

Aayush Shah

D1 - 19BCE245

18 April 2021

Practical 7

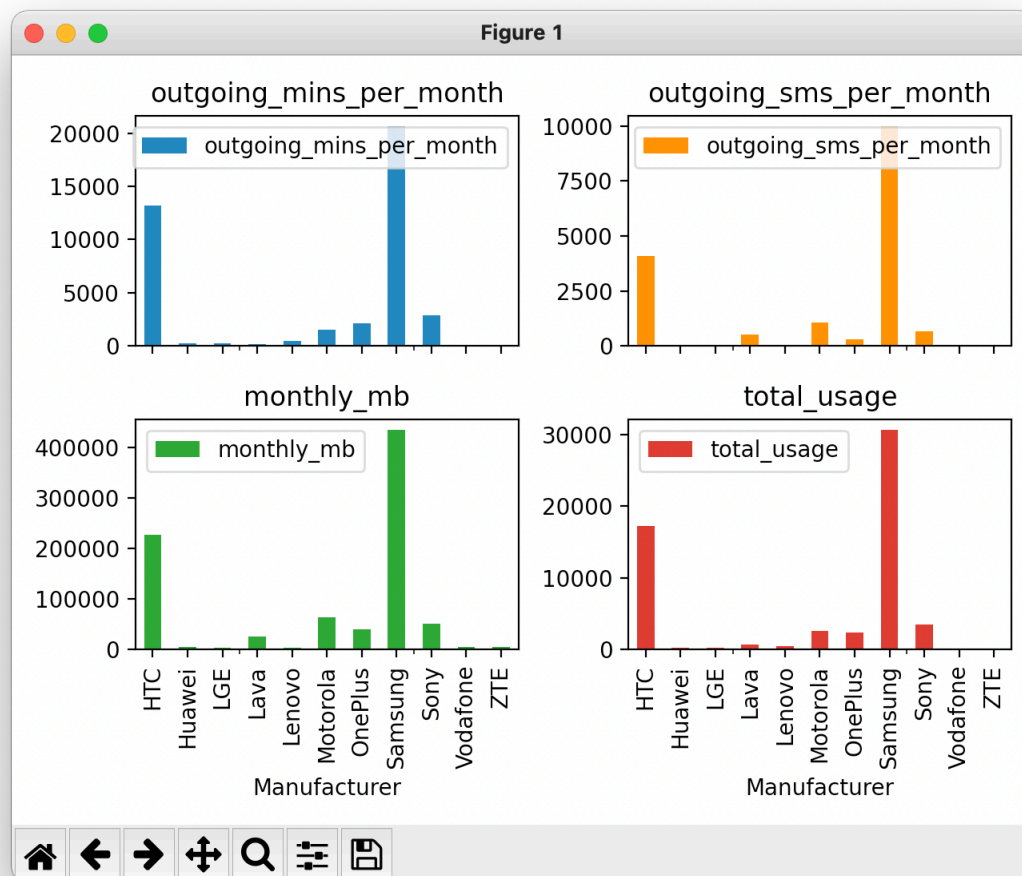
Develop a python program that reads the data from a given CSV file, which is having phone usage data using a different branded mobile phone. Determine if the usage patterns for users differ between different devices. For example, do users using Samsung devices use more call minutes than those using LG devices?

Code :

```
1. import matplotlib.pyplot as plt
2. import pandas as pd
3.
4. # Reading a comma-separated values (csv) file
5. user_data =
    pd.read_csv(io.BytesIO(uploaded['user_data.csv']))
6. devices_data =
    pd.read_csv(io.BytesIO(uploaded['device_data.csv']))
7. usage_data =
    pd.read_csv(io.BytesIO(uploaded['usage_data.csv']))
8. pd.set_option('display.max_columns', None)
9. pd.set_option('display.max_rows', 5)
10.
11. print("USER DATA : \n",user_data)
12. print("DEVICE DATA : \n",devices_data)
13. print("USAGE DATA : \n",usage_data)
14.
```

```
15.result_data = pd.merge(usage_data,
    user_data[['use_id', 'platform', 'device']], on='use_id',
    how='left')          # on='use_id' : Column (here, use_id) or
                           index level names to join on. || how='left' : use only keys
                           from left frame, similar to a SQL left outer join; preserve
                           key order.
16.print("DATA after merge1 : \n",result_data)
17.
18.devices_data.rename(columns={"Retail Branding":"Company"},
    inplace=True)          # inplace : Whether to return a new
                           DataFrame.
19.print("DATA after rename : \n",devices_data)
20.result_data = pd.merge(result_data,
    devices_data[['Company', 'Model']], left_on='device',
    right_on='Model', how='left')          # left_on : Column or
                           index level names to join on in the left DataFrame.
21.print("DATA after merge 2 : \n",result_data)
22.result_data['total_usage'] = result_data.iloc[:,
    :2].sum(axis=1)
23.print("DATA after adding sum column : \n",result_data)
24.
25.final_data =
    pd.DataFrame(result_data.groupby("Company").agg({
26.     "outgoing_mins_per_month": "sum",
27.     "outgoing_sms_per_month" : "sum",
28.     "monthly_mb": "sum",
29.     "total_usage": "sum"
30.}))
31.print("FINAL DATA : \n",final_data)
32.
33.final_data.plot(kind='bar', subplots=True, layout=(2,2))
34.plt.tight_layout()          # automatically adjusts subplot
    params so that the subplots fits in to the figure area.
35.plt.show()
```

Output (graph generated) :



Output (extra data) :

USER DATA :

	use_id	user_id	platform	platform_version
device \				
0	22782	26980	ios	10.2
iPhone7,2				
1	22783	29628	android	6.0
Nexus 5				
...
...				
270	23052	29727	ios	10.1
iPhone8,4				
271	23053	20257	android	5.1
Smart ultra 6				

	use_type_id
0	2
1	3

```

..          ...
270          3
271          1

```

[272 rows x 6 columns]

DEVICE DATA :

	Retail Branding	Marketing Name	Device
Model			
0	NaN	NaN	AD681H Smartfren
Andromax AD681H			
1	NaN	NaN	FJL21
FJL21			
...
...			
14544	tecmobile	OmnisOne	OmnisOne
Omnis One			
14545	ucall	EASY1	EASY1
EASY1			

[14546 rows x 4 columns]

USAGE DATA :

	outgoing_mins_per_month	outgoing_sms_per_month
monthly_mb use_id		
0	21.97	4.82
1557.33 22787		
1	1710.08	136.88
7267.55 22788		
..
..	...	
238	632.06	120.46
1453.16 25058		
239	488.70	906.92
3089.85 25220		

[240 rows x 4 columns]

DATA after merge1 :

	outgoing_mins_per_month	outgoing_sms_per_month
monthly_mb use_id \		
0	21.97	4.82
1557.33 22787		
1	1710.08	136.88
7267.55 22788		
..
..	...	
238	632.06	120.46
1453.16 25058		
239	488.70	906.92
3089.85 25220		

```

    platform    device
0    android    GT-I9505
1    android    SM-G930F
..          ...
238      NaN      NaN
239      NaN      NaN

```

[240 rows x 6 columns]

DATA after rename :

```

          Company Marketing Name      Device
Model
0          NaN          NaN      AD681H  Smartfren Andromax
AD681H
1          NaN          NaN      FJL21
FJL21
...          ...          ...          ...
...
14544  tecmobile      OmnisOne  OmnisOne
Omnis One
14545      ucall      EASY1      EASY1
EASY1

```

[14546 rows x 4 columns]

DATA after merge 2 :

```

          outgoing_mins_per_month  outgoing_sms_per_month
monthly_mb  use_id \
0          21.97          4.82
1557.33    22787
1          1710.08          136.88
7267.55    22788
..          ...          ...
..          ...
290          632.06          120.46
1453.16    25058
291          488.70          906.92
3089.85    25220

```

```

    platform    device  Company      Model
0    android    GT-I9505  Samsung    GT-I9505
1    android    SM-G930F  Samsung    SM-G930F
..          ...          ...          ...
290      NaN      NaN      NaN      NaN
291      NaN      NaN      NaN      NaN

```

[292 rows x 8 columns]

DATA after adding sum column :

```

          outgoing_mins_per_month  outgoing_sms_per_month
monthly_mb  use_id \

```

```

0          21.97          4.82
1557.33    22787
1          1710.08        136.88
7267.55    22788
..          ...          ...
..          ...
290        632.06        120.46
1453.16    25058
291        488.70        906.92
3089.85    25220

```

	platform	device	Company	Model	total_usage
0	android	GT-I9505	Samsung	GT-I9505	26.79
1	android	SM-G930F	Samsung	SM-G930F	1846.96
..
290	NaN	NaN	NaN	NaN	752.52
291	NaN	NaN	NaN	NaN	1395.62

[292 rows x 9 columns]

FINAL DATA :

monthly_mb \	Company	outgoing_mins_per_month	outgoing_sms_per_month
HTC		13193.09	4094.61
226339.43			
Huawei		244.58	28.50
4683.68			
...	
...			
Vodafone		42.75	46.83
5191.12			
ZTE		42.75	46.83
5191.12			

	total_usage
Company	
HTC	17287.70
Huawei	273.08
...	...
Vodafone	89.58
ZTE	89.58

[11 rows x 4 columns]