Just IT

B2W group

Apprenticeships | Training | Recruitment

Data Technician

Name:

Course Date:

Table of contents

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

```python
# FizzBuzz implementation in Python

# Loop through integers 1 to 100
output = []
for i in range(1, 101):
    if i % 3 == 0 and i % 5 == 0:
        output.append("fizzbuzz")
    elif i % 3 == 0:
        output.append("fizz")
    elif i % 5 == 0:
        output.append("buzz")
    else:
        output.append(str(i))

# Display the output as a space-separated string for readability
print(" ".join(output))
```

```
# FizzBuzz implementation in Python

# Loop through integers 1 to 100
output = []
for i in range(1, 101):
    if i % 3 == 0 and i % 5 == 0:
        output.append("fizzbuzz")
    elif i % 3 == 0:
        output.append("fizz")
    elif i % 5 == 0:
        output.append("buzz")
    else:
        output.append(str(i))

# Display the output as a space-separated string for readability
print(" ".join(output))

1 2 fizz 4 buzz fizz 7 8 fizz buzz 11 fizz 13 14 fizzbuzz 16 17 fizz 19 buzz fizz 22 23 fizz buzz 26 fizz 28 29 fizzbuzz
```

Full Result of code

1 2 fizz 4 buzz fizz 7 8 fizz buzz 11 fizz 13 14 fizzbuzz 16 17 fizz 19 buzz fizz 22 23 fizz buzz 26 fizz 28 29 fizzbuzz 31 32 fizz 34 buzz fizz 37 38 fizz buzz 41 fizz 43 44 fizzbuzz 46 47 fizz 49 buzz fizz 52 53 fizz buzz 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 fizz 82 83 fizz buzz 86 fizz 88 89 fizzbuzz 91 92 fizz 94 buzz fizz 97 98 fizz buzz

Day 3: Task 1

Using the 'student.csv' which can be downloaded here, 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."

1.

```
dataframe = pd.read_csv('student (1).csv')

print(dataframe)
```

```
     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
5     6     Alex John   Four    55    male
6     7  My John Rob  Fifth    78    male
7     8        Asruid   Five    85    male
8     9       Tes Qry    Six    78     NaN
9    10      Big John   Four    55  female
10   11        Ronald    Six    89  female
11   12         Recky    Six    94  female
12   13           Kty  Seven    88  female
13   14         Bigy   Seven    88  female
14   15      Tade Row     NaN    88    male
15   16         Gimmy   Four    88    male
16   17         Tumyu    Six    54    male
17   18         Honny   Five    75    male
18   19         Tinny   Nine    18    male
19   20        Jackly   Nine    65  female
20   21   Babby John    Four    69  female
21   22        Reggid  Seven    55  female
22   23         Herod  Eight    79    male
23   24    Tiddy Now   Seven    78    male
24   25     Giff Tow   Seven    88    male
25   26       Crelea   Seven    79    male
26   27          NaN  Three    81     NaN
27   28    Rojj Base   Seven    86  female
28   29  Tess Played   Seven    55    male
29   30    Reppy Red    Six    79  female
30   31  Marry Toeey    Four    88    male
31   32    Binn Rott  Seven    90  female
32   33    Kenn Rein    Six    96  female
33   34     Gain Toe   Seven    69    male
34   35   Rows Noump    Six    88  female
```

2.

```
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('student.csv')

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

	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

Next steps: Generate code with df | New interactive sheet

3.

```
[9]
✓ 0s
# Information about the Dataframe
df.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
```

4.

	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

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'."

1.

```
[56]
✓ 0s
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('student.csv')

df[['name', 'mark']]
```

	name	mark
0	John Deo	75
1	Max Ruin	85
2	Arnold	55
3	Krish Star	60
4	John Mike	60
5	Alex John	55
6	My John Rob	78
7	Asruid	85
8	Tes Qry	78
9	Big John	55
10	Ronald	89
11	Recky	94
12	Kty	88
13	Bigy	88
14	Tade Row	88
15	Gimmy	88
16	Tumyu	54
17	Honny	75

2.

```
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('student.csv')

df[['name']]
```

	name
0	John Deo
1	Max Ruin
2	Arnold
3	Krish Star
4	John Mike
5	Alex John
6	My John Rob
7	Asruid
8	Tes Qry
9	Big John
10	Ronald
11	Recky
12	Kty
13	Bigy

3.

```
[57]
✓ 0s
```

```python
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('student.csv')

df.head(3)
```

	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

Next steps: Generate code with df | New interactive sheet

4.

```python
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('student.csv')

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."

1.

```
df[df['mark']>= 60]
```

	id	name	class	mark	gender	passed
0	True	John Deo	Four	75	female	True
1	True	Max Ruin	Three	85	male	True
3	True	Krish Star	Four	60	female	True
4	True	John Mike	Four	60	female	True
6	True	My John Rob	Fifth	78	male	True
7	True	Asruid	Five	85	male	True
8	True	Tes Qry	Six	78	NaN	True
10	True	Ronald	Six	89	female	True
11	True	Recky	Six	94	female	True
12	True	Kty	Seven	88	female	True
13	True	Bigy	Seven	88	female	True
14	True	Tade Row	NaN	88	male	True
15	True	Gimmy	Four	88	male	True
17	True	Honny	Five	75	male	True
19	True	Jackly	Nine	65	female	True
20	True	Babby John	Four	69	female	True
22	True	Herod	Eight	79	male	True
23	True	Tiddy Now	Seven	78	male	True
24	True	Giff Tow	Seven	88	male	True
25	True	Crelea	Seven	79	male	True
26	True	NaN	Three	81	NaN	True
27	True	Rojj Base	Seven	86	female	True
29	True	Reppy Red	Six	79	female	True

2.

```
df.rename(columns={'mark': 'score', 'name': 'student_name'}, inplace=True)
print(df)
```

```
    id student_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
5    6    Alex John   Four     55    male
6    7  My John Rob  Fifth     78    male
7    8       Asruid   Five     85    male
8    9      Tes Qry    Six     78     NaN
9   10     Big John   Four     55  female
10  11       Ronald    Six     89  female
11  12        Recky    Six     94  female
12  13          Kty  Seven     88  female
13  14        Bigy   Seven     88  female
14  15    Tade Row     NaN     88    male
15  16       Gimmy   Four     88    male
16  17       Tumyu    Six     54    male
17  18       Honny   Five     75    male
18  19       Tinny   Nine     18    male
19  20      Jackly   Nine     65  female
20  21   Babby John   Four     69  female
21  22      Reggid  Seven     55  female
22  23       Herod  Eight     79    male
23  24   Tiddy Now  Seven     78    male
24  25    Giff Tow  Seven     88    male
25  26      Crelea  Seven     79    male
26  27         NaN  Three     81     NaN
27  28   Rojj Base  Seven     86  female
28  29  Tess Played  Seven     55    male
29  30   Reppy Red    Six     79  female
30  31  Marry Toeey   Four     88    male
31  32    Binn Rott  Seven     90  female
32  33    Kenn Rein    Six     96  female
33  34     Gain Toe  Seven     69    male
34  35   Rows Noump    Six     88  female
```

3.

```python
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('student.csv')

df[df['mark']>= 60]
df['passed'] = df['mark'] >= 60
print(df)
```

	id	name	class	mark	gender	passed
0	1	John Deo	Four	75	female	True
1	2	Max Ruin	Three	85	male	True
2	3	Arnold	Three	55	male	False
3	4	Krish Star	Four	60	female	True
4	5	John Mike	Four	60	female	True
5	6	Alex John	Four	55	male	False
6	7	My John Rob	Fifth	78	male	True
7	8	Asruid	Five	85	male	True
8	9	Tes Qry	Six	78	NaN	True
9	10	Big John	Four	55	female	False
10	11	Ronald	Six	89	female	True
11	12	Recky	Six	94	female	True
12	13	Kty	Seven	88	female	True
13	14	Bigy	Seven	88	female	True
14	15	Tade Row	NaN	88	male	True
15	16	Gimmy	Four	88	male	True
16	17	Tumyu	Six	54	male	False
17	18	Honny	Five	75	male	True
18	19	Tinny	Nine	18	male	False
19	20	Jackly	Nine	65	female	True
20	21	Babby John	Four	69	female	True
21	22	Reggid	Seven	55	female	False
22	23	Herod	Eight	79	male	True
23	24	Tiddy Now	Seven	78	male	True
24	25	Giff Tow	Seven	88	male	True
25	26	Crelea	Seven	79	male	True
26	27	NaN	Three	81	NaN	True
27	28	Rojj Base	Seven	86	female	True
28	29	Tess Played	Seven	55	male	False
29	30	Reppy Red	Six	79	female	True
30	31	Marry Toeey	Four	88	male	True
31	32	Binn Rott	Seven	90	female	True
32	33	Kenn Rein	Six	96	female	True
33	34	Gain Toe	Seven	69	male	True
34	35	Rows Noump	Six	88	female	True

```python
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('student.csv')

df[df['mark']>= 60]
df['passed'] = df['mark'] >= 60
df.drop(columns=['passed'], inplace=True)
print(df)
```

	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
5	6	Alex John	Four	55	male
6	7	My John Rob	Fifth	78	male
7	8	Asruid	Five	85	male
8	9	Tes Qry	Six	78	NaN
9	10	Big John	Four	55	female
10	11	Ronald	Six	89	female
11	12	Recky	Six	94	female
12	13	Kty	Seven	88	female
13	14	Bigy	Seven	88	female
14	15	Tade Row	NaN	88	male
15	16	Gimmy	Four	88	male
16	17	Tumyu	Six	54	male
17	18	Honny	Five	75	male
18	19	Tinny	Nine	18	male
19	20	Jackly	Nine	65	female
20	21	Babby John	Four	69	female
21	22	Reggid	Seven	55	female
22	23	Herod	Eight	79	male
23	24	Tiddy Now	Seven	78	male
24	25	Giff Tow	Seven	88	male
25	26	Crelea	Seven	79	male
26	27	NaN	Three	81	NaN
27	28	Rojj Base	Seven	86	female
28	29	Tess Played	Seven	55	male
29	30	Reppy Red	Six	79	female
30	31	Marry Toeey	Four	88	male
31	32	Binn Rott	Seven	90	female
32	33	Kenn Rein	Six	96	female
33	34	Gain Toe	Seven	69	male
34	35	Rows Noump	Six	88	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."

1.

```python
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('student.csv')

mean_marks_by_class = df.groupby('class')['mark'].mean()

# Display the result
print(mean_marks_by_class)
```

```
class
Eight    79.000000
Fifth    78.000000
Five     80.000000
Four     68.750000
Nine     41.500000
Seven    77.600000
Six      82.571429
Three    73.666667
Name: mark, dtype: float64
```

2.

```python
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('student.csv')

df.groupby('class').size().reset_index(name='student_count')
```

	class	student_count
0	Eight	1
1	Fifth	1
2	Five	2
3	Four	8
4	Nine	2
5	Seven	10
6	Six	7
7	Three	3

3.

```
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('student.csv')

average_mark_by_gender = df.groupby('gender', as_index=False)['mark'].mean()

# Display the result
print(average_mark_by_gender)
```

```
   gender       mark
0  female  77.312500
1    male  71.588235
```

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 >= 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."

1.

```
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('student.csv')

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

# Display the pivot table
print(pivot)
```

```
gender  female  male
class
Eight      NaN  79.0
Fifth      NaN  78.0
Five       NaN  80.0
Four      63.8  77.0
Nine      65.0  18.0
Seven     81.4  73.8
Six       89.2  54.0
Three      NaN  70.0
```

2.

```
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('student.csv')

# Define a function to assign grades based on mark
def assign_grade(mark):
    if mark >= 85:
        return 'A'
    elif mark >= 70:
        return 'B'
    elif mark >= 60:
        return 'C'
    else:
        return 'D'

# Apply the function to create the 'grade' column
df['grade'] = df['mark'].apply(assign_grade)

print(df)
```

```
    id          name  class  mark  gender grade
0    1      John Deo   Four    75  female     B
1    2      Max Ruin  Three    85    male     A
2    3        Arnold  Three    55    male     D
3    4   Krish Star    Four    60  female     C
4    5     John Mike   Four    60  female     C
5    6     Alex John   Four    55    male     D
6    7  My John Rob  Fifth    78    male     B
7    8        Asruid   Five    85    male     A
8    9       Tes Qry    Six    78     NaN     B
9   10      Big John   Four    55  female     D
10  11        Ronald    Six    89  female     A
11  12         Recky    Six    94  female     A
12  13           Kty  Seven    88  female     A
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
```

3.

```python
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('student.csv')

df.sort_values(by='mark', ascending=False, inplace=True)

print(df)
```

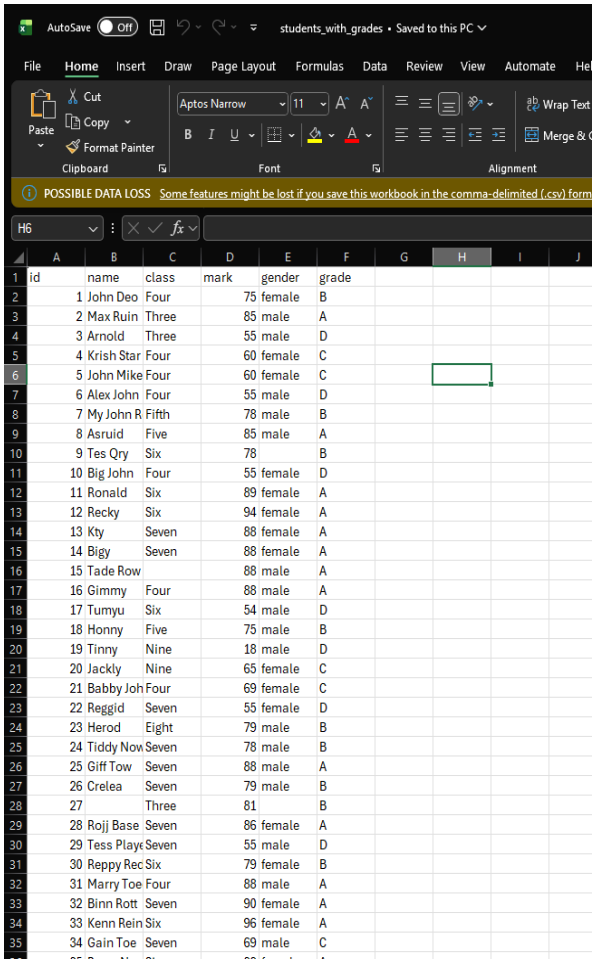
```
    id          name  class  mark  gender
32  33    Kenn Rein    Six    96  female
11  12        Recky    Six    94  female
31  32    Binn Rott  Seven    90  female
10  11       Ronald    Six    89  female
30  31  Marry Toeey   Four    88    male
34  35   Rows Noump    Six    88  female
24  25     Giff Tow  Seven    88    male
14  15     Tade Row    NaN    88    male
15  16        Gimmy   Four    88    male
12  13          Kty  Seven    88  female
13  14         Bigy  Seven    88  female
27  28    Rojj Base  Seven    86  female
7    8       Asruid   Five    85    male
1    2     Max Ruin  Three    85    male
26  27          NaN  Three    81     NaN
29  30    Reppy Red    Six    79  female
25  26       Crelea  Seven    79    male
22  23        Herod  Eight    79    male
6    7  My John Rob  Fifth    78    male
23  24    Tiddy Now  Seven    78    male
8    9      Tes Qry    Six    78     NaN
17  18        Honny   Five    75    male
0    1     John Deo   Four    75  female
33  34     Gain Toe  Seven    69    male
20  21   Babby John   Four    69  female
19  20       Jackly   Nine    65  female
3    4   Krish Star   Four    60  female
4    5    John Mike   Four    60  female
2    3       Arnold  Three    55    male
5    6    Alex John   Four    55    male
9   10     Big John   Four    55  female
21  22       Reggid  Seven    55  female
28  29  Tess Played  Seven    55    male
16  17        Tumyu    Six    54    male
18  19        Tinny   Nine    18    male
```

Exercise 6: Exporting Data

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

```python
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('student.csv')

# Define a function to assign grades based on mark
def assign_grade(mark):
    if mark >= 85:
        return 'A'
    elif mark >= 70:
        return 'B'
    elif mark >= 60:
        return 'C'
    else:
        return 'D'

# Apply the function to create the 'grade' column
df['grade'] = df['mark'].apply(assign_grade)

print(df)

df.to_csv('students_with_grades.csv', index=False)
```

```
    id         name  class  mark  gender grade
0    1     John Deo   Four    75  female     B
1    2     Max Ruin  Three    85    male     A
2    3       Arnold  Three    55    male     D
3    4   Krish Star   Four    60  female     C
4    5    John Mike   Four    60  female     C
5    6    Alex John   Four    55    male     D
6    7  My John Rob  Fifth    78    male     B
7    8       Asruid   Five    85    male     A
8    9      Tes Qry    Six    78     NaN     B
9   10    Big John   Four    55  female     D
10  11      Ronald    Six    89  female     A
11  12       Recky    Six    94  female     A
12  13         Kty  Seven    88  female     A
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
```



	A	B	C	D	E	F
1	id	name	class	mark	gender	grade
2	1	John Deo	Four	75	female	B
3	2	Max Ruin	Three	85	male	A
4	3	Arnold	Three	55	male	D
5	4	Krish Star	Four	60	female	C
6	5	John Mike	Four	60	female	C
7	6	Alex John	Four	55	male	D
8	7	My John R	Fifth	78	male	B
9	8	Asruid	Five	85	male	A
10	9	Tes Qry	Six	78		B
11	10	Big John	Four	55	female	D
12	11	Ronald	Six	89	female	A
13	12	Recky	Six	94	female	A
14	13	Kty	Seven	88	female	A
15	14	Bigy	Seven	88	female	A
16	15	Tade Row		88	male	A
17	16	Gimmy	Four	88	male	A
18	17	Tumyu	Six	54	male	D
19	18	Honny	Five	75	male	B
20	19	Tinny	Nine	18	male	D
21	20	Jackly	Nine	65	female	C
22	21	Babby Joh	Four	69	female	C
23	22	Reggid	Seven	55	female	D
24	23	Herod	Eight	79	male	B
25	24	Tiddy Now	Seven	78	male	B
26	25	Giff Tow	Seven	88	male	A
27	26	Crelea	Seven	79	male	B
28	27		Three	81		B
29	28	Roji Base	Seven	86	female	A
30	29	Tess Playe	Seven	55	male	D
31	30	Reppy Rec	Six	79	female	B
32	31	Marry Toe	Four	88	male	A
33	32	Binn Rott	Seven	90	female	A
34	33	Kenn Rein	Six	96	female	A
35	34	Gain Toe	Seven	69	male	C

Exercise 7: If finished early try visualising the results

Day 4: Task 1

Using the 'GDP (nominal) per Capita.csv' which can be downloaded here, 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

1.

```
[2]
✓ 0s
```

```
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('GDP (nominal) per Capita.csv')

print(df)
```

```
     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
..          ...               ...       ...           ...       ...
218         219            Malawi    Africa           496      2023
219         220       South Sudan    Africa           467      2023
220         221      Sierra Leone    Africa           415      2023
221         222       Afghanistan      Asia           611      2020
222         223           Burundi    Africa           249      2023

     WorldBank_Estimate  WorldBank_Year  UN_Estimate  UN_Year
0                234316            2021       234317     2021
1                157755            2020       169260     2021
2                133590            2021       133745     2021
3                100172            2021       101109     2021
4                114090            2021       112653     2021
..                  ...             ...          ...      ...
218                 635            2021          613     2021
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]
```

2.

```
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('GDP (nominal) per Capita.csv')


df.head(10)
```

	Unnamed: 0	Country/Territory	UN_Region	IMF_Estimate	IMF_Year	WorldBank_Estimate	WorldBank_Year	UN_Estimate	UN_Year
0	1	Monaco	Europe	0	0	234316	2021	234317	2021
1	2	Liechtenstein	Europe	0	0	157755	2020	169260	2021
2	3	Luxembourg	Europe	132372	2023	133590	2021	133745	2021
3	4	Ireland	Europe	114581	2023	100172	2021	101109	2021
4	5	Bermuda	Americas	0	0	114090	2021	112653	2021
5	6	Norway	Europe	101103	2023	89154	2021	89242	2021
6	7	Switzerland	Europe	98767	2023	91992	2021	93525	2021
7	8	Singapore	Asia	91100	2023	72794	2021	66822	2021
8	9	Isle of Man	Europe	0	0	87158	2019	0	0
9	10	Cayman Islands	Americas	0	0	86569	2021	85250	2021

Next steps: Generate code with df | New interactive sheet

3.

```
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('GDP (nominal) per Capita.csv')

df.tail(5)
```

	Unnamed: 0	Country/Territory	UN_Region	IMF_Estimate	IMF_Year	WorldBank_Estimate	WorldBank_Year	UN_Estimate	UN_Year
218	219	Malawi	Africa	496	2023	635	2021	613	2021
219	220	South Sudan	Africa	467	2023	1072	2015	400	2021
220	221	Sierra Leone	Africa	415	2023	480	2021	505	2021
221	222	Afghanistan	Asia	611	2020	369	2021	373	2021
222	223	Burundi	Africa	249	2023	222	2021	311	2021

4.

```
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('GDP (nominal) per Capita.csv')

df[['Country/Territory', 'UN_Region']]
```

	Country/Territory	UN_Region
0	Monaco	Europe
1	Liechtenstein	Europe
2	Luxembourg	Europe
3	Ireland	Europe
4	Bermuda	Americas
...	...	...
218	Malawi	Africa
219	South Sudan	Africa
220	Sierra Leone	Africa
221	Afghanistan	Asia
222	Burundi	Africa

223 rows × 2 columns

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 here. 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.

```
# number of countries per region
```

```
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('GDP (nominal) per Capita.csv')

countries_per_region = df.groupby('UN_Region').size()

print(df)
```

```
     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
..          ...               ...       ...           ...       ...
218         219            Malawi    Africa           496      2023
219         220       South Sudan    Africa           467      2023
220         221      Sierra Leone    Africa           415      2023
221         222       Afghanistan      Asia           611      2020
222         223           Burundi    Africa           249      2023

     WorldBank_Estimate  WorldBank_Year  UN_Estimate UN_Year
0                234316            2021       234317    2021
1                157755            2020       169260    2021
2                133590            2021       133745    2021
3                100172            2021       101109    2021
4                114090            2021       112653    2021
..                  ...             ...          ...     ...
218                 635            2021          613    2021
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]
```

```
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('GDP (nominal) per Capita.csv')

countries_per_region = df.groupby('UN_Region').size()

print(countries_per_region)
```

```
UN_Region
Africa      55
Americas    48
Asia        51
Europe      48
Oceania     20
World        1
dtype: int64
```

```
#What is European Union[n 1]?
```

```
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('GDP (nominal) per Capita.csv')

# List of EU member countries
eu_countries = [
    "Austria", "Belgium", "Bulgaria", "Croatia", "Cyprus", "Czech Republic",
    "Denmark", "Estonia", "Finland", "France", "Germany", "Greece", "Hungary",
    "Ireland", "Italy", "Latvia", "Lithuania", "Luxembourg", "Malta",
    "Netherlands", "Poland", "Portugal", "Romania", "Slovakia", "Slovenia",
    "Spain", "Sweden"
]

# Display the countries
for country in eu_countries:
    print(country)
```

```
Austria
Belgium
Bulgaria
Croatia
Cyprus
Czech Republic
Denmark
Estonia
Finland
France
Germany
Greece
Hungary
Ireland
Italy
Latvia
Lithuania
Luxembourg
Malta
Netherlands
Poland
Portugal
Romania
Slovakia
Slovenia
Spain
Sweden
```

```
# Countries in Europe below avarege
```

```
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('GDP (nominal) per Capita.csv')

europe_df = df[df['UN_Region'] == 'Europe'].copy()

print(europe_df)
```

```
    40          41                   Italy    Europe          36812       2023
    50          51                Slovenia    Europe          32214       2023
    51          52          Czech Republic    Europe          31368       2023
    52          53                   Spain    Europe          31223       2023
    53          54                 Estonia    Europe          31209       2023
    56          57               Lithuania    Europe          28094       2023
    58          59                Portugal    Europe          26012       2023
    59          60                  Latvia    Europe          25136       2023
    61          62                Slovakia    Europe          23457       2023
    62          63                  Greece    Europe          22595       2023
    69          70                 Croatia    Europe          20537       2023
    71          72                  Poland    Europe          19912       2023
    74          75                 Hungary    Europe          19385       2023
    77          78                 Romania    Europe          18530       2023
    86          87                Bulgaria    Europe          14893       2023
    89          90                  Russia    Europe          14403       2023
   102         103              Montenegro    Europe          11289       2023
   105         106                  Serbia    Europe          10849       2023
   111         112  Bosnia and Herzegovina    Europe           8223       2023
   114         115                 Belarus    Europe           7944       2023
   117         118          North Macedonia    Europe           7384       2023
   119         120                 Albania    Europe           7058       2023
   126         127                 Moldova    Europe           6342       2023
   132         133                  Kosovo    Europe           5641       2023
   142         143                 Ukraine    Europe           4654       2023

     WorldBank_Estimate  WorldBank_Year  UN_Estimate UN_Year
0                234316            2021       234317    2021
1                157755            2020       169260    2021
2                133590            2021       133745    2021
3                100172            2021       101109    2021
5                 89154            2021        89242    2021
6                 91992            2021        93525    2021
8                 87158            2019            0       0
12                68728            2021        69133    2021
```

```python
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('GDP (nominal) per Capita.csv')

# Using the most recent GDP estimates

def get_latest_gdp(row):
    return max(row['IMF_Estimate'], row['WorldBank_Estimate'], row['UN_Estimate'])

europe_df['Latest_GDP_per_Capita'] = europe_df.apply(get_latest_gdp, axis=1)

print(europe_df)
```

```
    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
5            6                  Norway    Europe        101103      2023
6            7             Switzerland    Europe         98767      2023
8            9             Isle of Man    Europe             0         0
12          13                 Iceland    Europe         75180      2023
13          14        Channel Islands     Europe             0         0
14          15          Faroe Islands     Europe             0         0
15          16                Denmark    Europe         68827      2023
17          18            Netherlands    Europe         61098      2023
19          20                Austria    Europe         56802      2023
21          22                 Sweden    Europe         55395      2023
22          23                Finland    Europe         54351      2023
23          24                Belgium    Europe         53377      2023
24          25             San Marino    Europe         52949      2023
27          28                Germany    Europe         51383      2023
32          33         United Kingdom    Europe         46371      2023
33          34                 France    Europe         44408      2023
34          35                Andorra    Europe         44387      2023
35          36     European Union[n 1]    Europe         39940      2023
39          40                  Malta    Europe         36989      2023
40          41                  Italy    Europe         36812      2023
50          51               Slovenia    Europe         32214      2023
51          52         Czech Republic    Europe         31368      2023
52          53                  Spain    Europe         31223      2023
53          54                Estonia    Europe         31209      2023
```

```python
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('GDP (nominal) per Capita.csv')

# Calculating the average GDP per capita
average_gdp = europe_df['Latest_GDP_per_Capita'].mean()
print(f"Average GDP per capita in Europe: ${average_gdp:,.0f}\n")
```

```
Average GDP per capita in Europe: $47,828
```

```python
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('GDP (nominal) per Capita.csv')

# Filtering countries below the average
below_avg_df = europe_df[europe_df['Latest_GDP_per_Capita'] < average_gdp]

print(below_avg_df)
```

```
     Unnamed: 0       Country/Territory UN_Region  IMF_Estimate  IMF_Year  \
32           33          United Kingdom    Europe         46371      2023
33           34                  France    Europe         44408      2023
34           35                 Andorra    Europe         44387      2023
35           36     European Union[n 1]    Europe         39940      2023
39           40                   Malta    Europe         36989      2023
40           41                   Italy    Europe         36812      2023
50           51                Slovenia    Europe         32214      2023
51           52          Czech Republic    Europe         31368      2023
52           53                   Spain    Europe         31223      2023
53           54                 Estonia    Europe         31209      2023
56           57               Lithuania    Europe         28094      2023
58           59                Portugal    Europe         26012      2023
59           60                  Latvia    Europe         25136      2023
61           62                Slovakia    Europe         23457      2023
62           63                  Greece    Europe         22595      2023
69           70                 Croatia    Europe         20537      2023
71           72                  Poland    Europe         19912      2023
74           75                 Hungary    Europe         19385      2023
77           78                 Romania    Europe         18530      2023
86           87                Bulgaria    Europe         14893      2023
89           90                  Russia    Europe         14403      2023
102         103              Montenegro    Europe         11289      2023
105         106                  Serbia    Europe         10849      2023
111         112  Bosnia and Herzegovina    Europe          8223      2023
114         115                 Belarus    Europe          7944      2023
117         118          North Macedonia    Europe          7384      2023
119         120                 Albania    Europe          7058      2023
126         127                 Moldova    Europe          6342      2023
132         133                  Kosovo    Europe          5641      2023
142         143                 Ukraine    Europe          4654      2023

     WorldBank_Estimate  WorldBank_Year  UN_Estimate UN_Year  \
32                46510            2021        46542    2021
33                43659            2021        44229    2021
34                42137            2021        42066    2021
35                38411            2021        31875    2021
```

```
## Which countries in Europe has higher GDP than UK?
```

```
import pandas as pd

# Load your DataFrame (example using a CSV file)
df = pd.read_csv('GDP (nominal) per Capita.csv')

# Filter for European countries
europe_df = df[df['UN_Region'] == 'Europe'].copy()

# Use the most recent available GDP estimate
def get_latest_gdp(row):
    return max(row['IMF_Estimate'], row['WorldBank_Estimate'], row['UN_Estimate'])

    # Apply the function to get the latest GDP per capita
europe_df['Latest_GDP_per_Capita'] = europe_df.apply(get_latest_gdp, axis=1)

# UK's GDP per capita
uk_gdp = europe_df[europe_df['Country/Territory'] == 'United Kingdom']['Latest_GDP_per_Capita'].values[0]

# Filter countries with GDP higher than the UK
higher_gdp_df = europe_df[europe_df['Latest_GDP_per_Capita'] > uk_gdp]

# Display results
print(f"UK GDP per capita: ${uk_gdp:,.0f}\n")
print("European countries with higher GDP per capita than the UK:\n")
print(higher_gdp_df[['Country/Territory', 'Latest_GDP_per_Capita']].sort_values(by='Latest_GDP_per_Capita', ascending=False).to_string(index=False))
```

```
UK GDP per capita: $46,542

European countries with higher GDP per capita than the UK:

Country/Territory  Latest_GDP_per_Capita
           Monaco                 234317
    Liechtenstein                 169260
       Luxembourg                 133745
          Ireland                 114581
           Norway                 101103
      Switzerland                  98767
      Isle of Man                  87158
          Iceland                  75180
  Channel Islands                  75153
    Faroe Islands                  69010
          Denmark                  68827
      Netherlands                  61098
           Sweden                  61029
```

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.