Data Visualization. Exploratory Data Analysis

Seeing what's inside our data and allowing others to see

Yordan Darakchiev

Technical Trainer

Software University

SoftUni

Software University

https://softuni.bg

Have a Question?

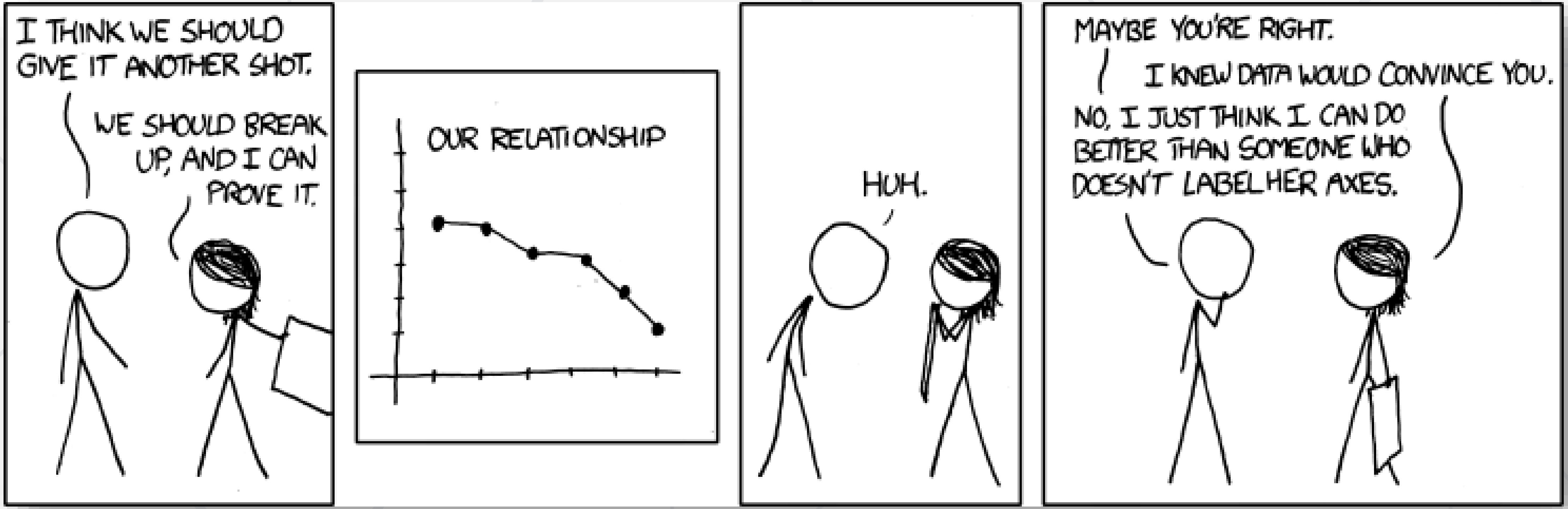
Software University

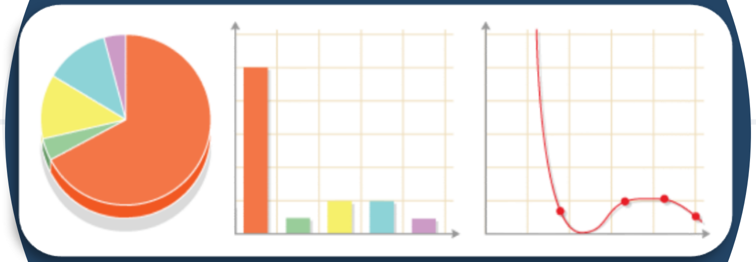
sli.do

#DataScience

Table of Contents

Be Careful…


I THINK WE SHOULD GIVE IT ANOTHER SHOT.
WE SHOULD BREAK UP, AND I CAN PROVE IT.
OUR RELATIONSHIP
HUH.
MAYBE YOU'RE RIGHT.
I KNEW DATA WOULD CONVINCE YOU.
NO, I JUST THINK I CAN DO BETTER THAN SOMEONE WHO DOESN'T LABEL HER AXES.

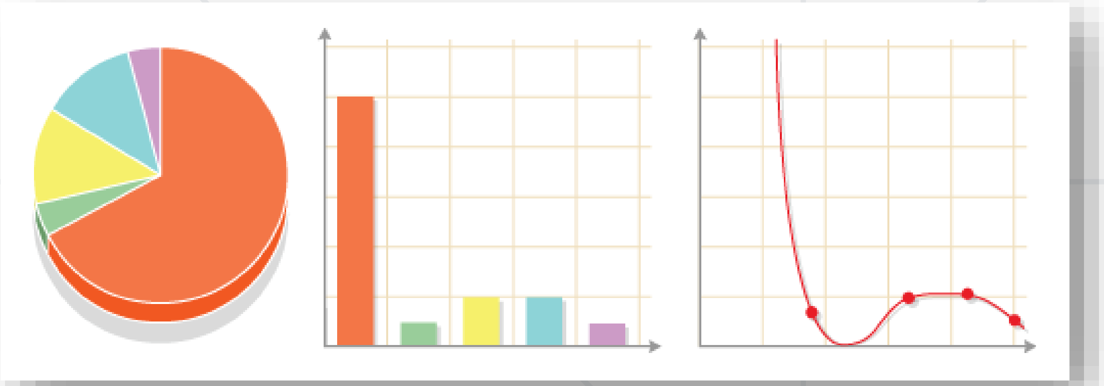


Main Concepts in Data Visualization

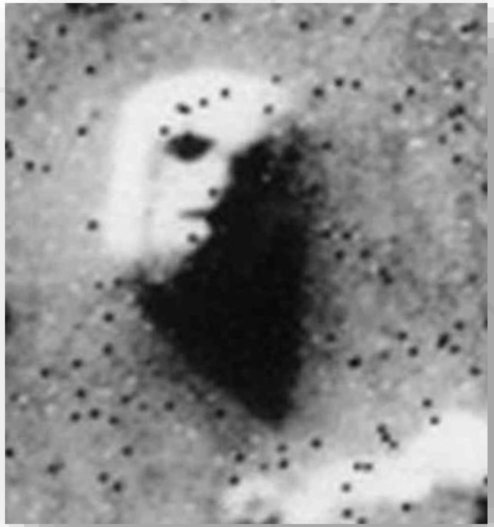
How to Tell the Right Story

Data Visualization

- We're amazingly good at spotting patterns
  - Trends over time, correlations, comparisons, ranges, etc.
- Visualizing data helps us understand the information better



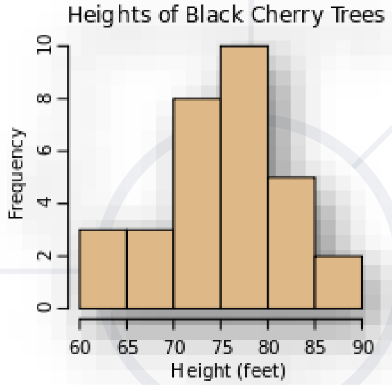
- By plotting different views on the data we can
  - Help ourselves explore and understand the data
  - Convey the stories in data to others
- Note that we're too good at spotting patterns
  - We can find patterns where they don't exist

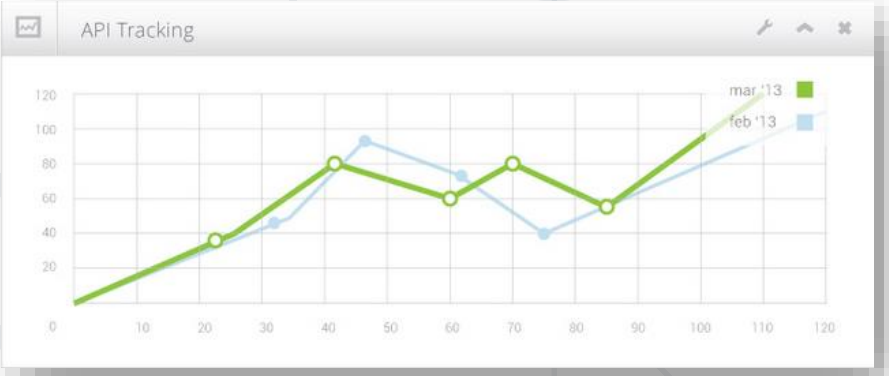


Knowing What (and When) to Plot

Software University

- Many types of graphs, each with its own purpose
  - **Histograms** – show distributions
  - **Boxplots** – show the range and skewness of values
  - **Bar charts** – show how different categories compare
  - **Line plots** – show how one (dependent) variable varies with respect to an independent variable (e.g. over time)
  - **Pie charts :(** – show relative sizes between parts of a whole
  - Don't forget that we can also display **single numbers** when they provide sufficient information





Knowing What (and When) to Plot

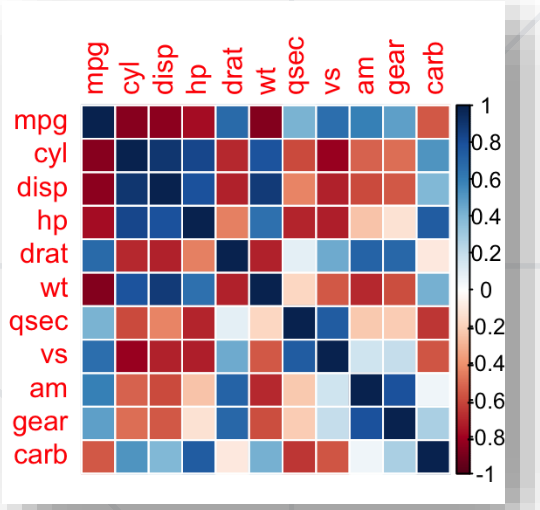
- Many more types of graphics depending on the context
- Choosing the right plot is a matter of **intuition**
  - The goal is to **present the message clearly**
    - I.e., "**tell the right story**"
- Two main kinds of visualizations
  - For scientific analysis and work – stricter rules
    - For exploratory analysis / quick references
  - For presenting results to non-specialists – we can be creative but we have to keep our message in mind
  - The results may be printed or viewed as a **dashboard**
- How many dimensions?
  - Each plot has two spatial dimensions, but we can add more using color, size, even animation

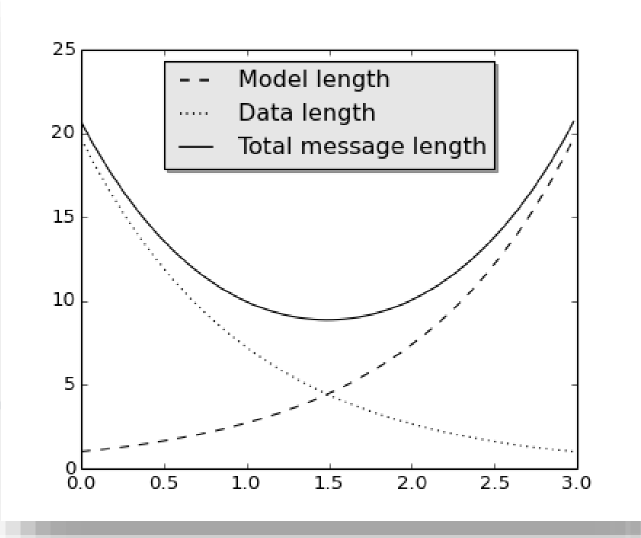
Knowing What (and When) to Plot

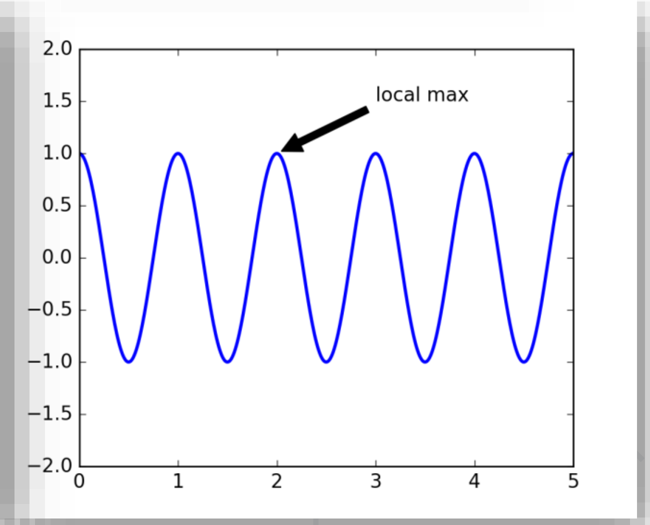
- To choose a plot type, think about
  - Numerical or categorical variables
  - Structure – spatial, temporal, etc.
  - Clustering
  - Relative size
- How can you know what all of these are?
  - Perform an **exploratory data analysis** first
  - "Play around" with the data
    - Check different measures (such as means, standard deviations, ranges, etc.), plot different charts
    - Explore the distributions and relations of variables
  - **Document** the **exploration process** and your **findings** to remember them later

Basic Rules

- Choose the appropriate chart type
  - If you can (and want), compare different types of charts
- Make your plot big enough to fit the plotting area
- When it's not obvious, add a title and a legend
- **Label the axes!**
- Optionally, point at interesting data
- Use marker size and color to convey information
- Don't strain the reader!







Plotting Basics

Starting out Easy

Plotting in matplotlib

- Quite easy to start plotting
- Very powerful
- There are many ways to do the same thing
- The documentation and examples are really good
  - We'll often end up consulting them, or the community
  - There are many options and it’s difficult to remember them all
- Importing the library

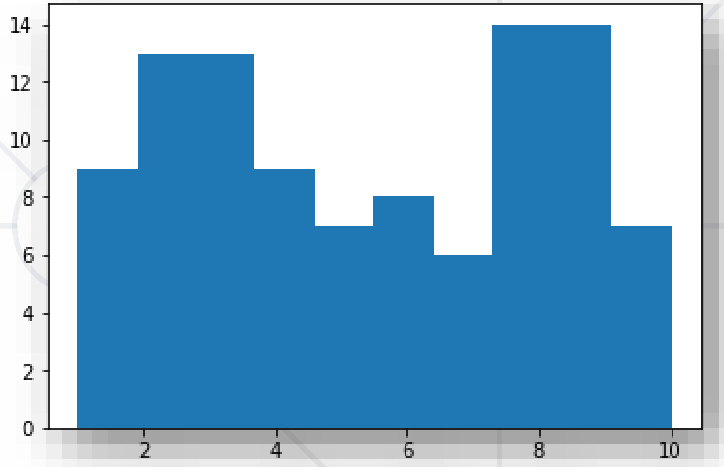
```
import matplotlib.pyplot as plt
```

- In Jupyter notebook, write the magic string %matplotlib inline in the first cell before importing
  - This will make plots appear as images in the notebook

Creating Simple Plots

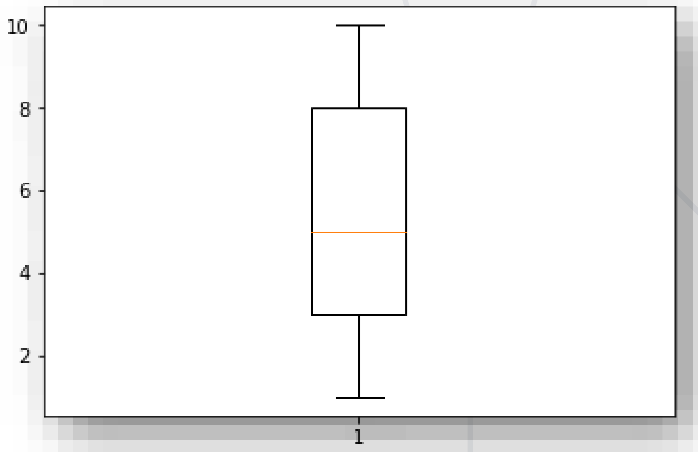
- Histogram
  - Shows the distribution of one variable

```
import numpy as np
values = np.random.randint(1, 11, 100)
plt.hist(values)
plt.show()
```



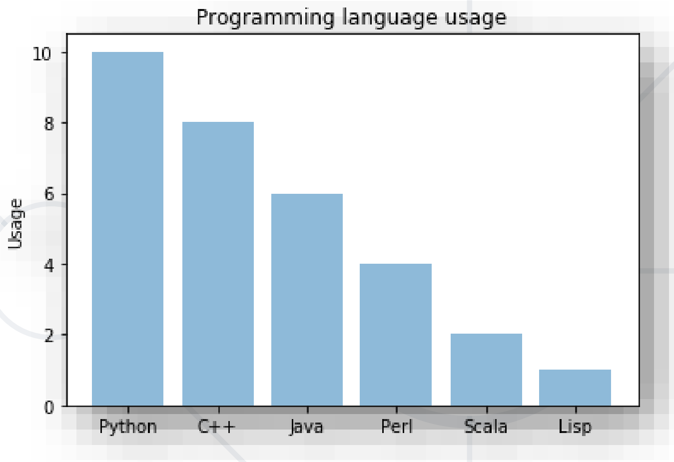
- Boxplot
  - Another way to show the distribution of one variable
  - May also be used to compare many distributions
  - *How to read* a boxplot

```
plt.boxplot(values)
plt.show()
```



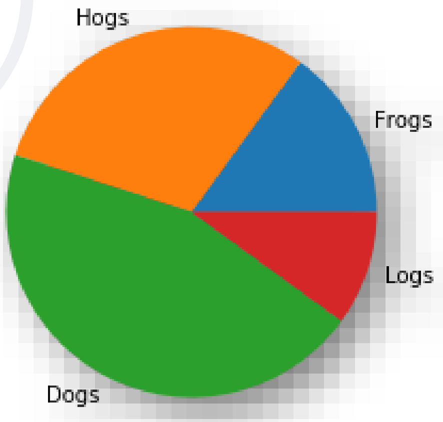
Creating Simple Plots

- Bar chart
  - Shows how one numeric value compares among different categories
  - Two variables: one categorical, one numerical
  - A little bit more difficult to plot
    - See a tutorial *here*



- **Although they look similar, histograms and bar charts are different!**
- Pie chart
  - Shows the relation of each part to the whole

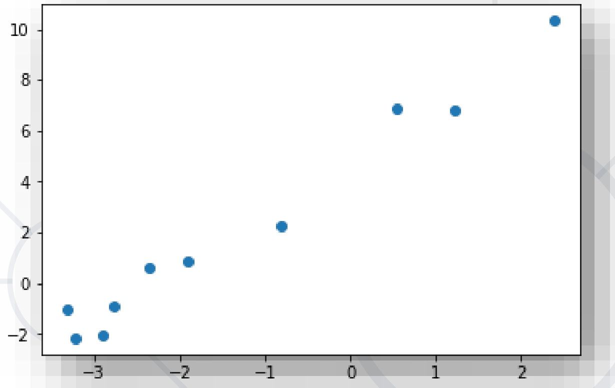
```
sizes = [15, 30, 45, 10]
plt.pie(sizes, labels = ["Frogs",
    "Hogs", "Dogs", "Logs"])
# Make the plot look circular
plt.gca().set_aspect("equal")
```



Creating Simple Plots

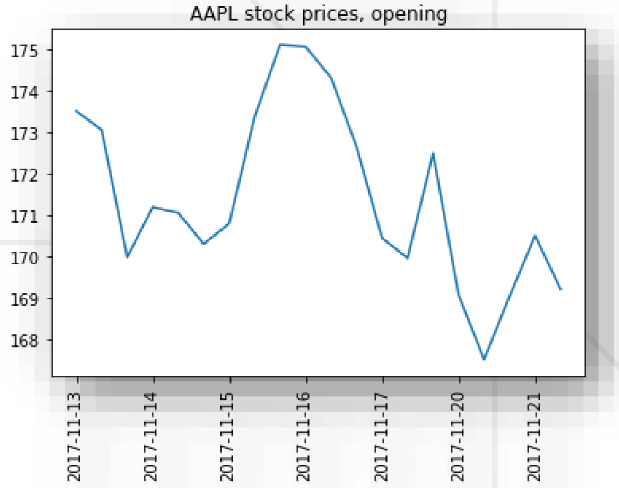
- Scatterplot (or scatter plot)
  - Shows how two variables compare
  - Can be used for displaying trends or correlations

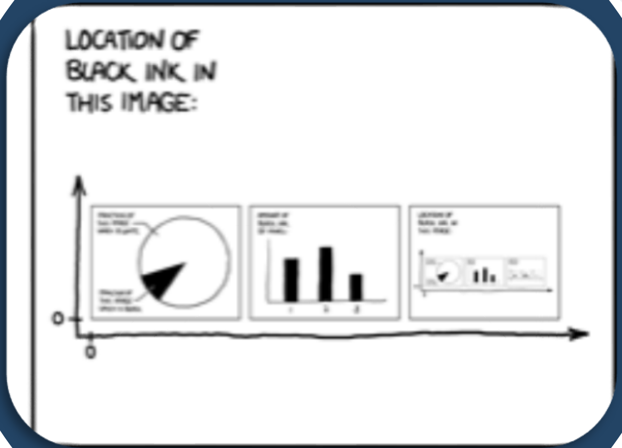
```
x = [-2.35, 1.22, -3.32, 2.39, -2.77,
        -3.21, 0.55, -0.81, -2.89, -1.9]
y = [0.58, 6.79, -1.01, 10.32, -0.9,
        -2.16, 6.87, 2.22, -2.05, 0.86]
plt.scatter(x, y)
plt.show()
```



- Line chart
  - Similar to scatterplot
  - Useful to show dependencies of two variables
    - If the horizontal axis is time – evolution

```
dates = ...
open_prices = ...
plt.plot(dates, open_prices)
plt.xticks(dates[::3], rotation = "vertical")
plt.title("AAPL stock prices, opening")
plt.show()
```

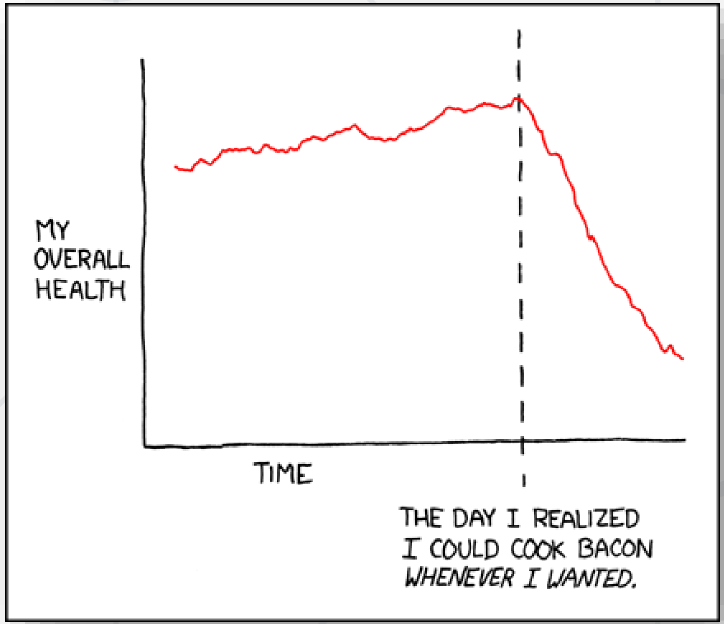


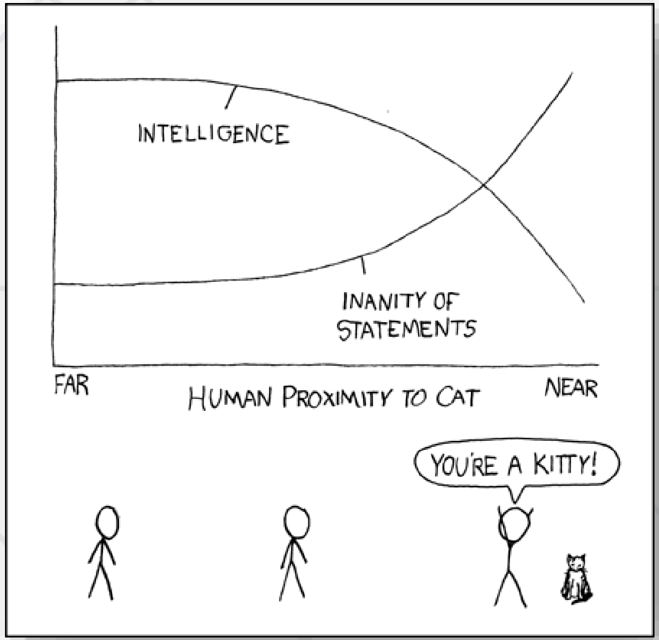


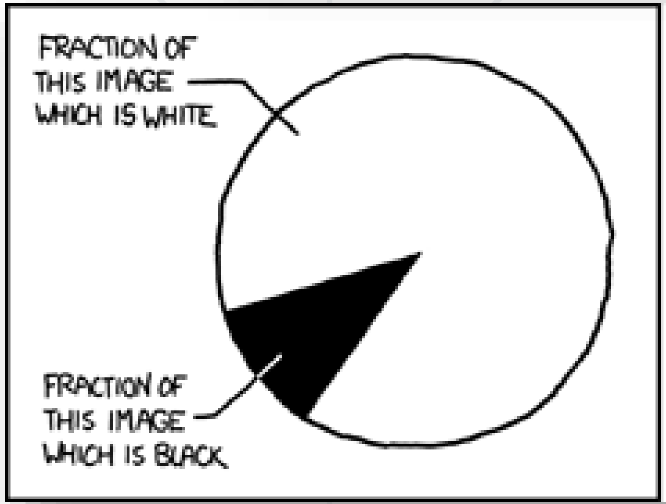
Data Visualization Examples

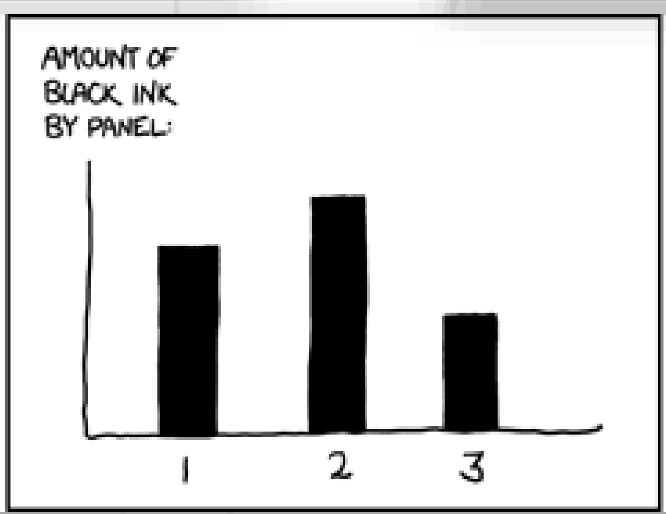
The Good, the Bad, the Ugly and the WTF

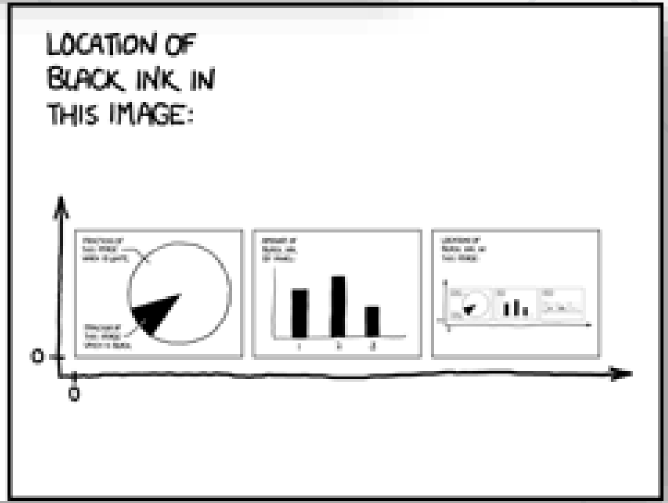
Some Examples





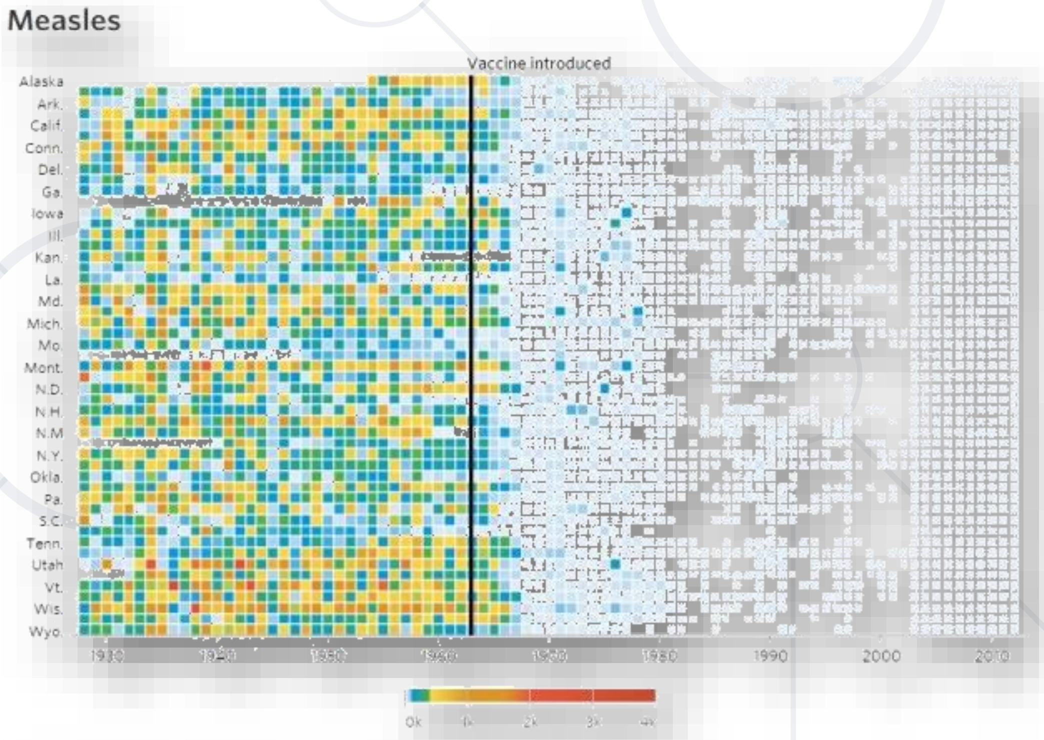






Infectious Diseases and Vaccines ✓

- Once again, a heatmap is used to convey disease spread
  - The legend has numbers
  - The *full chart* is interactive(provides all numbers)
- Makes use of temporal data
- Labels an interesting point
  - Allows us to compare "before" / "after"
- Conveys a clear message
  - Vaccines almost eradicated diseases



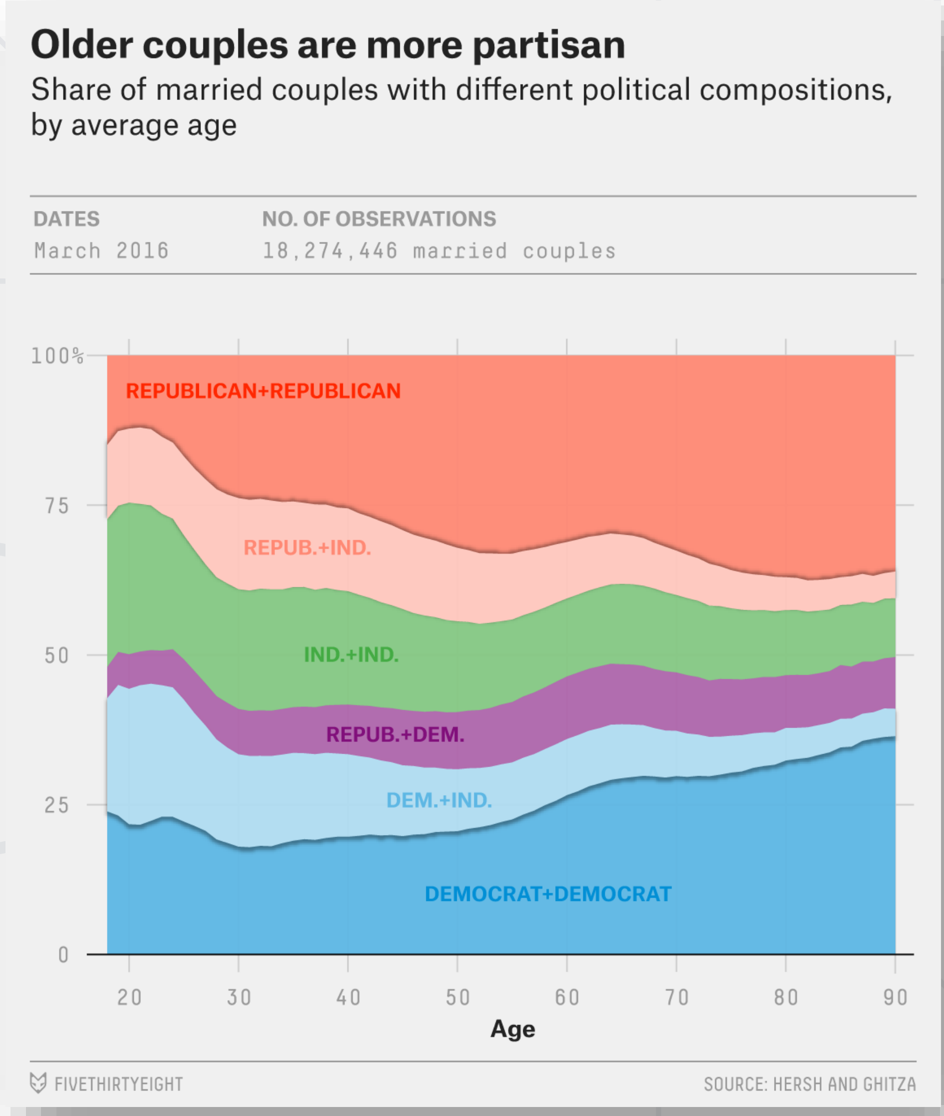
Note: CDC data from 2003-2012 comes from its Summary of Notifiable Diseases, which publishes yearly rather than weekly and counts confirmed cases as opposed to provisional ones.

Gay Marriage Acceptance ✓

- Good use of spatial structure (heatmaps)
- Overall message is clear
  - All states are in the "more" category
  - Some states are more accepting
- May additionally display numbers on the scale or on the map to give a quantitative view
  - This is an editorial, not a scientific plot so it's acceptable

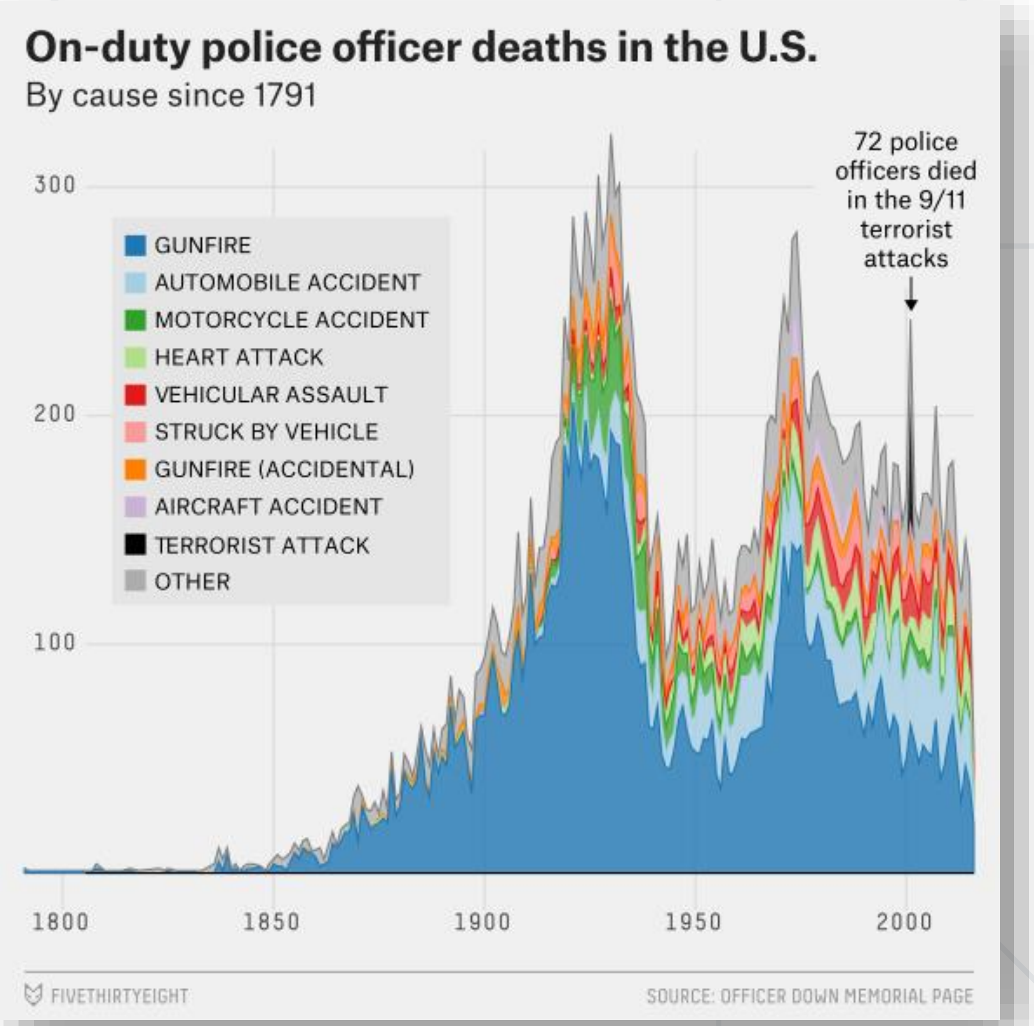
Political Views of Couples ✓

- Source: *FiveThirtyEight*
- Uses area plots to display relations
  - Allows comparison of different distributions for different ages
- Uses a clear color map
- Uses labels to make comparisons easier



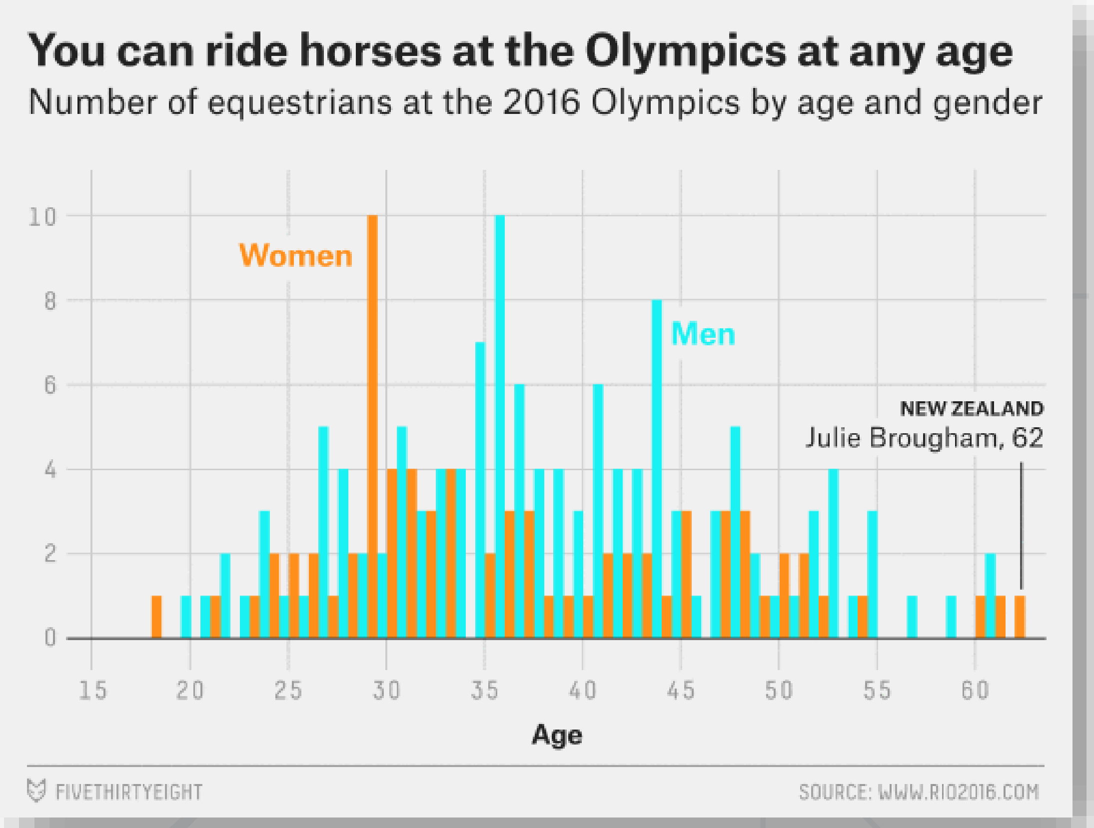
On-Duty Officer Deaths in the US ✓

- Source: *FiveThirtyEight*
- Uses a stacked area chart
  - Total area – **overall** tendency
  - Colored parts – tendency **for different causes**
  - Allows to inspect both
- Labels an interesting point



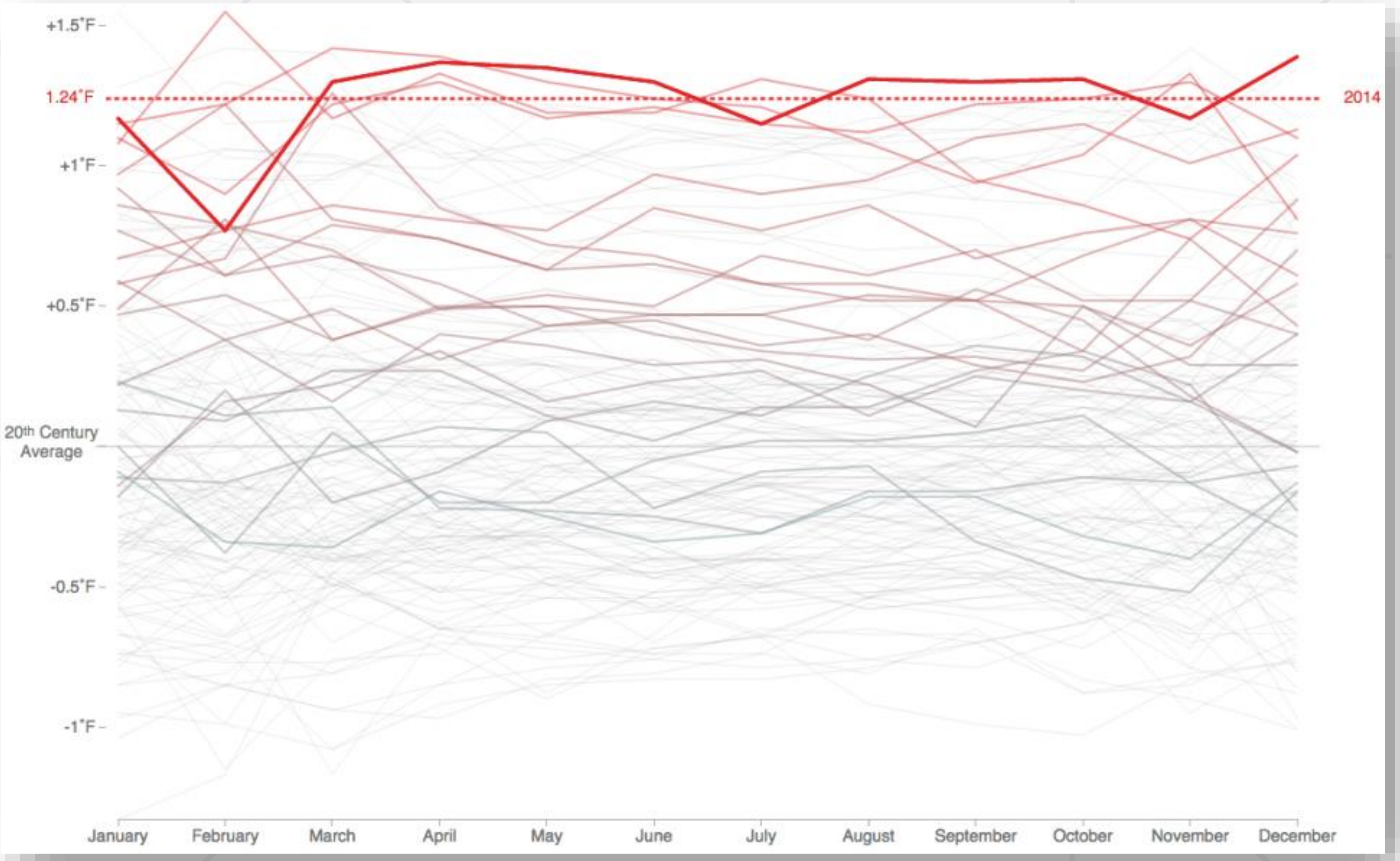
Horse Riders by Age and Gender ✓

- Source: *FiveThirtyEight*
- Histograms are relatively rare in non-scientific visuals
- Shows the two distributions clearly
- Conveys the message
  - The distributions are nearly identical
- Uses labels for outliers



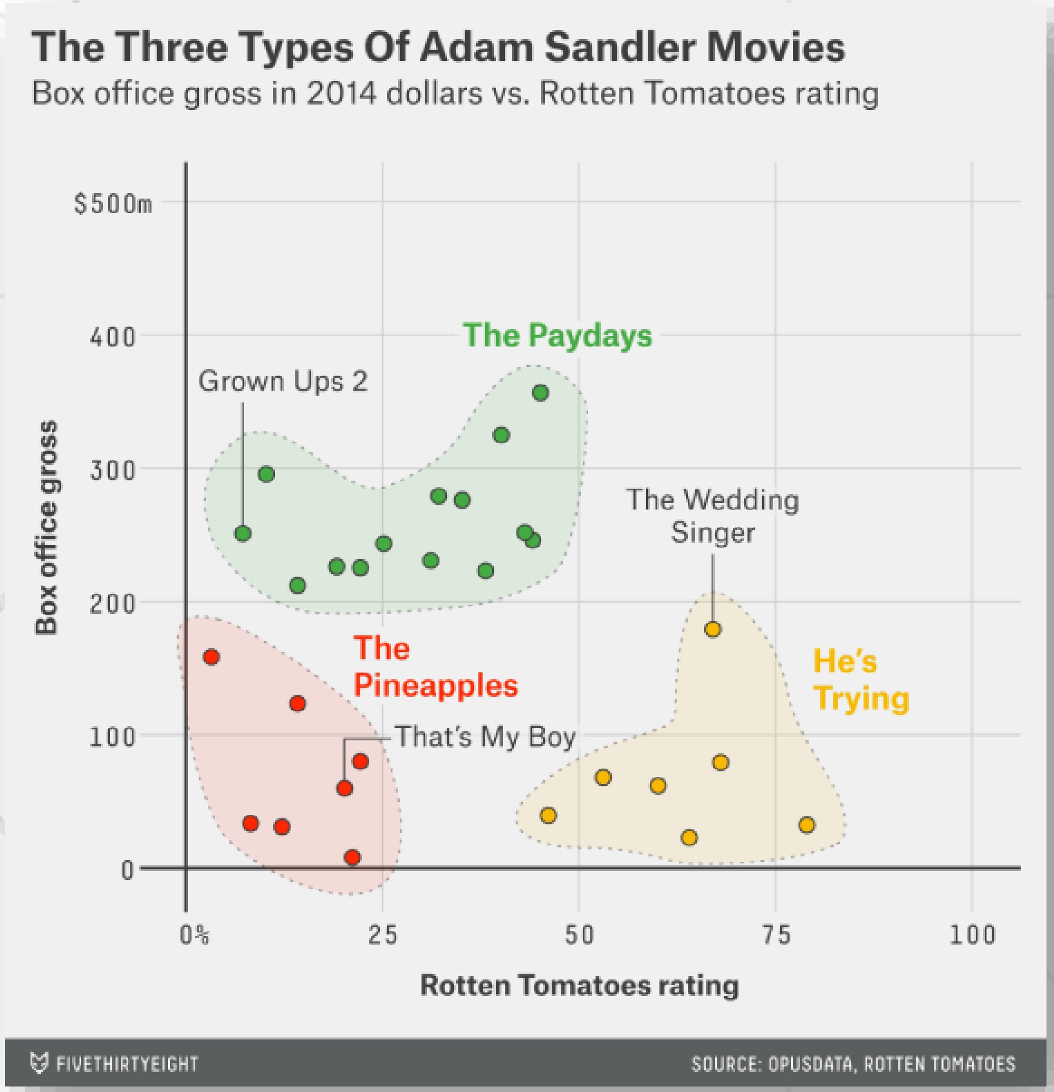
2014: The Hottest Year on Record ✓

- Source: *FlowingData*
- Presents temporal data in a classic way
- Uses color to show rising temperature
- Uses a thicker line to make it stand out in an otherwise very busy line plot



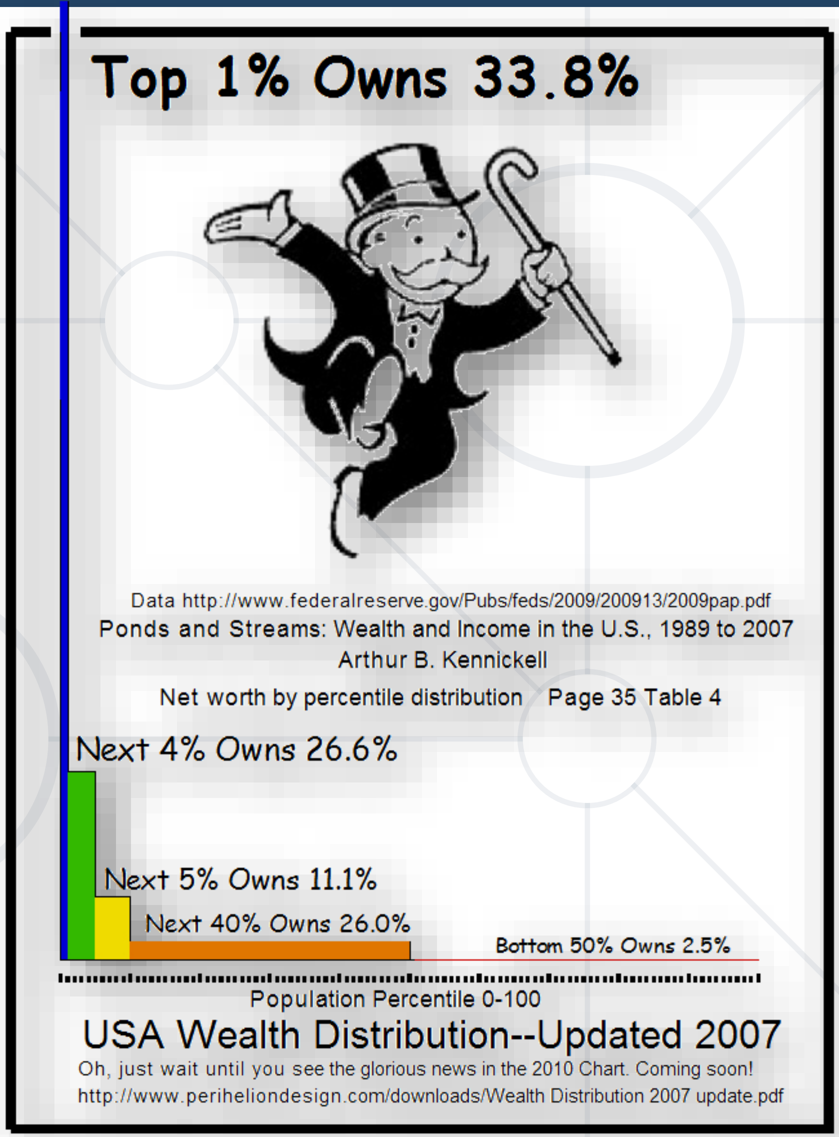
Types of Adam Sandler Movies ✓

- Source: *FiveThirtyEight*
- Presents a scatterplot of rating and profits
- Shows and labels clusters clearly
  - Uses different colors
- Conveys a clear message



USA Wealth Distribution ✗

- Source: WTFViz via Gizmodo
- Highly skewed and disproportional elements
  - It wants to convey the message of disproportionality but there are better ways (e. g. "cutting" the y-axis and displaying y-axis labels)
- Comic Sans(?!), useless image and useless text right in the middle of the chart



Too Much Pie ✗

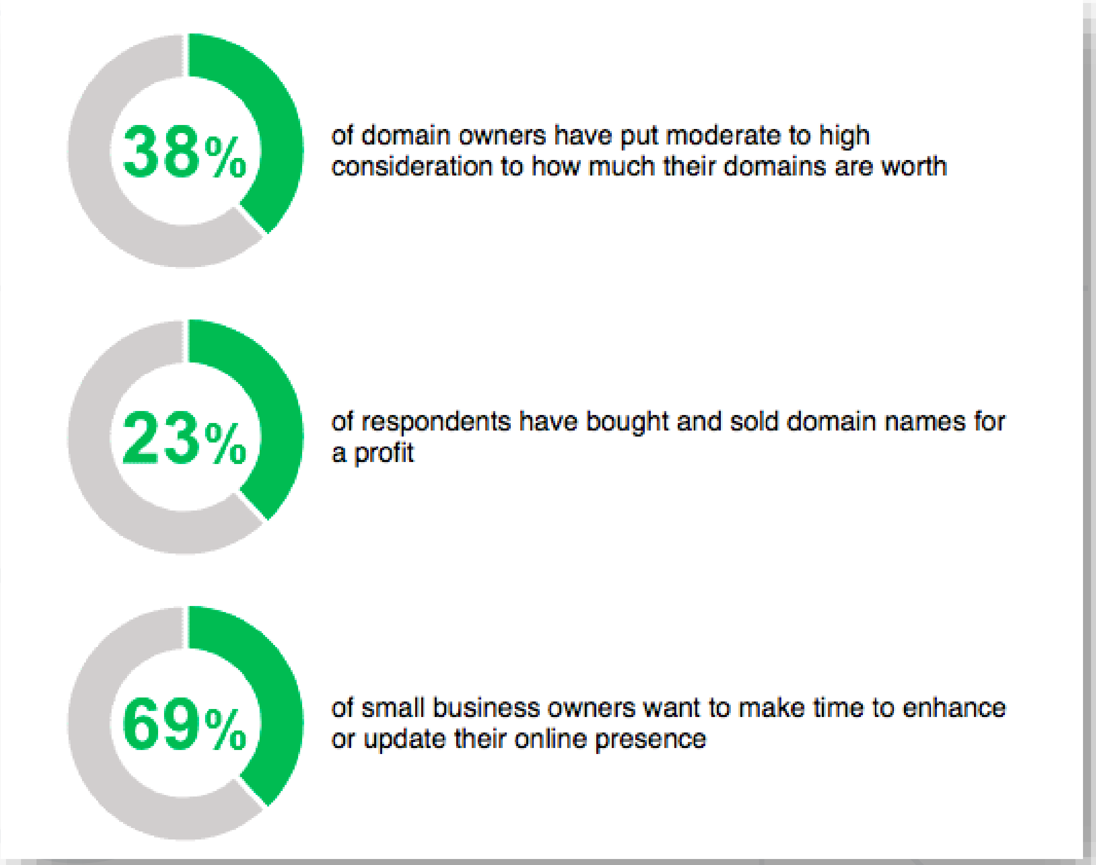
- Source: WTFViz via DesignRoast
- The parts of the pie add up to 188%
  - They're meant to be viewed on their own, e.g. there might be combinations of factors
- A pie chart is highly **not recommended** in this case
- Other than that, it shows good labels and a nice color scheme

Too Much Pie, Part 2 ✗

- Source: WTFViz (New Yorker)
- Once again, the pie chart makes no sense
- The values aren't related at all
- Why is there a world map?

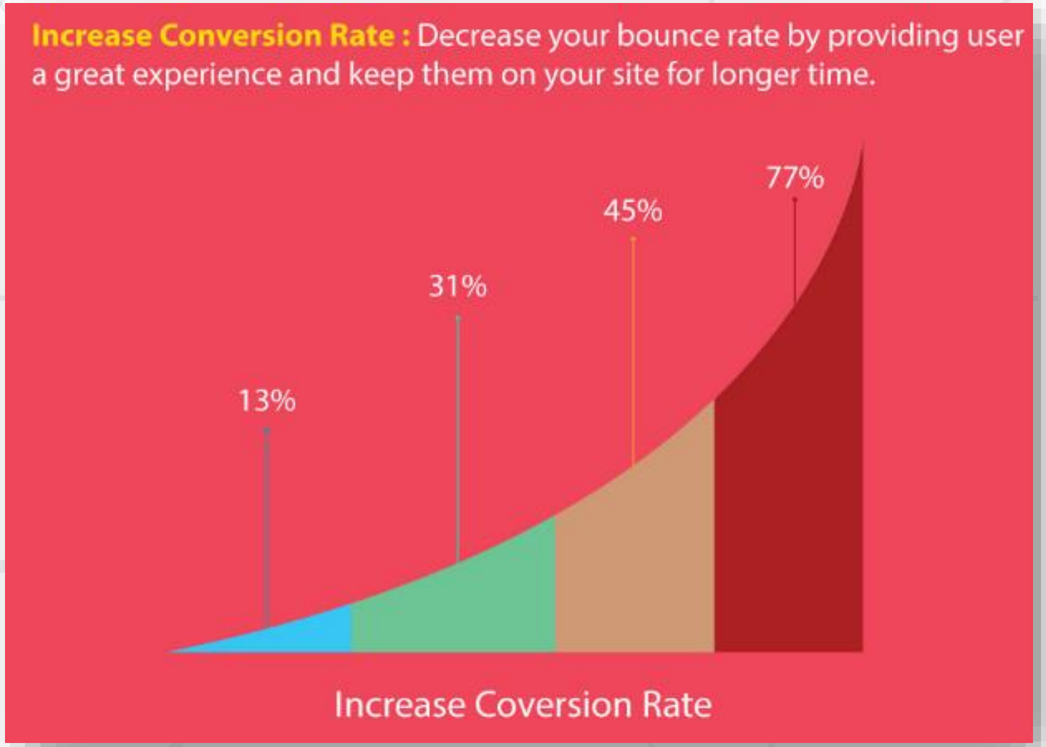
Too Much Pie, Part 3 ✗

- Source: Email from GoDaddy, WTFViz
- The numbers aren't related to the color of the rings at all
- Maybe just a programming mistake
  - Still, be careful and check your work



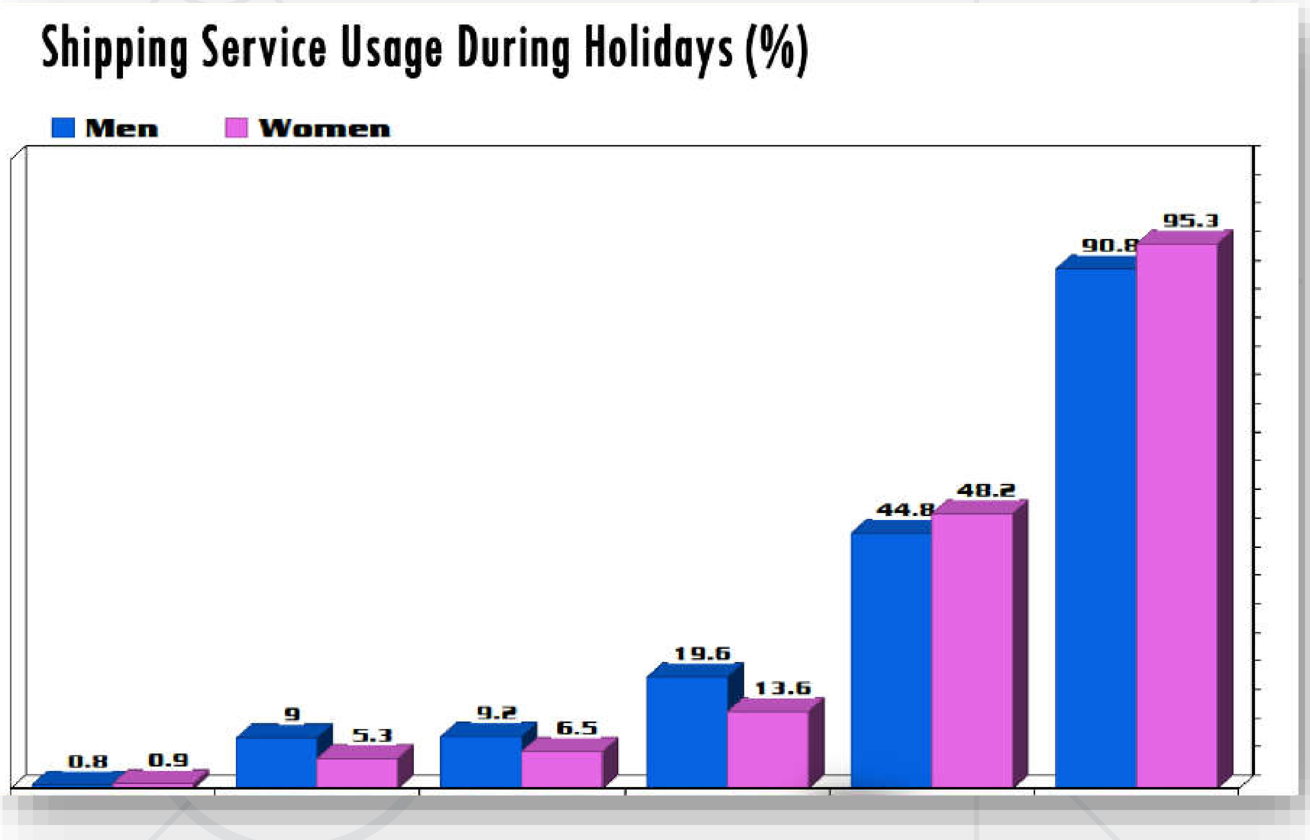
No Axes ✗

- Source: WTFViz via DesignRoast
- I don't know (and can't see) the real purpose
- No axis labels and no numbers (the bottom label is not the x-axis)
- Distracting background
  - Do you remember "Don't strain the reader"?



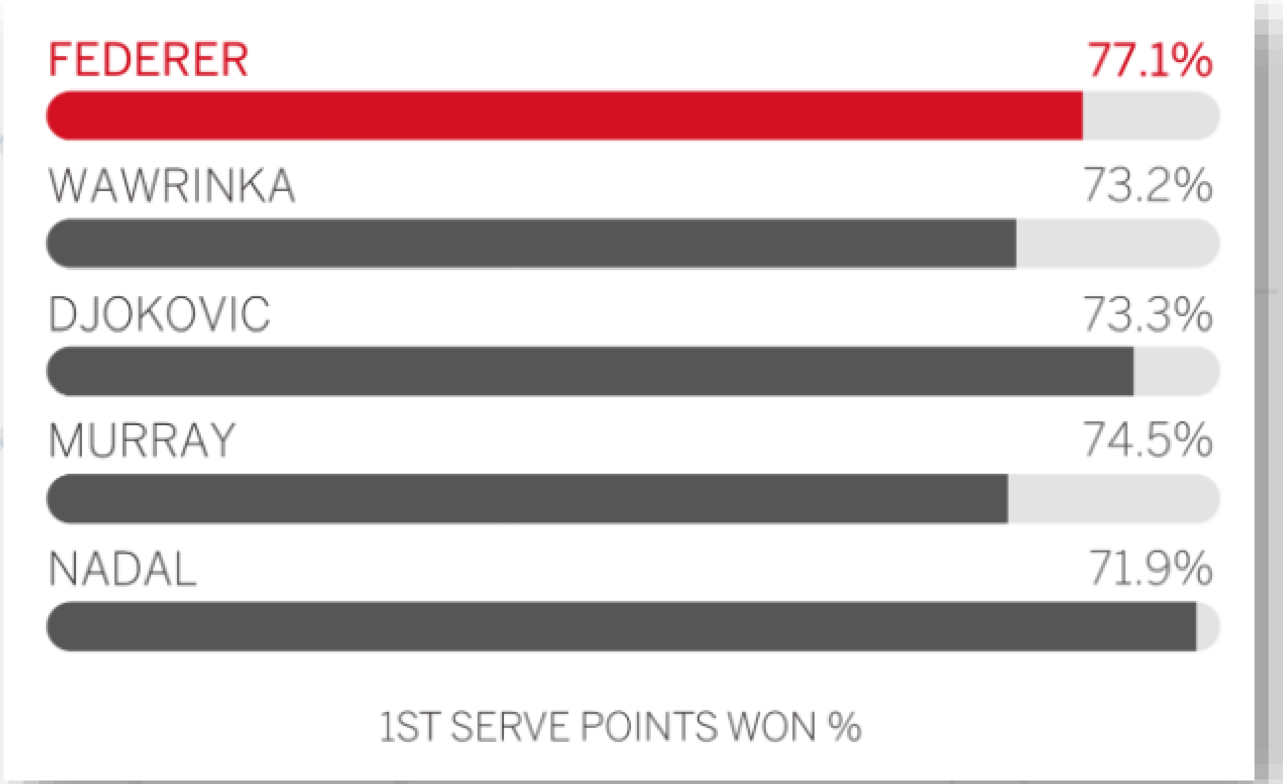
No Axes, Part 2 ✗

- Source: WTFViz via DesignRoast
- The categories are gone
  - The image is not trimmed, this is the entire chart
  - Are those different days, different products or something else?
  - Also, 3D doesn't give additional information
- Also, the design is kind of lame



Wrong Scales X

- Source: WTFViz via DesignRoast
- How come 71,9% is
  - Further than 77,1%
  - Close to full (looks like 95-98%)

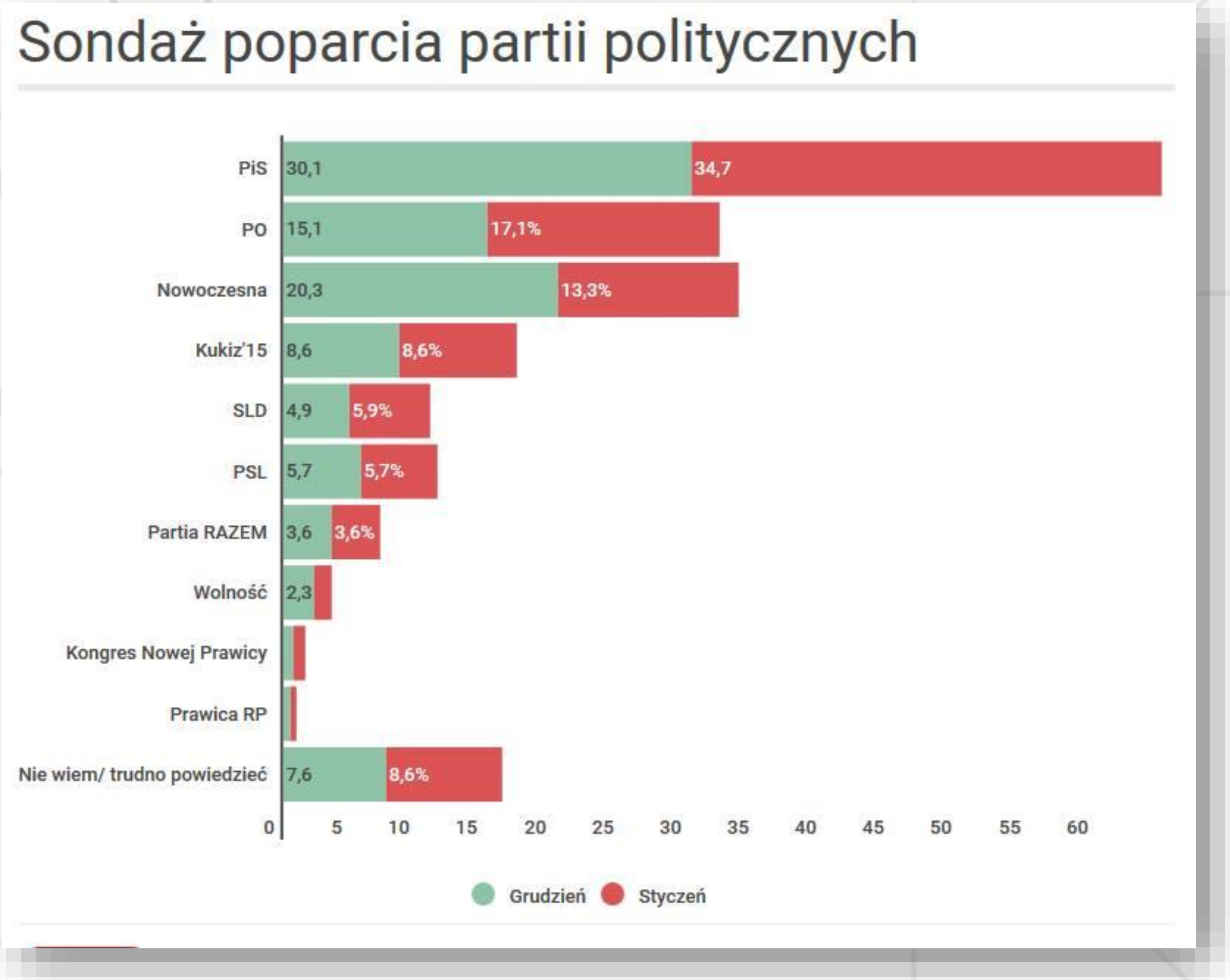


Double Scales that Make No Sense ✗

- Source: WTFViz
- The y-axis scales and the numbers on the bars represent different things
  - They're also on different scales (blue 11 is less than gray 10 but blue 4 seems very close to gray 18)
- Impossible to read and understand without additional explanation

Wrong Data ✗

- Source: WTFViz
- The chart looks OK
- Political party affirmation for December (blue) and January (red)
  - Why would one sum percentages like these? Makes no sense

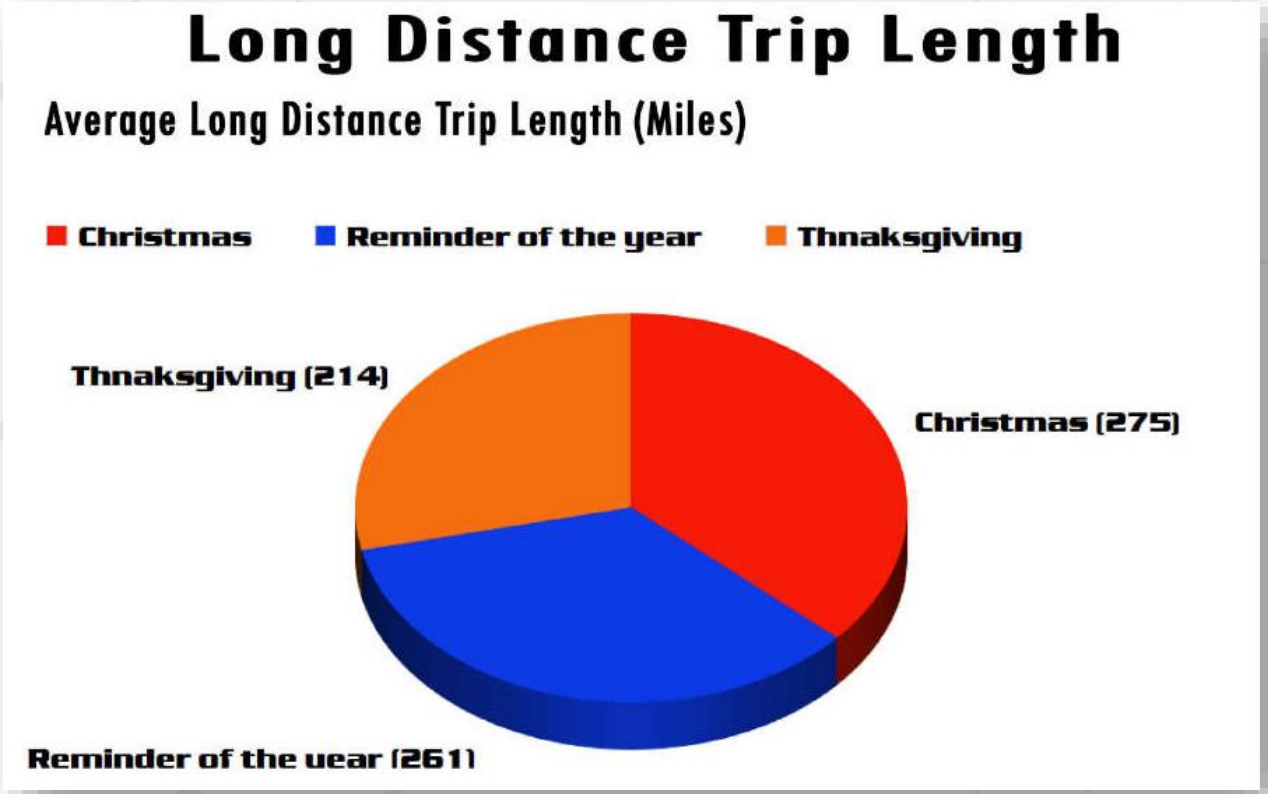


Wrong Data ✗

- Source: WTFViz
- Those burgers make data extremely difficult to compare
  - The numbers don't help very much

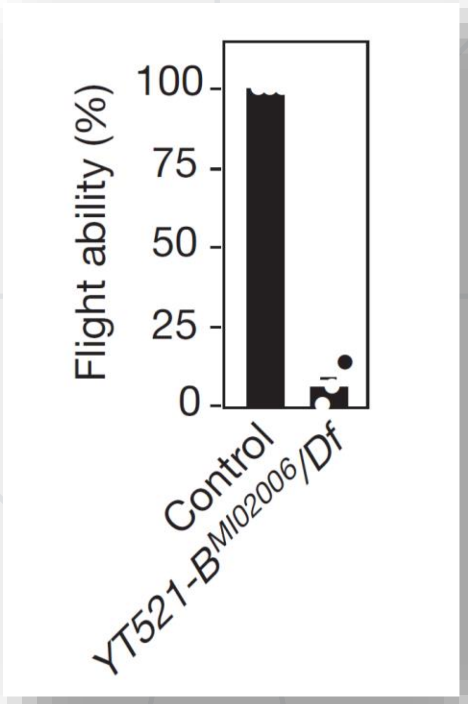
Wrong Information and Mistakes ✗

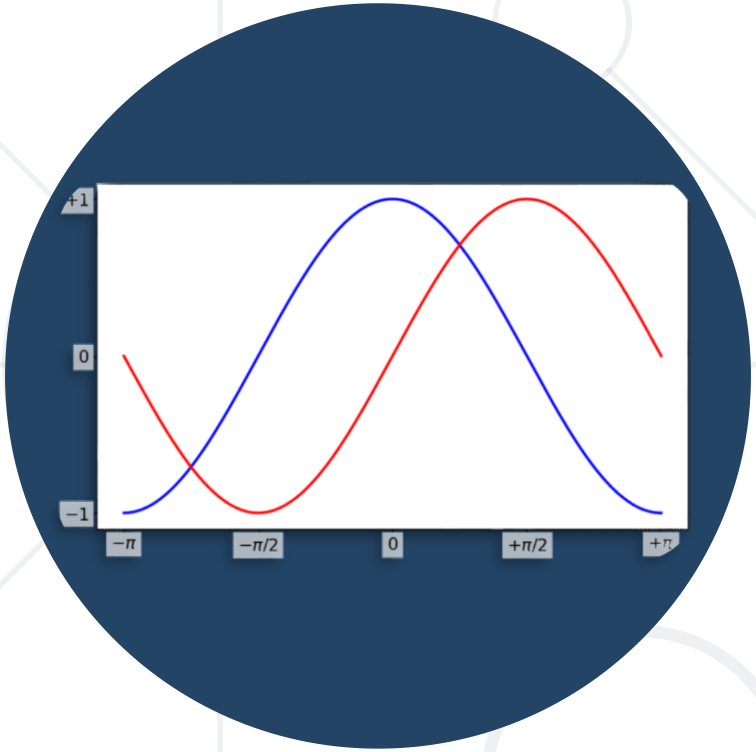
- Source: WTFViz
- Spelling errors
  - Also, unreadable font
- The pie chart conveys no information at all
  - Better – use a bar chart



Bar Chart Mistakes ✗

- Source: WTFViz
  - Original: Nature :(
- Bubbles make the values extremely difficult to compare
  - Where does the right bar end?
  - Where does even the left bar end?
- Below: Why are the bars warped?





Customizing Plots

Making Things Beautiful

Applying Styles

- Matplotlib has many default styles

```python
print(plt.style.available)
```

- Using a different style

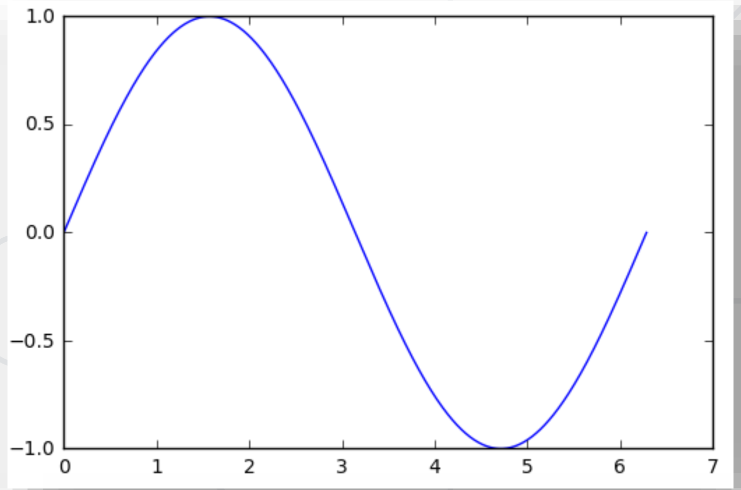
```python
plt.style.use("ggplot")
```

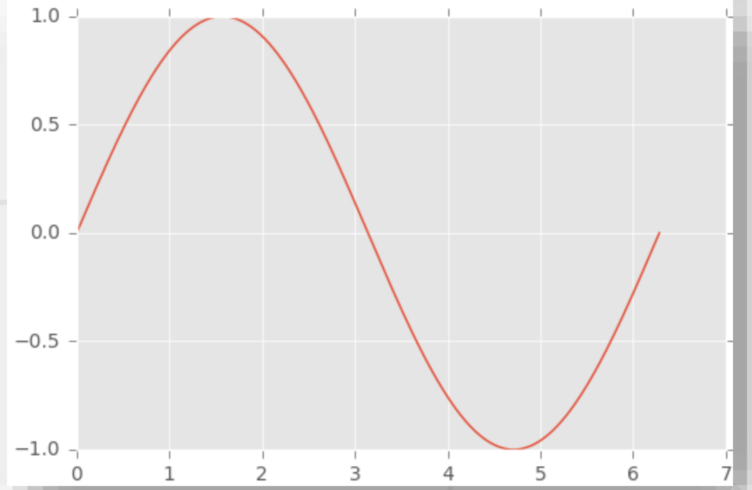
- Reverting to the default style

```python
plt.style.use("default")
```

  - In Jupyter notebook, `%matplotlib inline` uses its own styles

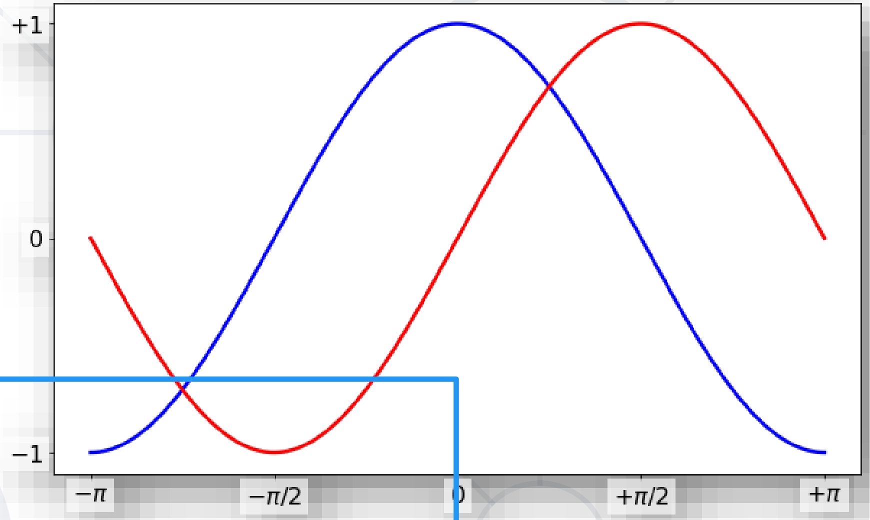
- Example: Draw a simple scatterplot, histogram and / or line chart in all different styles





Customizing Plots

- Every call to `plt.hist()`, `plt.plot()`, `plt.boxplot()`, etc., accepts many arguments
  - Colors, markers (type, size)
  - Axis limits and locations, axis labels
  - Data labels and additional text
  - Legend location and appearance




```python
cos_x = ... # [-pi; pi]
sin_x = ...
plt.figure(figsize = (10, 6))
plt.plot(x, cos_x, color = "blue", linewidth = 2.5, linestyle = "-")
plt.plot(x, sin_x, color = "red", linewidth = 2.5, linestyle = "-")
# Tick marks and labels
plt.xticks([-np.pi, -np.pi / 2, 0, np.pi / 2, np.pi],
    [r"$-\pi$", r"$-\pi/2$", r"$0$", r"$+\pi/2$", r"$+\pi$"])
plt.yticks([-1, 0, 1], [r"$-1$", r"$0$", r"$+1$"])
for label in ax.get_xticklabels() + ax.get_yticklabels():
    label.set_fontsize(16)
    label.set_bbox({facecolor: "white", edgecolor: "None", alpha: 0.65})
```

Example: Create a Customized Plot

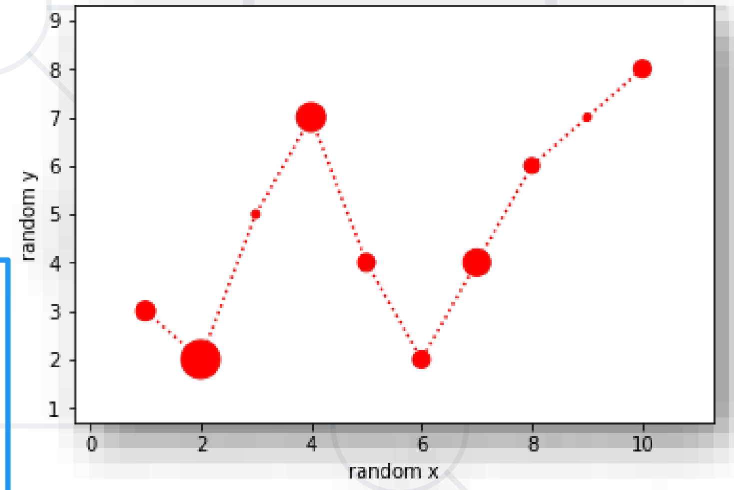
- Create a plot similar to the picture using the given data
  - This is to show that marker colors, sizes and types can be given as arrays – the elements are applied sequentially

```
x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
y = [3, 2, 5, 7, 4, 2, 4, 6, 7, 8]
y_radius = [10, 20, 4, 15, 9, 9, 14, 8, 4, 9]
```

```
# Note that s (for size) represents the area, not radius
plt.scatter(x, y, s = np.array(y_radius) ** 2, color = "red")
plt.plot(x, y, linestyle = "dotted", color = "red")

plt.xlim(np.min(x) - 1.3, np.max(x) + 1.3)
plt.ylim(np.min(y) - 1.3, np.max(y) + 1.3)
plt.xlabel("random x")
plt.ylabel("random y")

plt.show()
```



* Lab: Playing with matplotlib

- A very good part of matplotlib are the examples
  - See them *here*
- Many examples of common use cases
  - Creating multiple plots
  - Different types of plots: violin plot, residual plot, heatmap, etc.
  - Usages of color, shaded area, markers, labels, etc.
- Play with some of these examples to get a feel of what you can do with matplotlib
- Customize some of the examples
  - Read the docs to see all parameters

Exploratory Data Analysis (EDA)

Making Sense of our Data

Exploratory Data Analysis

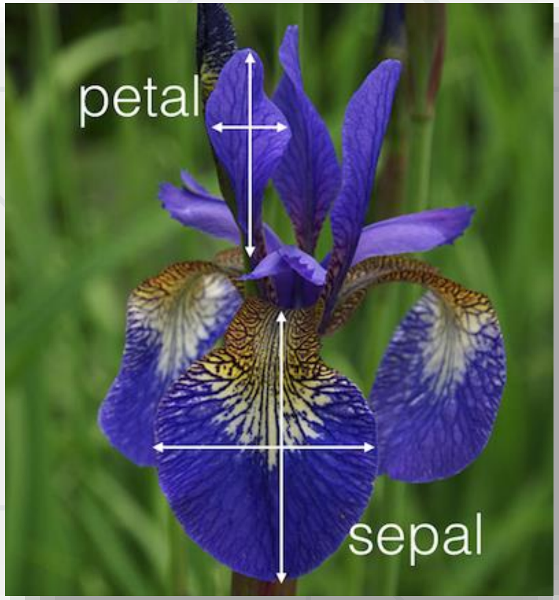
- A process to see what the data can tell us
  - Not tied to formal data modelling or hypothesis testing
- Many people have written about this
  - Most notably, John Tukey (1961)
- Like data cleaning, relies heavily on the scientist's intuition
- Objectives
  - Suggest hypotheses
  - Assess assumptions on which models will be based
  - Aid selection of features (feature engineering)
  - Provide a basis for further data collections

Analytic Graphs

- Good for explaining the dataset visually
  - Show distributions, relations, comparisons and causality even in multivariate data
- Principles of analytic (scientific) graphs
  - Show **comparisons**
  - Show **causality**
    - **Correlation does not imply causation**
  - Show **multivariate data** (many variables)
  - Integrate evidence from **multiple sources**
  - Describe and **document** evidence
  - **"Content is king"**
    - We have to have something interesting to report
- These principles apply to EDA as well

Exercise: Exploration of the Iris Dataset

- One of the most famous datasets in data science
  - Sizes of petals and sepals (see picture) for three classes of iris flowers
- Read, inspect and clean the *dataset*
  - Using the data cleaning approaches you already know
  - **Can we predict classes from sizes?**
- Inspect the distributions
  - Plot histograms and boxplots, print stats
    - Try different plot settings
  - Compare the quantities – scatterplots
    - In some cases, a "brute force" method might be useful – compare everything against everything else
  - Plot a correlation matrix



Lab: Exploration of the Iris Dataset

- Usually, we first perform univariate analysis, then go on to find correlations
  - Plot the entire distribution first
  - Start to break down by factors (in this case – the iris types)
  - Create additional columns if needed (data transforms)
  - Apply grouping, averaging and summing over groups to get an idea of possible "clusters" in the data
  - Inspect and plot certain data ranges (using filtering)
  - **Have fun with the data but don't forget the original question!**
- Next steps
  - After exploratory data analysis, we're usually able to form a hypothesis (a pair of hypotheses), model the data and check against the hypothesis
  - In other cases we can produce different visuals, graphics and dashboards to be used by others

* Lab: Exploration of the Iris Dataset

- We can also plot beautiful graphics using other packages (not `matplotlib`)
- An example of one such package is `seaborn`
  - Contains utility functions for some commonly used plots
  - Based on `matplotlib`
  - Read the docs *here*
    - It shows how to plot different distributions on their own and together, and also includes a little tutorial on an algorithm called KDE (kernel density estimation)
  - It also has other *tutorials* (such as *plotting linear correlations*)
  - It produces good-looking graphics but can lack customizability in some cases
- Other examples: *bokeh*, *plot.ly*, *ggplot*
  - And many more

Summary

- Main Concepts and Rules
- Creating Simple Plots
- Real-Life Examples: Good and Bad
- Customizing Plots
- Exploratory Data Analysis
  - Basic Guidelines
  - EDA as Part of the Data Science Process

Software University

Questions?

SoftUni

Software University

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

СофтУни БУДИТЕЛ

Finance Academy

SoftUni Diamond Partners

Software University

SmartIT

Coca-Cola HBC Bulgaria

SUPER HOSTING .BG

createX

Postbank

Решения за твоето утре

DRAFT KINGS

VIVACOM

AMBITIONED

INDEAVR

Serving the high achievers

SOFTWARE GROUP

PHAR VISION

Trainings @ Software University (SoftUni)

Software University

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity

Software
University

License