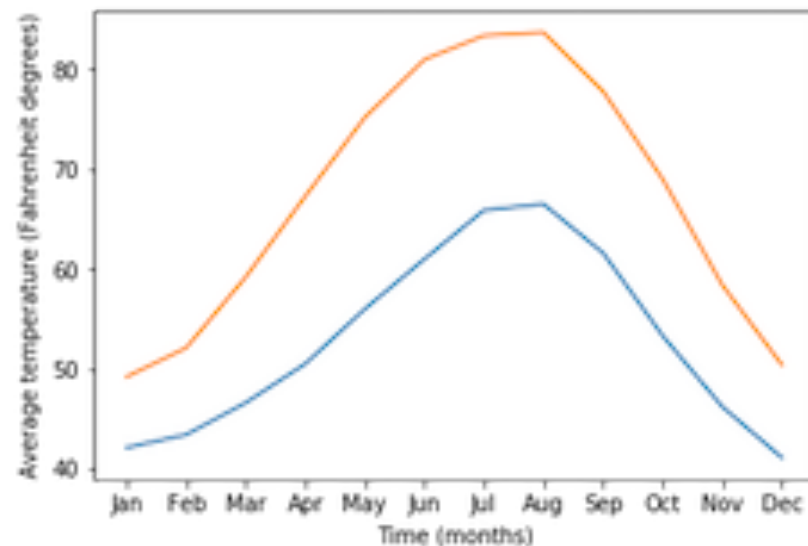


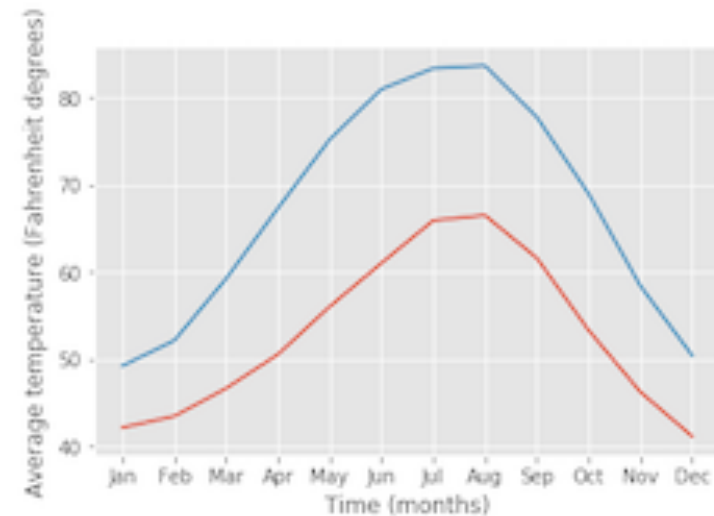
# Changing plot style

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots()
ax.plot(seattle_weather["MONTH"], seattle_weather["MLY-TAVG-NORMAL"])
ax.plot(austin_weather["MONTH"], austin_weather["MLY-TAVG-NORMAL"])
ax.set_xlabel("Time (months)")
ax.set_ylabel("Average temperature (Fahrenheit degrees)")
plt.show()
```



# Choosing a style

```
plt.style.use("ggplot")
fig, ax = plt.subplots()
ax.plot(seattle_weather["MONTH"], seattle_weather["MLY-TAVG-NORMAL"])
ax.plot(austin_weather["MONTH"], austin_weather["MLY-TAVG-NORMAL"])
ax.set_xlabel("Time (months)")
ax.set_ylabel("Average temperature (Fahrenheit degrees)")
plt.show()
```



# Back to the default

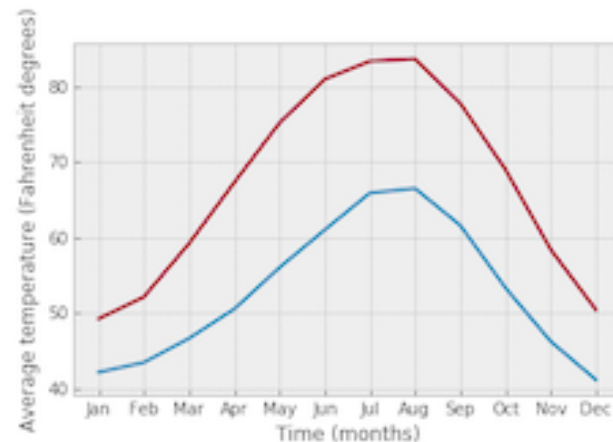
```
plt.style.use("default")
```

# The available styles

[https://matplotlib.org/gallery/style\\_sheets/style\\_sheets\\_refere](https://matplotlib.org/gallery/style_sheets/style_sheets_refere)

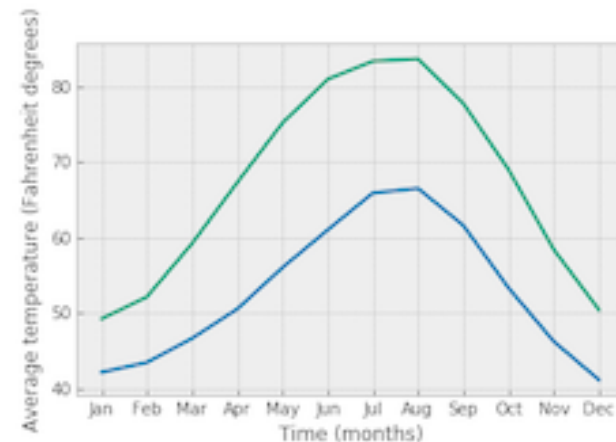
# The "bmh" style

```
plt.style.use("bmh")
fig, ax = plt.subplots()
ax.plot(seattle_weather["MONTH"], seattle_weather["MLY-TAVG-NORMAL"])
ax.plot(austin_weather["MONTH"], austin_weather["MLY-TAVG-NORMAL"])
ax.set_xlabel("Time (months)")
ax.set_ylabel("Average temperature (Fahrenheit degrees)")
plt.show()
```



# Seaborn styles

```
plt.style.use("seaborn-colorblind")
fig, ax = plt.subplots()
ax.plot(seattle_weather["MONTH"], seattle_weather["MLY-TAVG-NORMAL"])
ax.plot(austin_weather["MONTH"], austin_weather["MLY-TAVG-NORMAL"])
ax.set_xlabel("Time (months)")
ax.set_ylabel("Average temperature (Fahrenheit degrees)")
plt.show()
```



# Guidelines for choosing plotting style

- Dark backgrounds are usually less visible
- If color is important, consider choosing colorblind-friendly options
  - "seaborn-colorblind" or "tableau-colorblind10"
- If you think that someone will want to print your figure, use less ink
- If it will be printed in black-and-white, use the "grayscale" style

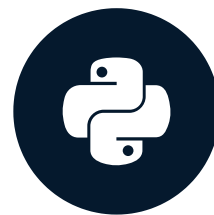
# Practice choosing the right style for you!

INTRODUCTION TO DATA VISUALIZATION WITH MATPLOTLIB



# Sharing your visualizations with others

INTRODUCTION TO DATA VISUALIZATION WITH MATPLOTLIB



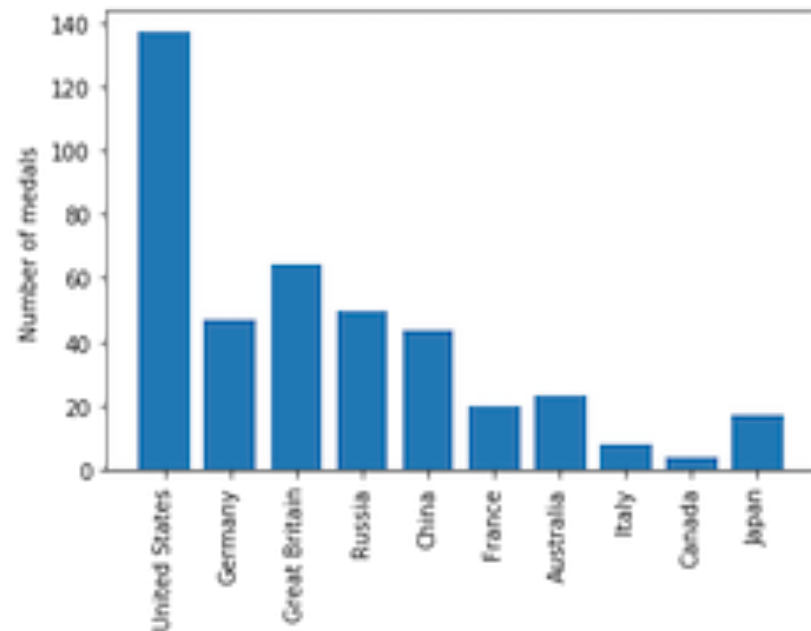
**Ariel Rokem**  
Data Scientist

# A figure to share

```
fig, ax = plt.subplots()

ax.bar(medals.index, medals["Gold"])
ax.set_xticklabels(medals.index, rotation=90)
ax.set_ylabel("Number of medals")

plt.show()
```



# Saving the figure to file

```
fig, ax = plt.subplots()

ax.bar(medals.index, medals["Gold"])
ax.set_xticklabels(medals.index, rotation=90)
ax.set_ylabel("Number of medals")

fig.savefig("gold_medals.png")
```

```
ls
```

```
gold_medals.png
```

# Different file formats

```
fig.savefig("gold_medals.jpg")
```

```
fig.savefig("gold_medals.jpg", quality=50)
```

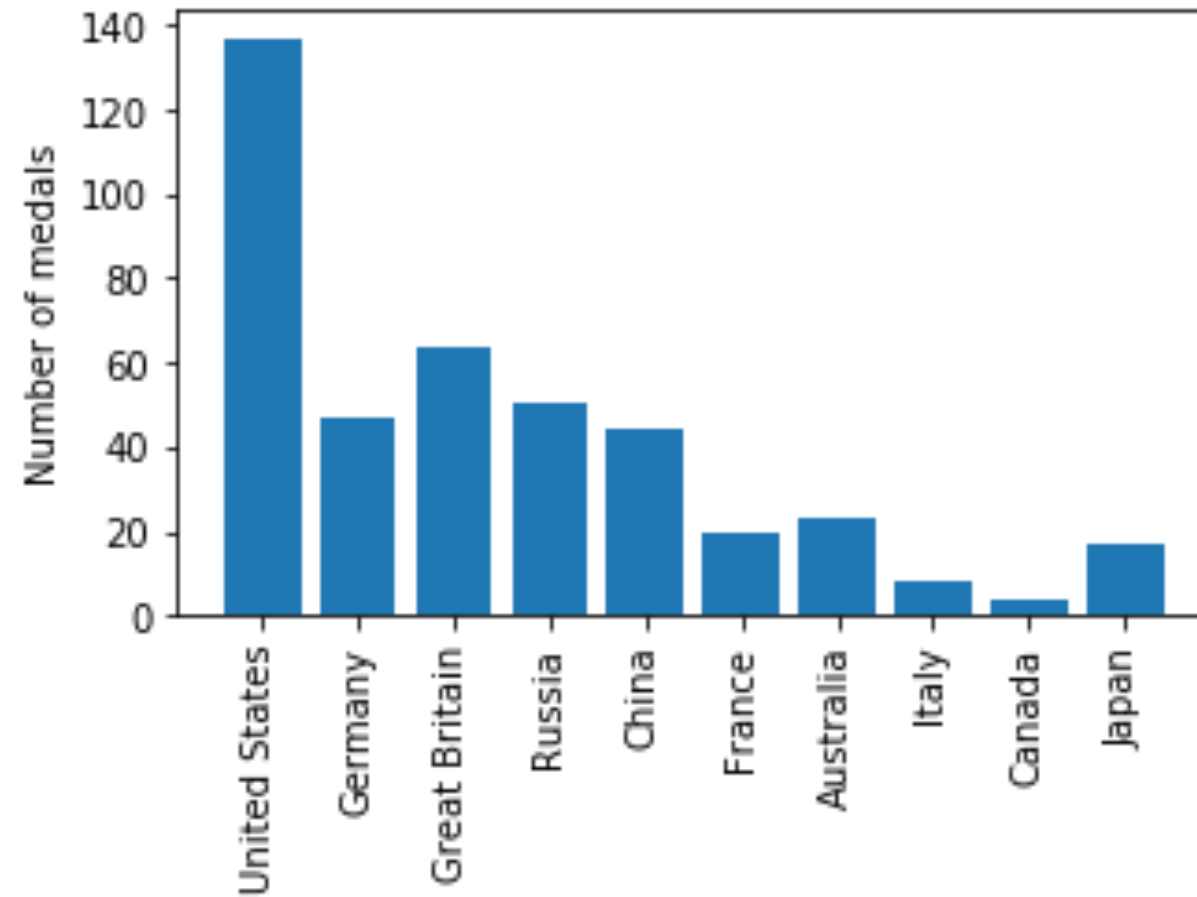
```
fig.savefig("gold_medals.svg")
```

# Resolution

```
fig.savefig("gold_medals.png", dpi=300)
```

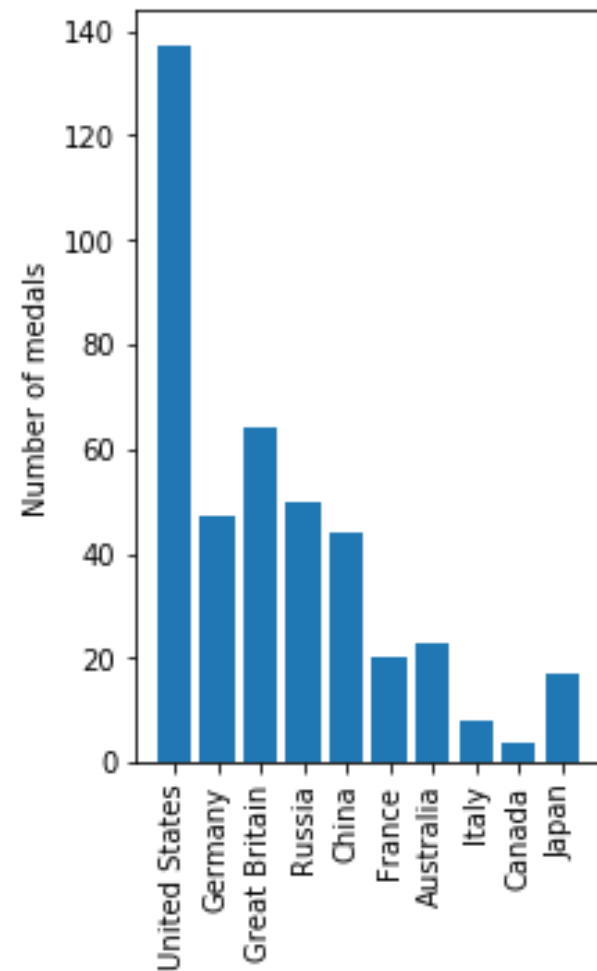
# Size

```
fig.set_size_inches([5, 3])
```



# Another aspect ratio

```
fig.set_size_inches([3, 5])
```



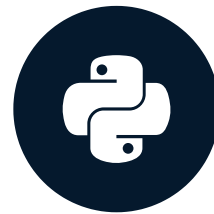
# Practice saving your visualizations!

INTRODUCTION TO DATA VISUALIZATION WITH MATPLOTLIB



# Automating figures from data

INTRODUCTION TO DATA VISUALIZATION WITH MATPLOTLIB



**Ariel Rokem**  
Data Scientist

# Why automate?

- Ease and speed
- Flexibility
- Robustness
- Reproducibility

# How many different kinds of data?

```
summer_2016_medals["Sport"]
```

```
ID
62      Rowing
65      Taekwondo
73      Handball
...
134759   Handball
135132   Volleyball
135205   Boxing
Name: Sport, Length: 976, dtype: object
```

# Getting unique values of a column

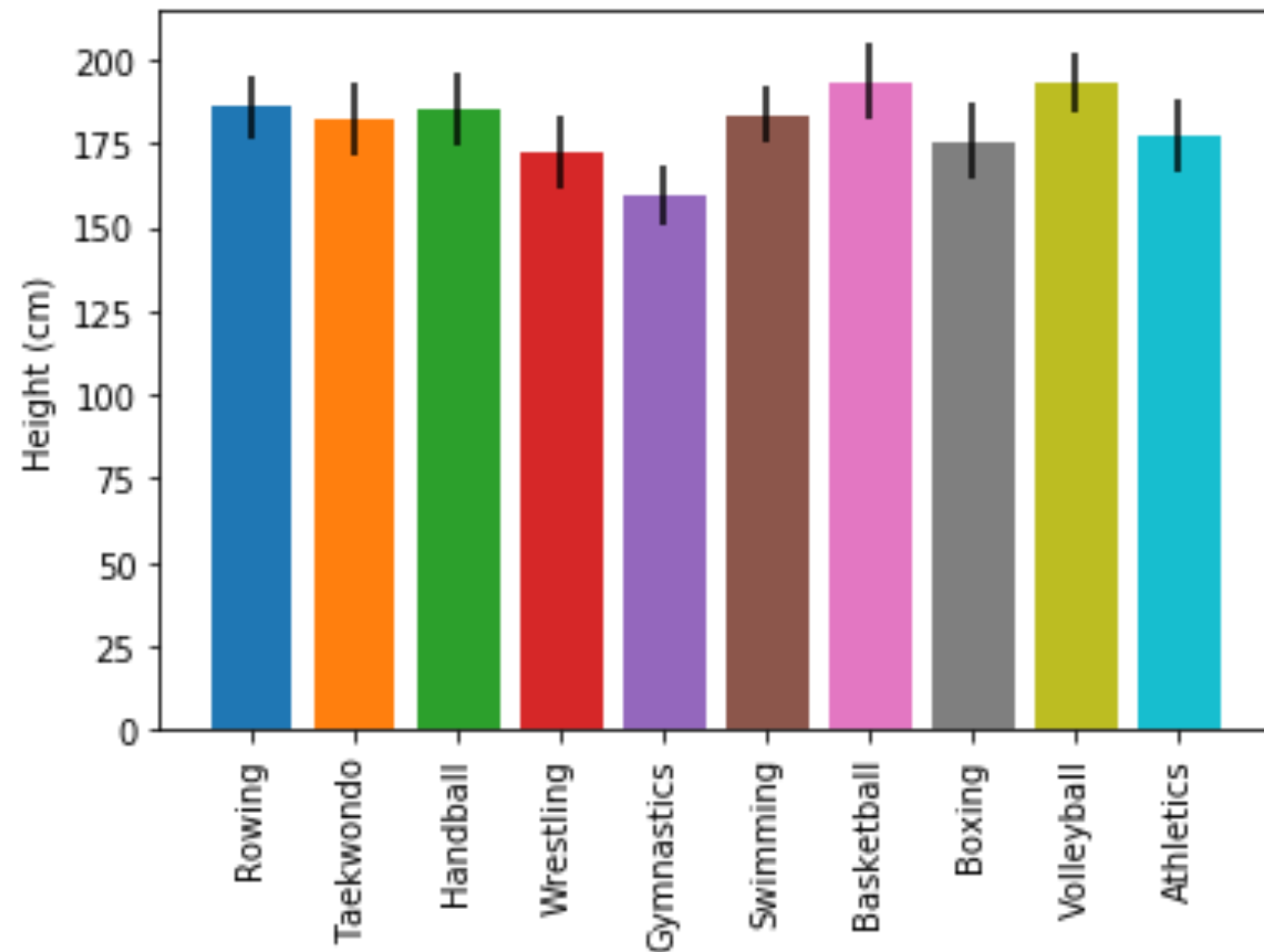
```
sports = summer_2016_medals["Sport"].unique()
print(sports)
['Rowing' 'Taekwondo' 'Handball' 'Wrestling'
 'Gymnastics' 'Swimming' 'Basketball' 'Boxing'
 'Volleyball' 'Athletics']
```

# Bar-chart of heights for all sports

```
fig, ax = plt.subplots()

for sport in sports:
    sport_df = summer_2016_medals[summer_2016_medals["Sport"] == sport]
    ax.bar(sport, sport_df["Height"].mean(),
           yerr=sport_df["Height"].std())
ax.set_ylabel("Height (cm)")
ax.set_xticklabels(sports, rotation=90)
plt.show()
```

# Figure derived automatically from the data

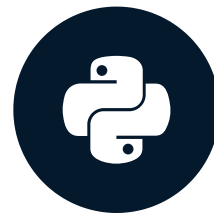


# Practice automating visualizations!

INTRODUCTION TO DATA VISUALIZATION WITH MATPLOTLIB

# Where to go next

INTRODUCTION TO DATA VISUALIZATION WITH MATPLOTLIB



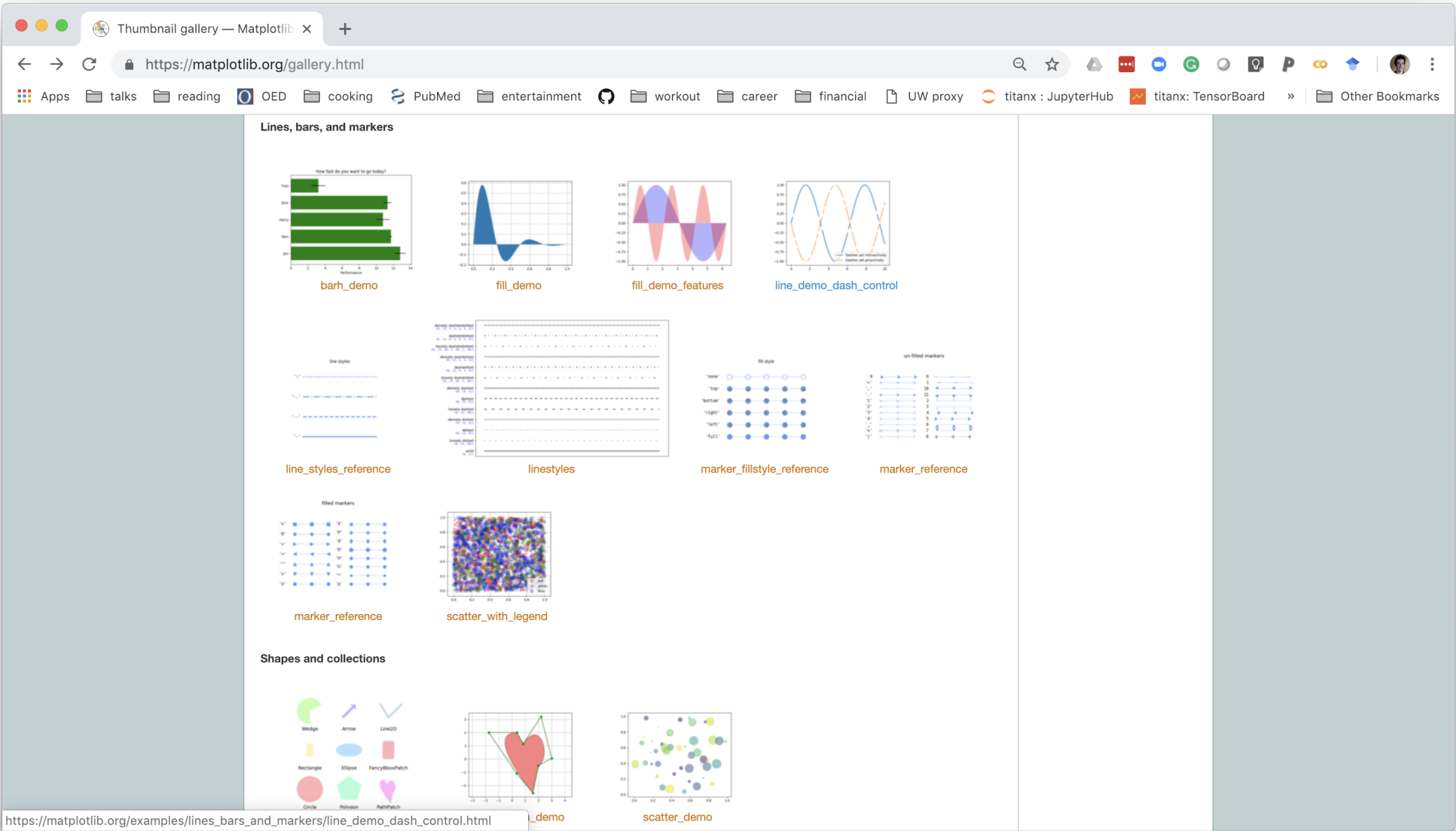
**Ariel Rokem**  
Data Scientist



# The Matplotlib gallery

<https://matplotlib.org/gallery.html>

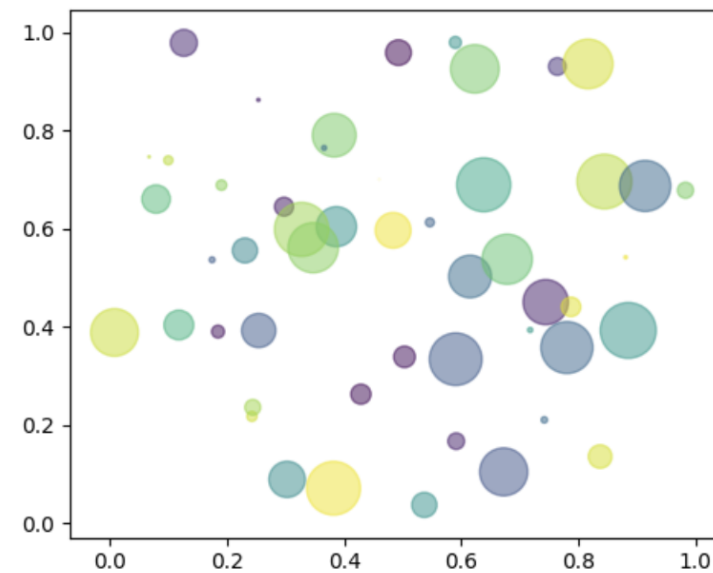
# Gallery of examples



# Example page with code

shapes\_and\_collections example code: scatter\_demo.py

([Source code](#), [png](#), [pdf](#))

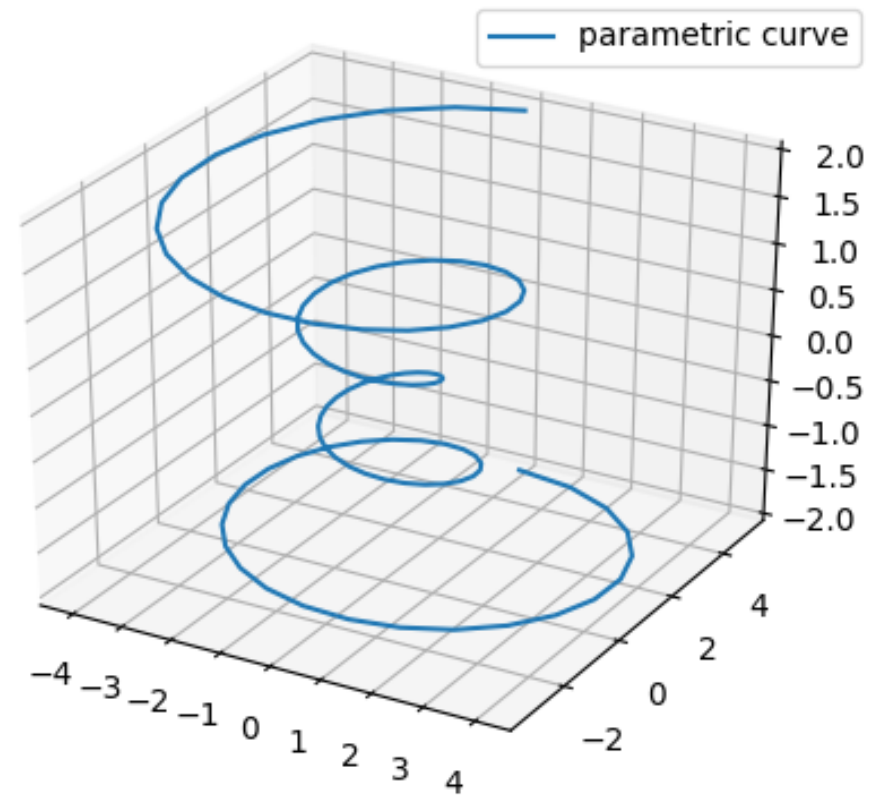


```
"""
Simple demo of a scatter plot.
"""
import numpy as np
import matplotlib.pyplot as plt

N = 50
x = np.random.rand(N)
y = np.random.rand(N)
colors = np.random.rand(N)
area = np.pi * (15 * np.random.rand(N))**2 # 0 to 15 point radii

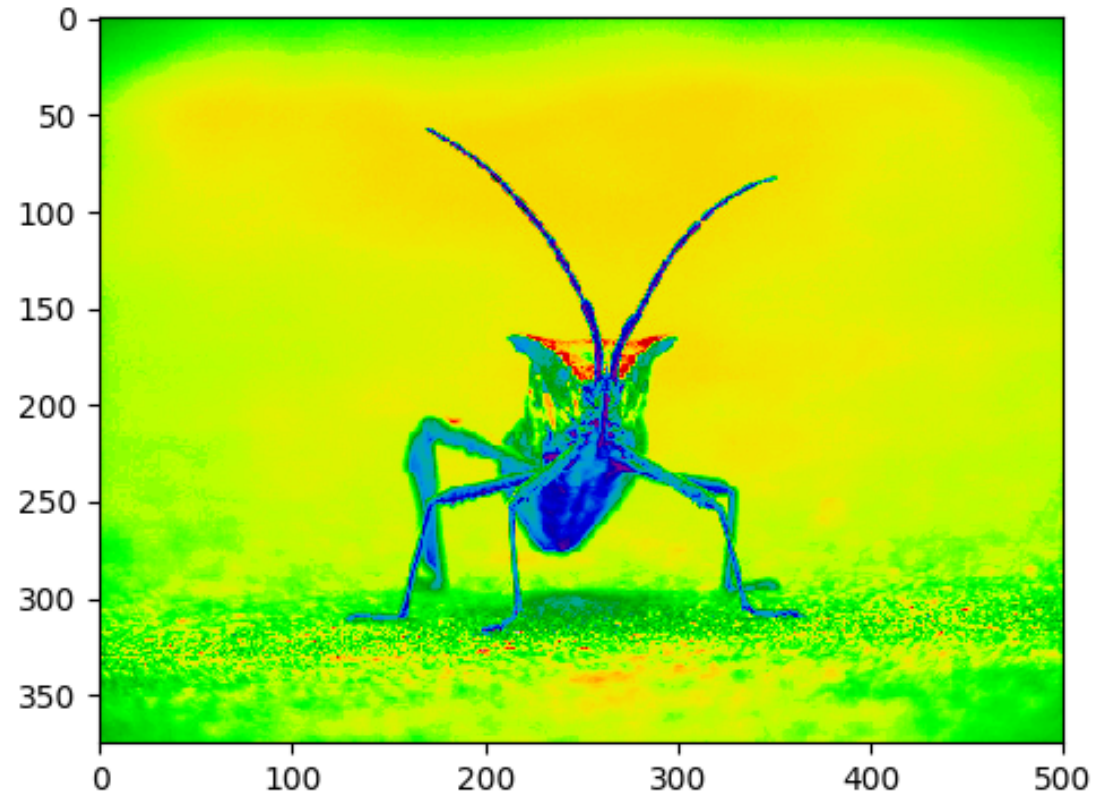
plt.scatter(x, y, s=area, c=colors, alpha=0.5)
plt.show()
```

# Plotting data in 3D



[https://matplotlib.org/mpl\\_toolkits/mplot3d/tutorial.html](https://matplotlib.org/mpl_toolkits/mplot3d/tutorial.html)

# Visualizing images with pseudo-color



[https://matplotlib.org/users/image\\_tutorial.html](https://matplotlib.org/users/image_tutorial.html)

# Animations

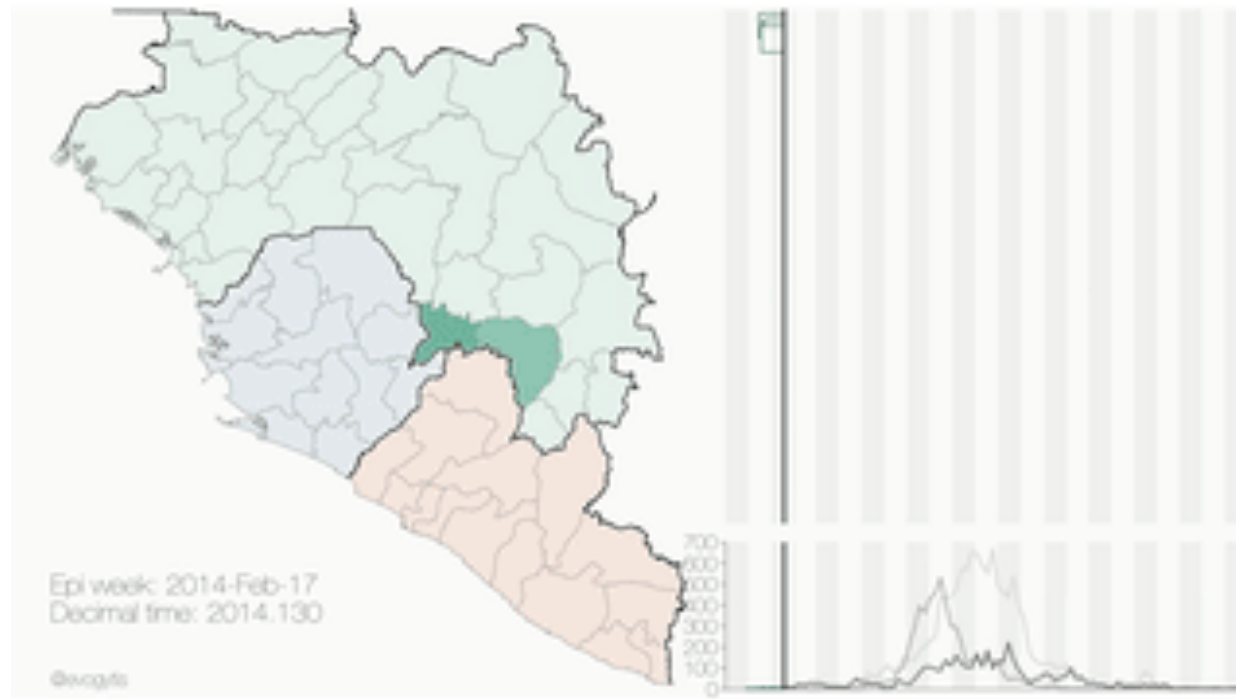


Image credit: **Gytis Dudas** and **Andrew Rambaut**

[https://matplotlib.org/api/animation\\_api.html](https://matplotlib.org/api/animation_api.html)

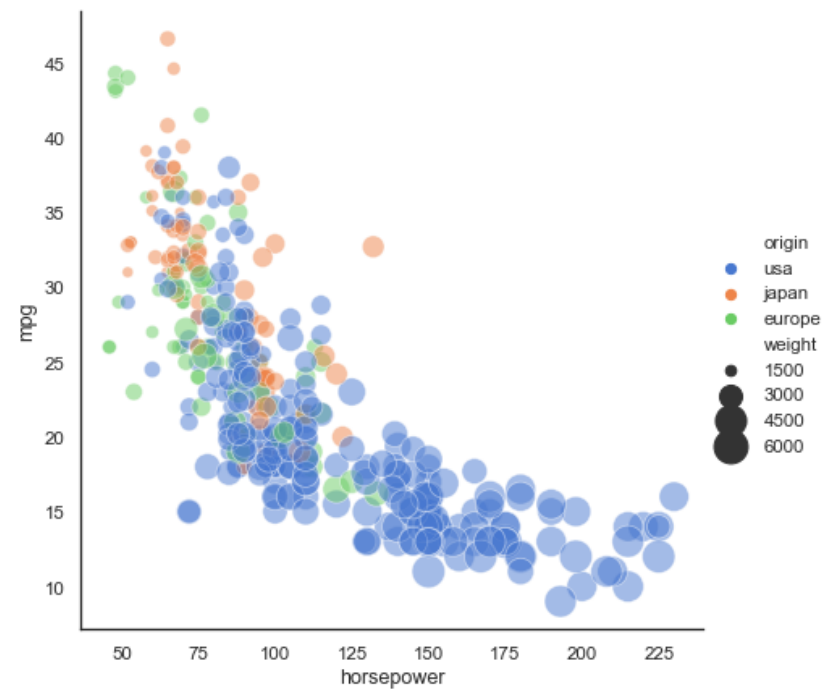
# Using Matplotlib for geospatial data



<https://scitools.org.uk/cartopy/docs/latest/>

# Pandas + Matplotlib = Seaborn

```
seaborn.relplot(x="horsepower", y="mpg", hue="origin", size="weight",  
               sizes=(40, 400), alpha=.5, palette="muted",  
               height=6, data=mpg)
```





# Seaborn example gallery

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

**Good luck  
visualizing your  
data!**

**INTRODUCTION TO DATA VISUALIZATION WITH MATPLOTLIB**