Introduction to Visualization

Professor Jeff Saltz
Professor Jeff Stanton

ischool.syr.edu
School of Information Studies
SYRACUSE UNIVERSITY

Data Science in the Real World

Whistleblower Unfriends Facebook

Data scientist urges oversight to alleviate social network's harm.

(Philly Inquirer, Oct 6th 2021)

Summary of Previous Learning

What you should know and be able to do at this point :

1. List major skills needed by data scientists
2. Describe a DS project with domain analysis and SMEs
3. Explain basic concepts of data modeling
4. Explain and use a data frame in R
5. Define the most common descriptive statistics and calculate them with R using appropriate functions
6. Demonstrate the development of a simple function in R
7. Describe the effects of randomness on sampling
8. Create and interpret a sampling distribution including defining the law of large numbers and the central limit theorem
9. Visualize a distribution and interpret a histogram
10. Describe multiple strategies for accessing external data from R
11. Use sqldf and/or other SQL query facilities from within R

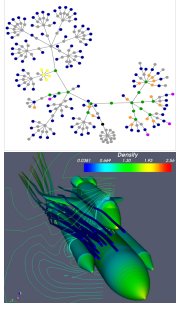
Learning Objectives for Today

- Explain the purposes and uses of visualization (by exploring some examples)
- Describe the most essential methods of representing numeric data visually
- Describe and put into practice the basic principles of visualization
- Generate basic visualizations using the ggplot2 package in R

Visualization Overview

Definitions

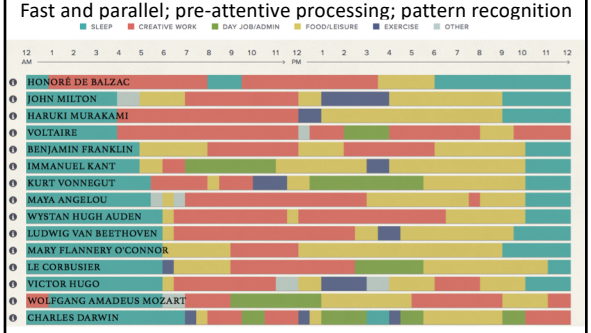
- *Information visualization* can be defined as, "**The creation and use of interactive visual representations of abstract data**."
- Distinctive from *scientific visualization* which is specifically concerned with data that have a well-defined representation in dimensional space



*Adapted from The ParaView Tutorial, Moreland

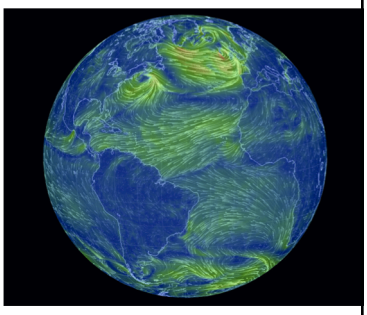
Vision is "Highest Bandwidth" Sense

Fast and parallel; pre-attentive processing; pattern recognition



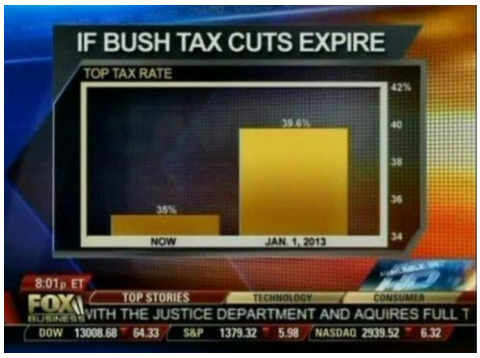
Visualization Components

- Color
- Size
- Texture
- Proximity
- Annotation
- Interactivity
  - Selection and filtering
  - Zoom
  - Animation

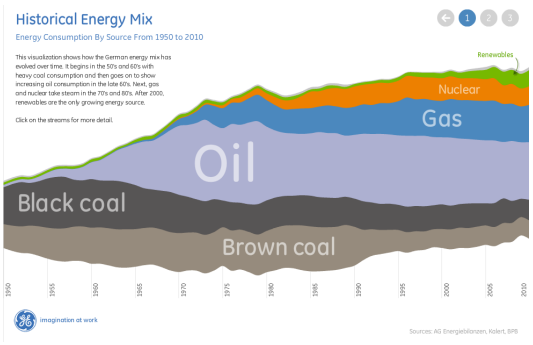


Analyzing Visualizations

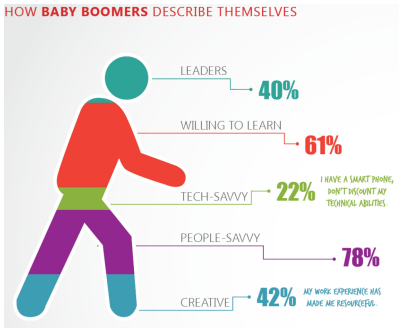
Bar Charts



Area Charts

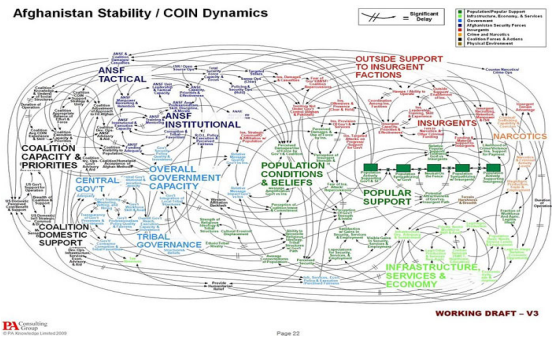


Area Charts



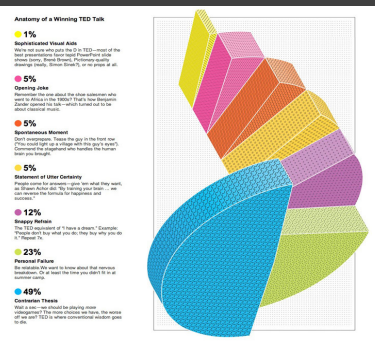
http://wpengine.com/blog/ugly-side-data-visualization/

Influence Flow Diagrams



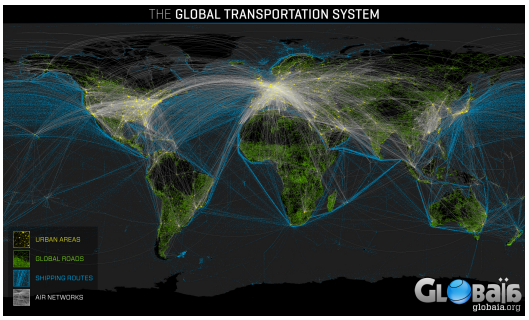
http://cdn.wpengine.com/wp-content/uploads/2014/03/Afghanistan-Stability-COIN-Dynamics.jpg

Pie Chart/Spiral Graph



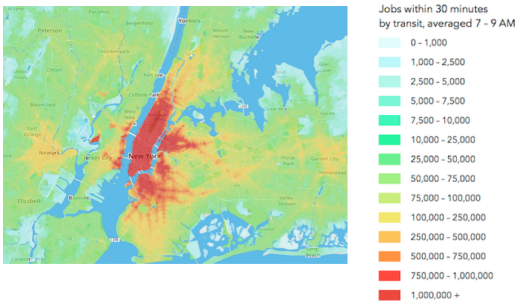
http://cdn.wpengine.com/wp-content/uploads/2014/03/Anatomy-of-a-winning-TED-Talk.jpg

Maps



http://globaia.org/portfolio/cartography-of-the-anthropocene/

Maps



http://www.wired.com/2014/10/americas-10-best-cities-commuting-public-transit/

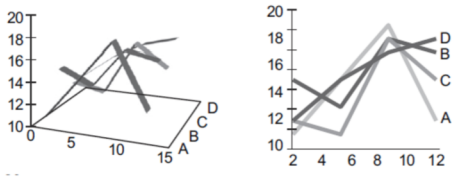
Ten Principles of Visualization

Adapted from Kelleher and Wagener (2011)

School of Information Studies
SYRACUSE UNIVERSITY
ischool.syr.edu

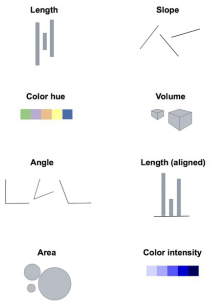
1. Simplicity

- Create the simplest graph that provides the information you want to convey (Tufte).



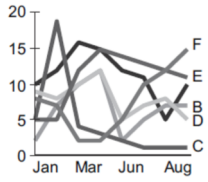
2. Encoding

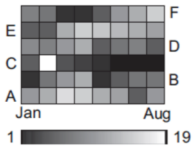
- Choose encoding that fits the attribute(s) you are trying to display (Chambers, 1983; Cleveland & McGill, 1984).



3. Patterns or Details

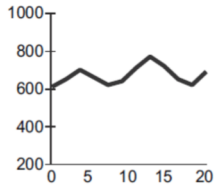
- Focus on visualizing patterns **or** on visualizing details (Few, 2004; Kosslyn & Chabris, 1992).

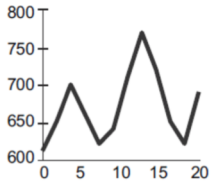




4. Truthful Ranges

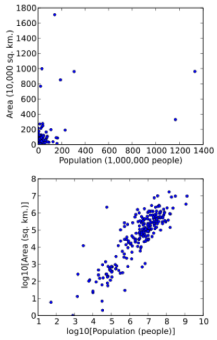
- Select meaningful axis ranges but avoid deceiving the viewer (Robbins,2005; Tufte, 2006; Strange, 2007)





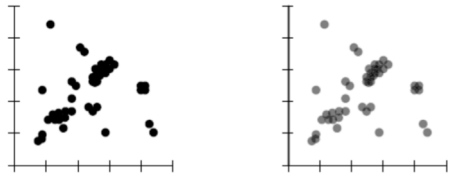
5. Scale Transformations

- Scale transformations can be used to improve interpretability (Cleveland, 1994 [p. 66, 95, 103]).



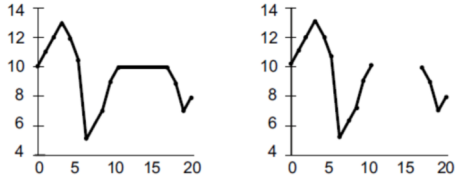
6. Show Density

- Plot overlapping points in a way that **density differences become apparent** (Few, 2009; Cleveland, 1994).



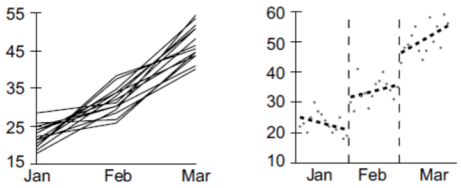
7. Make Connections

- Use lines when connecting sequential data in time-series plots (Strange, 2007).



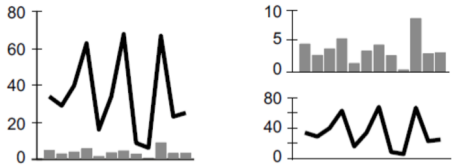
8. Meaningful Aggregation

- Aggregate data sets in meaningful ways (Cleveland & Devlin, 1980; Chambers, 1983)



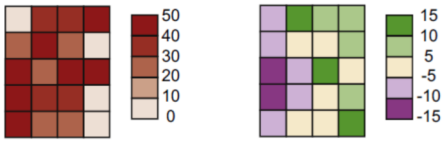
9. Facilitate Comparisons

- **Keep axis ranges similar** to compare variables accurately (Cleveland, 1994; Few, 2009).



10. Choose Sensible Colors

- Select an appropriate color scheme based on the type of data (Brewer, 1994; Harrower and Brewer, 2003).



Footnote on WCAG 2.0

- When designing visualizations, remember that about 8% of men (and 0.5% of women) have some form of colorblindness
- WCAG 2.0 is the Web Accessibility Content Guidelines, an international standard promoted by W3C
- *1.4.1 Use of Color: Color is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element. (Level A)*

genetics.thetech.org

The ggplot2 package

Professor Jeff Saltz

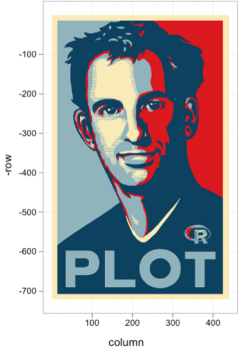
Professor Jeff Stanton

School of Information Studies
SYRACUSE UNIVERSITY

ischool.syr.edu

Hadley Wickham

- Chief scientist at R-Studio and self described "data nerd"
- GG = Grammar of Graphics
- ggplot2 released in 2007, has more than 2M downloads, most popular of the packages he wrote
- Part of Tidyverse



ggplot2

ggplot2 divides plot into three different fundamental parts:

Plot = Data + Aesthetics + Geometry

Every plot can be defined as follows:

- **Data** is a data frame.
- **Aesthetics** is used to indicate x and y variables. It can also be used to control the color, the size or the shape of points, the height of bars, etc.
- **Geometry** defines the type of display (histogram, box plot, line plot, density plot, dot plot, etc.)

Work with the Built-In mpg Data

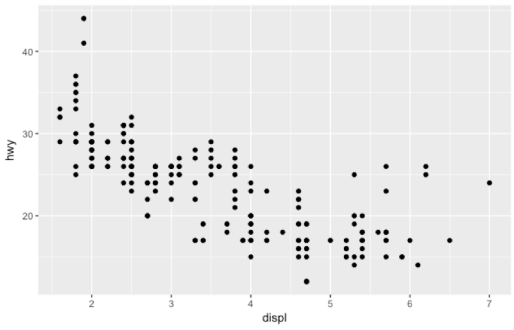
```
View(mpg)
```

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
	column 0: rownames										
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
4	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
7	audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
8	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
9	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
10	audi	a4 quattro	2.0	2008	4	manual(m6)	4	20	28	p	compact
11	audi	a4 quattro	2.0	2008	4	auto(s6)	4	19	27	p	compact
12	audi	a4 quattro	2.8	1999	6	auto(l5)	4	15	25	p	compact

Layering ggplot Specifications

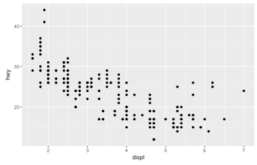
```
# The data
myPlot <- ggplot(mpg)

# The aesthetic
myPlot <- myPlot + aes(x=displ,y=hwy)

# The geometry
myPlot <- myPlot + geom_point()

myPlot # Invoke the plot to draw it
```

ggplot(mpg) + aes(x=displ, y=hwy) + geom_point()

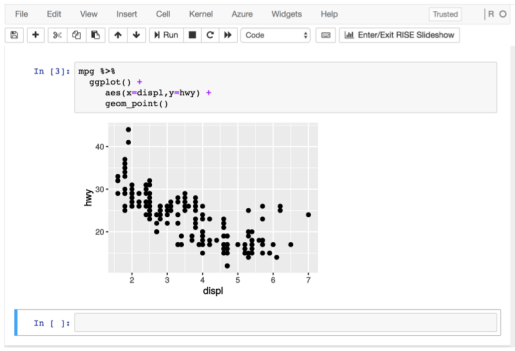


Using Tidyverse Style

```
# Using Pipes and the '+'
# to generate a scatter plot
mpg %>%
  ggplot() +
     aes(x=displ,y=hwy) +
     geom_point()
```

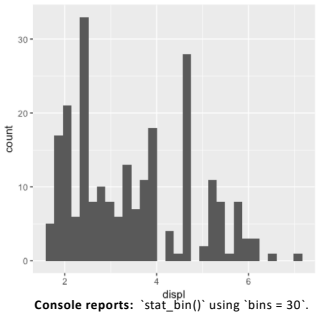


Using Tidyverse in a Notebook



Univariate Display

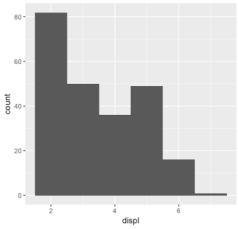
```
# Historgram
mpg %>%
  ggplot() +
  aes(x=displ)+
  geom_histogram()
```



Console reports: `stat_bin()` using `bins = 30`.
Pick better value with `binwidth`.

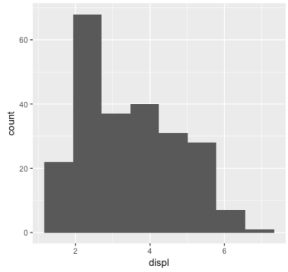
Univariate Display: Control Binwidth

```
# Define the logical
# unit for bins


mpg %>%
  ggplot() +
  aes(x=displ)+
  geom_histogram(binwidth=1)
```



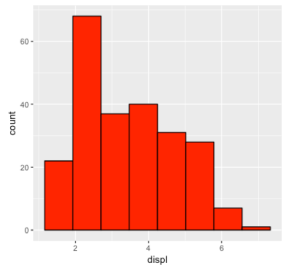
Univariate Display: Set Bin Count

```
# Define the
# Number of bins


mpg %>%
  ggplot() +
  aes(x=displ)+
  geom_histogram(bins=8)
```



Change Colors

```
# Define outline
# and fill color
```



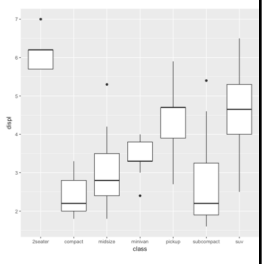

```
mpg %>%
  ggplot() +
  aes(x=displ)+
  geom_histogram(bins=8,
      fill="red",col="black")
```

Note ggplot Code Can Fit on One Line

```
#Rather than storing the plot specs and building
up piece by piece, you can combine everything
into one command line.
ggplot(mpg)+aes(x=displ) +
   geom_histogram(bins=8, fill="red",
                                  col="black")
```

Box Plot

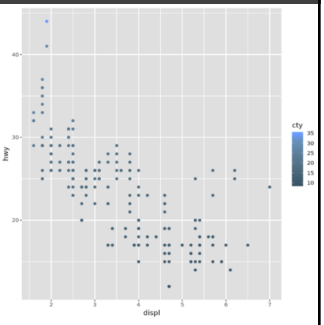
```
#Make a boxplot of:
# displ (y-variable)
# cars class (x-variable)
```



```
mpg %>%
  ggplot() +
    aes(x=class,y=displ) +
    geom_boxplot()
```

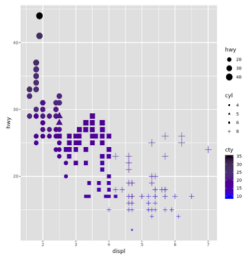
More Scatter Plots - Adding Color

```
#Fill the points
mpg %>%
  ggplot() +
  aes(x=displ,y=hwy) +
  geom_point(
      aes(color = cty))
```



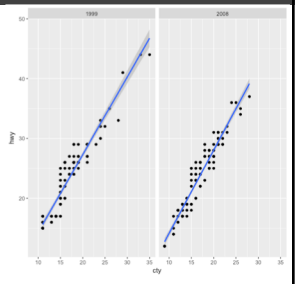
Adding More Attributes

```
#Why the need to mutate?
mpg %>%
  mutate(cyl=as.factor(cyl)) %>%
  ggplot() +
    aes(x=displ,y=hwy) +
    geom_point(aes(color = cty,
        shape=cyl, size=hwy)) +
        scale_color_gradient(
              low = "blue",
              high = "black")
```



Dual Scatterplot With Fitted Line

```
# Define 'facet'
# which is the multiple
# plots
```

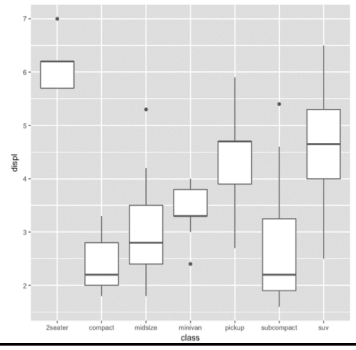



```
mpg %>%
  ggplot() +
  aes(x=cty,y=hwy) +
  geom_point() +
  facet_wrap(~year) +
  geom_smooth(method="lm")
```

Your Turn To Think!

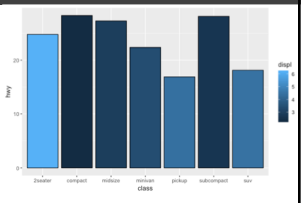
- Make a boxplot of **displ** (y-variable) comparing cars by **class** (x-variable).
- Important reminders:
  - geom_boxplot() is generally a bivariate plot, with a "factor" (grouping) variable on the x-axis.
  - Include both x and y in your aes() statement

ggplot(mpg)+ aes(x=class, y=displ)+geom_boxplot()



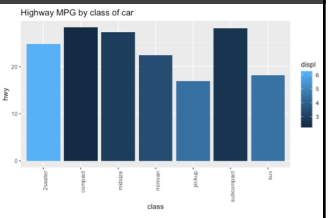
Bar Charts

```
# Use group_by
# to create data
#        for bars
```




```
mpg %>%
  group_by(class) %>%
  summarize(hwy=mean(hwy),
            displ=mean(displ)) %>%
  ggplot(aes(x = class, y=hwy))  +
  geom_col(color="black",
      aes(fill=displ))
```

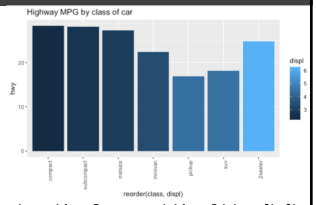
Bar Charts – Rotate Text




```
# Use theme
# to rotate text
# also define title
```

```
mpg %>%
  group_by(class) %>%
  summarize(hwy=mean(hwy), displ=mean(displ)) %>%
  ggplot(aes(x = class, y=hwy))  +
      geom_col(aes(fill=displ)) +
      theme(axis.text.x =
            element_text(angle = 90, hjust = 1)) +
      ggtitle("Highway MPG by class of car")
```

Bar Charts – reorder the columns




```
# Use 'reoder'
```

```
mpg %>%
  group_by(class) %>%
  summarize(hwy=mean(hwy), displ=mean(displ))  %>%
  ggplot(aes(x = reorder(class, displ), y=hwy))  +
  geom_col(aes(fill=displ)) +
  theme(axis.text.x =
          element_text(angle = 90, hjust = 1)) +
  ggtitle("Highway MPG by class of car")
```

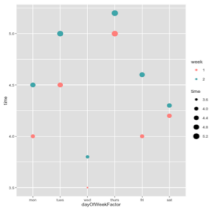
Explore a Timed Task

Define *'travel.df'* to be the dataset

	dayOfWeekFactor	time	week
1	mon	4.0	1
2	tues	4.5	1
3	wed	3.5	1
4	thurs	5.0	1
5	fri	4.0	1
6	sat	4.2	1
7	mon	4.5	2
8	tues	5.0	2
9	wed	3.8	2
10	thurs	5.2	2
11	fri	4.6	2
12	sat	4.3	2

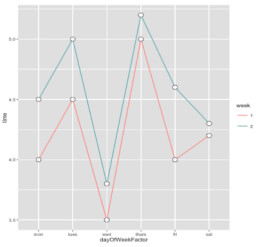
Show Points via a Scatter Plot

```
travel.df  %>%
  ggplot(
     aes(x=dayOfWeekFactor,
         y=time)) +
     geom_point(aes(size = time,
                    color=week))
```



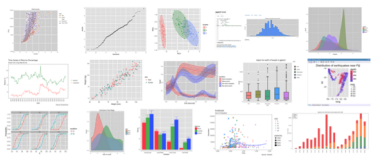
Show Line Plots

```
g <- ggplot(travel.df,
       aes(x = dayOfWeekFactor,
       group=week, color=week)) +
       geom_line(aes(y = time))

g <- g + geom_point(y=time,
        colour="black",
        size=4, shape=21, fill="white")

g <- g + ylab("time to NYC (in hours)") +
       ggtitle("compare weekly times")

g
```



Unexplored Potential

- Using color and marker shapes; time series; labels; titling; grouping; mapping (next week)



The Basic Geometries

- geom_histogram() – histograms (uni)
- geom_col() – bar plot (uni)
- geom_line() – lines connecting points (multi)
- geom_point() – points, scatterplots (multi)
- geom_jitter() – points that are slightly moved
- geom_boxplot() – grouped distributions (multi)
  - geom_violin() – like boxplot, but showing density
- geom_smooth() – fit a line or curve to a plot