Data Visualization

Jason Willwerscheid

2022-09-26

Introduction

- One use of data wrangling is to find **summary statistics** like minimum, maximum, mean, and median.

Introduction

- One use of data wrangling is to find **summary statistics** like minimum, maximum, mean, and median.
- A useful non-tidyverse way of getting all of these statistics at once is the `summary()` function.

Introduction

```
summary(bluebikes$tripduration / 60)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     1.02     7.78    13.17    28.99    22.33 43928.15
```

Introduction

- But if we want to know about a variable's entire **distribution**, then data visualization can come in handy.

Introduction

- But if we want to know about a variable's entire **distribution**, then data visualization can come in handy.
- Data visualization is also useful to explore *relationships* among variables.

Introduction

- But if we want to know about a variable's entire **distribution**, then data visualization can come in handy.
- Data visualization is also useful to explore *relationships* among variables.
- Different data types are better suited to different types of visualizations.

Introduction

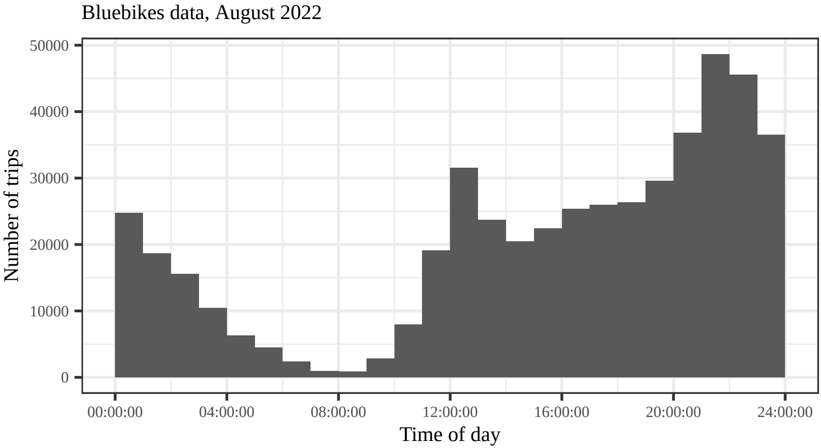
- But if we want to know about a variable's entire **distribution**, then data visualization can come in handy.
- Data visualization is also useful to explore *relationships* among variables.
- Different data types are better suited to different types of visualizations.
- Which types of visualization are you already familiar with?

Data Visualization: One Variable

For visualizing distributions of numeric variables, **histograms** are the most common choice.

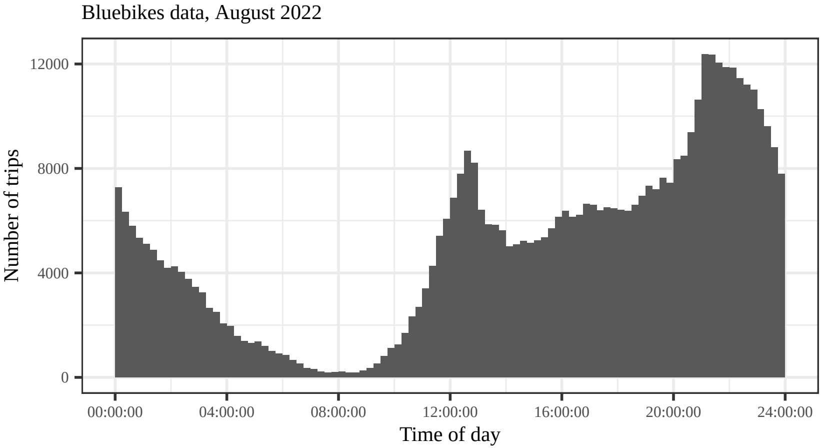
Data Visualization: One Variable

Histograms are easily understandable and can be plotted by hand. But you need to be careful about how to choose the number of **bins**, as different choices can give very different impressions. Compare:



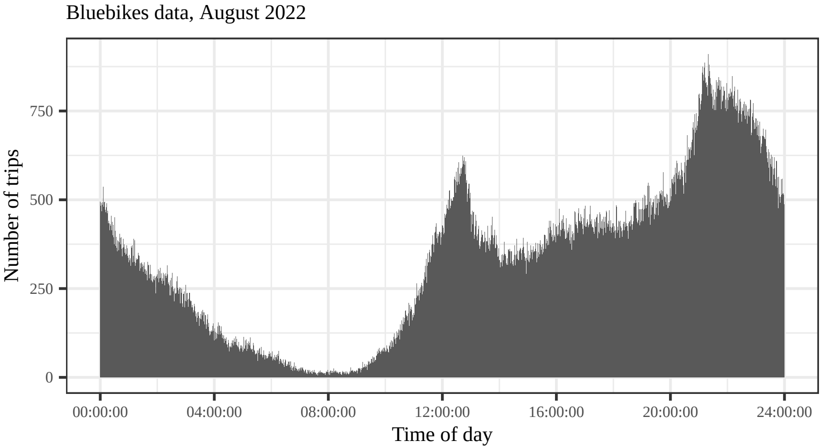
Data Visualization: One Variable

Histograms are easily understandable and can be plotted by hand, but you need to be careful about choosing the number of **bins**, as different choices can give very different impressions. Compare:



Data Visualization: One Variable

Histograms are easily understandable and can be plotted by hand, but you need to be careful about choosing the number of **bins**, as different choices can give very different impressions. Compare:

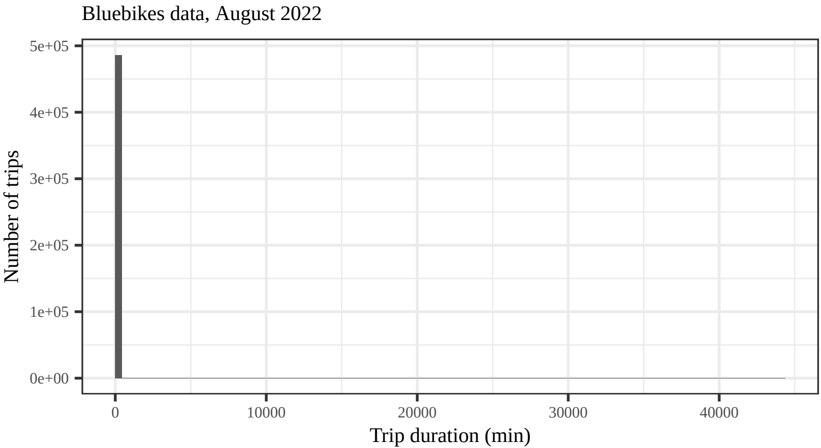


Data Visualization: One Variable

It's best to experiment with a number of different bin sizes and choose whichever one looks "smooth" but still captures important features of the distribution.

Data Visualization: One Variable

Here is another example from the Bluebikes dataset. What is the problem?



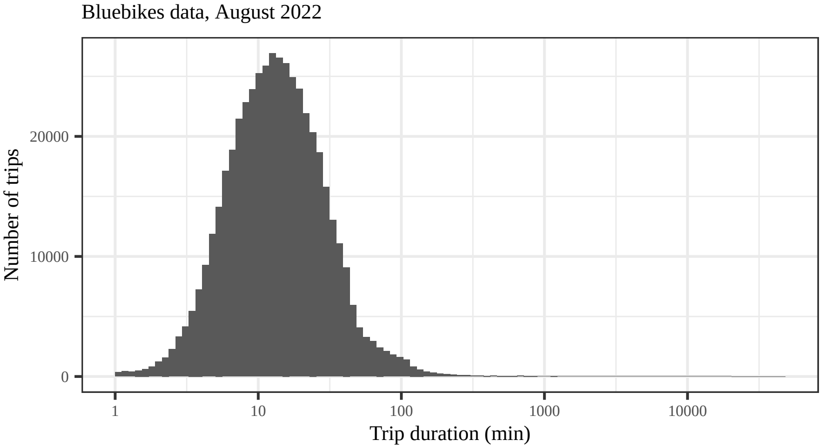
Data Visualization: One Variable

Here is another example from the Bluebikes dataset. What is the problem?

```
##    tripduration_days
## 1           30.50566
## 2           28.73723
## 3           27.89666
## 4           27.45263
## 5           27.20837
## 6           27.17883
## 7           26.63780
## 8           26.31271
## 9           26.03485
## 10          25.81250
```

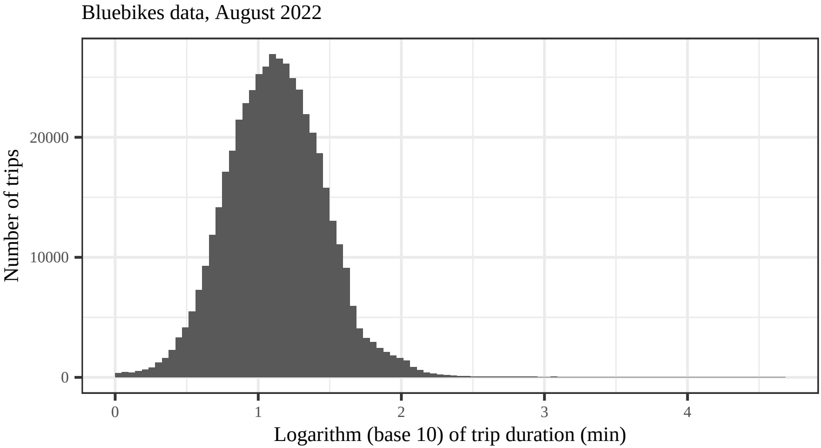
Data Visualization: One Variable

In cases like these, it can be helpful to use a **logarithmic scale**:



Data Visualization: One Variable

Using a logarithmic scale is essentially equivalent to **log transforming** the data; i.e., replacing each of the values with their logarithm:



Producing readable plots

Here is a good time to point out some features that make for good, readable figures:

- Always label your axes and include units (sec, min, hrs, etc.)

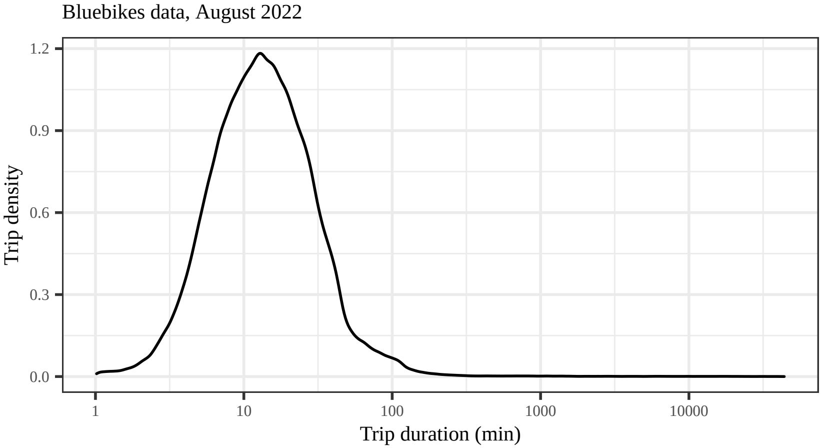
Producing readable plots

Here is a good time to point out some features that make for good, readable figures:

- Always label your axes and include units (sec, min, hrs, etc.)
- Always include a title that gives the context

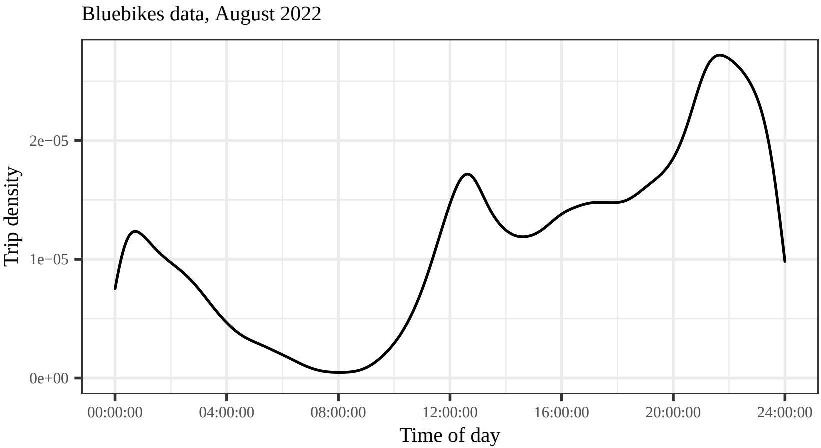
Data Visualization: One Variable

Another option for numeric variables is a **density plot**, which is more mathematically complex and cannot be plotted by hand, but usually gives “smooth” results without tinkering. Which do you prefer?



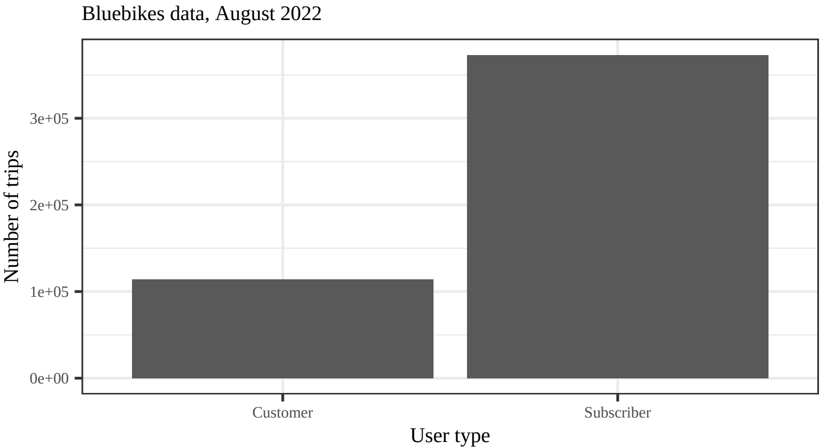
Data Visualization: One Variable

Density plots are not totally "automatic," however. Can you find the problem here?



Data Visualization: One Variable

For discrete data such as factors, histograms are not an option. One possibility is a bar plot:



Data Visualization: One Variable

Do you prefer the bar plot or a simple table?

```
bluebikes %>%
  group_by(usertype) %>%
  summarize(count = n())
```

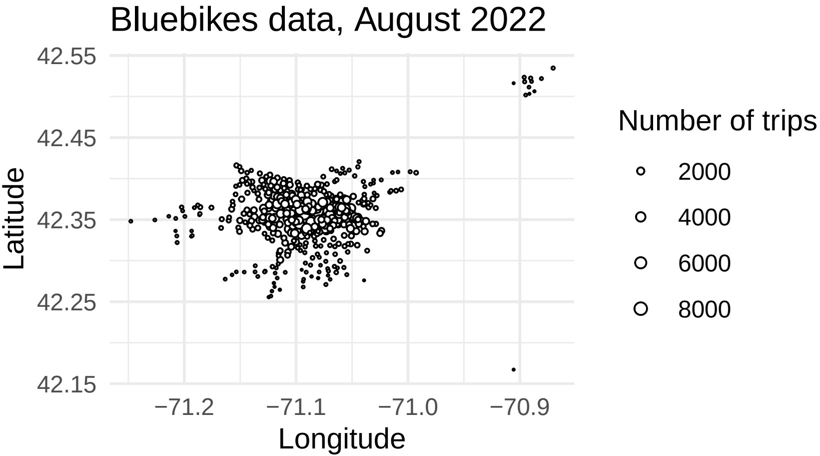
```
## # A tibble: 2 x 2
##   usertype    count
##   <chr>       <int>
## 1 Customer   114136
## 2 Subscriber 373065
```

Data Visualization: One Variable

There are 436 stations, so a bar plot would look terrible. Any ideas for how to visualize the number of trips per station?

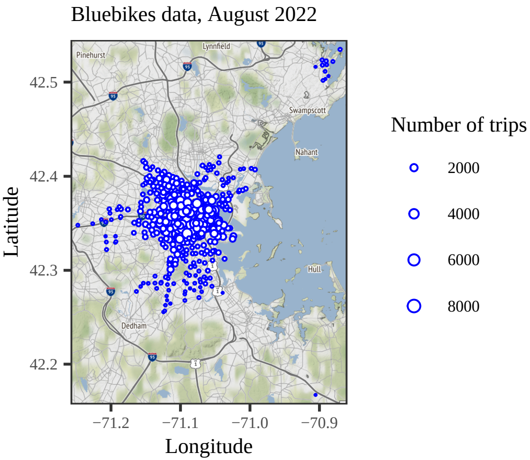
Data Visualization: One Variable

We have geographic coordinates; we might as well use them...



Data Visualization: One Variable

Even better:



Data Visualization: Two Variables

We've looked at continuous (numeric and datetime) variables and discrete variables (factors, but also logical and some integer variables). So we have three possibilities:

- Relate two continuous variables

Data Visualization: Two Variables

We've looked at continuous (numeric and datetime) variables and discrete variables (factors, but also logical and some integer variables). So we have three possibilities:

- Relate two continuous variables
- Relate a continuous variable to a discrete variable

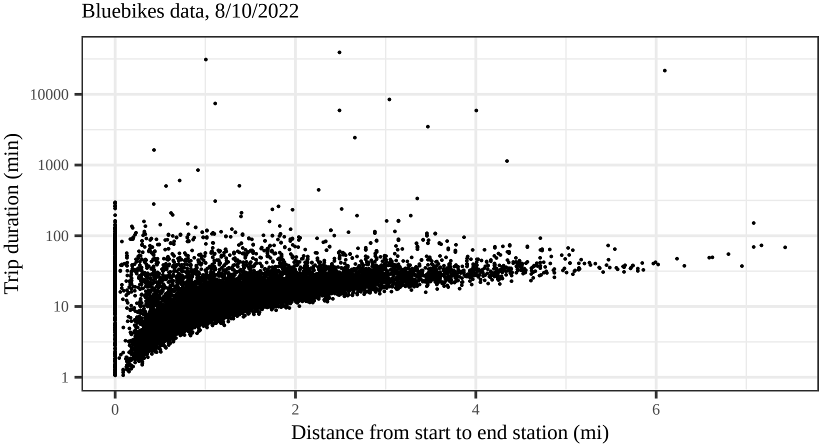
Data Visualization: Two Variables

We've looked at continuous (numeric and datetime) variables and discrete variables (factors, but also logical and some integer variables). So we have three possibilities:

- Relate two continuous variables
- Relate a continuous variable to a discrete variable
- Relate two discrete variables

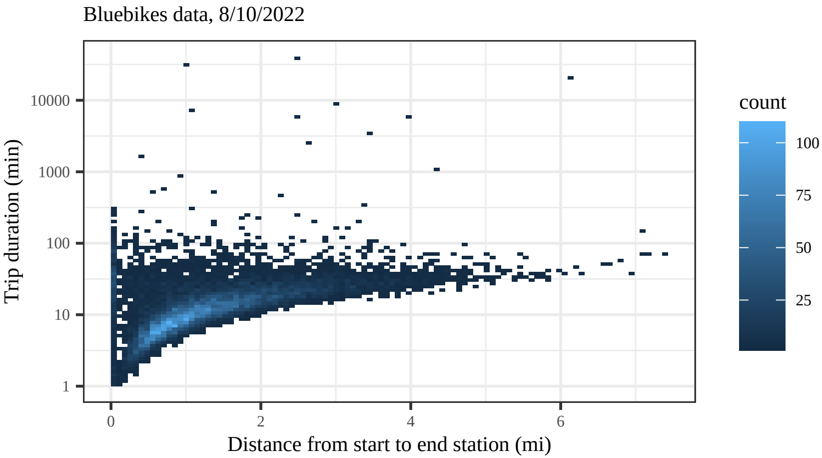
Data Visualization: Two Variables

To compare two numeric variables, a **scatterplot** is often your best choice. If there is a dependent and independent variable (i.e., cause and effect), the convention is to put the independent variable on the x-axis:



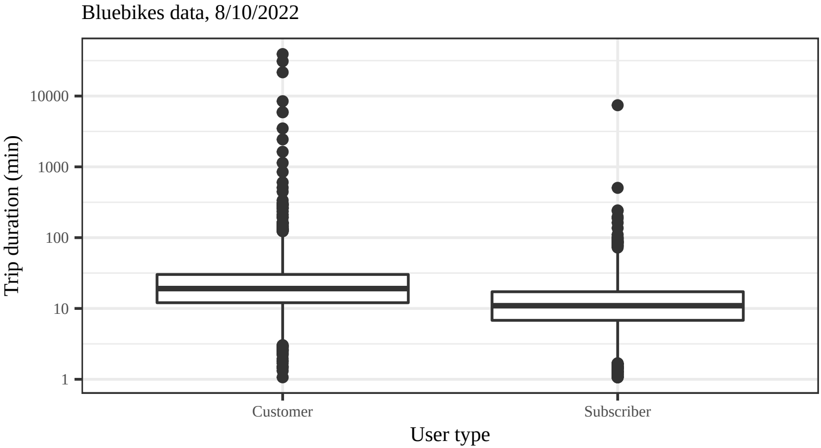
Data Visualization: Two Variables

With large datasets, we need to be wary of **overplotting**. The previous example looks better plotted as a **two-dimensional histogram**:



Data Visualization: Two Variables

A **boxplot** is a traditional way to compare a numeric variable across several levels of a factor variable:



Data Visualization: Two Variables

A boxplot contains a lot of information:

- The thick middle line is the median.

Data Visualization: Two Variables

A boxplot contains a lot of information:

- The thick middle line is the median.
- The shaded area is the data that falls between the first and third quartiles. This is known as the **interquartile range** (IQR).

Data Visualization: Two Variables

A boxplot contains a lot of information:

- The thick middle line is the median.
- The shaded area is the data that falls between the first and third quartiles. This is known as the **interquartile range** (IQR).
- The upper and lower ticks or "whiskers" give the maximum and minimum values that lie within 1.5 IQR of the first and third quartiles.

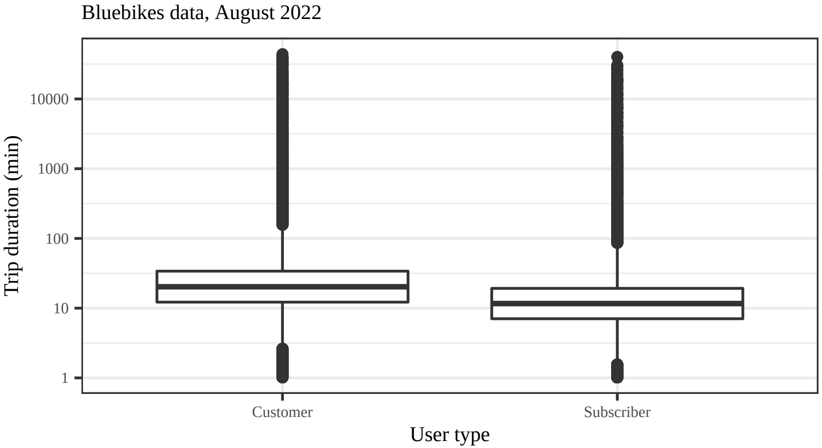
Data Visualization: Two Variables

A boxplot contains a lot of information:

- The thick middle line is the median.
- The shaded area is the data that falls between the first and third quartiles. This is known as the **interquartile range** (IQR).
- The upper and lower ticks or "whiskers" give the maximum and minimum values that lie within 1.5 IQR of the first and third quartiles.
- All other points are **outliers** (lying beyond 1.5 IQR of the first and third quartiles).

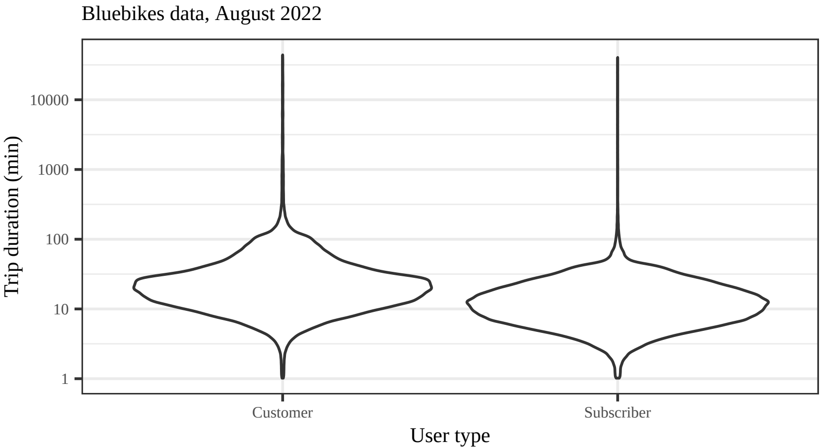
Data Visualization: Two Variables

Boxplots were designed to be drawn by hand, and are not very well suited to huge datasets. In particular, the definition of “outlier” seems pretty arbitrary:



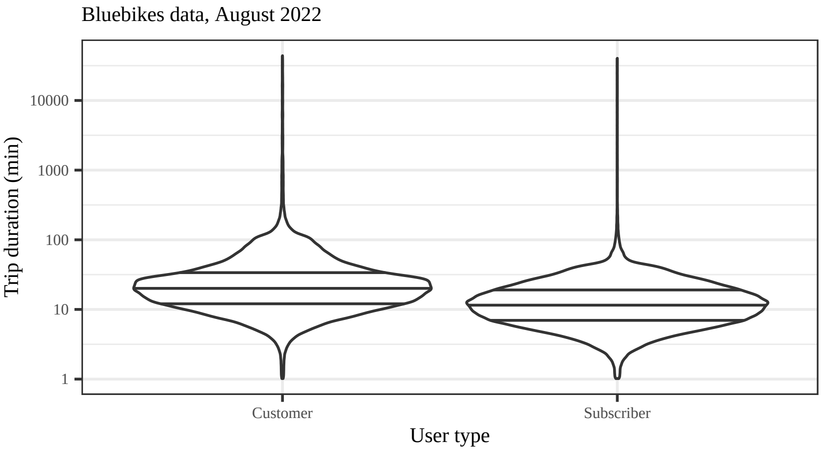
Data Visualization: Two Variables

The **violin plot** is a more modern invention. Unlike the boxplot, it cannot be drawn by hand and requires sophisticated computations. However, it does a better job of showing the overall shapes of the distributions:



Data Visualization: Two Variables

We can show quantiles as well:

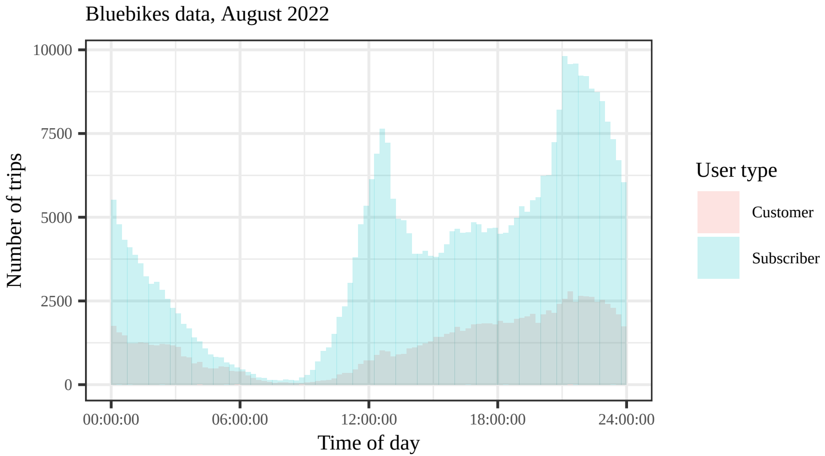


Data Visualization: Two Variables

What if we're interested in the relationship between user type and time of day? Box plots and violin plots don't make much sense (why?). Any ideas?

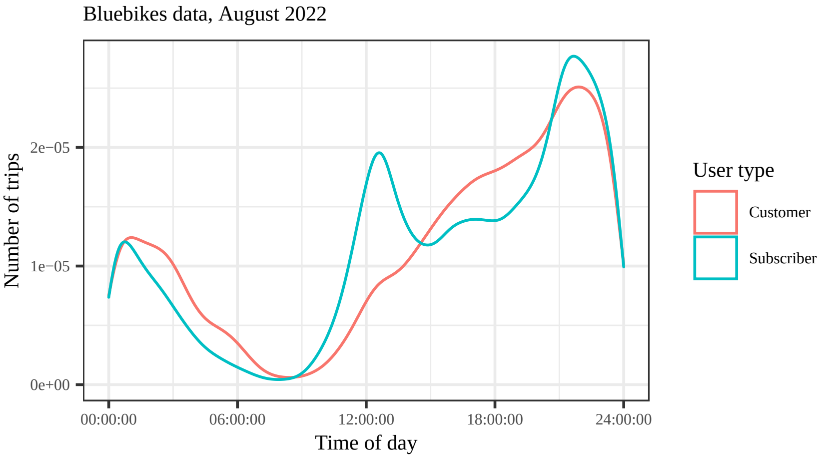
Data Visualization: Two Variables

We can use overlapping histograms:



Data Visualization: Two Variables

Or overlapping density plots:



Data Visualization: Two Variables

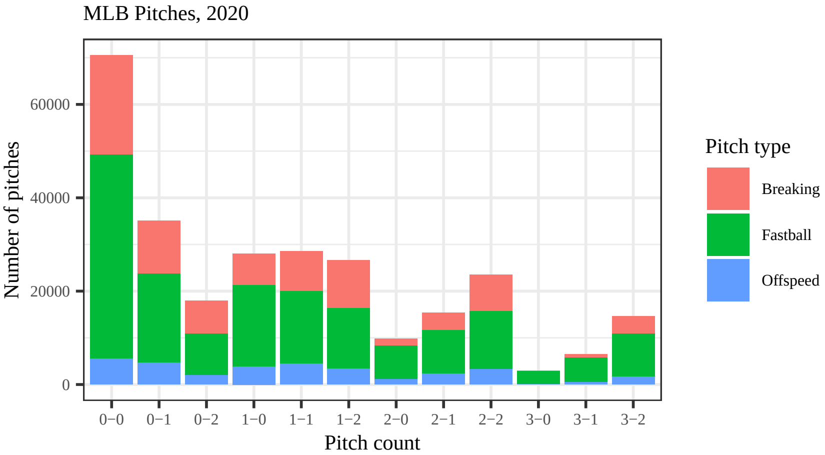
Showing relationships between two discrete variables that are both factors might be trickiest. Sometimes it's best to just go with a table:

```
## `summarise()` has grouped output by 'city'. You can over
## `.groups` argument.
```

```
## # A tibble: 2 x 3
## # Groups:   city [2]
##   city   Customer Subscriber
##   <chr>     <int>      <int>
## 1 Boston   113582     372790
## 2 Salem       554        275
```

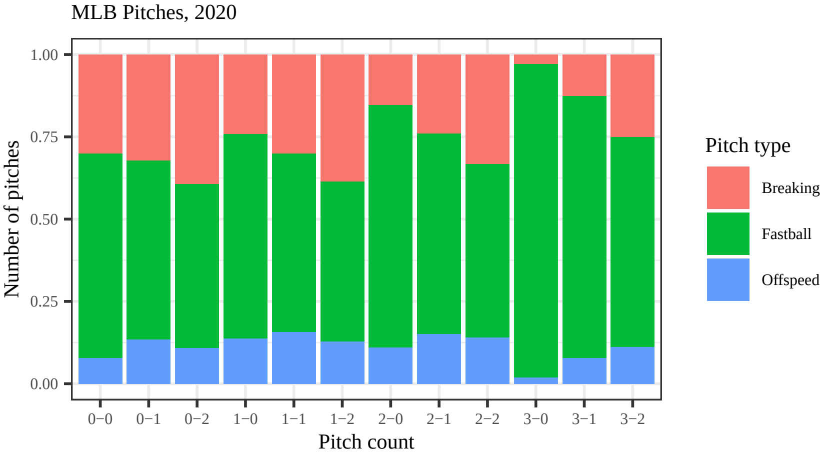
Data Visualization: Two Variables

A stacked bar chart can also be a good choice.



Data Visualization: Two Variables

If we're less interested in overall numbers, we can show proportions instead. Any further suggestions for improving this figure?



Data Visualization: Two Variables

We can split up balls and strikes and show three variables in a **mosaic plot**: