**Homework Week 1**

**Question 1 [20 points]**

Jupyter Notebooks (.ipynb) and Python files (.py) are both employed for coding in Python, yet they differ widely in framework, use, and adaptability, more so when it pertains to data analysis. Jupyter Notebooks are interactive environments that allow you to write and run code. This constitutes a web-based programme that enables you produce and distribute documents with live code, mathematics, visualisations, and written content. It is a great option for data cleansing and transformation, computational statistical modelling, data visualisation, machine learning, and other applications.

Advantages:

* Interactive interface: enables interactive way to explore and visualise data.
* Sharing: The notebook can be shared via pdf / github – helpful and insightful for collaboration
* Supports multiple languages – it can execute the code and show the output. Allows integration.

Disadvantages:

* Testing / writing it can be a challenge because of the notebook’s interactivity.
* Cells can be executed in any order – which can be misapprehension when sharing work with others.

Python Advantages:

* Python scripts are simple to construct, debug, and comprehend.
* High-performance workloads and massive data processing jobs are more efficient.

Disadvantages:

* Python scripts, unlike notebooks, do not support interactive and exploratory coding.
* Less suited to presentations and demonstrations: Python scripts, unlike Jupyter notebooks, are less appropriate for communicating analysis methods  due to the fact that they do not enable in-line graphics or explanatory text.

**Question 2 [10 points]**

A Pandas Series, is an a single-dimensional array-like object that may carry any data type. It consists of essentially a single data column which is capable of holding any data format (Integers, Strings, Floats,

import pandas as pd

data = {'a': 0., 'b': 1., 'c': 2.}

s = pd.Series(data)

print(s)

In contrast, a Pandas DataFrame is a two-dimensional , heterogeneous columnar data  with marked axes (rows and columns).

import pandas as pd

data = [{'a': 1, 'b': 2}, {'a': 5, 'b': 10, 'c': 20}]

df = pd.DataFrame(data)

print(df)

**Question 3 [10 points]**

Rectangular data, such as a Pandas DataFrame, is split into rows and columns in the same way that a table is. Each row offers an observation, and each column represents a feature. e.g. sql table. Non-rectangular data, on the other hand, does not conform to this structure and includes data types such as graphs, and unstructured data like images.

**Question 4 [10 points]**

1. Visualisations such box diagrams and scatter plots can instantly highlight outliers that must be handled before additional analysis or simulation

A scatter plot of cars against the cost of them, for instance, may clearly demonstrate cars that are exceptionally expensive or inexpensive taking into account their make  in a dataset of car prices.

1. Showing relationships : Scatter plots are capable of demonstrating the correlation between two variables, which may assist in decision-making in a commercial situation.

**Question 5 [50 points]**

**Each sub-question is worth 10 points.**

Using the *titanic dataset* which you can read into your notebook using the following code,

import pandas as pd

titanic = pd.read\_csv('https://raw.githubusercontent.com/mwaskom/seaborn-data/master/titanic.csv')

A screenshot of a computer

Description automatically generated

Codes :

#a

rows, cols = titanic.shape

print(f'The dataset has {rows} rows and {cols} columns.')

#b

print(titanic.describe())

#c

print(titanic.isna().sum())

#d

titanic.dropna(subset=['age'], inplace=True)

bins = range(0, 101, 10)

labels = ['0-10', '11-20', '21-30', '31-40', '41-50', '51-60', '61-70', '71-80', '81-90', '91-100']

titanic['age\_range'] = pd.cut(titanic['age'], bins=bins, labels=labels, include\_lowest=True)

print(titanic.head())

# e

print(titanic.groupby(['age\_range', 'sex']).size())

print(titanic.groupby(['age\_range', 'class']).size())

Output for all :

The dataset has 891 rows and 15 columns.

survived pclass age sibsp parch fare

count 891.000000 891.000000 714.000000 891.000000 891.000000 891.000000

mean 0.383838 2.308642 29.699118 0.523008 0.381594 32.204208

std 0.486592 0.836071 14.526497 1.102743 0.806057 49.693429

min 0.000000 1.000000 0.420000 0.000000 0.000000 0.000000

25% 0.000000 2.000000 20.125000 0.000000 0.000000 7.910400

50% 0.000000 3.000000 28.000000 0.000000 0.000000 14.454200

75% 1.000000 3.000000 38.000000 1.000000 0.000000 31.000000

max 1.000000 3.000000 80.000000 8.000000 6.000000 512.329200

survived 0

pclass 0

sex 0

age 177

sibsp 0

parch 0

fare 0

embarked 2

class 0

who 0

adult\_male 0

deck 688

embark\_town 2

alive 0

alone 0

dtype: int64

survived pclass sex age sibsp parch fare embarked class \

0 0 3 male 22.0 1 0 7.2500 S Third

1 1 1 female 38.0 1 0 71.2833 C First

2 1 3 female 26.0 0 0 7.9250 S Third

3 1 1 female 35.0 1 0 53.1000 S First

4 0 3 male 35.0 0 0 8.0500 S Third

who adult\_male deck embark\_town alive alone age\_range

0 man True NaN Southampton no False 21-30

1 woman False C Cherbourg yes False 31-40

2 woman False NaN Southampton yes True 21-30

3 woman False C Southampton yes False 31-40

4 man True NaN Southampton no True 31-40

age\_range sex

0-10 female 31

male 33

11-20 female 46

male 69

21-30 female 81

male 149

31-40 female 55

male 100

41-50 female 31

male 55

51-60 female 14

male 28

61-70 female 3

male 14

71-80 female 0

male 5

81-90 female 0

male 0

91-100 female 0

male 0

dtype: int64

age\_range class

0-10 First 3

Second 17

Third 44

11-20 First 18

Second 18

Third 79

21-30 First 40

Second 61

Third 129

31-40 First 49

Second 43

Third 63

41-50 First 37

Second 19

Third 30

51-60 First 25

Second 12

Third 5

61-70 First 11

Second 3

Third 3

71-80 First 3

Second 0

Third 2

81-90 First 0

Second 0

Third 0

91-100 First 0

Second 0

Third 0

dtype: int64