LESSON 5

EDA & Visualization: Part 1

Incorporating visualizations to aid our EDA

CSCI 3022 @ CU Boulder

Maribeth Oscamou

Content credit: Acknowledgments

Course Logistics: Your Second Week At A Glance

Mon 1/20	Tues 1/21	Wed 1/22	Thurs 1/23	Fri 1/24
NO SCHOOL - LABOR DAY		Attend & Participate in Class	HW 2 Due 11:59pm via Gradescope	In Class Quiz (beginning of class): Scope: Syllabus and Prerequisites Covered in HW 1 Attend & Participate in Class
			HW 1 feedback/ grades posted	HW 3 released (8am)

CC BY NC SA

Quiz 1 Details

- Pencil and paper quiz will take place the 15 minutes of class on Friday 1/24

- Scope: Syllabus AND Calculus and Discrete Structures prerequisite topics reviewed in HW 1 (there will NOT be any coding on Quiz 1)
- For a refresher on Prerequisite topics see the Prerequisite Review in Modules in Canvas
- You are allowed a calculator
- You are allowed a 2-sided 8.5” x11” crib sheet

CC BY NC SA

HW: Communicating Your Results

Before submitting the PDF of your HW check:

- Is all LaTeX correctly rendered (i.e. are math equations showing up correctly)?
- Are all lines of code and LaTeX showing on the page (not cutoff)?
    - Troubleshooting cutoff LaTeX : Break up long math statements into smaller ones.
    - Troubleshooting cutoff code - breakup code into multiple lines
        - use the \ symbol to indicate to Python that your code continues on to the next line

```
ee.groupby("Year") \
    .agg({"Popular vote":"mean", "%":"sum"}) \
    .rename(columns={"Popular vote":"avg Pop vote", "%":"sum%"})
```

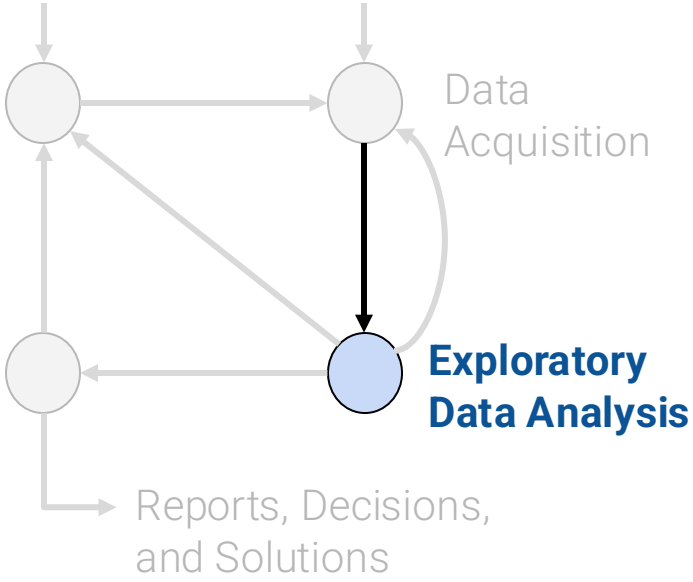
- Are all graphs in your answers showing up in the PDF?
    - Troubleshooting: See the Jupyter Notebook and LaTeX troubleshooting tips (Canvas Modules)

If you are noticing an issue with your plots not printing to the PDF here is the fix: **Disable autosave.** Autosave is enabled by default in JupyterLab.

- 1.) In the toolbar, go to `Settings`
- 2.) Disable `Autosave Documents`

Points will be deducted for HW that is submitted without completing these checks.

Plan for first 2 weeks



(Weeks 1 and 2)

EDA, Wrangling, and Data Visualization

Today's Roadmap

Finish Lesson 3: EDA & Wrangling

Start Lesson 5: EDA & Visualization Part 1

Wait What About Lesson 4?

Lesson 4: Pandas Bootcamp Part 2

- Covered in HW 2 (Question 1) as a video assignment

Lesson 5 Learning Objectives:

- **Use Python functions to visualize distributions of Qualitative Variables**

EDA & Visualization: Part 1

EDA & Visualization Part 1:

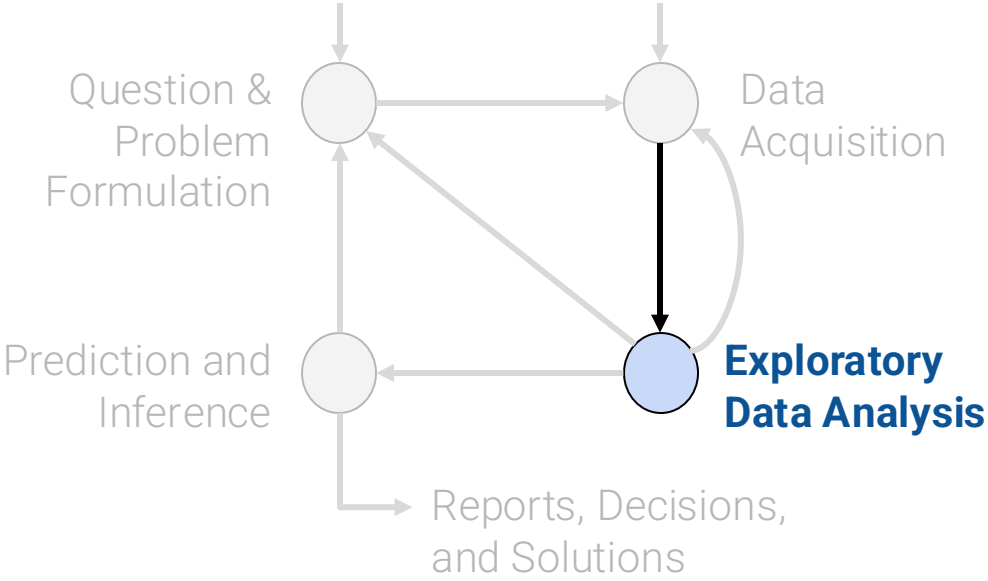
- Intro to Visualizing:
- Visualizing Distributions
  - Qualitative Variables

Intro to Visualizing Data

- Visualizing Data
  - Tools Used in this Course

CC BY NC SA

Data LifeCycle



EDA, Wrangling, and Data Visualization

Goals of Data Visualization

Goal 1: To **help your own understanding** of your data/results.

- Key part of exploratory data analysis.
- Summarize trends visually before in-depth analysis.
- Lightweight, iterative and flexible.

Goal 2: To **communicate results/conclusions to others**.

- Highly editorial and selective.
- Be thoughtful and careful!
- Fine-tuned to achieve a communications goal.
- Considerations: clarity, accessibility, and necessary context.

What do these goals imply?

Visualizations aren't a matter of making "pretty" pictures.

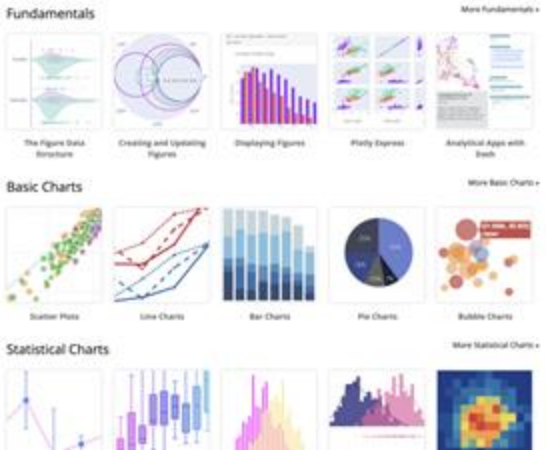
We need to do a lot of thinking about what stylistic choices communicate ideas most effectively.

CC BY NC SA

In this class we will focus on these 3 plotting libraries:

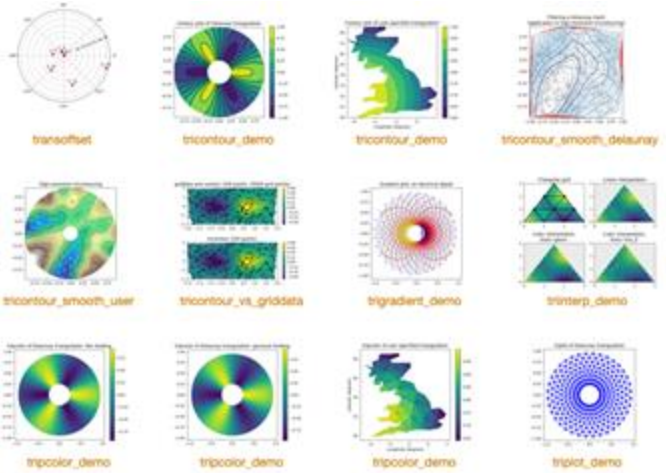
plotly

https://plotly.com/python/



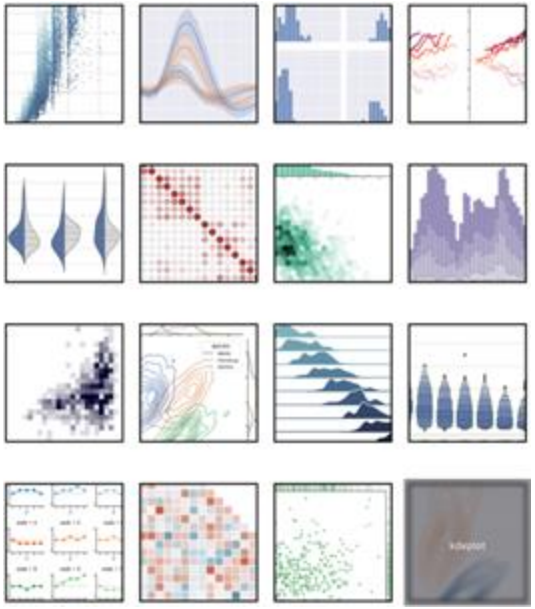
matplotlib

https://matplotlib.org/gallery.html



seaborn

https://seaborn.pydata.org/examples/index.html



The rich Python plotting ecosystem - this is not all!

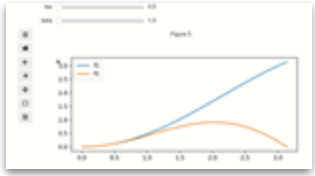
bokeh

Altair

seaborn

cartopy

mpl_interactions



plotly

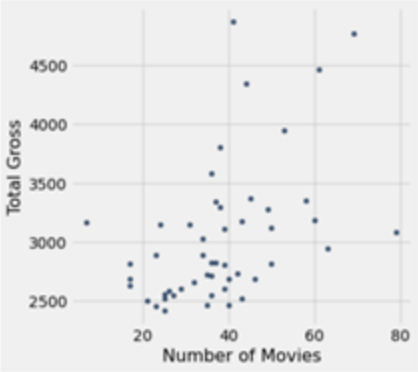
bqplot - Jupyter widgets

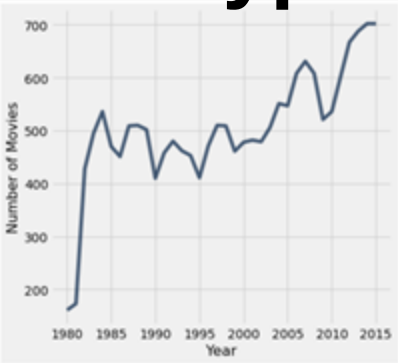
matplotlib

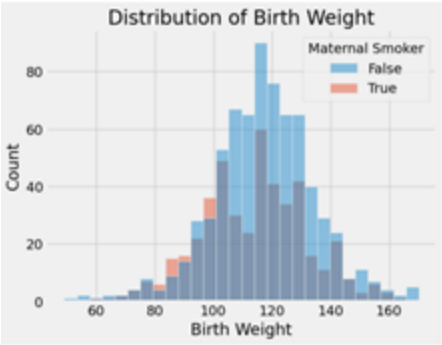
Visualizations

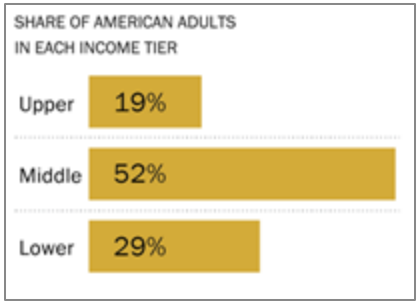
How you visualize your data depends on the

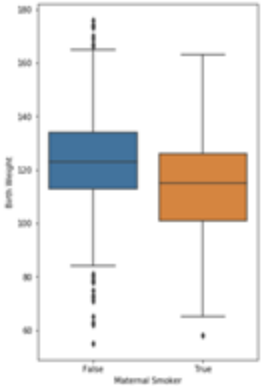
variable type of the data

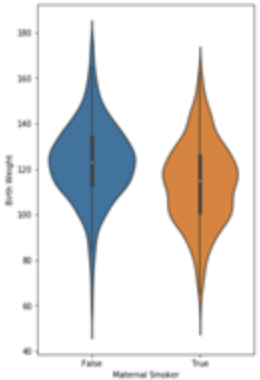


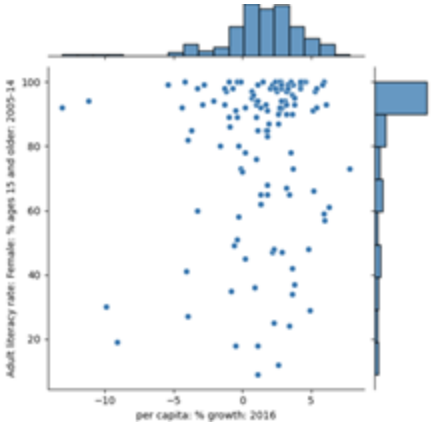












Learning Objectives

- **Use Python functions to visualize distributions of Qualitative Variables**

Visualizing Distributions

- Visualizing Distributions of Qualitative Variables:
  - Bar Charts
  - Python Code

CC BY NC SA

What Is a Distribution?

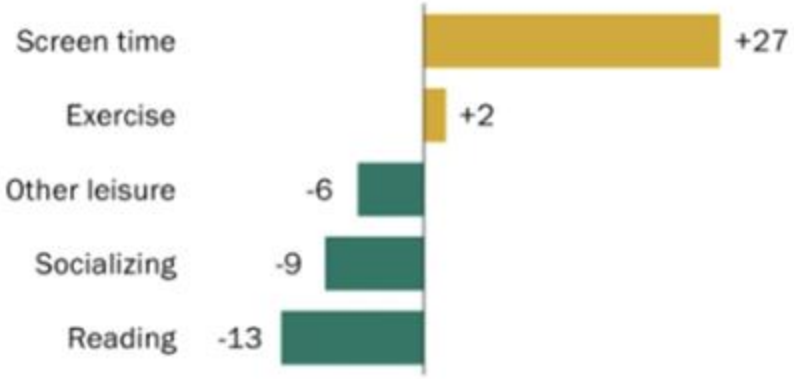
A **distribution** describes the frequency at which values of a variable occur.

- All values must be accounted for **once, and only once**.
- The total frequencies must **add up to 100%**, or to the number of values that we're observing.

CC BY NC SA

For older Americans, leisure time looks different today than it did a decade ago

Change in daily time use 2005-2015 (minutes), for people 60 and older

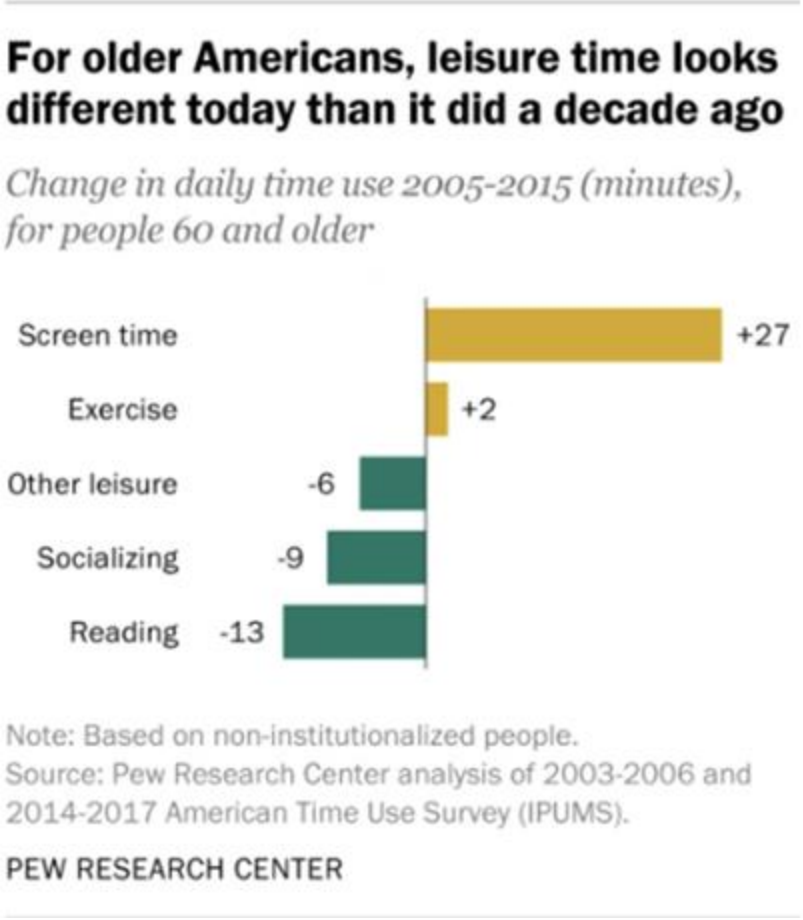


Note: Based on non-institutionalized people.
Source: Pew Research Center analysis of 2003-2006 and 2014-2017 American Time Use Survey (IPUMS).

PEW RESEARCH CENTER

Does this chart show a distribution?

CC BY NC SA



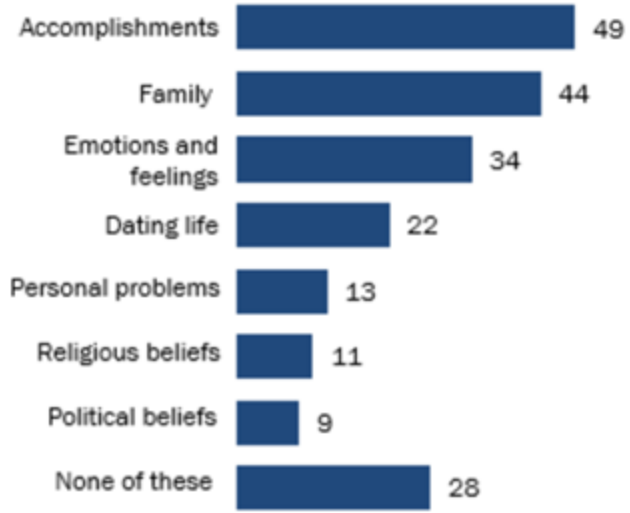
Does this chart show a distribution?

No.

- Individuals can be in more than one category.
- The numbers (and bar lengths) correspond to “time”, not the proportion or number of individuals in the category.

While about half of teens post their accomplishments on social media, few discuss their religious or political beliefs

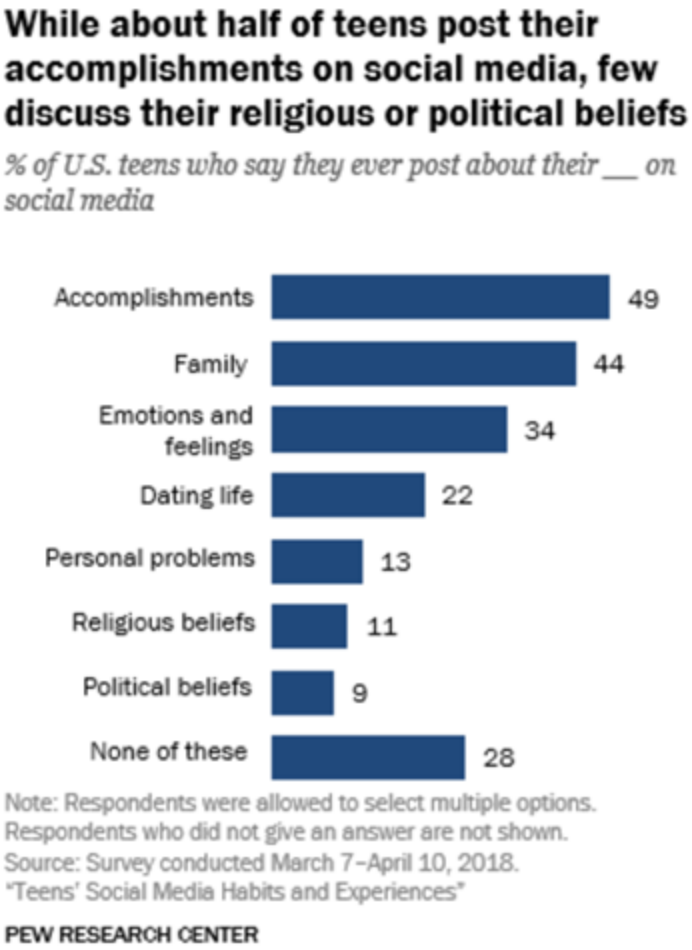
% of U.S. teens who say they ever post about their ___ on social media



Note: Respondents were allowed to select multiple options.
Respondents who did not give an answer are not shown.
Source: Survey conducted March 7–April 10, 2018.
"Teens' Social Media Habits and Experiences"

PEW RESEARCH CENTER

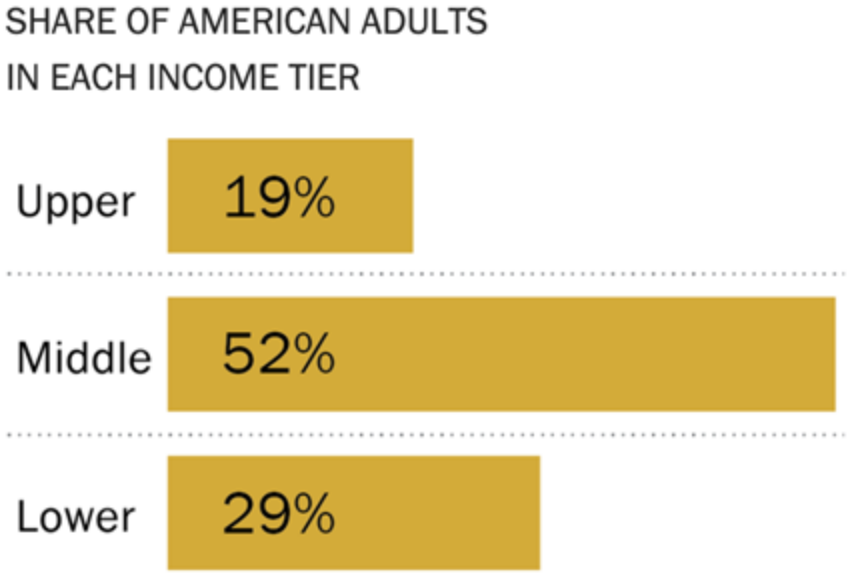
Does this chart show a distribution?



Does this chart show a distribution?

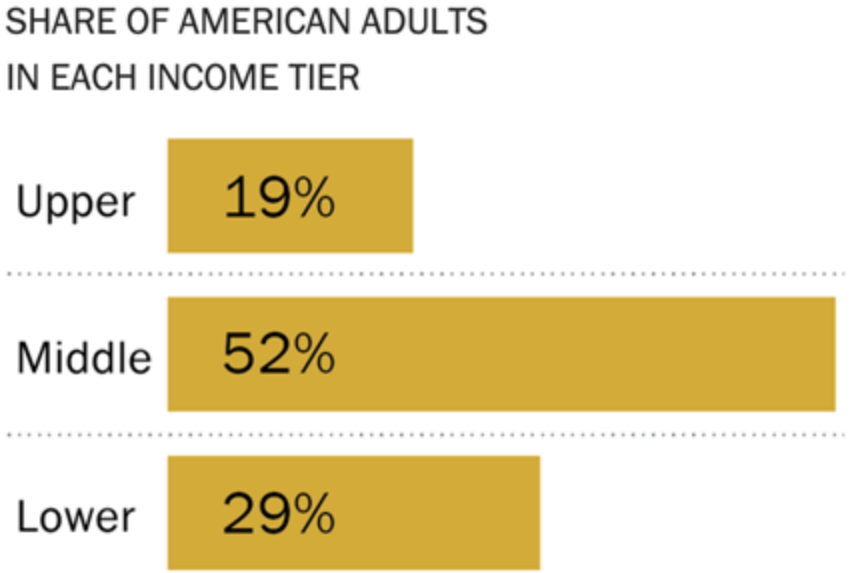
No.

- The chart does show percents of individuals in different categories!
- But, this is not a distribution because individuals can be in more than one category (see the fine print).



Does this chart show a distribution?

CC BY NC SA



Does this chart show a distribution?

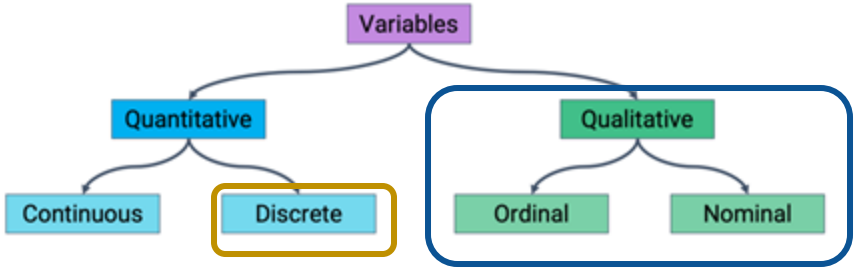
Yes!

- This chart shows the distribution of the qualitative ordinal variable "income tier."
- Each individual is in exactly one category.
- The values we see are the proportions of individuals in that category.
- Everyone is represented, as the total percentage is 100%.

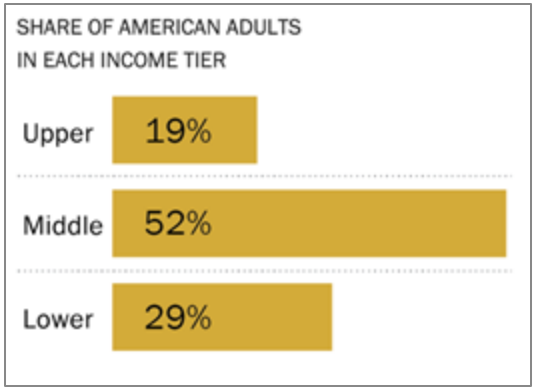
Bar Plots: Distributions of Qualitative Variables

Bar plots are the most common way of displaying the **distribution** of a **qualitative** variable.

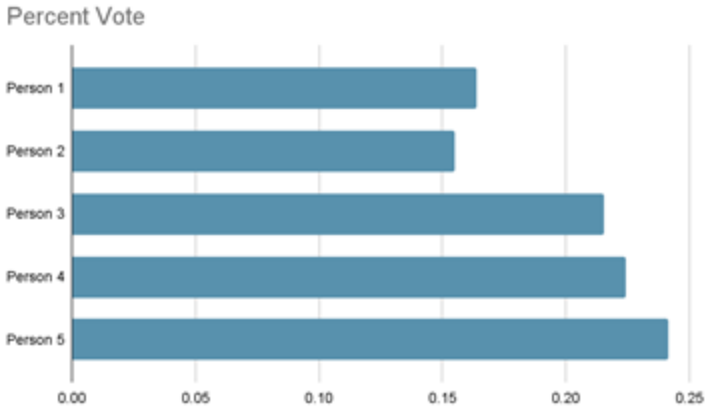
*Sometimes quantitative discrete data too, if there are few unique values.



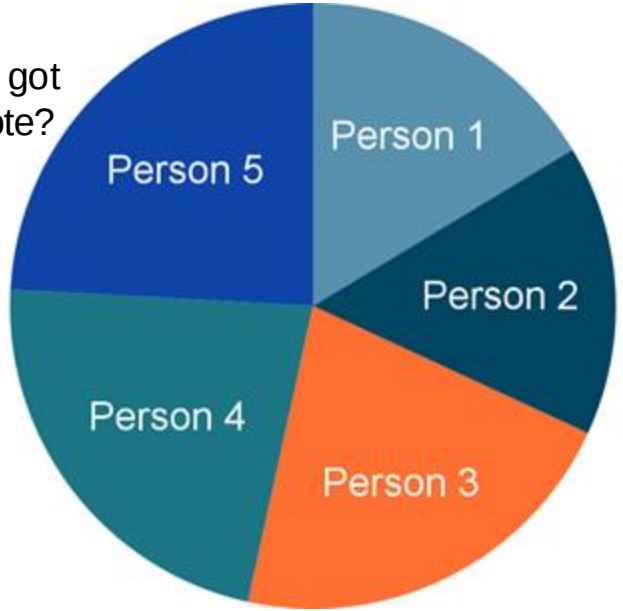
- Bar Charts:
  - One bar for each category
  - Length of bar is the percent (or count) of individuals in that category
  - Widths encode nothing: but bar widths should all be the same.
  - If ordinal - order of bars should reflect category order
  - Space between bars (not connected)
  - *Color* could indicate a sub-category (but not necessarily).



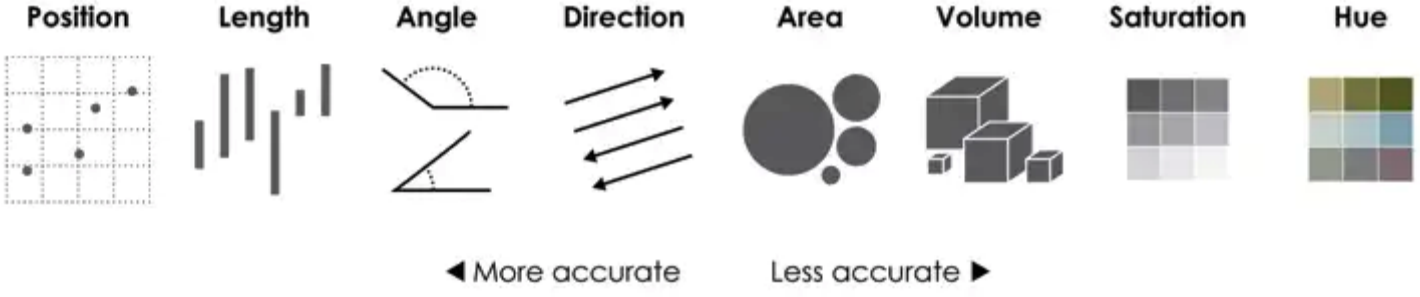
Why to Avoid Pie Charts



Which person got most of the vote?



Human Perception:



CC BY NC SA

https://uxdesign.cc/why-are-we-not-thinking-of-data-visualization-as-a-design-pattern-38e42ac86529

Example Dataset

We will be using data in the file baby.csv which contains data for 1774 mother-baby pairs in the 1960s. Here is what that looks like.

```
1 births = pd.read_csv('baby.csv')
```

```
1 births.head()
```

	Birth Weight	Gestational Days	Maternal Age	Maternal Height	Maternal Pregnancy Weight	Maternal Smoker
0	120	284	27	62	100	False
1	113	282	33	64	135	False
2	128	279	28	64	115	True
3	108	282	23	67	125	True
4	136	286	25	62	93	False

```
1 births.shape
```

```
(1174, 6)
```

CC BY NC SA

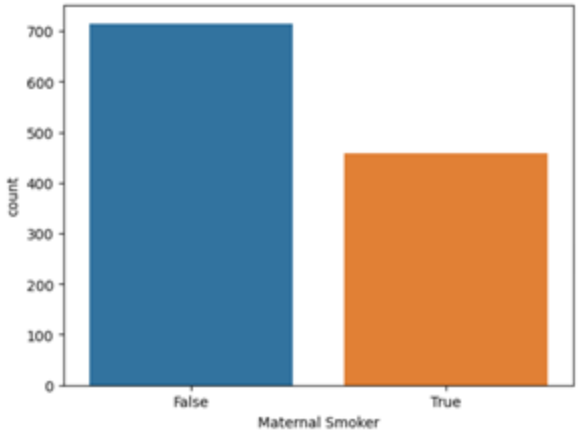
Bar Plots

`births['Maternal Smoker']` is a series containing True and False.

```
births['Maternal Smoker'].value_counts()
```

```
False    715
True     459
Name: Maternal Smoker, dtype: int64
```

We can visualize with a bar plot.



CC BY NC SA

Generating Bar Plots: Matplotlib

In this class we will mainly use 3 libraries for generating plots: Matplotlib, Seaborn and Plotly

Most Matplotlib plotting functions follow the same structure: We pass in a sequence (`list`, `array`, or `Series`) of values to be plotted on the x-axis, and a second sequence of values to be plotted on the y-axis.

```
import matplotlib.pyplot as plt
plt.plotting_function(x_values, y_values)
```

Matplotlib is typically given the alias `plt`

To add labels and a title:

```
plt.xlabel("x axis label")
plt.ylabel("y axis label")
plt.title("Title of the plot");
```

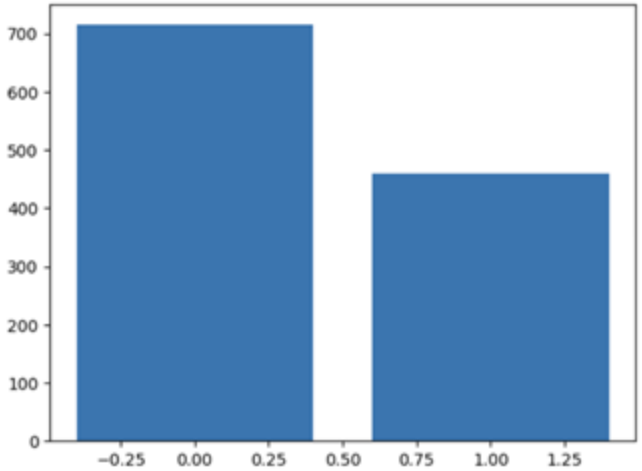
CC BY NC SA

Generating Bar Plots: Matplotlib

To create a bar plot in Matplotlib: `plt.bar( )`

```
babies = births['Maternal Smoker'].value_counts()
```

```
False    715
True     459
Name: Maternal Smoker, dtype: int64
```




```
plt.bar(babies.index,        babies.values);
```

x values

y values

CC BY NC SA

Generating Bar Plots: Seaborn

Seaborn plotting functions use a different structure: Pass in an entire `DataFrame`, then specify what column(s) to plot.

Seaborn is typically given the alias `sns`

```
import seaborn as sns
sns.plotting_function(data=df, x="x_col", y="y_col")
```

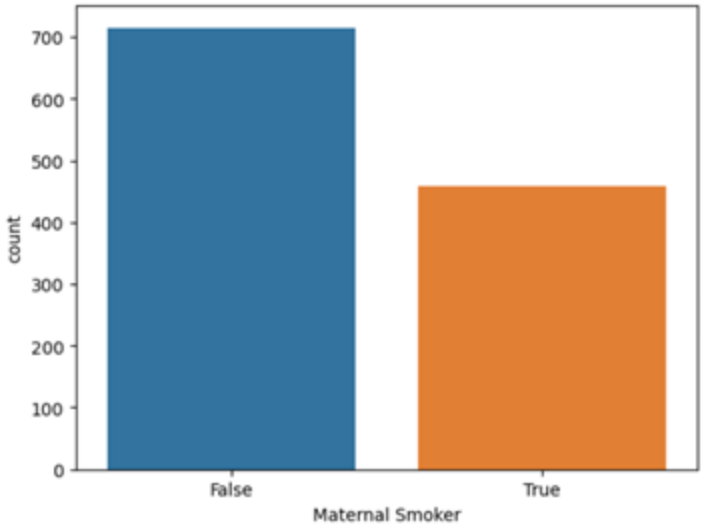
To add labels and a title, use the same syntax as before:

```
plt.xlabel("x axis label")
plt.ylabel("y axis label")
plt.title("Title of the plot");
```

CC BY NC SA

Generating Bar Plots: Seaborn

To create a bar plot in Seaborn: `sns.countplot( )`



`countplot` operates at a higher level of abstraction!

You give it the entire `DataFrame` and it does the counting for you.

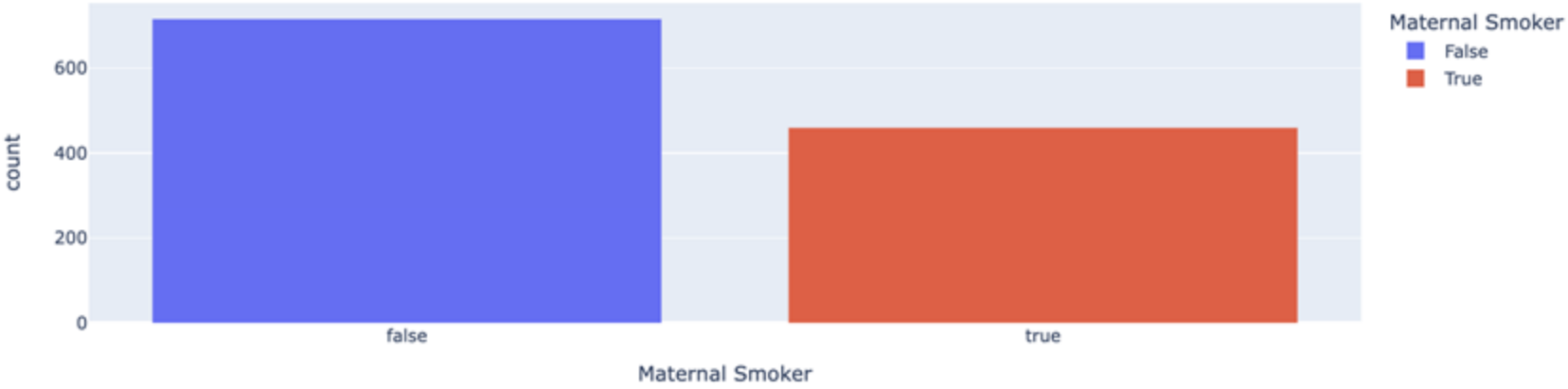
```
import seaborn as sns

sns.countplot(data = births, x = 'Maternal Smoker');
```

Generating Bar Plots: Plotly

Plotly follows the structure of Seaborn: Pass in an entire `DataFrame`, then specify what column(s) to plot.
Plotly plots are interactive (if you hover over them it will give you counts)

Plotly is typically given the alias `sns`

```
import plotly.express as px
px.plotting_function(data=df, x="x_col", y="y_col")
```

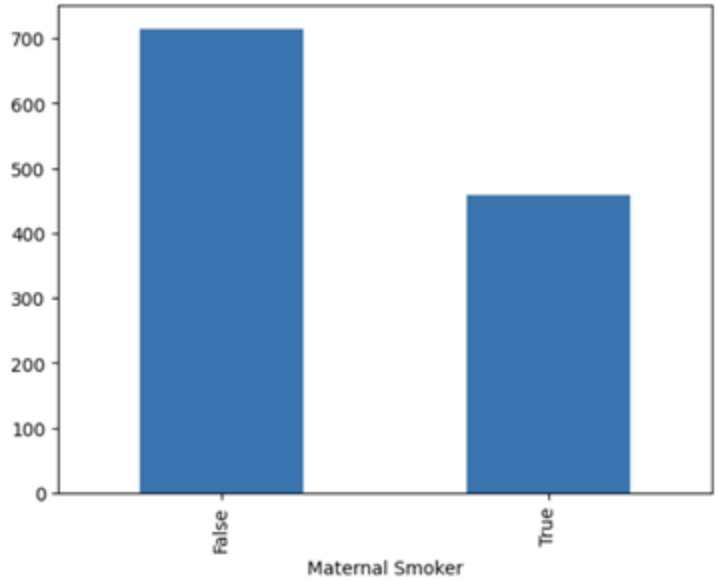


CC BY NC SA

Generating Bar Plots: pandas Native Plotting

To create a bar plot in native `pandas`: `.plot(kind='bar')`

```
births['Maternal Smoker'].value_counts().plot(kind='bar')
```



CC BY NC SA