

# Data Technician

**Name: Laiba Shaukat**

**Course Date: 12 May 2025**

**Table of contents**

Day 2: Task 1 .....	3
Day 3: Task 1 .....	4
Exercise 1: Loading and Exploring the Data .....	4
Exercise 2: Indexing and Slicing .....	6
Exercise 3: Data Manipulation.....	8
Exercise 4: Aggregation and Grouping .....	8
Exercise 5: Advanced Operations .....	9
Exercise 6: Exporting Data.....	11
Exercise 7: If finished early try visualising the results.....	11
Day 4: Task 1 .....	12
Day 4: Task 2 .....	14
Course Notes .....	16
Additional Information.....	17



## Day 2: Task 1

It is a common software development interview question to create the below with a certain programming language. Create the below using Python syntax, test it and past the completed syntax and output below.

FizzBuzz:

Go through the integers from 1 to 100.

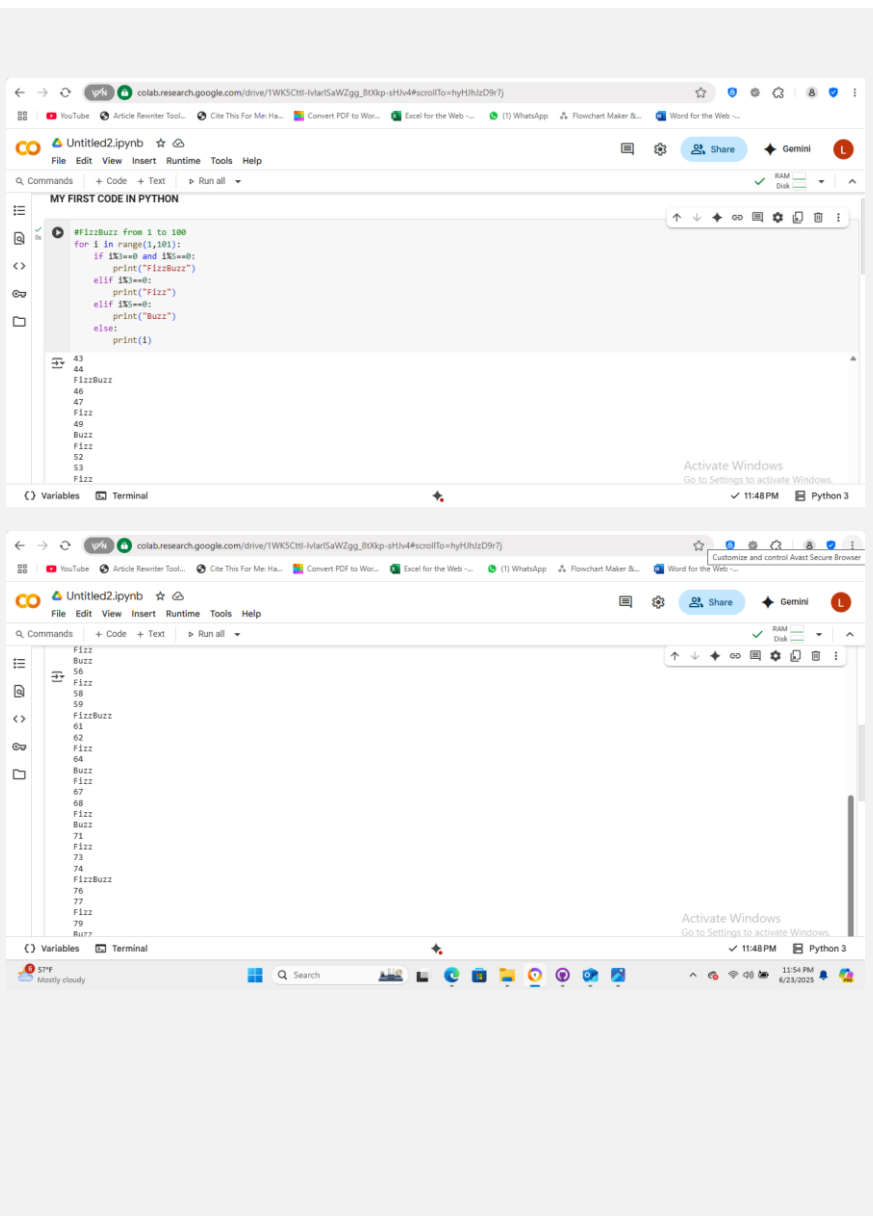
If a number is divisible by 3, print "fizz."

If a number is divisible by 5, print "buzz."

If a number is both divisible by 3 and by 5, print "fizzbuzz."

Otherwise, print just the number.

Paste your completed work to the right

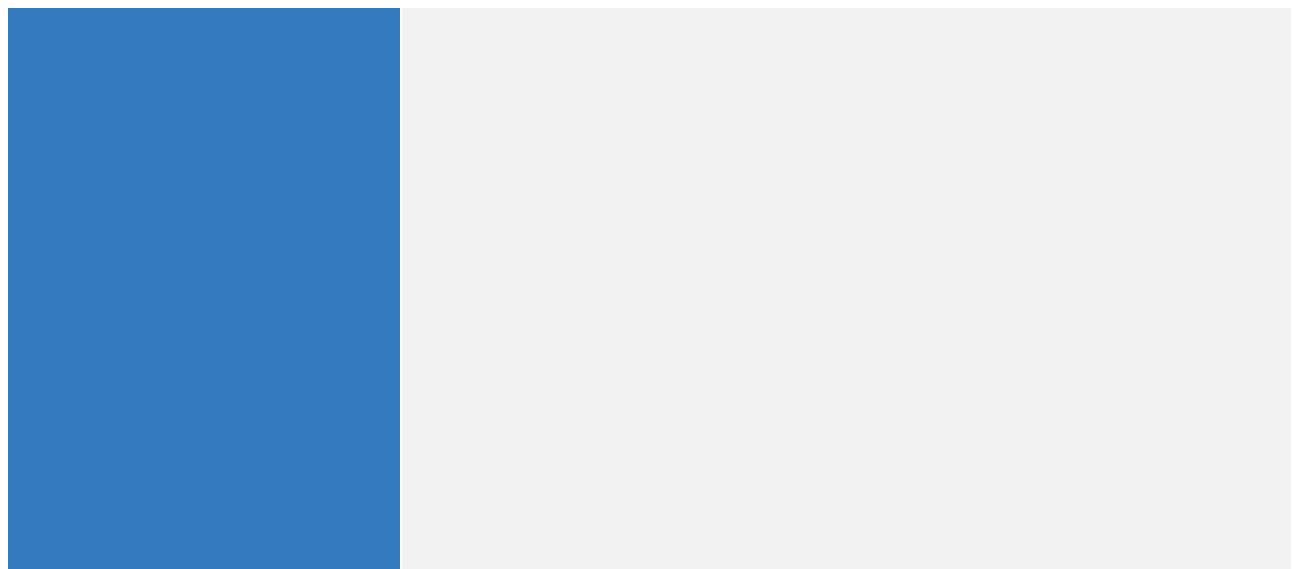


The image displays two screenshots of a Jupyter Notebook interface. The top screenshot shows the code for the FizzBuzz program, which iterates from 1 to 100 and prints 'fizz' for multiples of 3, 'buzz' for multiples of 5, and 'fizzbuzz' for multiples of both. The bottom screenshot shows the output of the program, which is a list of numbers from 1 to 100, with 'fizz' replacing multiples of 3, 'buzz' replacing multiples of 5, and 'fizzbuzz' replacing multiples of both.

```
#FizzBuzz from 1 to 100
for i in range(1,101):
    if i%3==0 and i%5==0:
        print("FizzBuzz")
    elif i%3==0:
        print("fizz")
    elif i%5==0:
        print("buzz")
    else:
        print(i)
```

43  
44  
FizzBuzz  
46  
47  
Fizz  
49  
Buzz  
Fizz  
52  
53  
Fizz

Fizz  
56  
Fizz  
58  
59  
FizzBuzz  
61  
62  
Fizz  
64  
Buzz  
Fizz  
67  
68  
Fizz  
Buzz  
71  
Fizz  
73  
74  
FizzBuzz  
76  
77  
Fizz  
79  
Buzz




## Day 3: Task 1

Download the 'student.csv', complete the below exercises as a group and paste your input and output. Although this is a group activity, everyone should have the below answered so it supports your portfolio:

### Exercise 1: Loading and Exploring the Data

1. Question: "Write the code to read a CSV file into a Pandas DataFrame."
2. Question: "Write the code to display the first 5 rows of the DataFrame."
3. Question: "Write the code to get the information about the DataFrame."
4. Question: "Write the code to get summary statistics for the DataFrame."




Day3\_Notebook.ipynb
☆
🔗

File
Edit
View
Insert
Runtime
Tools
Help

🔍 Commands
+ Code
+ Text
▶ Run all ▼

☰

0s [18]

```
# 1. Read the file into a table called df
df = pd.read_csv('student.csv')

# 3. Peek at the first 5 rows to check your data
df.head()
```

🔍

↔

	id	name	class	mark	gender
0	1	John Deo	Four	75	female
1	2	Max Ruin	Three	85	male
2	3	Arnold	Three	55	male
3	4	Krish Star	Four	60	female
4	5	John Mike	Four	60	female

📊

📈

File
Edit
View
Insert
Runtime
Tools
Help

🔍 Commands
+ Code
+ Text
▶ Run all ▼

☰

0s ▶

```
df = pd.read_csv('student.csv')

# 2. Peek at the first 5 rows to check your data
df.head()

# 3. Check the shape (rows & columns)
print("Shape (rows, columns):", df.shape)

# 4. Check the data types
print("\nData types:")
print(df.dtypes)

# 5. List all column names
print("\nColumns:")
print(list(df.columns))

# 6. See basic statistics
print("\nDescriptive stats:")
print(df.describe())

# 7. Get full info
print("\nInfo:")
df.info()
```

🔍

↔

🔑

📁

{} Variables

📄 Terminal

◆

File Edit View Insert Runtime Tools Help

Q Commands

+ Code + Text

▶ Run all ▼

Shape (rows, columns): (35, 5)

⌵

⌵

<>

🔑

📁

Data types:

id int64

name object

class object

mark int64

gender object

dtype: object

Columns:

['id', 'name', 'class', 'mark', 'gender']

Descriptive stats:

id mark

count 35.000000 35.000000

mean 18.000000 74.657143

std 10.246951 16.401117

min 1.000000 18.000000

25% 9.500000 62.500000

50% 18.000000 79.000000

75% 26.500000 88.000000

max 35.000000 96.000000

File Edit View Insert Runtime Tools Help

Q Commands

+ Code + Text

▶ Run all ▼

⌵

⌵

<>

🔑

📁

Info:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 35 entries, 0 to 34

Data columns (total 5 columns):

# Column Non-Null Count Dtype

---

0 id 35 non-null int64

1 name 34 non-null object

2 class 34 non-null object

3 mark 35 non-null int64

4 gender 33 non-null object

dtypes: int64(2), object(3)

memory usage: 1.5+ KB

## Exercise 2: Indexing and Slicing

1. Question: "Write the code to select the 'name' column."
2. Question: "Write the code to select the 'name' and 'mark' columns."
3. Question: "Write the code to select the first 3 rows."
4. Question: "Write the code to select all rows where the 'class' is 'Four'."

Data Technician | Workbook | v1.

Page 6 of 17

FileEditViewInsertRuntimeToolsHelp

Q Commands+ Code+ Text▶ Run all▼

⋮

🔍

<>

🔑

📁

▶

1. # Select the "Name" column  
df["name"]

2. # Select the "Name" and "Mark" columns  
df[["name", "mark"]]

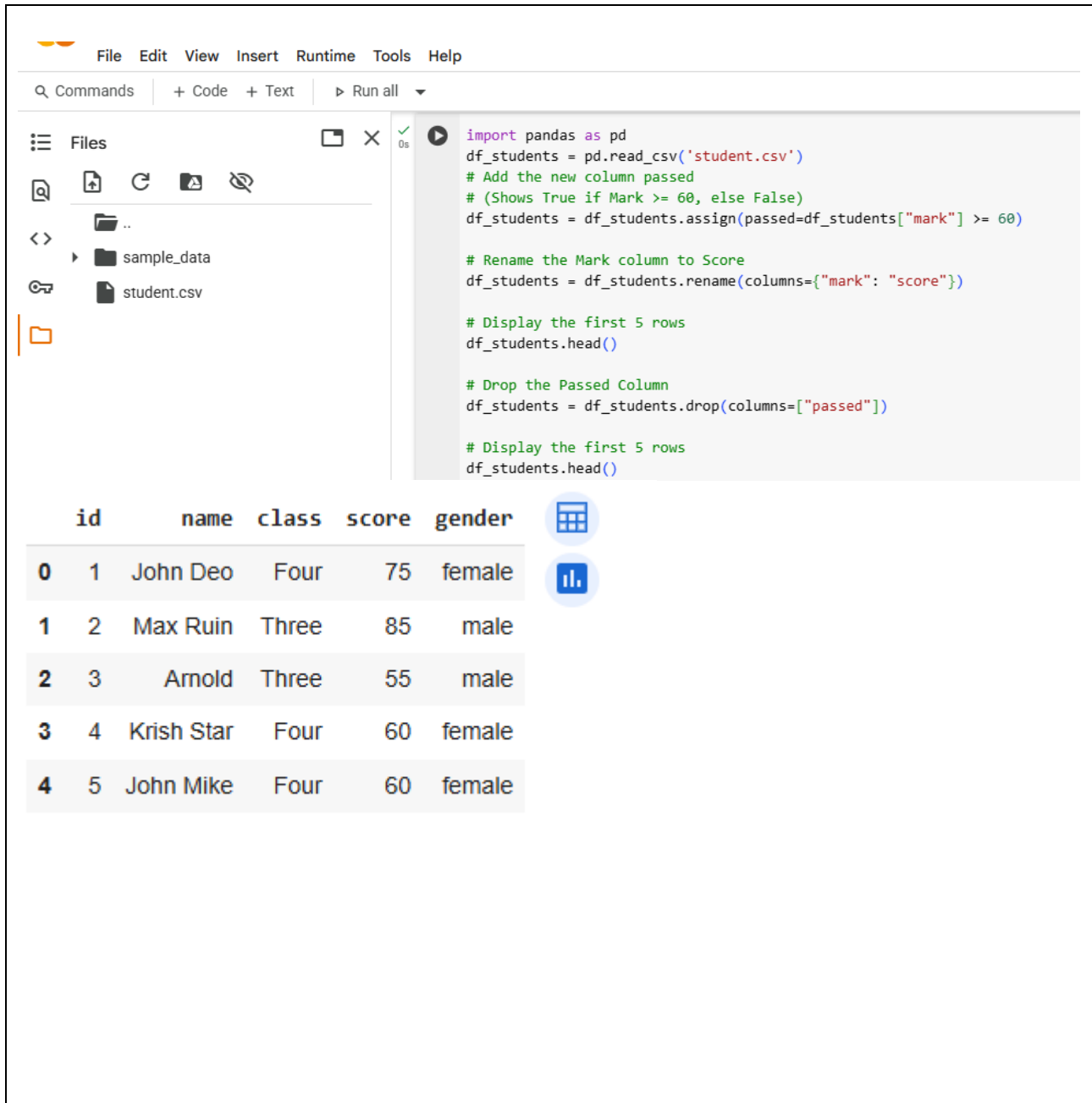
3. # Select the first 3 rows  
df.head(3)

4. # Select all rows where class is four  
df[df["class"] == "Four"]

	id	name	class	mark	gender
0	1	John Deo	Four	75	female
3	4	Krish Star	Four	60	female
4	5	John Mike	Four	60	female
5	6	Alex John	Four	55	male
9	10	Big John	Four	55	female
15	16	Gimmy	Four	88	male
20	21	Babby John	Four	69	female
30	31	Marry Toeey	Four	88	male

## Exercise 3: Data Manipulation

1. Question: "Write the code to add a new column 'passed' that indicates whether the student passed (mark >= 60)."
2. Question: "Write the code to rename the 'mark' column to 'score'."
3. Question: "Write the code to drop the 'passed' column."



The screenshot shows a Jupyter Notebook interface with a file explorer on the left, a code editor in the center, and a DataFrame output at the bottom. The file explorer shows a directory structure with 'sample\_data' and 'student.csv'. The code editor contains the following Python code:

```
import pandas as pd
df_students = pd.read_csv('student.csv')
# Add the new column passed
# (Shows True if Mark >= 60, else False)
df_students = df_students.assign(passed=df_students["mark"] >= 60)

# Rename the Mark column to Score
df_students = df_students.rename(columns={"mark": "score"})

# Display the first 5 rows
df_students.head()

# Drop the Passed Column
df_students = df_students.drop(columns=["passed"])

# Display the first 5 rows
df_students.head()
```

The output shows a DataFrame with the following columns: id, name, class, score, gender. The data is as follows:

	id	name	class	score	gender
0	1	John Deo	Four	75	female
1	2	Max Ruin	Three	85	male
2	3	Arnold	Three	55	male
3	4	Krish Star	Four	60	female
4	5	John Mike	Four	60	female

## Exercise 4: Aggregation and Grouping

1. Question: "Write the code to group the DataFrame by the 'class' column and calculate the mean 'mark' for each group."
2. Question: "Write the code to count the number of students in each class."
3. Question: "Write the code to calculate the average mark for each gender."





The screenshot shows a Jupyter Notebook environment. The left sidebar displays the file explorer with a folder named 'sample\_data' and a file named 'student.csv'. The main area contains a code cell with the following Python code:

```
import pandas as pd
df_students = pd.read_csv('student.csv')
df = pd.DataFrame(df_students)

# Group by Class and calculate mean Mark
df.groupby("class")["mark"].mean()

# Count the number of students in each class
df['class'].value_counts()

# Calculate Average mark for each gender
df.groupby("gender")["mark"].mean()
```

Below the code cell, the output is displayed as a table:

mark	
gender	
female	77.312500
male	71.588235

The output also indicates the data type: `dtype: float64`.

## Exercise 5: Advanced Operations

1. Question: "Write the code to create a pivot table with 'class' as rows, 'gender' as columns, and 'mark' as values."
2. Question: "Write the code to create a new column 'grade' where marks  $\geq 85$  are 'A', 70-84 are 'B', 60-69 are 'C', and below 60 are 'D'."
3. Question: "Write the code to sort the DataFrame by 'mark' in descending order."

File Edit View Insert Runtime Tools Help

Q Commands

+ Code + Text

▶ Run all

Files

..

sample\_data

student.csv

import pandas as pd

df\_students = pd.read\_csv('student.csv')

df = pd.DataFrame(df\_students)

# Class as rows, Gender as columns, Mark as values

pivot\_table = df.pivot\_table(index='class', columns='gender', values='mark', aggfunc='mean')

print(pivot\_table)

# Create a Grade column

def assign\_grade(mark):

if mark >= 85:

return 'A'

elif mark >= 70:

return 'B'

elif mark >= 60:

return 'C'

else:

return 'D'

df['grade'] = df['mark'].apply(assign\_grade)

print(df)

# Sort the DataFrame by Mark in descending order

df\_sorted = df.sort\_values(by='mark', ascending=False)

print(df\_sorted)

Disk

70.27 GB available

Variables

Terminal

9:06 AM

File Edit View Insert Runtime Tools Help

Q Commands

+ Code + Text

▶ Run all

Files

..

sample\_data

student.csv

print(df\_sorted)

13 14 Bigy Seven 88 female A

14 15 Tade Row NaN 88 male A

15 16 Gimmy Four 88 male A

16 17 Tumyu Six 54 male D

17 18 Honny Five 75 male B

18 19 Tinny Nine 18 male D

19 20 Jackly Nine 65 female C

20 21 Babby John Four 69 female C

21 22 Reggid Seven 55 female D

22 23 Herod Eight 79 male B

23 24 Tiddy Now Seven 78 male B

24 25 Giff Tow Seven 88 male A

25 26 Crelea Seven 79 male B

26 27 NaN Three 81 NaN B

27 28 Rojj Base Seven 86 female A

28 29 Tess Played Seven 55 male D

29 30 Reppy Red Six 79 female B

30 31 Marry Toeey Four 88 male A

31 32 Binn Rott Seven 90 female A

32 33 Kenn Rein Six 96 female A

33 34 Gain Toe Seven 69 male C

34 35 Rows Noump Six 88 female A

id name class mark gender grade

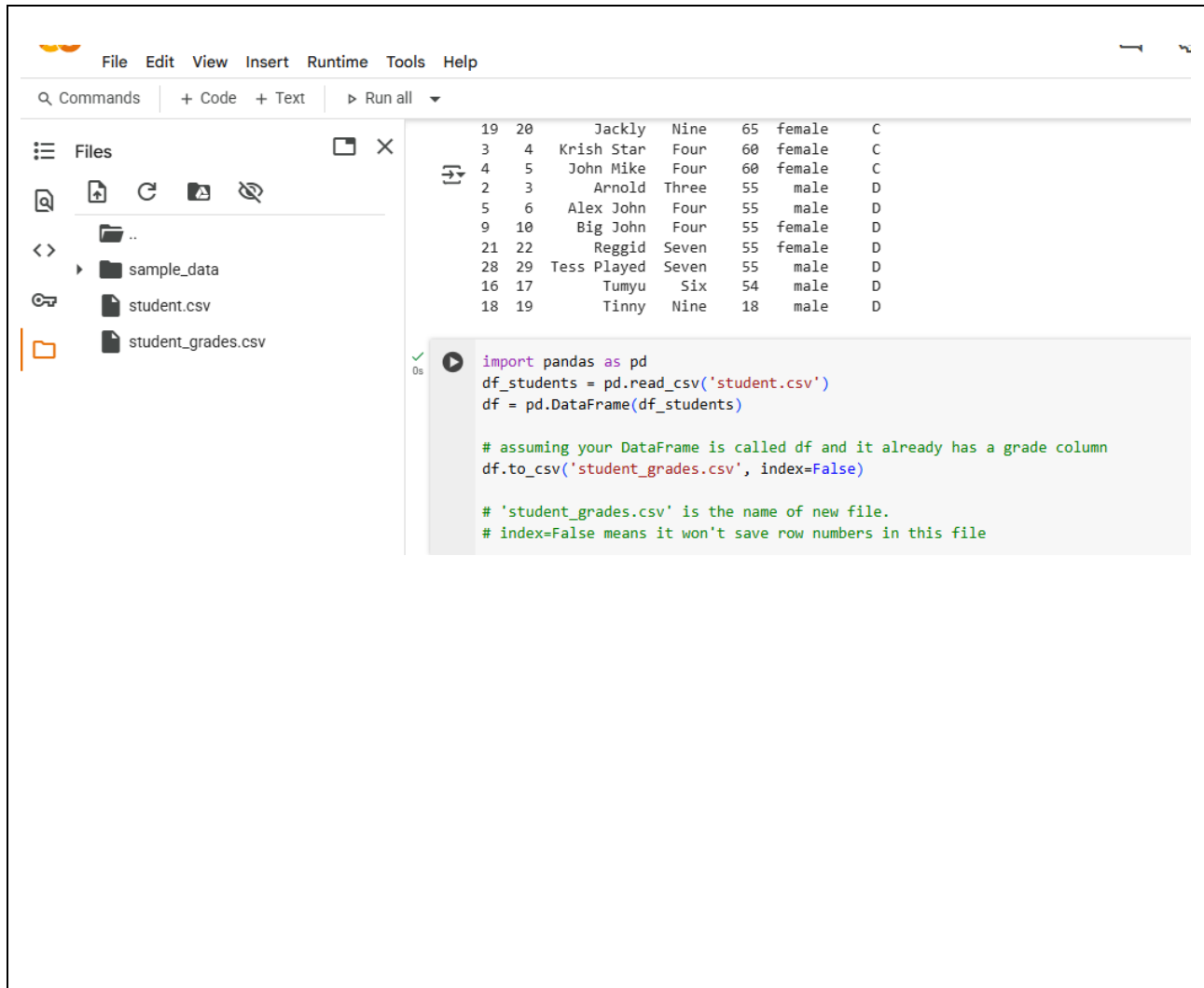
Disk

70.27 GB available



## Exercise 6: Exporting Data

1. Question: "Write the code to save the DataFrame with the new 'grade' column to a new CSV file."



The screenshot shows a Jupyter Notebook interface. On the left, the 'Files' sidebar displays a directory structure with 'sample\_data' containing 'student.csv' and 'student\_grades.csv'. The main area shows a DataFrame with 10 rows and 7 columns. The code cell below the DataFrame contains the following Python code:

```
import pandas as pd
df_students = pd.read_csv('student.csv')
df = pd.DataFrame(df_students)

# assuming your DataFrame is called df and it already has a grade column
df.to_csv('student_grades.csv', index=False)

# 'student_grades.csv' is the name of new file.
# index=False means it won't save row numbers in this file
```

## Exercise 7: If finished early try visualising the results



## Day 4: Task 1

Using the 'GDP (nominal) per Capita.csv' which can be downloaded from the shared Folder, complete the below exercises and paste your input and output. Work individually, but we will work and support each other in the room.

- Read and save the 'GDP (nominal) per Capita' data to a data frame called "df" in Jupyter notebook
- Print the first 10 rows
- Print the last 5 rows
- Print 'Country/Territory' and 'UN\_Region' columns



File Edit View Insert Runtime Tools Help

Q Commands

+ Code + Text

▶ Run all

Files

..

sample\_data

GDP (nominal) per Capita.csv

```

# Import pandas
import pandas as pd

# Read the CSV file into a DataFrame named df
df = pd.read_csv('GDP (nominal) per Capita.csv')

# Display the DataFrame
print(df)

# Print the first 10 rows
print(df.head(10))

# Print the last 5 rows
print(df.tail(5))

# Print only the 'Country/Territory' and 'UN_Region' columns
print(df[['Country/Territory', 'UN_Region']])

```

218

...

...

613

...

219

1072

2015

400

2021

220

480

2021

505

2021

What can I help you build?

Disk 70.29 GB available

Variables Terminal

File Edit View Insert Runtime Tools Help

Q Commands

+ Code + Text

▶ Run all

Files

..

sample\_data

GDP (nominal) per Capita.csv

```
print(df[['Country/Territory', 'UN_Region']])
```

218

...

...

613

...

219

1072

2015

400

2021

220

480

2021

505

2021

221

369

2021

373

2021

222

222

2021

311

2021

[223 rows x 9 columns]

Unnamed: 0

Country/Territory

UN\_Region

IMF\_Estimate

IMF\_Year

\

0

1

Monaco

Europe

0

0

1

2

Liechtenstein

Europe

0

0

2

3

Luxembourg

Europe

132372

2023

3

4

Ireland

Europe

114581

2023

4

5

Bermuda

Americas

0

0

5

6

Norway

Europe

101103

2023

6

7

Switzerland

Europe

98767

2023

7

8

Singapore

Asia

91100

2023

8

9

Isle of Man

Europe

0

0

9

10

Cayman Islands

Americas

0

0

WorldBank\_Estimate

WorldBank\_Year

UN\_Estimate

UN\_Year

0

2021

2021

2021

What can I help you build?

Disk 70.29 GB available

Variables Terminal

File Edit View Insert Runtime Tools Help

Q Commands

+ Code + Text

▶ Run all

Files

..

sample\_data

GDP (nominal) per Capita.csv

```
print(df[['Country/Territory', 'UN_Region']])
```

218

...

...

613

...

219

1072

2015

400

2021

220

480

2021

505

2021

221

369

2021

373

2021

222

222

2021

311

2021

[223 rows x 9 columns]

Unnamed: 0

Country/Territory

UN\_Region

IMF\_Estimate

IMF\_Year

\

0

1

Monaco

Europe

0

0

1

2

Liechtenstein

Europe

0

0

2

3

Luxembourg

Europe

132372

2023

3

4

Ireland

Europe

114581

2023

4

5

Bermuda

Americas

0

0

5

6

Norway

Europe

101103

2023

6

7

Switzerland

Europe

98767

2023

7

8

Singapore

Asia

91100

2023

8

9

Isle of Man

Europe

0

0

9

10

Cayman Islands

Americas

0

0

WorldBank\_Estimate

WorldBank\_Year

UN\_Estimate

UN\_Year

0

2021

2021

2021

What can I help you build?

Disk 70.29 GB available

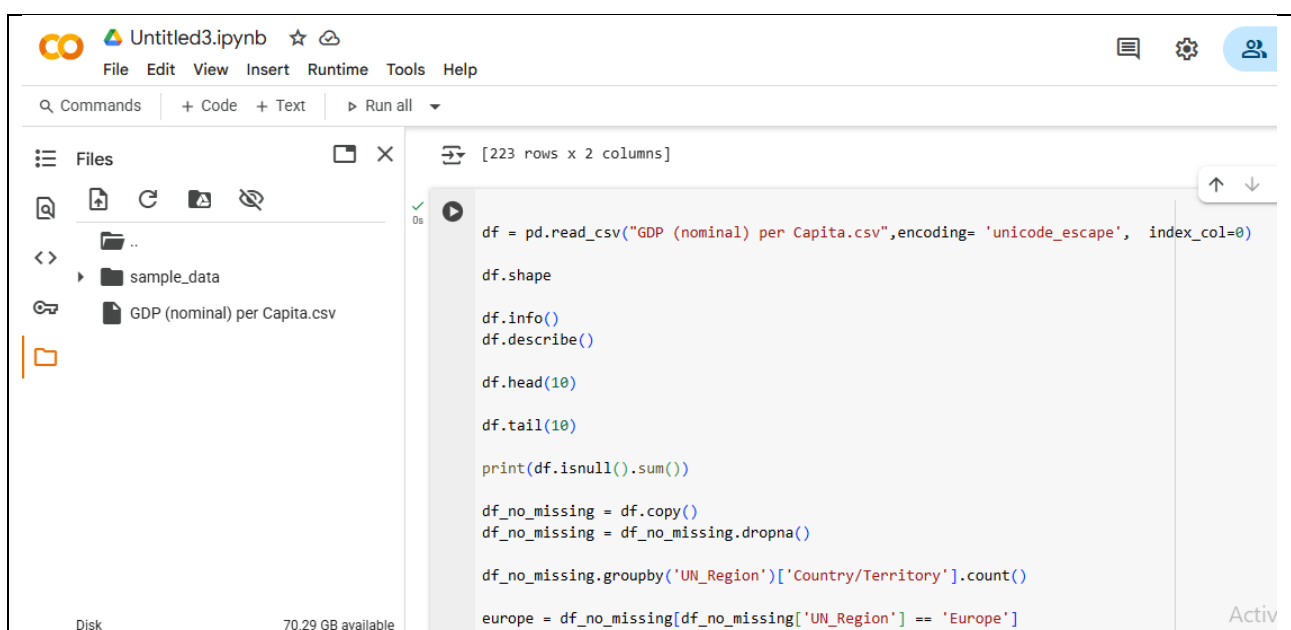


## Day 4: Task 2

Back with 'GDP (nominal) per Capita'. As a group, import and work your way through the Day\_4\_Python\_Activity.ipynb notebook which can be found on the shared Folder. There are questions to answer, but also opportunities to have fun with the data – paste your input and output below.

Once complete, and again as a group, work with some more data and have some fun – there is no set agenda for this section, other than to embed the skills developed this week. Paste your input and output below and upon return we'll discuss progress made.

[Additional data found here.](#)



The screenshot shows a Jupyter Notebook titled 'Untitled3.ipynb'. The left sidebar displays a file explorer with a folder named 'sample\_data' containing a file 'GDP (nominal) per Capita.csv'. The main area shows a code cell with the following Python code:

```
[223 rows x 2 columns]

df = pd.read_csv("GDP (nominal) per Capita.csv",encoding= 'unicode_escape', index_col=0)

df.shape

df.info()
df.describe()

df.head(10)

df.tail(10)

print(df.isnull().sum())

df_no_missing = df.copy()
df_no_missing = df_no_missing.dropna()

df_no_missing.groupby('UN_Region')['Country/Territory'].count()

europe = df_no_missing[df_no_missing['UN_Region'] == 'Europe']
```

The bottom status bar indicates 'Disk' and '70.29 GB available'.

Untitled3.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Files

sample\_dataGDP (nominal) per Capita.csv

```
europe = df_no_missing[df_no_missing['UN_Region'] == 'Europe']
print(europe.head())

average_imf = df_no_missing['IMF_Estimate'].mean()

below_average = europe[europe['IMF_Estimate'] < average_imf]

below_average

uk_gdp = df[df['Country/Territory'] == 'United Kingdom']['IMF_Estimate'].values[0]

print(uk_gdp)

higher_than_uk = europe[europe['IMF_Estimate'] > uk_gdp]

print("\nEuropean countries with higher GDP than the UK:")

print(higher_than_uk[['Country/Territory', 'IMF_Estimate']].sort_values(by='IMF_Estimate', ascending=False))

print("Average IMF GDP estimate per UN region:")

print(df.groupby('UN_Region')['IMF_Estimate'].mean())
```

Disk70.29 GB available

Untitled3.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Files

sample\_dataGDP (nominal) per Capita.csv

```
print(df.groupby('UN_Region')['IMF_Estimate'].mean())

print("Total World Bank GDP estimate per year:")

print(df.groupby('WorldBank_Year')['WorldBank_Estimate'].sum())
```

IMF\_Estimate0
IMF\_Year0
WorldBank\_Estimate0
WorldBank\_Year0
UN\_Estimate0
UN\_Year0
dtype: int64

	Country/Territory	UN_Region	IMF_Estimate	IMF_Year	WorldBank_Estimate
1	Monaco	Europe	0	0	234316
2	Liechtenstein	Europe	0	0	157755
3	Luxembourg	Europe	132372	2023	133590
4	Ireland	Europe	114581	2023	100172
6	Norway	Europe	101103	2023	89154

	WorldBank_Year	UN_Estimate	UN_Year
1	2021	234317	2021
2	2020	169260	2021
3	2021	133745	2021
4	2021	101109	2021
6	2021	101109	2021

Disk70.29 GB available

Variables

Terminal



## Course Notes

It is recommended to take notes from the course, use the space below to do so, or use the revision guide shared with the class:





We have included a range of additional links to further resources and information that you may find useful, these can be found within your revision guide.

### **END OF WORKBOOK**

**Please check through your work thoroughly before submitting and update the table of contents if required.**

**Please send your completed work booklet to your trainer.**

