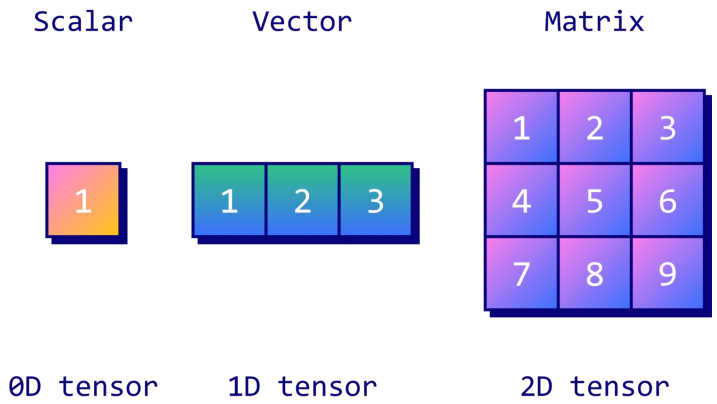
Python Tools for Machine Learning and Data Analysis (2)

Common Libraries for Data Processing

- numpy
- pandas
- matplotlib
- seaborn

numpy

- A unified platform for data processing
- Its data structure, numpy array (np.array), is very useful for processing multi-dimensional data (tensors) we see in machine learning



pandas

- A unified platform for processing tables of data
- The main structure it processes is the DataFrame (pd.DataFrame)

Arrays

- For example, the Python list [1, 2, 3, 4] can be thought of as a 4-dimensional vector (4-dimensional 1D array, the terms can be confusing!)
- This corresponds to one data point with four features

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.1	3.5	1.4	0.2

Data source: https://www.kaggle.com/datasets/saurabh00007/iriscsv

Arrays

- [[1, 2, 3, 4],
  [5, 6, 7, 8],
  [9, 10, 11, 12]] (a nested Python list) can be thought of as a 3x4 matrix (2D array)
- This corresponds to 3 data points with 4 features each

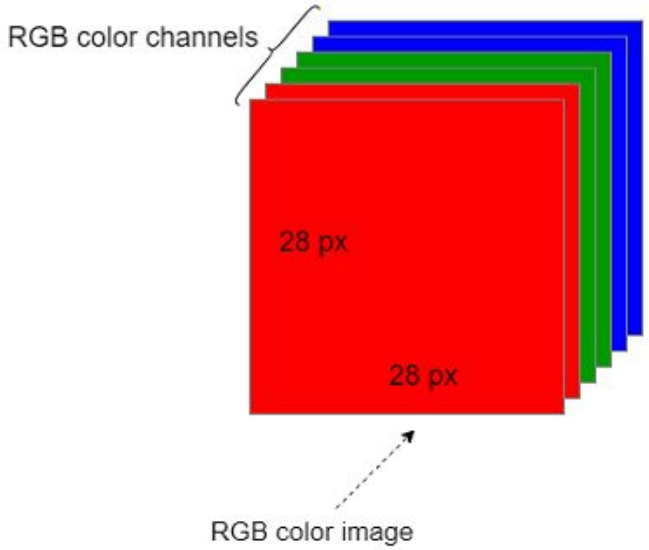
:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2

- Data source: https://www.kaggle.com/datasets/saurabh00007/iriscsv

Arrays

- Many things that we process in machine learning (e.g., images) can be implemented as multi-dimensional arrays (a.k.a. tensors which will be discussed later)
- e.g., an image can be represented as a 3D array



Source: https://towardsdatascience.com/rgb-color-image-compression-using-principal-component-analysis-fce3f48dfdd0/

Creating numpy Arrays

- Convert a Python list into an np.array
- e.g., t = np.array([1, 2, 3, 4]) # 4-dimensional vector
- Fill in values and specify the shape
- e.g., t = np.zeros([3, 3]) # 3x3 matrix, np.array([[0, 0, 0], [0, 0, 0], [0, 0, 0]])

Shapes

- Details the number of elements for each dimension of the arrays
- Shapes of arrays can be expressed as arrays
- e.g., the shape of a 3-dimensional vector is [3], the shape of a 3x3 matrix is [3, 3]
- The shape of an array is described in its .shape attribute
- e.g., np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]]).shape -> (3, 3

Shapes

- Try out the numpy array functions on https://www.kaggle.com/code/carsoncheng/numpy-array-processing/edit

pandas

- Kaggle course: https://www.kaggle.com/learn/pandas

pandas DataFrame

- Create a pandas DataFrame: from a dictionary (e.g., pd.DataFrame({‘a’: [1, 2, 3], ‘b’: [4, 5, 6]})) or from a file (pd.read_csv(“your_file.csv”))

In [2]:
```
pd.DataFrame({'Yes': [50, 21], 'No': [131, 2]})
```

Out[2]:

	Yes	No
0	50	131
1	21	2

Image source: https://www.kaggle.com/code/residentmario/creating-reading-and-writing

pandas DataFrame

- View the first few rows of the DataFrame df: df.head()

```
In [9]: wine_reviews.head()
```

Out[9]:

	Unnamed: 0	country	description	designation	points	price	province	region_1	region_2	taster_name	taster_twitter_handle
0	0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	NaN	Kerin O'Keefe	@kerinokeefe
1	1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	NaN	NaN	Roger Voss	@vossroger
2	2	US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Willamette Valley	Willamette Valley	Paul Gregutt	@paulgwine
3	3	US	Pineapple rind, lemon pith and orange	Reserve Late Harvest	87	13.0	Michigan	Lake Michigan Shore	NaN	Alexander Peartree	NaN

Image source: https://www.kaggle.com/code/residentmario/creating-reading-and-writing

pandas DataFrame

- Extracting certain rows of the DataFrame

```
In [16]: reviews.loc[reviews.country == 'Italy']
```

Out[16]:

	country	description	designation	points	price	province	region_1	region_2	taster_name	taster_twitter_handle	title
0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	NaN	Kerin O'Keefe	@kerinokeefe	Nicosia 2013 Vulkà Bianco (Etna)
6	Italy	Here's a bright, informal red that opens with ...	Belsito	87	16.0	Sicily & Sardinia	Vittoria	NaN	Kerin O'Keefe	@kerinokeefe	Terre di Giurfo 2013 Belsito Frappato (Vittoria)
...	...	...	...	...	...	...	...	...	...	...	...
129961	Italy	Intense aromas of wild cherry, baking spice, t...	NaN	90	30.0	Sicily & Sardinia	Sicilia	NaN	Kerin O'Keefe	@kerinokeefe	COS 2013 Frappato (Sicilia)
129962	Italy	Blackberry, cassis, grilled herb and toasted a...	Sàgana Tenuta San Giacomo	90	40.0	Sicily & Sardinia	Sicilia	NaN	Kerin O'Keefe	@kerinokeefe	Cusumano 2012 Sàgana Tenuta San Giacomo Nero d...

Image source: https://www.kaggle.com/code/residentmario/indexing-selecting-assigning

Data Visualization

- Before applying AI / machine learning or any other techniques, you should first get an understanding of the data using exploratory data analysis (EDA)
- Different properties of the data warrant different processing and modeling steps
- It is a good idea to visualize the data before you start thinking about approaches to modeling

Data Visualization

- https://www.kaggle.com/learn/data-visualization

Data Visualization

- Main libraries: matplotlib and seaborn
- matplotlib for figure construction and graphing; seaborn for graphing
- import matplotlib.pyplot as plt
- import seaborn as sns

In [1]:

```
import pandas as pd
pd.plotting.register_matplotlib_converters()
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
print("Setup Complete")
```

```
Setup Complete
```

Image source: https://www.kaggle.com/code/alexisbcook/hello-seaborn

Useful Plots

- Line charts

```
In [5]: # Line chart showing daily global streams of each song
        sns.lineplot(data=spotify_data)

Out[5]: <AxesSubplot:xlabel='Date'>
```

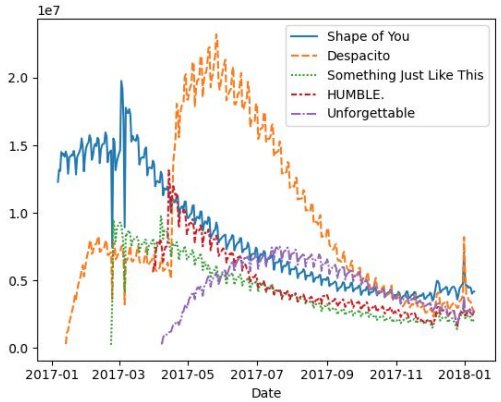


Image source: https://www.kaggle.com/code/alexisbcook/line-charts

Useful Plots

- Heatmaps

```
In [5]:
# Set the width and height of the figure
plt.figure(figsize=(14,7))

# Add title
plt.title("Average Arrival Delay for Each Airline, by Month")

# Heatmap showing average arrival delay for each airline by month
sns.heatmap(data=flight_data, annot=True)

# Add label for horizontal axis
plt.xlabel("Airline")

Out[5]:
Text(0.5, 47.7222222222222, 'Airline')
```

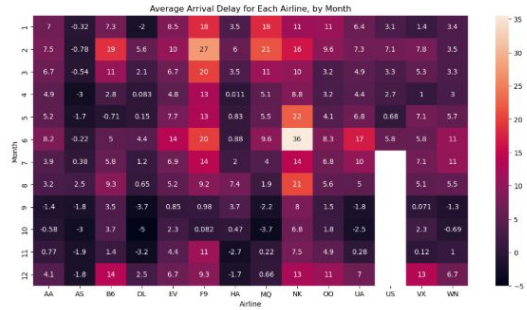


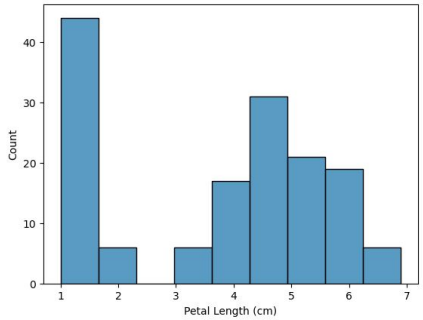
Image source: https://www.kaggle.com/code/alexisbcook/bar-charts-and-heatmaps

Useful Plots

- Histograms and distribution plots

In [3]:
```
# Histogram
sns.histplot(iris_data['Petal Length (cm)'])
```

Out[3]:
```
<AxesSubplot:xlabel='Petal Length (cm)', ylabel='Count'>
```



In [4]:
```
# KDE plot
sns.kdeplot(data=iris_data['Petal Length (cm)'], shade=True)
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:2: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.
```

Out[4]:
```
<AxesSubplot:xlabel='Petal Length (cm)', ylabel='Density'>
```

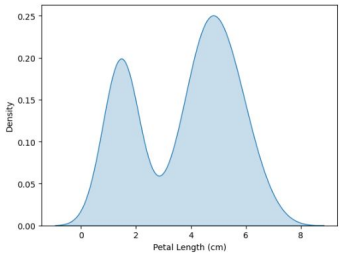


Image source: https://www.kaggle.com/code/alexisbcook/distributions

Useful Plots

- Scatterplots

```
In [6]: sns.scatterplot(x=insurance_data['bmi'], y=insurance_data['charges'], hue=insurance_data['smoke
r'])

Out[6]: <AxesSubplot:xlabel='bmi', ylabel='charges'>
```

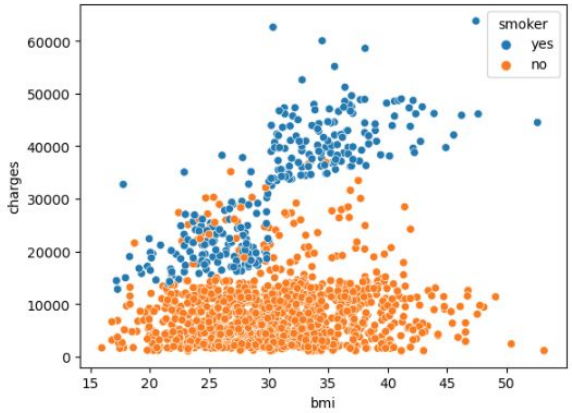


Image source: https://www.kaggle.com/code/alexisbcook/scatter-plots