Jone Paesmans
Özgün Ünver

Developing our own small Python program following the waterfall model

Requirements (problem description)

At Vrije Universiteit Brussel, there are three Data Stewards helping researchers with their questions about Research Data Management. Researchers can send their questions to a shared inbox and depending on the faculty they belong to; the corresponding data steward will answer the email:

- *Jone* helps researchers in the faculties of Sciences and Bioengineering Sciences (**WE**), and Engineering (**IR**)
- *Pieter* helps researchers in the faculties of Medicine and Pharmacy (**GF**), and Physical Education and Physiotherapy (**LK**)
- *Özgün* helps researchers in the faculties of Languages and Humanities (**LW**), Multidisciplinary Institute for Teacher Education (**MILO**), Psychology and Educational Sciences (**PE**), Law and Criminology (**RC**), and Social Sciences and Solvay Business School (**SW**)

At the end of the year, they are required to report the number of questions they have handled to the administration of the university for statistical purposes. In order to report these numbers in an efficient way, a Python code will be written that counts how many questions they received per faculty and how many of these questions were dealt with by which Data Steward.

Design

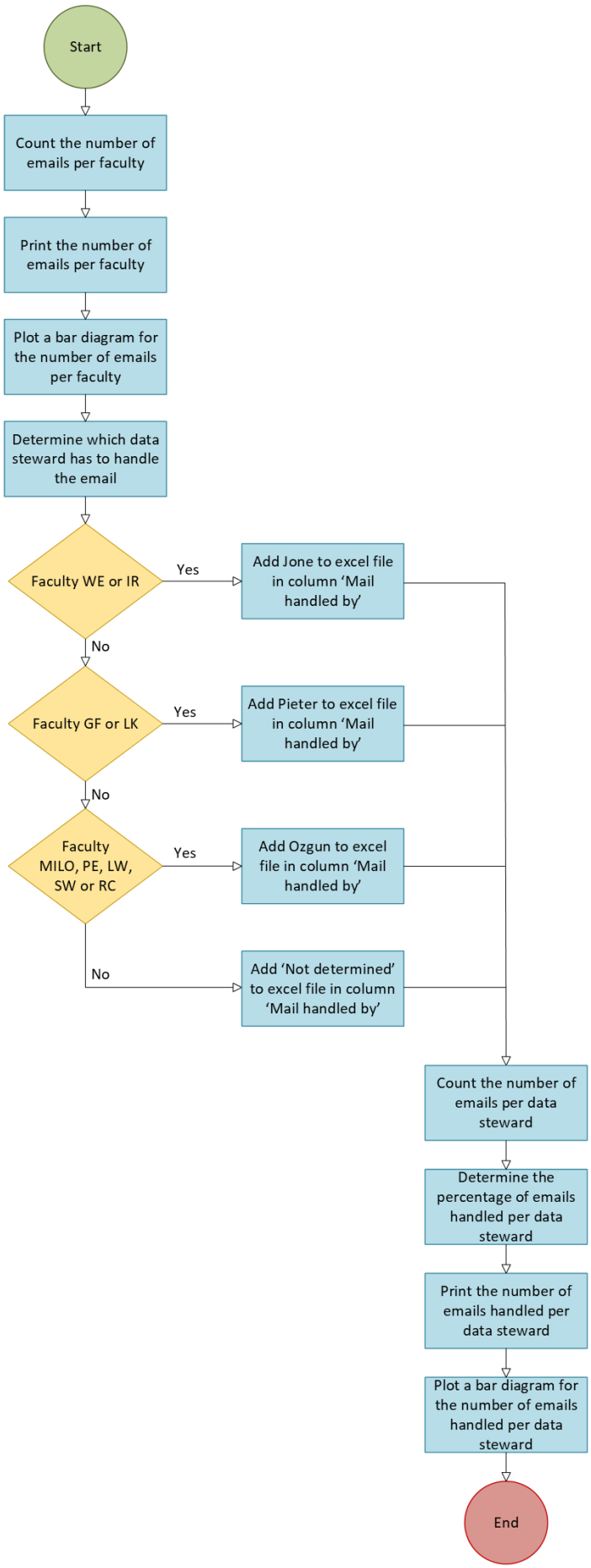
The input file is an Excel file where emails are matched with the faculty name of the researchers. Following task should be executed:

- Count and print the number of emails per faculty and plot a bar diagram with this information
- Determine which Data Steward should handle the email based on the faculty:
    - If WE or IR: Jone
    - If GF or LK: Pieter
    - If MILO, PE, SW, RC, or LW: Özgün
- Add the name of the Data Steward in a new column (i.e. “Mail handled by”) in the Excel file
- Count and print the number of emails dealt with per Data Steward and plot a bar diagram with this information
- Determine the percentage of questions each Data Steward has dealt with

Needed libraries: pandas, numpy, matplotlib.pyplot

Note: Since this was the first time we (Özgün and Jone) had to write code in Python, ChatGPT 3.5 was used to help create a well working program...

Below is a visual process flowchart of the design:


Start
Count the number of emails per faculty
Print the number of emails per faculty
Plot a bar diagram for the number of emails per faculty
Determine which data steward has to handle the email
Faculty WE or IR
Yes
Add Jone to excel file in column 'Mail handled by'
No
Faculty GF or LK
Yes
Add Pieter to excel file in column 'Mail handled by'
No
Faculty MILO, PE, LW, SW or RC
Yes
Add Ozgun to excel file in column 'Mail handled by'
No
Add 'Not determined' to excel file in column 'Mail handled by'
Count the number of emails per data steward
Determine the percentage of emails handled per data steward
Print the number of emails handled per data steward
Plot a bar diagram for the number of emails handled per data steward
End

Implementation

See file “Assignment_2.7_final.py” with additional input file “task_2.7_dataset.xlsx”.

In case something goes wrong with the Python-file, the code is also added below.

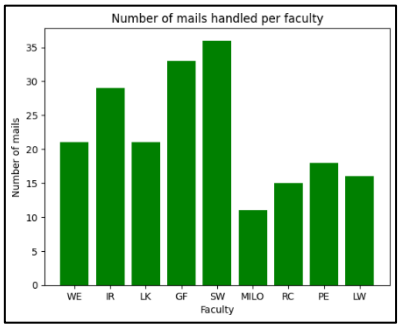
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Read excel file
file_name = "task_2.7_dataset.xlsx"
df = pd.read_excel(file_name)

# Count occurrences of faculty in the 'Faculty' column
faculty_counts = df['Faculty'].value_counts()

# Extract counts for specific faculty values
faculty_values = ['WE', 'IR', 'LK', 'GF', 'SW', 'MILO', 'RC', 'PE', 'LW']
faculty_counts_list = [faculty_counts.get(value, 0) for value in faculty_values]

# Print occurrences of faculty values
for value, count in zip(faculty_values, faculty_counts_list):
    print(f"Occurrences of {value}: {count}")

# Create a bar chart for the number of mails per faculty
plt.bar(faculty_values, faculty_counts_list, color='green')
plt.xlabel('Faculty')
plt.ylabel('Number of mails')
plt.title('Number of mails handled per faculty')
plt.show()

# Function to determine the name based on faculty
def assign_name(row):
    if row['Faculty'] in ['WE', 'IR']:
        return 'Jone'
    elif row['Faculty'] in ['LK', 'GF']:
        return 'Pieter'
    elif row['Faculty'] in ['MILO', 'PE', 'RC', 'LW', 'SW']:
        return 'Ozgun'
    else:
        return 'Not determined'

# Add Jone, Pieter or Ozgun depending on faculty
df['Mail handled by'] = df.apply(assign_name, axis=1)
df.to_excel(file_name, index=False)

# Count occurrences of names in the 'Mail handled by' column
name_counts = df['Mail handled by'].value_counts()
```

```
# Define names dynamically
names = name_counts.index.tolist()

# Extract counts for specific names
counts = np.array([name_counts.get(name, 0) for name in names])

# Calculate the total count
total_count = counts.sum()

# Calculate the percentage
percentages = (counts / total_count) * 100

# Print occurrences and percentages
for name, count, percentage in zip(names, counts, percentages):
    print(f"Occurrences of {name}: {count} ({percentage:.2f}%)")

# Create a bar chart for the number of mails handled per data steward
plt.bar(names, counts, color=['purple', 'orange', 'blue'])
plt.xlabel('Name of data steward')
plt.ylabel('Number of emails')
plt.title('Number of mails handled per data steward')
plt.show()
```

Verification (testing)

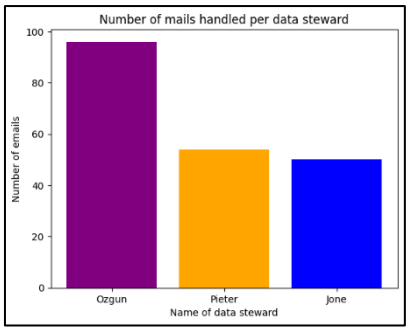
Output from Python code

Occurrences of WE: 21
Occurrences of IR: 29
Occurrences of LK: 21
Occurrences of GF: 33
Occurrences of SW: 36
Occurrences of MILO: 11
Occurrences of RC: 15
Occurrences of PE: 18
Occurrences of LW: 16



	A	B	C	D
1	Mail number	Faculty	Mail handled by	
2	1	PE	Ozgun	
3	2	PE	Ozgun	
4	3	MILO	Ozgun	
5	4	RC	Ozgun	
6	5	LW	Ozgun	
7	6	LW	Ozgun	
8	7	LW	Ozgun	
9	8	SW	Ozgun	
10	9	SW	Ozgun	
11	10	SW	Ozgun	
12	11	SW	Ozgun	
13	12	SW	Ozgun	
14	13	WE	Jone	
15	14	WE	Jone	
16	15	WE	Jone	
17	16	WE	Jone	
18	17	IR	Jone	
19	18	IR	Jone	
20	19	IR	Jone	
21	20	IR	Jone	
22	21	IR	Jone	
23	22	IR	Jone	
24	23	GF	Pieter	

Occurrences of Özgün: 96 (48.00%)
Occurrences of Pieter: 54 (27.00%)
Occurrences of Jone: 50 (25.00%)



Check via Excel

To verify that our code calculated the right numbers, we checked the number of faculties one by one on Excel and found that all numbers were correct:

	A	B
1		
2	**Faculty**	**Count of Faculty**
3	SW	36
4	GF	33
5	IR	29
6	WE	21
7	LK	21
8	PE	18
9	LW	16
10	RC	15
11	MILO	11
12	**Grand Total**	**200**

Data Steward	**# DMPs handled**		**% of #DMPs handled**	
Özgün	96	=SUM(B3+B8+B9+B10+B11)	48	=(96/200)*100
Jone	50	=SUM(B5+B6)	25	=(25/200)*100
Pieter	54	=SUM(B4+B7)	27	=(27/200)*100