Programming task

August 5, 2020

0.1 Importing the packages

[111]:
```
# Importing the necessary packages
import pandas as pd
import os
import xlrd
import matplotlib as plt
%autosave 25
```

```
Autosaving every 25 seconds
```

0.2 Reading the data from the directory and storing it as a dataframe

[112]:
```
# Reading the csv files from the directory
areas=pd.read_csv("./areas.csv")
visits=pd.read_csv("./visits.csv")
```

[113]:
```
# Reading the text file in to a dictionary
# with users as key and beneficiary as values
d = {}
with open("users.txt") as f:
    for line in f:
            (key, val) = line.split()
            d[str(key)] = val

# Converting the dictionary into a dataframe which later will be used to form a␣
 ↪master table

users=pd.DataFrame(list(d.items()), index=range(1,28))
users.columns=["users","beneficiaries"]
```

0.3 Checking the data for any null values

[114]:
```
areas.head(5)
```

[114]:

	Beneficiary	Area
0	beneficiary_575	district_1

```
1  beneficiary_2703  district_1
2  beneficiary_2682  District_1
3  beneficiary_2615  district_1
4  beneficiary_1186  District_1
```

[115]:
```
visits.head(5)
```

[115]:
```
      Type       Beneficiary
0  Clinic   beneficiary_933
1    Home  beneficiary_1969
2    Home  beneficiary_1297
3  Clinic  beneficiary_2411
4  Clinic   beneficiary_545
```

[116]:
```
visits.isnull().sum()
```

[116]:
```
Type           0
Beneficiary    0
dtype: int64
```

[117]:
```
areas.isnull().sum()
```

[117]:
```
Beneficiary    0
Area           0
dtype: int64
```

[118]:
```
users.isnull().sum()
```

[118]:
```
users            0
beneficiaries    0
dtype: int64
```

0.4 Creating a master data by merging all the 3 dataframes

[120]:
```
# Merging the visit and areas data using inner join on beneficiary column
merged_data=pd.merge(visits,areas,on='Beneficiary')
```

[121]:
```
# Assigning each user with beneficiary inorder to identify which beneficiary is␣
 ↪associated with
# which user(Health care worker)
d1 = {k: oldk for oldk, oldv in d.items() for k in oldv.split(',')}

# Creating a new colum User which contains the userid
merged_data['User']=merged_data.Beneficiary.map(d1)
```

[122]:
```
# Viewing master data which contains
# which beneficiary is associated with user in which district and where have␣
 ↪they recieved healthcare
merged_data.tail(5)
```

[122]:

	Type	Beneficiary	Area	User
14457	Home	beneficiary_81	district_5	user_23:
14458	Clinic	beneficiary_609	DISTRICT_6	user_14:
14459	Clinic	beneficiary_1927	District_5	user_6:
14460	Home	beneficiary_2407	District_14	user_15:
14461	Clinic	beneficiary_2451	District_14	user_19:

[123]:
```
# Cleaning the data
# District names( Area colum) had some inconsistencies which had mix of␣
 ↪lowercase and uppercase names
# For eg: district_1 & District_1 are the same. To avoid inconsistenicies in␣
 ↪casing, all the names
# were converted into uppercase

merged_data['Area']=merged_data['Area'].str.upper()
```

0.5 (A) a visual representation of which districts have beneficiaries that are not receiving sufficient amounts of healthcare

[124]:
```
# Using the pivot function to find which count of beneficiaries in each␣
 ↪district to have
# recieved healthcare from user
p1=pd.
 ↪pivot_table(merged_data,columns=['Area'],values=["Beneficiary"],aggfunc=[len]).
 ↪T
```

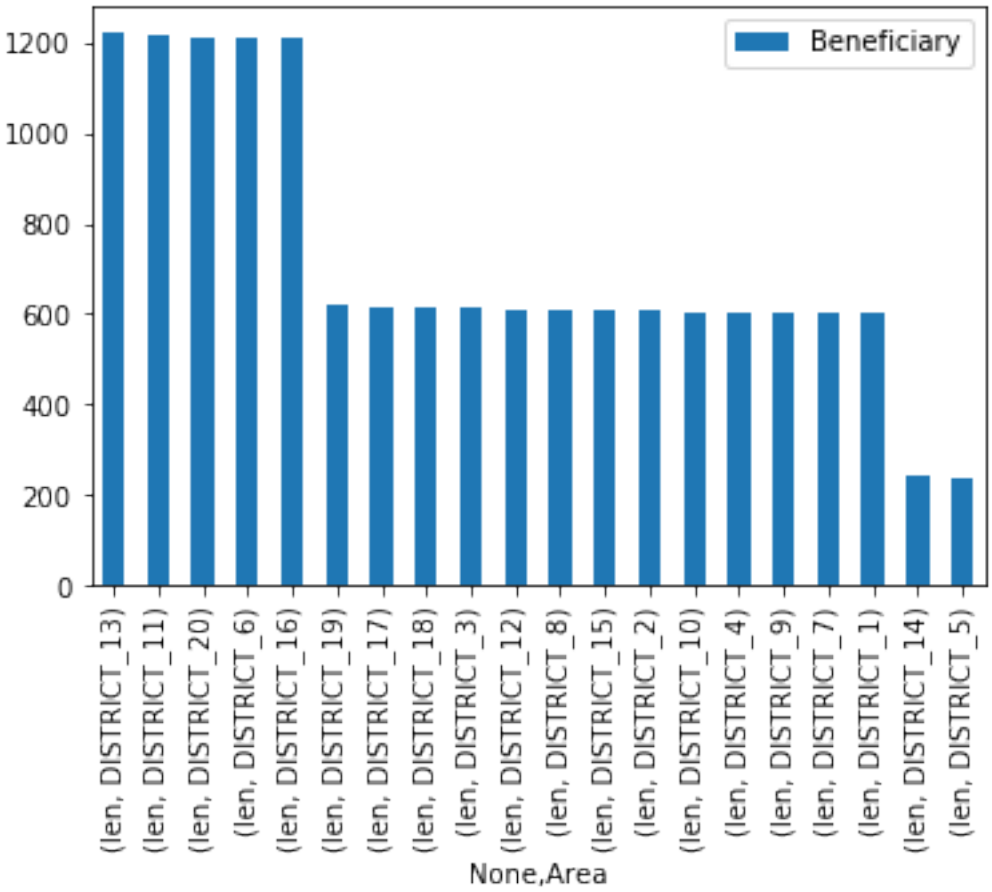
[134]:
```
p2.head(4)
```

[134]:

		Beneficiary
	User	
len	user_10:	612
	user_11:	609
	user_12:	608
	user_13:	611

[126]:
```
# Sorting the values in descending order and plotting
p1.sort_values('Beneficiary', ascending=False).plot.bar(stacked=True)
```

[126]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1ef3f5da208>
```



1 Interpretation from the plot

1.1 From the above plot following can be seen clearly:

1.1.1 1.District_14 and District_5 have beneficiaries that have been recieving the least amount of healthcare.

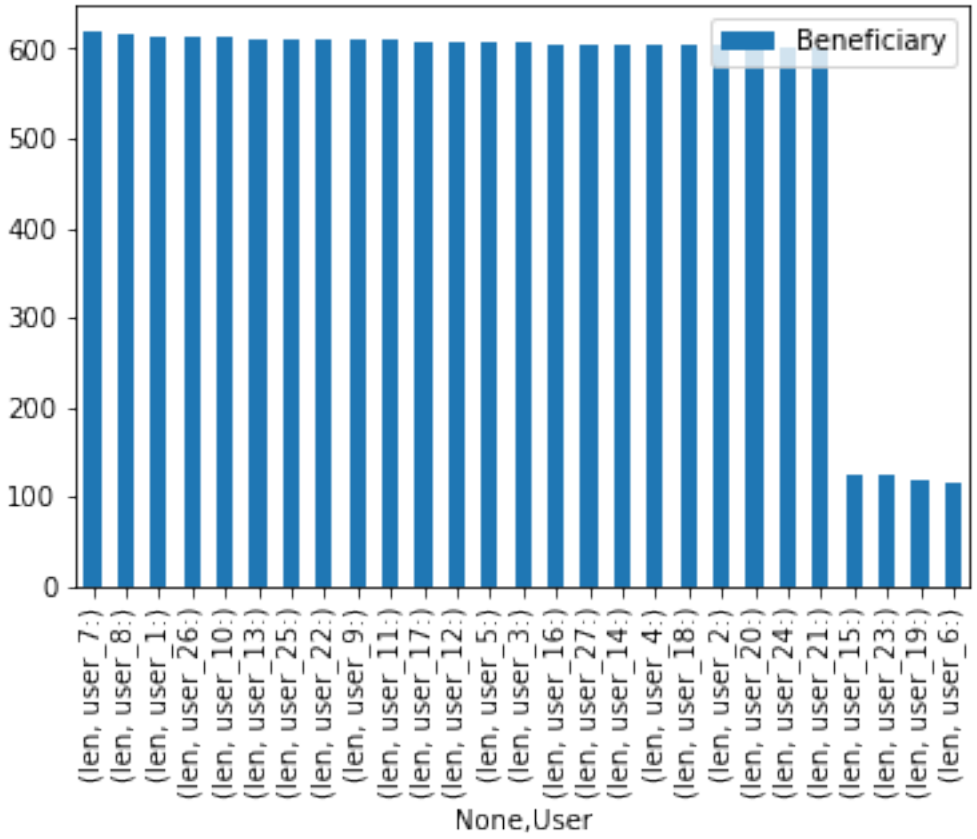
1.1.2 2.District 13,11,20,6,16 have beneficiaries that have been recieving the most amount of healthcare.

1.2 (B) a visual representation of which users are over/underperforming

[128]:
```
# Using the pivot function to find which count of beneficiaries to have␣
 ↪recieved healthcare from user
p2=pd.
 ↪pivot_table(merged_data,columns=['User'],values=["Beneficiary"],aggfunc=[len]).
 ↪T
```

[129]:
```
# Sorting the values in descending order and plotting
p2.sort_values('Beneficiary', ascending=False).plot.bar(stacked=True)
```

[129]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1ef3f6a4908>
```



2 Interpretation from the plot

2.1 From the above plot following can be seen clearly:

2.1.1 1. User 15,23,19,6 have been underperforming when compared with other users

[132]:
```
p1.describe()
```

[132]:

	Beneficiary
count	20.000000
mean	723.100000
std	311.473561
min	238.000000
25%	603.750000
50%	610.000000

```
75%     765.750000
max    1222.000000
```

3 Analysis of the insights

3.0.1 a. For the first part (A) analysis, we can see that on an average 610 beneficiaries have been recieving the healthcare in each district.

3.0.2 b. 2 out 20 districts have recieved the minimum healtcare for the beneficiaries.

3.0.3 c. 5 districts have recieved the max healthcare for the beneficiaries and lie above 75th percentile.

[133]:
```
p2.describe()
```

[133]:
```
        Beneficiary
count     27.000000
mean     535.629630
std      176.793915
min      115.000000
25%      603.000000
50%      606.000000
75%      610.000000
max      618.000000
```