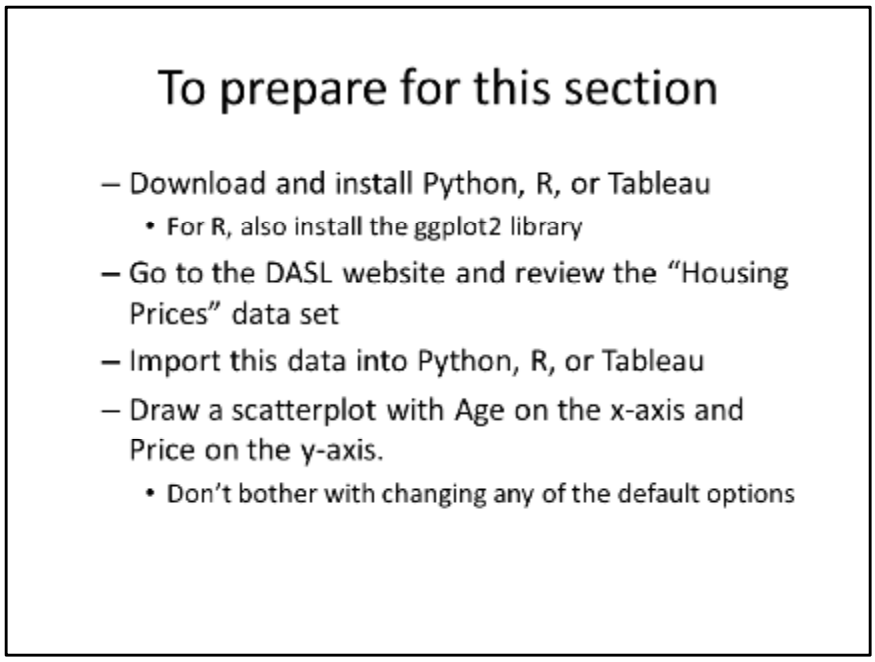
Data visualization - 01 - points

Steve Simon

6/21/2019

Notes about this talk

- This slide should not to be included in the final presentation
- Present 00-overview first
- After this talk, either 02-bars or 03-lines



Please do some work now, to get ready for the work in this section. If you do not already have Python, R, or Tableau installed on your computer, please do so now.

I want you to import a particular data file and draw a simple scatterplot.

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.

Choose “Saratoga House Prices” (not “Saratoga Houses”)

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.

Python code

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

```
((Python code to be added later))
```

R code

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

```
library(ggplot2)
fn <-
"https://dasl.datadescription.com/download/data/3
275"
saratoga_houses <- read.table(fn, header=TRUE,
sep="\x09")
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 make some minor changes to get this to work.

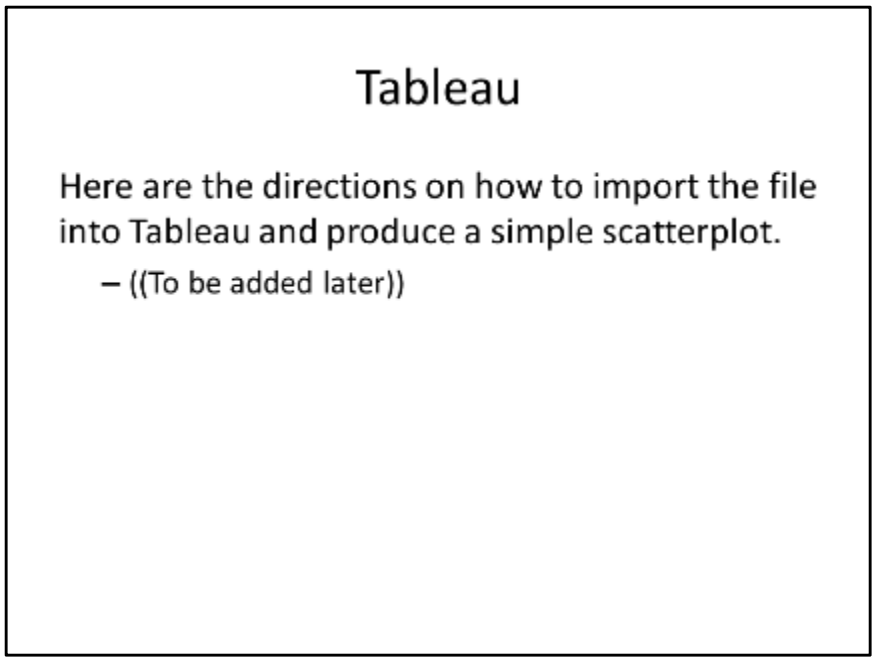
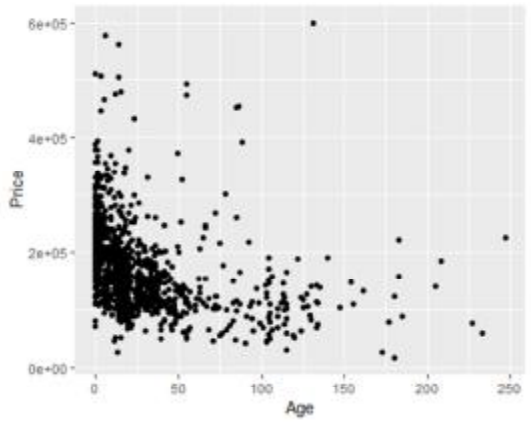
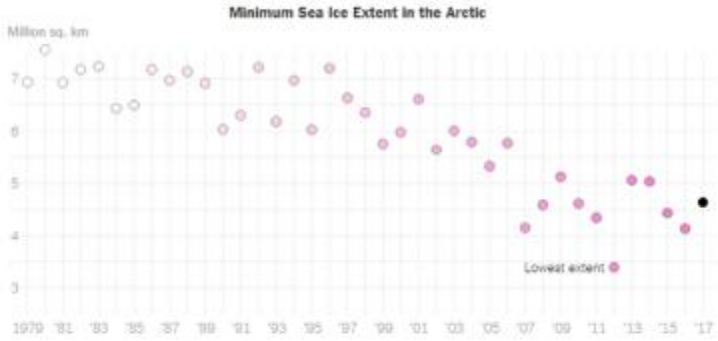


Tableau uses a graphical user interface, so there is no "program" to run. Here are the steps you need to take to get the data in and produce a simple scatterplot.

What your scatterplot should look like (R version)



Group exercise (1 of 2)



Scatterplot of Arctic ice levels

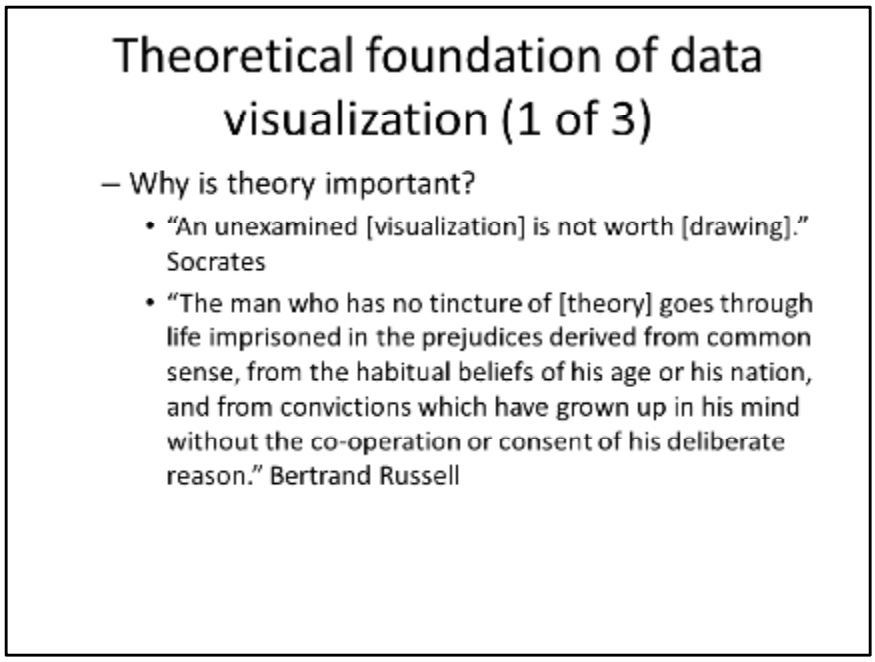
This is one of two graphs. It was published in

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

Split into pairs. Review the article briefly (about 5 minutes) and look at the graph. Explain to your partner what the graph is trying to show. Your partner will get a different graph and do the same thing with you listening this time.

Group exercise (2 of 2)

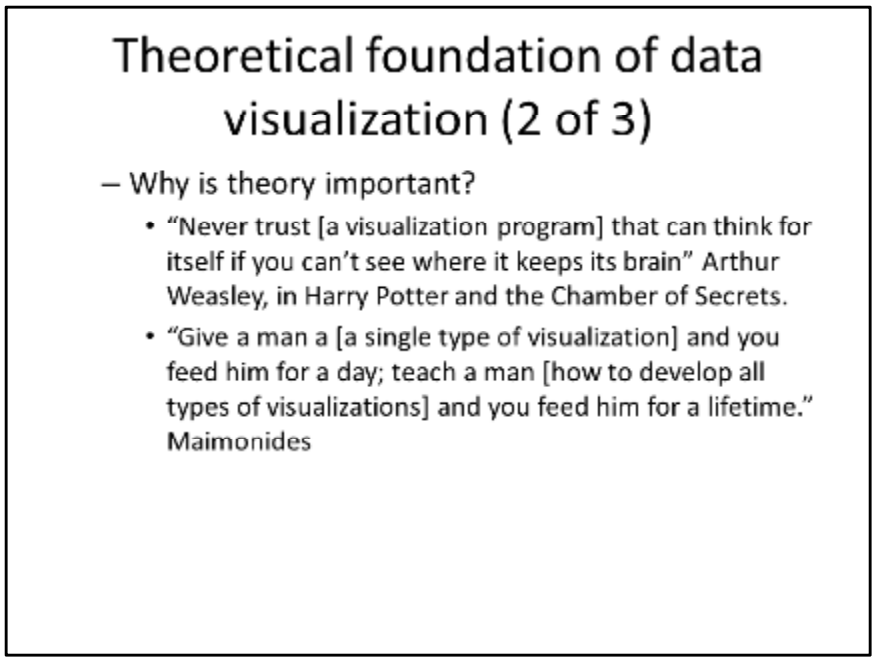
((Image is not yet available.))



I'm a big fan of theory, but I understand that not everyone else is. I found some quotes to support my perspective, although I had to make a few minor corrections.

Like Socrates, I am always curious what is "under the hood." If a visualization works, why wouldn't you want to dig down and figure out why it works?

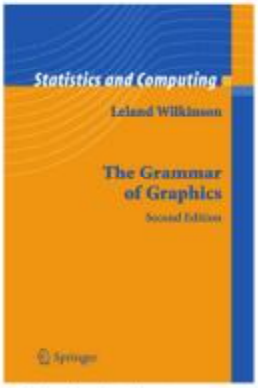
It's more than that, though. Bertrand Russell warns you about the dangers of not having an underlying theoretical framework. Without it, your visualizations will be limited by your own prejudices and convictions that were derived without careful thought or reason.



I also distrust software that makes certain choices for you. It got Ginny Weasley into lots of trouble in J.K. Rowling's Harry Potter and the Chamber of Secrets. I call it the fault of default principle. To be fair, the people who write these visualization software programs are a lot smarter than I am and their choices are on the mark way more often than mine might be. But you have to think carefully about your visualizations and understand where they are coming from. Redraw them as often as you rewrite your text.

But most importantly, Maimonides, the famous Jewish philosopher of the twelfth century, explains the benefits of knowing how and why. If you understand the theoretical framework of visualization, you are capable of drawing a lifetime of different visualizations.

Theoretical foundation of data visualization (3 of 3)



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 inPython, 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

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

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.

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.

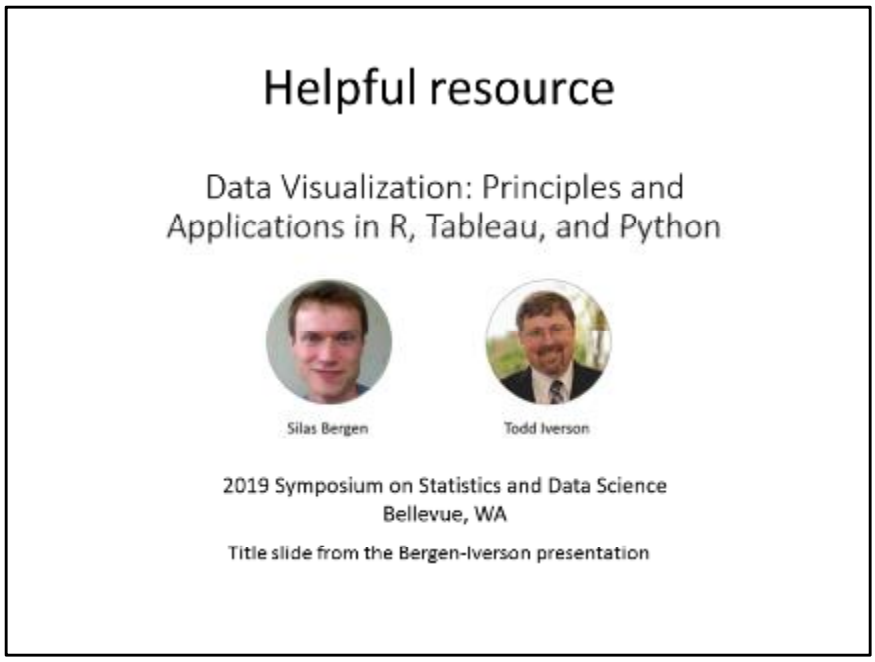
Visualization before Wilkinson (3 of 3)

```
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)
```

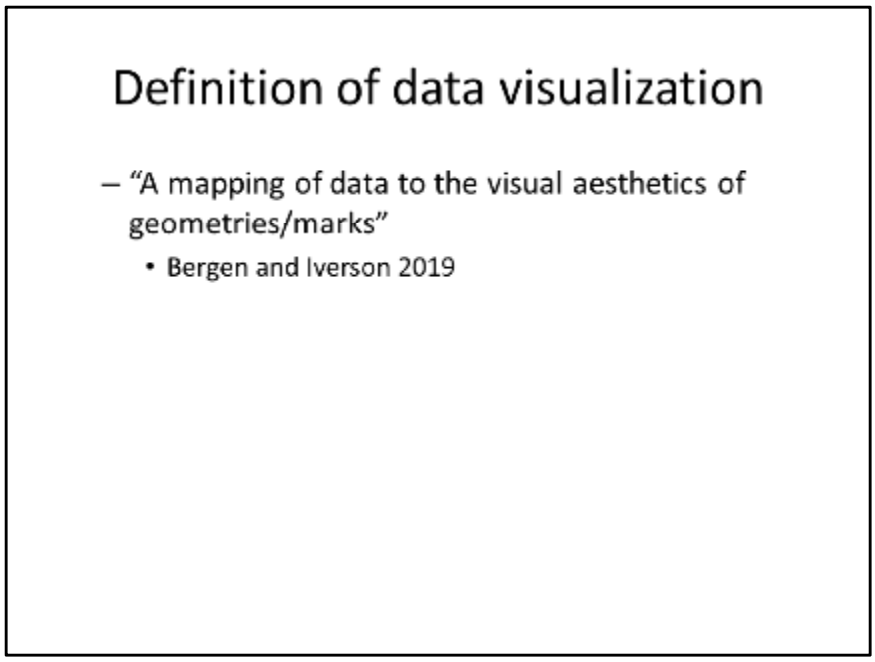
Excerpt from the R help file for the boxplot function

Here's the help function from the program R for the boxplot function. Notice how each function has a different set of arguments, listed in a different order and with different default options. This is only the beginning of the parade of confusion. There is a pie function for pie charts, a contour function for contour plots, a persp3d function for three dimensional surfaces, a 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.



In this section, I am going to borrow heavily from a short course I attended 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 iversion sdss2019 data visualization.



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.

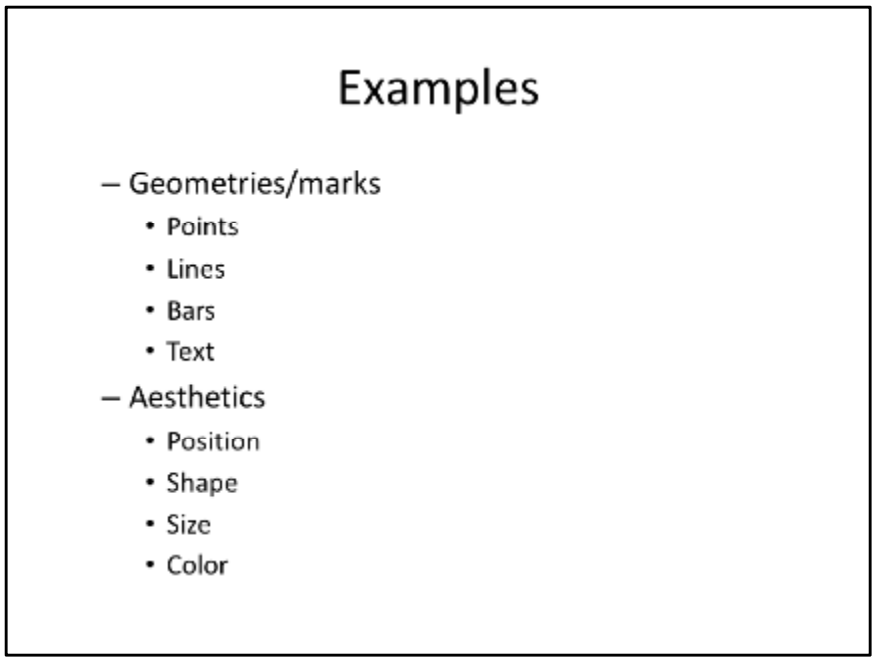
Data. I hope I don't have to define data other than to say that it is an interesting set of numbers. I won't talk about non-numeric data like text in this workshop. Ideally these numbers have enough structure that you can put them into a rectangular grid like a spreadsheet or database table.

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.



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

There are four (more or less) major visual properties of points, lines, bars, and text.

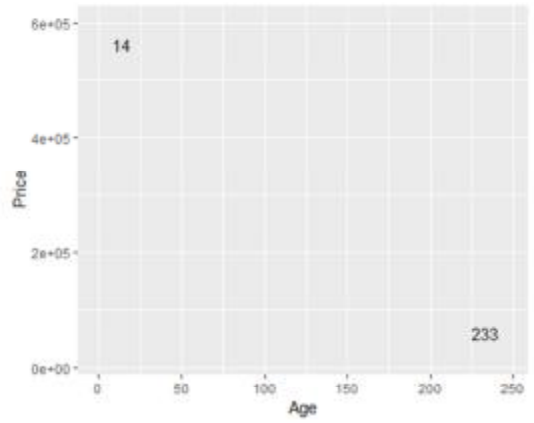
The aesthetics fall into five major classes: position, shape, size, color, and text.

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.

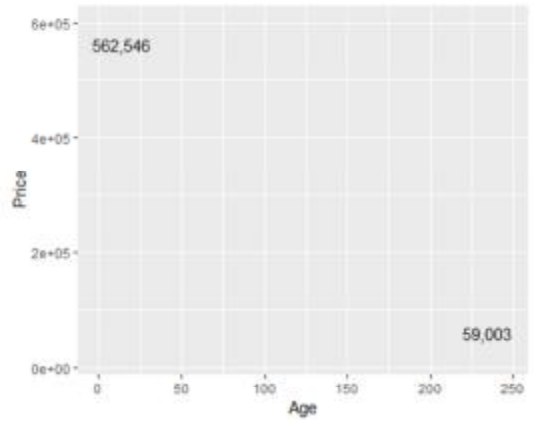
Notes about this talk

- This slide should not be included in the final presentation
- I am considering a flipped classroom approach. If I do this, the flipped part of the classroom would end here and the in class portion would start after the next slide, except I would have a brief review first.

Aesthetics for points - location (1 of 2)



Aesthetics for points - location (2 of 2)



On your own

– Revise the plot so that the location of the points represents x=Bedrooms and Y=Price.

Wait before showing

Here's the Python code.

```
((To be added later))
```

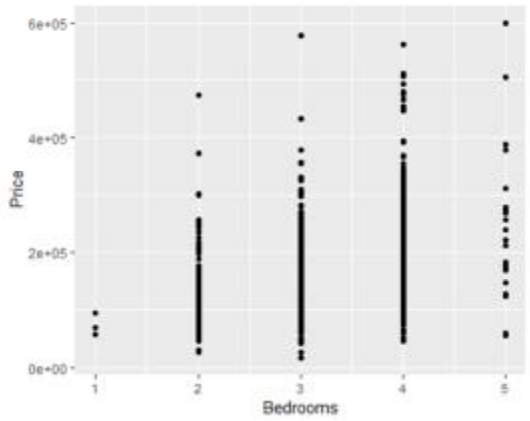
Here's the R code.

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

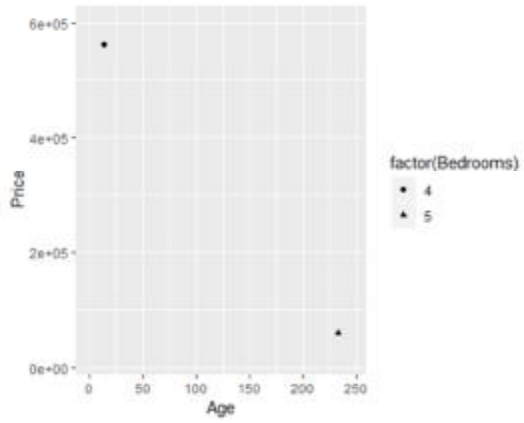
Here are the steps in Tableau.

((To be added later))

What your visualization might look like.



Aesthetics for points - shape



On your own

– Draw a plot where the location is x=Age and y=Price and the symbol represents the number of bedrooms.

Wait before showing

Here's the Python code.

```
((To be added later))
```

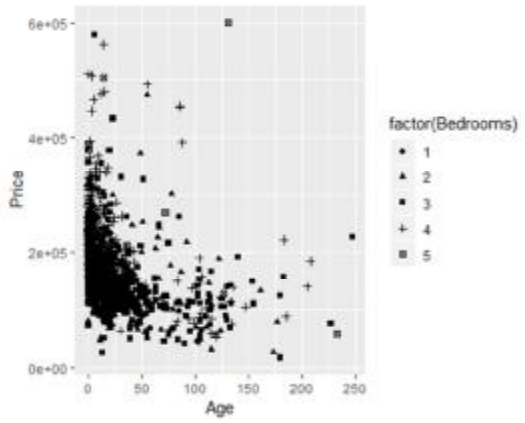
Here's the R code.

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

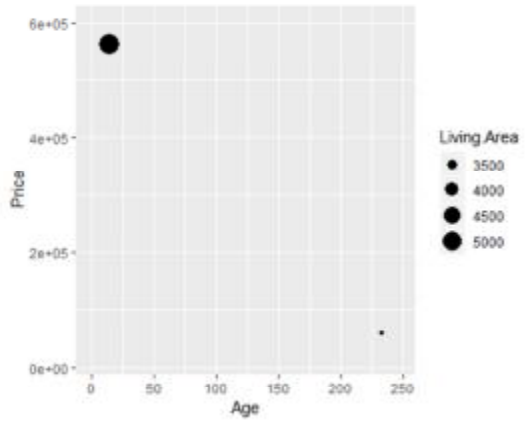
Here are the steps in Tableau.

((To be added later))

What your visualization might look like.



Aesthetics for points - size



On your own

– Draw a plot where the location is x=Age and y=Price and the size represents the living area.

Wait before showing

Here's the Python code.

```
((To be added later))
```

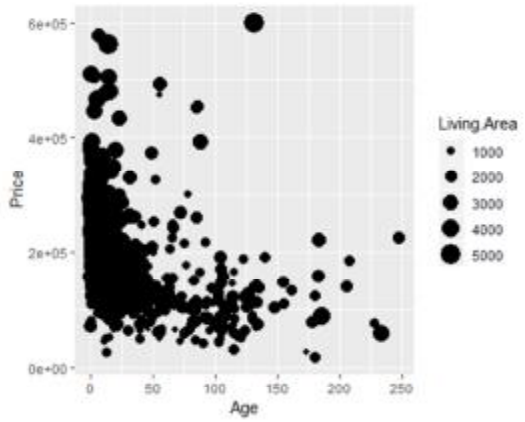
Here's the R code.

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

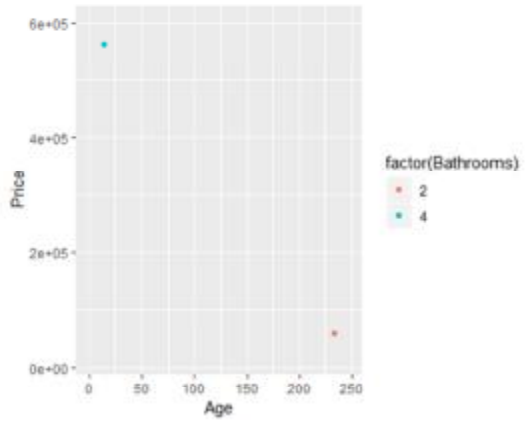
Here are the steps in Tableau.

((To be added later))

What your visualization might look like.



Aesthetics for points - color (1 of 2)



On your own

- Draw a plot where the location is x=Age and y=Price and the color represents the number of bathrooms.

Wait before showing

Here's the Python code.

```
((To be added later))
```

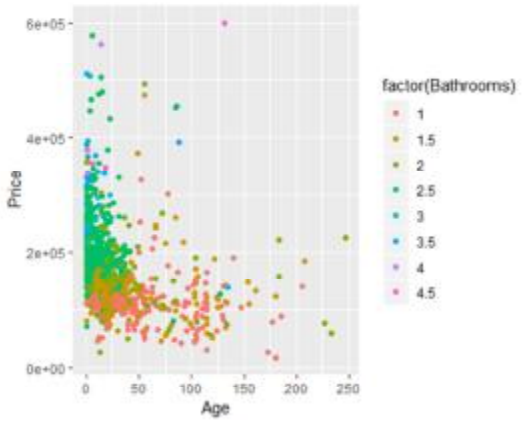
Here's the R code.

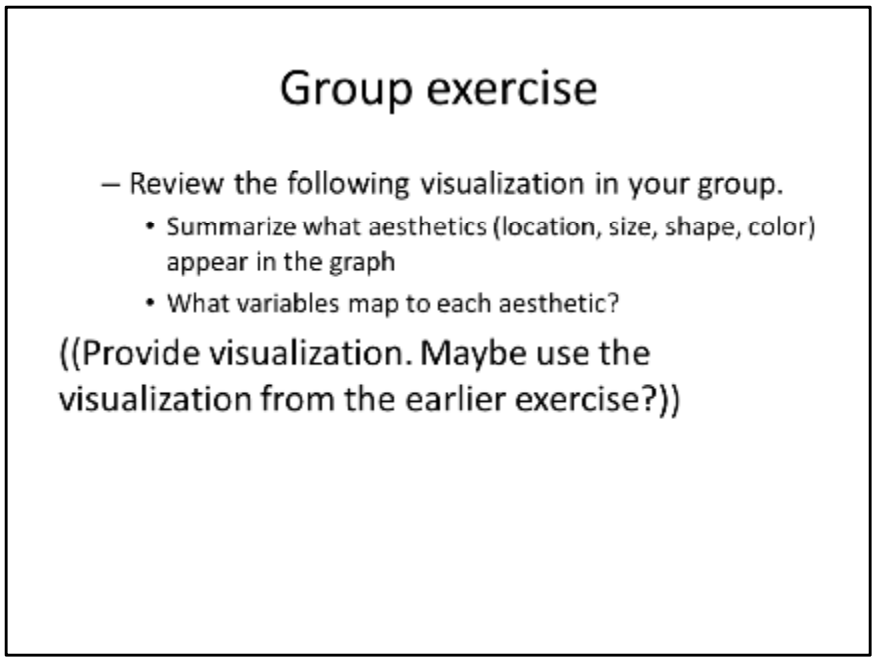
```
ggplot(saratoga_houses, aes(x=Age, y=Price)) +
  geom_point(aes(color=factor(Bathrooms)))
```

Here are the steps in Tableau.

((To be added later))

What your visualization might look like.





I want you to review the visualizations that you discussed earlier. With you partner review the visualization again. Talk about the aesthetics and what variables map to each aesthetic.

Panels

((Show an example of panels and explain how they work))

Some 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

Don’t try to squeeze in too much.

((Show and example with four variables: shape and color))

Double up to emphasize

((Show an example where shape and color are mapped from the same variable))

Shape is only good for categories

((Explain why))

Shape and size don't mix

((Explain why))

On your own

((Find a totally different data set and get the students to draw four different visualizations. Have them divide into groups that like the same visualization software and have each person do a different visualization.))

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