EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis

- Exploratory Data Analysis, or EDA is a way to analyze data sets, often using data visualization methods, to summarize the main characteristics of the data.
- EDA helps in better understanding the different features of the data, the relationship between them, and in determining the statistical techniques appropriate for the data set.
- matplotlib and seaborn libraries are pretty good for EDA.

Univariate Analysis (1/7)

- A dataset may contain one or more features/variables/columns.
- Univariate Analysis provides summary statistics only on one variable.
- Univariate Analysis only describes the data and helps identify any patterns in the data.

Categorical vs Continuous Data

- There are two types of data;
  - Categorical/Discrete, e.g., spam vs no spam, male vs female
  - Continuous, e.g., Age of population
- EDA is performed differently on the two types of data.

Univariate Analysis (2/7)

- Consider the following dataframe containing 5 features containing information about iris plants.
- The last column is categorical, while the rest are continuous.

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Virginica
146	6.3	2.5	5.0	1.9	Virginica
147	6.5	3.0	5.2	2.0	Virginica
148	6.2	3.4	5.4	2.3	Virginica
149	5.9	3.0	5.1	1.8	Virginica

150 rows × 5 columns

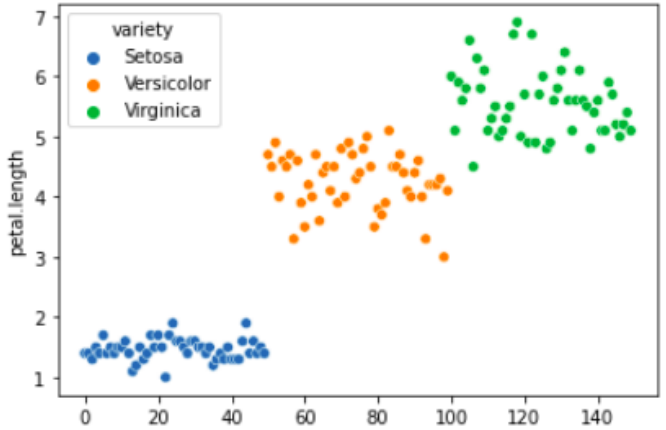
Univariate Analysis – Continuous Data (3/7)

Scatter Plot

- We can use a number of graphs to perform EDA on continuous data.
- In the given figure, we use the .scatterplot() function of the seaborn library to plot the 'petal.length' column of the dataframe.
- The hue parameter assigns different color to the data points based on the category they belong to in the column specified in the hue parameter.

```
x_axis = df.index
y_axis = df['petal.length']
sns.scatterplot(x=x_axis, y=y_axis, hue=df['variety'])
```

```
<AxesSubplot:ylabel='petal.length'>
```



Meta Brains

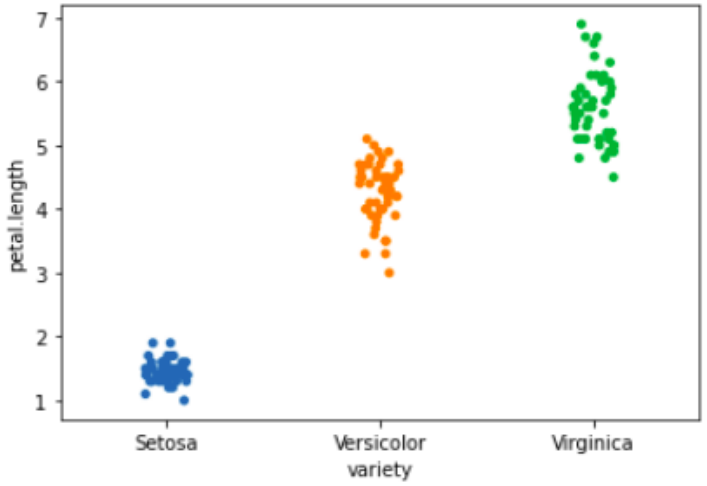
Univariate Analysis – Continuous Data (4/7)

Strip Plot

- Strip plots are also a good way to analyze the distribution of the variables for each category.
- In the given figure, we have plotted ‘petal.length’ column for each category in the ‘variety’ column.
- On the Y-axis, we have the distribution of each category, and on the X-axis we have the categories.

```
sns.stripplot(x=df['variety'], y=df['petal.length'])
```

```
<AxesSubplot:xlabel='variety', ylabel='petal.length'>
```



Meta Brains

Univariate Analysis – Continuous Data (5/7)

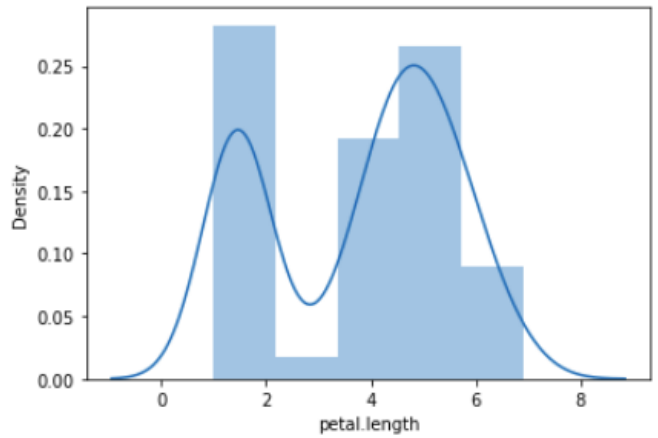
Distribution Plot

- To find the distribution of a variable/feature/column, use the .distplot() function of the seaborn library.
- In this figure, we plot the distribution of the ‘petal.length’ column.

```
sns.distplot(df['petal.length'])
```

```
/home/waqar/anaconda3/lib/python3.8/site-packages/seaborn/distributions.py:2551: FutureWar
ning: `distplot` is a deprecated function and will be removed in a future version. Please
adapt your code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
<AxesSubplot:xlabel='petal.length', ylabel='Density'>
```



Meta Brains

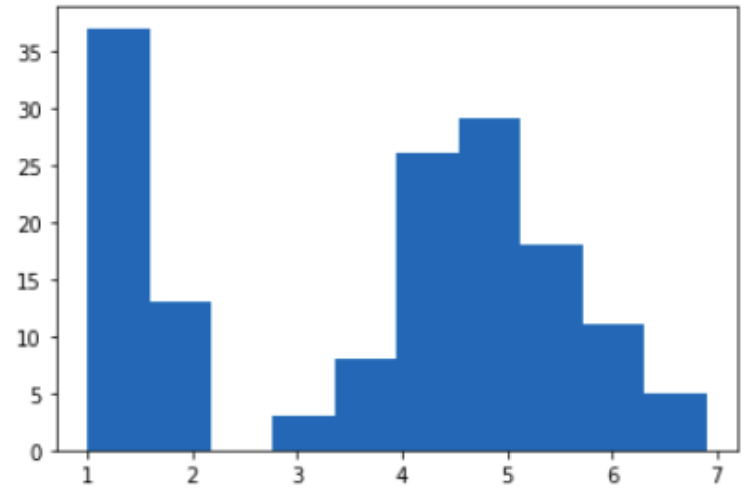
Univariate Analysis – Continuous Data (6/7)

Histograms

- To analyze the frequency of values, we can plot a histogram using the .hist() method of the matplotlib library.
- In this figure, we plot the frequency of values in the ‘petal.length’ column.

```
plt.hist(df['petal.length'])
```

```
(array([37., 13.,  0.,  3.,  8., 26., 29., 18., 11.,  5.]),
 array([1.  , 1.59, 2.18, 2.77, 3.36, 3.95, 4.54, 5.13, 5.72, 6.31, 6.9 ]),
 <BarContainer object of 10 artists>)
```

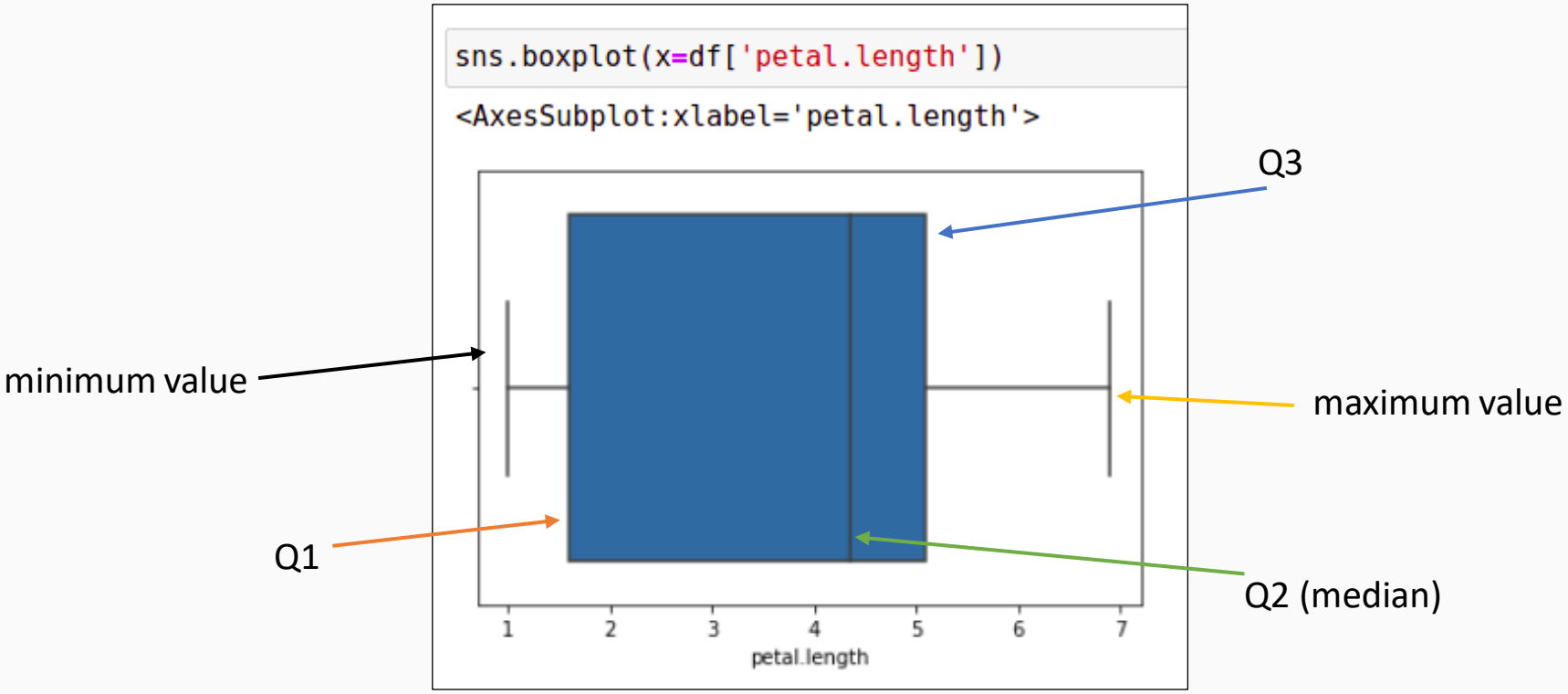


Meta Brains

Univariate Analysis – Continuous Data (7/7)

Box Plot

- A box plot provides great insights into the data using 5-number summary; minimum value, first quartile, second quartile, third quartile, and maximum value.
- We can use the .boxplot() function of either matplotlib or seaborn.



Meta Brains

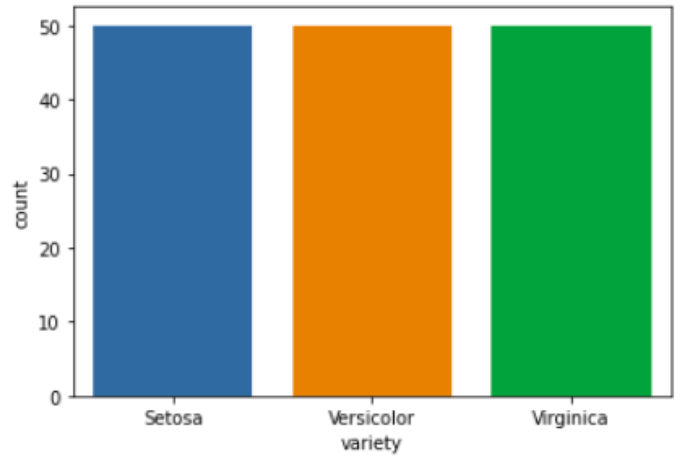
Univariate Analysis – Categorical Data (1/2)

Count Plot

- We can use count plots to perform EDA on categorical data.
- The .countplot() function of the seaborn library plots the total count of each value as a bar chart.
- In the given figure, we see that we have equal instances of each category in the dataframe.

```
sns.countplot(x=df['variety'])
```

```
<AxesSubplot:xlabel='variety', ylabel='count'>
```



Meta Brains

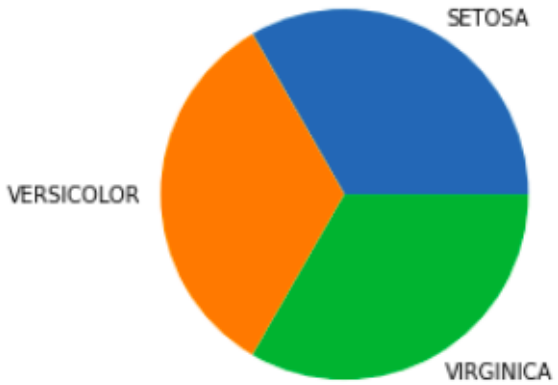
Univariate Analysis – Categorical Data (2/2)

Pie Chart

- We can also visualize the proportion of each category using a pie chart as shown in the figure.

```
_labels = ['SETOSA','VERSICOLOR','VIRGINICA']
plt.pie(df['variety'].value_counts(), labels = _labels)
```

```
([<matplotlib.patches.Wedge at 0x7f18c505e5b0>,
  <matplotlib.patches.Wedge at 0x7f18c505ea90>,
  <matplotlib.patches.Wedge at 0x7f18c505ef10>],
 [Text(0.5499999702695115, 0.9526279613277875, 'SETOSA'),
  Text(-1.0999999999999954, -1.0298943258065002e-07, 'VERSICOLOR'),
  Text(0.5500001486524352, -0.9526278583383436, 'VIRGINICA')])
```



Meta Brains

Bivariate Analysis – Continuous & Continuous (1/4)

- Bivariate Analysis is used to study the relationship between exactly two variables/features/columns of the data set.
- Consider the following dataframe.

```
df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

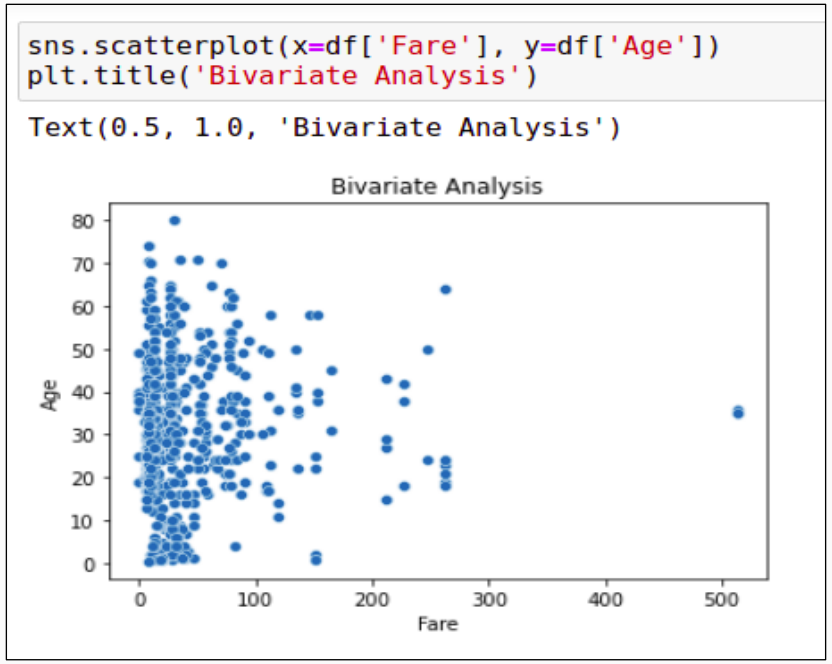
Bivariate Analysis – Continuous & Continuous (2/4)

Scatter Plot

- If the two variables in question are both continuous, we can plot a scatter plot between them to get an idea of their distribution.
- In the given figure, we plot the ‘Age’ column of the dataframe against the ‘Fare’ column, both of which are continuous.
- It is evident that the two features are independent of each other.

```
sns.scatterplot(x=df['Fare'], y=df['Age'])
plt.title('Bivariate Analysis')
```

```
Text(0.5, 1.0, 'Bivariate Analysis')
```



Meta Brains

Bivariate Analysis – Continuous & Continuous (3/4)

Correlation

- We can also find the correlation between two continuous features.
- If the correlation is high, the two features are significantly related.
- Use the .corr() function to compute correlation between two features.

```
df[['Fare', 'Age']].corr()
```

	Fare	Age
Fare	1.000000	0.096067
Age	0.096067	1.000000

Meta Brains

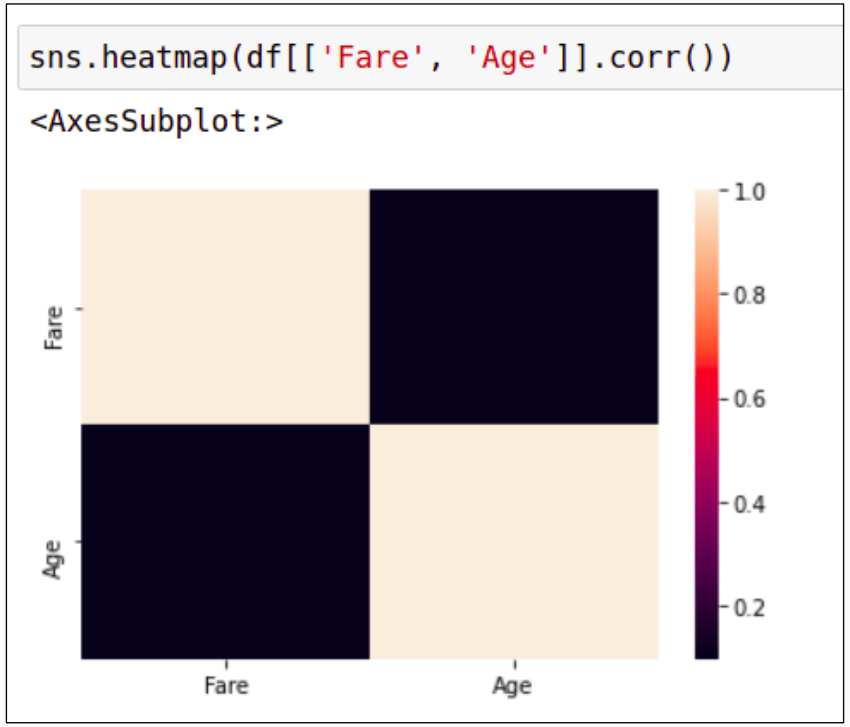
Bivariate Analysis – Continuous & Continuous (4/4)

Heatmp

- We can also use a heatmap to graphically visualize the correlation between something.
- Use the .heatmap() method of the seaborn library to create a heatmap.
- Provide the correlation dataframe that we created in the previous slide as an argument to the .heatmap()

```
sns.heatmap(df[['Fare', 'Age']].corr())
```

```
<AxesSubplot:>
```



Meta Brains

Bivariate Analysis – Categorical & Categorical (1/3)

- Consider the following dataframe from the previous slide.
- We will see how we can use bar plot to identify any relation between two categorical variables, namely ‘Pclass’ and ‘Survived’.

```
df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

Meta Brains

Bivariate Analysis – Categorical & Categorical (2/3)

- We first select the two columns that we are interested in, namely ‘Pclass’ and ‘Survived’.
- We then group this new dataframe based on ‘Pclass’ column and apply the .sum() function to find total number of survivors from each category in the Pclass.
- The output of this part is shown below.

```
survived_ratio = df[['Pclass', 'Survived']].groupby('Pclass').sum()
survived_ratio
```

Pclass	Survived
1	136
2	87
3	119

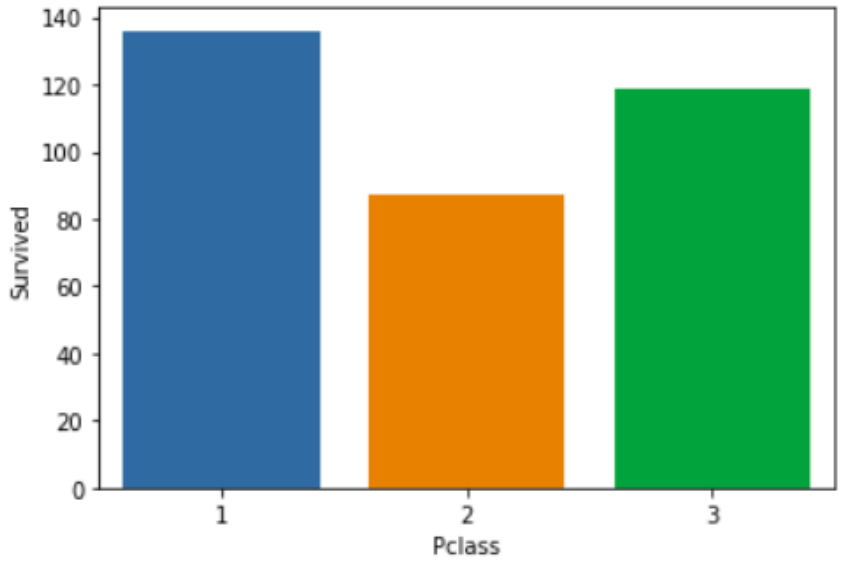
Meta Brains

Bivariate Analysis – Categorical & Categorical (3/3)

- We then plot a bar plot, with ‘Pclass’ on the X-axis.
- As evident, more passengers in ‘Pclass’ 1 were able to survive than those from ‘Pclass’ 2 or 3.

```
sns.barplot(x=survived_ratio.index, y=survived_ratio['Survived'])
```

```
<AxesSubplot:xlabel='Pclass', ylabel='Survived'>
```



Meta Brains

Bivariate Analysis – Continuous & Categorical (1/3)

- Sometimes, we want to find out the relationship between a continuous and a categorical variable.
- Consider the following dataframe from the previous slide.

```
df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

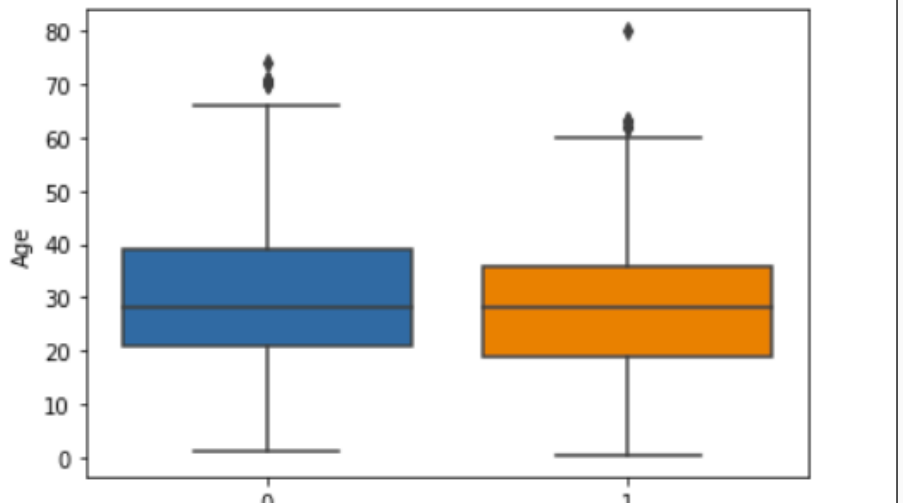
Bivariate Analysis – Continuous & Categorical (2/3)

Box Plot

- We can use a box plot to perform EDA on continuous and categorical data.
- In the given figure, we plot 'Age' (continuous feature) against 'Survived' (categorical feature).
- The plot suggests that the younger people had a greater chance of survival.

```
sns.boxplot(x=df['Survived'], y=df['Age'])
```

```
<AxesSubplot:xlabel='Survived', ylabel='Age'>
```



Meta Brains

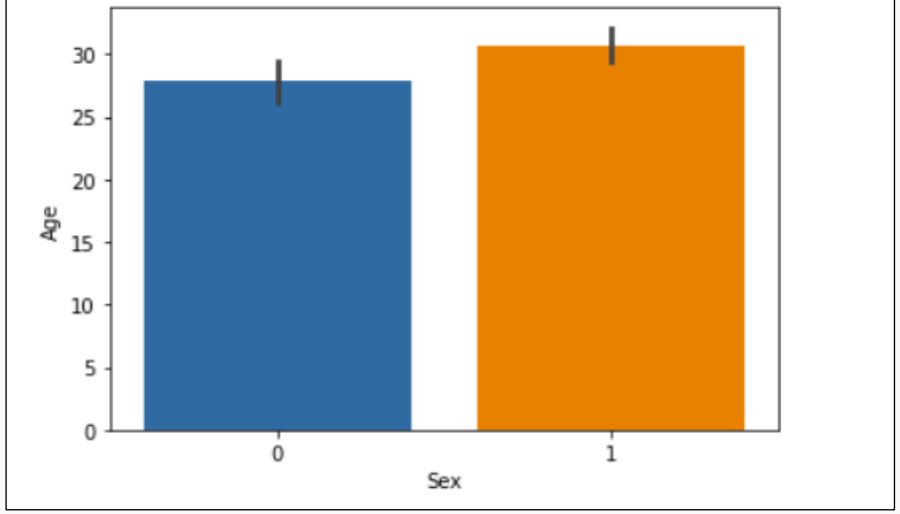
Bivariate Analysis – Continuous & Categorical (3/3)

Bar Plot

- Bar plots can also be used for bivariate analysis of a continuous and a categorical feature.
- In the given figure we plot 'Age' (continuous feature) against 'Sex' (categorical feature). 1 in the 'Sex' column represents male and 0 represents 'female' passengers.
- The bar plot suggests that elderly passengers were mostly male.

```
sns.barplot(x=df['Sex'], y=df['Age'])
```

```
<AxesSubplot:xlabel='Sex', ylabel='Age'>
```



Meta Brains

Detecting Outliers

- Outliers/anomalies in the data are the observations that do not fit into the standard pattern of the data.
- In chapetr 4, we discussed major techniques to detect outliers/anomalies e.g.;
  - Median-based anomaly detection
  - Mean-based anomaly detection
  - Z-score-based anomaly detection
  - IQR-based anomaly detection
- In this chapter, we will learn different ways to treat such outliers/anomalies in the data.

Outliers Treatment (1/3)

Trimming Outliers

- One naïve way of treating outliers is to remove them from the data. However, this approach is not very good.
- Outliers can be removed from the data in several ways.
- Consider the given Series. It is safe to say that the value 150 is an outlier in the data.

```
0      1
1      2
2      3
3      6
4      7
5      8
6    150
dtype: int64
```

Meta Brains

Outliers Treatment (2/3)

Trimming Outliers

- We delete the rows of the Series for which the absolute value of the Z-score is bigger than 1.5, which means that the values lie outside of 1.5 standard deviations of the data.

```
x = pd.Series([1, 2, 3, 6, 7, 8, 150])
mean = x.mean()
std = x.std()
z_scores = abs((x-mean)/std)
z_scores
```

```
0    0.441104
1    0.422941
2    0.404778
3    0.350288
4    0.332125
5    0.313962
6    2.265198
dtype: float64
```

```
outliers_removed = x[z_scores <= 1.5]
outliers_removed
```

```
0    1
1    2
2    3
3    6
4    7
5    8
dtype: int64
```

Meta Brains

Outliers Treatment (3/3)

Mean/Median Imputation

- We can also replace outliers with either mean or median.
- In the given figure, we detect outliers in the data using Z-score and replace them with the median value of the Series.

```
x = pd.Series([1, 2, 3, 6, 7, 8, 150])
mean = x.mean()
std = x.std()
median = np.median(x)
z_scores = abs((x-mean)/std)
median
```

```
6.0
```

```
x[z_scores > 1.5] = median
x
```

```
0    1
1    2
2    3
3    6
4    7
5    8
6    6
dtype: int64
```

Meta Brains

Categorical Variable Transformation (1/3)

- We discussed about numerical variable transformation in chapter 4 using normalization and standardization.
- In this chapter we will discuss categorical variable transformation.
- There are several ways to transform categorical variable to make them more meaningful for machines, we will discuss only some of them.
- Consider the following dataframe. We will transorm the 'Sex' variable.

	Name	Sex
0	Braund, Mr. Owen Harris	male
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female
2	Heikkinen, Miss. Laina	female
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female
4	Allen, Mr. William Henry	male

Meta Brains

Categorical Variable Transformation (2/3)

Label Encoding

- In label encoding, we replace categorical data with numbers.
- We replace male with 1 and female with 0.

```
df['Sex'].replace({'male':1, 'female':0}, inplace=True)
df
```

	Name	Sex
0	Braund, Mr. Owen Harris	1
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0
2	Heikkinen, Miss. Laina	0
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0
4	Allen, Mr. William Henry	1

Meta Brains

Categorical Variable Transformation (3/3)

Frequency Encoding

- In this encoding method, we replace each value in the categorical variable by its frequency.
- In the given figure, we replaced male and female labels in the 'Sex' column with their frequencies.

```
freq = df['Sex'].value_counts()/len(df['Sex'])
freq
```

```
female    0.6
male      0.4
Name: Sex, dtype: float64
```

```
df['Sex'].replace({'male':freq['male'], 'female':freq['female']}, inplace=True)
df
```

	Name	Sex
0	Braund, Mr. Owen Harris	0.4
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0.6
2	Heikkinen, Miss. Laina	0.6
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0.6
4	Allen, Mr. William Henry	0.4

Meta Brains

Resources

- https://www.kaggle.com/residentmario/univariate-plotting-with-pandas
- https://purnasaigudikandula.medium.com/exploratory-data-analysis-beginner-univariate-bivariate-and-multivariate-habberman-dataset-2365264b751

Meta Brains