# Part 2 - Data Visualization

## Seaborn Information

seaborn practice datasets – sns.get\_dataset\_names()

df = sns.load\_dataset(‘dowjones’)

## Seaborn

Import seaborn as sns

plt.figure(figsize=(8,8), dpi=150) # provides a bit better chart

### Line Plot

sns.lineplot(data=pokemon, x=”HP”, y=”Attack”)

### Scatter Plot

sns.scatterplot(data=pokemon, x=”HP”, y=”Attack”, (optional) hue=”Type”, col=”Type”)

### Relationship Plot

sns.relplot(data=pokemon, x=”HP”, y=”Attack”, (optional) hue=”Type”, col=”Type”, col\_wrap=3)

### Regression Plot – Plot data and a linear regression model fit.

sns.regplot(data=pokemon, x=”HP”, y=”Attack”)

### Pair Plot – relationships of all numeric features

sns.pairplot(pokemon)

sns.scatterplot(x='age', y='physical\_score', data=df, hue='test\_result', alpha=0.8)

### Histogram Plot

sns.histplot(data=pokemon, x="Attack") (optional bins=10 or other number)

### Kernel Density Estimate Plot

sns.kdeplot(data=pokemon, x="Attack")

### Combined Histogram and KDE Plot

sns.histplot(data=pokemon, x="Attack", bins=8, kde=True)

### Distribution Plot

sns.displot(data=pokemon, x="Attack", bins=10, col="Type", col\_wrap=3)

### Categorical Plots

sns.stripplot(data=pokemon, x='Type', y='Attack')

sns.catplot(kind='strip', data=pokemon, x='Type', y='Attack', aspect=2)

### Categorical Box Plot

sns.catplot(kind='box', data=pokemon, x='Type', y='Attack', aspect=2)

### Categorical Violin Plot

sns.catplot(kind='violin', data=pokemon, x='Type', y='Attack', aspect=2)

### Categorical Bar Plot

sns.catplot(kind='bar', data=pokemon, x='Type', y='Attack', aspect=2)

### Categorical Count Plot

sns.catplot(kind='count', data=pokemon, x='Type', aspect=2)

### Count Plot

sns.countplot(data=df, x='test\_result')

### Box Plot

sns.boxplot(x='test\_result', y='physical\_score', data=df)

## Matplotlib

import matplotlib as plt

Pandas include a built-in implementation of Matplotlib.

df.plot(kind=’line’) # other types are available

**Example Line Plot**

years = list(map(str, range(1980, 2014)))

df\_canada.loc(‘country\_wanted’, years).plot(kind=’line’)

plt.title(‘Plot Title’)

plt.ylabel(‘ylabel’)

plt.xlabel(‘xlabel’)

plt.show()

**Example Area Plot**

df\_canada.sort\_values([‘Total’], ascending=False, inplace=True)

df.top5 = df\_canada.head()

df.top5 = df\_top5[years].transpose

df\_top5.plot(kind=’area’)

plt.title(‘Plot Title’)

plt.ylabel(‘ylabel’)

plt.xlabel(‘xlabel’)

plt.show()

**Example Histogram**

count, bin\_edges = np.histogram(df\_canada[‘2013’])

df\_canada[‘2013’].plot(kind=’hist’, xticks=bin\_edges)

plt.title(‘Plot Title’)

plt.ylabel(‘ylabel’)

plt.xlabel(‘xlabel’)

plt.show()

**Example Bar Chart**

years = list(map(str, range(1980, 2014)))

df\_Iceland = df.canada.loc[‘Iceland’, years]

c = [‘colors’, ‘colors’] # list of colors equal to the number of ticks

df.Iceland.plot(kind=’bar’, color = c, edge\_color=’black’) # you can also use ‘barh’

plt.title(‘Plot Title’)

plt.ylabel(‘ylabel’)

plt.xlabel(‘xlabel’)

plt.show()

**Example Pie Chart**

df\_continents = df\_canada.groupby(‘Continent’, axis=0).sum()

df\_continents[‘total’].plot(kind=pie)

plt.title(‘Plot Title’)

plt.show()

**Example Scatter Plot**

df\_total = df\_canada[‘year’, ‘total’]

df\_total.plot(kind=’scatter’, x=’year’, y=’total’)

plt.title(‘Plot Title’)

plt.ylabel(‘ylabel’)

plt.xlabel(‘xlabel’)

plt.show()

**Example Waffle Chart**

#install pywaffle

!pip install pywaffle

#import Waffle from pywaffle

from pywaffle import Waffle

#Set up the Waffle chart figure

fig = plt.figure(FigureClass = Waffle,

rows = 20, columns = 30, #pass the number of rows and columns for the waffle

values = df\_dsn['Total'], #pass the data to be used for display

cmap\_name = 'tab20', #color scheme

legend = {'labels': [f"{k} ({v})" for k, v in zip(df\_dsn.index.values,df\_dsn.Total)],

'loc': 'lower left', 'bbox\_to\_anchor':(0,-0.1),'ncol': 3}

#notice the use of list comprehension for creating labels

#from index and total of the dataset

)

#Display the waffle chart

plt.show()

# Multiple Histograms

Create multiple histograms from features from the same dataframe.

columns=[‘price’, ‘speed’, ‘ram’] # target features

data = computers # # target data frame

fig, ax = plt.subplots(len(columns), layout=’tight’)

for col, ax in zip(columns, ax.flat):

sns.histplot(data, x=col, ax=ax)

ax.set\_title(f”Histogram of {col}”)

**# For multiple Box Plots**

fig, ax = plt.subplots(3)

for col, ax in zip(columns, ax.flat):

sns.boxplot(data=data, x=col, ax =ax)

## Feature Correlations - Heatmap

## Imports

import seaborn as sns

import matplotlib.pyplot as plt

import numpy as np

df = sns.load\_dataset('mpg')

df.head()

# Drop non-numeric features

df.drop('origin', axis=1, inplace=True)

df.drop('name', axis=1, inplace=True)

df.head()

df.corr()

my\_corr\_mat = df.corr()

## Heatmap Example

sns.heatmap(my\_corr\_mat, cmap='plasma')

plt.figure(figsize=(12,6), dpi=250)

sns.heatmap(my\_corr\_mat, cmap='plasma', vmin=0.5, vmax=1);

plt.figure(figsize=(12,6), dpi=250)

sns.heatmap(my\_corr\_mat, cmap='plasma', vmin=-1, vmax=1);

plt.figure(figsize=(12,6), dpi=250)

sns.heatmap(my\_corr\_mat, cmap='plasma', center=0.5, vmin=0.5, vmax=1);

plt.figure(figsize=(12,6), dpi=250)

sns.heatmap(my\_corr\_mat, cmap='plasma', center=0, vmin=-0.25, vmax=0.25, annot=True);

plt.figure(figsize=(12,6), dpi=250)

plt.figure(figsize=(12,6), dpi=250)

sns.heatmap(my\_corr\_mat, annot=True, cmap='magma', vmin=-1, vmax=1);

plt.figure(figsize=(12,6), dpi=250)

sns.heatmap(my\_corr\_mat, annot=True, cmap='magma', vmin=-1, vmax=1, linewidths=2, linecolor='red');

plt.figure(figsize=(12,6), dpi=250)

sns.heatmap(my\_corr\_mat, annot=True, cmap='magma', vmin=-1, vmax=1, square=True);

plt.figure(figsize=(12,6), dpi=250)

sns.heatmap(my\_corr\_mat, annot=True, cmap='magma', vmin=-1, vmax=1, square=True, mask=np.triu(my\_corr\_mat));

plt.figure(figsize=(12,6), dpi=250)

sns.heatmap(my\_corr\_mat, annot=True, cmap='mako', vmin=-1, vmax=1, square=True);

plt.figure(figsize=(12,6), dpi=250)

sns.heatmap(my\_corr\_mat, annot=True, cmap='vlag', vmin=-1, vmax=1, square=True);

plt.figure(figsize=(12,6), dpi=250)

sns.heatmap(my\_corr\_mat, annot=True, cmap='icefire', vmin=-1, vmax=1, square=True);

plt.figure(figsize=(12,6), dpi=250)

sns.heatmap(my\_corr\_mat, annot=True, cmap='icefire', fmt=".2f", vmin=-1, vmax=1, square=True, mask = np.triu(my\_corr\_mat));

plt.figure(figsize=(12,6), dpi=250)

sns.heatmap(my\_corr\_mat, annot=True, cmap='coolwarm', fmt=".2f", vmin=-1, vmax=1, square=True);

plt.figure(figsize=(12,6), dpi=250)

sns.heatmap(my\_corr\_mat, annot=True, cmap='Spectral', fmt=".2f", vmin=-1, vmax=1, square=True);

plt.figure(figsize=(12,6), dpi=250)

sns.heatmap(my\_corr\_mat, annot=True, cmap='Spectral', fmt=".2f", vmin=-1, vmax=1, square=True, mask = np.triu(my\_corr\_mat));