Python is really a beautiful and amazing programming language. It's really intuitive and in a simplified form for all who are committed to working with its vast libraries that makes execution of tasks simply

I thought I've experienced it all with Python until i ventured into packages/libraries like numpy pandas, matplotlib, scipy, seaborn, scikit_learn etc. It opened me up to a whole new world. My excitement new no bounds as well as the possibility of what i can do with it. It gave me a whole new learning and practical experience

This session contains my starting point trying out my learning and application of numpy and pandas libraries. And I'll upload more of it in subsequent sessions. The reading text that i used and will recommend asides other video courses was "Python for Data Analysis by Wes McKinney".

Follow with the instructions and you can replicate all I've done and more

Pandas supports loading and reading of files like excel, csv, html or from a database. Each of these requires its special command prompt or code

First, you'll import the pandas library and you'll load the data, which is an excel file. Note that the command used is specifically for loading excel. To load other files require different command. The file is a dummy data from my end.

In [6]:
```
import pandas as pd
file = pd.read_excel("Excursion Withdrawals.xls")
file.head(10)
```

Out[6]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5
0	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	Items	Quantity	Price	Amount
4	NaN	NaN	Fuelling of Coaster Bus	5	13775	68875
5	NaN	NaN	Fuelling of Small Bus	4	10875	43500
6	NaN	NaN	Departmental T-shirt	NaN	40000	40000
7	NaN	NaN	Accommodation (Nnewi)	NaN	32000	32000
8	NaN	NaN	Feeding	NaN	28000	28000
9	NaN	NaN	Renting Of Sound System	4	2000	8000

Note that the file can be loaded in such a way that most of the unwanted rows will be filtered out. But for now, we'll go about that in a counter intuitive manner

The file has been loaded and .head() reads the first 10 lines of the data

The first thing you should do after loading the data is to familiarize yourself with the data by getting several information about it. Examples of that is included below

In [7]:
```
file.shape
```

Out[7]:
```
(16, 6)
```

In [8]:
```
file.size
```

Out[8]:
```
96
```

In [9]:
```
file.dtypes
```

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

In [30]:
```
file.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16 entries, 0 to 15
Data columns (total 6 columns):
Unnamed: 0    0 non-null float64
Unnamed: 1    0 non-null float64
Unnamed: 2    12 non-null object
Unnamed: 3    5 non-null object
Unnamed: 4    11 non-null object
Unnamed: 5    10 non-null object
dtypes: float64(2), object(4)
memory usage: 576.0+ bytes
```

In [31]:
```
file.describe()
```

Out[31]:

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

Now, we'll drop i.e delete the unwanted rows. Note that i assigned the outcome to another variable. I can choose to retain the outcome in the same variable container by setting "inplace" to be True.

In [11]:
```
file1 = file.drop([0,1,2,10,14,15])
```

I checked the outcome of the action to know what the Dataset looks like and have a clue of what next to do

In [12]:
```
file1
```

Out[12]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5
3	NaN	NaN	Items	Quantity	Price	Amount
4	NaN	NaN	Fuelling of Coaster Bus	5	13775	68875
5	NaN	NaN	Fuelling of Small Bus	4	10875	43500
6	NaN	NaN	Departmental T-shirt	NaN	40000	40000
7	NaN	NaN	Accommodation (Nnewi)	NaN	32000	32000
8	NaN	NaN	Feeding	NaN	28000	28000
9	NaN	NaN	Renting Of Sound System	4	2000	8000
11	NaN	NaN	Fixing of Bus	NaN	25000	25000
12	NaN	NaN	Drivers	4	10000	40000
13	NaN	NaN	Miscellaneous	NaN	15000	15000

Next, i deleted some of the unwanted columns. Notice that i set "axis = 1" in the syntax and line of code below

In [13]:
```
file2 = file1.drop(["Unnamed: 0","Unnamed: 1"], axis = 1)
```

In [14]:
```
file2
```

Out[14]:

	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5
3	Items	Quantity	Price	Amount
4	Fuelling of Coaster Bus	5	13775	68875
5	Fuelling of Small Bus	4	10875	43500
6	Departmental T-shirt	NaN	40000	40000
7	Accommodation (Nnewi)	NaN	32000	32000
8	Feeding	NaN	28000	28000
9	Renting Of Sound System	4	2000	8000
11	Fixing of Bus	NaN	25000	25000
12	Drivers	4	10000	40000
13	Miscellaneous	NaN	15000	15000

Now, i renamed the columns of the dataset to make it more sensible and relatable

In [15]:
```
file2.columns = ['ITEMS','QUANTITY', 'PRICE','AMOUNT']
```

In [16]:
```
file2
```

Out[16]:

	ITEMS	QUANTITY	PRICE	AMOUNT
3	Items	Quantity	Price	Amount
4	Fuelling of Coaster Bus	5	13775	68875
5	Fuelling of Small Bus	4	10875	43500
6	Departmental T-shirt	NaN	40000	40000
7	Accommodation (Nnewi)	NaN	32000	32000
8	Feeding	NaN	28000	28000
9	Renting Of Sound System	4	2000	8000
11	Fixing of Bus	NaN	25000	25000
12	Drivers	4	10000	40000
13	Miscellaneous	NaN	15000	15000

And again, I dropped another column that i found irrelevant in the dataset

In [17]:
```
file3 = file2.drop(3)
```

In [18]:
```
file3
```

Out[18]:

	ITEMS	QUANTITY	PRICE	AMOUNT
4	Fuelling of Coaster Bus	5	13775	68875
5	Fuelling of Small Bus	4	10875	43500
6	Departmental T-shirt	NaN	40000	40000
7	Accommodation (Nnewi)	NaN	32000	32000
8	Feeding	NaN	28000	28000
9	Renting Of Sound System	4	2000	8000
11	Fixing of Bus	NaN	25000	25000
12	Drivers	4	10000	40000
13	Miscellaneous	NaN	15000	15000

In [20]:
```
file3.isna().sum()
```

Out[20]:
```
ITEMS       0
QUANTITY    5
PRICE       0
AMOUNT      0
dtype: int64
```

There are some values in the Dataset that are "nan" which means not a number. It's been revealed that 5 null values are present in the quantity column. These values needs to be treated to make the Dataset clean. It's either you drop them or you correct them by replacing them with another value. There are many ways to go about this but in this case, it'll be replaced with a number using the fillna method

In [21]:
```
file4 = file3.fillna(1)
```

In [22]:
```
file4
```

Out[22]:

	ITEMS	QUANTITY	PRICE	AMOUNT
4	Fuelling of Coaster Bus	5	13775	68875
5	Fuelling of Small Bus	4	10875	43500
6	Departmental T-shirt	1	40000	40000
7	Accommodation (Nnewi)	1	32000	32000
8	Feeding	1	28000	28000
9	Renting Of Sound System	4	2000	8000
11	Fixing of Bus	1	25000	25000
12	Drivers	4	10000	40000
13	Miscellaneous	1	15000	15000

In [23]:
```
link = list(file4["ITEMS"])
```

In [24]:
```
link
```

Out[24]:
```
['Fuelling of Coaster Bus',
 'Fuelling of Small Bus',
 'Departmental T-shirt',
 'Accommodation ( Nnewi )',
 'Feeding',
 'Renting Of Sound System',
 'Fixing of Bus',
 'Drivers',
 'Miscellaneous']
```

I gave the dataset a new index label that's more befitting. And i deleted the column that i used

In [25]:
```
file4.index = link
```

In [26]:
```
file4
```

Out[26]:

	ITEMS	QUANTITY	PRICE	AMOUNT
Fuelling of Coaster Bus	Fuelling of Coaster Bus	5	13775	68875
Fuelling of Small Bus	Fuelling of Small Bus	4	10875	43500
Departmental T-shirt	Departmental T-shirt	1	40000	40000
Accommodation (Nnewi)	Accommodation (Nnewi)	1	32000	32000
Feeding	Feeding	1	28000	28000
Renting Of Sound System	Renting Of Sound System	4	2000	8000
Fixing of Bus	Fixing of Bus	1	25000	25000
Drivers	Drivers	4	10000	40000
Miscellaneous	Miscellaneous	1	15000	15000

In [29]:
```
file5 = file4.drop("ITEMS", axis = 1)
```

In [28]:
```
file5
```

Out[28]:

	QUANTITY	PRICE	AMOUNT
Fuelling of Coaster Bus	5	13775	68875
Fuelling of Small Bus	4	10875	43500
Departmental T-shirt	1	40000	40000
Accommodation (Nnewi)	1	32000	32000
Feeding	1	28000	28000
Renting Of Sound System	4	2000	8000
Fixing of Bus	1	25000	25000
Drivers	4	10000	40000
Miscellaneous	1	15000	15000

In [32]:
```
file5.describe()
```

Out[32]:

	QUANTITY	PRICE	AMOUNT
count	9.000000	9.000000	9.000000
mean	2.444444	19627.777778	33375.000000
std	1.740051	12261.572623	17793.959649
min	1.000000	2000.000000	8000.000000
25%	1.000000	10875.000000	25000.000000
50%	1.000000	15000.000000	32000.000000
75%	4.000000	28000.000000	40000.000000
max	5.000000	40000.000000	68875.000000

The Dataset is now clean and ready for analysis. I just checked the Dataset with some methods to. confirm its new state. I checked its info,statistics, null values and data types.

In [33]:
```
file5.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 9 entries, Fuelling of Coaster Bus to Miscellaneous
Data columns (total 3 columns):
QUANTITY    9 non-null int64
PRICE       9 non-null int64
AMOUNT      9 non-null int64
dtypes: int64(3)
memory usage: 252.0+ bytes
```

In [34]:
```
file5.describe()
```

Out[34]:

	QUANTITY	PRICE	AMOUNT
count	9.000000	9.000000	9.000000
mean	2.444444	19627.777778	33375.000000
std	1.740051	12261.572623	17793.959649
min	1.000000	2000.000000	8000.000000
25%	1.000000	10875.000000	25000.000000
50%	1.000000	15000.000000	32000.000000
75%	4.000000	28000.000000	40000.000000
max	5.000000	40000.000000	68875.000000

In [35]:
```
file5.isnull().sum()
```

Out[35]:
```
QUANTITY    0
PRICE       0
AMOUNT      0
dtype: int64
```

In [36]:
```
file5.notnull().sum()
```

Out[36]:
```
QUANTITY    9
PRICE       9
AMOUNT      9
dtype: int64
```

In [37]:
```
file5.dtypes
```

Out[37]:
```
QUANTITY    int64
PRICE       int64
AMOUNT      int64
dtype: object
```

There are several ways and more efficient methods to get this done. This is majorly the basics. In subsequent lessons, you'll learn the advanced methods of data cleaning, data wrangling and data manipulation

The best way to learn is by doing. So practice with these basics and you'll learn advanced techniques in subsequent lessons. ❤️❣️❤️❣️❤️

In [ ]:
```
```