Just IT

B2W group

Apprenticeships | Training | Recruitment

Data Technician

Name:

Course Date:

Table of contents

Day 2: Task 1

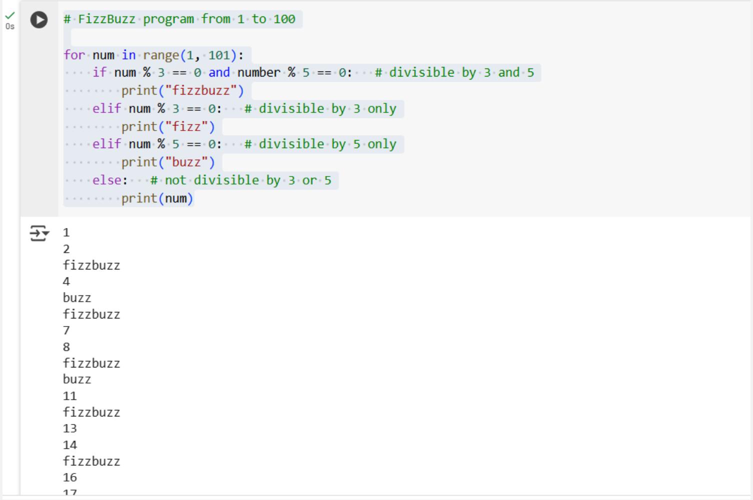
It is a common software development interview question to create the below with a certain programming language. Create the below using Python syntax, test it and past the completed syntax and output below.

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

```
# FizzBuzz program from 1 to 100


for num in range(1, 101):
    if num % 3 == 0 and number % 5 == 0:   # divisible by 3 and 5
        print("fizzbuzz")
    elif num % 3 == 0:   # divisible by 3 only
        print("fizz")
    elif num % 5 == 0:   # divisible by 5 only
        print("buzz")
    else:   # not divisible by 3 or 5
        print(num)
```



```
# FizzBuzz program from 1 to 100

for num in range(1, 101):
    if num % 3 == 0 and number % 5 == 0:   # divisible by 3 and 5
        print("fizzbuzz")
    elif num % 3 == 0:   # divisible by 3 only
        print("fizz")
    elif num % 5 == 0:   # divisible by 5 only
        print("buzz")
    else:   # not divisible by 3 or 5
        print(num)
```

```
1
2
fizzbuzz
4
buzz
fizzbuzz
7
8
fizzbuzz
buzz
11
fizzbuzz
13
14
fizzbuzz
16
17
```

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."

Q:1 df=pd.read_csv(‘filename.csv’)
Q:2 df.head(); #head function by default shows 5 rows only
Q3: df.info()
Q4: df.describe()

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'."

Name_col=df.loc[:,’name’]
Name_col=df.loc[:,[‘name’, ‘mark’]]

Name_col=df.loc[0:3]

```
class_four = df[df['class'] == 'Four']
```

Exercise 3: Data Manipulation

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

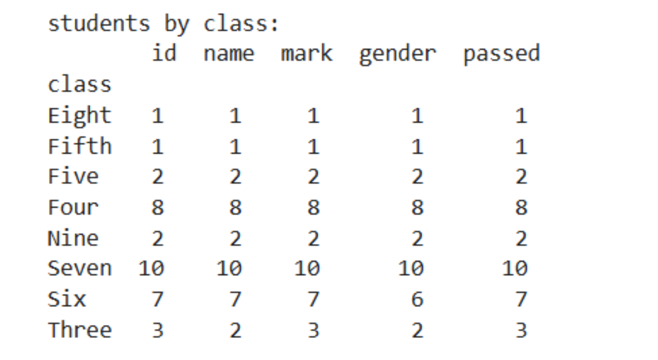
2. Question: "Write the code to rename the 'mark' column to 'score'."
3. Question: "Write the code to drop the 'passed' column."

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


print(df['passed'])


df_renamed.rename(columns={'mark': 'score'})

df = df.drop('passed', axis=1)
```

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."

```
Q1: class_mean = df.groupby('class')['mark'].mean()
print(class_mean)

Q2: # I did the following count function but i did not get exact result. My code
was


count_of_std=df.groupby('class').count()
print("students by class:", count_of_std) # it displays all the dframe and not
exact count of students.
```



```
students by class:
        id  name  mark  gender  passed
class
Eight    1     1     1       1       1
Fifth    1     1     1       1       1
Five     2     2     2       2       2
Four     8     8     8       8       8
Nine     2     2     2       2       2
Seven   10    10    10      10      10
Six      7     7     7       6       7
Three    3     2     3       2       3
```

```
# then i asked chatgpt in learning version and it explains size() function i
think it makes more sense according to the question. So the new code is:
```

```
count_of_std = df.groupby('class').size()

print("Students by class:\n", count_of_std)
```

```
Students by class:
 class
Eight     1
Fifth     1
Five      2
Four      8
Nine      2
Seven    10
Six       7
Three     3
dtype: int64
```

Q3:
```
Average_by_gender = df.groupby('gender')['mark'].mean()
print("Students by class:\n")
display(Average_by_gender)
```

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."

Q1:
```
pivot_table = df.pivot_table(values="mark", index="class", columns="gender", 
aggfunc="mean")
print(pivot_table)
```

```
Q2: df['grade'] = df['mark'].apply(
    lambda x: 'A' if x >= 85 else
              'B' if x >= 70 else
              'C' if x >= 60 else
              'D'
)

print(df)
```

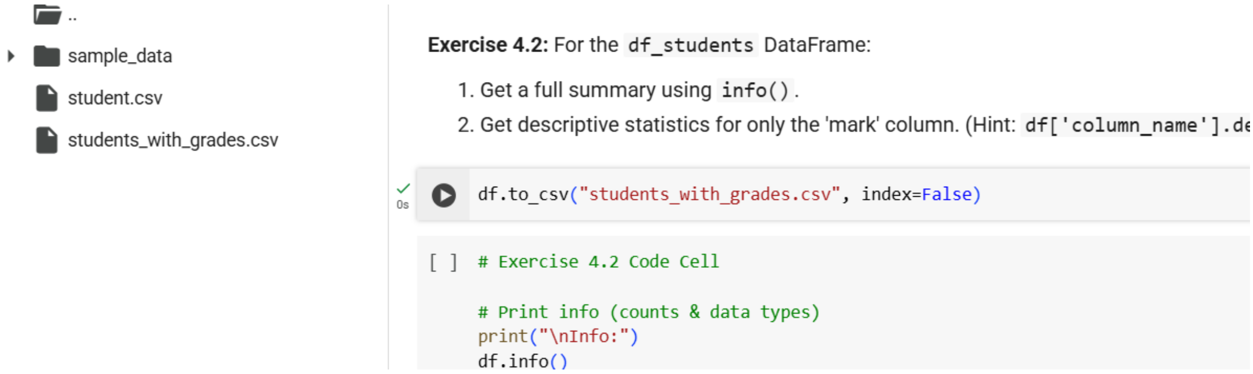
#Q3:

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

Exercise 6: Exporting Data

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

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



..

sample_data

student.csv

students_with_grades.csv

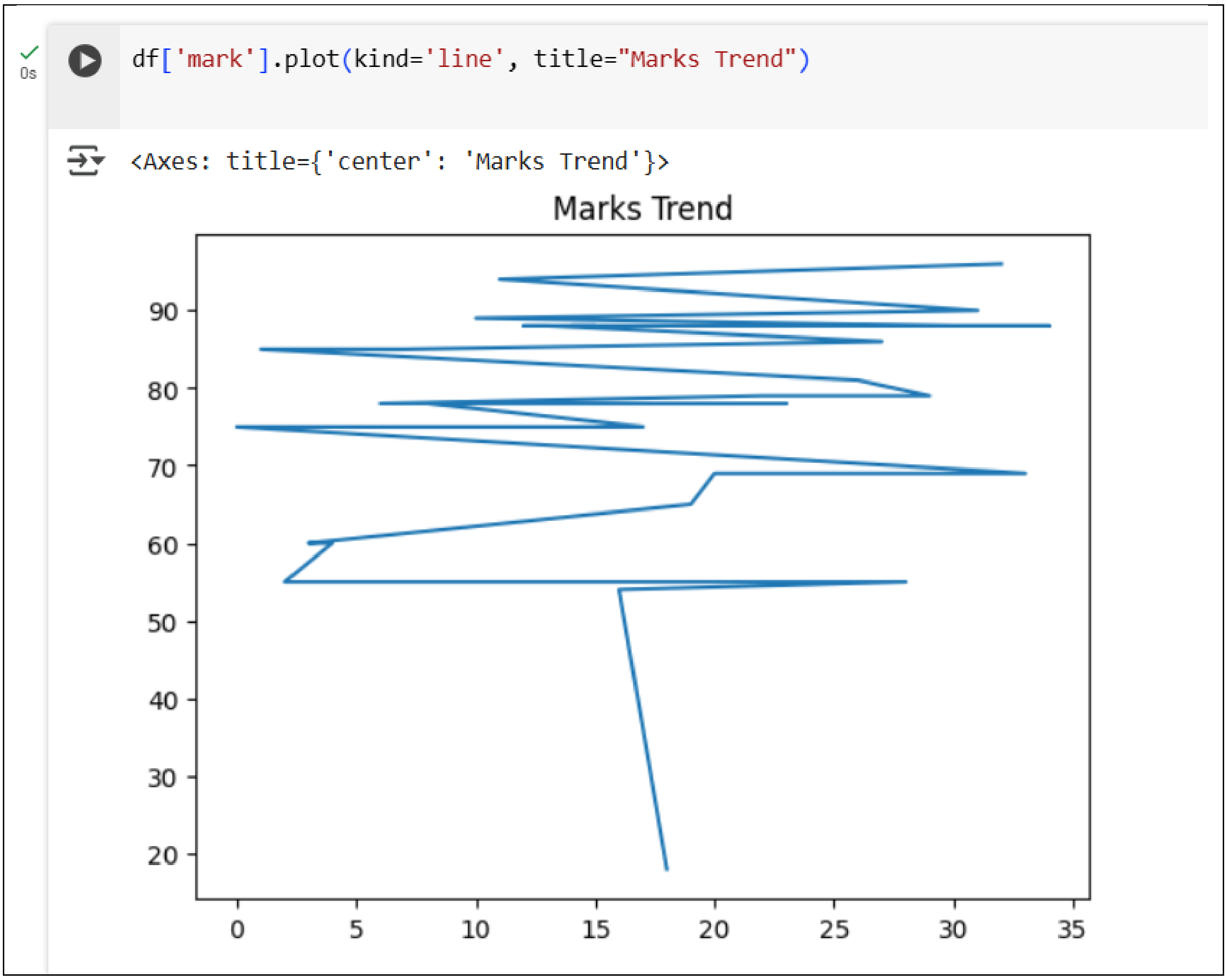
Exercise 4.2: For the `df_students` DataFrame:

1. Get a full summary using `info()`.
2. Get descriptive statistics for only the 'mark' column. (Hint: `df['column_name'].de`

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

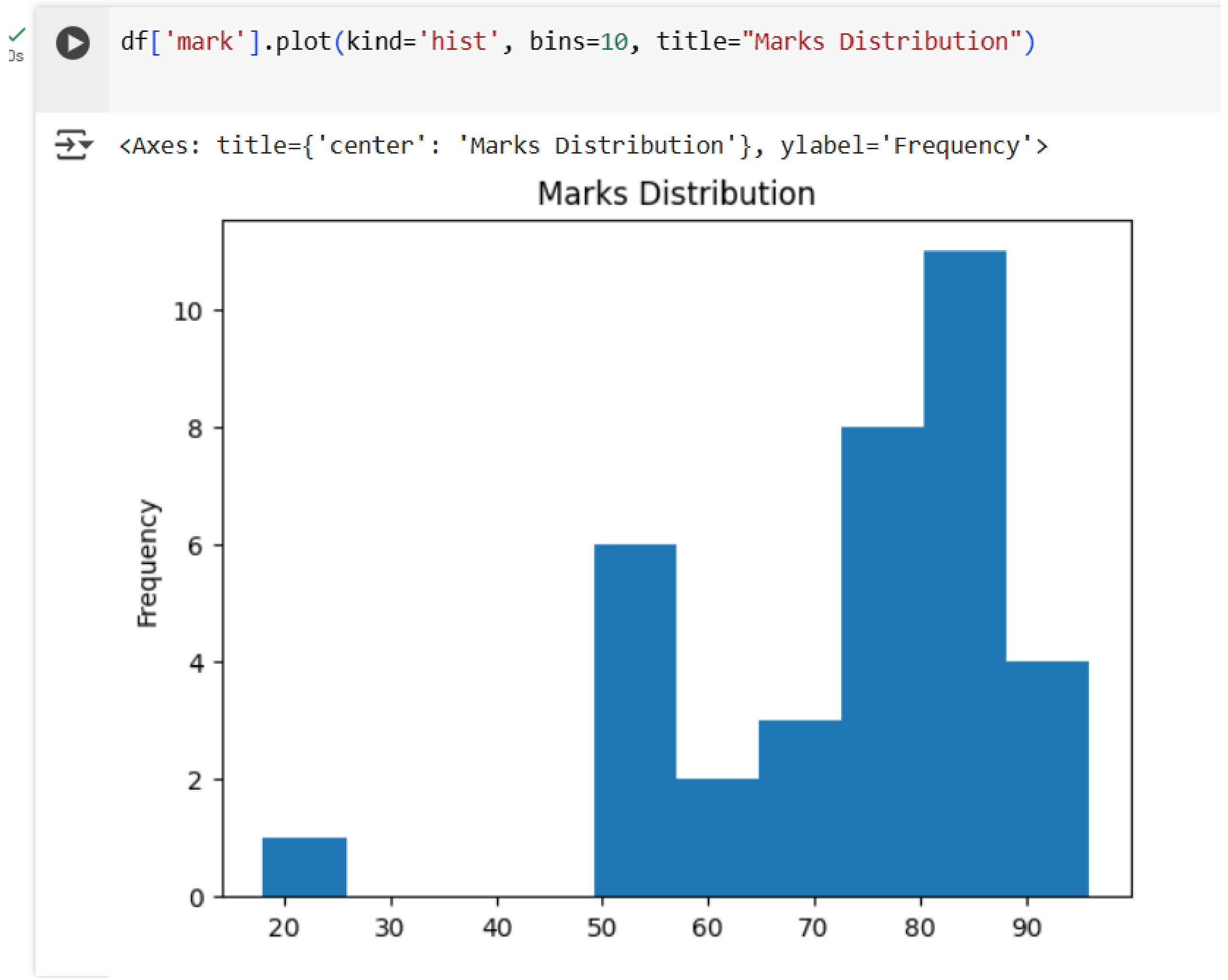
```
# Exercise 4.2 Code Cell

# Print info (counts & data types)
print("\nInfo:")
df.info()
```

Exercise 7: If finished early try visualising the results



```
df['mark'].plot(kind='line', title="Marks Trend")
```

```
<Axes: title={'center': 'Marks Trend'}>
```



```
df['mark'].plot(kind='hist', bins=10, title="Marks Distribution")
```

```
<Axes: title={'center': 'Marks Distribution'}, ylabel='Frequency'>
```

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 Jyputer notebook
- Print the first 10 rows

- Print the last 5 rows
- Print ‘Country/Territory’ and ‘UN_Region’ columns

```
two_cols  = df.loc[:, [' Country/Territory', ' UN_Region']]
```

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.

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

eu_count = (df['Country/Territory'] == "European Union[n 1]").sum()

print("European Union occurs:", eu_count, "times")

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




# Countries in Europe below avarege

uk_gdp = df.loc[df["Country/Territory"] == "United Kingdom",
"UN_Estimate"].values[0]
print("UK GDP per capita (UN):", uk_gdp)


europe_df = df['Country/Territory'].loc[(df['UN_Region'] == 'Europe') &
(df['UN_Estimate'] > uk_gdp)]



#Display results
print(europe_df)


#uk_gdp = df.loc[df['Country/Territory'] == 'United Kingdom',
'WorldBank_Estimate'].values[0]
#print(f"Countries in European
Union:\n{df['Country/Territory'].loc[(df['UN_Region'] == 'Europe') &
(df['WorldBank_Estimate'] > uk_gdp)]}")
```

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.