EXPERIMENT NO:2

DATA PREPROCESSING AND VISUALIZATION USING PYTHON

Reading the data:

```
In [4]: import pandas as pd
        import numpy as np
        from pandas import read_csv

In [10]: df = pd.read_csv('tennisdata.csv')

In [11]: print(df)
```

```
                  Player1              Player2  Gender  Round  Result  \
0             Lukas Lacko       Novak Djokovic    Male      1       0
1          Leonardo Mayer       Albert Montanes    Male      1       1
2        Marcos Baghdatis         Denis Istomin    Male      1       0
3         Dmitry Tursunov       Michael Russell    Male      1       1
4             Juan Monaco        Ernests Gulbis    Male      1       0
5        Santiago Giraldo           Sam Querrey    Male      1       0
6              Dudi Sela       Jarkko Nieminen    Male      1       0
7          Fabio Fognini    Alex Bogomolov Jr.    Male      1       1
8            David Guez       Richard Gasquet    Male      1       0
9      Nikolay Davydenko          Lukasz Kubot    Male      1       1
10   Pablo Carreno Busta       Julien Benneteau    Male      1       0
11        Tommy Robredo            Lukas Rosol    Male      1       1
12         Samuel Groth         Vasek Pospisil    Male      1       0
13         Nicolas Mahut          Matthew Ebden    Male      1       0
14       Alejandro Falla       Mikhail Kukushkin    Male      1       1
15     Stanislas Wawrinka          Andrey Golubev    Male      1       1
```

Reading data using head function:

```
In [12]: df.head()
```

Out[12]:

	Player1	Player2	Gender	Round	Result	FNL1	FNL2	FSP.1	FSW.1	SSP.1	...	BPW.2	NPA.2	NPW.2	TPW.2	ST1.2	ST2.2	ST3.2	ST4.2	ST5.2	UF
0	Lukas Lacko	Novak Djokovic	Male	1	0	0.0	3	61	35	39	...	8	8.0	9.0	101.0	6	7	6.0	NaN	NaN	
1	Leonardo Mayer	Albert Montanes	Male	1	1	3.0	0	61	31	39	...	0	NaN	NaN	42.0	1	3	1.0	NaN	NaN	
2	Marcos Baghdatis	Denis Istomin	Male	1	0	0.0	3	52	53	48	...	13	12.0	16.0	126.0	6	7	6.0	NaN	NaN	
3	Dmitry Tursunov	Michael Russell	Male	1	1	3.0	0	53	39	47	...	7	NaN	NaN	79.0	2	2	3.0	NaN	NaN	
4	Juan Monaco	Ernests Gulbis	Male	1	0	1.0	3	76	63	24	...	5	16.0	28.0	127.0	1	6	7.0	6.0	NaN	

Statistics of the data set:

```
In [14]: df.describe()
```

Out[14]:

	Round	Result	FNL1	FNL2	FSP.1	FSW.1	SSP.1	SSW.1	ACE.1	DBF.1	...	BPW.2	NPA.2	1
count	127.000000	127.000000	126.000000	127.000000	127.000000	127.000000	127.000000	127.000000	124.000000	126.000000	...	127.000000	108.000000	
mean	1.944882	0.551181	1.595238	1.393701	61.118110	37.314961	38.881890	16.897638	6.895161	3.833333	...	6.070866	11.453704	
std	1.274332	0.499343	1.089430	1.169435	7.817983	16.885563	7.817983	8.683897	7.403382	2.750273	...	4.445760	7.558352	
min	1.000000	0.000000	0.000000	0.000000	42.000000	8.000000	18.000000	1.000000	0.000000	0.000000	...	0.000000	0.000000	
25%	1.000000	0.000000	1.000000	0.000000	56.000000	24.000000	34.000000	10.500000	2.000000	2.000000	...	3.000000	6.000000	
50%	1.000000	1.000000	2.000000	1.000000	61.000000	35.000000	39.000000	15.000000	4.500000	3.000000	...	5.000000	10.000000	
75%	2.000000	1.000000	2.000000	2.000000	66.000000	49.500000	44.000000	23.000000	9.000000	6.000000	...	8.000000	15.000000	
max	7.000000	1.000000	3.000000	3.000000	82.000000	109.000000	58.000000	43.000000	41.000000	13.000000	...	20.000000	37.000000	

8 rows × 40 columns

Getting Datatypes of Dataset:

```
In [16]: df.dtypes
```

Out[16]:

```
Player1     object
Player2     object
Gender      object
Round        int64
Result       int64
FNL1       float64
FNL2         int64
FSP.1        int64
FSW.1        int64
SSP.1        int64
SSW.1        int64
ACE.1      float64
DBF.1      float64
WNR.1        int64
UFE.1        int64
BPC.1        int64
BPW.1        int64
NPA.1      float64
NPW.1      float64
TPW.1        int64
ST1.1        int64
ST2.1        int64
ST3.1      float64
ST4.1      float64
ST5.1      float64
FSP.2        int64
FSW.2        int64
```

Cleaning of the data

Finding out missing values and count them

```
In [13]: df.isnull().sum()
```

```
Out[13]: Player1      0
         Player2      0
         Gender       0
         Round        0
         Result       0
         FNL1         1
         FNL2         0
         FSP.1        0
         FSW.1        0
         SSP.1        0
         SSW.1        0
         ACE.1        3
         DBF.1        1
         WNR.1        0
         UFE.1        0
         BPC.1        0
         BPW.1        0
         NPA.1       21
         NPW.1       21
         TPW.1        0
         ST1.1        0
         ST2.1        0
```

Converting categorical strings to numbers

```
cat_columns = ['Player1', 'Gender']
# df[cat_columns].apply(lambda x: x.cat.codes)
df.head()
```

	Player1	Player2	Gender	Round	Result	FNL1	FNL2	FSP.1	FSW.1	SSP.1	...	BPW.2	NPA.2	NPW.2	TPW.2	ST1.2	ST2.2	ST3.2	ST4.2	ST5.2	UF
0	Lukas Lacko	Novak Djokovic	Male	1	0	0.0	3	61	35	39	...	8	8.0	9.0	101.0	6	7	6.0	NaN	NaN	
1	Leonardo Mayer	Albert Montanes	Male	1	1	3.0	0	61	31	39	...	0	NaN	NaN	42.0	1	3	1.0	NaN	NaN	
2	Marcos Baghdatis	Denis Istomin	Male	1	0	0.0	3	52	53	48	...	13	12.0	16.0	126.0	6	7	6.0	NaN	NaN	
3	Dmitry Tursunov	Michael Russell	Male	1	1	3.0	0	53	39	47	...	7	NaN	NaN	79.0	2	2	3.0	NaN	NaN	
4	Juan Monaco	Ernests Gulbis	Male	1	0	1.0	3	76	63	24	...	5	16.0	28.0	127.0	1	6	7.0	6.0	NaN	

5 rows × 43 columns

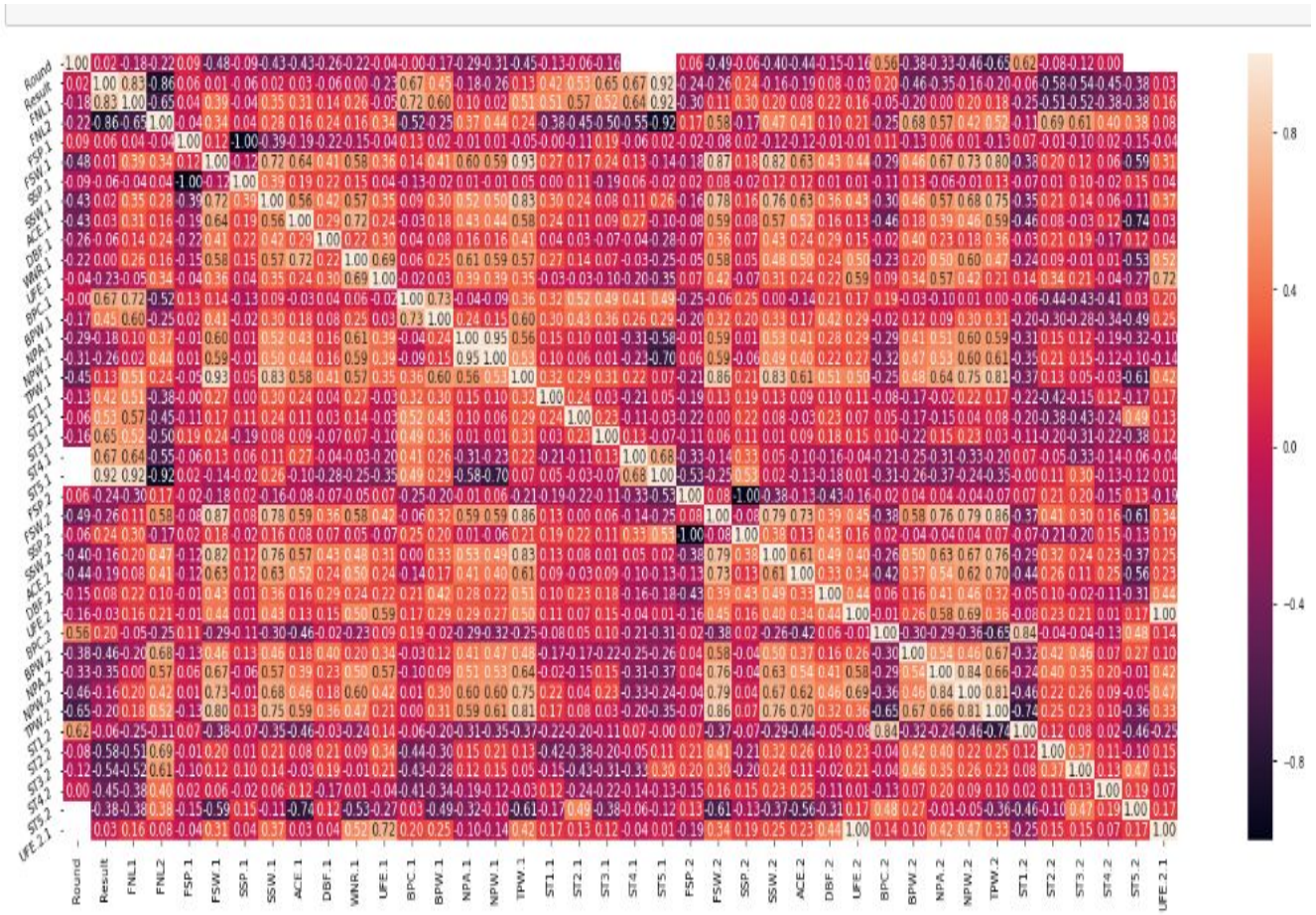
Replacing all the NaN values with mean value :

```
In [17]: df.fillna(df.mean(),inplace=False)
```

	Player1	Player2	Gender	Round	Result	FNL1	FNL2	FSP.1	FSW.1	SSP.1	...	BPW.2	NPA.2	NPW.2	TPW.2	ST1.2	ST2.2	ST3.2	ST4.2
0	Lukas Lacko	Novak Djokovic	Male	1	0	0.000000	3	61	35	39	...	8	8.000000	9.000000	101.0	6	7	6.000000	4.226415
1	Leonardo Mayer	Albert Montanes	Male	1	1	3.000000	0	61	31	39	...	0	11.453704	14.669811	42.0	1	3	1.000000	4.226415
2	Marcos Baghdatis	Denis Istomin	Male	1	0	0.000000	3	52	53	48	...	13	12.000000	16.000000	126.0	6	7	6.000000	4.226415
3	Dmitry Tursunov	Michael Russell	Male	1	1	3.000000	0	53	39	47	...	7	11.453704	14.669811	79.0	2	2	3.000000	4.226415
4	Juan Monaco	Ernests Gulbis	Male	1	0	1.000000	3	76	63	24	...	5	16.000000	28.000000	127.0	1	6	7.000000	6.000000
5	Santiago Giraldo	Sam Querrey	Male	1	0	1.000000	3	65	51	35	...	7	14.000000	17.000000	122.0	6	6	3.000000	7.000000
6	Dudi Sela	Jarkko Nieminen	Male	1	0	2.000000	3	68	73	32	...	17	25.000000	36.000000	173.0	3	7	6.000000	6.000000

Coorelation:

```
import seaborn as sns
corr= df.corr()
sns.set_context("notebook", font_scale=1.0, rc={"lines.linewidth": 3})
plt.figure(figsize=(13,7))
a = sns.heatmap(corr, annot=True, fmt='.2f')
rotx = a.set_xticklabels(a.get_xticklabels(), rotation=90)
roty = a.set_yticklabels(a.get_yticklabels(), rotation=30)
```



Finding Outliers:

```
def find_outliers(x):
q1=x.quantile(.25)
q3=x.quantile(.75)
iqr=q3-q1
floor=q1-1.5*iqr
ceiling=q3+1.5*iqr
outliers_indices=list(x.index[(x<floor) | (x>ceiling)])
outliers_values=list(x[outliers_indices])
```

```
Round_indices,Round_values = find_outliers(df['Round'])
print("Outliers for Round")
print(np.sort(Round_values))
```

```
Outliers for Round
[4 4 4 4 4 4 4 4 5 5 5 5 6 6 7]
```

```
return outliers_indices,outliers_values
```

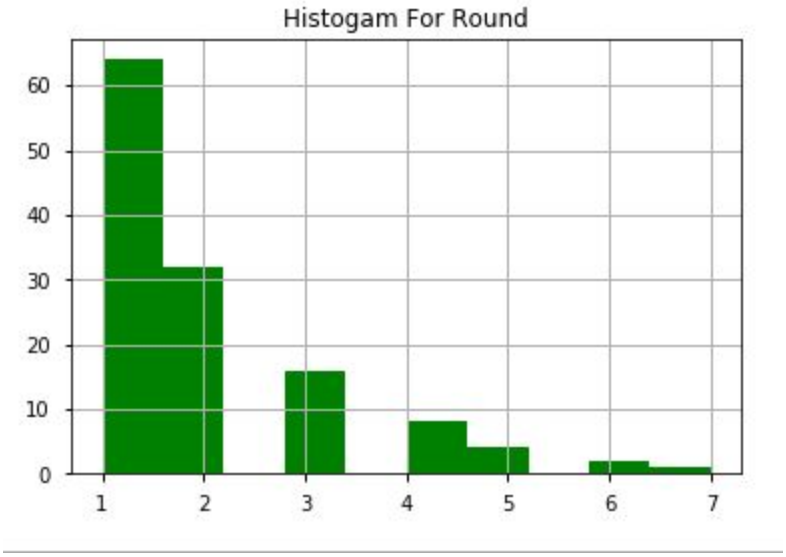
```
In [34]: FNL2_indices,FNL2_values = find_outliers(df['FNL2'])
         print("Outliers for FNL2")
         print(np.sort(FNL2_values))
```

```
Outliers for FNL2
[]
```

Visualization:

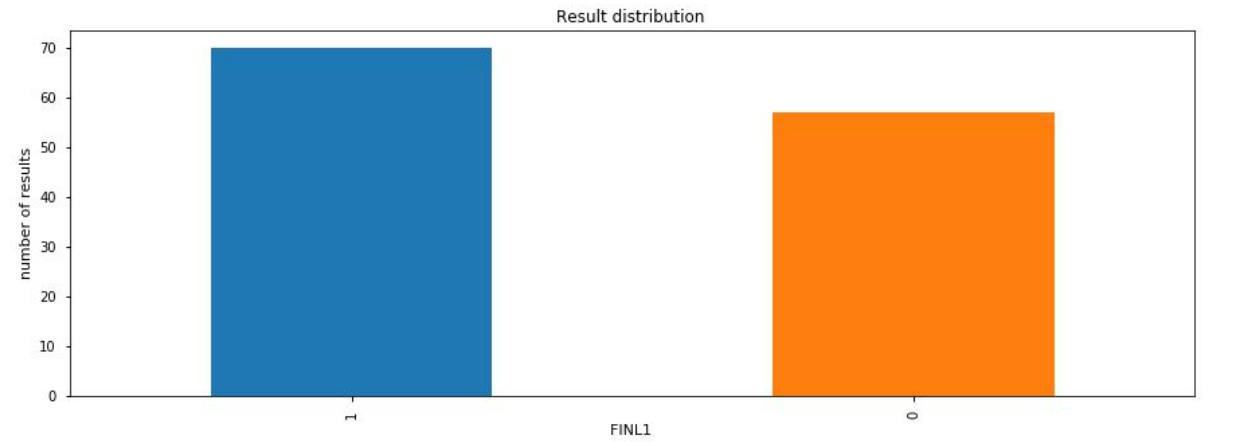
1.

```
df['Round'].hist(color='green')
plt.title("Histogam For Round");
```



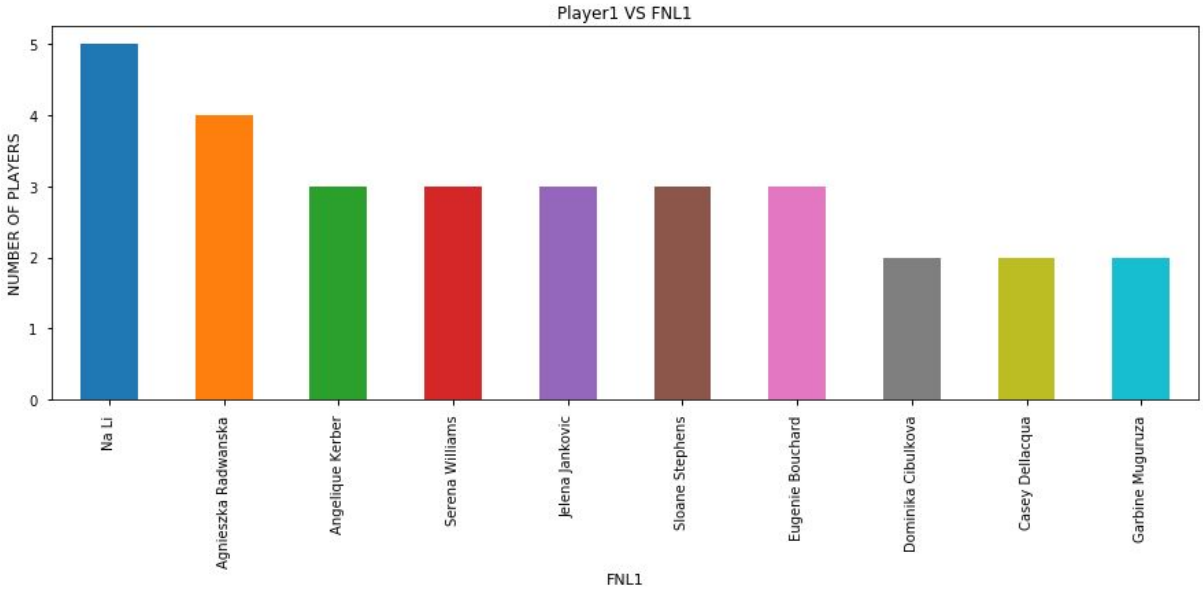
EXPLANATION :It is determined from above figure that round 1 and 2 has maximum players and 6 and 7 has very less number of players

2. df.Result.value_counts().nlargest(2).plot(kind='bar',
figsize=(15,5))
plt.title("Result distribution ")
plt.ylabel('number of results')
plt.xlabel('FINL1 ');



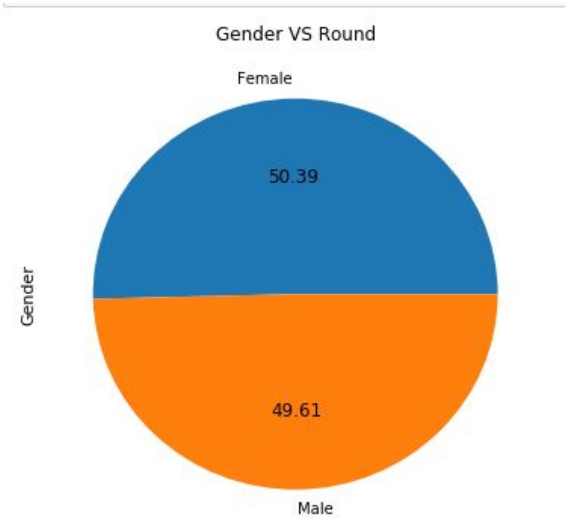
EXPLANATION : It can be determined from the diagram that the number of successfull players is a more than the number of unsuccessful players.

3. df.Player1.value_counts().nlargest(10).plot(kind='bar', figsize=(15,5))
plt.title("Player1 VS FNL1")
plt.ylabel('NUMBER OF PLAYERS')
plt.xlabel('FNL1');



EXPLANATION : The above diagram is for player1 vs FINL1 which determines the number of players played for different rounds in FINL1

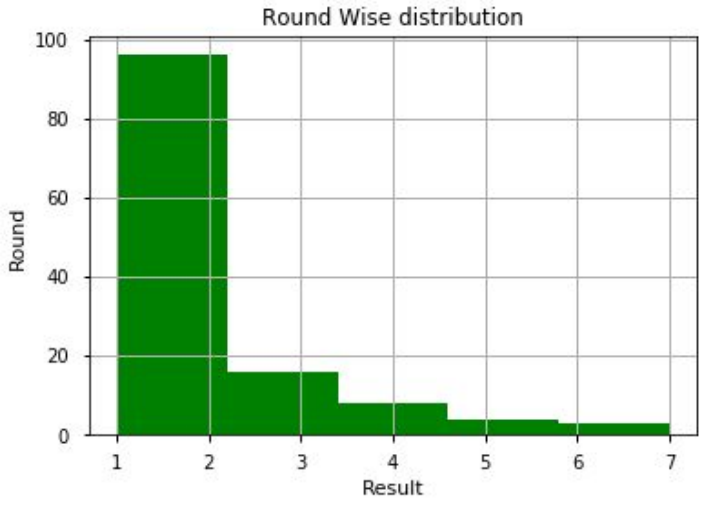
4. data['Gender'].value_counts().plot.pie(figsize=(6,6),autopct='%.2f')
plt.title('Gender VS Round');



EXPLANATION : From above diagram we determine that no of female players in round1 are more than the no of male players

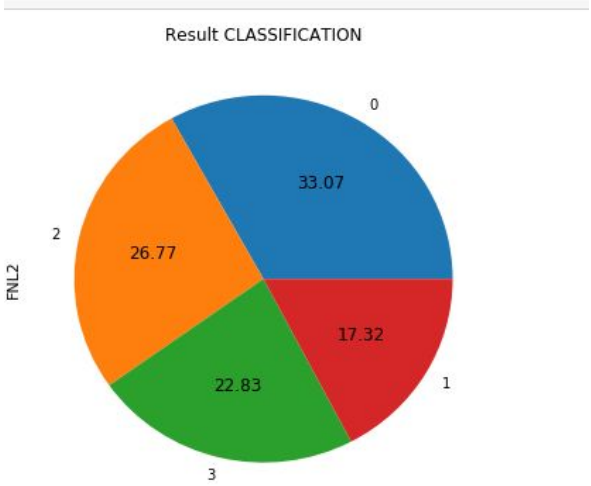
5. df.Round.hist(bins=5,color='green');
plt.title("Round Wise distribution")
plt.ylabel('Round')
plt.xlabel('Result')

```
Text(0.5,0,u'Result')
```



EXPLANATION : From above diagram we can determine distribution of players across all 7 rounds.we see that players are decreasing gradually till last round

6. df['FNL2'].value_counts().plot.pie(figsize=(6,6),autopct='%.2f')
plt.title('Result CLASSIFICATION');



EXPLANATION: The Figure determines for Player2 and FINL2.From above diagram we determine the number of games won by player 2 .