Lecture 3

Statistical graphs

The type of statistical graph employed to display data depends on the kind of data to be plotted. Categorical or qualitative data is graphed very differently than numerical or quantitative data. One very common error students make is plotting data in the wrong kind of graph. Some graphing programs, like Excel, will allow you to make all kinds of meaningless charts. So, it's important that before you graph anything, you take a moment to consider what kind of data you have and which type of plot would be the best one to display it with.

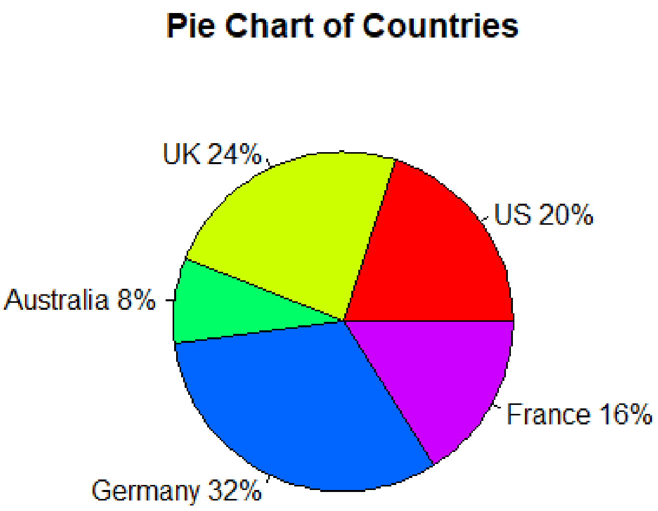
Qualitative data

There are two main types of graphs you can use to plot categorical data:

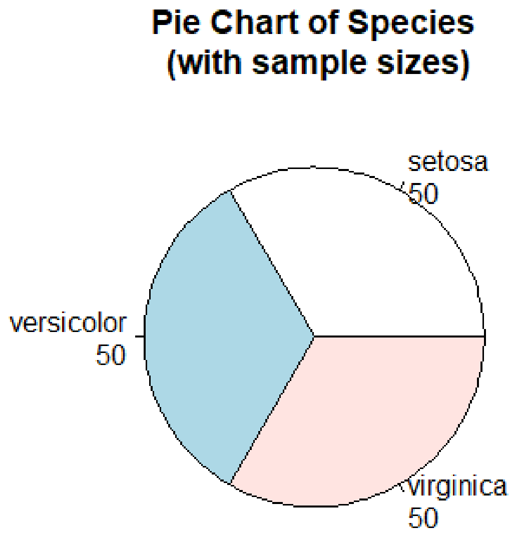
- Bar graphs
- Pie charts

There is a variation on the bar graph, a Pareto chart, that we are also going to discuss.
Both types of graphs are generally built from frequency tables. You should first summarize your data into a table before thinking about plotting. This avoids many common errors. Some programs can summarize for you and plot from raw data, but when getting started, you should decouple these steps to ensure that you are not just getting something meaningless.

The pie chart is the most restrictive type of graph here. It can plot only one variable at a time. It is meant to display relative percentages that meaningfully add up to a total (100%). It should not be used to display something just because it is a percentage. Pie charts are also difficult to read if there are too many categories. A good rule of thumb is no more than 7 categories.



This particular pie chart is fictional data and so the title doesn't tell us what the data represents. This is otherwise a well-designed pie chart with a title, and percentages on each slice. The labels can go next to the slices or in a legend.



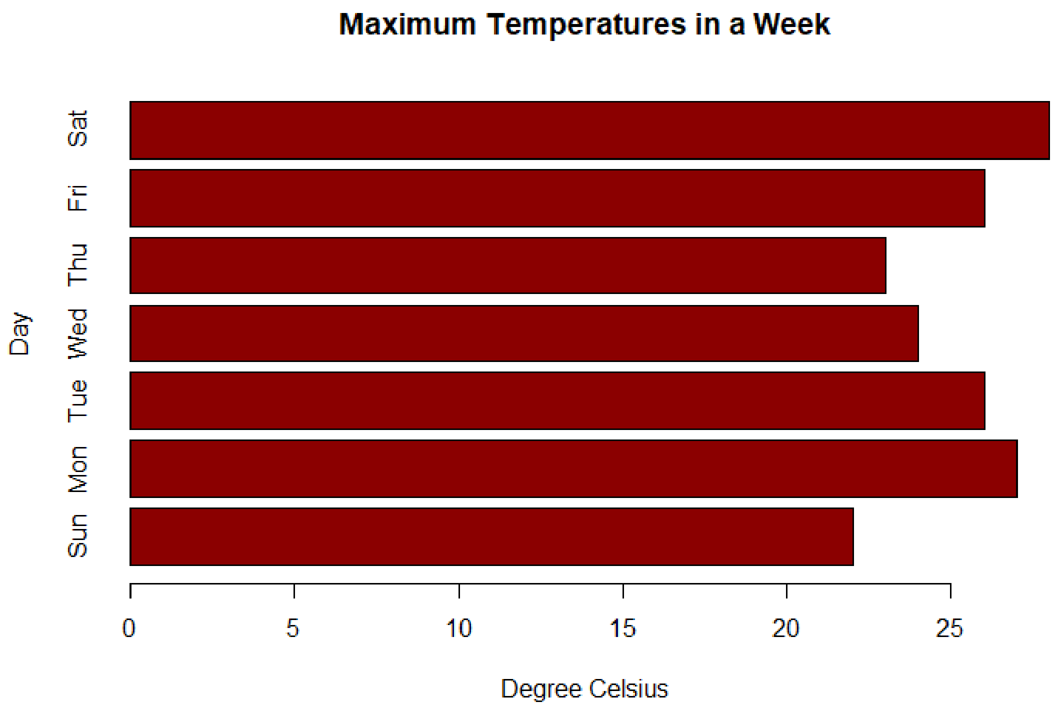
It’s also possible to create a pie chart that displays values (here sample sizes) rather than percentages. This one is built from the iris dataset built into R.

These pie charts were built in base R and have limited functionality since they tend to be dis-preferred by R designers. However, there are other packages that improve functionality similar to what Excel provides, including 3D options. But, it is generally preferred that you avoid 3D effects in graphs. While they look pretty, they can be misleading. Even pie charts can be problematic because we are better at judging height than area (and even worse at volume), so bar graphs are easier to read.

Bar graphs can be used both to compare frequencies (counts or percentages) or other summary statistics within categories (such as average income within various job categories). A variation on the simple bar graph is a Pareto chart, where one variable is plotted with the bars sorted by their frequency (usually largest to smallest).

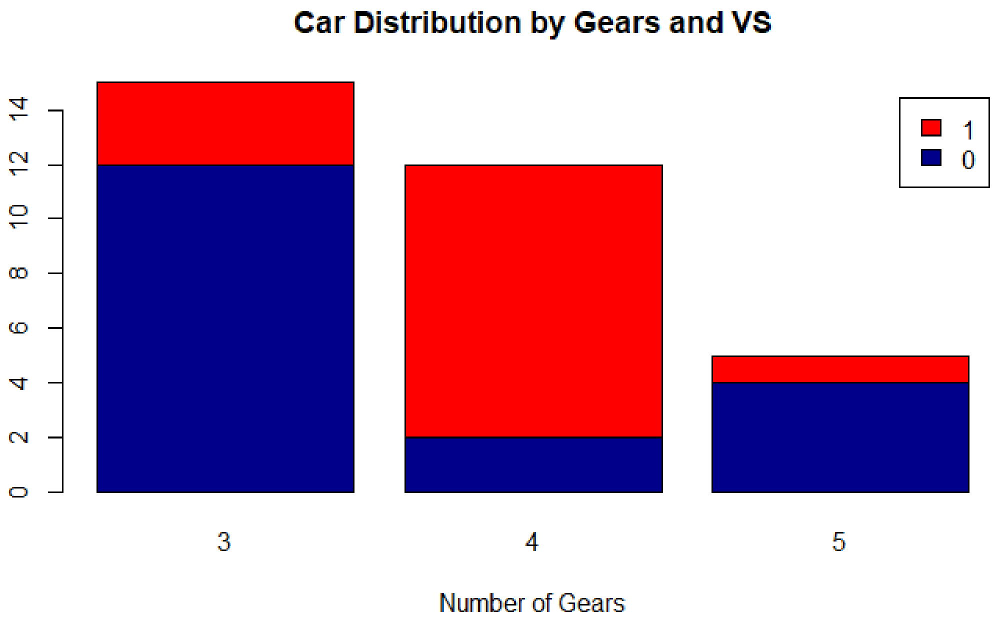
Bar graphs have many variations. They can be plotted vertically (standard) or horizontally. They can plot more than one variable (clustered or stacked). And using frequencies or percentages, and when plotting more than one variable, relative frequencies (percent stacked) can give each bar the effect of having something like a comparative pie chart.

A horizontally orientated bar graph is shown below. Notice this is a well-labeled graph with axis titles, and a descriptive title. It’s not displaying only counts here, but another variable separated by the categories.

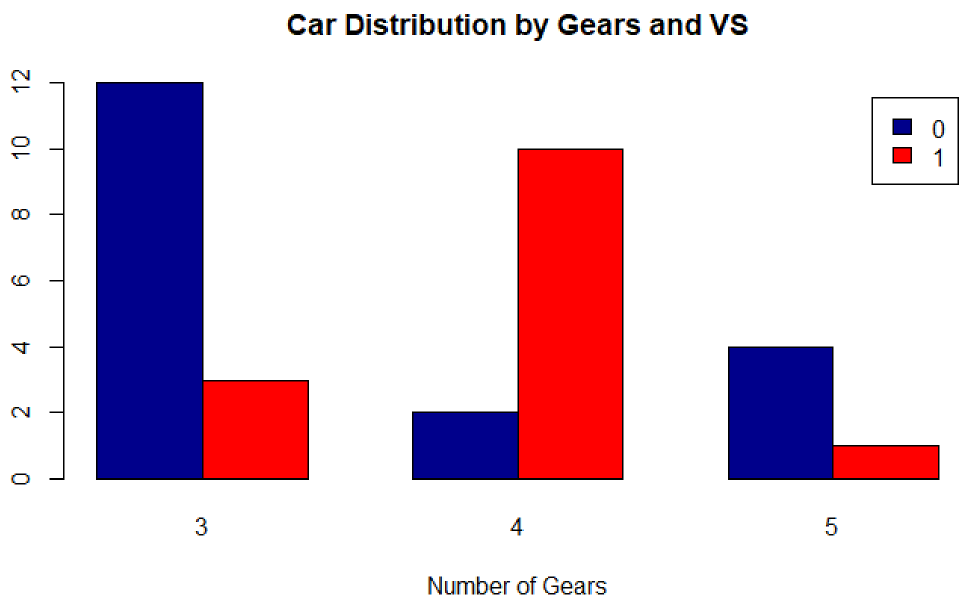


(when saving, be careful with the size of the plot to make sure everything is displaying properly).

We can stack bars to add another dimension to our data (to compare two categorical variables).

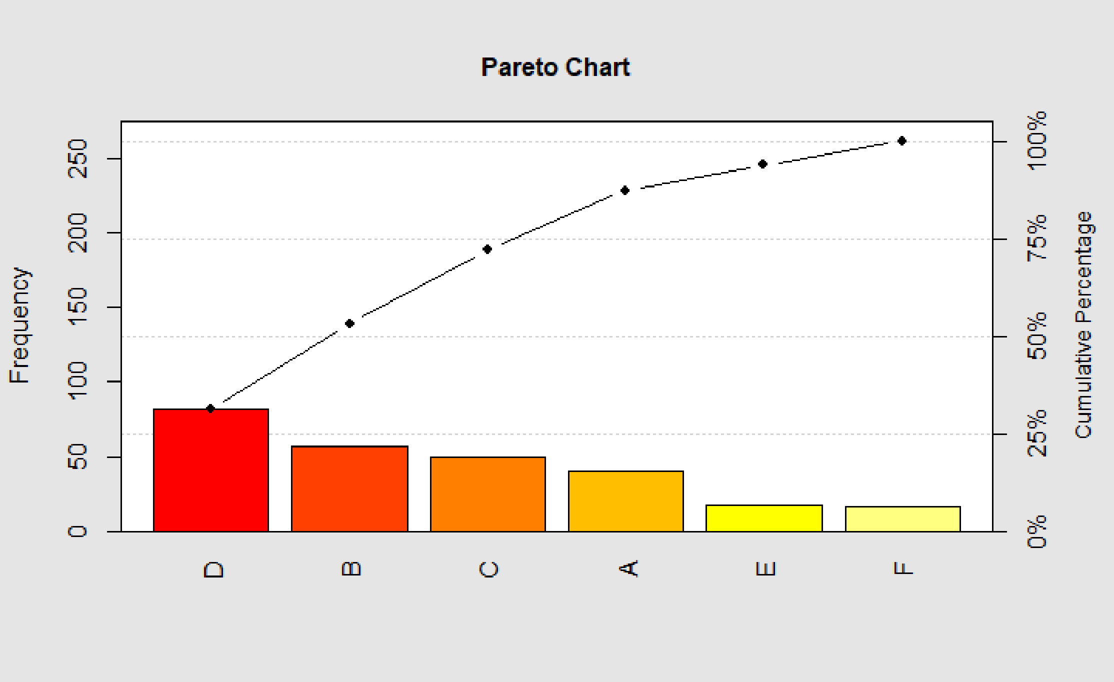


Or we can plot the same data as groups with clusters of bars.



A Pareto chart generally speaking is an ordered bar graph, by order of the heights of the bars. While it is possible to see Pareto charts with just the ordered bars, it is also quite comment to see a cumulative frequency line plotted over top of it, with the percentages marked on the axis on the right side. This is a type of combinate chart with the ogive graph we'll discuss below.

This data is fictional, thus the generic graph title and missing axis title on the horizontal axis.



Quantitative data

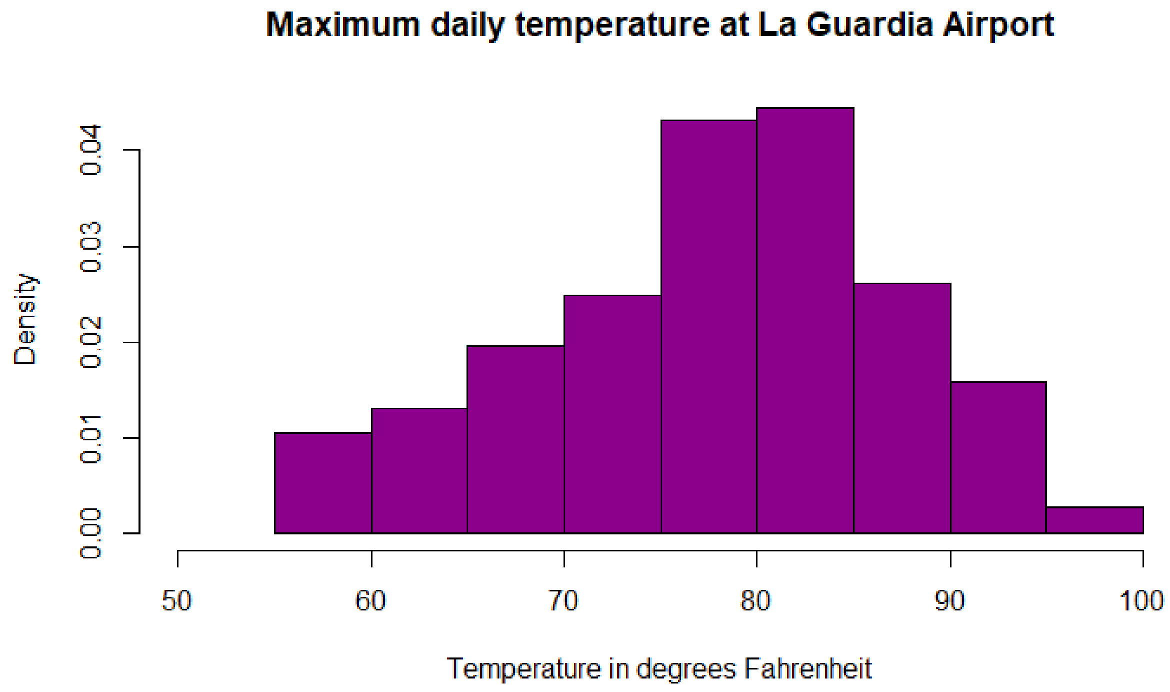
Not all numerical data in a dataset is meaningful and worth plotting. Often datasets replace names of customers or respondents with numbers. This number is not a measurement and it should never be plotted. Think of it as an index. In R, you may want to consider removing it from the dataset altogether before proceeding.

If you are not sure what a particular number in a dataset means, consider going to the data dictionary or other information you may have on the data before plotting it to make sure it is not a dummy variable. Interpreting your data correctly before plotting will save a lot of headaches.

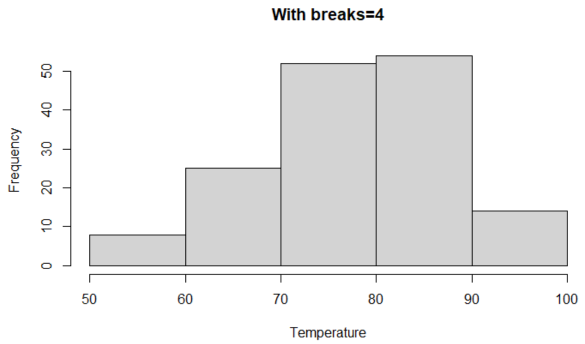
Histograms are typically built (when done by hand) from frequency tables like bar graphs. The numerical data is binned into groups and then the histogram is built from that now categorized data. Technology can do that categorizing for us now, but it is going on behind the scenes. Discrete data can sometimes bin itself (with each discrete value being one bin), but this depends on how many values the data can take on. For example, you might be able to do this for the number of children in a family, but not for ages.

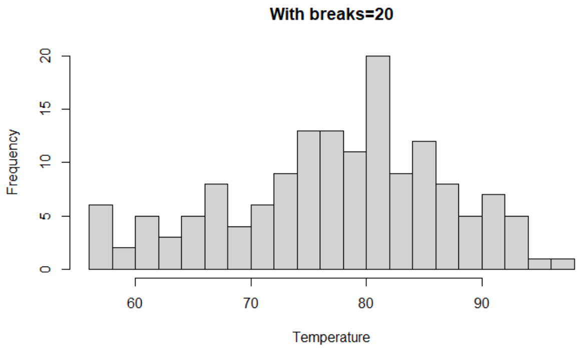
The rule of thumb for histograms is very general: 5-20 bins. The number of bins depends on the amount of data to be plotted, but you should experiment with the number of bins because the shape of the graph will change subtly. In general, though, too few bins will not do a good job showing the shape of the distribution, and too many may produce other problems like empty categories, a pancake graph with little or no structure.

To create by hand, you would find the range of the data and then divide by the number of bins you want. Round up a little if you want to keep that number of bins (if you round up too much, you may lose bins), round down if you are okay adding an extra bin. This number is called the bin width. Start at or near the minimum and begin creating your classes. Each new bin starts by adding another bin width to the previous class. Count the number of observations in each bin. Then make a bar graph of the data.



Changing the number of bins can make the graph look a little differently.



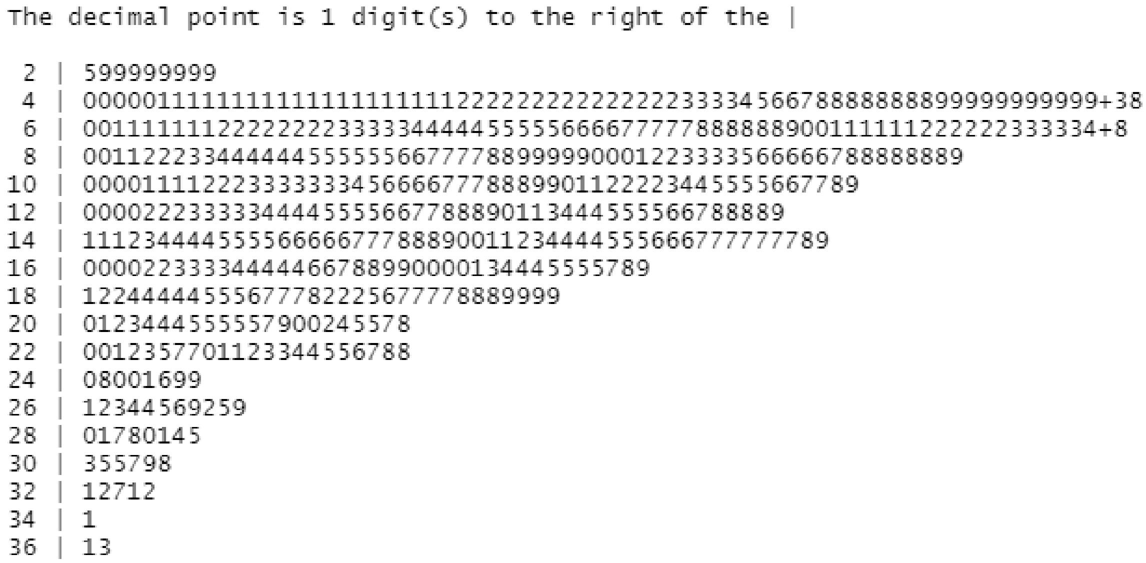


The first graph above has 4 breaks (5 bins) and does look okay without distorting the data too much. The second graph on the right has 20 breaks (21 bins) and it looks like this is too many bins. There is too much noise in the bin heights. Fewer bins would be less noisy.

Shape of the distributions.
Numerical plots like histograms, and as we'll see later, density plots, display data it a way that lets us talk about the way data is distributed. Some other graph types as well. Can be distributed in a couple of broad categories most of the time. The data might be symmetric (typically a bell-shape). If it has a long left tail, we call it left or negatively skewed. If it has a long right tail, we call it right or positively skewed. If it has no peaks, we can call it uniform. If the data has two peaks we can call it bimodal. All of these descriptions are approximate. Real data will never look perfectly symmetric or perfectly uniform.

The graph of temperature above is a little left-skewed.

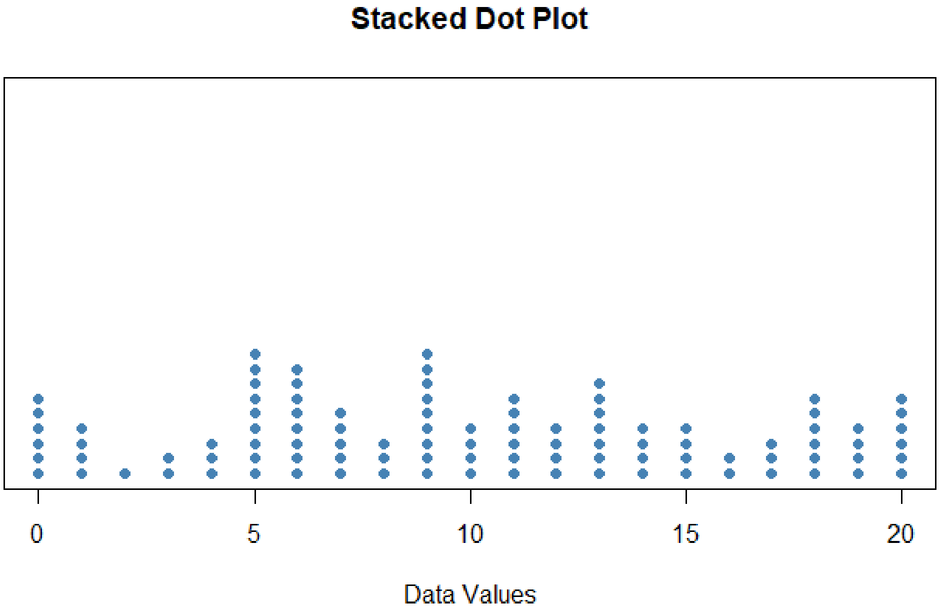
A stem plot, or a stem-and-leaf plot, is a graph that is similar to a histogram but which displays the original data values. How one bins the data is more limited, but you do get a chance to see the shape of the distribution. R can produce a stem plot as a text plot, not as an image (at least in base R). The following is a screenshot of a stem plot from chick weight data in R. This is a large data set and an atypical example for a stemplot. You can see that the bins 4 and 6 are actually too big to display entirely. This graph has a strong right (positive) skew. Stem plots should have a legend (as seen in the top) to show how to interpret the numbers in the graph.



```
 The decimal point is 1 digit(s) to the right of the |

   2 | 599999999
   4 | 00000111111111111111111112222222222222223333456678888888899999999999+38
   6 | 0011111112222222233333444445555566667777788888890011111122222233333 4+8
   8 | 00112223344444455555566777788999990001223333566666788888889
  10 | 0000111122233333334566667778889901122223445555667789
  12 | 00002223333344445555667788890113444555566788889
  14 | 11123444455556666677788890011234444555666777777789
  16 | 00002233334444466788990000134445555789
  18 | 12244444555677782225677778889999
  20 | 0123444555557900245578
  22 | 0012357701123344556788
  24 | 08001699
  26 | 12344569259
  28 | 01780145
  30 | 355798
  32 | 12712
  34 | 1
  36 | 13
```

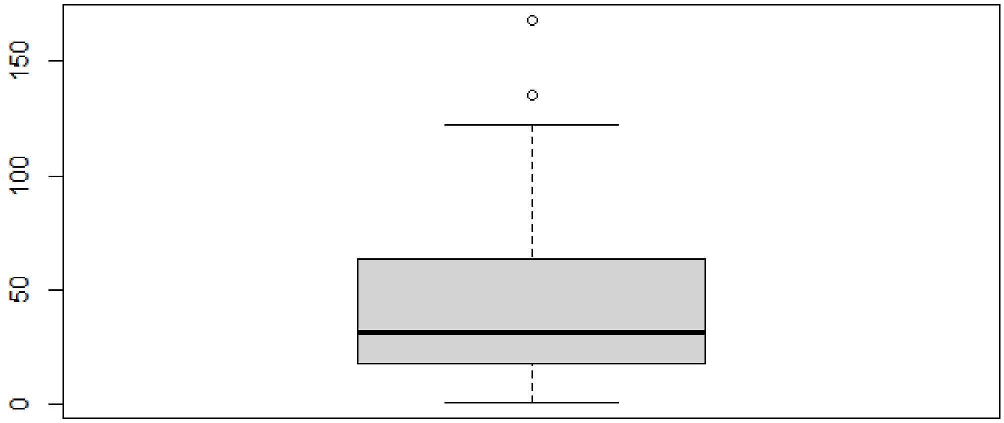
A dot plot is another graph that is similar to a histogram, but uses (generally) one dot per observation. Sometimes this type of display is also used for categorical data. It's similar to a histogram, but better for small data sets.



The data in the graph above is simulated data, so the labels are more general than for genuine data.

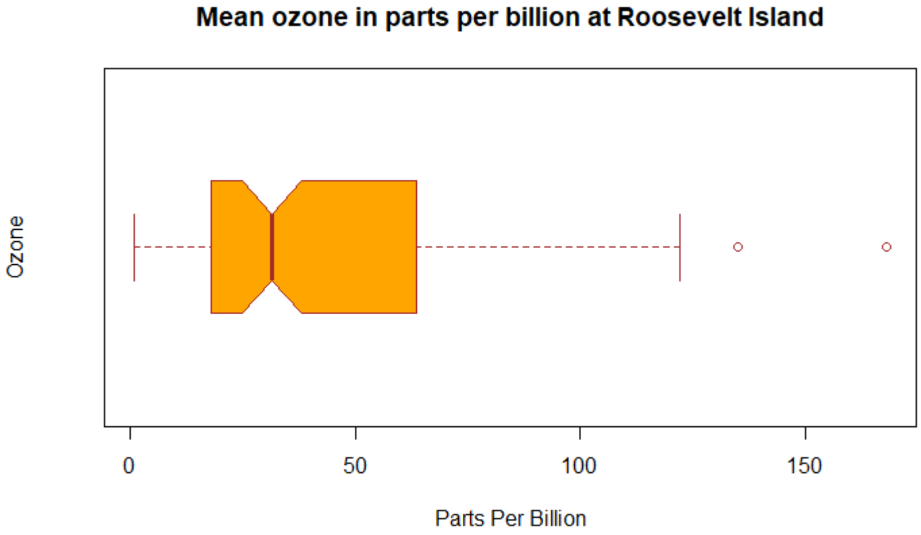
When searching for how to create dot plots, take care since there are other kinds of dot plots that can be used in place of boxplots (swarm plots), or for comparing different observations of grouped data, and other examples.

A Box plot (or a box-and-whisker plot) is a quick and easy method to make a graph of data that can be done by hand without too much trouble. The simplest versions can be built from the five-number summary: min, Q1, median, Q3, max. More sophisticated versions consider the fences and plot extreme values. Some versions also mark the mean on the graph (to compare to the median). The central box in a box plot is the middle 50% and goes from Q1 to Q3.



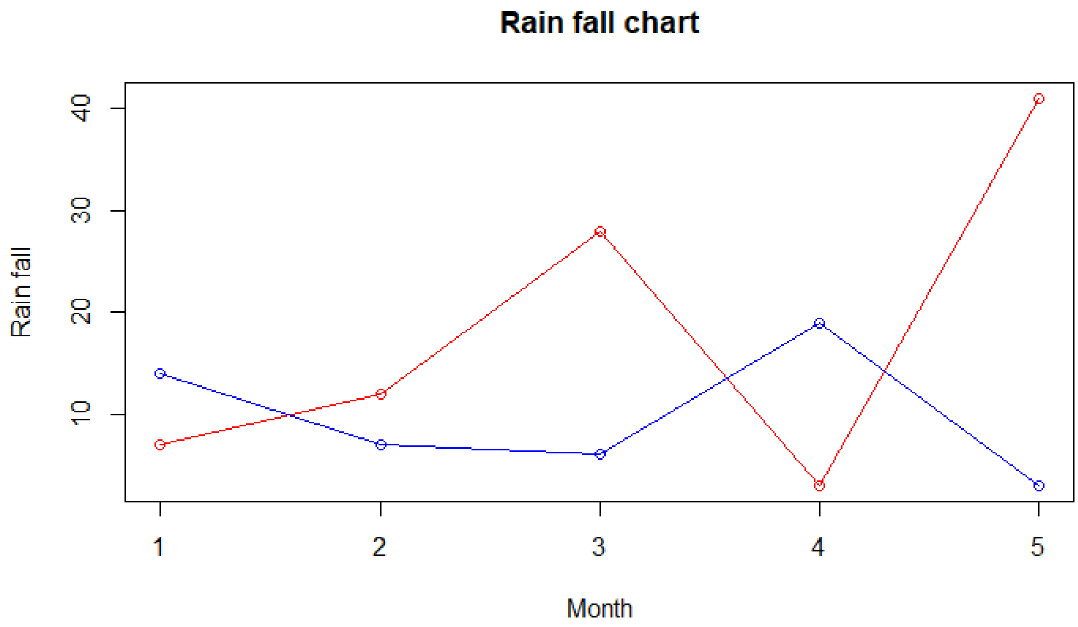
This boxplot is displaying two extreme values. Really does need a title and axis labels, though.

There are other variations on the box plot. You can change the orientation to horizontal instead. You can add color. Other variations include a notched box plot.



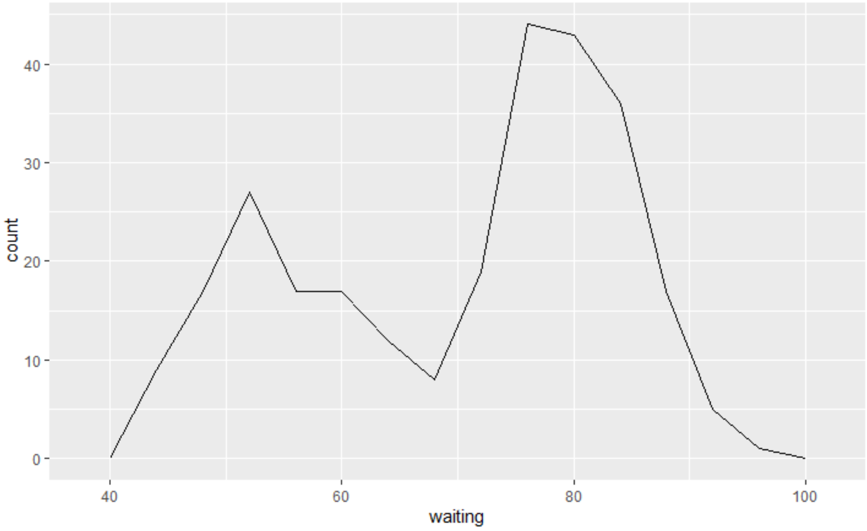
As we'll see below, you can also plot multiple boxplots side-by-side to compare data more easily than you can with a histogram.

A line graph connects the values on the vertical axis with lines. Occasionally it may be used in place of a histogram, but usually its used for time series. Ordered data must go on the horizontal axis (i.e. time), and then observations at each time go on the vertical axis.



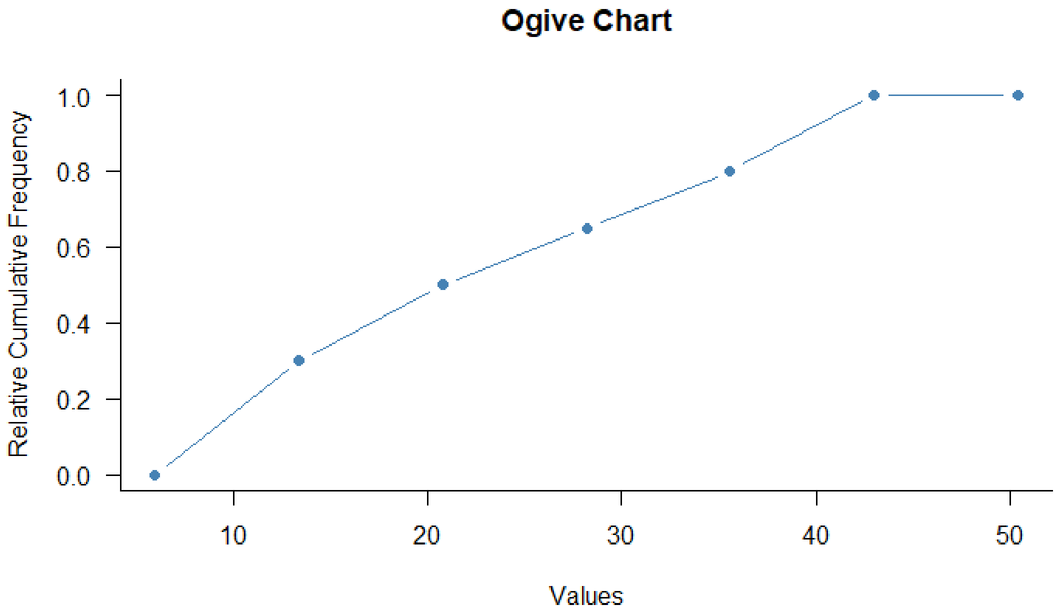
Often, you will plot only one line on a line graph, but as shown here, it can also be used to compare data, perhaps temperatures from two different cities in the same month.

A frequency polygon is essentially just a line graph (usually for discrete data) that plots frequencies.



(this graph was built in ggplot rather than base R).

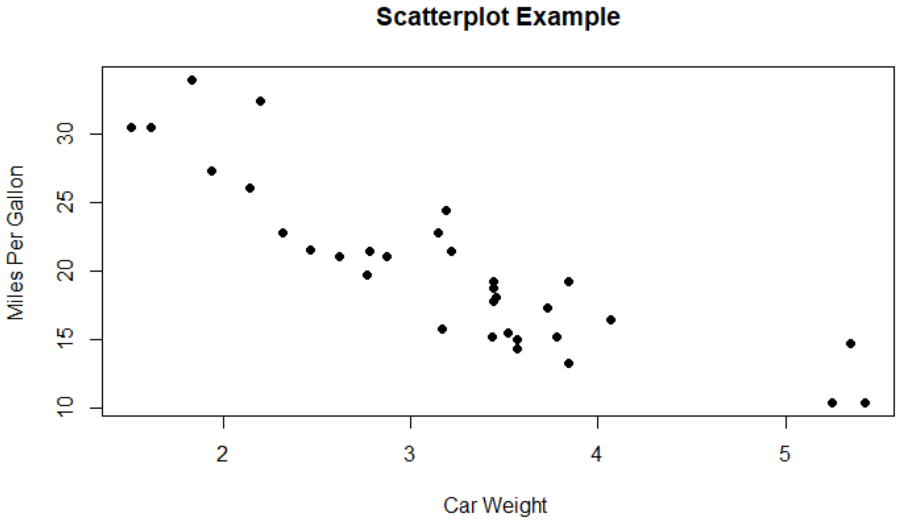
Sometimes you may want to shade below the line (making it an area plot), which is also okay. The ogive graph is basically the same idea except that the frequencies are cumulative.



Comparing data

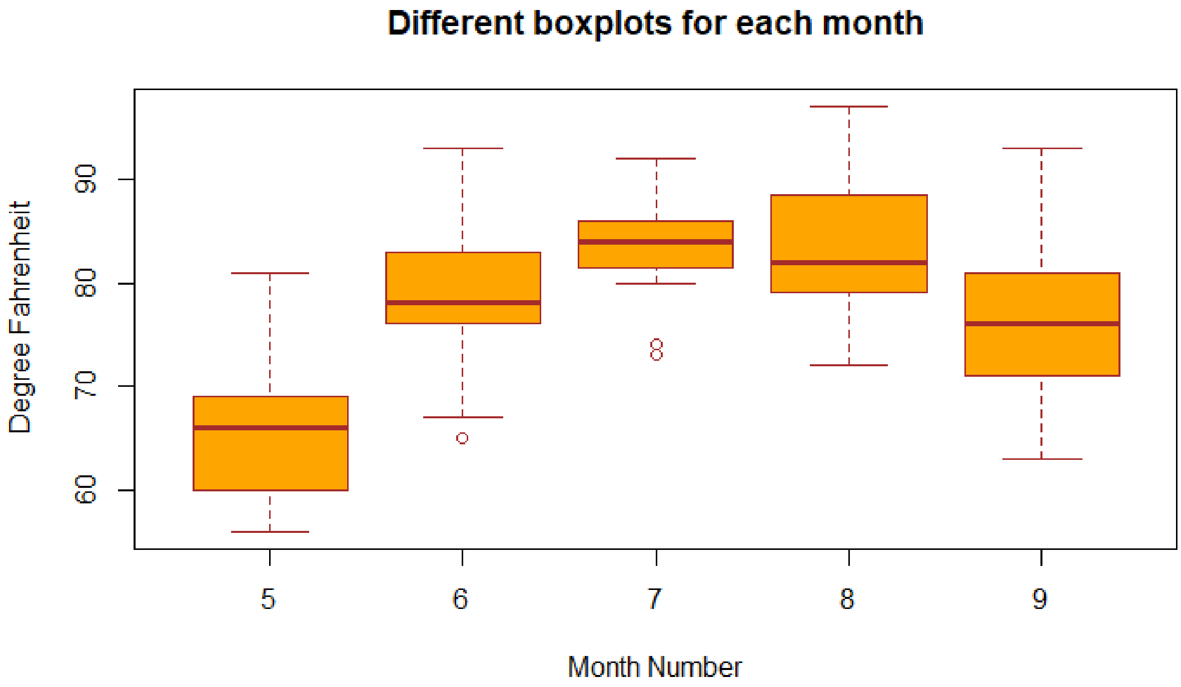
We've already seen how line graphs can be used to compare data of the same measurements in different categories, but what if we wanted to understand the relationship between two numerical observations? Then we need a scatterplot.

Scatterplots plot pairs of observations in a plane. Each pair is plotted with a dot.

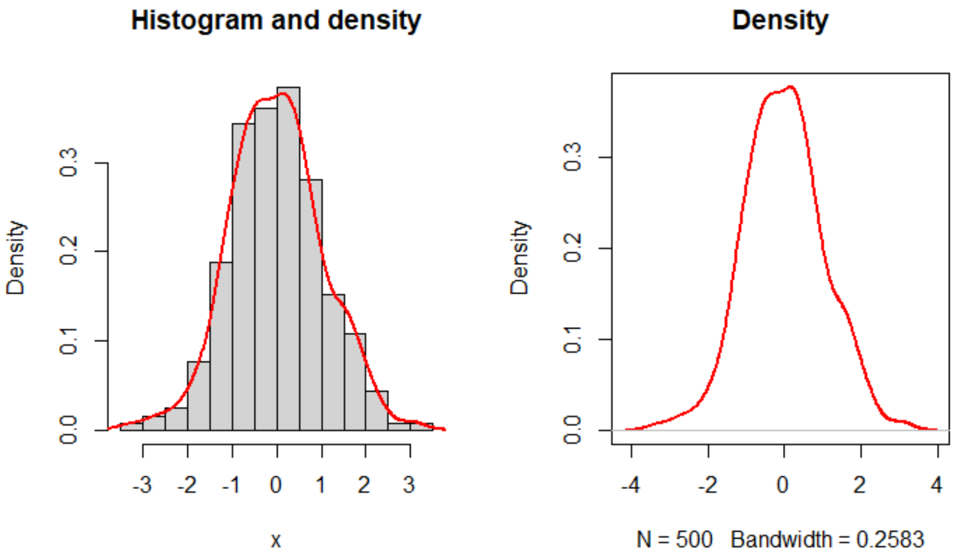


Scatterplots will be our go-to tool when we do regression. Then we'll look at adding regression lines to the plot, plotting in 3 dimensions and other features.

Comparative box plots let you compare numerical data in different categories.

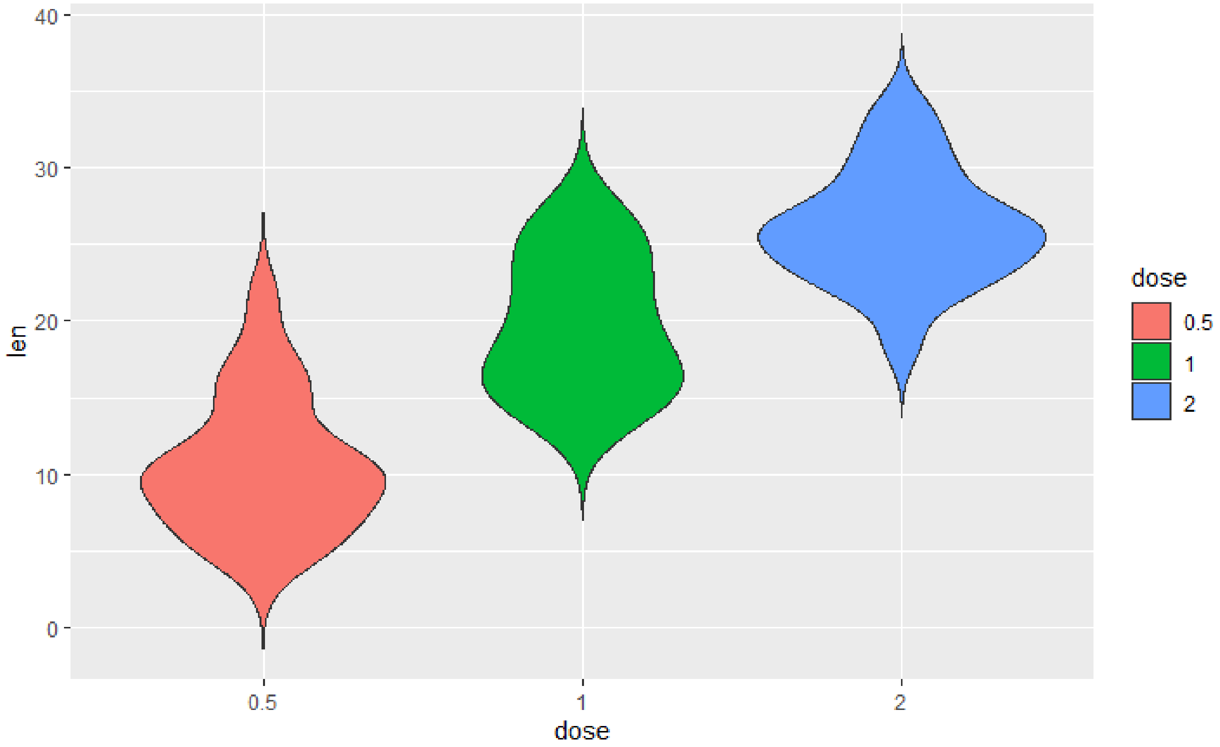


Other plots
As we become more sophisticated, we will adapt plots and experiment with more sophisticated graph types.



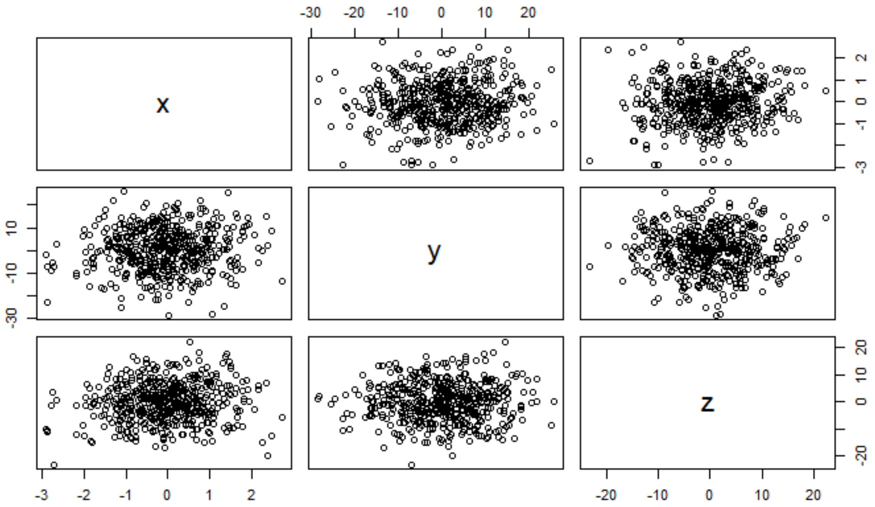
Density plots are built on histograms and try to smooth out the shape of the histogram curve. You can change the bandwidth to get smoother or rougher curves.

Violin plots combine density plots with boxplots.



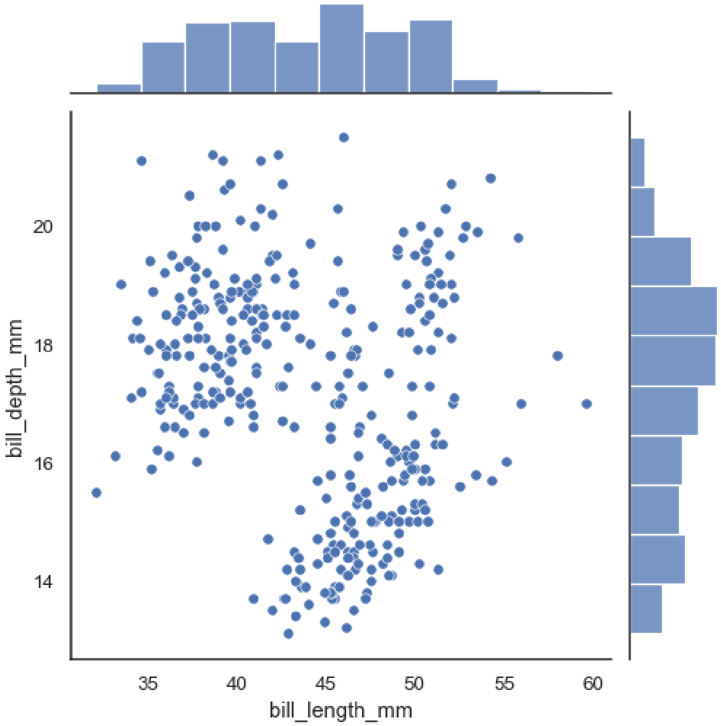
There are a wide variety of variations on the simple boxplot including dot plots (with jitter), swarm plots, and others. Each type has additional options you can add to make them more useful for whatever purpose you have.

Pair plots are great for getting an overview of the data and how they might be related to each other, by creating multiple scatterplots.



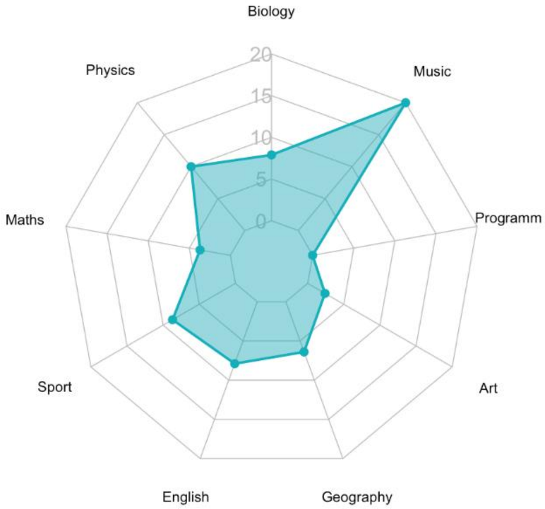
Some versions of pair plots will automatically make dummies of categorical data for you, and others will plot histograms on the diagonals. The data used for this plot is simulated, but in real data this is a good way to spot relationships. We'll look at these again more next semester.

Some examples of other types of graphs we may also want to experiment with as time goes on.
Joint plots:



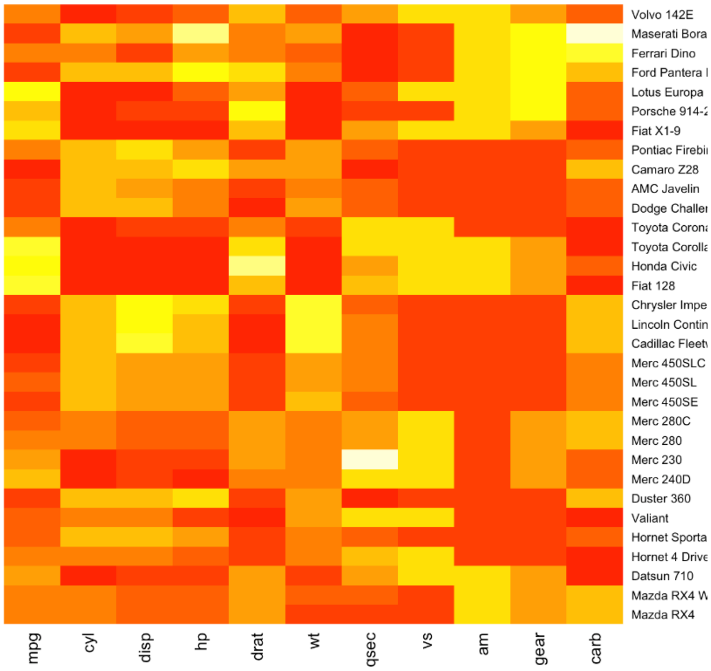
(This one is from python.)

Radar or spider plots.



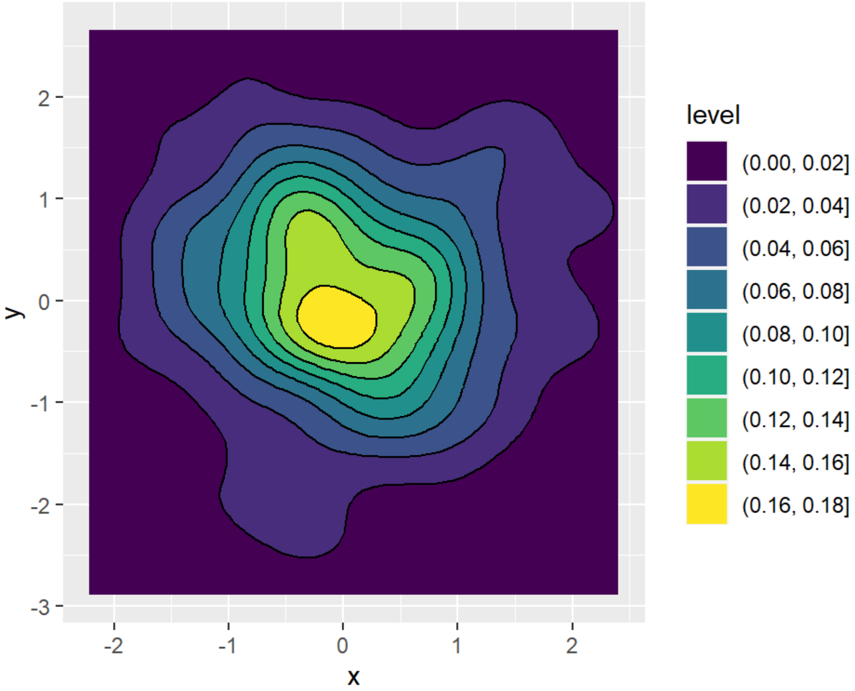
(These kind of plots can be problematic, so use with care.)

Heat maps.



Can be uses for plotting 3 values against each other, but be cautious about using with categorical data which is inherently unordered like this example.

2D density curve or contour plots.



We'll run into other kinds of graphs as we go, but there are a lot of options out there as we try to understand our data and communicate what we discovered to others.

Communication is why we turn to graphs, so it's an important aspect of graphs to think about. Not just so that we understand it, but so that other people do too. That's why it's important to think about the audience for your graphs, especially for the final versions, but also for the intermediate stage ones. If you walk away from your analysis and come back to it after a long time, will you still remember what the graph was displaying? What about someone who has never seen the original data?

Properties of good graphs

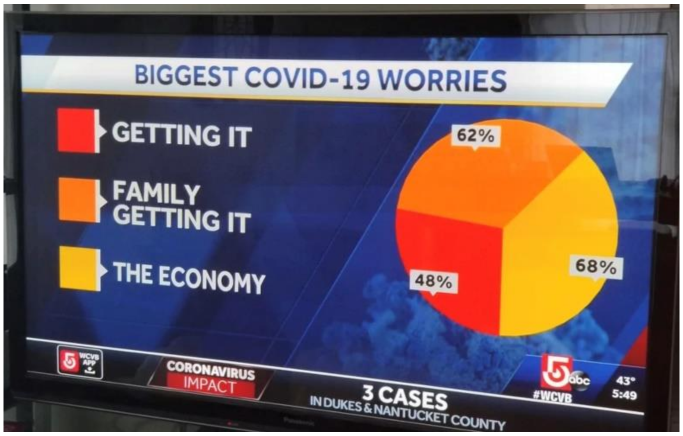
Good graphs should be able to be interpreted, stand alone, and not be misleading. In general, they should have:

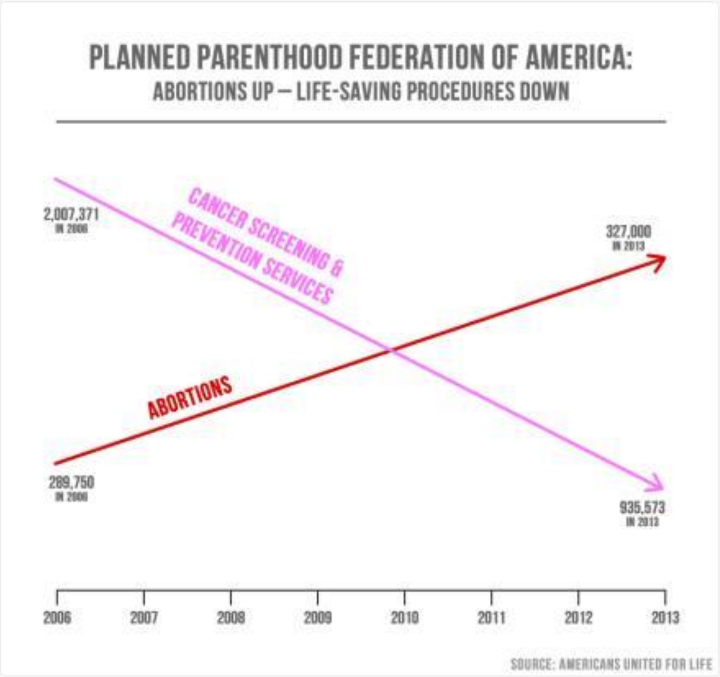
- Descriptive titles
- Axis titles (pie slices with percentages)
- A legend if the graph plots more than one variable
- Avoid misleading effects such as 3D
- An easy to interpret color scheme (there are color schemes to avoid issues with color-blindness, for example).

What makes a bad graph?

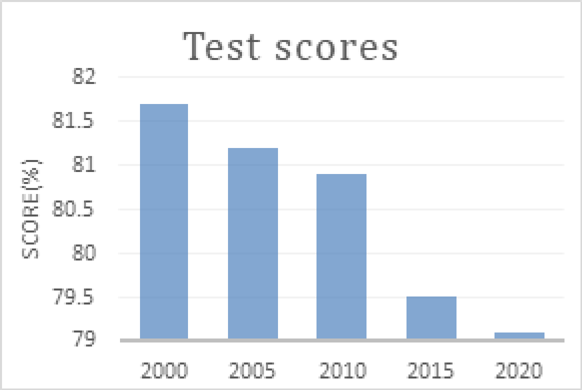
- Is the graph misleading? Does the bar graph start at 0? If not, it might be exaggerating differences.
- Is the graph in 3D? Our brains don't process area or volume as readily as height only information. Pretty sometimes can be the enemy of effectively communicating information.
- Do you know what is being graphed? Is the title vague? Do you have appropriate units on your axis labels? Is anything missing?
- Is it the right graph for the data? Is it the right graph for the kind of data you have? Do the variables you are plotting have meaning?
- If you can't interpret your graph, neither can anyone else.
- Don't try to put too much data into one graph.

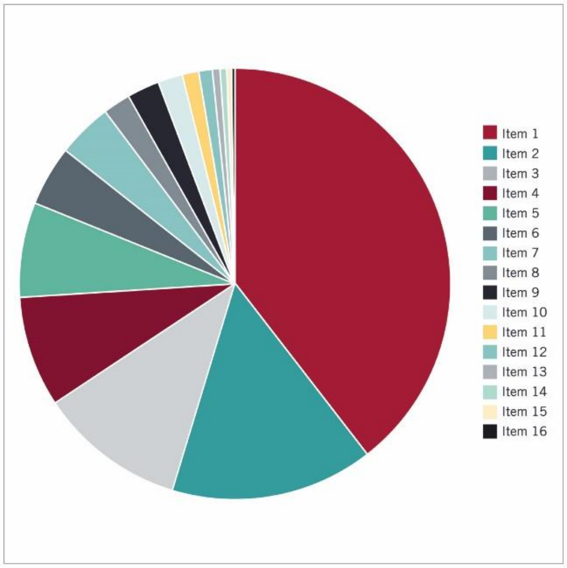
We can find all kinds of examples of bad graphs in the media and online. Sometimes people are actively trying to mislead. Sometimes, it's because they aren't thinking about the message their graph conveys, or are trying to be too cute. Often, simpler is better.

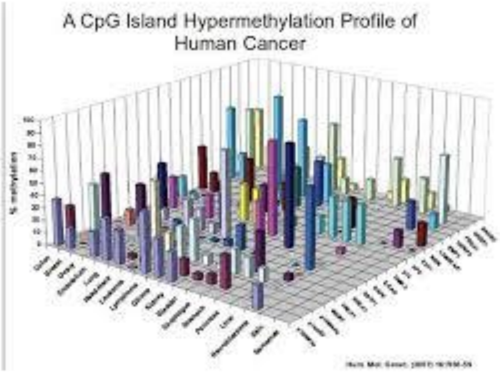


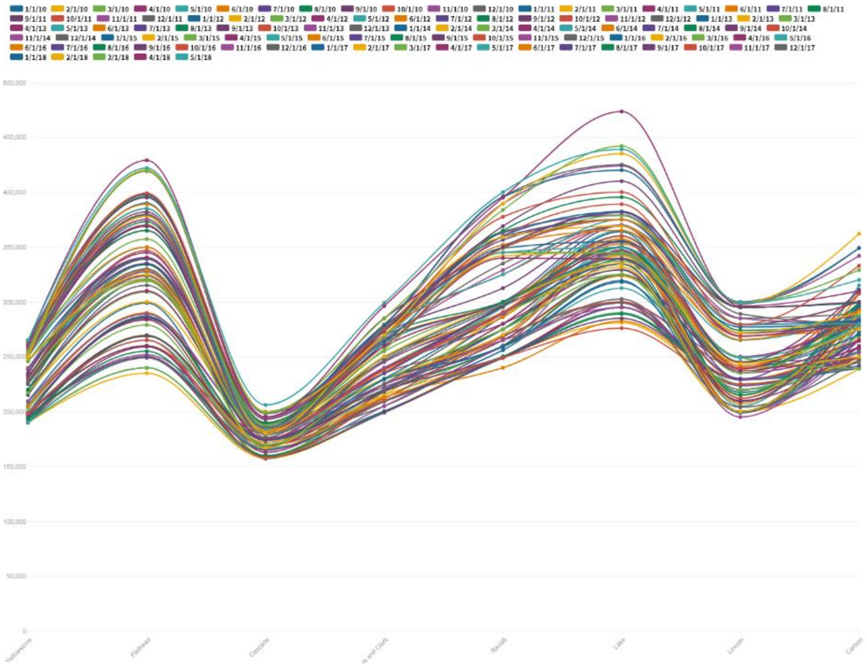


SOURCE: AMERICANS UNITED FOR LIFE









The point is, just because you can do it, doesn't mean you should.

Resources:

1. https://www.statmethods.net/graphs/pie.html
2. https://faculty.ksu.edu.sa/sites/default/files/probability_and_statistics_for_engineering_and_the_sciences.pdf
3. https://assets.openstax.org/oscms-prodcms/media/documents/IntroductoryStatistics-OP_i6tAI7e.pdf
4. https://www.statmethods.net/graphs/bar.html
5. https://www.r-bloggers.com/2021/08/how-to-create-pareto-chart-in-r/
6. https://www.google.com/search?q=examples+of+bad+graphs&tbm=isch&chips=q:examples+of+misleading+graphs,g_1:media:BgG3awE_fis%3D&rlz=1C1RXQR_enUS988US988&hl=en&sa=X&ved=2ahUKEwj3mZvxnor4AhWJEGIAHYXDA3sQ4lYoAnoECAEQIg&biw=1583&bih=757
7. https://www.datamentor.io/r-programming/histogram/
8. https://www.tutorialgateway.org/stem-and-leaf-plot-in-r/
9. https://www.datamentor.io/r-programming/box-plot/

10. https://r-graphics.org/recipe-distribution-freqpoly
11. https://www.statology.org/ogive-graph-in-r/
12. https://www.tutorialspoint.com/r/r_line_graphs.htm
13. https://www.statmethods.net/graphs/scatterplot.html
14. https://r-coder.com/density-plot-r/
15. http://www.sthda.com/english/wiki/ggplot2-violin-plot-quick-start-guide-r-software-and-data-visualization
16. https://www.geeksforgeeks.org/how-to-create-and-interpret-pairs-plots-in-r/
17. https://seaborn.pydata.org/generated/seaborn.jointplot.html
18. https://www.datanovia.com/en/blog/beautiful-radar-chart-in-r-using-fmsb-and-ggplot-packages/
19. https://r-graph-gallery.com/215-the-heatmap-function.html
20. https://r-charts.com/correlation/contour-plot-ggplot2/