Math 300 NTI Lesson 4

Boxplots and Barcharts

Professor Bradley Warner

June, 2022

Contents

Objectives

1. Create and interpret boxplots using `ggplot()`. This includes quartiles, interquartile range, whiskers, and outliers.
2. Compare and contrast boxplots, barcharts, histograms, and pie charts.
3. Create and interpret barcharts for one or two categorical variables using `ggplot()`.
4. Create tables to summarize one or two categorical variables.

Reading

Chapter 2.7 - 2.9

Lesson

Remember that you will be running this more like a lab than a lecture. You want them using `R` and answering questions.

Work through the learning checks LC2.22 - LC2.37.

- Again, you want to emphasize that we are starting to look at different plots. These plots change based on the nature of our data. The boxplot was developed prior to widespread use of computers. It is comparable to the use of a histogram and/or density plot. It is meant for a single quantitative variable. It a second categorical variable is added, we can generate side-by-side boxplots.
- The barchart is for a categorical, qualitative, variables. It can be argued that a table is just as informative. If a second categorical variable is added, we must consider how to visually represent it. A table is often better than a barchart especially for a single variable. We will `tidyverse` and base code to create a table in our solution.

- We are doing some more data wrangling. This is a preview of material to come.
- We added a theme to our plot in this lesson to make the plot more appropriate for presentation to an audience. In an exploratory analysis, we don't care about axis labels or plot themes.
- Clarify the use of `fill` and `color`. The former is the fill color and the later the color of the bounding box.
- See the following for more on the principles of visualizing data and technical details.
- Read 2.9.2 and help students understand specifying arguments to an `R` function. Remember, what do we want `R` to do? What does `R` need to do this?

Setup

```
library(nycflights13)
library(ggplot2)
library(dplyr)
```

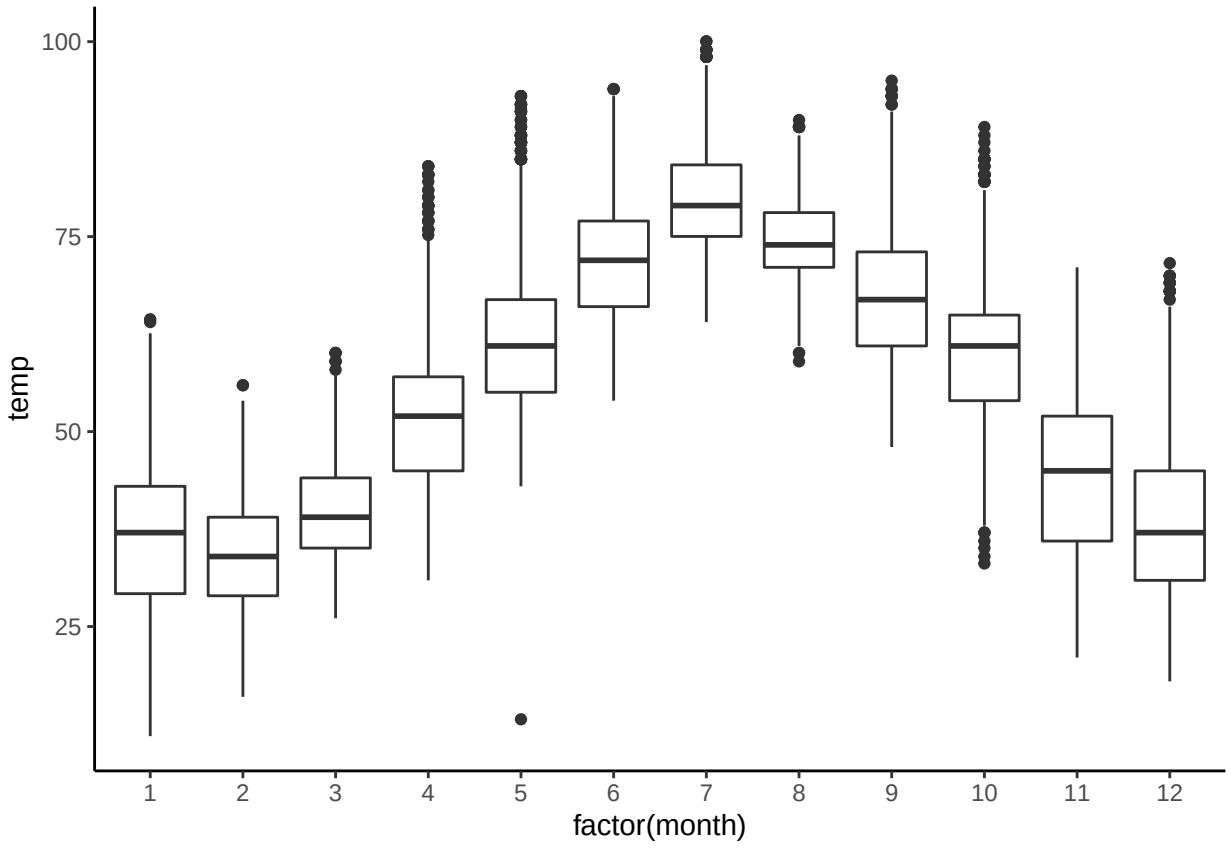
Create the side-by-side boxplots from the book.

Explain why we have to use factor in `aes()` function.

Walk through explaining the elements of the boxplot.

```
# Boxplots in the reading of section 2.7, adding a different theme.

ggplot(data = weather, mapping = aes(x = factor(month), y = temp)) +
  geom_boxplot() +
  theme_classic()
```



LC 2.22 (Objective 1)

(LC2.22) What does the dot at the bottom of the plot for May correspond to? Explain what might have occurred in May to produce this point.

Solution: It appears to be an outlier. Let's revisit the use of the `filter` command to hone in on it. We want all data points where the `month` is 5 and `temp<25`

```
#Explore the outlier.
weather %>%
  filter(month == 5 & temp < 25)
```

```
## # A tibble: 1 x 15
##   origin  year month   day  hour  temp  dewp humid wind_dir wind_speed wind_gust
##   <chr>  <int> <int> <int> <int> <dbl> <dbl> <dbl>    <dbl>      <dbl>     <dbl>
## 1 JFK     2013     5     8    22  13.1  12.0  95.3       80       8.06        NA
## # ... with 4 more variables: precip <dbl>, pressure <dbl>, visib <dbl>,
## #   time_hour <dttm>
```

There appears to be only one hour and only at JFK that recorded 13.1 F (-10.5 C) in the month of May. This is probably a data entry mistake! Why wasn't the weather at least similar at EWR (Newark) and LGA (LaGuardia)? Can we drop it from further analysis?

LC 2.23 (Objective 1)

(LC2.23) Which months have the highest variability in temperature? What reasons do you think this is?

Solution: We are now interested in the **spread** of the data. One measure, some of you may have seen previously, is the standard deviation. But in this plot we can read off the Interquartile Range (IQR):

- The distance from the 1st to the 3rd quartiles i.e. the length of the boxes
- You can also think of this as the spread of the **middle 50%** of the data

Just from eyeballing it, it seems

- November has the biggest IQR, i.e. the widest box, so has the most variation in temperature
- August has the smallest IQR, i.e. the narrowest box, so is the most consistent temperature-wise

Here's how we compute the exact IQR values for each month (we'll see this more in depth when we learn more about data wrangling later in the course):

- `group` the observations by `month` then
- for each `group`, i.e. `month`, `summarize` it by applying the summary statistic function `IQR()`, while making sure to skip over missing data via `na.rm=TRUE` then
- `arrange` the table in `desc`ending order of `IQR`

```
# Which month has the most variability in temperature?
weather %>%
  group_by(month) %>%
  summarize(IQR = IQR(temp, na.rm = TRUE)) %>%
  arrange(desc(IQR))
```

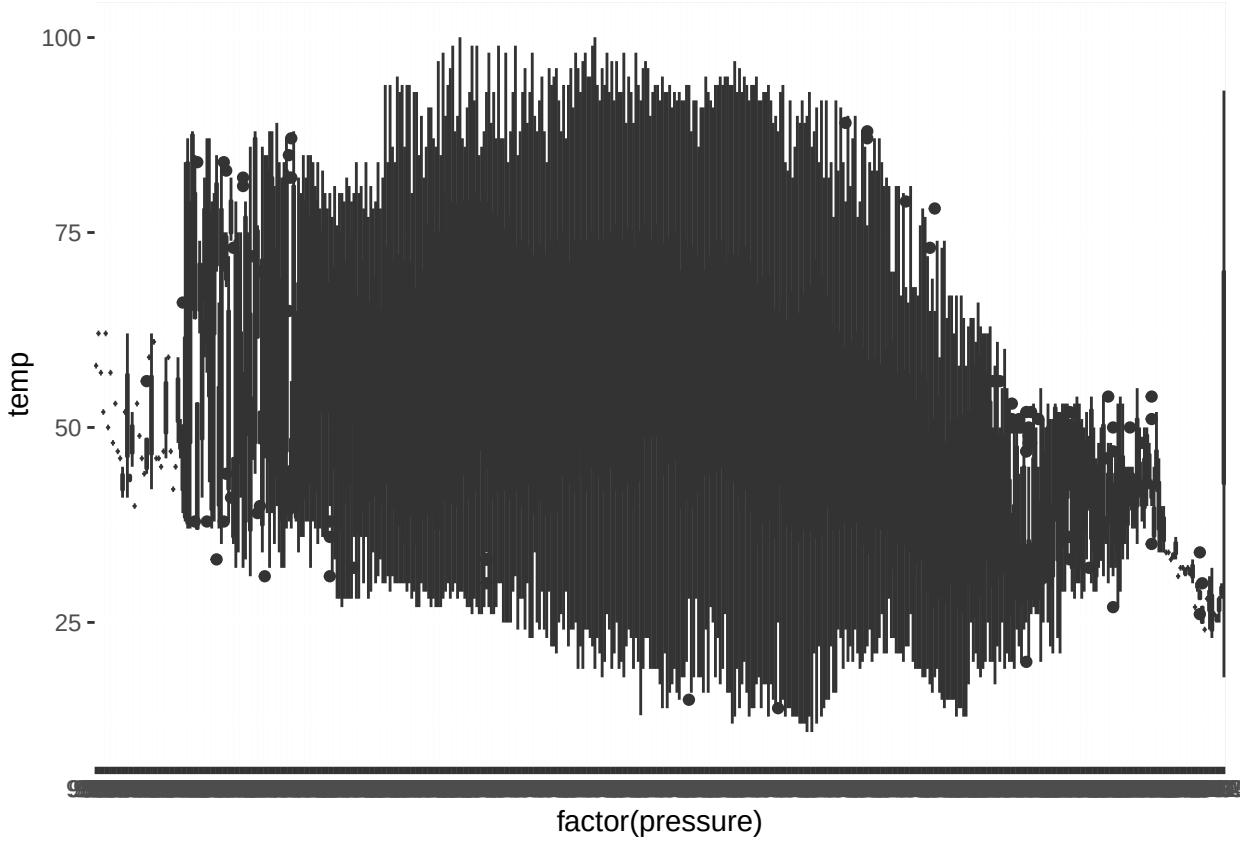
```
## # A tibble: 12 x 2
##    month   IQR
##    <int> <dbl>
##  1    11 16.0
##  2    12 14.0
##  3     1 13.8
##  4     9 12.1
##  5     4 12.1
##  6     5 11.9
##  7     6 11.0
##  8    10 11.0
##  9     2 10.1
## 10     7  9.18
## 11     3  9
## 12     8  7.02
```

LC 2.24 (Objective 1)

(LC2.24) We looked at the distribution of the numerical variable `temp` split by the numerical variable `month` that we converted to a categorical variable using the `factor()` function. Why would a boxplot of `temp` split by the numerical variable `pressure` similarly converted to a categorical variable using the `factor()` not be informative?

Solution: Because there are 12 unique values of `month` yielding only 12 boxes in our boxplot. There are many more unique values of `pressure` (469 unique values in fact), because values are to the first decimal place. This would lead to 469 boxes, which is too many for people to digest.

```
# Side-by-side boxplots that make a poor graph.
ggplot(data = weather, mapping = aes(x = factor(pressure), y = temp)) +
  geom_boxplot()
```



LC 2.25 (Objective 2)

(LC2.25) Boxplots provide a simple way to identify outliers. Why may outliers be easier to identify when looking at a boxplot instead of a faceted histogram?

Solution: In a histogram, the bin corresponding to where an outlier lies may not by high enough for us to see. In a boxplot, they are explicitly labeled separately.

LC 2.26 (Objective 2)

(LC2.26) Why are histograms inappropriate for visualizing categorical variables?

Solution: Histograms are for numerical variables i.e. the horizontal part of each histogram bar represents an interval, whereas for a categorical variable each bar represents only one level of the categorical variable.

LC 2.27 (Objective 2)

(LC2.27) What is the difference between histograms and barplots?

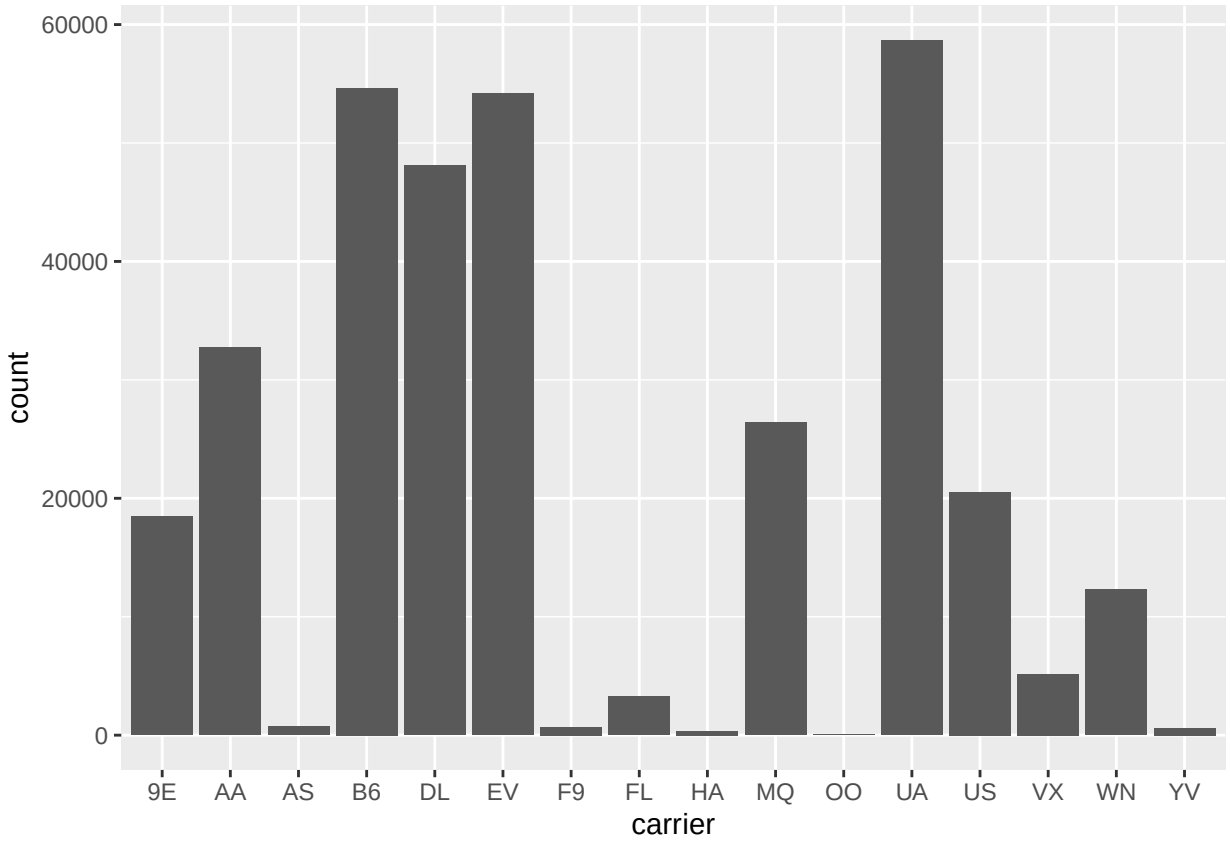
Solution: Again histograms visualize a quantatative, numerical, variable whilst a barchart visualizes a categorical, qualatative, variable.

LC 2.28 (Objective 3, 4)

(LC2.28) How many Envoy Air flights departed NYC in 2013?

Solution: Envoy Air is carrier code `MQ` and thus 26397 flights departed NYC in 2013. This can be done with a plot or table. The plot makes it difficult to know the exact value. We can use some data wrangling skills to find the answer.

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




```
flights %>%
  count(carrier) %>%
  arrange(desc(n))
```

```
## # A tibble: 16 x 2
##    carrier     n
##    <chr>   <int>
```

```
##  1 UA      58665
##  2 B6      54635
##  3 EV      54173
##  4 DL      48110
##  5 AA      32729
##  6 MQ      26397
##  7 US      20536
##  8 9E      18460
##  9 WN      12275
## 10 VX       5162
## 11 FL       3260
## 12 AS        714
## 13 F9        685
## 14 YV        601
## 15 HA        342
## 16 OO         32
```

```
#Base code
table(flights$carrier)
```

```
##
##    9E    AA    AS    B6    DL    EV    F9    FL    HA    MQ    OO    UA    US
## 18460 32729   714 54635 48110 54173   685  3260   342 26397    32 58665 20536
##    VX    WN    YV
##  5162 12275   601
```

LC 2.29 (Objective 3, 4)

(LC2.29) What was the seventh highest airline in terms of departed flights from NYC in 2013? How could we better present the table to get this answer quickly?

Solution: The answer is US, AKA U.S. Airways, with 20536 flights. However, picking out the seventh highest airline when the rows are sorted alphabetically by carrier code is difficult. This would be easier to do if the rows were sorted by number. We'll learn how to do this in the chapter on data wrangling.

```
flights %>%
  count(carrier) %>%
  arrange(desc(n))
```

```
## # A tibble: 16 x 2
##    carrier     n
##    <chr>   <int>
##  1 UA      58665
##  2 B6      54635
##  3 EV      54173
##  4 DL      48110
##  5 AA      32729
##  6 MQ      26397
##  7 US      20536
##  8 9E      18460
##  9 WN      12275
## 10 VX       5162
## 11 FL       3260
```

```
## 12 AS        714
## 13 F9        685
## 14 YV        601
## 15 HA        342
## 16 OO         32
```

LC 2.30 (Objective 2)

(LC2.30) Why should pie charts be avoided and replaced by barplots?

Solution: In our **opinion**, comparisons using horizontal lines are easier than comparing angles and areas of circles.

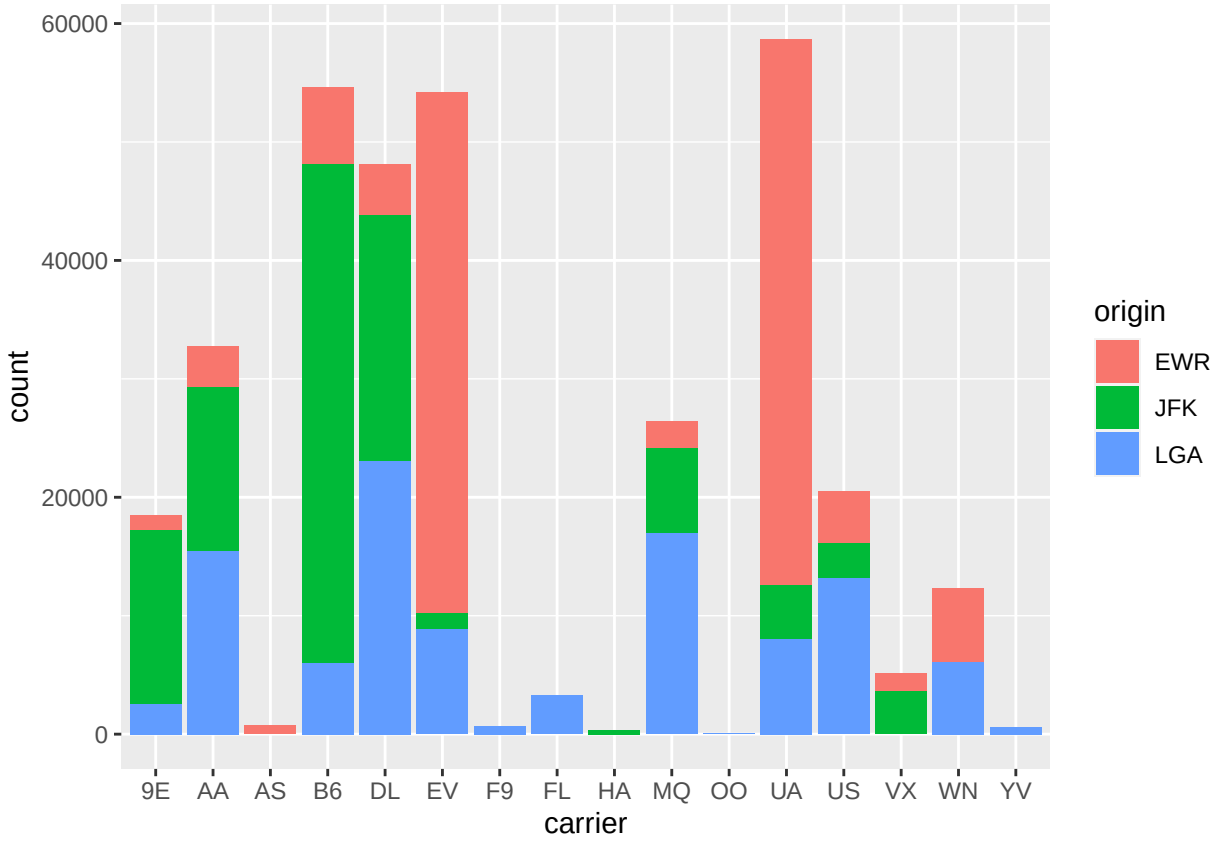
LC 2.31 (Objective 2)

(LC2.31) What is your opinion as to why pie charts continue to be used?

Solution: In our **opinion**, pie charts are generally considered as a poorer method for communicating data than bar charts. People's brains are not as good at comparing the size of angles because there is no scale, and in comparison, it is much easier to compare the heights of bars in a bar charts. However, pie charts have been used historical and it is difficult to overcome this momentum. Also, many software packages, such as Excel, make it is to generate pie charts.

LC 2.32 (Objective 3)

```
# Code needed for this problem
ggplot(data = flights, mapping = aes(x = carrier, fill = origin)) +
  geom_bar()
```

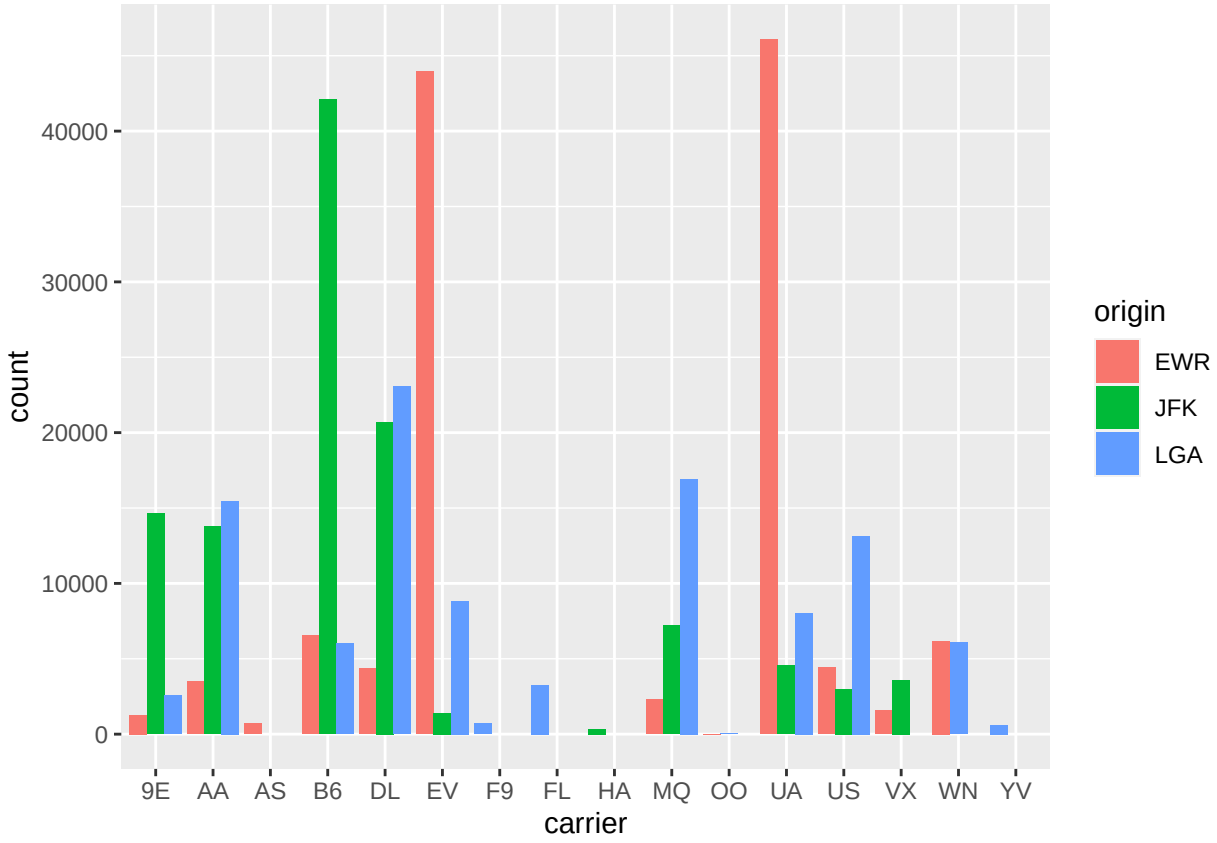


(LC2.32) What kinds of questions are not easily answered by looking at the above figure?

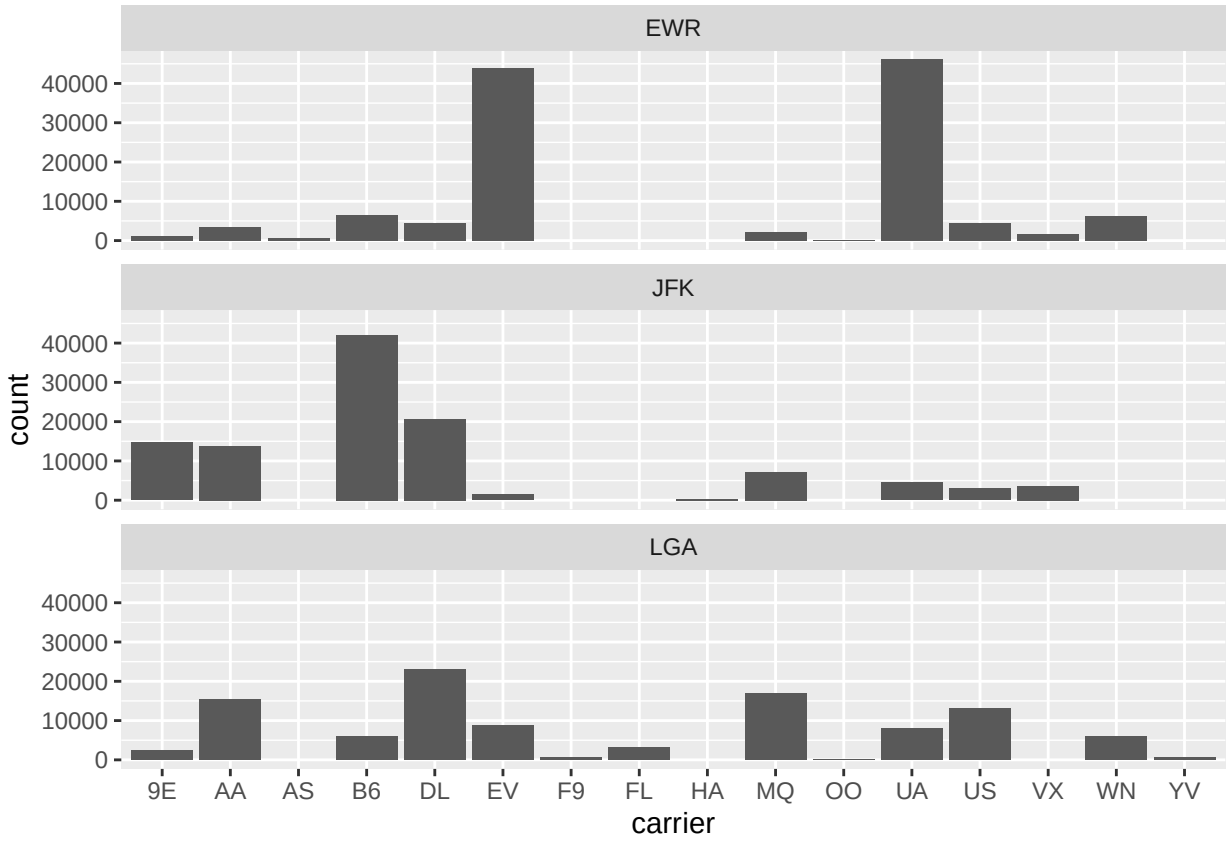
Solution: Because the red, green, and blue bars don't all start at 0 (only light blue does), it makes comparing counts hard. Notice we can generate percentages within each carrier by using `position()` in the geom.

Here are a couple of different ways to do the plots.

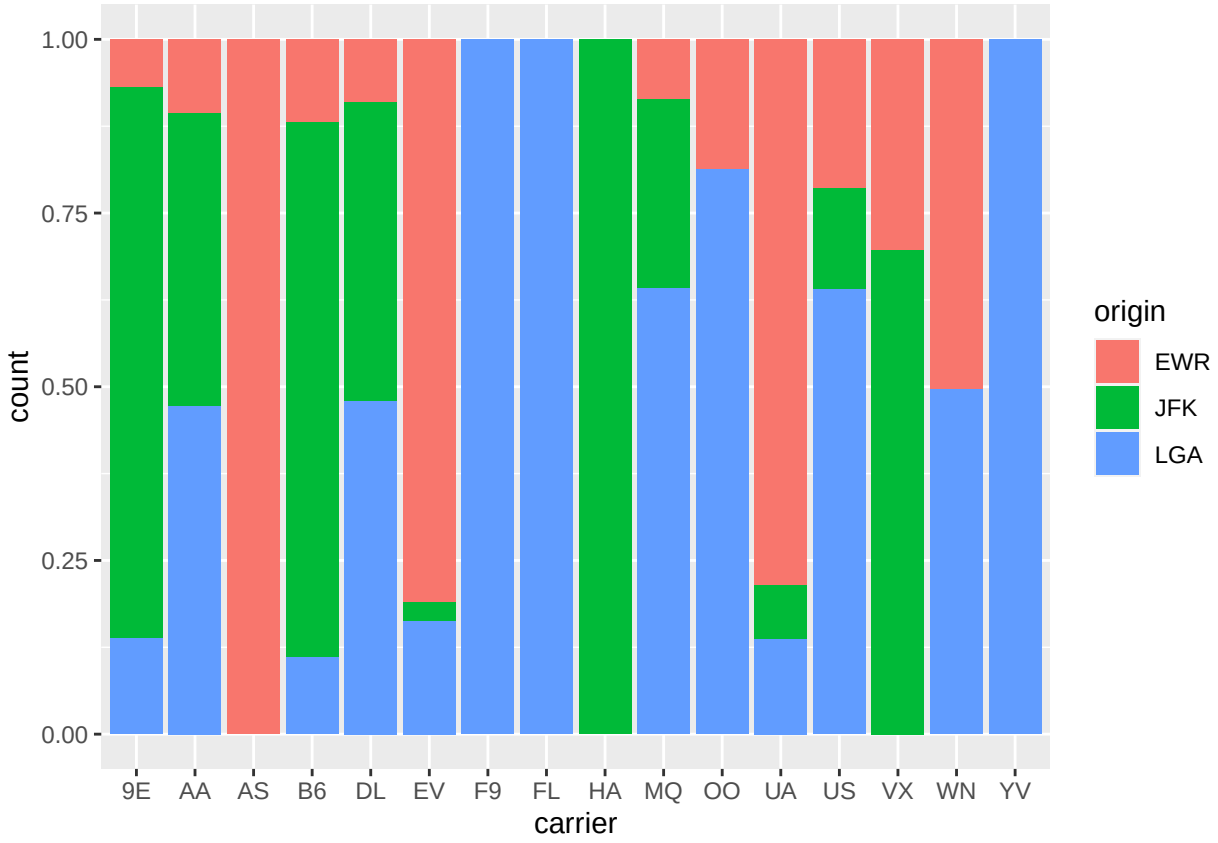
```
ggplot(data = flights, mapping = aes(x = carrier, fill = origin)) +
  geom_bar(position = position_dodge(preserve = "single"))
```




```
ggplot(data = flights, mapping = aes(x = carrier)) +
  geom_bar() +
  facet_wrap(~ origin, ncol = 1)
```




```
# Percentage
ggplot(data = flights, mapping = aes(x = carrier, fill = origin)) +
  geom_bar(position='fill')
```



LC 2.33 (Objective 3)

(LC2.33) What can you say, if anything, about the relationship between airline and airport in NYC in 2013 in regards to the number of departing flights?

Solution: The different airlines prefer different airports. For example, United is mostly a Newark carrier and JetBlue is a JFK carrier. If airlines didn't prefer airports, each color would be roughly one third of each bar in the original plot.

LC 2.34 (Objective 2)

(LC2.34) Why might the side-by-side (AKA dodged) barplot be preferable to a stacked barplot in this case?

Solution: We can easily compare the different airports for a given carrier using a single comparison line i.e. things are lined up.

LC 2.35 (Objective 2)

(LC2.35) What are the disadvantages of using a side-by-side (AKA dodged) barplot, in general?

Solution: It is hard to get totals for each airline.

LC 2.36 (Objective 2)

(LC2.36) Why is the faceted barplot preferred to the side-by-side and stacked barplots in this case?

Solution: Not that different than using side-by-side; depends on how you want to organize your presentation. The side-by-side makes it easier to compare airports with a carriers versus the faceted plot makes it easier to compare carriers within airports.

LC 2.37 (Objective 2)

(LC2.37) What information about the different carriers at different airports is more easily seen in the faceted barplot?

Solution: Now we can also compare the different carriers **within** a particular airport easily too. For example, we can read off who the top carrier for each airport is easily using a single horizontal line.

Documenting software

- File creation date: 2022-06-04
- R version 4.1.3 (2022-03-10)
- `ggplot2` package version: 3.3.6
- `dplyr` package version: 1.0.9
- `nycflights13` package version: 1.0.2