Data visualization, scatterplots

Steve Simon

Created 2019-08-16

Introduction, Modules

- Scatterplots
- Bar charts
- Line plots
- Surface plots (optional)
- Maps (optional)

This workshop is split up into three to five modules. The basic modules cover scatterplats, bar charts, and line plots. There are optional advanced modules on surface plots and maps.

Introduction, Module components

- Preparation
- Exercise, what is the message?
- Tutorial
- Fundamentals
- Basic exercises
- Recommendations
- Exercise, identify features
- Short quiz
- On your own

Each module will start with some basic preparations that you should do before the class starts or at the very beginning of the class..

Then you will review some data visualizations that appeats in newpspaper or web articles. Your goal is to identify (in 25 words or less) the message is that the visualization is trying to convey. The

Each module will include a tutorial on perceptual issues that you need to understand in order to design effective data visulalizations.

Then you will review some fundamental commands that you need to know to draw basic data visualizations. Warning: some of the visualizations you will be asked to produce will look terribly ugly. That's okay. You're just learning the basic programming steps for now. Later, you will see how to apply these steps to make better looking and more effective visualizations.

Then you will learn some general recommendations on how to produce effective visualizations.

You will return to the newspaper articles and and relate features of the data visualizations to concepts discussed in the previous lecture.

You will have to answer a short quiz, not for a grade, but to reinforce some of the important points in the module.

Finally, you will work on your own with a different data set, trying to create effective data visualizations similar to the ones presented in each module.

Introduction, Software agnosticism

- This course will show examples using
  - Python,
  - R, and
  - Tableau
- I do not play favorites

I am a big believer in software agnosticism. That means that when I teach something, I teach it with the expectation that the software used to do the assignments is software of YOUR choosing. I have my own preferences, but those should not be your preferences.

It's a lot more work to teach a course that is not dependent on a particular software system, but I do not know what the best software choice would be for you. In this class, I will try to show examples using Python, R, and Tableau. I realize that there are other good choice, but I'm hoping that most of you will be happy with one of these three choices. Within Python, I will use the altair package. In R, I will use ggplot2. I realize that there are other graphics packages in these two languages, but altair and ggplot2 rely on modern graphics principles, so I will restrict my attention to these packages.

Tableau is a commercial product. If you don't currently have access to Tableau, the company offers a free version, Tableau Public. It has all the features of Tableau, but you have to store any data visualizations on a public server. That's just fine for someone like me who uses teaching examples with publicly available data sets. If you are using private or proprietary data, you need to pay the money for the commercial

version.

Introduction, What software should you use?

- Use the software you like best
- What does your boss use?
- What do your co-workers use?
- What software are you most comfortable with?

If you are not sure what software package to use in this class, let me offer a few suggestions. First, your boss may have a strong opinion about what software that you should use. You can go to your boss and say "My teacher is a really smart guy says that the _____ package in _____ is the best choice for data visualization." Try it and see what happens. Nothing, I suspect. One of the great tragedies in life is that the wise advice you get in this class carries very little weight in the real world.

If your boss doesn't care, see what most of your co-workers are using. They may not be as smart as I am (put on a false air of pride here) but they are a lot closer to your cubicle when this workshop ends and you have to find a quick answer.

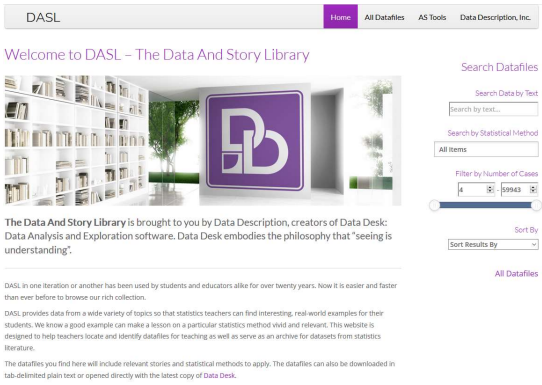
There's also a comfort level here. Tableau develops its visualizations using a graphical user interface. Python and R are programming languages. A graphical interface is great for getting work done quickly. A programming language is great for reproducibility and reusability. What fits your working style better? I don't know and it would be arrogant of me to make the presumption that I do know.

One more consideration. Some of you in this class are "ringers." You already know visualization better than I do because you've been doing it for longer, with bigger and

more complex data sets. You're just here to see if I know one or two things that you don't already know. If you're a ringer, take the challenge of learning a new software system. It will keep you from getting too bored when I talk about all these things that you already know better than I do.

Preparation, DASL

https://dasl.datadescription.com/

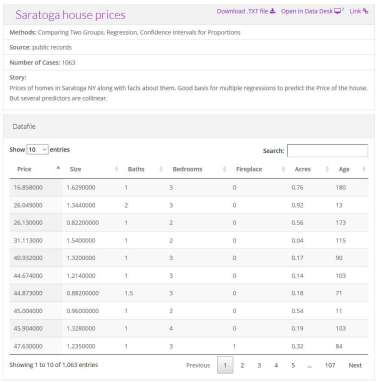


Main page of DASL website

DASL is an acronym for Data And Story Library. It used to sit on a website, Statlib, at Carnegie Mellon, but the company, Data Description, which makes a data analysis program, DataDesk, took over when the Statlib site went dark. It's a very nice site for small data sets useful for teaching.

I want to use a file on housing prices in Saratoga, New York, and you can find it through the search function on the main page. Look for housing or Saratoga, and you'll find it pretty quickly.

Preparation, Saratoga House Prices



Saratoga house prices

Download .TXT file | Open in Data Desk | Link

Methods: Comparing Two Groups, Regression, Confidence Intervals for Proportions

Source: public records

Number of Cases: 1063

Story:
Prices of homes in Saratoga NY along with facts about them. Good basis for multiple regressions to predict the Price of the house. But several predictors are collinear.

Datafile

Show 10 entries Search:

Price	Size	Baths	Bedrooms	Fireplace	Acres	Age
16.858000	1.6290000	1	3	0	0.76	180
26.049000	1.3440000	2	3	0	0.92	13
26.130000	0.82200000	1	2	0	0.56	173
31.113000	1.5400000	1	2	0	0.04	115
40.932000	1.3200000	1	3	0	0.17	90
44.674000	1.2140000	1	3	0	0.14	103
44.873000	0.88200000	1.5	3	0	0.18	71
45.004000	0.96000000	1	2	0	0.54	11
45.904000	1.3280000	1	4	0	0.19	103
47.630000	1.2350000	1	3	1	0.32	84

Showing 1 to 10 of 1,063 entries Previous 1 2 3 4 5 ... 107 Next

Description of the Saratoga house prices dataset

There are two files actually, that look very similar. You want the “Saratoga House Prices” file and not the one called “Saratoga Houses”. The Saratoga House Prices file has 1063 records and the variables are Price, Living.Area, Bathrooms, Bedrooms, Fireplaces, Lot.Size, Age, Fireplace.

This is the file that you want to download and run a scatterplot.

Actually, you can just use the CSV file that I thoughtfully provide.

Preparation, Advice if things don't work

- Download the file
- Tweak the file
  - Remove variable names in first line
  - Change missing value codes
  - Change the delimiter
  - Look for inconsistencies
  - Convert the format

These things never work right the first time. If computers worked the first time and every time, we'd all be getting paid the minimum wage. I'm glad to help if you have any problems importing this or any other files, but here is some general advice that you might want to try first.

Some software systems allow you to download directly from the web. This is fast and easy and convenient, but if it doesn't work download the file and see if you can import it directly.

If this doesn't work, open the file in a text editing program like notepad and see if you can make some minor changes that allow you to import the file.

If the first line of code has variable names, see if you can import the file without the variable names.

Look at the code for missing values. Some systems use a single dot for a missing value, and others use the letters "NA". Sometimes converting the missing value code to a different code will help.

Most files have a delimiter, a character that separates one variable from the next. This could be a space, a comma, or a tab character. You may have better luck if you do a careful search and replace, changing from one delimiter to another.

Look for inconsistencies, such as some invalid lines at the bottom of the file or a line with one too few variables or an unmatched quote mark. Fix these and try again.

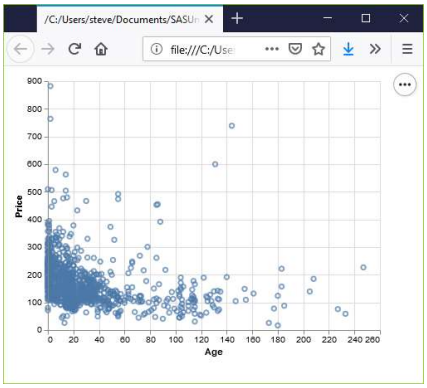
See if you can read the file into a different program like a spreadsheet and export it in a different format.

Preparation, Python code

```
import pandas as pd
import altair as alt
df = pd.read_csv("data/houses.csv")
ch = alt.Chart(df).mark_point().encode(
    x='Age',
    y='Price'
)
ch.save("/images/python-scatterplot.html")
```

Here is the Python code that will download the data and create a simple scatterplot. You may need to adapt the names of the files or the directories where they are stored.

Preparation, Python output



Basic scatterplot using Python

See if you can get a graph that looks something like this. If your graph uses different colors, scaling, etc. don't worry. I just want to know at this point that you can produce any sort of reasonable graph.

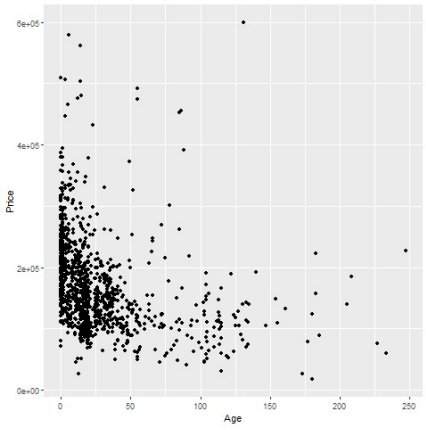
Preparation, R code

Here is the R code that will download the data and create a simple scatterplot.

```
library(ggplot2)
saratoga_houses <- read.csv("data/houses.csv")
ggplot(saratoga_houses, aes(x=Age, y=Price)) +
  geom_point()
```

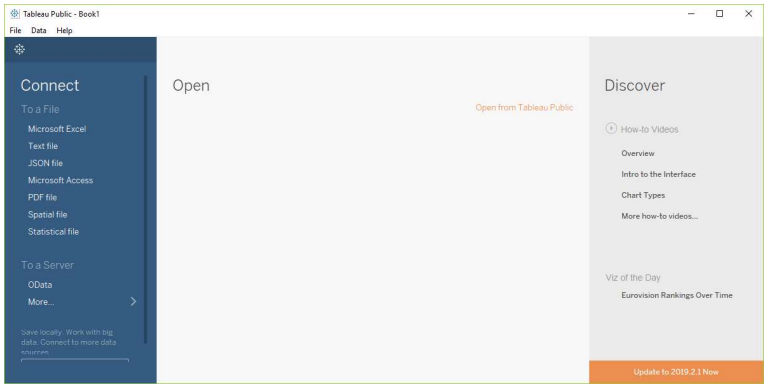
Here's a brief bit of R code that should work. You may need to adapt the names of the files or the directories where they are stored.

Preparation, R output



Again, see if you can get a graph that looks something like this. If your graph uses different colors, scaling, etc. don't worry. I just want to know at this point that you can produce any sort of reasonable graph.

Preparation, Tableau Open dialog box



Screenshot of Tableau software, main screen

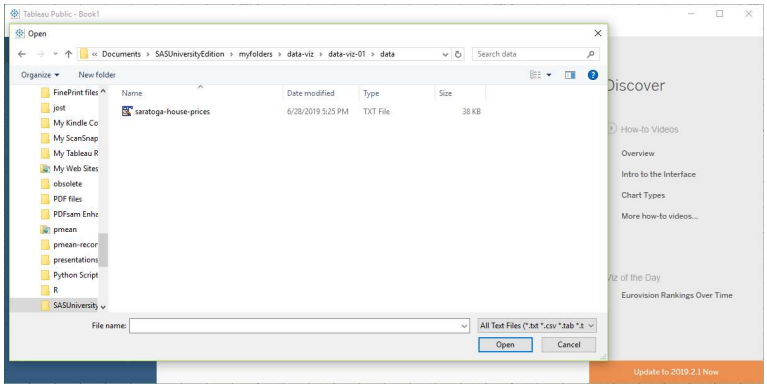
Before you start with Tableau, download the Saratoga Housing Prices file to your local computer.

Tableau uses a graphical user interface, so there is no “program” to run. Here are the steps you need to take to get the data in and produce a simple scatterplot.

This is what Tableau looks like when you open it up. It may appear slightly differently on your computer system.

Select “Text file” from the left side menu bar or “File | Open” from the main menu.

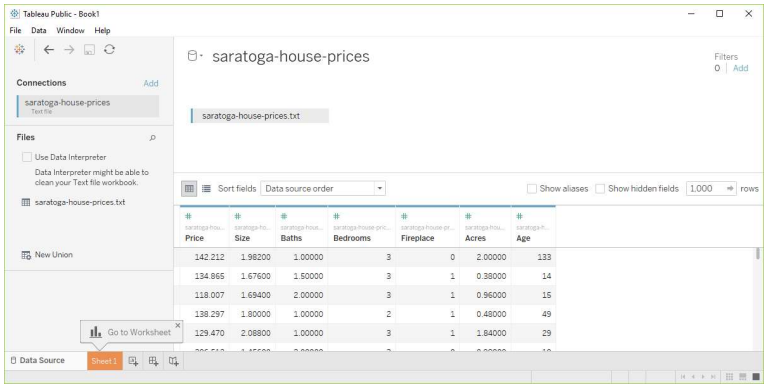
Preparation, Tableau, Select your file



Screenshot of Tableau software, file open dialog box

Find the proper location on your computer where you stored the downloaded file and open it.

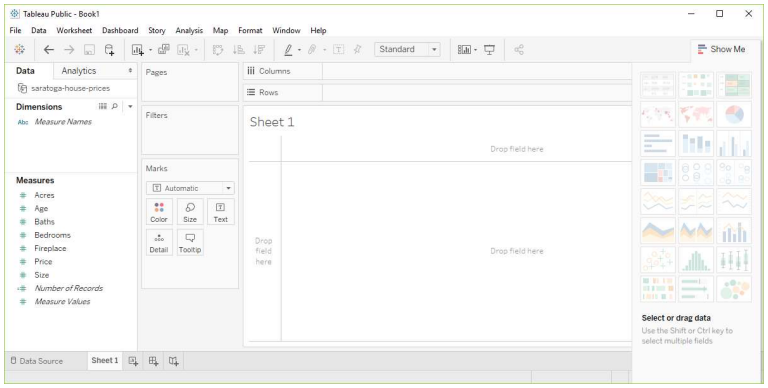
Preparation, Tabelau data preview



Screenshot of Tableau software with data imported

Your screen should look something like this if you imported the data correctly. Click on the sheet1 tab in the lower left corner to open up a blank visualization page.

Preparation, Tableau, start visualization

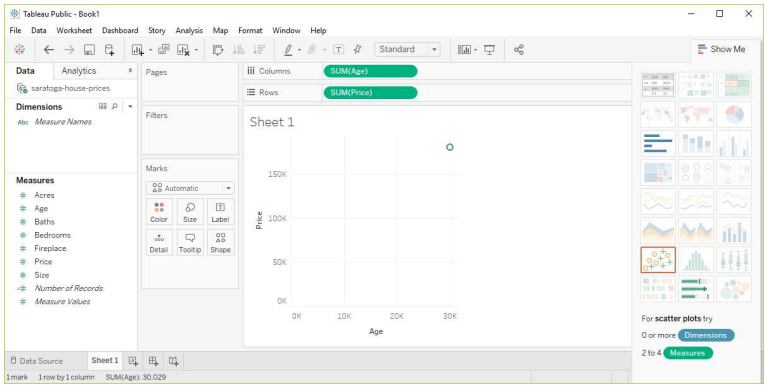


Screenshot of Tableau software, blank visualization page

Tableau will try to classify your data as measures or dimensions and will also try to decide whether they are categorical (designated by blue pills) or continuous (designated by green pills). Don't worry too much about this now, other than to note that changing the designations that Tableau makes will change how you visualize things.

Drag the variable Age into the Columns field and drag Price into the Rows field

Preparation, Tableau, Identify columns and rows

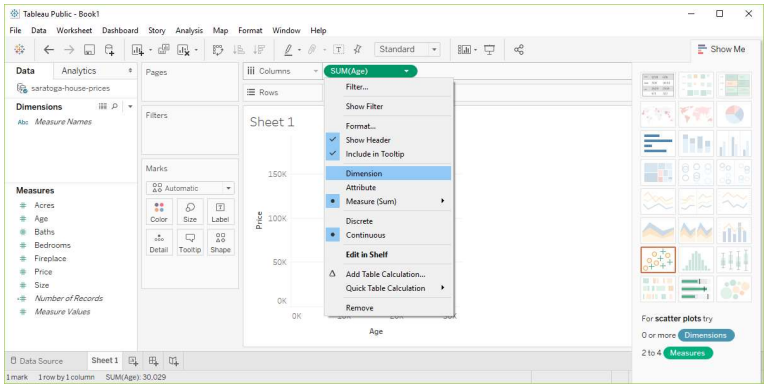


Screenshot of Tableau software after initial drag and drop

Tableau makes some educated guesses about what you want. It thinks that you are interested in aggregating age and price and plots a single data point with the sum of all the ages of the houses on the X axis and the sum of all the prices on the Y axis.

This is not what you want, but that's okay. Better a wrong guess that you can correct than no guess at all.

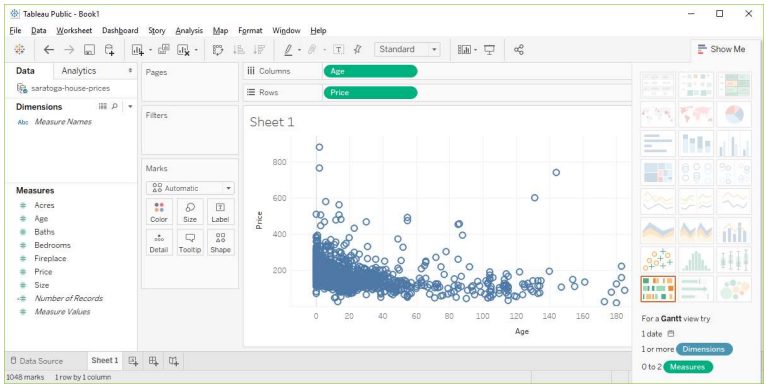
Preparation, Tableau, change columns



Screenshot of Tableau software changing from measure to dimension

Click on the green Sum(Age) and change it from Measure(Sum) to Dimension. Don't freak out when the graph goes all bonkers on you. Click on the green Sum(Price) and do the same thing. Then you'll get a nice basic scatterplot.

Preparation, Tableau output



Screenshot of Tableau software changing from measure to dimension

Did you get something that looks roughly like this? Pat yourself on the back for a job well done.

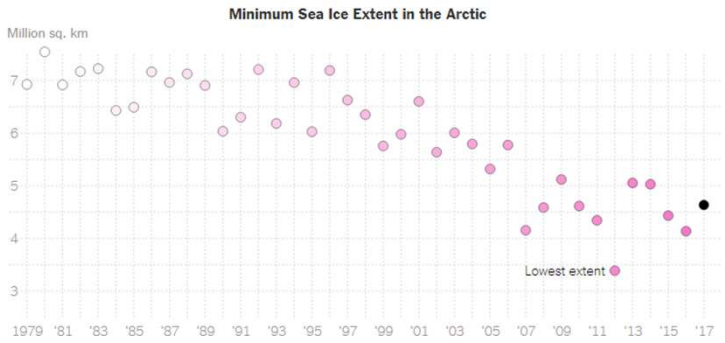
Graphs in the news, group exercise

– Review one of these graphs/newspaper articles

- We Charted Arctic Sea Ice for Nearly Every Day Since 1979. You'll See a Trend.
- We Read 150 Privacy Policies. They Were an Incomprehensible Disaster.
- How Medicine Became the Stealth Family Friendly Profession.

The following images are taken from various newspaper articles. Look at the graph and read/skim the article.

Graphs in the news, Arctic ice levels

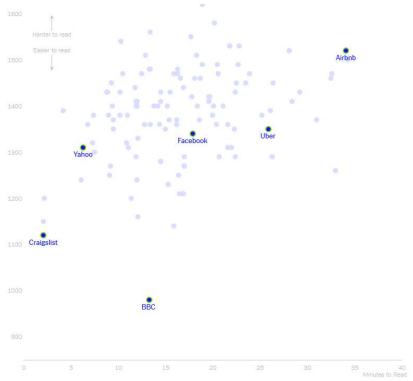


Scatterplot of Arctic ice levels

This image was found in a newpaper article,

Popvich, N., Fountain, H., & Pearce, A. (2017, September 22). We Charted Arctic Sea Ice for Nearly Every Day Since 1979. You'll See a Trend. - The New York Times. The New York Times. Retrieved from https://www.nytimes.com/interactive/2017/09/22/climate/arctic-sea-ice-shrinking-trend-watch.html

Graphs in the news, Privacy policies

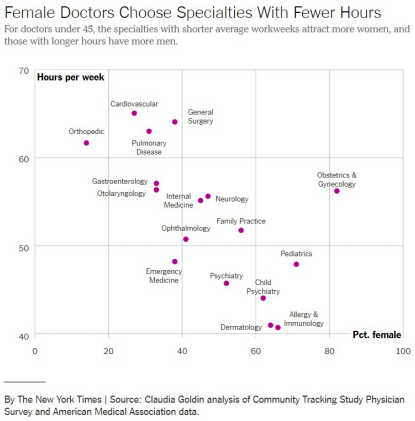


Scatterplot of length versus reader difficulty

This image was found in a newpaper article,

Kevin Litman-Navarro. We Read 150 Privacy Policies. They Were an Incomprehensible Disaster. The New York Times. Retrieved from https://www.nytimes.com/interactive/2019/06/12/opinion/facebook-google-privacy-policies.html

Graphs in the news, Family friendly profession



This image was found in a newpaper article,

Claire Cain Miller (2019, August 21). How Medicine Became the Stealth Family Friendly Profession. The New York Times. Retrieved from https://nytimes.com/2019/08/21/upshot/medicine-family-friendly-profession-women.html

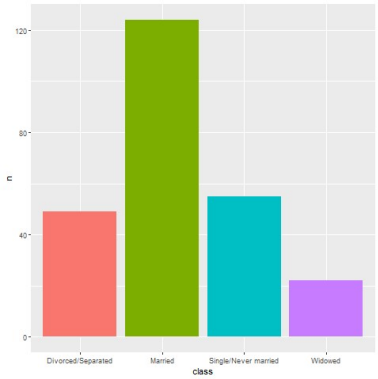
Graphs in the news, what is the message?

- Your group will be assigned one particular graph and newspaper article
- Read/skim the article and examine the graph
- What is the message?
  - Summarize in 25 words or less.

The following images are taken from various newspaper articles or press releases. Look at the graph and read/skim the article.

What message do you think the journalist is trying to convey with this graph. Summarize this message in 25 words or less.

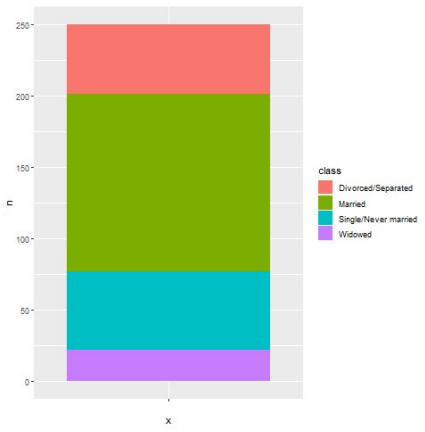
Evaluation, Which is better? A bar chart…



Bar chart, with four bars side by side

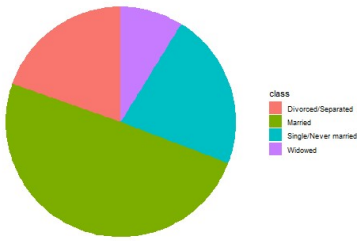
Here's a bar chart dispaying the counts for people in four categories for marital status, divorced, married, single, widowed.

Evaluation, … a stacked bar chart, …



Here's the same data, but the bars are now stacked in a single column.

Evaluation, … or a pie chart



Pie chart

There’s one more obvious choice. You can display the counts in a pie chart.

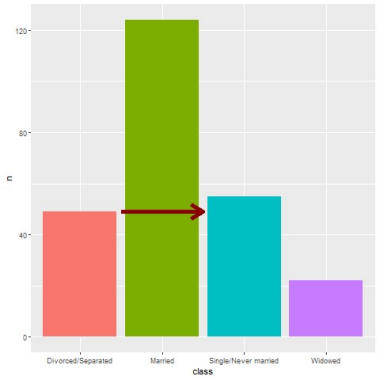
Evaluation, Better in what way?

– Two most important criteria
  - Speed
  - Accuracy

The answer really depends on what question you are asking. There are a variety of questions that you might ask.

You can run an experiment (people have done this) where you randomize and show half of them a bar chart and half of them a pie chart. Then you ask a question. Then you note the speed and accuracy of the response. Depending on the question, sometimes pie charts give faster and more accurate answers. Sometimes bar charts give faster and more accurate answers. It turns out that the results match up nicely with what we know about the psychology of perception.

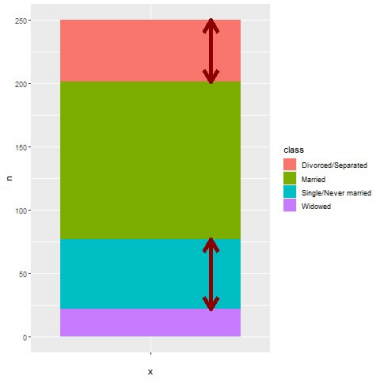
Evaluation, Which percentage is larger? Bar chart is best!



Bar chart, projecting one bar to another

If you ask the question, which percentage is larger, the percentage for single or the percentage for divorced/separated, the bar chart is the winner, hands-down. The comparison involves a simple horizontal projection. You can do this quickly and accurately.

Evaluation, Second best, stacked bar chart



Stacked bar chart, comparing lengths

The second best choice for answering this question is the stacked bar chart. You have to compare the lengths of two bars which are not aligned. It takes a bit longer to make this judgement, and it is harder to provide an accurate answer when the bars are very similar in length. But it is still not too difficult.

Evaluation, Worst, pie chart

Pie chart, comparing two angles

For the pie chart, you have to judge which wedge of the pie is bigger by looking at the area of the wedges, but actually most people make assessments in a pie chart by looking at the interior angle. You can see that the interior angle is bigger for the single group, but it is a harder judgement to make quickly and it is a harder judgement to make accurately.

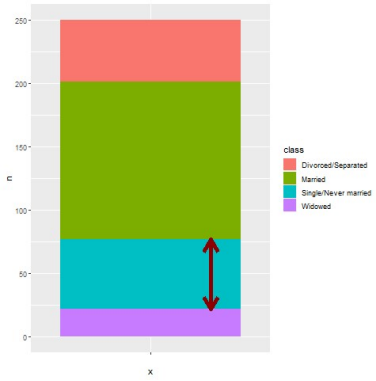
Now I need to note that this is not my opinion. It is a fact established by empirical study. The first bar chart, which allows you to make comparisons using relative position, produces faster and more accurate answers.

Evaluation, What fraction of people are single? Best is pie chart

Pie chart, estimating a percentage for one wedge

But wait! If you ask a different question, what percentage of people in your sample are single, the pie chart does the best. The interior angle for the single wedge is just a bit under 90 degrees, and that tells you quickly, and fairly accurately, that the percentage is a bit under 25%.

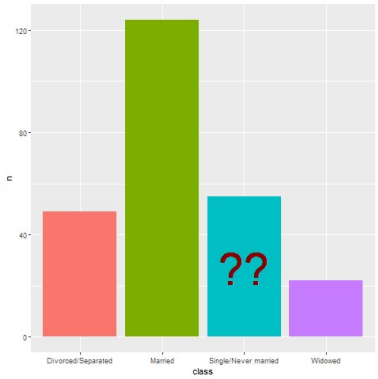
Evaluation, The stacked bar chart is second best



Stacked bar chart, estimating percentage

You can ask the same question for the stacked bar chart, but it will take longer and be less accurate. It's easy to split a bar in half, but you will have much harder perceptual task to split it into quarters. So deciding whether that single bar is a bit less than 25% or a bit more than 25% can't be done as well.

Evaluation, The side-by-side bar chart is hopeless.

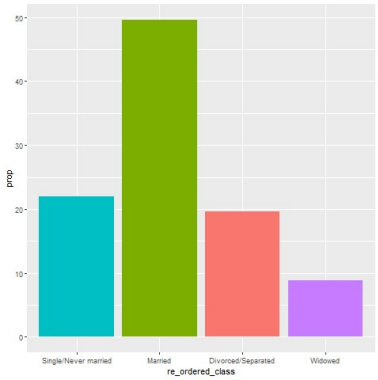


Bar chart, trying to assess percentage questions

Trying to answer a percentage question for a side by side bar chart is pretty much hopeless. What you have to do is visually stack the bars and then divide the bars into quarters.

When you ask percentage questions, all methods do pretty well for percentages near 0% and 100%. All methods also do well for percentages around half, or 50%. The pie chart also does well for percentages close to 25% and 75%. It is easy visually to split a pie into four equal pieces. Just look for the 90 degree angles. The empirical research supports this. Speed and accuracy of percentage judgements are about the same for bar charts and pie charts, except around 25% and 75% where the pie chart is markedly superior.

Evaluation, Maybe the “hopeless” bar chart isn’t so hopeless



Bar chart, with bars re-ordered and a percentage value on the Y axis

Things are never totally hopeless, however. If your goal is to simplify the estimation of the percentage of your sample in the single/never married category, change your y axis from counts to percentages. Also, place the most important bar closest to the Y axis. Horizontal projections are easiest when the distances you have to project are very short.

There’s lots of other little things you can do. We’ll talk a lot more about bars in the second part of this workshop.

Evaluation, Summary

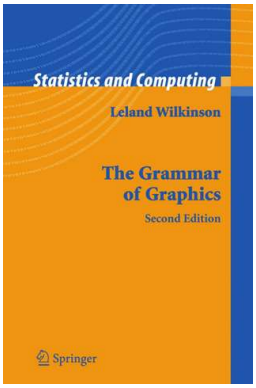
- Judging effectiveness of a graph
  - Speed
  - Accuracy
- Pie chart better for estimating percentages
- Any graph can be improved

The way that you judge the effectiveness of a graph is by speed and accuracy. How quickly you answer a question about the graph, and how accurate your answer is.

The bar chart is best for comparing one bar to another, but if you want to estimate the percentage of a whole, then a pie chart is better, particularly for percentages around 25% and 75%.

Even a bad graph can often be improved with some minor changes.

Grammar, Theoretical foundation of data visualization



Front cover of the book, The Grammar of Graphics

Most of the current designers of data visualization software have based their work on the theoretical foundations of Leland Wilkinson. This includes ggplot2 in r, altair in Python, and Tableau, among others. Dr. Wilkinson wrote a book, The Grammar of Graphics, in 1999 (second edition in 2006) that laid out the principles for the development of pretty much any data visualizaton that you could imagine. The work is mathematically rigorous, and I do not recommend that you read this book unless you enjoy that sort of thing. I do want to highlight a few of the fundamental ideas in the book

Grammar, Visualization before Wilkinson (1 of 3)

```
barplot(height, …)

# S3 method for default
barplot(height, width = 1, space = NULL,
        names.arg = NULL, legend.text = NULL, beside = FALSE,
        horiz = FALSE, density = NULL, angle = 45,
        col = NULL, border = par("fg"),
        main = NULL, sub = NULL, xlab = NULL, ylab = NULL,
        xlim = NULL, ylim = NULL, xpd = TRUE, log = "",
        axes = TRUE, axisnames = TRUE,
        cex.axis = par("cex.axis"), cex.names = par("cex.axis"),
        inside = TRUE, plot = TRUE, axis.lty = 0, offset = 0,
        add = FALSE, ann = !add && par("ann"), args.legend = NULL, …)

# S3 method for formula
barplot(formula, data, subset, na.action,
        horiz = FALSE, xlab = NULL, ylab = NULL, …)
```

Excerpt from the R help file for the barplot function

I'm going to do one of those bad things that a presenter should never do. I'm going to show you a series of three slides that is intended to create a sense of confusion. I'm doing this because I want you to appreciate how bad things were before Leland Wilkinson came along.

There's a saying that a camel is a horse designed by a committee. Well, this slide and the next two slides represent some very ugly looking camels.

Here's the help function from the program R for the barplot function. This function and the following were developed before Wilkinson's work and show the problem without using his framework.

I don't want you to study this help file closely. I just want to emphasize two points.

First, there are a dizzying number of options, 30 for the barplot. Maybe this is unavoidable. But what follows when you look at the next help function is the real problem.

When you switch from one function to another, from one visualization method to

another, the options change. This is bad because it makes it harder for you to learn new graphical display methods. Even after you learn them, you will have a difficult time remembering which options and which defaults go with which graphical displays.

Grammar, Visualization before Wilkinson (2 of 3)

```
hist(x, …)

# S3 method for default
hist(x, breaks = "Sturges",
     freq = NULL, probability = !freq,
     include.lowest = TRUE, right = TRUE,
     density = NULL, angle = 45, col = NULL, border = NULL,
     main = paste("Histogram of" , xname),
     xlim = range(breaks), ylim = NULL,
     xlab = xname, ylab,
     axes = TRUE, plot = TRUE, labels = FALSE,
     nclass = NULL, warn.unused = TRUE, …)
```

Excerpt from the R help file for the hist function

Here's the help function from the program R for the hist function. The default option for the borders of the bars of a barplot differ from the option for the borders of the bars of a histogram. This makes no sense. A histogram is different than a bar chart, but not that much different.

The way that you determine the limits for the x-axis and the y-axis differ. Again, how is a histogram so much different from a barplot that you need different methods for deciding something like this?

Grammar, Visualization before Wilkinson (3 of 3)

```
boxplot(x, …)

# S3 method for formula
boxplot(formula, data = NULL, …, subset, na.action = NULL,
        xlab = paste(names(mf)[-response], collapse = " : "),
        ylab = names(mf)[ response],
        add = FALSE, ann = !add,
        drop = FALSE, sep = ".", lex.order = FALSE)

# S3 method for default
boxplot(x, …, range = 1.5, width = NULL, varwidth = FALSE,
        notch = FALSE, outline = TRUE, names, plot = TRUE,
        border = par("fg"), col = NULL, log = "",
        pars = list(boxwex = 0.8, staplewex = 0.5, outwex = 0.5),
         ann = !add, horizontal = FALSE, add = FALSE, at = NULL)
```

Excerpt from the R help file for the boxplot function

If that isn't confusing enough, look at the help file for the boxplot.

It's pure chaos. Each function has a different set of arguments, listed in a different order and with different default options. This continues when you look at help for the pie function for pie charts, the contour function for contour plots, the persp3d function for three dimensional surfaces, the stem function for stem and leaf diagrams, and many others.

Adopting the framework developed in The Grammar of Graphics provides you with one stop shopping. It is a bit daunting at first, because it includes everything and the kitchen sink. But once you get comfortable with it, you will find that each new visualization that you try uses the same syntax, more or less.

Grammar, Helpful resource

Data Visualization: Principles and Applications in R, Tableau, and Python

Silas Bergen

Todd Iverson

2019 Symposium on Statistics and Data Science
Bellevue, WA

Title slide from the Bergen-Iverson presentation

In this section, I am going to borrow heavily from a short course presented at the 2019 Symposium on Statistics and Data Science. The presenters are nice enough to share their materials on their github site. You can find it easily with a google search of bergen iverson sdss2019 data visualization.

Grammar, Definition of data visualization

– "A mapping of data to the visual aesthetics of geometries/marks"
  - Bergen and Iverson 2019

A definition of visualization, based on the Grammar of Graphics framework is provided in the Bergan and Iverson presentation that I mentioned on the previous slide.

There are four nouns in this definition.

Data. I hope I don't have to define data other than to say that it is an interesting set of numbers. I won't talk about non-numeric data like text in this workshop. Ideally these numbers have enough structure that you can put them into a rectangular grid like a spreadsheet or database table.

Aesthetics is a work that Dr. Wilkinson likes, but I'm not so sure that I care for it. An aesthetic is a visual feature.

The compound noun geometries/marks is a deliberate choice of Bergen and Iverson. If you use ggplot2 in R, you will be more comfortable with the noun geometries. If you use altair in Python, or if you use Tableau, you will be more comfortable with the noun marks.

Mapping means a transformation. You are taking data and converting it into various visual features.

It will help to see some examples.

Grammar, Examples

- Geometries/marks
  - Points
  - Lines
  - Bars
- Aesthetics
  - Position
  - Shape
  - Size
  - Color

Think of geometries/marks are ink placed on a sheet of paper. They could represent points, lines, or bars, among other things.

Geometries/marks have several major visual properties, known as aesthetics. The aesthetics include position, shape, size, and color.

Not every geometry/mark will have every possible aesthetic. Some of these aesthetics can be combined to great effect, but sometimes they work antagonistically. Do consider every possible aesthetic in your graph, but intentionally ignoring an aesthetic can sometimes work to your advantage. Some aesthetics map very nicely to continuous data, but others only work well with categorical data.

Grammar, Geometries/marks in Python

– mark_point

– mark_line

– mark_bar

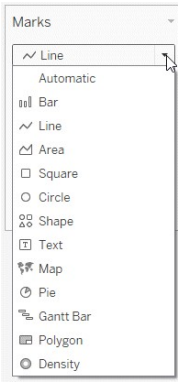
You specify the type of geometry/mark in Python using a range of mark functions.

Grammar, Geometries/marks in R

- geom_point
- geom_line
- geom_bar

You specify the type of geometry/marks in R using a range of geom functions.

Grammar, Geometries/marks in Tableau



Screenshot of the Marks pulldown menu

Tableau will try to guess what type of marks to use, but if it guesses wrong, you can correct things with the Marks pull down menu. This menu gives you options for (among other things) Bar, Line, and Shape (points).

Note that square and circle are special cases of points. This makes things a bit confusing, unfortunately. Always choose Shape because it gives you more latitude to change things.

Grammar, Mapping to aesthetics in Python

- encode function
  - x=
  - y=
  - shape=
  - size=
  - color=
- Example
  - alt.Chart(*data*).mark_point().encode(x='*var1*', y='*var2*', size='*var3*', shape='*var4*', color='*var5*')

In Python, the mapping is done with the encode function. The location is represented by x= and y= arguments. Size, shape, and color are mapped using the size=, shape=, and color= arguments.

Grammar, Mapping to aesthetics in R

- aes function
  - x=
  - y=
  - size=
  - shape=
  - color=
- Example
  - ggplot(*data*, aes(x=*var1*, y=*var2*)) +
  - geom_point(aes(size=*var3*, shape=*var4*, color=*var5*))

In R, the mapping is done with the aes (short for aesthetics) function. The location is represented by x= and y= arguments. Size, shape, and color are mapped using the size=, shape=, and color= arguments.

Grammar, Mapping to aesthetics in Tableau

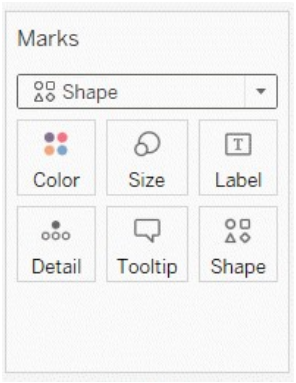


Tableau buttons below Marks

Tableau accomplishes mapping through a drag and drop interface. Drag a variable to the Columns field and a second variable to the Rows field to specify the x and y locations. Drag a variable on top of the size, shape, and color icons to map those variables appropriately.

Grammar, changing variable types in Python

- Q: Quantitative
  - Use for continuous variables
- O: Ordinal
  - Use for ordered categories
- N: nominal
  - Use for unordered categories
- T: Temporal
  - Use for time variables

All of these graphic packages will make assumptions about what the data represents, and set certain default graphic types on the basis of these guesses. Those choices are usally good, but when they are not, then you can override these guesses.

In Altair/Python, you use letter codes.

The letter Q (quantitative) will tell the system that you want to treat this variable as continuous.

The letter O (Ordinal) will tell the system that you want to treat this variable as ordered categories.

The letter N (Nomina) will tell the system that you want to treat this variable as unordered categories.

The letter T (Temporal) will tell the system that you want to treat this variable as a measure of time.

Grammar, Changing variable types in R

- as.numeric()
  - Use for continuous variables
- as.character(), as.factor()
  - Use for categorical variables
- as.Date()
  - Use for date variables

There are lots of conversion functions in R, and it beyond the scope of the class to cover them. But sometimes you will find in R that you need to convert the variable type before you do certain analyses. If you want to represent a variable as continuous, the as.numeric function will often do the trick. If you want to represent a variable as categorical, then try the as.character or as.factor functions. For temporal data, try the as.Date function.

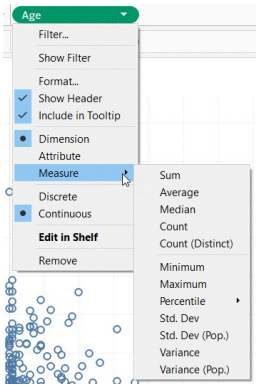
Grammar, Changing variable types in Tableau

- Dimension Discrete (blue pill)
  - Use for categorical variables
- Dimension Continuous (green pill)
  - Use for continuous variables
- Measure
  - Use for summary measures (average, count, etc.)

In Tableau, use Dimension Discrete to designate catergorical data. The color of the "pill" for your variable will change to blue when you make this choice. Use Dimension Continuous for continuous data. The color of the "pill" for your variable will change to green when you make this choice.

Use Measure when you want to display not the invidual values within a group but rather an aggregate or summary statistic, such as a mean or a count.

Grammar, Changing variable types in Tableau



Variable pull down menu in Tableau

This image shows the pull down menu for a variable with the choices you get.

Grammar, Summary

- "A mapping of data to the visual aesthetics of geometries/marks"
- Geometries/marks
  - Point
  - Bar
  - Line
- Aesthetics
  - Location
  - Size
  - Shape
  - Color

Data visualization is a mapping of data.

Geometries/marks include points, bars, and lines. Aesthetics include location, size, shape, and color.

These ideas are implemented in Python (Altair), R (ggplot2), and Tableau.

Fundamentals, Review basic scatterplot

– Python code

```
ch = alt.Chart(df).mark_point().encode(
    x='Age', y='Price'
)
```

– R code

```
ggplot(saratoga_houses, aes(x=Age, y=Price)) +
  geom_point()
```

– Tableau

- (Drag and drop)

You've already drawn a basic scatterplot in Python, R, or Tableau. All three systems choose a slightly different visualization, but, for the most part, the graphs look fairly nice. You can change some of those default options.

Fundamentals, Changing default options

- Python code

```
.mark_point(shape="square", color="green").
```

- R code

```
geom_point(shape="square", color="green")
```

- Tableau
  - Click on the shape and color buttons

In Python, you can change the default for every data point inside the mark_point function. In R, you do this inside the geom_point function. In Tableau, you have to click on the color and shape buttons to change the defaults.

Note that you may want to change the color or shape only for some of the data points. This is done inside the encode function in Altair/Python. In R, this is done inside the aes function. In Tableau, you drag and drop certain variables on top of buttons for color and shape.

Exercise, change the color and shape

- Use the Saratoga housing data set.
- Change the default color to any color you like
- Change the default shape to any shape you like

Take the scatterplot that you have already drawn and change the color of the points. Any color is fine as long as it is different from the default color. Change the shape of the points as well to any shape you want.

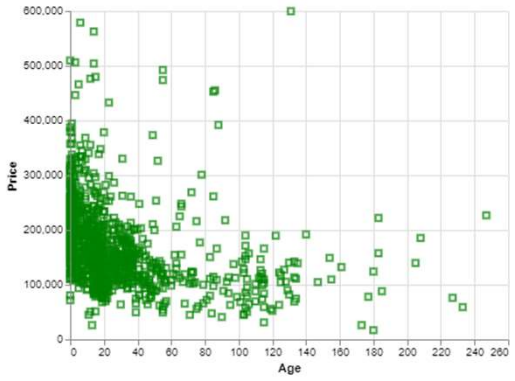
Exercise, Python code

– Here's the Python code.

```
ch = alt.Chart(df).mark_point(
    shape="square",
    color="green"
).encode(
    x='Age',
    y='Price'
)
```

In Python, you change defaults inside the mark_point function.

Exercise, Python output



Python scatterplot with green squares

Here's what my output looks like. You are welcome to experiment with different colors and different shapes.

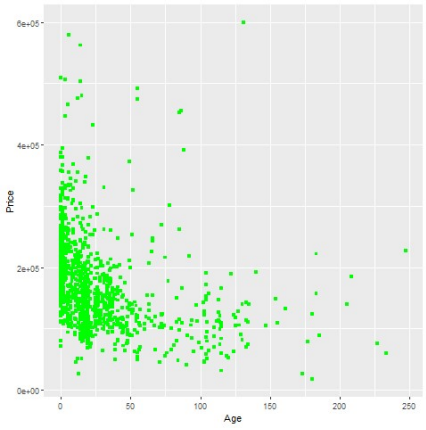
Exercise, R code

– Here's the R code

```
ggplot(saratoga_houses, aes(x=Age, y=Price)) +
  geom_point(shape="square", color="green")
```

The change in default shape and color are found in the geom_point function.

Exercise, R output



This is the R output. Try other options if you have time.

Exercise, Tableau output

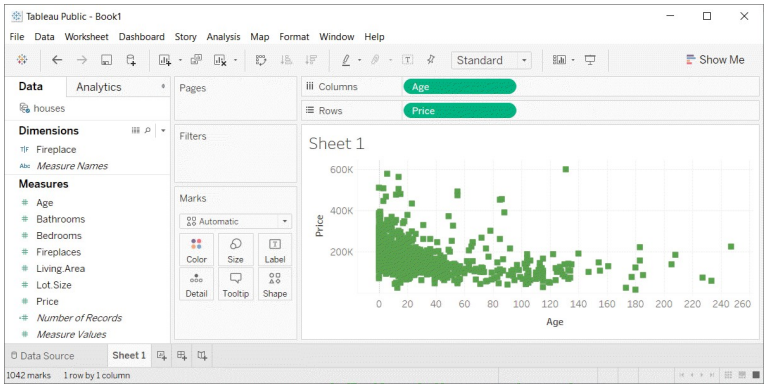


Tableau scatterplot with green squares

Here is the Tableau output.

Fundamentals, Aesthetics for points - location (1 of 2)

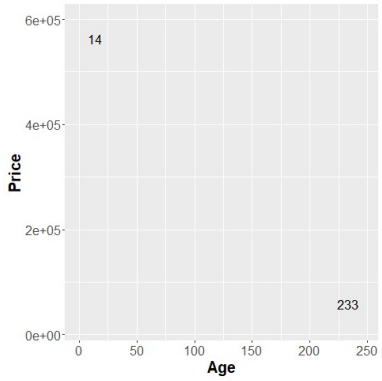


Illustration of location aesthetics

This plot show only two data points, and is labeled with the variable age, corresponding to the X location.

The point on the left is a young house, only 14 years old. The point on the right is an old house, 233 years old.

Fundamentals, Aesthetics for points - location (2 of 2)

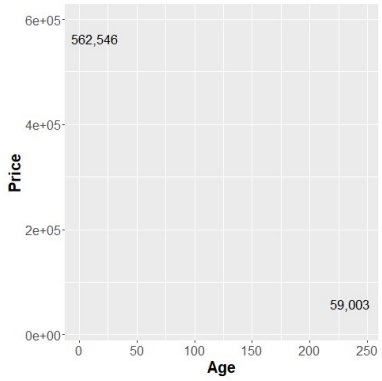


Illustration of location aesthetics

This plot is labeled with the variable Price, corresponding to the Y location.

The point in the up high is an expensive house, over a half million dollars. The point down low a cheap house, about one-tenth of the price.

Exercise, change the location

- Use the Saratoga housing data set.
- Revise the plot so that the location of the points represents X=Bedrooms and Y=Price.

Go back to the plot you just drew. Modify it so that the x location is Bedrooms and not Age.

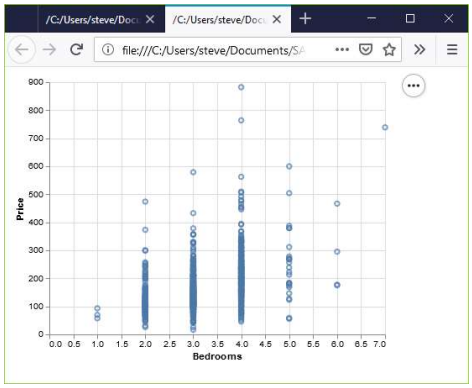
Exercise, Python code

– Here's the Python code.

```
ch = alt.Chart(df).mark_point().encode(
    x='Bedrooms',
    y='Price'
)
```

You specify the x and y locations inside the encode function.

Exercise, Python output



Python scatterplot of bedrooms and price

This is what the Python graph should look like. There is a general upward trend. Houses with more bedrooms tend to cost more.

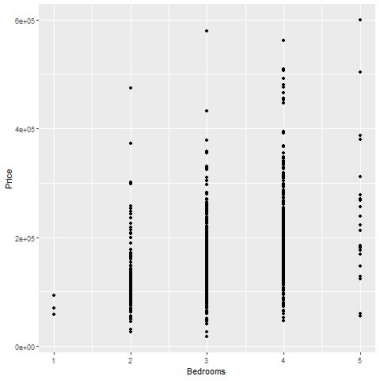
Exercise, R code

– Here's the R code

```
ggplot(
  saratoga_houses,
  aes(x=Bedrooms, y=Price)) +
  geom_point())
```

In R, the X and Y locations go inside the aes function. Normally the aes function sits insider the ggplot function.

Exercise, R output



Scatterplot of bedrooms and price

This is the plot in R. It shows a general upward trend.

Exercise, Tableau steps

- (Take your existing scatterplot of age and price)
- Click on the Age pill
  - Choose the remove option
- Drag Bedrooms to the Columns field
  - Change to Dimension, Continuus

In Tableau, click on Age in the Columns field and choose the remove option. The graph looks a bit weird with no columns, but ignore it. Drag Bedrooms over to the Columns field. Tableau wants to use a sum, but you want individual data points. So click on SUM(Bedrooms) and change it to a Dimension.

Exercise, Tableau output

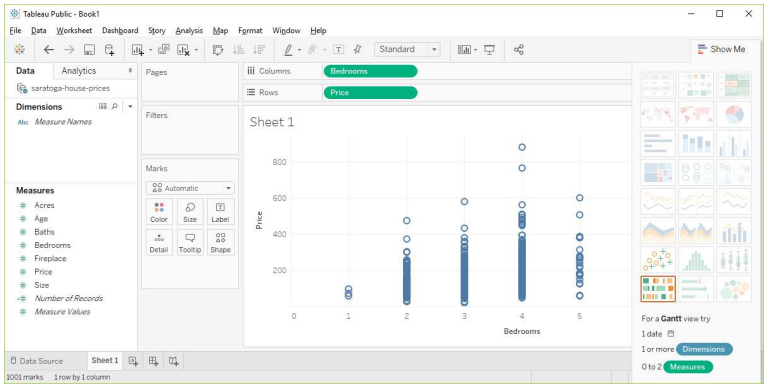


Tableau scatterplot of Bedrooms and Price

Here is the Tableau output.

Fundamentals, Aesthetics for points - shape

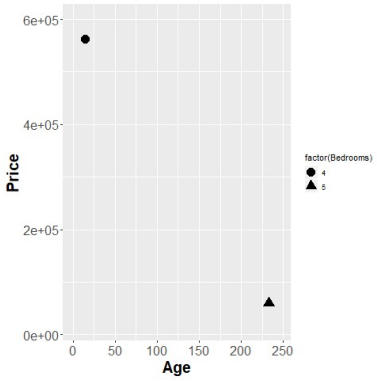


Illustration of shape aesthetics

The house in the upper left corner has four bedrooms, and the house in the lower right corner has five bedrooms. You use circles and triangles to designate this and the legend on the right hand side tells you how to decipher the symbols.

I want to note here that this plot is effectively showing three dimensions, Age, Price, and Bedrooms, even though it is restricted to a two dimensional screen.

I also want to point out that this may not be the best way to visualize the relationship among these three variables.

Exercise, change the shape

- Use the Saratoga housing data set.
- Draw a plot of all of the data where
  - X=Age,
  - Y=Price,
  - Symbol=number of bedrooms.

Revisit your scatterplot. Draw a plot with x representing the age of the house, y representing the price, and the symbol representing the number of bedrooms.

Exercise, Python code

– Here's the Python code.

```
ch = alt.Chart(df).mark_point().encode(
    x='Age',
    y='Price',
    shape='Bedrooms:N'
)
```

In Python, when you want the shape to vary by the levels of a particular variable, you specify this in the encode function. This Python program has three variables that are mapped. Age is mapped to the X location, Price is mapped to the Y location, and Bedrooms is mapped to various shapes.

Bedrooms is on a continuous scale, and you have to tell Python that you want to treat it as if it were nominal.

Exercise, Python output

Python scatterplot mapping bedrooms to shape

This is what the plot looks like. There is a terrible mix of shapes with no obvious pattern.

Exercise, R code

– Here's the R code.

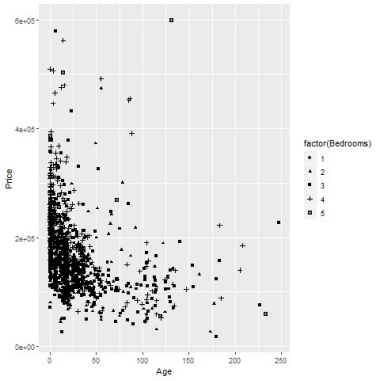
```
ggplot(saratoga_houses, aes(x=Age, y=Price)) +
  geom_point(aes(shape=factor(Bedrooms)))
```

Here is the R code. When the shape of a point is dependent on a variable, you need to specify that variable inside the aes function. You could have defined it inside the existing aes function in the first line of code, but you can also add an aes function to the geom_point function.

I like the second option because it makes the code a bit easier to read.

Bedrooms is numeric, and by default in R is treated as a continuous variable. You can convert it to a nominal variable with the factor function.

Exercise, R output



R scatterplot mapping bedrooms to shape

Here is what the R graph looks like. I don't particularly like this graph. It is confusing, especially with all the overprinting.

Exercise, Tableau output

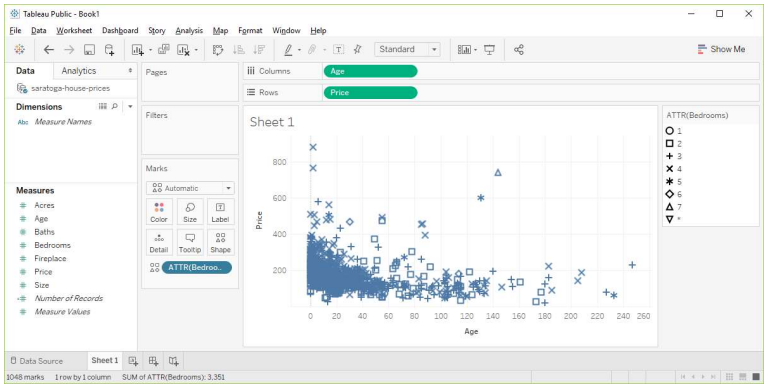


Tableau scatterplot using shapes to represent Bedrooms

Here are the steps in Tableau. First revert to the earlier scatterplot where Age is in the Column fields and Price is in the Rows field. Then drag and drop Bedrooms on top of the Shape icon. Change from SUM(Bedrooms) to Dimension Categorical.

Fundamentals, Aesthetics for points - size

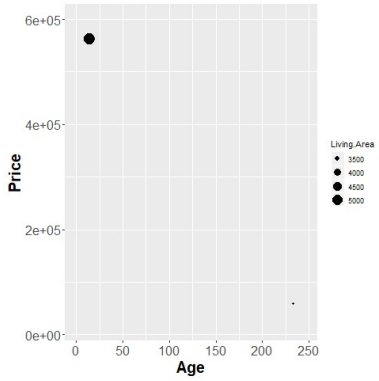


Illustration of size aesthetics

You can also use the size of a point to represent a third dimension. Here is a plot where the larger house, the house with more living area has a big circle and the house with less living area has a small circle.

Exercise, change the size

- Use the Saratoga housing data set.
- Draw a plot of all of the data where
  - X=Age,
  - Y=Price,
  - Size=Living.Area.

Revisit your scatterplot. Draw a plot with x representing the age of the house, y representing the price, and the size representing the living area.

Exercise, Python code

– Here's the Python code

```
ch = alt.Chart(df).mark_point().encode(
    x='Age',
    y='Price',
    size='Living.Area'
)
```

For Altair/Python, you define the variable associated with size in the encode function.

Exercise, Python output

Python scatterplot mapping living area to size

This plot is a bit of a jumble, but you do notice that the very very high prices are all large circles, meaning spacious houses.

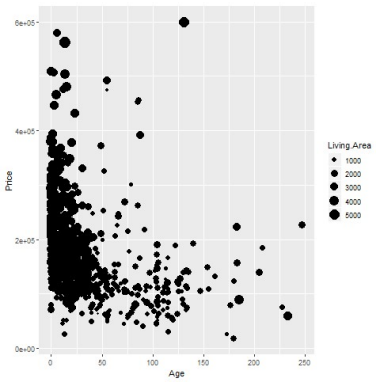
Exercise, R code

– Here's the R code.

```
ggplot(saratoga_houses, aes(x=Age, y=Price)) +
  geom_point(aes(size=Living.Area))
```

With R, you put the variable associated with size in the aes function. Here, I placed this particular aes function in the geom_point function, but it would be okay to specify it as part of the earlier aes function inside the ggplot function.

Exercise, R output



R graph using size to represent Living Area

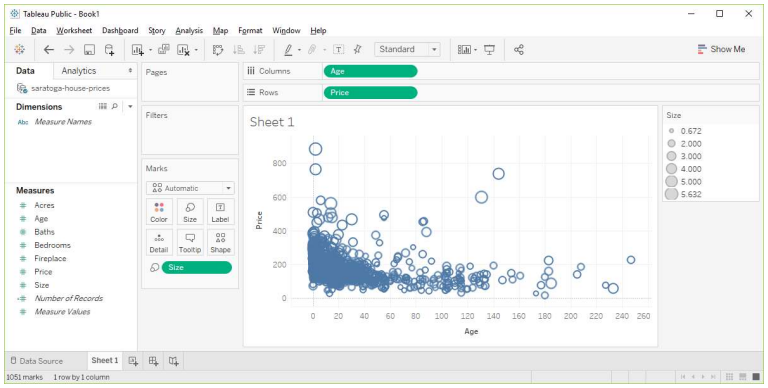
This plot shows the same pattern as earlier. The spacious houses tend to be clustered near the top of the graph, the expensive part of the graph.

Exercise, Tableau steps

- Drag Age to the Columns field, Price to Rows field
  - Change to Dimension, Continous (green pill)
- Drag Living.Area to the Shape button

Here are the steps to create a scatterplot in Tableau where the size of the data points varies.

Exercise, Tableau output



Visualizaion using Living.Area as the size

In Tableau, you drag and drop the Living.Area variable on the Size icon. The results are pretty much the same.

Fundamentals, Aesthetics for points - color (1 of 2)

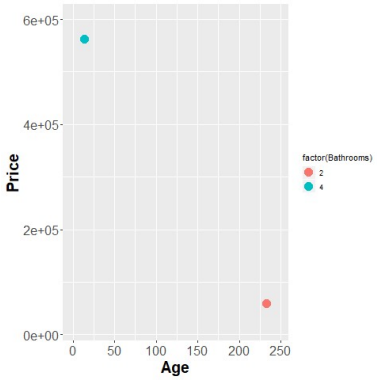
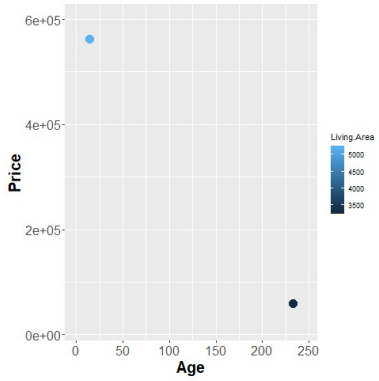


Illustration of color aesthetics

This plot shows a blue point representing a four bathroom house in the upper left corner and an orange point representing a two bathroom house in the lower right corner.

Fundamentals, Aesthetics for points - color (2 of 2)



R graph using color to represent Living.Area

Notice how the legend has changed. All of the graphic packages use a set of discrete, easily distinguishable colors for categorical data. For continuous data, these graphic packages use a gradient.

We'll discuss discrete colors versus gradients in a separate module.

Exercise, change the color

- Use the Saratoga housing data set.
- Draw a plot of all of the data where
  - X=Age,
  - Y=Price,
  - Color=Bathrooms.

Revisit your scatterplot. Draw a plot with x representing the age of the house, y representing the price, and the color representing the number of bathrooms.

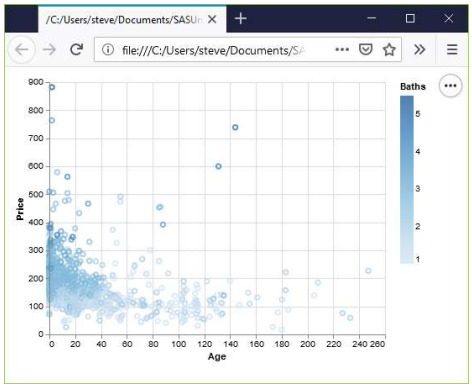
Exercise, Python code

– Here's the Python code.

```
ch = alt.Chart(df).mark_point().encode(
    x="Age",
    y="Price",
    color="Baths")
```

In Altair/Python, the variable associated with different colors goes inside the encode function.

Exercise, Python output



Python scatterplot mapping bathrooms to color

Notice that Altair/Python used a gradient of colors. This is a good choice.

The darker colors, associated with a larger number of bathrooms cluster to the left. Newer houses tend to have more bathrooms.

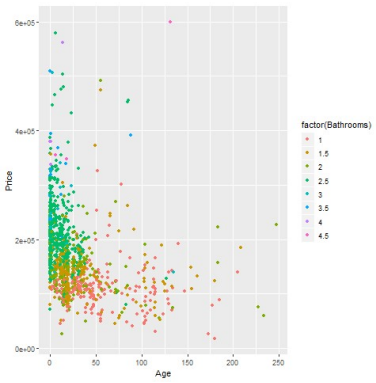
Exercise, R code

– Here's the R code.

```
ggplot(saratoga_houses, aes(x=Age, y=Price)) +
  geom_point(aes(color=factor(Bathrooms)))
```

The color variable is defined inside the aes function.

Exercise, R output



R scatterplot mapping bathrooms to color

I deliberately defined Baths as categorical with the factor function, and notice the use of discrete colors rather than a gradient.

Exercise, Tableau output

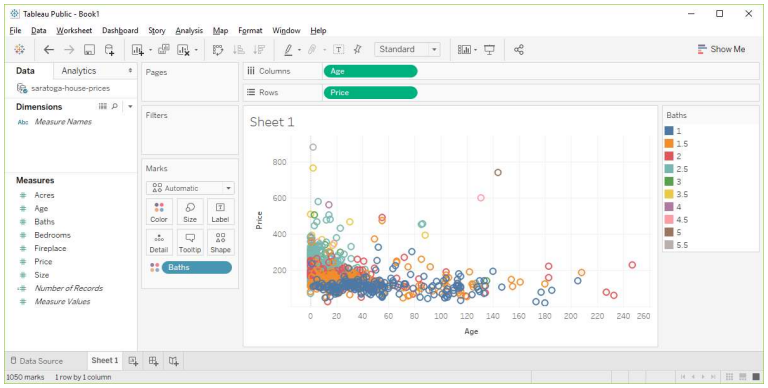


Tableau scatterplot mapping Bathrooms to color

I deliberately defined Baths as Dimension Categorical (blue pill) to show the contrast. Notice that the colors are discrete well separated values.

Fundamentals, summary

- In this section, you learned how to use Python, R, and/or Tableau to
  - Assign variables to the x and y position of a graph
  - Change the defaults for aesthetics like shape and color
  - Assign a third variable to shape, size, or color

Pat yourself on the back because you've alredy learned a lot. You know how to assign variables to the x and y positions of a graph. You can change the default appearance of the points on that graph. You can include a third variable in your graph, represented by shape, size, or color.

(Note to myself) The use of panels is an important aspect of visualization, and it is unclear to me where best to introduce it. Possibly it could be another dimension of location. You have your x position and your y position, but you also have a position within one of several panels.

Recommendations, outline of topics

– Solving problems with overprinting
– Don’t mix shape and size
– Double up for emphasis
– Shape is only good for categories
– Size is only good for continuous variables.

Here are some general recommendations for scatterplots.

First, if you have a lot of data, your scatterplot may look like a large black blob in one corner of your graph with a few stgray points in the other corners. Or it may look like a large elliptical blob in the center of the graph. Either way, there are several strategies to help you see patterns that might otherwise be lost with overprinting.

Another important lesson is to not let both the size and the shape aesthetic to vary within the same graph. At the same time, you can sometimes double up aesthetics, letting both the shape and the color, for example, to both represent the same variable.

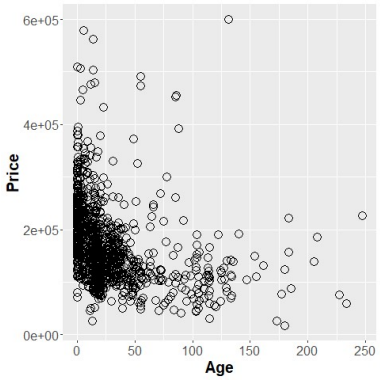
The other important issue is that shape is a useful aesthetic only for categorical data, while size is only good for continuous variables.

Recommendations, Solutions to excessive overprinting

- Open symbols
- Small points
- Opacity
- Log scale

The plots that you have drawn tend to have difficulty with overprinting. Too many data points crammed into a small area produces a large solid blob that is hard to interpret. There are four possible solutions: open symbols, small points, opacity, and a log scale.

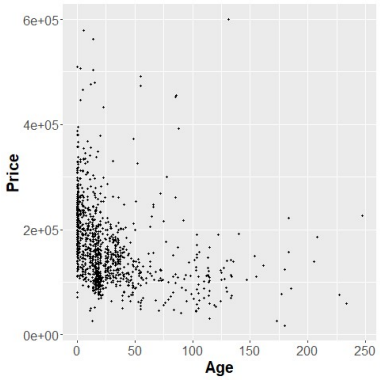
Recommendations, Overprinting - open symbols



R graph using open circles

Open symbols have less trouble with overprinting. They use less ink and it is easier to disentanlge two or three partially overlapping symbols.

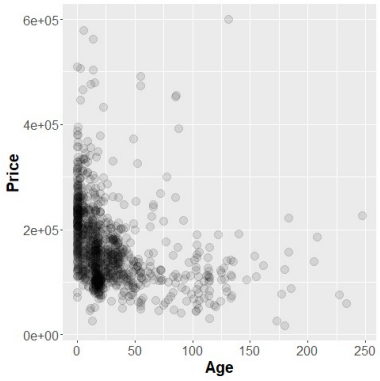
Recommendations, Overprinting - small points



Graph using small points

Smaller data points have less trouble with overprinting.

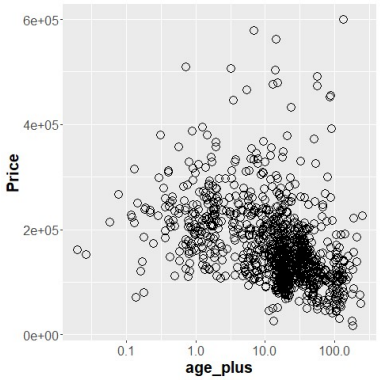
Recommendations, Overprinting - opacity



Graph using opacity

If you make the data points somewhat translucent, so that you can see what it behind it, this prevents a massive black blob from forming. You have to experiement a bit with the level of translucence.

Recommendations, Overprinting - log scale

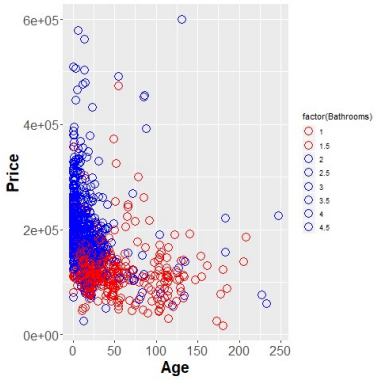


Graph using log scales

The log function stretches the small values and squeeze the large values. Here the log scale is used on the x-axis only and the points that are all piled on top of one another on the left side of the graph are stretched apart. The outlying points on the right, the very, very old houses are fairly rare and waste some of the space that you might need for the far more numerous new houses. The log squeezes these few old houses together. The points are more evenly distributed across the x-axis when you use a log scale, so you can see more details and have less overprinting.

It doesn't work perfectly, and sometimes the log scale is like jumping from the frying pan into the fire. But it is easy enough to try, so what the heck!

Recommendations, Don't try to squeeze in too much.

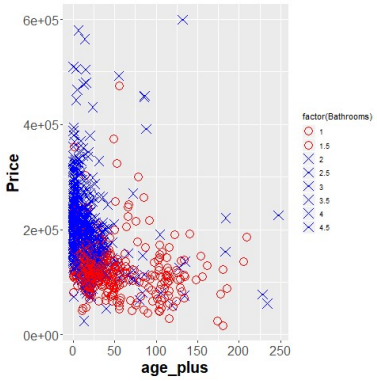


Graph using only two colors

Sometimes less is more, and that is especially true for colors. Rather than assign eight different colors to the eight different number of bathrooms, why not just have two colors. Red for a small number of bathrooms (1.5 or less) and blue for a large number of bathrooms (2 or more).

Sure you are losing some information, but the increasing clarity of the pattern more than compensates. Note that I deliberately included redundant values in the legend to better emphasize the choice I made. I would not recommend this for a final graph.

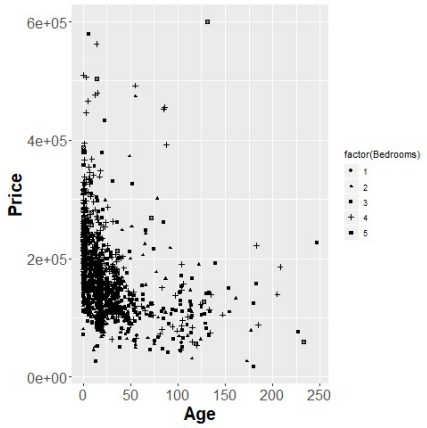
Recommendations, Double up to emphasize



Graph using colors and shapes

It's okay to map a single variable to more than one aesthetic. In fact, this often helps emphasize that variable. Here, both the shape and the color are associated with the number of bathrooms. It makes things a bit easier to pick out.

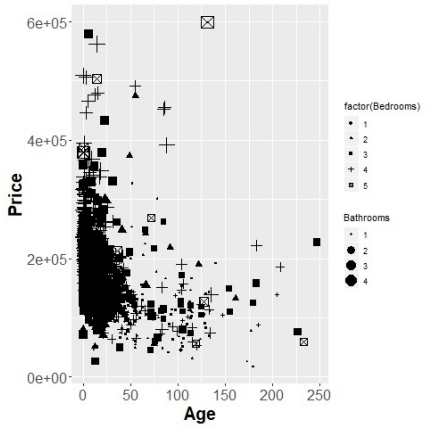
Recommendations, Use shape only for categorical variables



First, shape makes sense only for categorical data. If you have a large number of values, as is typical for a continuous variable, you run out of symbols before you run out of categories.

Even if your continuous variable only has a small number of values (e.g., Bedrooms which can equal 1, 2, 3, or 4 in our data set), don't use shape. Shape has no natural order.

Recommendations, Shape and size don’t mix



Also size and shape don’t mix. You can really compare two diffent shapes, such as an X and an O on size. Is the size related to the diameter, the area, or the perimeter? There’s no really good answer, so it’s asking too much of a graph that asks you to decide which is larger among points represent different shapes. Either vary the shape or vary the size, but not both.

Recommendations, summary

- Overprinting
  - Open symbols
  - Small points
  - Opacity
  - Log transformation
- Don't try to squeeze in too much
- Double up to emphasize
- Shape is for categorical variables
- Size is for continuous variables
- Size and shape don't mix

Quick quiz (1 of 4)

Data visualization is a mapping of data to the visual aesthetics of geometries/marks. Some examples of visual aesthetics include (choose all that apply)

1. Size
2. Points
3. Shapes

Answer: 1 and 3 (size and shapes)

Quick quiz (2 of 4)

The log transformation works by (choose the best answer)

1. Stretching all data values equally.
2. Stretching the small values and squeezing the large values.
3. Stretching the large values and squeezing the small values.

Answer: 2 (strethcing the small values and squeezing the large values)

Quick quiz (3 of 4)

Strategies that can sometimes help when you have a lot of problems with overprinting include (choose all that apply)

1. Using open circles
2. Using large size points
3. Using translucent points

Answer: 1 and 3 (using open circles and translucent points)

Quick quiz (4 of 4)

The two visual aesthetics that doen't work well together are (choose the best answer)

1. color and size
2. color and shape
3. size and shape

Answer: 3 (size and shape)

Graphs in the news, what are the aesthetics?

- What aesthetics (location, shape, size, color) are used?
- What aesthetics are not used?
- What variables are mapped to which aesthetics?

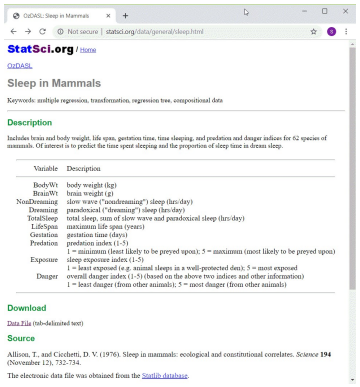
Review the same newspaper article and graph that you used earlier. Identify the various aesthetics that are used or are not used in your graph. What variables are associated with which aesthetics?

Advanced exercise

- There is a second data set on sleep in mammals. You can find a brief description of this data set at
  - http://www.statsci.org/data/general/sleep.html
- You can download the actual data at
  - http://www.statsci.org/data/general/sleep.txt

This is an interesting data set. You can download from the web, or I can provide a comma separated value format for you to use.

Data dictionary



StatSci.org / Home

OzDASL

Sleep in Mammals

Keywords: multiple regression, transformation, regression tree, compositional data

Description

Includes brain and body weight, life span, gestation time, time sleeping, and predation and danger indices for 62 species of mammals. Of interest is to predict the time spent sleeping and the proportion of sleep time in dream sleep.

Variable	Description
BodyWt	body weight (kg)
BrainWt	brain weight (g)
NonDreaming	slow wave ("nondreaming") sleep (hrs/day)
Dreaming	paradoxical ("dreaming") sleep (hrs/day)
TotalSleep	total sleep, sum of slow wave and paradoxical sleep (hrs/day)
LifeSpan	maximum life span (years)
Gestation	gestation time (days)
Predation	predation index (1-5) 1 = minimum (least likely to be preyed upon); 5 = maximum (most likely to be preyed upon)
Exposure	sleep exposure index (1-5) 1 = least exposed (e.g. animal sleeps in a well-protected den); 5 = most exposed
Danger	overall danger index (1-5) (based on the above two indices and other information) 1 = least danger (from other animals); 5 = most danger (from other animals)

Download

Data File (tab-delimited text)

Source

Allison, T., and Cicchetti, D. V. (1976). Sleep in mammals: ecological and constitutional correlates. *Science* **194** (November 12), 732-734.

The electronic data file was obtained from the Statlib database.

Data dictionary for sleep file

This is the data dictionary for this file, from the website listed above.

Advanced exercise

- There are some interesting relationships among the variables. Explore whatever strikes you as interesting. Some possibiilities include
  - bodywt and predation
  - gestation and lifespan
  - exposure and totalsleep
- Draw a visualization
  - illustrates an intersting interrelationships
  - use a third variable for shape, size, or color

Correlations

	BodyWt	BrainWt	NonDreaming	Dreaming	TotalSleep	LifeSpan	Gestation	Predation	Exposure	Danger
BodyWt	1.00	0.93	-0.38	-0.11	-0.31	0.30	0.65	0.06	0.34	0.13
BrainWt	0.93	1.00	-0.37	-0.11	-0.36	0.51	0.75	0.03	0.37	0.15
NonDreaming	-0.38	-0.37	1.00	0.51	0.96	-0.38	-0.59	-0.32	-0.54	-0.48
Dreaming	-0.11	-0.11	0.51	1.00	0.73	-0.30	-0.45	-0.45	-0.54	-0.58
TotalSleep	-0.31	-0.36	0.96	0.73	1.00	-0.41	-0.63	-0.40	-0.64	-0.59
LifeSpan	0.30	0.51	-0.38	-0.30	-0.41	1.00	0.61	-0.10	0.36	0.06
Gestation	0.65	0.75	-0.59	-0.45	-0.63	0.61	1.00	0.20	0.64	0.38
Predation	0.06	0.03	-0.32	-0.45	-0.40	-0.10	0.20	1.00	0.62	0.92
Exposure	0.34	0.37	-0.54	-0.54	-0.64	0.36	0.64	0.62	1.00	0.79
Danger	0.13	0.15	-0.48	-0.58	-0.59	0.06	0.38	0.92	0.79	1.00

Correlation matrix among sleep variables

You might find this table of correlations helpful when deciding what relationships to explore.

Summary

- "A mapping of data to the visual aesthetics of geometries/marks"
  - Points are a type of geometry/mark
  - Aesthetics for points include location, shape, size, color
- Basic tips
  - Don't try to squeeze in too much
  - Double up to emphasize
  - Shape is only good for categories
  - Shape and size don't mix