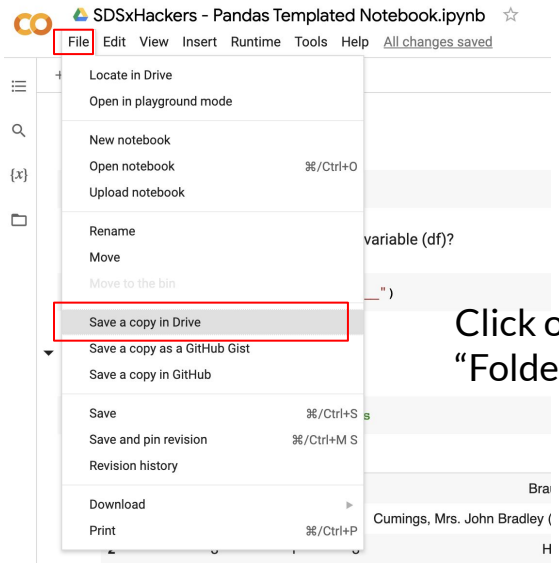
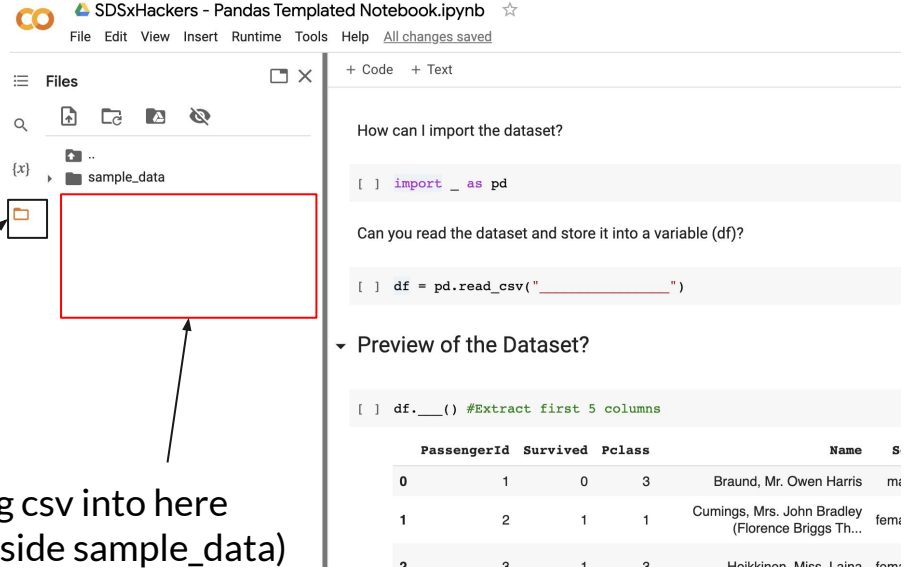
For earlybirds - Setting up of Practical Notebook

1) **Link to Practice Notebook: https://tinyurl.com/sdspandaspractice**
2) Save a copy of "**SDSxHackers - Pandas Templated Notebook.ipynb**" to your own drive.
3) Download dataset: **tinyurl.com/sdspandasdataset**
4) Drag and drop 'Titanic Train Dataset.csv' into **your Copy** of Files of Google Colab Notebook

Step 2



Step 4



Click on this "Folder"

Drag csv into here (outside sample_data)

Data Manipulation with Pandas Workshop

Dylan, Eugene, Hui Xuan

Before we start...

Link to Dataset
tinyurl.com/sdspandasdataset

Link to Pandas Example Notebook (For Reference)
https://colab.research.google.com/drive/1MToIR-vBouNwks3gsR4mmkC5hdacRuFk?usp=sharing

1. Folder containing the Dataset has been shared with participants!
2. Download csv file (Dataset) from Google Drive Link
3. Open Pandas_Example.ipynb
4. Save a copy to your drive (Files -> Save a copy in Drive)
5. Drag and drop csv folder into Google Colab Notebook (Files)
6. Test and Run the code along during Theory!

PollEv

PollEv.com/eugeneganche909

Overview of Lesson

1. Introduction to Pandas
2. Introduction to Dataset (Titanic Dataset from Kaggle)
3. Theory of Pandas Functions
4. Practical of Pandas Functions
5. Conclusion

Introduction to Pandas

What is pandas?

- Open Source Data Structures Library in Python
- Easy to perform Data Transformations and Data Cleaning with minimal code.
- Highly integrated with other popular libraries such as NumPy and Matplotlib for Data Visualisation
- Widely used by Data Analysts and Data Scientists to perform Data Manipulation

Introduction to Titanic Dataset

Titanic Dataset

The sinking of the Titanic is one of the most infamous shipwrecks in history.

Now, let's play the role of analysts who will be playing around with the dataset.

This dataset is generally used for predicting whether a passenger would survive or not. (Beyond the Scope of this Workshop)

Our goal is to learn how to manipulate data using this sample dataset.

Data Dictionary

Variable	Definition	Key
survival	Survived or not	0 = No, 1 = Yes
pclass	Ticket Class	1 = 1st, 2 = 2nd
sex	Sex	
Age	Age in Years	
sibsp	# of siblings / spouses on Titanic	
parch	# of parents / children on Titanic	
fare	Passenger fare	
cabin	Cabin Number	
embarked	Port of Embarkation	

For Reference with Slides

Notebook with example codes and outputs for reference

https://tinyurl.com/sdspandasexample

No need to run or set up anything, just for reference!

Example Notebook for your reference

Theory of Pandas Functions

Before we begin...

Introducing the Pandas DataFrame and Pandas Series objects

```
0                              Braund, Mr. Owen Harris
1    Cumings, Mrs. John Bradley (Florence Briggs Th...
2                               Heikkinen, Miss. Laina
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)
4                             Allen, Mr. William Henry
5                                    Moran, Mr. James
6                                McCarthy, Mr. Timothy J
7                          Palsson, Master. Gosta Leonard
8    Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)
9                  Nasser, Mrs. Nicholas (Adele Achem)
10                     Sandstrom, Miss. Marguerite Rut
Name: Name, dtype: object
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	NaN	C
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	G6	S

Series

- 1-dimensional array storing data for 1 column of a DataFrame
- Contains both the Index and values in the respective rows for the particular column

DataFrame

- 2-dimensional array storing data from multiple columns
- Can be thought of as a collection of Series objects

Disclaimer: Some functions are only available to either the Series or DataFrame

Before we begin...

Importing the Pandas Package + Reading in the Dataset

```
import pandas as pd

df = pd.read_csv("/content/Titanic Train Dataset.csv")
```

Viewing the DataFrame

df.shape

Shows the number of rows and columns in DataFrame in format (rows, columns)

```
df.shape #output rows x columns of df
```

```
(11, 11)
```

Output: Tuple representing the row x column dimensions of DataFRame

df.columns

Shows the names of all columns in the DataFrame

```
df.columns #output names of cols in df
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

Output: Index containing the names of all columns in DataFrame

df.info()

Shows number of non-null values and datatypes of all columns

```
df.info() #no. of non-null values and datatype
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11 entries, 0 to 10
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  11 non-null     int64  
 1   Survived     11 non-null     int64  
 2   Pclass       11 non-null     int64  
 3   Name         11 non-null     object 
 4   Sex          11 non-null     object 
 5   Age          10 non-null     float64
 6   SibSp        11 non-null     int64  
 7   Parch        11 non-null     int64  
 8   Fare         11 non-null     float64
 9   Cabin        4 non-null      object 
 10  Embarked     11 non-null     object 
dtypes: float64(2), int64(5), object(4)
memory usage: 1.1+ KB
```

Output: String with number of non-null values per column and datatype accepted for each column

df.describe()

Shows statistical summary for values in all numerical columns

```
df.describe() #statistics summary columns
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	11.000000	11.000000	11.000000	10.000000	11.000000	11.000000	11.000000
mean	6.000000	0.545455	2.363636	25.700000	0.727273	0.363636	26.082564
std	3.316625	0.522233	0.924416	16.021167	0.904534	0.674200	22.605970
min	1.000000	0.000000	1.000000	2.000000	0.000000	0.000000	7.250000
25%	3.500000	0.000000	1.500000	16.000000	0.000000	0.000000	8.254150
50%	6.000000	1.000000	3.000000	26.500000	1.000000	0.000000	16.700000
75%	8.500000	1.000000	3.000000	35.000000	1.000000	0.500000	40.966650
max	11.000000	1.000000	3.000000	54.000000	3.000000	2.000000	71.283300

Output: DataFrame with number of values (count), mean/standard deviation and min/max/quintile values

df.isnull()

Shows if the value in a specific row/column is a null value (only NAN values count)

```
df.isnull() #show if value in cell is null
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	True	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	True	False
5	False	False	False	False	False	True	False	False	False	True	False
6	False	False	False	False	False	False	False	False	False	False	False
7	False	False	False	False	False	False	False	False	False	True	False
8	False	False	False	False	False	False	False	False	False	True	False
9	False	False	False	False	False	False	False	False	False	True	False
10	False	False	False	False	False	False	False	False	False	False	False

Output: DataFrame that shows True if value is NAN and False otherwise

```
df.isnull().sum() #count of null values
```

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            1
SibSp          0
Parch          0
Fare           0
Cabin          7
Embarked       0
```

Output: Number of NAN values in each column

df.head(n)

Viewing first n rows in Pandas DataFrame

```
df.head(n=4) #preview n rows in df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S

Output: DataFrame with first n rows (n=4)

Changing Datatypes

df.astype()

Converts all values in a DataFrame into a specified Datatype (int, string, float etc)

```
df["Pclass"] #Before: Integer
```

```
0     3
1     1
2     3
3     1
4     3
5     3
6     1
7     3
8     3
9     2
10    3
Name: Pclass, dtype: int64
```

Output: Series that shows all values in column (Pclass)

```
df["Pclass"].astype('str') #After: String
```

```
0     3
1     1
2     3
3     1
4     3
5     3
6     1
7     3
8     3
9     2
10    3
Name: Pclass, dtype: object
```

Output: Series that shows all values in column (Pclass)

Selecting/Filtering

Recalling contents of the DataFrame

Take note of the column names and index numbers

```
df #shows whole df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	NaN	C
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	G6	S

df[[column names]]

Shows DataFrame with columns with the specified names

```
df[[“PassengerId”, “Age”]] #output column with Passenger ID and Age
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	NaN	C
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	G6	S

	PassengerId	Age
0	1	22.0
1	2	38.0
2	3	26.0
3	4	35.0
4	5	35.0
5	6	NaN
6	7	54.0
7	8	2.0
8	9	27.0
9	10	14.0
10	11	4.0

Output: DataFrame with only the specified columns

df.loc[rows, columns]

Shows DataFrame with rows from columns with the specified names

```
df.loc[:, [“Age”, “PassengerId”]] #output all rows in columns Age and Passenger ID
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	NaN	C
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	G6	S

	Age	PassengerId
0	22.0	1
1	38.0	2
2	26.0	3
3	35.0	4
4	35.0	5
5	NaN	6
6	54.0	7
7	2.0	8
8	27.0	9
9	14.0	10
10	4.0	11

Output: DataFrame with all rows in the specified columns

df.loc[rows, columns]

Shows DataFrame with rows from columns with the specified names

```
df.loc[[0,5], :] #output row labels 0 and 5 with all columns
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	NaN	C
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	G6	S

Output: DataFrame with all columns in the specified row labels

df.iloc[row_index, col_index]

Shows DataFrame with rows and columns filtered by index number

```
df.iloc[:, [0,5]] #output all rows, but only columns of index 0 and 5
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	NaN	C
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	G6	S

→

	PassengerId	Age
0	1	22.0
1	2	38.0
2	3	26.0
3	4	35.0
4	5	35.0
5	6	NaN
6	7	54.0
7	8	2.0
8	9	27.0
9	10	14.0
10	11	4.0

Output: DataFrame with only the specified rows and columns

df[df[condition on column]]

Shows all rows whose value in specific column fulfill a specific condition

```
df[df["Age"] > 30] #output all rows where value in Age column > 30
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	NaN	C
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	G6	S

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S

Output: DataFrame with rows that fulfill the conditions

Multiple boolean indexing

Shows all rows whose value in specific column fulfill multiple conditions

```
df[(df[“Age”] > 30) & (df[“Sex”] == “male”)] #Output rows where Age > 30 and Sex = male
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	NaN	C
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	G6	S

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S

& -> AND, | -> OR

Output: DataFrame with rows that fulfill the conditions

df[Column] = value

Changes all values in the DataFrame column to specified value

```
df4 = df.loc[df[“Age”] > 30] #filter df4 as rows where Age > 30
df4[“Age”] = 10 #values in whole Age col for df4 changed to 10
```

```
df4 #shows whole df4
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	10	1	0	71.2833	C85	C
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	10	1	0	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	10	0	0	8.0500	NaN	S
6	7	0	1	McCarthy, Mr. Timothy J	male	10	0	0	51.8625	E46	S

df.iloc[row_index] = value

Changes all values in the specified row to given value

```
df5 = df4.copy() #creates a deep copy of df4 (all rows where Age > 30)
    df5.iloc[0] = 0 #values in whole Age col for df4 changed to 10
```

```
df5 #shows whole df5
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
1	0	0	0	0	0	0	0	0	0.0000	0	0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	10	1	0	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	10	0	0	8.0500	NaN	S
6	7	0	1	McCarthy, Mr. Timothy J	male	10	0	0	51.8625	E46	S

Simultaneous Filtering and Indexing

Do both in a single line!

```
df[df["Age"] > 30]["Sex"] #Chaining of index
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	NaN	C
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	G6	S

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S

Output: Series representing the column Sex

```
1    female
3    female
4      male
6      male
Name: Sex, dtype: object
```

Applying Functions

df.apply(<function_name>)

Applying a function onto a column of dataframe

```
def has_survived(x):
 if x == 1:
   return True
 else:
   return False
```

```
df["Survived"].apply(has_survived)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	NaN	C
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	G6	S

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	1	False	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
1	2	True	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	3	True	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	4	True	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	5	False	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
5	6	False	3	Moran, Mr. James	male	NaN	0	0	8.4583	NaN	Q
6	7	False	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
7	8	False	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	NaN	S
8	9	True	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	NaN	S
9	10	True	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	NaN	C
10	11	True	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	G6	S

Quiz Time!!!

PollEv

PollEv.com/eugeneganche909

Qn 1. How do we find the number of rows and columns in a df?

I. len(df) & len(df.columns)

II. df.shape()

III. len(df.rows) & len(df.columns)

- I only
- I & II
- All of the above
- II & III
- II only

Qn 1. How do we find the number of rows and columns in a df?

I. len(df) & len(df.columns)

II. df.shape()

III. len(df.rows) & len(df.columns)

I only
I & II
All of the above
II & III
II only

Qn 1. How do we find the number of rows and columns in a df?

I. len(df) & len(df.columns)

II. df.shape()

III. len(df.rows) & len(df.columns)

- I only
- I & II
- All of the above
- II & III
- II only

Qn 2A context

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	NaN	C
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	G6	S

df.loc[2:10, ["PassengerId", "Age", "Sex"]]

→ ?

A

	PassengerId	Age	Sex
2	3	26.0	female
3	4	35.0	female
4	5	35.0	male
5	6	NaN	male
6	7	54.0	male
7	8	2.0	male
8	9	27.0	female
9	10	14.0	female

B

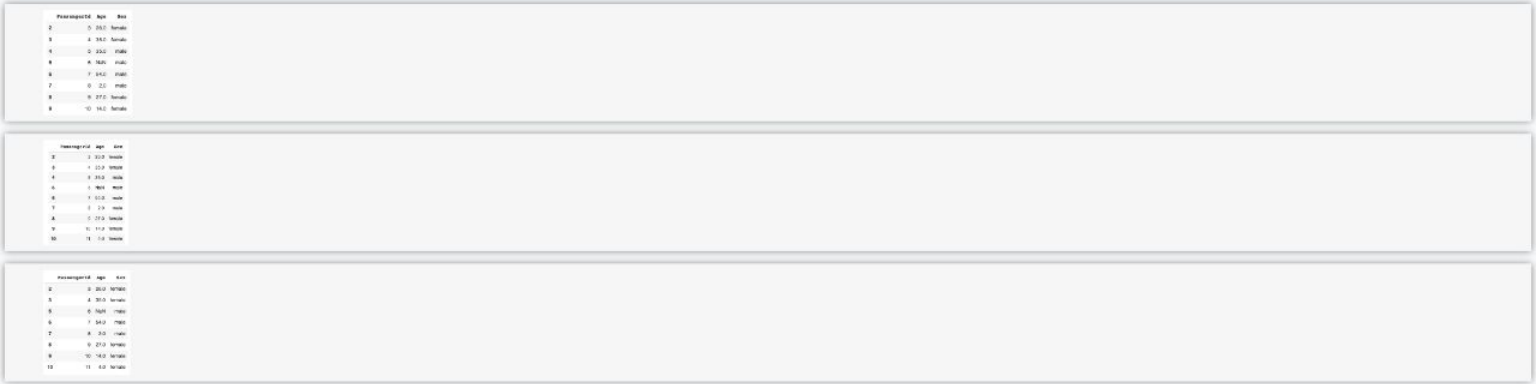
	PassengerId	Age	Sex
2	3	26.0	female
3	4	35.0	female
4	5	35.0	male
5	6	NaN	male
6	7	54.0	male
7	8	2.0	male
8	9	27.0	female
9	10	14.0	female
10	11	4.0	female

C

	PassengerId	Age	Sex
2	3	26.0	female
3	4	35.0	female
5	6	NaN	male
6	7	54.0	male
7	8	2.0	male
8	9	27.0	female
9	10	14.0	female
10	11	4.0	female

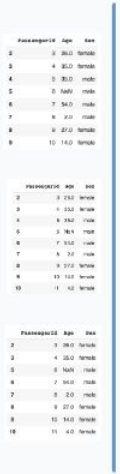
Qn 2A. What would be the result of the df following this line of code?

df.loc[2:10, ["PassengerId", "Age", "Sex"]]



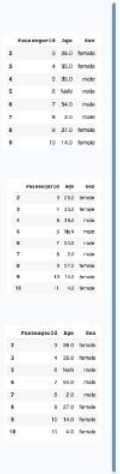
Qn 2A. What would be the result of the df following this line of code?

df.loc[2:10, ["PassengerId", "Age", "Sex"]]



Qn 2A. What would be the result of the df following this line of code?

df.loc[2:10, ["PassengerId", "Age", "Sex"]]



Qn 2B context

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	NaN	C
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	G6	S

df.iloc[2:10,[1,5]]

?

A

	PassengerId	Survived	Age
2	3	1	26.0
3	4	1	35.0
4	5	0	35.0
5	6	0	NaN
6	7	0	54.0
7	8	0	2.0
8	9	1	27.0
9	10	1	14.0
10	11	1	4.0

B

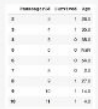
	Survived	Age
2	1	26.0
3	1	35.0
4	0	35.0
5	0	NaN
6	0	54.0
7	0	2.0
8	1	27.0
9	1	14.0

C

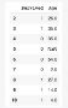
	Survived	Age
2	1	26.0
3	1	35.0
4	0	35.0
5	0	NaN
6	0	54.0
7	0	2.0
8	1	27.0
9	1	14.0
10	1	4.0

Qn 2B. What would be the result of the df following this line of code?

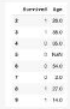
df.iloc[2:10, [1,5]]



A



B



C

Qn 2B. What would be the result of the df following this line of code?

df.iloc[2:10, [1,5]]

A

B

C

Qn 2B. What would be the result of the df following this line of code?

df.iloc[2:10, [1,5]]

A

B

C

Qn 3. What do you think would happen to the df following these lines?

1. df["Age"]=df["Age"].astype(str)

2. df["Age"]=df["Age"].astype(int)

- Oh no! Error being raised at line 1
- 'Age' column data type would convert from float to str, then to int. from str to int
- Oh no! Error being raised at line 2

Qn 3. What do you think would happen to the df following these lines?

1. df["Age"]=df["Age"].astype(str)

2. df["Age"]=df["Age"].astype(int)

Oh no! Error being raised at line 1

'Age' column data type would convert from float to str, then to int. from str to int

Oh no! Error being raised at line 2

Qn 3. What do you think would happen to the df following these lines?

1. df["Age"]=df["Age"].astype(str)

2. df["Age"]=df["Age"].astype(int)

Oh no! Error being raised at line 1

'Age' column data type would convert from float to str, then to int. from str to int

Oh no! Error being raised at line 2

Qn 4. Is it possible to apply a function where we reference multiple columns within a row and create a new column? (Eg. create a new column "net profit" using profit and cost)

	Revenue	Cost
0	200.5	185.0
1	150.6	100.3

sample1_df

Yes, it is possible. **A**

No way! **B**

Create a new column "net profit" using profit and cost

```
sample1_df['profit']=sample1_df.apply(lambda row: row['Revenue']-row['Cost'], axis=1)
```

	Revenue	Cost	profit
0	200.5	185.0	15.5
1	150.6	100.3	50.3

sample1_df after the code

Break Time

Practical Time

Statistics

df.groupby('Column').agg("agg_function")

Group by specified column and perform and aggregating function across the other columns
Agg_function: "min", "max", "sum" , etc.

```
df7.groupby('Sex').agg("mean")
```

Sex	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
female	6.5	1.0	2.166667	24.00	0.666667	0.5	31.702067
male	5.4	0.0	2.600000	28.25	0.800000	0.2	19.339160

Output: Mean values of other columns, grouped by "Sex"

df.groupby(Column).<agg_function>()

Alternative Method of Grouping and Aggregating (More Common)

`df.groupby("Sex").mean()`

Sex	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
female	6.5	1.0	2.166667	24.00	0.666667	0.5	31.702067
male	5.4	0.0	2.600000	28.25	0.800000	0.2	19.339160

Output: Mean values of other columns, grouped by "Sex"

`df.groupby("Sex").sum()`

Sex	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
female	39	6	13	144.0	4	3	190.2124
male	27	0	13	113.0	4	1	96.6958

Think about what data is useful. Is summing up here actually useful?

Output: Summing values of other columns, grouped by "Sex"

df.corr()

Shows the correlation matrix of all numerical data

```
df.corr()
```

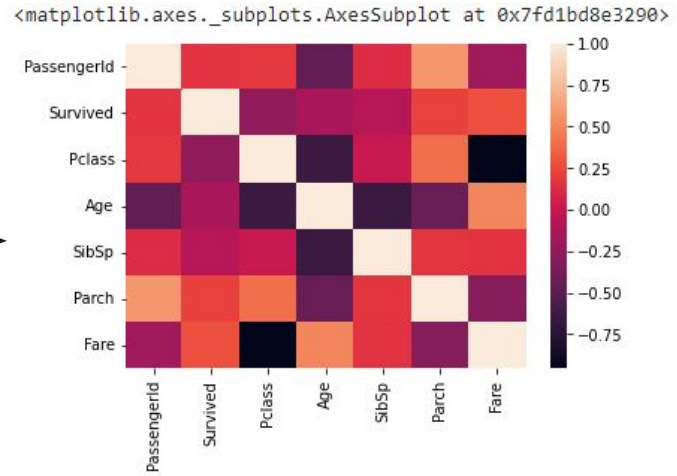
	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	0.173205	0.195698	-0.456264	0.133333	0.581378	-0.171006
Survived	0.173205	1.000000	-0.244804	-0.136987	-0.076980	0.232379	0.285602
Pclass	0.195698	-0.244804	1.000000	-0.644048	0.010872	0.408422	-0.956986
Age	-0.456264	-0.136987	-0.644048	1.000000	-0.653575	-0.424524	0.509039
SibSp	0.133333	-0.076980	0.010872	-0.653575	1.000000	0.178885	0.161232
Parch	0.581378	0.232379	0.408422	-0.424524	0.178885	1.000000	-0.290590
Fare	-0.171006	0.285602	-0.956986	0.509039	0.161232	-0.290590	1.000000

Output: DataFrame representing Correlation matrix of variables in df

A colorful representation of corr()

```
import seaborn as sns
sns.heatmap(df.corr())
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	0.173205	0.195698	-0.456264	0.133333	0.581378	-0.171006
Survived	0.173205	1.000000	-0.244804	-0.136987	-0.076980	0.232379	0.285602
Pclass	0.195698	-0.244804	1.000000	-0.644048	0.010872	0.408422	-0.956986
Age	-0.456264	-0.136987	-0.644048	1.000000	-0.653575	-0.424524	0.509039
SibSp	0.133333	-0.076980	0.010872	-0.653575	1.000000	0.178885	0.161232
Parch	0.581378	0.232379	0.408422	-0.424524	0.178885	1.000000	-0.290590
Fare	-0.171006	0.285602	-0.956986	0.509039	0.161232	-0.290590	1.000000



Output: A color map based on the numbers in the correlation matrix

Manipulating Data with Special Considerations

df.dropna(*optional: [columns]*)

Drops rows with null values in all/specified columns

```
df8 = df.copy()
df8.dropna(subset=[“Age”, “Cabin”])
```

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
6	0	3	Moran, Mr. James	male	NaN	0	0	8.4583	NaN	Q
7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	NaN	S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	NaN	S
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	NaN	C
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	G6	S

Output: DataFrame without rows that contain null values in columns Age and Cabin

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	G6	S

df.fillna(*value, inplace*...)*

Fills null values in DataFrame with value, or using a method

```
df8.fillna("NONE", inplace = True)
OR
df8 = df8.fillna("NONE", inplace = False)

df8 #view df8 after filling NAN
```

Output: DataFrame with all null values filled with "NONE"

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
6	0	3	Moran, Mr. James	male	NaN	0	0	8.4583	NaN	Q
7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	NaN	S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	NaN	S
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	NaN	C
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	G6	S

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NONE	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NONE	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NONE	S
6	0	3	Moran, Mr. James	male	NONE	0	0	8.4583	NONE	Q
7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	NONE	S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	NONE	S
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	NONE	C
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	G6	S

df.fillna(*value, inplace*...*)

Fills null values in DataFrame with value, or using a method

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NONE	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NONE	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NONE	S
6	0	3	Moran, Mr. James	male	NONE	0	0	8.4583	NONE	Q
7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	NONE	S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	NONE	S
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	NONE	C
11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	G6	S

`df8.info()` →

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11 entries, 0 to 10
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  11 non-null     int64
 1   Survived     11 non-null     int64
 2   Pclass       11 non-null     int64
 3   Name         11 non-null     object
 4   Sex          11 non-null     object
 5   Age          11 non-null     object
 6   SibSp        11 non-null     int64
 7   Parch        11 non-null     int64
 8   Fare         11 non-null     float64
 9   Cabin        11 non-null     object
 10  Embarked     11 non-null     object
dtypes: float64(1), int64(5), object(5)
memory usage: 1.1+ KB
```

```
df[df8['Age'] > 30] #ERROR
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-163-7c3e68901722> in <module>
----> 1 df[df8['Age'] > 30]

5 frames
/usr/local/lib/python3.7/dist-packages/pandas/_libs/ops.pyx in pandas._libs.ops.scalar_compare()

TypeError: '>' not supported between instances of 'str' and 'int'
```

df.fillna(*method …)*

Fills null values in DataFrame with value, or using a method

Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
male	22.0	1	0	7.2500	NaN	S
female	38.0	1	0	71.2833	C85	C
female	26.0	0	0	7.9250	NaN	S
female	35.0	1	0	53.1000	C123	S
male	35.0	0	0	8.0500	NaN	S
male	NaN	0	0	8.4583	NaN	Q
male	54.0	0	0	51.8625	E46	S
male	2.0	3	1	21.0750	NaN	S
female	27.0	0	2	11.1333	NaN	S
female	14.0	1	0	30.0708	NaN	C
female	4.0	1	1	16.7000	G6	S

```
df8.fillna(method = "ffill")
```

Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
male	22.0	1	0	7.2500	NaN	S
female	38.0	1	0	71.2833	C85	C
female	26.0	0	0	7.9250	C85	S
female	35.0	1	0	53.1000	C123	S
male	35.0	0	0	8.0500	C123	S
male	35.0	0	0	8.4583	C123	Q
male	54.0	0	0	51.8625	E46	S
male	2.0	3	1	21.0750	E46	S

```
df8.fillna(method = "bfill")
```

Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
male	22.0	1	0	7.2500	C85	S
female	38.0	1	0	71.2833	C85	C
female	26.0	0	0	7.9250	C123	S
female	35.0	1	0	53.1000	C123	S
male	35.0	0	0	8.0500	E46	S
male	54.0	0	0	8.4583	E46	Q
male	54.0	0	0	51.8625	E46	S
male	2.0	3	1	21.0750	G6	S

Merge/Concatenate

pd.concat([df_i,df_j,...])

Add ≥ 2 dfs across rows

df_top

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

df_bottom

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C

```
df9 = pd.concat([df_top, df_btm])
```

pd.concat([df_i,df_j,...])

df9

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C

pd.concat([df_i,df_j,...], axis=1)

Add ≥ 2 dfs across columns

df_left

PassengerId	Survived	Pclass	Name	Sex	Age
1	0	3	Braund, Mr. Owen Harris	male	22.0
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0
3	1	3	Heikkinen, Miss. Laina	female	26.0
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0
5	0	3	Allen, Mr. William Henry	male	35.0
6	0	3	Moran, Mr. James	male	NaN
7	0	1	McCarthy, Mr. Timothy J	male	54.0
8	0	3	Palsson, Master. Gosta Leonard	male	2.0
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0

df_right

SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	A/5 21171	7.2500	NaN	S
1	0	PC 17599	71.2833	C85	C
0	0	STON/O2. 3101282	7.9250	NaN	S
1	0	113803	53.1000	C123	S
0	0	373450	8.0500	NaN	S
0	0	330877	8.4583	NaN	Q
0	0	17463	51.8625	E46	S
3	1	349909	21.0750	NaN	S
0	2	347742	11.1333	NaN	S
1	0	237736	30.0708	NaN	C

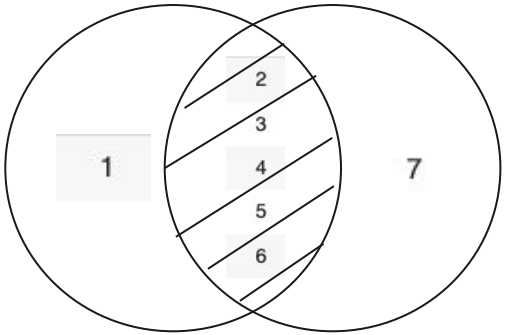
```
df10=pd.concat([df_left,df_right], axis=1)
```

pd.concat([df_i,df_j,...], axis=1)

`df10`

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C

pd.merge([df_i,df_j], on=column_name)



df_merge1

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

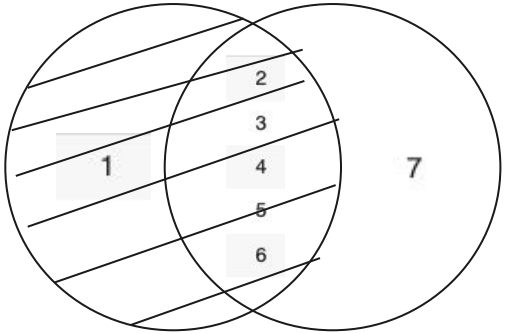
df_merge2

PassengerId	Ticket	Age
2	PC 17599	38.0
3	STON/O2. 3101282	26.0
4	113803	35.0
5	373450	35.0
6	330877	NaN
7	17463	54.0

```
df_merge1.merge(df_merge2, on= 'PassengerId')
```

PassengerId	Name	Ticket	Age
2	Cumings, Mrs. John Bradley (Florence Briggs Th...	PC 17599	38.0
3	Heikkinen, Miss. Laina	STON/O2. 3101282	26.0
4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	113803	35.0
5	Allen, Mr. William Henry	373450	35.0
6	Moran, Mr. James	330877	NaN

pd.merge(..., how= 'left')



df_merge1

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

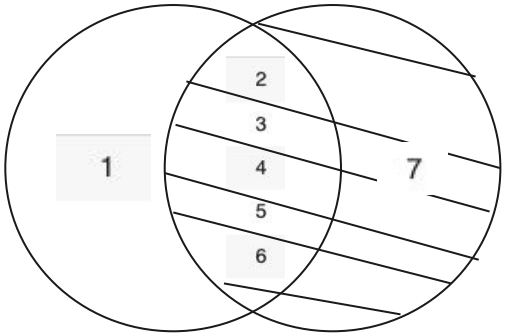
df_merge2

PassengerId	Ticket	Age
2	PC 17599	38.0
3	STON/O2. 3101282	26.0
4	113803	35.0
5	373450	35.0
6	330877	NaN
7	17463	54.0

```
df_merge1.merge(df_merge2, on= 'PassengerId',
how = 'left')
```

PassengerId	Name	Ticket	Age
1	Braund, Mr. Owen Harris	NaN	NaN
2	Cumings, Mrs. John Bradley (Florence Briggs Th...	PC 17599	38.0
3	Heikkinen, Miss. Laina	STON/O2. 3101282	26.0
4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	113803	35.0
5	Allen, Mr. William Henry	373450	35.0
6	Moran, Mr. James	330877	NaN

pd.merge(..., how= 'right')



df_merge1

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

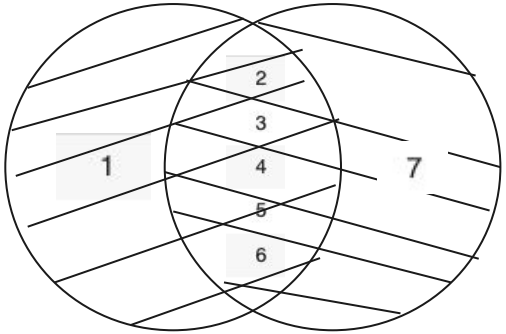
df_merge2

PassengerId	Ticket	Age
2	PC 17599	38.0
3	STON/O2. 3101282	26.0
4	113803	35.0
5	373450	35.0
6	330877	NaN
7	17463	54.0

```
df_merge1.merge(df_merge2, on= 'PassengerId',
how = 'right')
```

PassengerId	Name	Ticket	Age
2	Cumings, Mrs. John Bradley (Florence Briggs Th...	PC 17599	38.0
3	Heikkinen, Miss. Laina	STON/O2. 3101282	26.0
4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	113803	35.0
5	Allen, Mr. William Henry	373450	35.0
6	Moran, Mr. James	330877	NaN
7	NaN	17463	54.0

pd.merge(..., how= 'outer')



df_merge1

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

df_merge2

PassengerId	Ticket	Age
2	PC 17599	38.0
3	STON/O2. 3101282	26.0
4	113803	35.0
5	373450	35.0
6	330877	NaN
7	17463	54.0

```
df_merge1.merge(df_merge2, on= 'PassengerId',
how = 'outer')
```

PassengerId	Name	Ticket	Age
1	Braund, Mr. Owen Harris	NaN	NaN
2	Cumings, Mrs. John Bradley (Florence Briggs Th...	PC 17599	38.0
3	Heikkinen, Miss. Laina	STON/O2. 3101282	26.0
4	Futrelle, Mrs. Jacques Heath (Lily May Peel)	113803	35.0
5	Allen, Mr. William Henry	373450	35.0
6	Moran, Mr. James	330877	NaN
7	NaN	17463	54.0

Quiz Time!!!

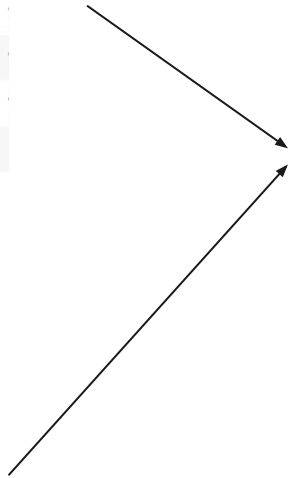
Qn 5 context

df_merge1

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

df_merge2

PassengerId	Ticket	Age
2	PC 17599	38.0
3	STON/O2. 3101282	26.0
4	113803	35.0
5	373450	35.0
6	330877	NaN
7	17463	54.0



How do I find all the passengers with name but do not have an age using df_merge1 and df_merge2?

Qn 5. How do I find all the passengers with name but do not have an age using df_merge1 and df_merge2?

- Merge the 2 tables via INNER join, then filter the joined tables accordingly using isnull() on "Age" column
- Merge the 2 tables via LEFT join, then filter the joined tables accordingly using isnull() on "Age" column
- Merge the 2 tables via RIGHT join, then filter the joined tables accordingly using isnull() on "Age" column

Qn 5. How do I find all the passengers with name but do not have an age using df_merge1 and df_merge2?

Merge the 2 tables via INNER join, then filter the joined tables accordingly using isnull() on "Age" column

Merge the 2 tables via LEFT join, then filter the joined tables accordingly using isnull() on "Age" column

Merge the 2 tables via RIGHT join, then filter the joined tables accordingly using isnull() on "Age" column

Qn 5. How do I find all the passengers with name but do not have an age using df_merge1 and df_merge2?

Merge the 2 tables via INNER join, then filter the joined tables accordingly using isnull() on "Age" column

Merge the 2 tables via LEFT join, then filter the joined tables accordingly using isnull() on "Age" column

Merge the 2 tables via RIGHT join, then filter the joined tables accordingly using isnull() on "Age" column

Qn 5 suggested code

```
df_after_merge=df_merge1.merge(df_merge2, on= 'PassengerId', how='left')
df_after_merge[df_after_merge['Age'].isnull()]
```

Qn 6 context

	Revenue	Cost	type
0	200.5	185.0	gadget
1	150.6	100.3	gadget
2	130.0	10.0	rock
3	80.0	5.0	NaN
4	5.0	6.0	stationery

sample2_df

sample_df.dropna(inplace=True)

	Revenue	Cost	type
0	200.5	185.0	gadget
1	150.6	100.3	gadget
2	130.0	10.0	rock
4	5.0	6.0	stationery

sample2_df

Qn 6A. Given the scneario where I perform dropna function, what would happen if I perform the following line of code?

sample2_df.loc[3,]

- Error
- No Error. It will still output me the last row.

Qn 6A. Given the scneario where I perform dropna function, what would happen if I perform the following line of code?

sample2_df.loc[3,]

Error	✓ 0%
No Error. It will still output me the last row.	

Qn 6A. Given the scneario where I perform dropna function, what would happen if I perform the following line of code?

sample2_df.loc[3,]

Error	✓ 0%
No Error. It will still output me the last row.	

Qn 6B. Given the scneario where I perform dropna function, what would happen if I perform the following line of code?

sample2_df.iloc[3,]

No Error, it will output the last row

Error

Qn 6B. Given the scneario where I perform dropna function, what would happen if I perform the following line of code?

sample2_df.iloc[3,]

No Error, it will output the last row	✓ 0%
Error	

Qn 6B. Given the scneario where I perform dropna function, what would happen if I perform the following line of code?

sample2_df.iloc[3,]

No Error, it will output the last row	✓ 0%
Error	

```
Revenue              5.0
Cost                 6.0
type          stationery
Name: 4, dtype: object
```

Practical 2

Conclusion

You have learnt how to do the following...

1. Viewing the Dataframe
2. Changing datatypes
3. Selecting / Indexing/ Filtering
4. Applying functions
5. Statistical functions (Aggregations)
6. Handling missing values
7. Merging / Concatenate

Workshop Materials Tinyurl

https://tinyurl.com/sdspandasmaterials

Feedback Form

https://tinyurl.com/PandasHackerSDS