CUNY SPS DATA606 LAB2

Chinedu Onyeka

9/11/2021

Some define statistics as the field that focuses on turning information into knowledge. The first step in that process is to summarize 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 airports in 2013. We will generate simple graphical and numerical summaries of data on these flights and explore delay times. Since this is a large data set, along the way you'll also learn the indispensable skills of data processing and subsetting.

Getting started

Load packages

In this lab, we will explore and visualize the data using the **tidyverse** suite of packages. The data can be found in the companion package for OpenIntro labs, **openintro**.

Let's load the packages.

```
library(tidyverse)
library(openintro)
```

```
data(nycflights)
```

```
names(nycflights)
```

```
##  [1] "year"      "month"     "day"       "dep_time"  "dep_delay" "arr_time"
##  [7] "arr_delay" "carrier"   "tailnum"   "flight"    "origin"    "dest"
## [13] "air_time"  "distance"  "hour"      "minute"
```

```
glimpse(nycflights)
```

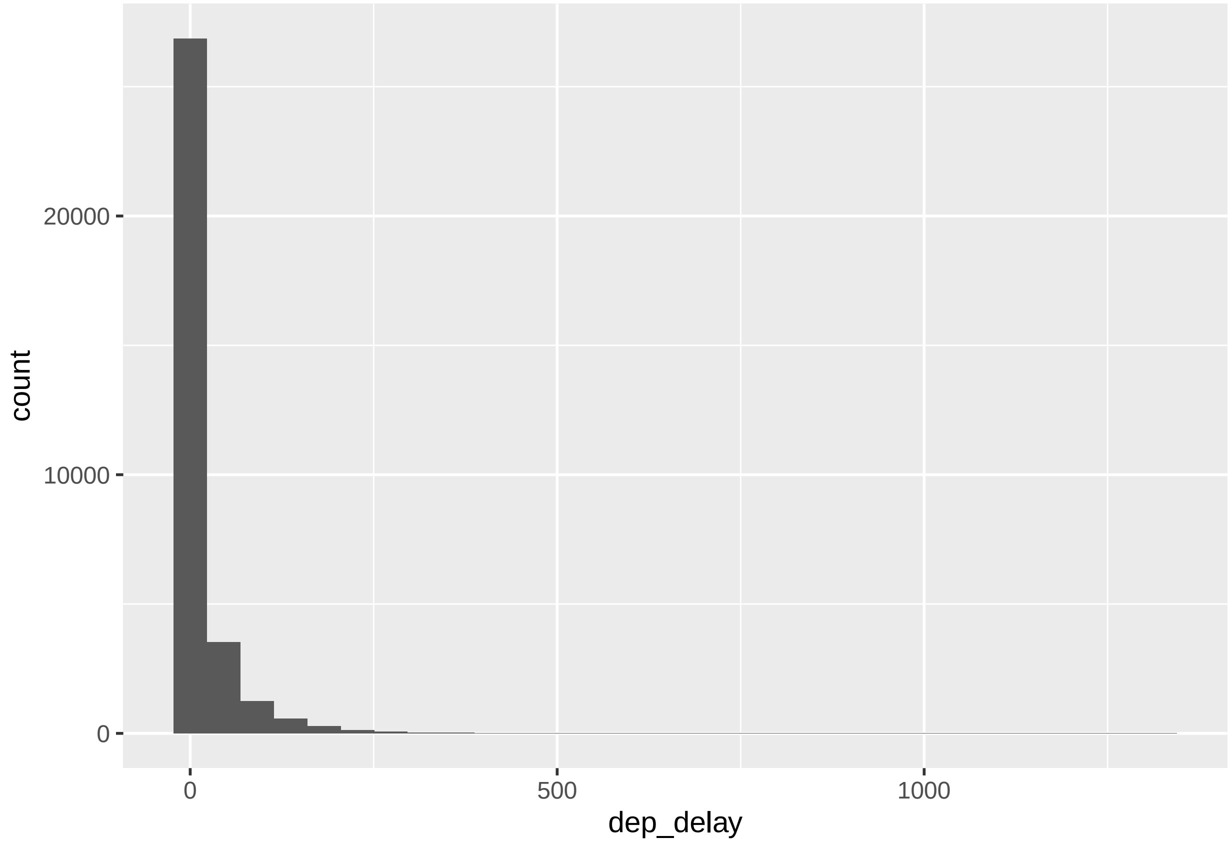
```
## Rows: 32,735
## Columns: 16
## $ year      <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, ~
## $ month     <int> 6, 5, 12, 5, 7, 1, 12, 8, 9, 4, 6, 11, 4, 3, 10, 1, 2, 8, 10~
## $ day       <int> 30, 7, 8, 14, 21, 1, 9, 13, 26, 30, 17, 22, 26, 25, 21, 23, ~
## $ dep_time  <int> 940, 1657, 859, 1841, 1102, 1817, 1259, 1920, 725, 1323, 940~
## $ dep_delay <dbl> 15, -3, -1, -4, -3, -3, 14, 85, -10, 62, 5, 5, -2, 115, -4, ~
## $ arr_time  <int> 1216, 2104, 1238, 2122, 1230, 2008, 1617, 2032, 1027, 1549, ~
## $ arr_delay <dbl> -4, 10, 11, -34, -8, 3, 22, 71, -8, 60, -4, -2, 22, 91, -6, ~
## $ carrier   <chr> "VX", "DL", "DL", "DL", "9E", "AA", "WN", "B6", "AA", "EV", ~
## $ tailnum   <chr> "N626VA", "N3760C", "N712TW", "N914DL", "N823AY", "N3AXAA", ~
## $ flight    <int> 407, 329, 422, 2391, 3652, 353, 1428, 1407, 2279, 4162, 20, ~
## $ origin    <chr> "JFK", "JFK", "JFK", "JFK", "LGA", "LGA", "EWR", "JFK", "LGA~
## $ dest      <chr> "LAX", "SJU", "LAX", "TPA", "ORF", "ORD", "HOU", "IAD", "MIA~
## $ air_time  <dbl> 313, 216, 376, 135, 50, 138, 240, 48, 148, 110, 50, 161, 87,~
## $ distance  <dbl> 2475, 1598, 2475, 1005, 296, 733, 1411, 228, 1096, 820, 264,~
## $ hour      <dbl> 9, 16, 8, 18, 11, 18, 12, 19, 7, 13, 9, 13, 8, 20, 12, 20, 6~
## $ minute    <dbl> 40, 57, 59, 41, 2, 17, 59, 20, 25, 23, 40, 20, 9, 54, 17, 24~
```

Analysis

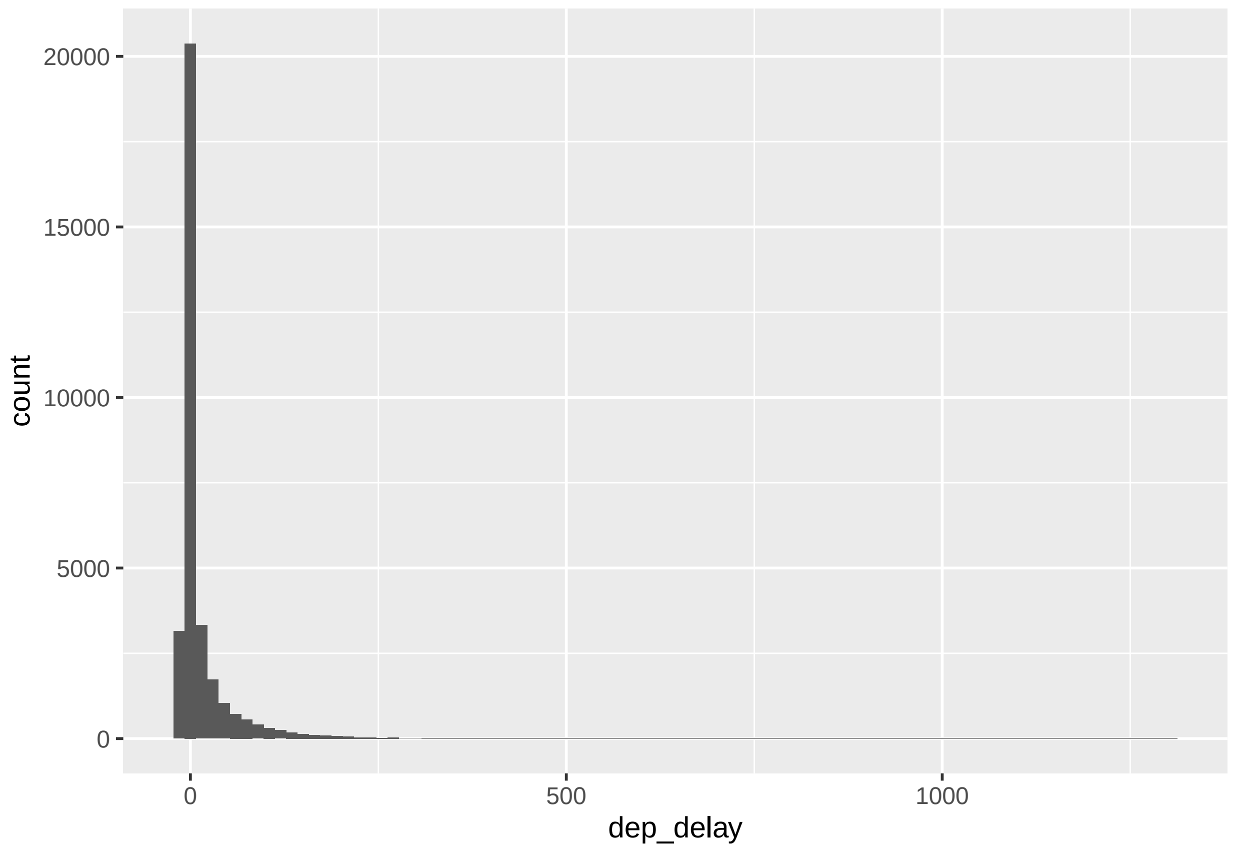
Departure delays

Let's start by examing the distribution of departure delays of all flights with a histogram.

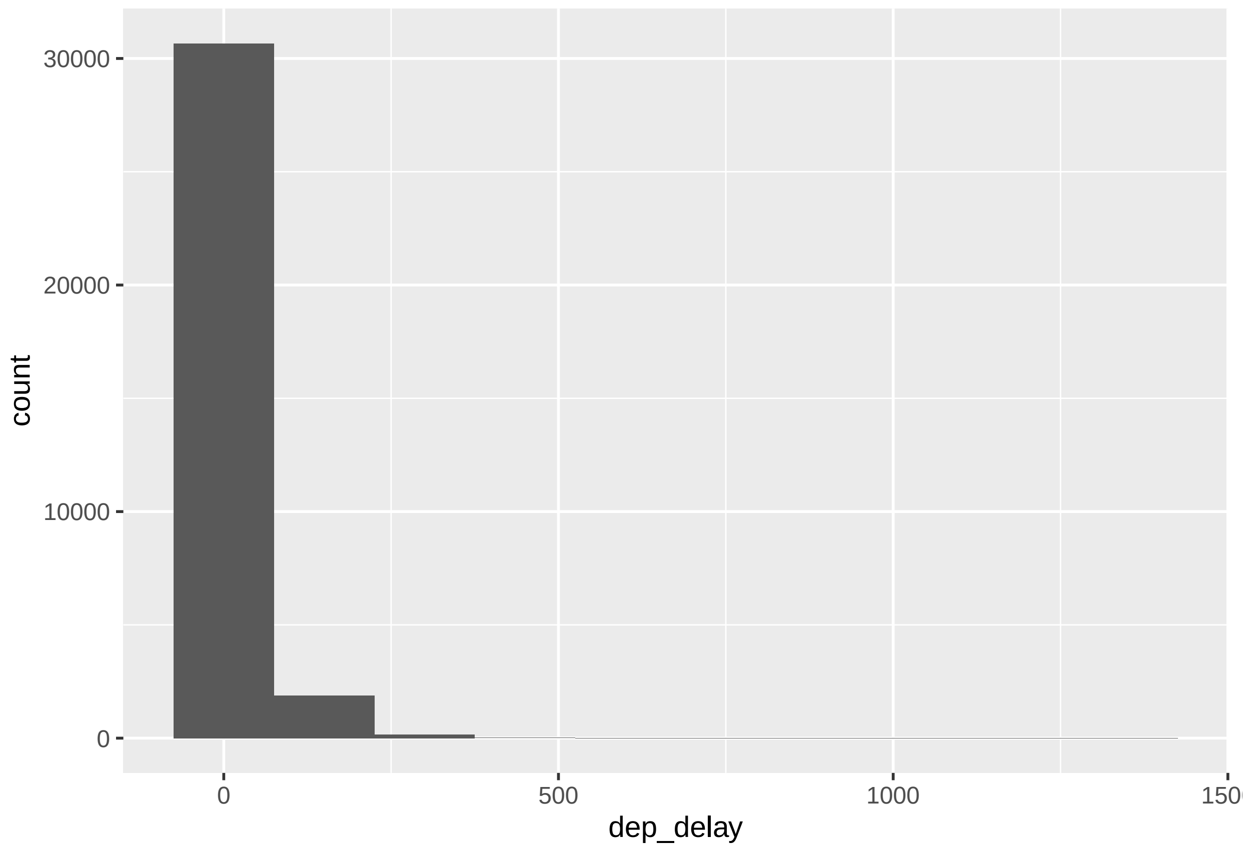
```
ggplot(data = nycflights, aes(x = dep_delay)) +
  geom_histogram()
```



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



```
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?

Solution 1:

These three histograms show how differences in bin width can affect or obscure data. The third histogram completely obscured the distribution of data and barely showed a count for a broad category of values. The second histogram showed a spike somehwere near zero which was not clearly visible in the other two histograms.The first histogram is better than the third in displaying the distribution, but still obscured some of the first data. In my opinion, I prefer the second histogram as it seemed less obscuring than others to me. In summary, one need be careful in selecting the optimal number of bins (or bin width) to use for histograms as there is no single answer for all cases. Too many bins may end up making the histogram pointless and showing each data as a single point while having bigger bins may completely obscure the data.

2. 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?

Solution 2:

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

```
## [1] 68
```

```
sfo_feb_flights %>% count()
```

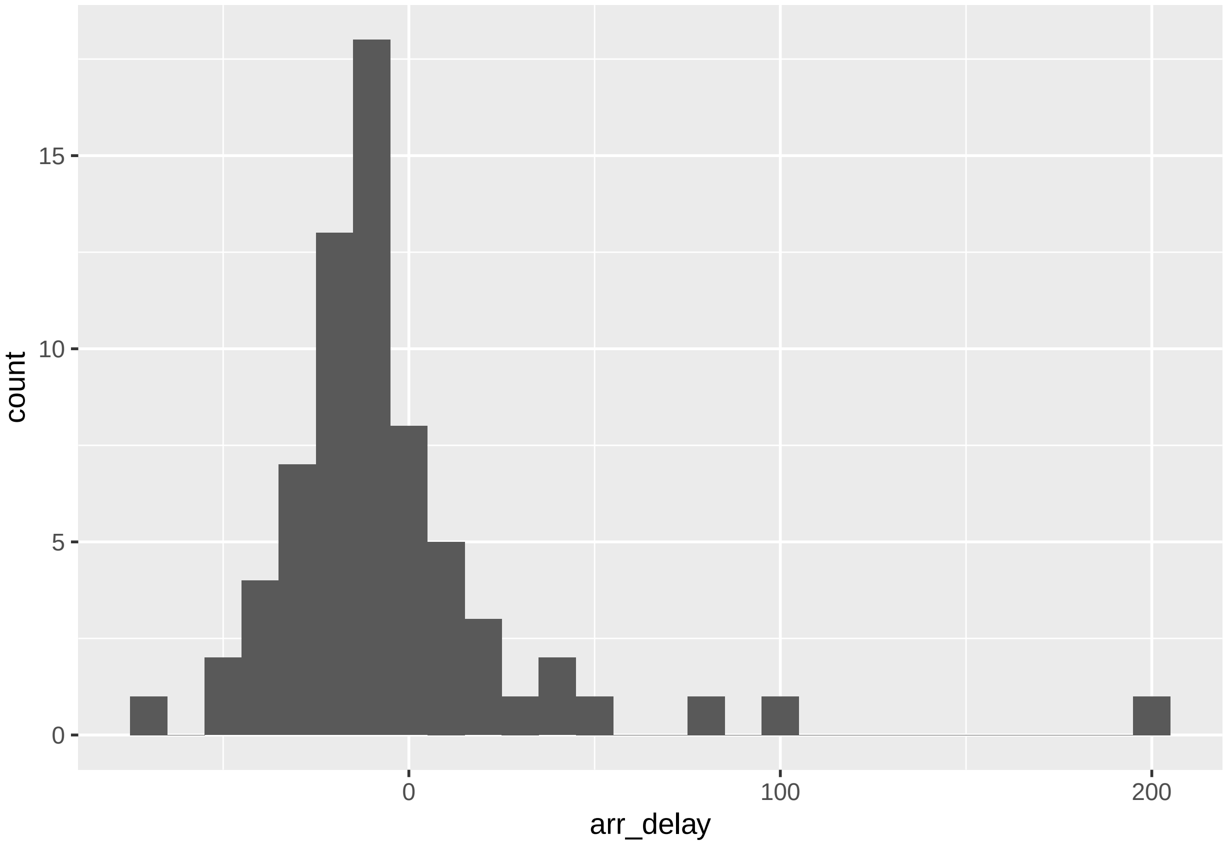
```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    68
```

The number of flights that meet this criteria is 68 flights as seen from the results of the nrow function of base R or the count function of dplyr.

3. 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.

Solution 3:

```
ggplot(data = sfo_feb_flights, aes(x = arr_delay)) + geom_histogram(binwidth = 10)
```



```
sfo_feb_flights %>% summarise(mean_ad = mean(arr_delay), median_ad = median(arr_delay),
                              IQR_ad = IQR(arr_delay), n_flights = n())
```

```
## # A tibble: 1 x 4
##   mean_ad median_ad IQR_ad n_flights
##     <dbl>     <dbl>  <dbl>     <int>
## 1    -4.5       -11   23.2        68
```

4. 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?

Solution 4:

```
sfo_feb_flights %>% group_by(carrier) %>%
  summarise(median_ad = median(arr_delay), iqr_ad = IQR(arr_delay), n_flights = n())
```

```
## # A tibble: 5 x 4
##   carrier median_ad iqr_ad n_flights
##   <chr>       <dbl>  <dbl>     <int>
## 1 AA            5     17.5        10
## 2 B6          -10.5   12.2         6
## 3 DL          -15     22          19
## 4 UA          -10     22          21
## 5 VX          -22.5   21.2        12
```

The carriers with the most variable arrival delays are DL and UA since they both have the highest IQR(Inter quartile range).

5. 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?

Solution 5:

```
nycflights %>%
  group_by(month) %>%
  summarise(mean_dd = mean(dep_delay), median_dd = median(dep_delay)) %>%
  arrange(desc(mean_dd), desc(median_dd))
```

```
## # A tibble: 12 x 3
##    month mean_dd median_dd
##    <int>   <dbl>     <dbl>
##  1     7   20.8          0
##  2     6   20.4          0
##  3    12   17.4          1
##  4     4   14.6         -2
##  5     3   13.5         -1
##  6     5   13.3         -1
##  7     8   12.6         -1
##  8     2   10.7         -2
##  9     1   10.2         -2
## 10     9    6.87        -3
## 11    11    6.10        -2
## 12    10    5.88        -3
```

The pro of choosing the lowest mean departure delay is that the mean will take outliers into account while the pro of choosing the lowest median is that the median will give the value at the center of the data.The con of choosing the lowest mean departure delay is that it can be misleading if there are outliers with substantial values (maybe a delay as a result of unusual events) that will make the mean unusually high and not be representative of the actual situation while the con of choosing the median is that it will not account for outliers or unusual events. In this scenario the month with the lowest mean departure delay is October(10) and the months with the highest mean departure delay are July (7), June(6), and December(12). We can understand easily that June and July are peak summer travel times and December is Christmas where travel is usually also high. The months of September, October, and November have low travel delays which may be as a result of low travel demand or traffic.

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

Solution 6:

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

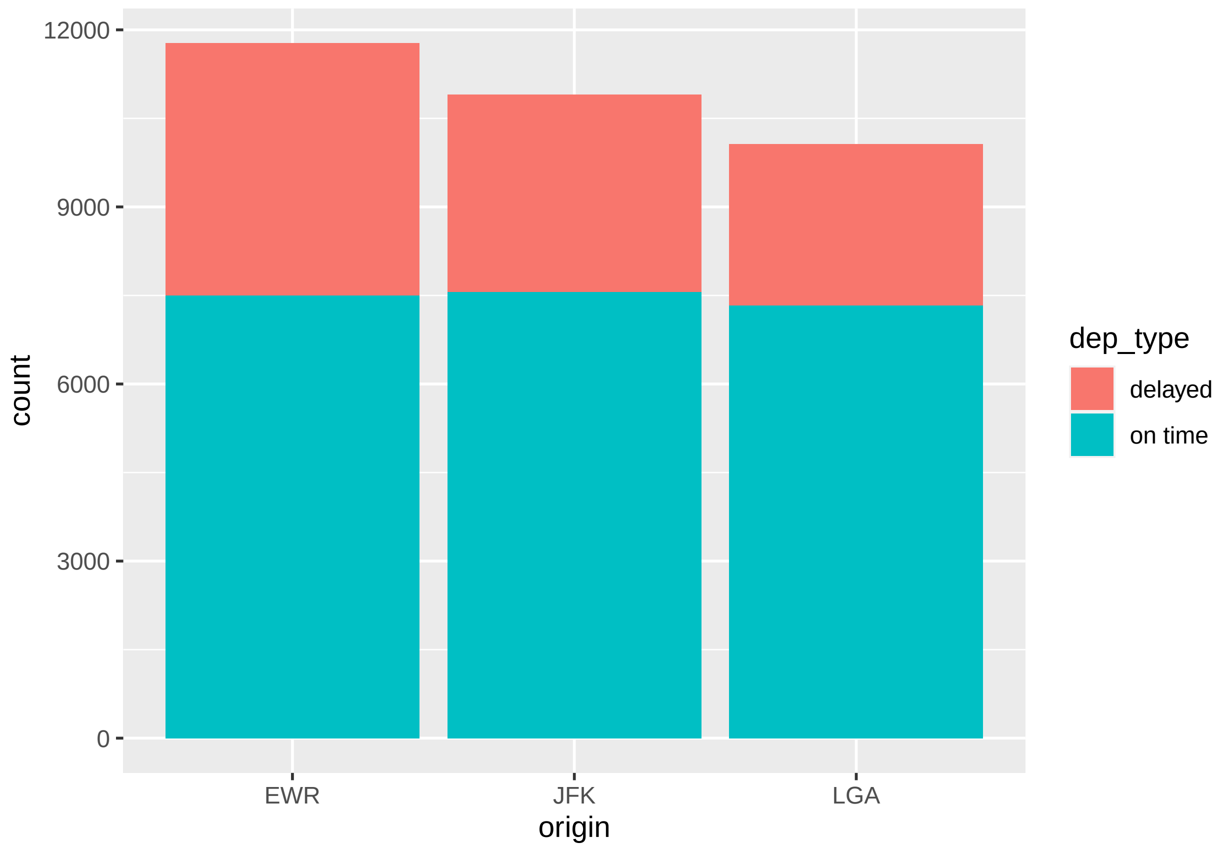
```
nycflights %>%
  group_by(origin) %>%
  summarise(ot_dep_rate = sum(dep_type == "on time") / n()) %>%
  arrange(desc(ot_dep_rate))
```

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

The airport with the highest on time departure rate is LGA and hence, I will choose to fly out of LGA (LaGuardia Airport)

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()
```



From the bar plots, it appears that EWR has the highest count of on time flights while LGA has the lowest count of on time flights. However, the proportion of on time flights out of LGA is higher than the other airports as seen from the summarized data above. This goes to show that visualizations should be accompanied with some descriptive statistics to make better decisions as required.

---

7. 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.

Solution 7:

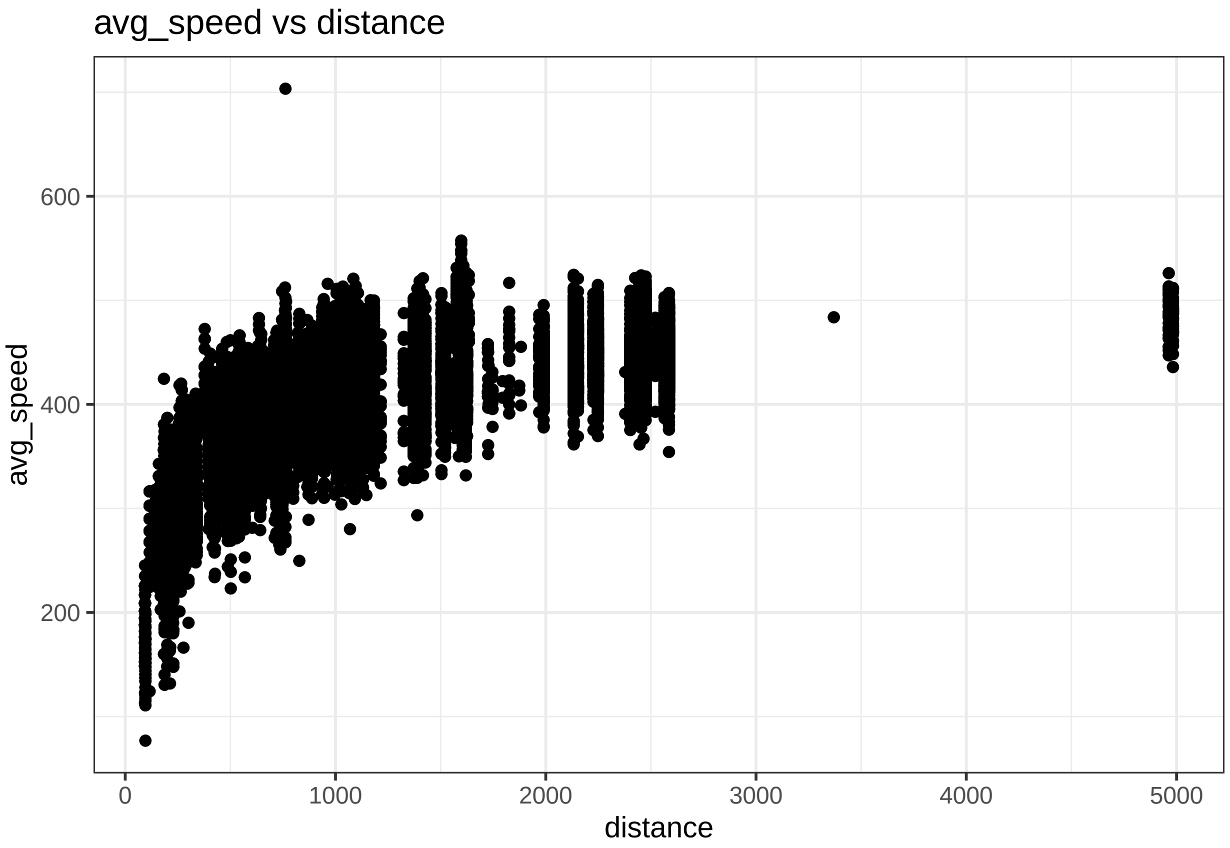
```
nycflights <- nycflights %>% mutate(avg_speed = distance/(air_time/60))
head(nycflights %>% select(distance, air_time, avg_speed))
```

```
## # A tibble: 6 x 3
##   distance air_time avg_speed
##      <dbl>    <dbl>     <dbl>
## 1     2475      313      474.
## 2     1598      216      444.
## 3     2475      376      395.
## 4     1005      135      447.
## 5      296       50      355.
## 6      733      138      319.
```

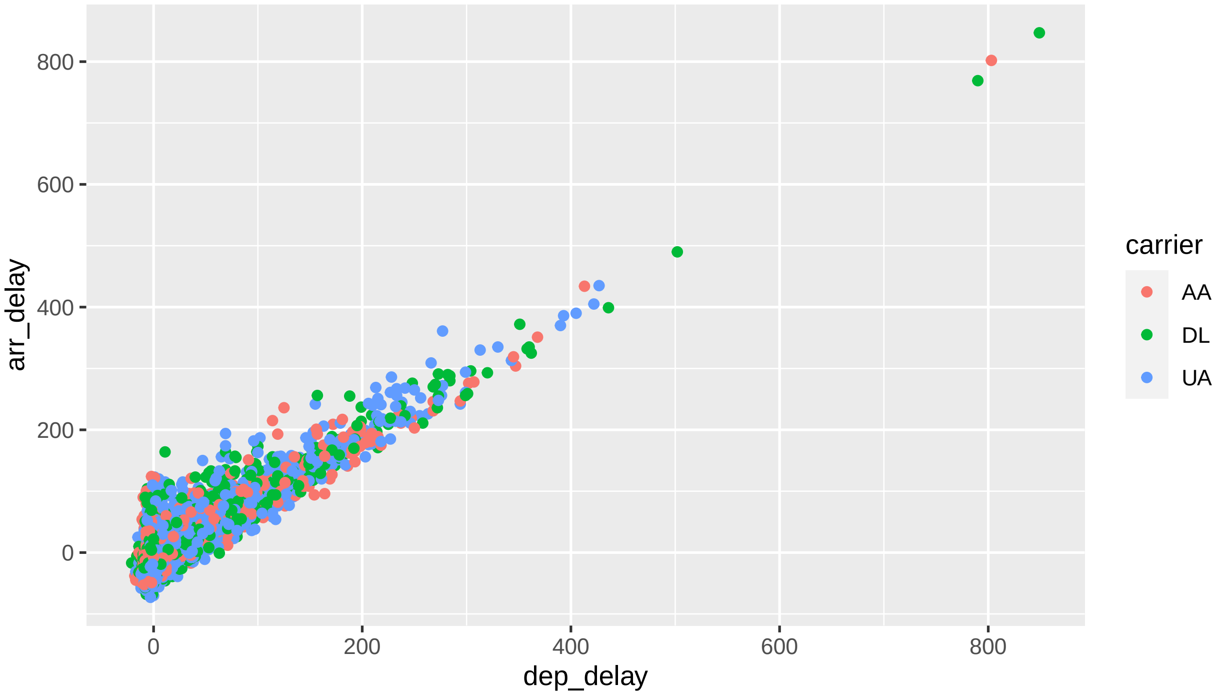
8. Make a scatterplot of avg_speed vs. distance. Describe the relationship between average speed and distance. **Hint:** Use geom_point().

Solution 8:

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

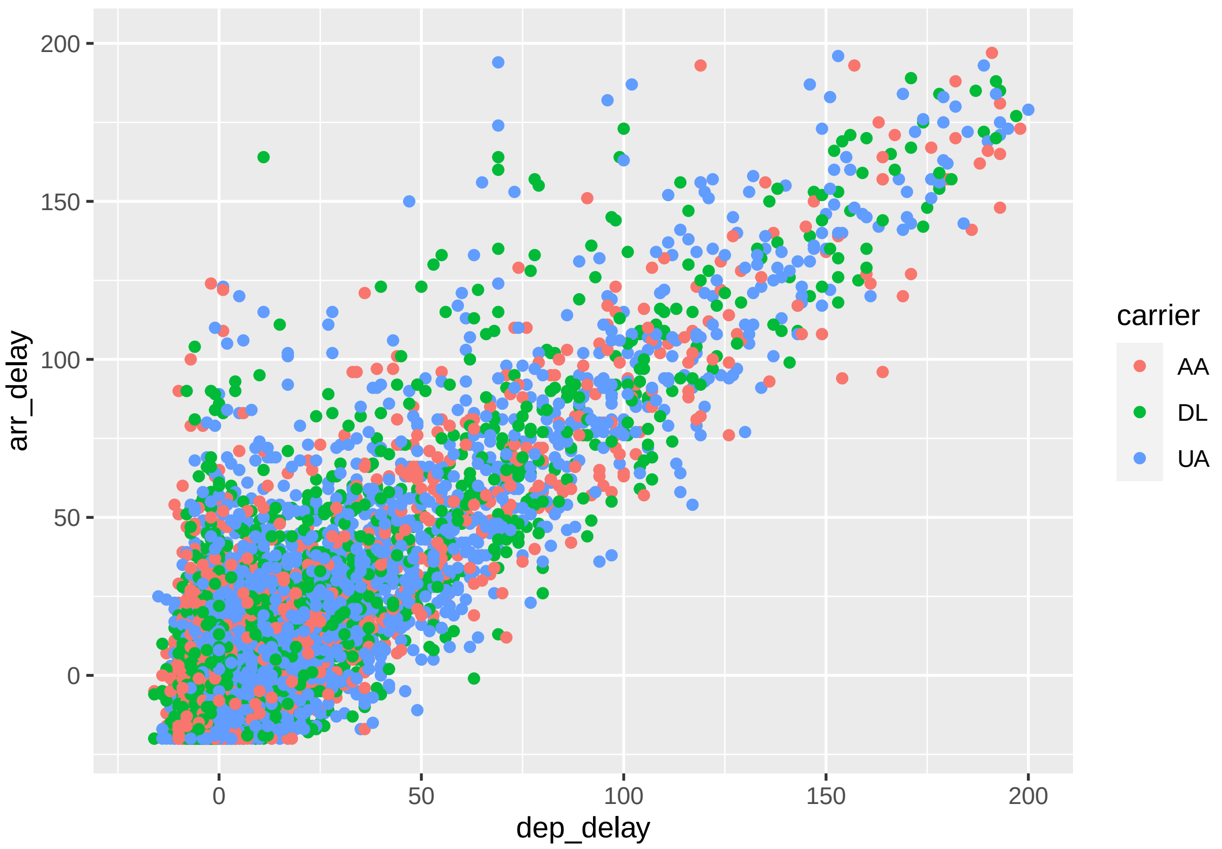


9. Replicate the following plot. **Hint:** The data frame plotted only contains flights from American Airlines, Delta Airlines, and United Airlines, and the points are colored by carrier. Once you replicate the plot, determine (roughly) what the cutoff point is for departure delays where you can still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



Solution 9:

```
dl_aa_ua <- nycflights %>%
  filter(carrier == "AA" | carrier == "DL" | carrier == "UA")
ggplot(data = dl_aa_ua, aes(x = dep_delay, y = arr_delay, color = carrier)) +
  geom_point() + xlim(-20, 200) + ylim(-20, 200)
```



The cut off point is around 23 - 25 minutes after which you start to see an upward trend or say increase in arrival time as departure time increases.