Data Visualization: matplotlib

Nazgul Rakhimzhanova

IITU 2021

OUTLINE

- **Previously**
- **Data variables** and visualization
- **Visualization** with matplotlib
- **Readings**

PREVIUOSLY

- Discussed about **data variables**
- Did overview of the **Pandas package**
- Practiced **Pandas** functions and objects

What do you remember about each activity?

COURSE SCHEDULE

week	Mid Term (weeks 01-07)	End Term (weeks 08-14)	week
01	Intro: Data Science Area and open source tools for Data Science	Statistics: Distribution – Lognormal, Exponential	08
02	NumPy package for data science	Sampling and Estimation	09
03	Pandas package for data science	Correlation and Covariance	10
04	Visualization with matplotlib	Hypothesis testing	11
05	Statistics: Distribution – Normal	Decision Tree	12
06	Exploratory Data Analysis (EDA)	Linear Regression	13
07	**Summary for 6 weeks QA session**	**Summary for 6 weeks QA session**	**14**
15	**Course summary**		

PREVIUOSLY

Have you thought about additional DS methods applications?

DATA

DATA → INFORMATION → KNOWLEDGE → UNDESTANDING

How to make information **understandable**?

DATA

What is **visualization**?

DATA VISUALIZATION

- **Based on (non-visual) data.** A visualization's purpose is the communication of data. That means that the data must come from something that is abstract or at least not immediately visible (like the inside of the human body). This rules out photography and image processing. Visualization transforms from the invisible to the visible.
- **Produce an image.** It may seem obvious that a visualization has to produce an image, but that is not always so clear. Also, the visual must be the primary means of communication, other modalities can only provide additional information. If the image is only a small part of the process, it is not visualization.
- **The result must be readable and recognizable.** The most important criteria is that the visualization must provide a way to learn something about the data. Any transformation of non-trivial data into an image will leave out information, but there must be at least some relevant aspects of the data that can be read. The visualization must also be recognizable as one and not pretend to be something else

DATA VISUALIZATION

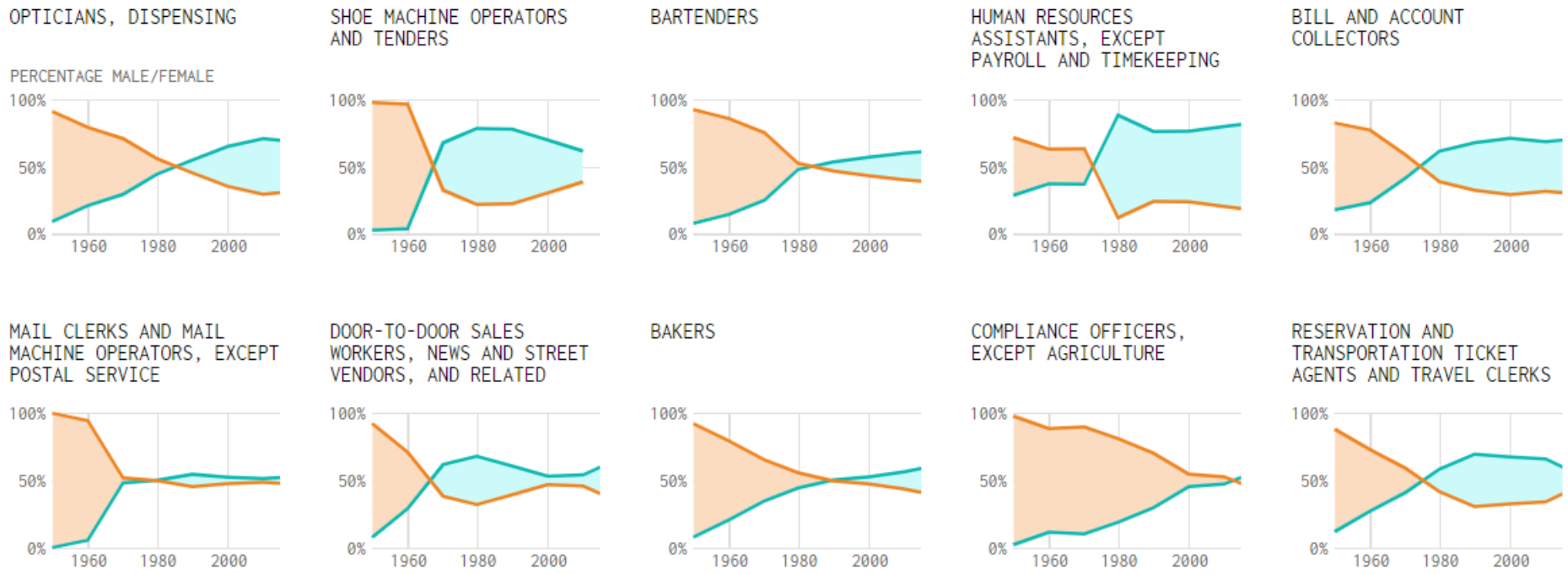
Common general types of data visualization:

- Charts
- Tables
- Graphs
- Maps
- Infographics
- Dashboards

DATA VISUALIZATION

ONCE MALE, SHIFTING TO MAJORITY FEMALE

As more women entered the workforce, many occupations saw a shift from mostly male to a majority or more female. Here are the 10 biggest changes.

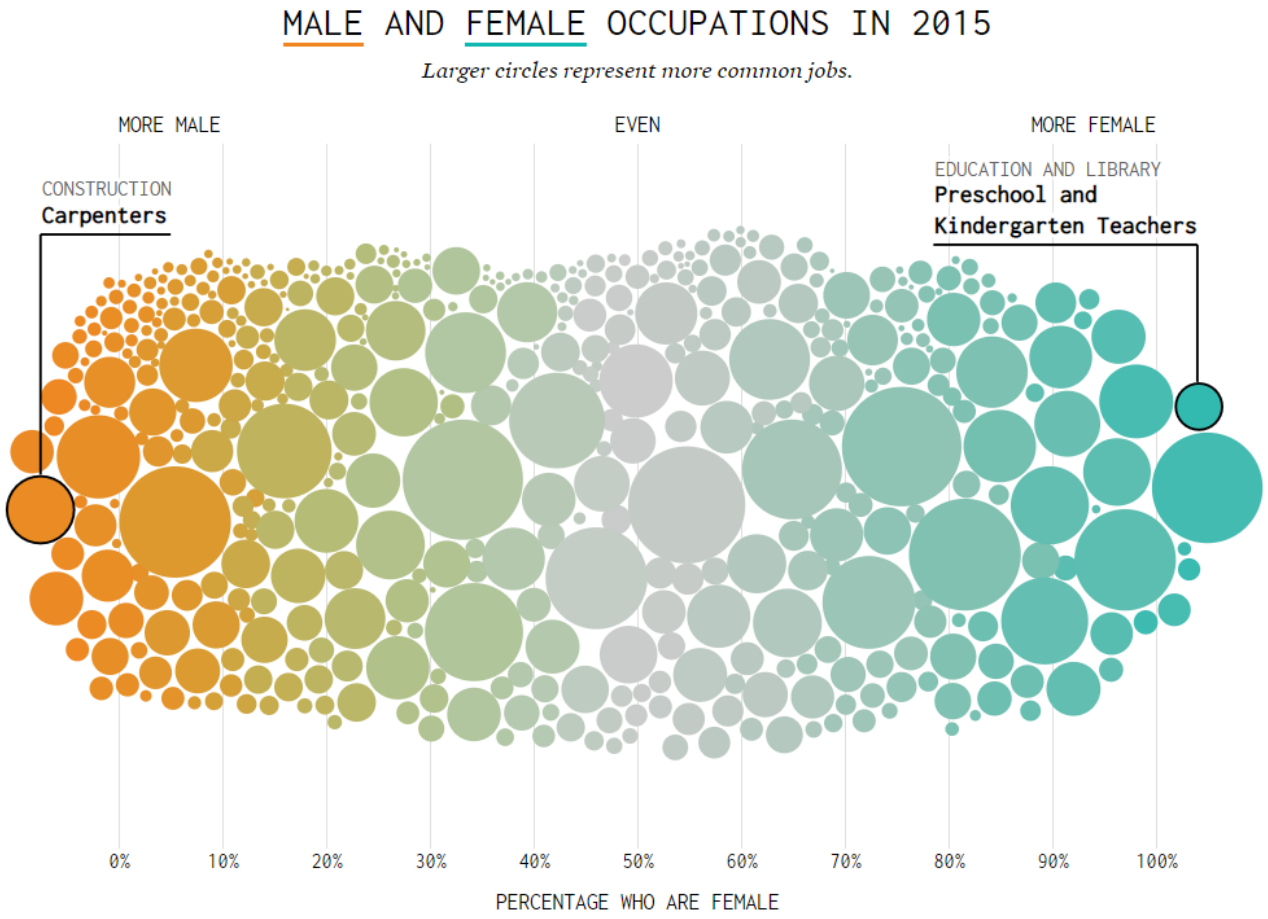


DATA VISUALIZATION

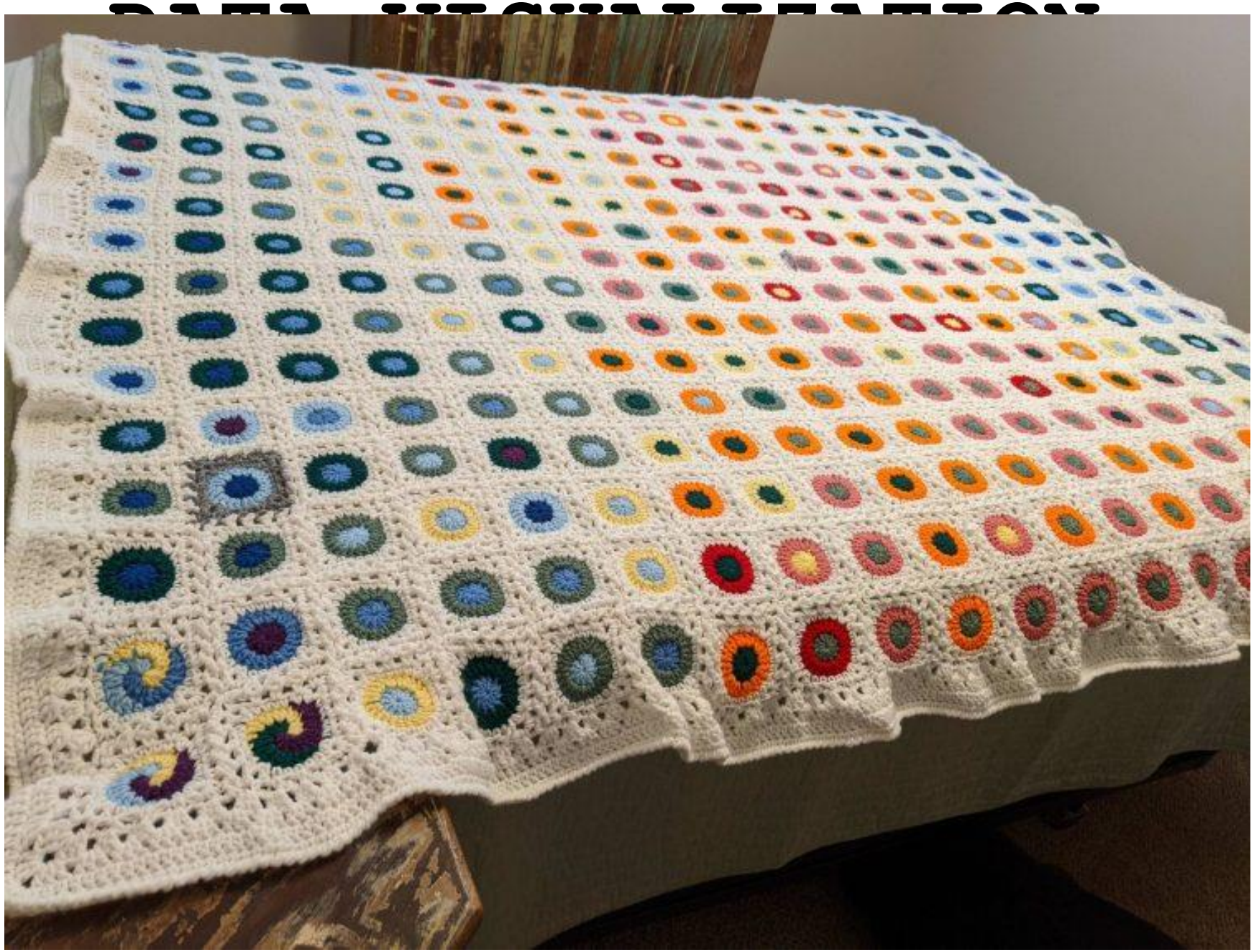
https://flowingdata.com/2017/06/05/how-many-friends-you-need-to-have-a-birthday-every-day-of-the-year/

DATA VISUALIZATION

https://flowingdata.com/2017/09/11/most-female-and-male-occupations-since-1950/



DATA VISUALIZATION

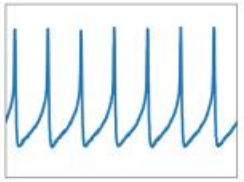
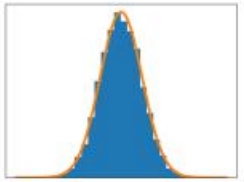
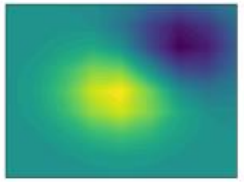
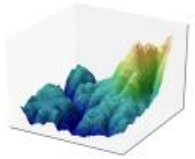


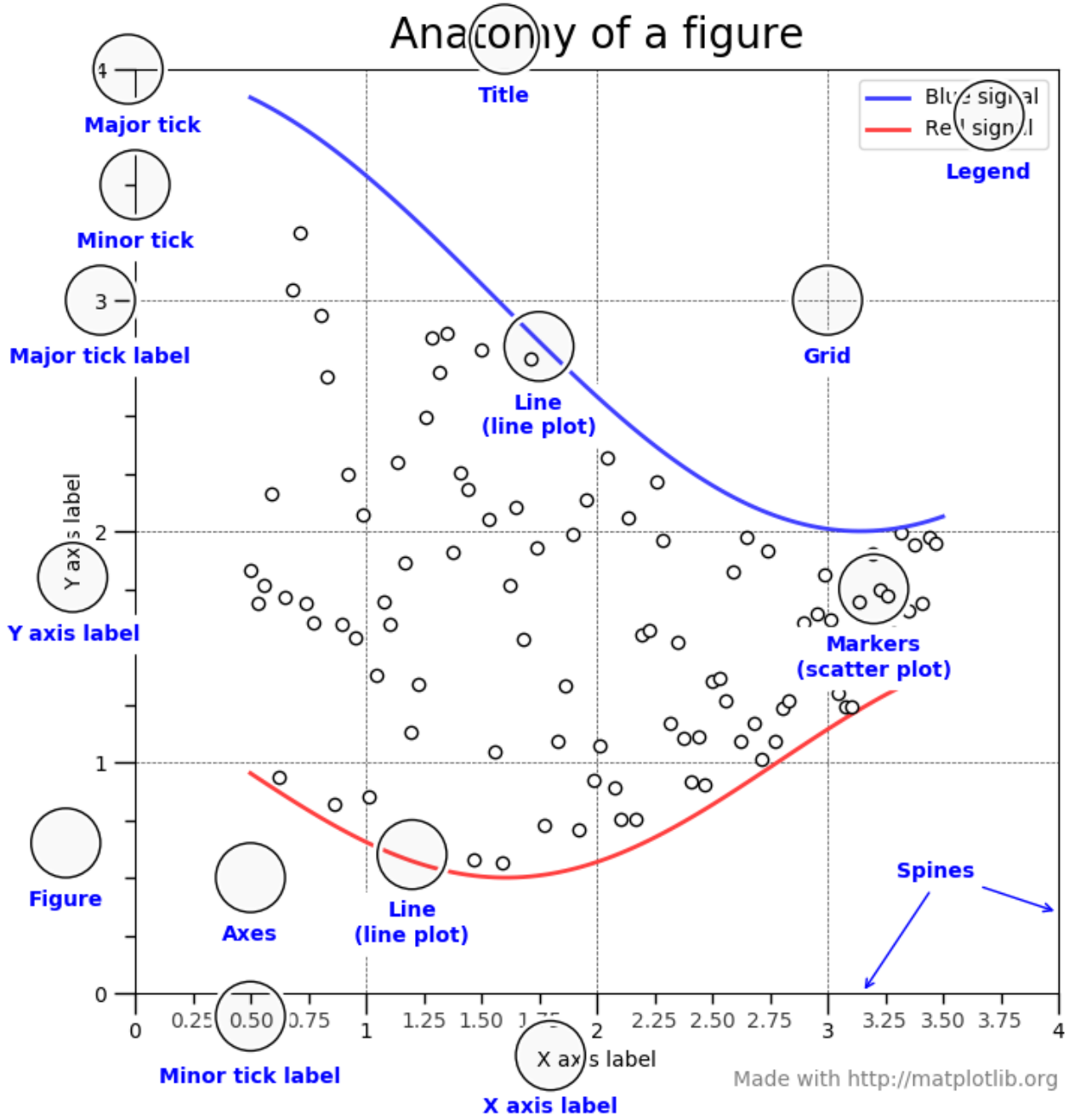
DATA VISUALIZATION

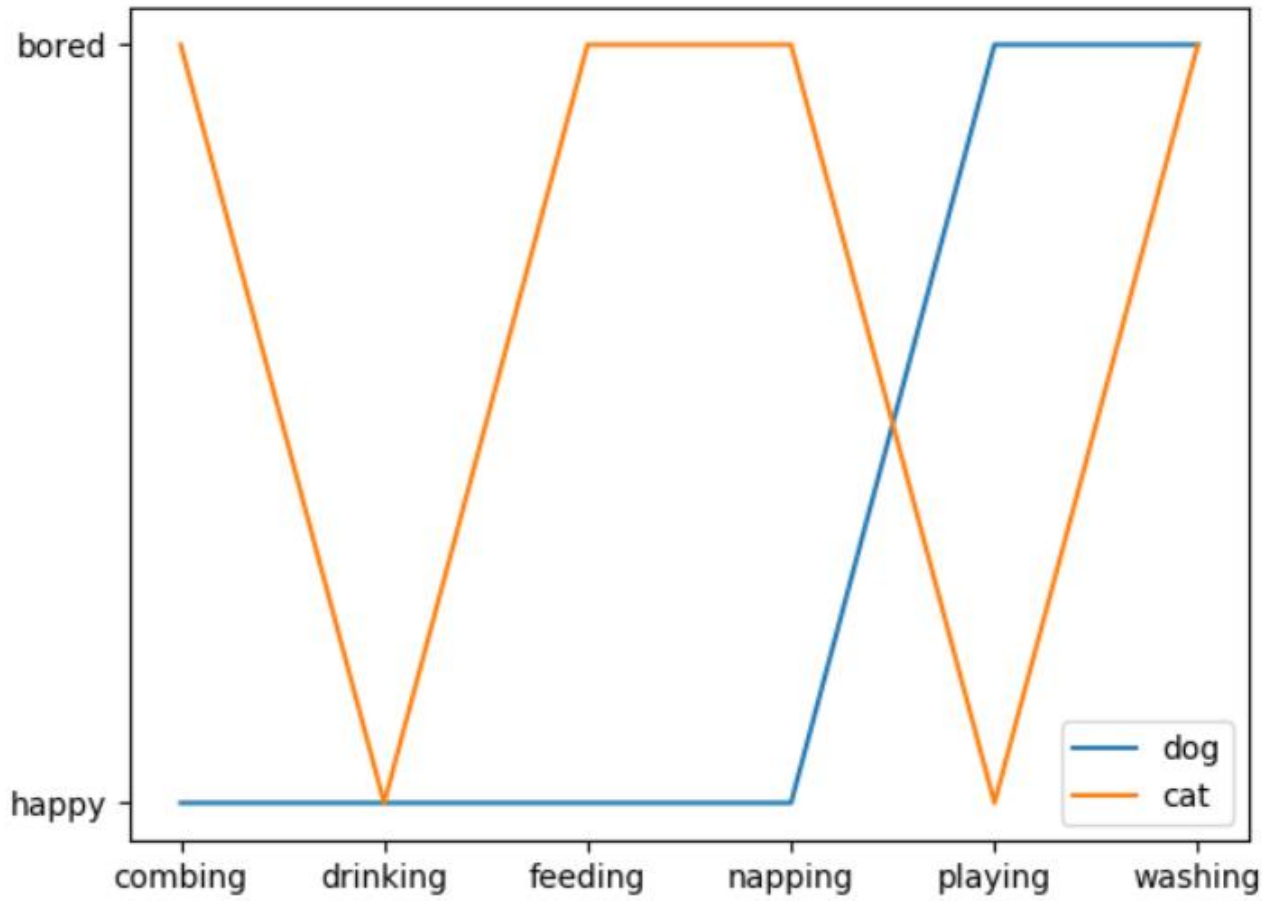
- To explore data
- To communicate data

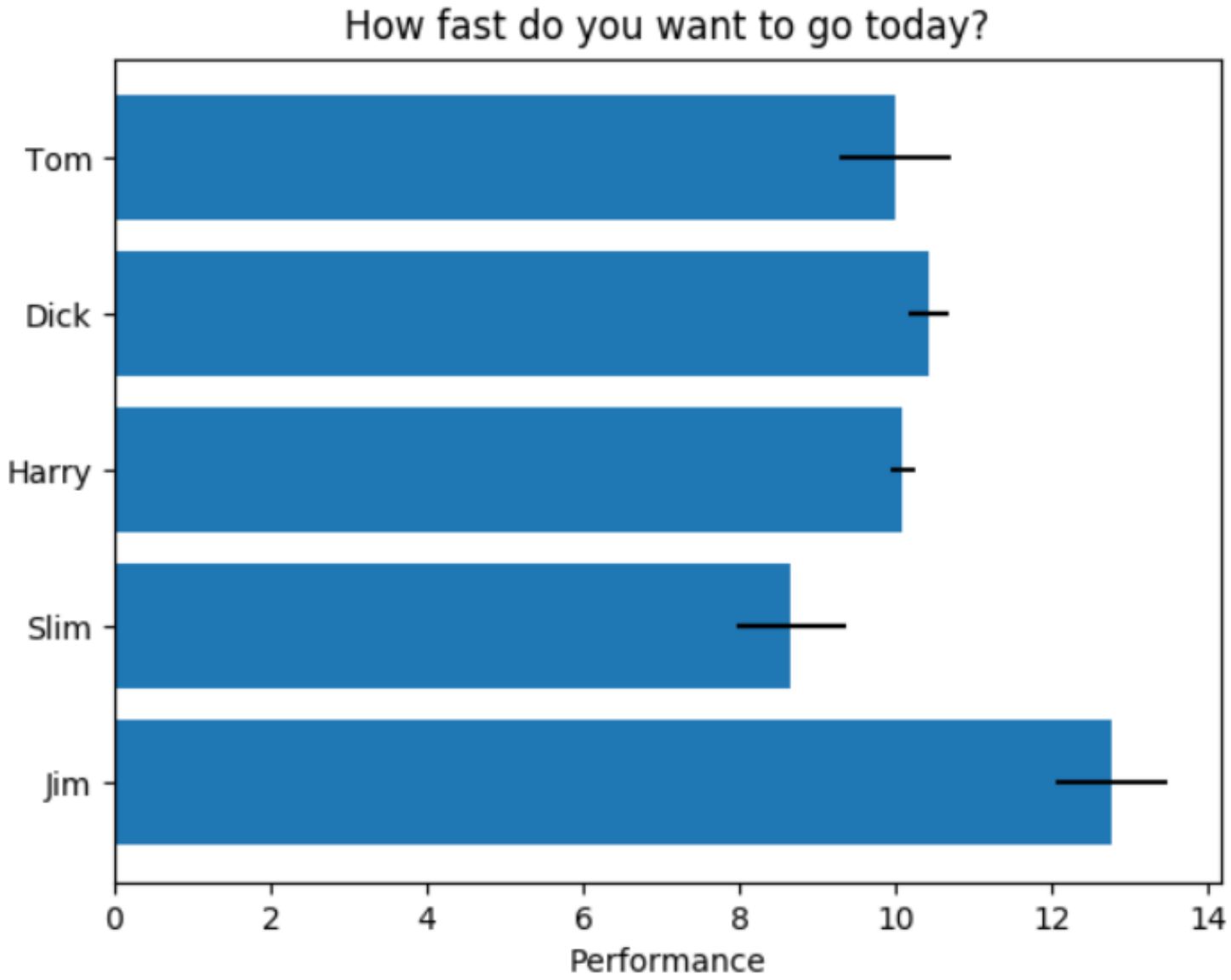
matplotlib

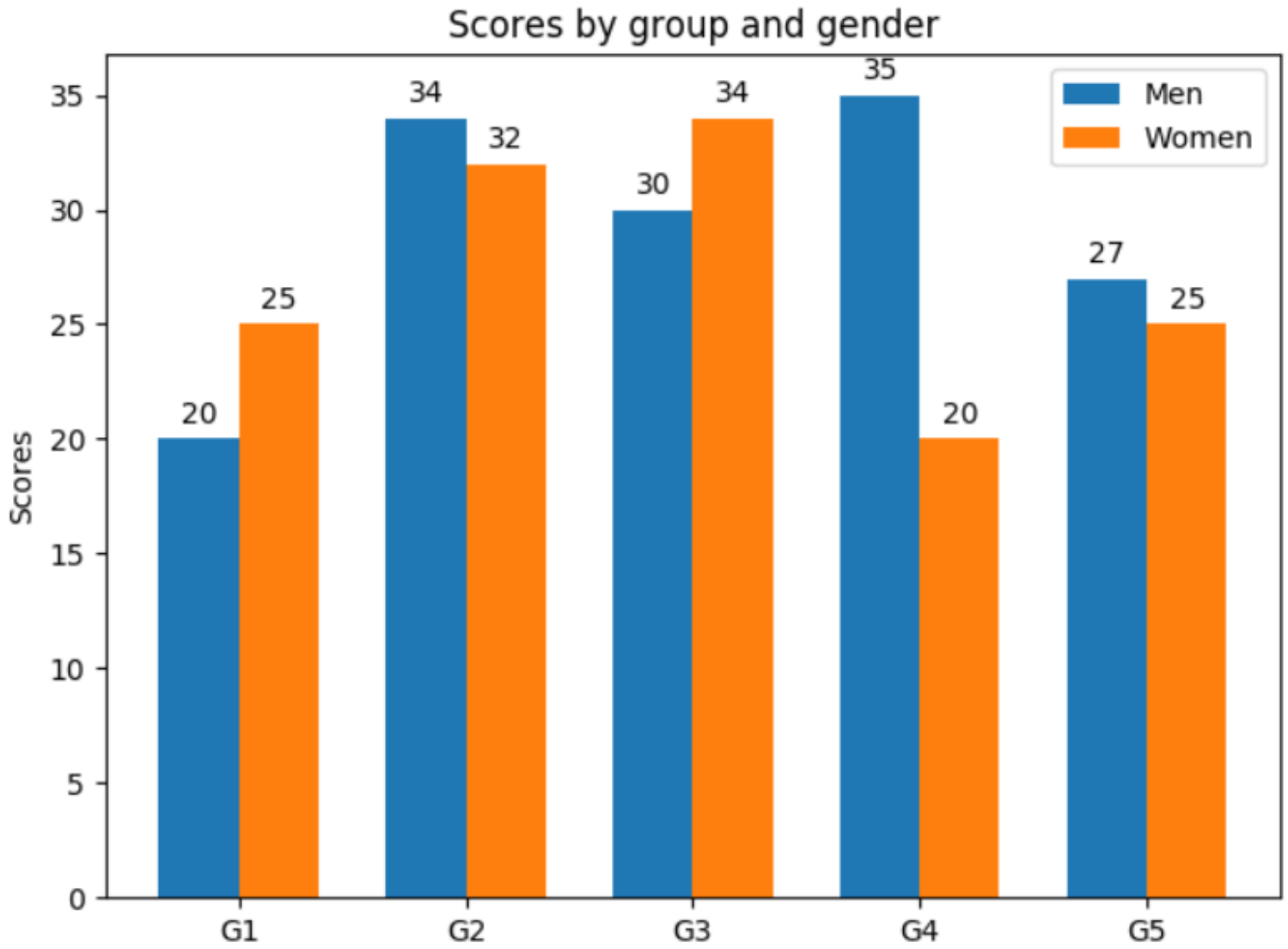
- **Matplotlib** is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.
- **For simple** plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython.


Anatomy of a figure
Title
Major tick
Minor tick
Major tick label
Legend
Grid
Line
(line plot)
Y axis label
Markers
(scatter plot)
Figure
Axes
Line
(line plot)
Spines
Minor tick label
X axis label
Made with http://matplotlib.org


bored
happy
combing
drinking
feeding
napping
playing
washing
dog
cat

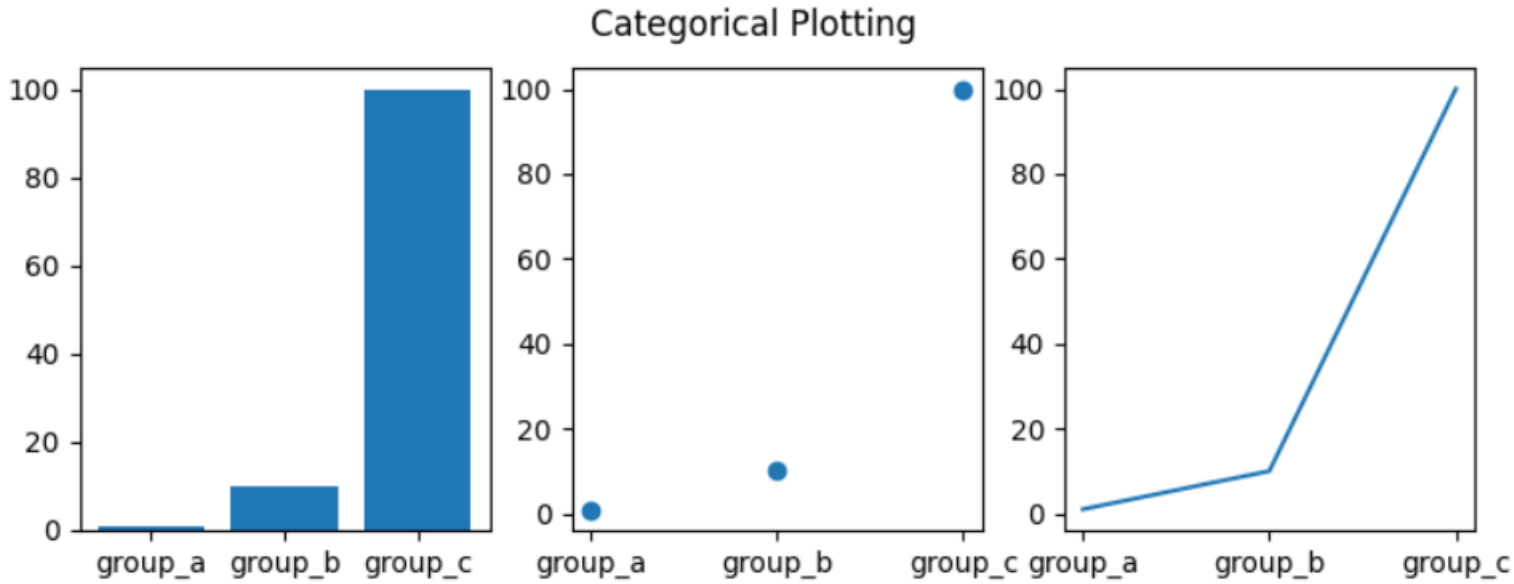

How fast do you want to go today?
Tom
Dick
Harry
Slim
Jim
0
2
4
6
8
10
12
14
Performance


Scores by group and gender
Men
Women
Scores
0
5
10
15
20
25
30
35
20
25
34
32
30
34
35
20
27
25
G1
G2
G3
G4
G5

```
names = ['group_a', 'group_b', 'group_c']
values = [1, 10, 100]

plt.figure(figsize=(9, 3))

plt.subplot(131)
plt.bar(names, values)
plt.subplot(132)
plt.scatter(names, values)
plt.subplot(133)
plt.plot(names, values)
plt.suptitle('Categorical Plotting')
plt.show()
```



https://matplotlib.org/stable/tutorials/introductory/pyplot.html#sphx-glr-tutorials-introductory-pyplot-py

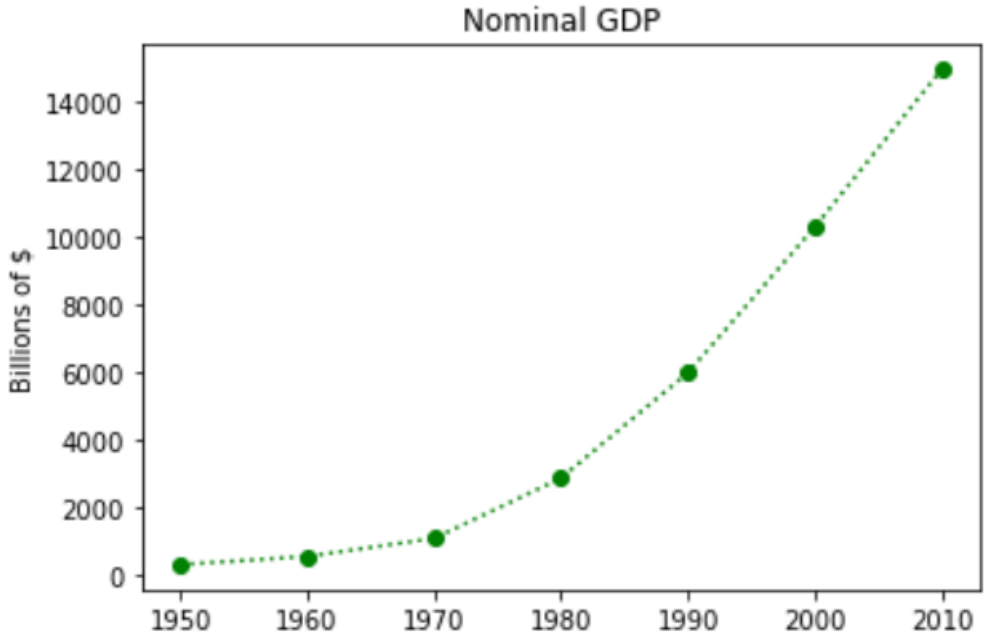
How to visualize the best?

#	Years	GDP, bill $
1	1950	300,2
2	1960	543,3
3	1970	1075,9
4	1980	2862,5
5	1990	5979,6
6	2000	10289,7
7	2010	14958,3

Line Charts

In [3]:
```
years=[1950,1960,1970,1980,1990,2000,2010]
gdp=[300.2,543.3,1075.9,2862.5,5979.6,10289.7,14958.3]
```

In [23]:
```
plt.plot(years, gdp, color='green', marker = 'o', linestyle = 'dotted')
plt.title('Nominal GDP')
plt.ylabel('Billions of $')
plt.show()
```



Line Charts

- **Line** graphs are used to track changes over short and long periods of time.
- **Line** graphs can also be used to compare changes over the same period of time for more than one group.

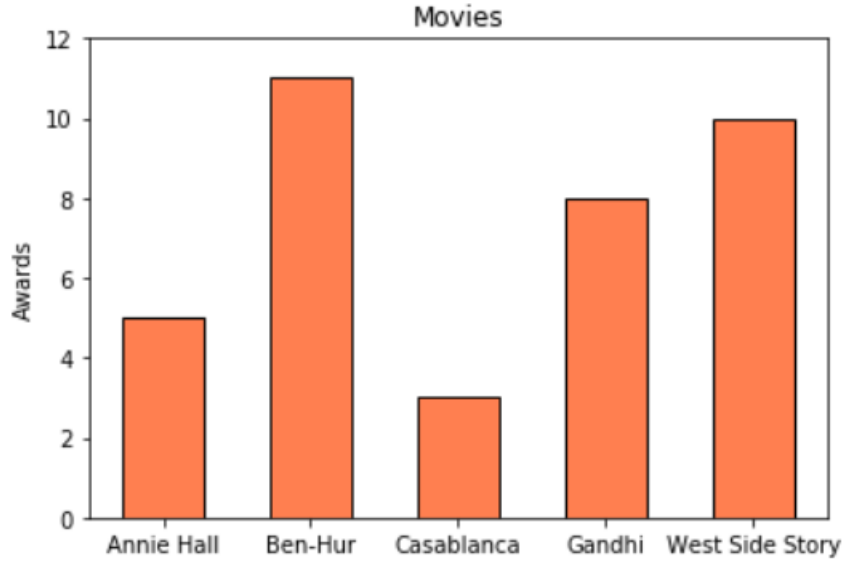
How to visualize the best?

#	Picture	Oscars
1	Annie Hall	5
2	Ben-Hur	11
3	Casablanca	3
4	Gandhi	8
5	West Side Story	10

Bar Charts

In [27]:
```
movies=["Annie Hall", "Ben-Hur", "Casablanca", "Gandhi", "West Side Story"]
oscars = [5, 11, 3, 8, 10]
```

In [48]:
```
xs=[i+0.1 for i,_ in enumerate(movies)]
plt.bar(xs, oscars, 0.55, color = '#ff7f50', edgecolor = 'black')
plt.ylabel('Awards');
plt.title('Movies');
plt.xticks([i+0.1 for i,_ in enumerate(movies)], movies)
plt.yticks(np.arange(0, 13, 2))
plt.show()
```



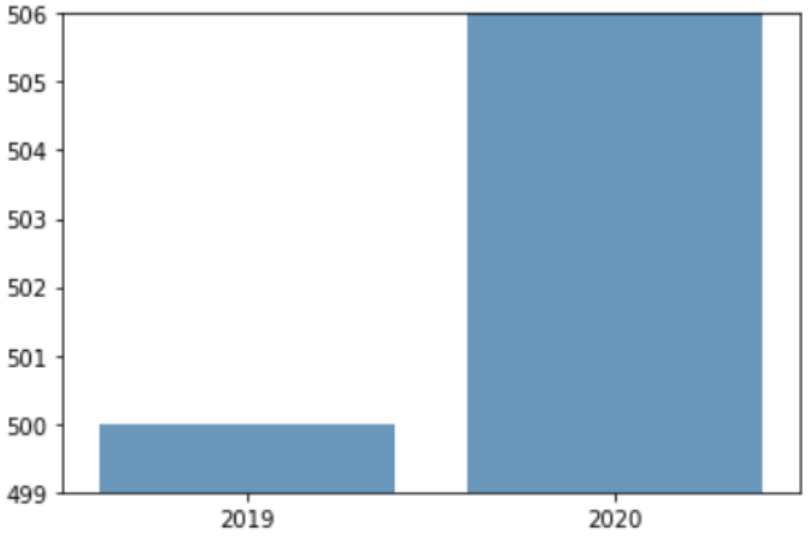
Bar Charts

- **Bar** graphs are used to compare things between different groups or to track changes over time
- **However,** when trying to measure change over time, bar graphs are best when the changes are larger.

Tricky Bar Charts

In [125]:
```
m = [500, 506]
years = [2019, 2020]
```

In [144]:
```
plt.bar(years, m, 0.8, color = '#6897bb')
plt.xticks(years)
plt.axis([2018.5, 2020.5, 499, 506])
plt.ticklabel_format(useOffset=False)
plt.show()
```

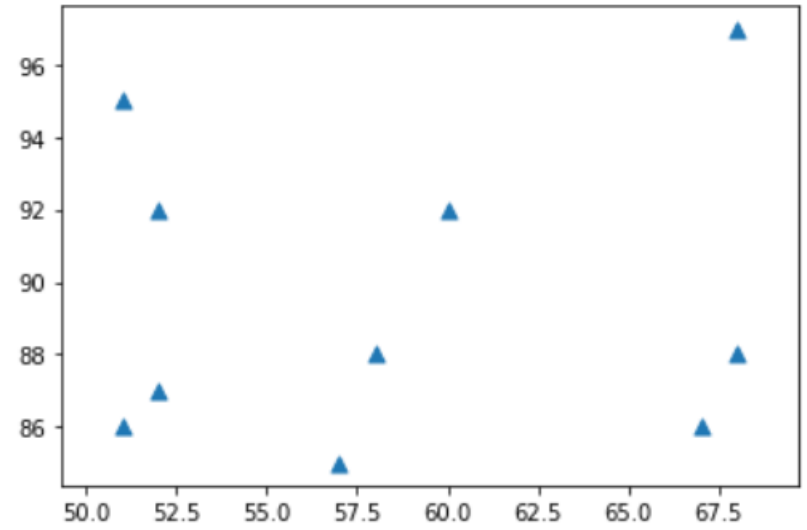


How to visualize the best?

Temperature	Rainfall
27	2,00
28,9	1,33
34,5	1,33
44,1	1,05
42,4	1,00
25,3	1,54
32,9	1,54

Scatter plots

In [164]:
```
n = np.random.randint(50, 70, 10)
f = np.random.randint(80, 100, 10)
plt.scatter(n, f, 50, marker = '^')
plt.axis('equal')
plt.show()
```



Scatter plots

- **A scatterplots** is the right choice for visualizing the relationship between two paired sets of data

Initial Data

	B	C	D	E	F	G	H
	Date	**Day**	**Temperature**	**Rainfall**	**Flyers**	**Price**	**Sales**
	01.01.2017	Sunday	27	2,00	15	0,3	10
	02.01.2017	Monday	28,9	1,33	15	0,3	13
	03.01.2017	Tuesday	34,5	1,33	27	0,3	15
	04.01.2017	Wednesday	44,1	1,05	28	0,3	17
	05.01.2017	Thursday	42,4	1,00	33	0,3	18
	06.01.2017	Friday	25,3	1,54	23	0,3	11
	07.01.2017	Saturday	32,9	1,54	19	0,3	13
	08.01.2017	Sunday	37,5	1,18	28	0,3	15
	09.01.2017	Monday	38,1	1,18	20	0,3	17

What to visualize?

Readings

- https://matplotlib.org/stable/index.html
- https://matplotlib.org/stable/tutorials/introductory/usage.html#sphx-glr-tutorials-introductory-usage-py
- **Data Science from Scratch**, Book by Joel Grus