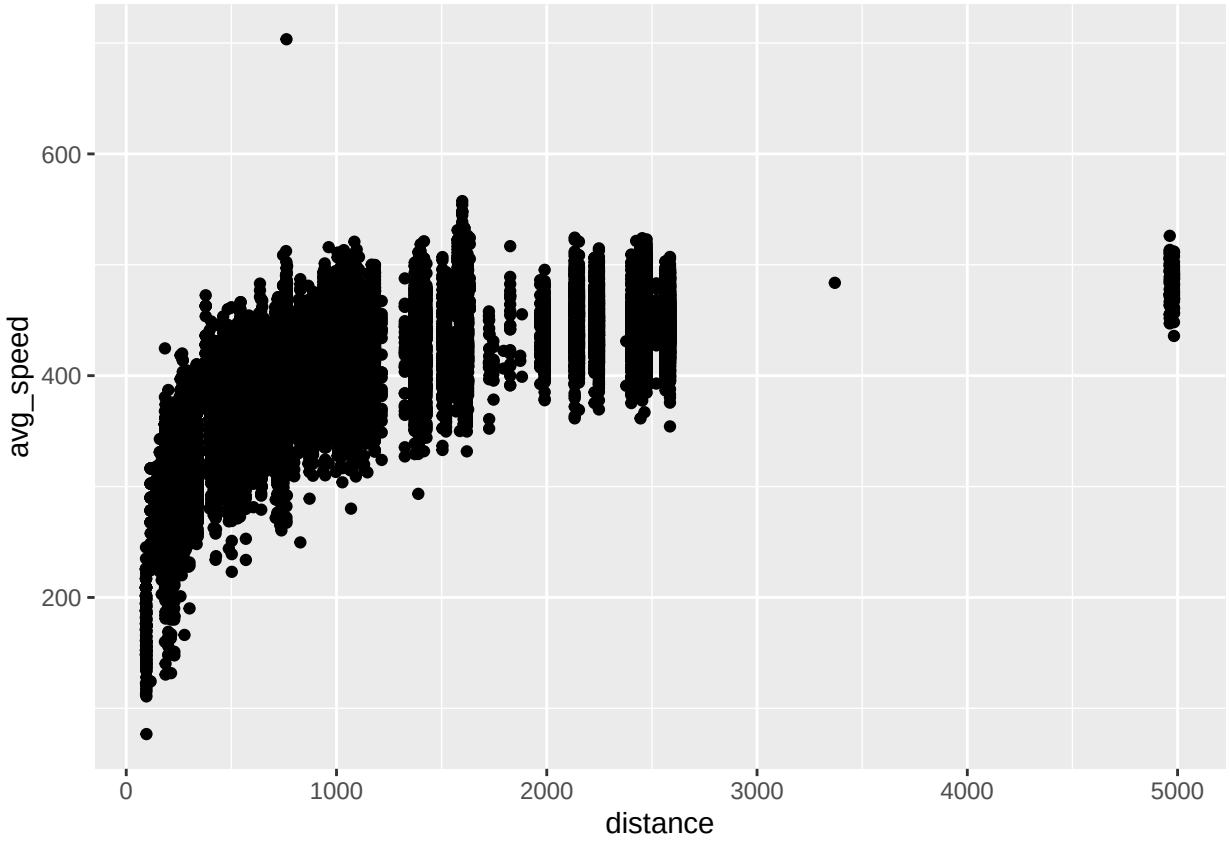
More Practice

1. 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). **Hint:** Average speed can be calculated as distance divided by number of hours of travel, and note that `air_time` is given in minutes.

```
nycflights <- nycflights %>%
  mutate(avg_speed =  nycflights$distance/(nycflights$air_time/60))
```

1. Make a scatterplot of `avg_speed` vs. `distance`. Describe the relationship between average speed and distance. **Hint:** Use `geom_point()`.

```
ggplot(data = nycflights, aes(x = distance , y = avg_speed)) +  geom_point()
```



I see a general increase in average speed as distance increases. I also notice the absence of destinations between 2800 and 4500 miles

```
subset(nycflights, distance > 2800 & distance < 4500)[c("dest","distance")]
```

```
## # A tibble: 1 x 2
##   dest  distance
##   <chr>    <dbl>
## 1 ANC       3370
```

I also notice some suspect data

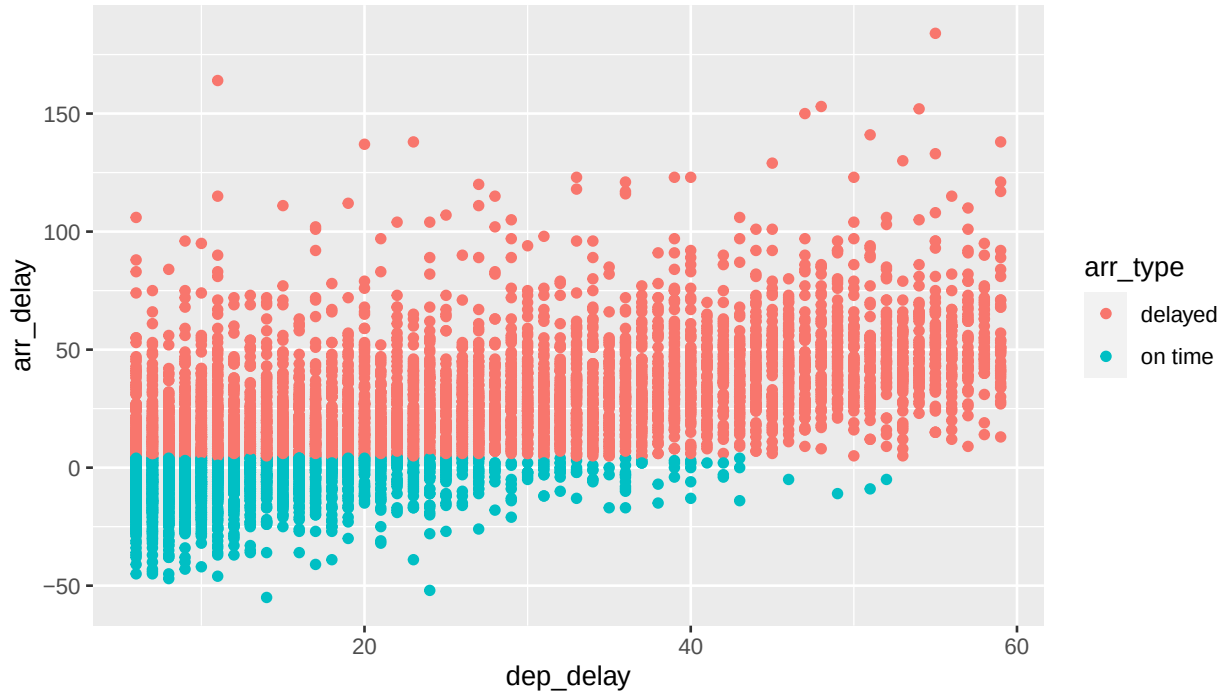
```
subset(nycflights, avg_speed > 600)[c("air_time","distance","avg_speed")]
```

```
## # A tibble: 1 x 3
##   air_time distance avg_speed
##      <dbl>    <dbl>     <dbl>
## 1       65      762      703.
```

1. Replicate the following plot. **Hint:** The data frame plotted only contains flights from American Airlines, Delta Airlines, and United Airlines, and the points are `colored` by `carrier`. Once you replicate the plot, determine (roughly) what the cutoff point is for departure delays where you can still expect to get to your destination on time.

I dont care about flights that departed on time so Ill filter them out. Then Ill plot the remaining by arriving on time or delayed. Also to make it easier to read, Ill filter out the siginficant departure delays.

```
delayed<-5
too_delayed_to_care<-60

tardy_flights<-subset(nycflights,dep_delay > delayed)

# create a category to display them by color
tardy_flights <- tardy_flights %>%
mutate(arr_type = ifelse(arr_delay < delayed, "on time", "delayed"))

tardy <- tardy_flights  %>%
  filter(dep_delay < too_delayed_to_care)
ggplot(data = tardy, aes(x = dep_delay, y = arr_delay, color = arr_type)) +
  geom_point()
```



It looks to me even with a small departure delay, you would still expect to arrive a little bit late