Hello, welcome to another lesson on python and the data science libraries (Numpy,Pandas and Matplotlib). This is another dummy dataset that i worked on as i learned and grow on this career path. I want you to connect deeply with these dataset and feel encouraged about what you're doing. We all do it a little poorly until we get better.That's why i share my journey and my growth process.

In this very dataset, the data is the same as that of the last lesson. But the approach is about using advance techniques to achieve the same task while writing lesser lines of code to achieve the task. Also, Matplotlib Library for Data Visualization will be introduced in this lesson.

And to make it more interesting, you'll join 2 datasets together which you'll read from separate pages of this excel file which will be imported

So fasten your seatbelts and Let's Get Started !!!

The first step is to import the necessary libraries using standard convention. And this time you're going to import matplotlib. Then import the data and read some chunk of it ¶

Notice the line of code that reads "%matplotlib inline".

This will ensure that everytime you write a command to plot, it'll be displayed without additional effort. Skipping this step will mean you have to type "plt.show()" everytime you make a plot.

So, save yourself some stress !

In [1]:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:

```
dataset = pd.read_excel("Excursion Portfolio.xls", sheet_name = 0)
dataset.head(10)
```

Out[2]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unname
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
4	NaN	NaN	NaN	NaN	S/N	Name	Status	Amount Paid	T-sh
5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Ty
6	NaN	NaN	NaN	NaN	1	Victor T. Na'Allah	Member	12000	Lo Slee
7	NaN	NaN	NaN	NaN	2	Olajide Mattew	Member	12000	Lo Slee
8	NaN	NaN	NaN	NaN	3	Abel Modu Timothy	Member	12000	Lo Slee
9	NaN	NaN	NaN	NaN	4	Egwim Jones Udojuaku	Member	12000	Lo Slee

Check the summary description and statistics of the Dataset to familiarize yourself with the data

In [3]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58 entries, 0 to 57
Data columns (total 11 columns):
Unnamed: 0     0 non-null float64
Unnamed: 1     0 non-null float64
Unnamed: 2     0 non-null float64
Unnamed: 3     0 non-null float64
Unnamed: 4     52 non-null object
Unnamed: 5     50 non-null object
Unnamed: 6     50 non-null object
Unnamed: 7     51 non-null object
Unnamed: 8     41 non-null object
Unnamed: 9     41 non-null object
Unnamed: 10    7 non-null object
dtypes: float64(4), object(7)
memory usage: 3.5+ KB
```

```
In [4]: dataset.describe()
```

Out[4]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3
count	0.0	0.0	0.0	0.0
mean	NaN	NaN	NaN	NaN
std	NaN	NaN	NaN	NaN
min	NaN	NaN	NaN	NaN
25%	NaN	NaN	NaN	NaN
50%	NaN	NaN	NaN	NaN
75%	NaN	NaN	NaN	NaN
max	NaN	NaN	NaN	NaN

```
In [5]: dataset.shape
```

```
Out[5]: (58, 11)
```

```
In [6]: dataset.size
```

```
Out[6]: 638
```

```
In [7]: dataset.ndim
```

```
Out[7]: 2
```

```
In [8]: dataset.dtypes
```

```
Out[8]: Unnamed: 0     float64
        Unnamed: 1     float64
        Unnamed: 2     float64
        Unnamed: 3     float64
        Unnamed: 4      object
        Unnamed: 5      object
        Unnamed: 6      object
        Unnamed: 7      object
        Unnamed: 8      object
        Unnamed: 9      object
        Unnamed: 10     object
        dtype: object
```

Since a visual clue to the data reveals that some rows and columns contains nan, use "dropna" to delete all rows and columns that contains "nan" across the columns and down the rows respectively

```
In [9]: dataset.dropna(how = 'all', inplace = True)
```

```
In [10]: dataset.dropna(how = "all", axis = 1, inplace = True)
```

In [11]:

```
dataset.head()
```

Out[11]:

	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10
4	S/N	Name	Status	Amount Paid	T-shirt	NaN	Comments
5	NaN	NaN	NaN	NaN	Type	Amount	NaN
6	1	Victor T. Na'Allah	Member	12000	Long Sleeve	2500	NaN
7	2	Olajide Mattew	Member	12000	Long Sleeve	2500	NaN
8	3	Abel Modu Timothy	Member	12000	Long Sleeve	2500	NaN

Create a new column label and assign it accordingly

In [12]:

```
new_col = list(dataset.loc[4])
new_col
```

Out[12]:

```
['S/N', 'Name', 'Status', 'Amount Paid', 'T-shirt', nan, 'Comments']
```

In [13]:

```
new_col[5] = 'T-Shirt Amount'
new_col
```

Out[13]:

```
['S/N',
 'Name',
 'Status',
 'Amount Paid',
 'T-shirt',
 'T-Shirt Amount',
 'Comments']
```

In [14]:

```
dataset.columns = new_col
dataset.head()
```

Out[14]:

	S/N	Name	Status	Amount Paid	T-shirt	T-Shirt Amount	Comments
4	S/N	Name	Status	Amount Paid	T-shirt	NaN	Comments
5	NaN	NaN	NaN	NaN	Type	Amount	NaN
6	1	Victor T. Na'Allah	Member	12000	Long Sleeve	2500	NaN
7	2	Olajide Mattew	Member	12000	Long Sleeve	2500	NaN
8	3	Abel Modu Timothy	Member	12000	Long Sleeve	2500	NaN

Create a new index label and assign it accordingly and

In [15]:

```
new_index = list(dataset['Name'])
new_index
```

```
Out[15]: ['Name',
          nan,
          "Victor T. Na'Allah",
          'Olajide Mattew',
          'Abel Modu Timothy',
          'Egwim Jones Udojuaku',
          'Nwachukwu Emmanuel Benedict',
          'Adole John A.',
          'Faleti Ayodeji Peter',
          'Ayantoye Ridwan Ayomide',
          'Marvellous  T. Isaac',
          'Shenge Raphael Saarshatar',
          'Reuben O.Enoch',
          'Mercy Ajayi',
          'Michael Kpoco',
          'Saad',
          'Terzungwe Caleb',
          'Faith',
          'Gonet Zion',
          'Hawwau Adeboyin Adeyemo\nFor => Jubrin Omeiza',
          'Nicholas Otonoku',
          'Timothy Ignitus Agbor',
          'Madumche Chidibere',
          'Nwokocha Ethelbert',
          'Effiong Ubon Alasi ',
          'Daniel Overcomer',
          'Aguwa Wisdom',
          'Ganiyu Mujeeb',
          'Hezekiel Joel',
          'Shitu Mustapha Ibrahim',
          'Orogu Francis Israel',
          'Agha Elizabeth',
          'Muhammed Zainab',
          'Isaac Priscilla',
          'Paul Elizabeth Ladi',
          'Omaji Samuel Owoicho',
          'Esinome Abraham',
          'Umeh Audu Ayigba',
          'Sarah Kauna Edoja',
          'Ibrahim Hussein Chado',
          'Fatima Ganiyu',
          'Daleng Elisha Nandi',
          'Ukande Aondongu Cephas',
          'De-Gold David Tarki',
          'Shekinah Ajibola',
          'Adanu David',
          'Oche Muscle',
          'Ocheje Jeremiah',
          'Oguntowo Basit Ifedolapo',
          'Simeon Iganga',
          'Caleb Onuoja Aaron',
          nan,
          nan]
```

There are some " nan" in. the index list i just created. To avoid having issues dealing with them, I decided to give it a proper name below

In [16]:

```
new_index[1] = "count"
```

In [17]:

```
new_index[-1] = "total"
```

In [18]:

```
new_index[-2] = "type"
```

In [19]:
```
new_index
```

Out[19]:
```
['Name',
 'count',
 "Victor T. Na'Allah",
 'Olajide Mattew',
 'Abel Modu Timothy',
 'Egwim Jones Udojuaku',
 'Nwachukwu Emmanuel Benedict',
 'Adole John A.',
 'Faleti Ayodeji Peter',
 'Ayantoye Ridwan Ayomide',
 'Marvellous  T. Isaac',
 'Shenge Raphael Saarshatar',
 'Reuben O.Enoch',
 'Mercy Ajayi',
 'Michael Kpoco',
 'Saad',
 'Terzungwe Caleb',
 'Faith',
 'Gonet Zion',
 'Hawwau Adeboyin Adeyemo\nFor => Jubrin Omeiza',
 'Nicholas Otonoku',
 'Timothy Ignitus Agbor',
 'Madumche Chidibere',
 'Nwokocha Ethelbert',
 'Effiong Ubon Alasi ',
 'Daniel Overcomer',
 'Aguwa Wisdom',
 'Ganiyu Mujeeb',
 'Hezekiel Joel',
 'Shitu Mustapha Ibrahim',
 'Orogu Francis Israel',
 'Agha Elizabeth',
 'Muhammed Zainab',
 'Isaac Priscilla',
 'Paul Elizabeth Ladi',
 'Omaji Samuel Owoicho',
 'Esinome Abraham',
 'Umeh Audu Ayigba',
 'Sarah Kauna Edoja',
 'Ibrahim Hussein Chado',
 'Fatima Ganiyu',
 'Daleng Elisha Nandi',
 'Ukande Aondongu Cephas',
 'De-Gold David Tarki',
 'Shekinah Ajibola',
 'Adanu David',
 'Oche Muscle',
 'Ocheje Jeremiah',
 'Oguntowo Basit Ifedolapo',
 'Simeon Iganga',
 'Caleb Onuoja Aaron',
 'type',
 'total']
```

In [20]:
```
dataset.index = new_index
```

In [21]:
```
dataset.head()
```

Out[21]:

	S/N	Name	Status	Amount Paid	T-shirt	T-Shirt Amount	Comments
Name	S/N	Name	Status	Amount Paid	T-shirt	NaN	Comments
count	NaN	NaN	NaN	NaN	Type	Amount	NaN
Victor T. Na'Allah	1	Victor T. Na'Allah	Member	12000	Long Sleeve	2500	NaN
Olajide Mattew	2	Olajide Mattew	Member	12000	Long Sleeve	2500	NaN
Abel Modu Timothy	3	Abel Modu Timothy	Member	12000	Long Sleeve	2500	NaN

Now, delete the irrelevant rows and columns in the dataset at this stage.

In [22]:
```
dataset.drop(['Name','count','type','total'], inplace = True)
dataset.head()
```

Out[22]:

	S/N	Name	Status	Amount Paid	T-shirt	T-Shirt Amount	Comments
Victor T. Na'Allah	1	Victor T. Na'Allah	Member	12000	Long Sleeve	2500	NaN
Olajide Mattew	2	Olajide Mattew	Member	12000	Long Sleeve	2500	NaN
Abel Modu Timothy	3	Abel Modu Timothy	Member	12000	Long Sleeve	2500	NaN
Egwim Jones Udojuaku	4	Egwim Jones Udojuaku	Member	12000	Long Sleeve	2500	NaN
Nwachukwu Emmanuel Benedict	5	Nwachukwu Emmanuel Benedict	Member	12000	Long Sleeve	2500	NaN

In [23]:
```
dataset.drop(['S/N', 'Name', 'Comments'], axis = 1, inplace = True)
```

```
In [24]: dataset.head()
```

Out[24]:

	Status	Amount Paid	T-shirt	T-Shirt Amount
Victor T. Na'Allah	Member	12000	Long Sleeve	2500
Olajide Mattew	Member	12000	Long Sleeve	2500
Abel Modu Timothy	Member	12000	Long Sleeve	2500
Egwim Jones Udojuaku	Member	12000	Long Sleeve	2500
Nwachukwu Emmanuel Benedict	Member	12000	Long Sleeve	2500

```
In [25]: dataset.describe()
```

Out[25]:

	Status	Amount Paid	T-shirt	T-Shirt Amount
count	49	48	39	39
unique	2	7	3	3
top	Member	12000	Long Sleeve	2500
freq	44	38	18	18

Now, it's time to deal with the null data in this dataset. If you decide to drop the null values at this stage, you'll be losing a large chunk of valuable data. A better alternative is to fill the null values as the case requires. My approach is written below

```
In [26]: dataset["Amount Paid"].fillna(0, inplace = True)
```

```
In [27]: dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 49 entries, Victor T. Na'Allah to Caleb Onuoja Aaron
Data columns (total 4 columns):
Status            49 non-null object
Amount Paid       49 non-null int64
T-shirt           39 non-null object
T-Shirt Amount    39 non-null object
dtypes: int64(1), object(3)
memory usage: 1.1+ KB
```

```
In [28]: dataset["T-shirt"].fillna("Long Sleeve", inplace = True)
```

```
In [29]: dataset["T-Shirt Amount"].fillna(2500, inplace = True)
```

```
In [30]: dataset["T-shirt"].replace({"NIL":"No Sleeve"}, inplace = True)
```

```
In [31]: dataset["T-Shirt Amount"].replace({"NIL":0}, inplace = True)
```

In [32]:
```
dataset.index.name = "Name"
dataset.head()
```

Out[32]:

	Status	Amount Paid	T-shirt	T-Shirt Amount
Name				
Victor T. Na'Allah	Member	12000	Long Sleeve	2500
Olajide Mattew	Member	12000	Long Sleeve	2500
Abel Modu Timothy	Member	12000	Long Sleeve	2500
Egwim Jones Udojuaku	Member	12000	Long Sleeve	2500
Nwachukwu Emmanuel Benedict	Member	12000	Long Sleeve	2500

Here, i realized the "Shirt Amount" column is an integer but it's not assigned the correct data type. Hence, i manipulated it to bring it back to the right data type. Just understand the tools at your disposal and the task you want to execute. You can always get creative about how to go about it

In [33]:
```
shortamount = np.array(dataset["T-Shirt Amount"], dtype = np.int64)
shortamount
```

Out[33]:
```
array([2500, 2500, 2500, 2500, 2500, 2100, 2100, 2500, 2100, 2100, 250
0,
       2500, 2100, 2500, 2100, 2500, 2500, 2500, 2100, 2500, 2500, 210
0,
       2500, 2100, 2500, 2500, 2500, 2500, 2100, 2100, 2500, 2100, 210
0,
       2100, 2100, 2100, 2500,    0,    0, 2500, 2500, 2500, 2500,    
0,
          0, 2100, 2500, 2500, 2500], dtype=int64)
```

```
In [34]: dataset["T-Shirt Amount"] = shortamount
         dataset["T-Shirt Amount"].head(20)
```

```
Out[34]: Name
         Victor T. Na'Allah                                 2500
         Olajide Mattew                                     2500
         Abel Modu Timothy                                  2500
         Egwim Jones Udojuaku                               2500
         Nwachukwu Emmanuel Benedict                        2500
         Adole John A.                                      2100
         Faleti Ayodeji Peter                               2100
         Ayantoye Ridwan Ayomide                            2500
         Marvellous  T. Isaac                               2100
         Shenge Raphael Saarshatar                          2100
         Reuben O.Enoch                                     2500
         Mercy Ajayi                                        2500
         Michael Kpoco                                      2100
         Saad                                               2500
         Terzungwe Caleb                                    2100
         Faith                                              2500
         Gonet Zion                                         2500
         Hawwau Adeboyin Adeyemo\nFor => Jubrin Omeiza    2500
         Nicholas Otonoku                                   2100
         Timothy Ignitus Agbor                              2500
         Name: T-Shirt Amount, dtype: int64
```

The dataset is clean and ready for further analysis a d exploration to derive deep insights.

But wait a minute, remember i mentioned the Dataset you'll be dealing with is 2 contained in the same file. You'll read the second file now, clean and merge it with the first one you've worked on

The second dataset has been imported and read. Notice that in the line of code used to import the file, "sheet_name" was set to 1. So it read the second sheet of the source file. When it's not included, that means it's on it's defaults that's set to 0. And automatically reads the first page of the source file

In [35]:

```
dataset1 = pd.read_excel("Excursion Portfolio.xls", sheet_name = 1)
dataset1.head()
```

Out[35]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unn
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1	NaN	NaN	NaN	NaN	S/N	Name	Type	T-shirt	
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Type	A
3	NaN	NaN	NaN	NaN	1	Abuh E. Rasheed	Exco	Short Sleeve	
4	NaN	NaN	NaN	NaN	2	Didi Chukwuebuka V.	Exco	Long Sleeve	

Drop the rows and columns of the dataset that contains "nan" all true

In [36]:

```
dataset1.dropna(how = 'all', inplace = True)
```

In [37]:

```
dataset1.dropna(how = 'all', axis = 1, inplace = True)
```

In [38]:

```
dataset1.head()
```

Out[38]:

	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8
1	S/N	Name	Type	T-shirt	NaN
2	NaN	NaN	NaN	Type	Amount
3	1	Abuh E. Rasheed	Exco	Short Sleeve	2100
4	2	Didi Chukwuebuka V.	Exco	Long Sleeve	2500
5	3	Jonathan Ajiboye	Exco	Short Sleeve	2100

Next, give your index label and column label appropriate names to make them more relatable.

Then drop the irrelevant rows and columsns respectively

In [39]:

```
new_col1 = list(dataset1.loc[1])
new_col1
```

Out[39]:

```
['S/N', 'Name', 'Type', 'T-shirt', nan]
```

In [40]:

```
new_col1[2] = "Status"
new_col1[-1] = "T-Shirt Amount"
new_col1
```

Out[40]:

```
['S/N', 'Name', 'Status', 'T-shirt', 'T-Shirt Amount']
```

In [41]:

```
dataset1.columns = new_col1
```

In [42]:

```
dataset1.drop([1,2,16,17,27], inplace = True)
```

In [43]:

```
dataset1.index = dataset1.Name
```

In [44]:

```
dataset1.drop("Name", axis =1, inplace = True)
```

In [45]:

```
dataset1.drop("S/N", axis = 1, inplace = True)
```

This first Dataset has a column called "Amount Paid", but that column is absent in the secomd dataset. Hence i created a new column in the second Dataset with the same name and values of zero all true.

I used the numpy zeros function to create the array and used that to create the required in the second Dataset.

Like i mentioned earlier on, just know your tools and understand the task you want to execute. There's no limitation to unleashing your creativity

In [46]:

```
new_array = np.zeros((13,1), dtype = np.int64)
new_array
```

Out[46]:

```
array([[0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0],
       [0]], dtype=int64)
```

```
In [47]: dataset1["Amount Paid"] = new_array
         dataset1.head()
```

Out[47]:

	Status	T-shirt	T-Shirt Amount	Amount Paid
Name				
Abuh E. Rasheed	Exco	Short Sleeve	2100	0
Didi Chukwuebuka V.	Exco	Long Sleeve	2500	0
Jonathan Ajiboye	Exco	Short Sleeve	2100	0
Yusuff Fatorisa	Exco	Short Sleeve	2100	0
Gabriel Alkali	Exco	Long Sleeve	2500	0

Like the first dataset, the "Shirt _Amount" column consist of integers but have the wrong data type. So i manipulated it bring it back to the right data type

```
In [48]: shortamount1 = np.array(dataset1["T-Shirt Amount"], dtype = np.int64)
         shortamount1
```

```
Out[48]: array([2100, 2500, 2100, 2100, 2500, 2100, 2500, 2100, 2500, 2500, 250
         0,
                2500, 2500], dtype=int64)
```

```
In [49]: dataset1["T-Shirt Amount"] = shortamount1
         dataset1["T-Shirt Amount"]
```

```
Out[49]: Name
         Abuh E. Rasheed        2100
         Didi Chukwuebuka V.    2500
         Jonathan Ajiboye       2100
         Yusuff Fatorisa        2100
         Gabriel Alkali         2500
         Ebube Onuigbo          2100
         Sanusi Muhammad        2500
         Ameh Sunday            2100
         Bawa Vincent           2500
         Bala Monday            2500
         Mukaila Abdul-Rafiu    2500
         Cynthia John           2500
         Victor Chumnon         2500
         Name: T-Shirt Amount, dtype: int64
```

```
In [50]: dataset1.info()

<class 'pandas.core.frame.DataFrame'>
Index: 13 entries, Abuh E. Rasheed to Victor Chumnon
Data columns (total 4 columns):
Status              13 non-null object
T-shirt             13 non-null object
T-Shirt Amount      13 non-null int64
Amount Paid         13 non-null int64
dtypes: int64(2), object(2)
memory usage: 748.0+ bytes
```

```
In [51]: recol = ["Status", "Amount Paid", "T-shirt", "T-Shirt Amount"]
         recol

Out[51]: ['Status', 'Amount Paid', 'T-shirt', 'T-Shirt Amount']
```

```
In [52]: dataset1.reindex(columns = recol)
         dataset1.head()
```

Out[52]:

	Status	T-shirt	T-Shirt Amount	Amount Paid
Name				
Abuh E. Rasheed	Exco	Short Sleeve	2100	0
Didi Chukwuebuka V.	Exco	Long Sleeve	2500	0
Jonathan Ajiboye	Exco	Short Sleeve	2100	0
Yusuff Fatorisa	Exco	Short Sleeve	2100	0
Gabriel Alkali	Exco	Long Sleeve	2500	0

Both dataset are clean and ready. So use concat function to merge them together

i also rewrite it back into an excel file.

Now it's time for

"DATA VISUALIZATION"

In [53]:

```
comb_dataset = pd.concat([dataset,dataset1])
comb_dataset.tail()
```

```
/data/user/0/ru.iiec.pydroid3/files/arm-linux-androideabi/lib/python3.
7/site-packages/ipykernel_launcher.py:1: FutureWarning: Sorting becaus
e non-concatenation axis is not aligned. A future version
of pandas will change to not sort by default.

To accept the future behavior, pass 'sort=False'.

To retain the current behavior and silence the warning, pass 'sort=Tru
e'.

  """Entry point for launching an IPython kernel.
```

Out[53]:

	Amount Paid	Status	T-Shirt Amount	T-shirt
Name				
Bawa Vincent	0	Exco	2500	Long Sleeve
Bala Monday	0	Exco	2500	Long Sleeve
Mukaila Abdul-Rafiu	0	Exco	2500	Long Sleeve
Cynthia John	0	Exco	2500	Long Sleeve
Victor Chumnon	0	Exco	2500	Long Sleeve

In [54]:

```
comb_dataset.to_excel("ExcurPort.xlsx")
```

I made use of the "groupby" fuction to extract the required data and plot the Visualization

In [55]:

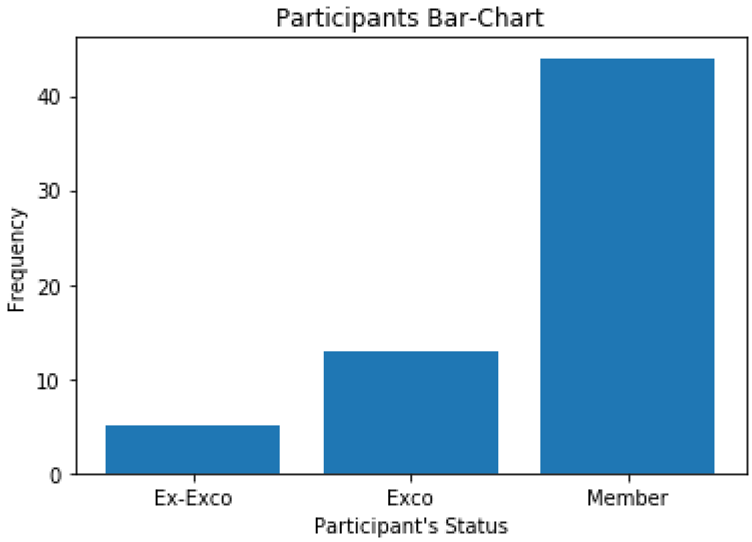
```
class1 = comb_dataset["Amount Paid"].groupby(comb_dataset["Status"])
class11 = class1.count()
class11
crass = list(class11.index)
crass
```

Out[55]:

```
['Ex-Exco', 'Exco', 'Member']
```

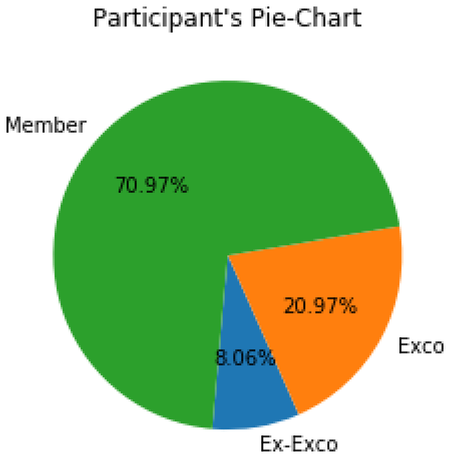
In [56]:

```
plt.bar(crass,class11)
plt.title("Participants Bar-Chart")
plt.xlabel("Participant's Status")
plt.ylabel("Frequency")
plt.savefig("F-Dset.png")
```



In [57]:

```
plt.pie(class11, labels = crass, startangle = 265, autopct ="%1.2f%%")
plt.title("Participant's Pie-Chart")
plt.savefig("F-Dset1.png")
plt.show()
```



In [58]:

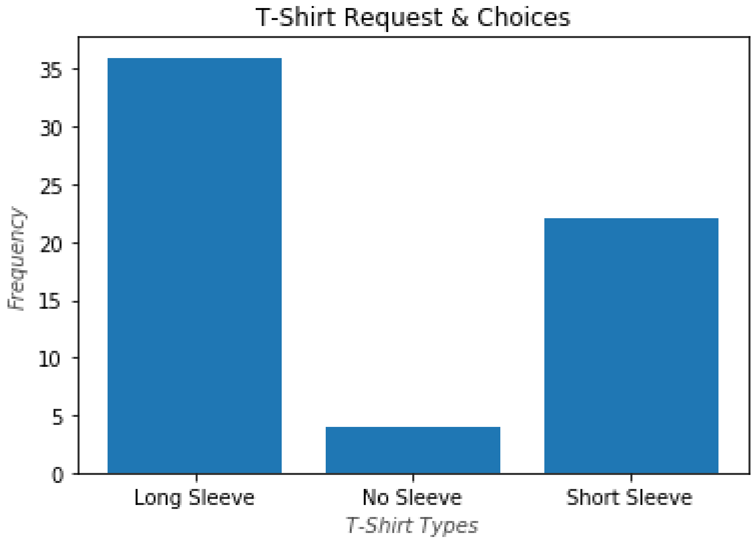
```
class4 = comb_dataset["T-Shirt Amount"].groupby(comb_dataset["T-shirt"
])
class4.unique()
class41 = class4.count()
crass41 = list(class41.index)
crass41
```

Out[58]:

```
['Long Sleeve', 'No Sleeve', 'Short Sleeve']
```

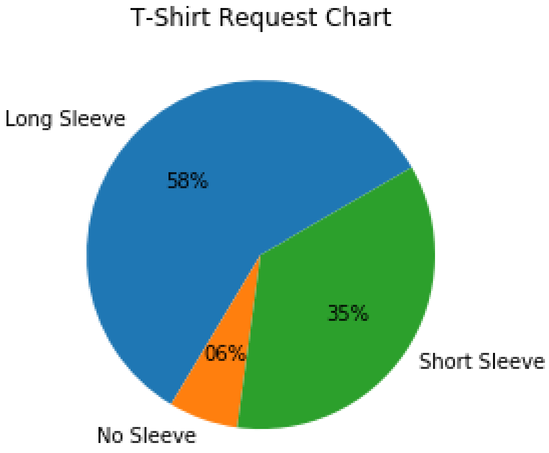
In [59]:

```
plt.bar(crass41, class41)
plt.title("T-Shirt Request & Choices", fontstyle = 'normal')
plt.xlabel("T-Shirt Types", alpha = 0.7, fontstyle = 'italic')
plt.ylabel("Frequency", alpha = 0.7, fontstyle = 'italic')
plt.savefig("F-Dset3.png")
```



In [60]:

```
plt.pie(class41, labels = crass41, autopct ="%1.2d%%", startangle = 30
)
plt.title("T-Shirt Request Chart")
plt.savefig("F-Dset4.png")
```



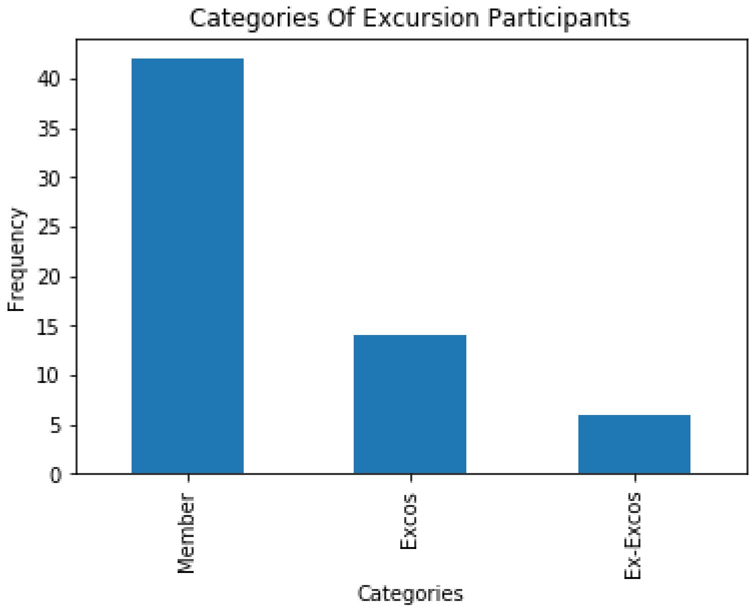
In [61]:

```
parts = [0,0.9,8001,12001]
parts_label = ("Excos","Ex-Excos","Member")
```

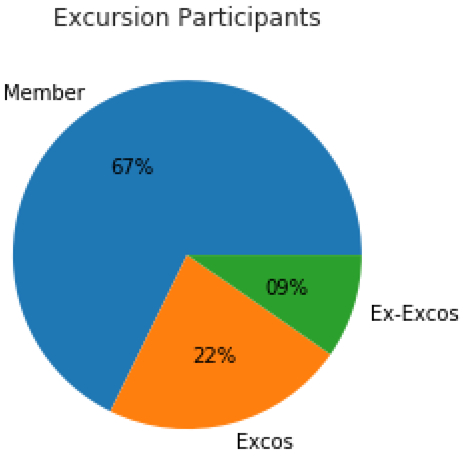
In [62]:
```
class2 = pd.cut(comb_dataset["Amount Paid"], parts, labels = parts_lab
el, right = False)
class21 = pd.value_counts(class2)
class21
```

Out[62]:
```
Member      42
Excos       14
Ex-Excos     6
Name: Amount Paid, dtype: int64
```

In [63]:
```
class21.plot(kind = 'bar')
plt.title("Categories Of Excursion Participants")
plt.xlabel("Categories")
plt.ylabel("Frequency")
plt.savefig("F-Dset5.png")
```



In [64]:
```
class21.plot(kind = "pie", autopct ="%1.2d%%", startangle = 0)
plt.ylabel('')
plt.title("Excursion Participants",alpha = 0.85)
plt.savefig("F-Dset6.png")
```



In [65]:

```
tparts = [0,5,2101,2501]
tparts_label = ("No Sleeve","Short Sleeve", "Long Sleeve")
```

In [66]:

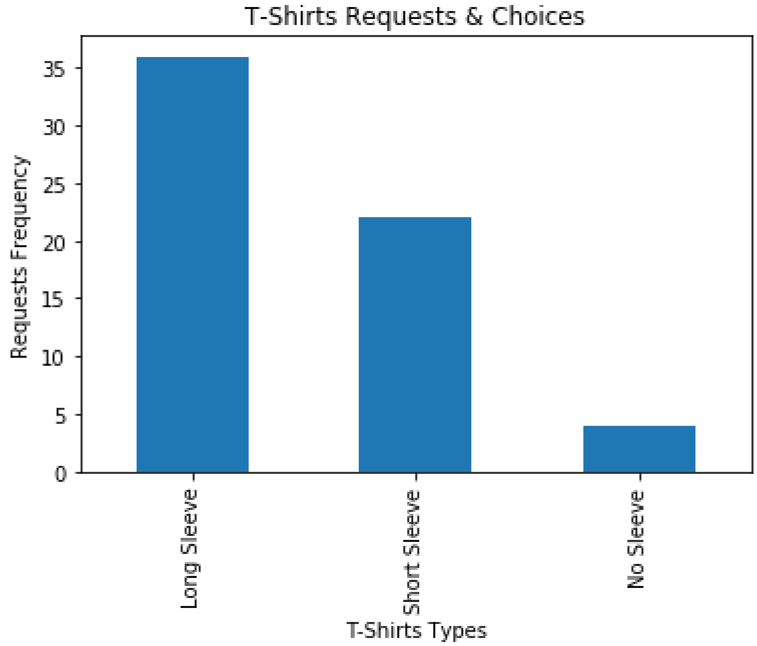
```
class3 = pd.cut(comb_dataset["T-Shirt Amount"],tparts, labels = tparts
_label, right = False)
class31 = pd.value_counts(class3)
class31
```

Out[66]:

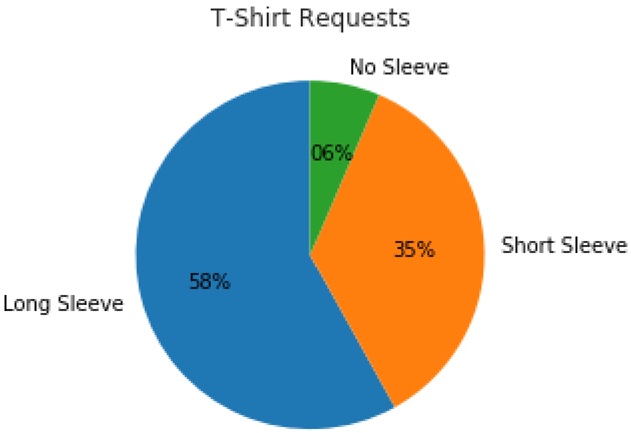
```
Long Sleeve     36
Short Sleeve    22
No Sleeve        4
Name: T-Shirt Amount, dtype: int64
```

In [67]:

```
class31.plot(kind = "bar")
plt.title("T-Shirts Requests & Choices")
plt.xlabel("T-Shirts Types")
plt.ylabel("Requests Frequency")
plt.savefig("F-Dset7.png")
```



```
In [68]: class31.plot(kind = "pie", startangle = 90, autopct = '%1.2d%%', label
         = ("LONG","SHORT"))
         plt.ylabel("")
         plt.title("T-Shirt Requests", alpha = 0.85)
         plt.savefig("F-Dset8.png")
```



There's still a whole bunch of amazing things you can do using the "Matplotlib Library". But I'll leave you to do more exploration and discoveries on your own

```
In [69]: comb_dataset["Status"].unique()

Out[69]: array(['Member', 'Ex-Exco', 'Exco'], dtype=object)
```

```
In [70]: comb_dataset["T-shirt"].unique()

Out[70]: array(['Long Sleeve', 'Short Sleeve', 'No Sleeve'], dtype=object)
```

```
In [71]: comb_dataset["T-Shirt Amount"].unique()

Out[71]: array([2500, 2100,    0], dtype=int64)
```

```
In [72]: comb_dataset["Amount Paid"].unique()

Out[72]: array([12000,  7000,  5000,  8000, 10000,     0,  9000,  9500],
               dtype=int64)
```

In [73]:

```
comb_dataset[comb_dataset["Amount Paid"] == 12000]
```

Out[73]:

	Amount Paid	Status	T-Shirt Amount	T-shirt
Name				
Victor T. Na'Allah	12000	Member	2500	Long Sleeve
Olajide Mattew	12000	Member	2500	Long Sleeve
Abel Modu Timothy	12000	Member	2500	Long Sleeve
Egwim Jones Udojuaku	12000	Member	2500	Long Sleeve
Nwachukwu Emmanuel Benedict	12000	Member	2500	Long Sleeve
Adole John A.	12000	Member	2100	Short Sleeve
Faleti Ayodeji Peter	12000	Member	2100	Short Sleeve
Ayantoye Ridwan Ayomide	12000	Member	2500	Long Sleeve
Marvellous T. Isaac	12000	Member	2100	Short Sleeve
Shenge Raphael Saarshatar	12000	Member	2100	Short Sleeve
Reuben O.Enoch	12000	Member	2500	Long Sleeve
Mercy Ajayi	12000	Member	2500	Long Sleeve
Saad	12000	Member	2500	Long Sleeve
Faith	12000	Member	2500	Long Sleeve
Gonet Zion	12000	Member	2500	Long Sleeve
Hawwau Adeboyin Adeyemo\nFor => Jubrin Omeiza	12000	Member	2500	Long Sleeve
Nicholas Otonoku	12000	Member	2100	Short Sleeve
Timothy Ignitus Agbor	12000	Member	2500	Long Sleeve
Madumche Chidibere	12000	Member	2500	Long Sleeve
Effiong Ubon Alasi	12000	Member	2500	Long Sleeve
Daniel Overcomer	12000	Member	2100	Short Sleeve
Aguwa Wisdom	12000	Member	2500	Long Sleeve
Ganiyu Mujeeb	12000	Member	2500	Long Sleeve
Hezekiel Joel	12000	Member	2500	Long Sleeve
Shitu Mustapha Ibrahim	12000	Member	2500	Long Sleeve
Orogu Francis Israel	12000	Member	2100	Short Sleeve
Agha Elizabeth	12000	Member	2100	Short Sleeve
Muhammed Zainab	12000	Member	2500	Long Sleeve
Isaac Priscilla	12000	Member	2100	Short Sleeve
Paul Elizabeth Ladi	12000	Member	2100	Short Sleeve
Esinome Abraham	12000	Member	2100	Short Sleeve
Umeh Audu Ayigba	12000	Member	2100	Short Sleeve
Ukande Aondongu Cephas	12000	Member	2500	Long Sleeve

	Amount Paid	Status	T-Shirt Amount	T-shirt
Name				
De-Gold David Tarki	12000	Member	2500	Long Sleeve
Shekinah Ajibola	12000	Member	2500	Long Sleeve
Ocheje Jeremiah	12000	Member	2100	Short Sleeve
Oguntowo Basit Ifedolapo	12000	Member	2500	Long Sleeve
Simeon Iganga	12000	Member	2500	Long Sleeve

In [74]:

```
comb_dataset[comb_dataset["Amount Paid"] == 8000]
```

Out[74]:

	Amount Paid	Status	T-Shirt Amount	T-shirt
Name				
Nwokocha Ethelbert	8000	Ex-Exco	2100	Short Sleeve
Omaji Samuel Owoicho	8000	Ex-Exco	2100	Short Sleeve
Sarah Kauna Edoja	8000	Member	2500	Long Sleeve
Caleb Onuoja Aaron	8000	Ex-Exco	2500	Long Sleeve

In [75]:

```
comb_dataset[comb_dataset["Amount Paid"] < 12000]
```

Out[75]:

	Amount Paid	Status	T-Shirt Amount	T-shirt
Name				
Michael Kpoco	7000	Ex-Exco	2100	Short Sleeve
Terzungwe Caleb	5000	Ex-Exco	2100	Short Sleeve
Nwokocha Ethelbert	8000	Ex-Exco	2100	Short Sleeve
Omaji Samuel Owoicho	8000	Ex-Exco	2100	Short Sleeve
Sarah Kauna Edoja	8000	Member	2500	Long Sleeve
Ibrahim Hussein Chado	10000	Member	0	No Sleeve
Fatima Ganiyu	0	Member	0	No Sleeve
Daleng Elisha Nandi	9000	Member	2500	Long Sleeve
Adanu David	9000	Member	0	No Sleeve
Oche Muscle	9500	Member	0	No Sleeve
Caleb Onuoja Aaron	8000	Ex-Exco	2500	Long Sleeve
Abuh E. Rasheed	0	Exco	2100	Short Sleeve
Didi Chukwuebuka V.	0	Exco	2500	Long Sleeve
Jonathan Ajiboye	0	Exco	2100	Short Sleeve
Yusuff Fatorisa	0	Exco	2100	Short Sleeve
Gabriel Alkali	0	Exco	2500	Long Sleeve
Ebube Onuigbo	0	Exco	2100	Short Sleeve
Sanusi Muhammad	0	Exco	2500	Long Sleeve
Ameh Sunday	0	Exco	2100	Short Sleeve
Bawa Vincent	0	Exco	2500	Long Sleeve
Bala Monday	0	Exco	2500	Long Sleeve
Mukaila Abdul-Rafiu	0	Exco	2500	Long Sleeve
Cynthia John	0	Exco	2500	Long Sleeve
Victor Chumnon	0	Exco	2500	Long Sleeve

In [76]:

```
comb_dataset[comb_dataset["Status"] == "Member"]
```

Out[76]:

	Amount Paid	Status	T-Shirt Amount	T-shirt
Name				
Victor T. Na'Allah	12000	Member	2500	Long Sleeve
Olajide Mattew	12000	Member	2500	Long Sleeve
Abel Modu Timothy	12000	Member	2500	Long Sleeve
Egwim Jones Udojuaku	12000	Member	2500	Long Sleeve
Nwachukwu Emmanuel Benedict	12000	Member	2500	Long Sleeve
Adole John A.	12000	Member	2100	Short Sleeve
Faleti Ayodeji Peter	12000	Member	2100	Short Sleeve
Ayantoye Ridwan Ayomide	12000	Member	2500	Long Sleeve
Marvellous T. Isaac	12000	Member	2100	Short Sleeve
Shenge Raphael Saarshatar	12000	Member	2100	Short Sleeve
Reuben O.Enoch	12000	Member	2500	Long Sleeve
Mercy Ajayi	12000	Member	2500	Long Sleeve
Saad	12000	Member	2500	Long Sleeve
Faith	12000	Member	2500	Long Sleeve
Gonet Zion	12000	Member	2500	Long Sleeve
Hawwau Adeboyin Adeyemo\nFor => Jubrin Omeiza	12000	Member	2500	Long Sleeve
Nicholas Otonoku	12000	Member	2100	Short Sleeve
Timothy Ignitus Agbor	12000	Member	2500	Long Sleeve
Madumche Chidibere	12000	Member	2500	Long Sleeve
Effiong Ubon Alasi	12000	Member	2500	Long Sleeve
Daniel Overcomer	12000	Member	2100	Short Sleeve
Aguwa Wisdom	12000	Member	2500	Long Sleeve
Ganiyu Mujeeb	12000	Member	2500	Long Sleeve
Hezekiel Joel	12000	Member	2500	Long Sleeve
Shitu Mustapha Ibrahim	12000	Member	2500	Long Sleeve
Orogu Francis Israel	12000	Member	2100	Short Sleeve
Agha Elizabeth	12000	Member	2100	Short Sleeve
Muhammed Zainab	12000	Member	2500	Long Sleeve
Isaac Priscilla	12000	Member	2100	Short Sleeve
Paul Elizabeth Ladi	12000	Member	2100	Short Sleeve
Esinome Abraham	12000	Member	2100	Short Sleeve
Umeh Audu Ayigba	12000	Member	2100	Short Sleeve
Sarah Kauna Edoja	8000	Member	2500	Long Sleeve

	Amount Paid	Status	T-Shirt Amount	T-shirt
Name				
Ibrahim Hussein Chado	10000	Member	0	No Sleeve
Fatima Ganiyu	0	Member	0	No Sleeve
Daleng Elisha Nandi	9000	Member	2500	Long Sleeve
Ukande Aondongu Cephas	12000	Member	2500	Long Sleeve
De-Gold David Tarki	12000	Member	2500	Long Sleeve
Shekinah Ajibola	12000	Member	2500	Long Sleeve
Adanu David	9000	Member	0	No Sleeve
Oche Muscle	9500	Member	0	No Sleeve
Ocheje Jeremiah	12000	Member	2100	Short Sleeve
Oguntowo Basit Ifedolapo	12000	Member	2500	Long Sleeve
Simeon Iganga	12000	Member	2500	Long Sleeve

In [77]:

```
extract = comb_dataset[comb_dataset.Status == "Member"]
extract.head()
```

Out[77]:

	Amount Paid	Status	T-Shirt Amount	T-shirt
Name				
Victor T. Na'Allah	12000	Member	2500	Long Sleeve
Olajide Mattew	12000	Member	2500	Long Sleeve
Abel Modu Timothy	12000	Member	2500	Long Sleeve
Egwim Jones Udojuaku	12000	Member	2500	Long Sleeve
Nwachukwu Emmanuel Benedict	12000	Member	2500	Long Sleeve

Try using the "set_index()" and "reset_index()" methods and have a feel of advanced techniques of data manipulation

In [78]:

```
extract1 = extract.reset_index()
extract1.head()
```

Out[78]:

	Name	Amount Paid	Status	T-Shirt Amount	T-shirt
0	Victor T. Na'Allah	12000	Member	2500	Long Sleeve
1	Olajide Mattew	12000	Member	2500	Long Sleeve
2	Abel Modu Timothy	12000	Member	2500	Long Sleeve
3	Egwim Jones Udojuaku	12000	Member	2500	Long Sleeve
4	Nwachukwu Emmanuel Benedict	12000	Member	2500	Long Sleeve

In [79]:

```
extract11 = extract1.set_index(["Name","T-shirt"])
extract11.head()
```

Out[79]:

		Amount Paid	Status	T-Shirt Amount
Name	T-shirt			
Victor T. Na'Allah	Long Sleeve	12000	Member	2500
Olajide Mattew	Long Sleeve	12000	Member	2500
Abel Modu Timothy	Long Sleeve	12000	Member	2500
Egwim Jones Udojuaku	Long Sleeve	12000	Member	2500
Nwachukwu Emmanuel Benedict	Long Sleeve	12000	Member	2500

This are just a few of the exciting and amazing this the Python libraries can do to help you achieve your designated tasks.

Remember, the best way to learn is by doing, so

Practice ! Practice !! Practice !!!

I wish you a beautiful ride on this exciting journey.

Happy Learning

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: