In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
df= pd.read_csv('hospital_data.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	Date	Medication Revenue	Lab Cost	Consultation Revenue	Doctor Type	Financial Class	Patient Type	Entry Time	Co
0	11/4/2019 0:00	$1,183.22	$10.00	$20.17	ANCHOR	HMO	OUTPATIENT	8:35:45	
1	11/6/2019 0:00	$738.48	$-	$15.00	ANCHOR	INSURANCE	OUTPATIENT	19:19:16	
2	11/2/2019 0:00	$660.00	$-	$21.17	ANCHOR	HMO	OUTPATIENT	10:46:52	
3	11/6/2019 0:00	$600.00	$-	$-	ANCHOR	MEDICARE	OUTPATIENT	9:38:34	
4	11/1/2019 0:00	$591.60	$-	$12.00	ANCHOR	INSURANCE	OUTPATIENT	11:16:21	

In [4]:

```
df.tail()
```

Out[4]:

	Date	Medication Revenue	Lab Cost	Consultation Revenue	Doctor Type	Financial Class	Patient Type	Entry Time
29994	11/13/2019 0:00	$-	$-	$-	ANCHOR	PRIVATE	OUTPATIENT	14:01:24

	Date	Medication Revenue	Lab Cost	Consultation Revenue	Doctor Type	Financial Class	Patient Type	Entry Time
29995	11/13/2019 0:00	$-	$-	$-	ANCHOR	PRIVATE	OUTPATIENT	14:01:33
29996	11/13/2019 0:00	$-	$-	$-	ANCHOR	PRIVATE	OUTPATIENT	14:01:41
29997	11/13/2019 0:00	$-	$-	$-	LOCUM	CORPORATE	OUTPATIENT	14:01:42
29998	11/13/2019 0:00	$-	$-	$-	ANCHOR	PRIVATE	OUTPATIENT	14:01:55

In [5]:

```
df.shape
```

Out[5]:

```
(29999, 12)
```

In [6]:

```
df.describe()
```

Out[6]:

	Date	Medication Revenue	Lab Cost	Consultation Revenue	Doctor Type	Financial Class	Patient Type	Entr Tim
count	29999	29999	29999	29999	29999	29999	29999	2999
unique	13	4280	200	241	3	5	1	2062
top	11/11/2019 0:00	$-	$-	$-	ANCHOR	INSURANCE	OUTPATIENT	18:07:4
freq	3618	11937	28566	5576	21914	9931	29999	

In [7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29999 entries, 0 to 29998
Data columns (total 12 columns):
 #   Column                    Non-Null Count  Dtype
```

```
---  ------                  --------------  -----
 0   Date                    29999 non-null  object
 1    Medication Revenue     29999 non-null  object
 2     Lab Cost              29999 non-null  object
 3    Consultation Revenue   29999 non-null  object
 4   Doctor Type             29999 non-null  object
 5   Financial Class         29999 non-null  object
 6   Patient Type            29999 non-null  object
 7   Entry Time              29999 non-null  object
 8   Post-Consultation Time  29999 non-null  object
 9   Completion Time         29999 non-null  object
 10  Doctor                  29999 non-null  object
 11  Patient ID              29999 non-null  object
dtypes: object(12)
memory usage: 2.7+ MB
```

In [8]:

```
df.dtypes
```

Out[8]:

```
Date                          object
 Medication Revenue           object
  Lab Cost                    object
 Consultation Revenue         object
Doctor Type                   object
Financial Class               object
Patient Type                  object
Entry Time                    object
Post-Consultation Time        object
Completion Time               object
Doctor                        object
Patient ID                    object
dtype: object
```

In [9]:

```
df['Date'] = pd.to_datetime(df['Date'])
```

In [10]:

```
df.dtypes
```

Out[10]:

```
Date                          datetime64[ns]
 Medication Revenue                   object
```

```
  Lab Cost                           object
 Consultation Revenue                object
Doctor Type                          object
Financial Class                      object
Patient Type                         object
Entry Time                           object
Post-Consultation Time               object
Completion Time                      object
Doctor                               object
Patient ID                           object
dtype: object
```

In [11]:

```
df.columns
```

Out[11]:

```
Index(['Date', ' Medication Revenue ', '  Lab Cost ', ' Consultation Re
venue ',
       'Doctor Type', 'Financial Class', 'Patient Type', 'Entry Time',
       'Post-Consultation Time', 'Completion Time', 'Doctor', 'Patient
ID'],
      dtype='object')
```

In [12]:

```
df.columns=df.columns.str.strip()
```

In [13]:

```
df.columns
```

Out[13]:

```
Index(['Date', 'Medication Revenue', 'Lab Cost', 'Consultation Revenu
e',
       'Doctor Type', 'Financial Class', 'Patient Type', 'Entry Time',
       'Post-Consultation Time', 'Completion Time', 'Doctor', 'Patient
ID'],
      dtype='object')
```

In [14]:

```
df['Entry Time'] = pd.to_datetime(df['Entry Time'])
df['Post-Consultation Time'] = pd.to_datetime(df['Post-Consultation Tim
e'])
df['Completion Time'] = pd.to_datetime(df['Completion Time'])
df.dtypes
```

Out[14]:

```
Date                          datetime64[ns]
Date                          datetime64[ns]
Medication Revenue                    object
Lab Cost                              object
Consultation Revenue                  object
Doctor Type                           object
Financial Class                       object
Patient Type                          object
Entry Time                    datetime64[ns]
Post-Consultation Time        datetime64[ns]
Completion Time               datetime64[ns]
Doctor                                object
Patient ID                            object
dtype: object
```

In [15]:

```
def clean_text(x):
    x=x.replace('$','')
    x=x.replace(',','')
    x=x.replace('-','')
    x=x.replace(' ','')
    x=pd.to_numeric(x)
    return x
df['Medication Revenue']=df['Medication Revenue'].apply(clean_text)
df.head()
```

Out[15]:

	Date	Medication Revenue	Lab Cost	Consultation Revenue	Doctor Type	Financial Class	Patient Type	Entry Time	Consu
0	2019-11-04	1183.22	$10.00	$20.17	ANCHOR	HMO	OUTPATIENT	2021-01-08 08:35:45	2021 09
1	2019-11-06	738.48	$-	$15.00	ANCHOR	INSURANCE	OUTPATIENT	2021-01-08 19:19:16	2021 2
2	2019-11-02	660.00	$-	$21.17	ANCHOR	HMO	OUTPATIENT	2021-01-08 10:46:52	2021 1

	Date	Medication Revenue	Lab Cost	Consultation Revenue	Doctor Type	Financial Class	Patient Type	Entry Time	Consu
3	2019-11-06	600.00	$-	$-	ANCHOR	MEDICARE	OUTPATIENT	2021-01-08 09:38:34	2021 1(
4	2019-11-01	591.60	$-	$12.00	ANCHOR	INSURANCE	OUTPATIENT	2021-01-08 11:16:21	2021 12

In [16]:

```
df['Lab Cost']=df['Lab Cost'].apply(clean_text)
df['Consultation Revenue']=df['Consultation Revenue'].apply(clean_text)
df.dtypes
```

Out[16]:

```
Date                      datetime64[ns]
Medication Revenue               float64
Lab Cost                         float64
Consultation Revenue             float64
Doctor Type                       object
Financial Class                   object
Patient Type                      object
Entry Time                datetime64[ns]
Post-Consultation Time    datetime64[ns]
Completion Time           datetime64[ns]
Doctor                            object
Patient ID                        object
dtype: object
```

In [17]:

```
df.head()
```

Out[17]:

	Date	Medication Revenue	Lab Cost	Consultation Revenue	Doctor Type	Financial Class	Patient Type	Entry Time	P Consulta
0	2019-11-04	1183.22	10.0	20.17	ANCHOR	HMO	OUTPATIENT	2021-01-08 08:35:45	2021-0 09:1

	Date	Medication Revenue	Lab Cost	Consultation Revenue	Doctor Type	Financial Class	Patient Type	Entry Time	Consulta
1	2019-11-06	738.48	NaN	15.00	ANCHOR	INSURANCE	OUTPATIENT	2021-01-08 19:19:16	2021-0 21:0
2	2019-11-02	660.00	NaN	21.17	ANCHOR	HMO	OUTPATIENT	2021-01-08 10:46:52	2021-0 11:5
3	2019-11-06	600.00	NaN	NaN	ANCHOR	MEDICARE	OUTPATIENT	2021-01-08 09:38:34	2021-0 10:5
4	2019-11-01	591.60	NaN	12.00	ANCHOR	INSURANCE	OUTPATIENT	2021-01-08 11:16:21	2021-0 12:0

In [18]:

```
df['Total Revenue']=df['Medication Revenue']+df['Consultation Revenue']
+df['Lab Cost']
df.head()
```

Out[18]:

	Date	Medication Revenue	Lab Cost	Consultation Revenue	Doctor Type	Financial Class	Patient Type	Entry Time	Consulta
0	2019-11-04	1183.22	10.0	20.17	ANCHOR	HMO	OUTPATIENT	2021-01-08 08:35:45	2021-0 09:1
1	2019-11-06	738.48	NaN	15.00	ANCHOR	INSURANCE	OUTPATIENT	2021-01-08 19:19:16	2021-0 21:0
2	2019-11-02	660.00	NaN	21.17	ANCHOR	HMO	OUTPATIENT	2021-01-08 10:46:52	2021-0 11:5
3	2019-11-06	600.00	NaN	NaN	ANCHOR	MEDICARE	OUTPATIENT	2021-01-08 09:38:34	2021-0 10:5

	Date	Medication Revenue	Lab Cost	Consultation Revenue	Doctor Type	Financial Class	Patient Type	Entry Time	F Consulta
4	2019-11-01	591.60	NaN	12.00	ANCHOR	INSURANCE	OUTPATIENT	2021-01-08 11:16:21	2021-0 12:0

In [19]:
```
df=df.fillna(0)
df.head()
```

Out[19]:

	Date	Medication Revenue	Lab Cost	Consultation Revenue	Doctor Type	Financial Class	Patient Type	Entry Time	F Consulta
0	2019-11-04	1183.22	10.0	20.17	ANCHOR	HMO	OUTPATIENT	2021-01-08 08:35:45	2021-0 09:1
1	2019-11-06	738.48	0.0	15.00	ANCHOR	INSURANCE	OUTPATIENT	2021-01-08 19:19:16	2021-0 21:0
2	2019-11-02	660.00	0.0	21.17	ANCHOR	HMO	OUTPATIENT	2021-01-08 10:46:52	2021-0 11:5
3	2019-11-06	600.00	0.0	0.00	ANCHOR	MEDICARE	OUTPATIENT	2021-01-08 09:38:34	2021-0 10:5
4	2019-11-01	591.60	0.0	12.00	ANCHOR	INSURANCE	OUTPATIENT	2021-01-08 11:16:21	2021-0 12:0

What time has the most visitors?

In [20]:
```
df['Entry Hour']=df['Entry Time'].dt.hour
```

```
df.head()
```

Out[20]:

	Date	Medication Revenue	Lab Cost	Consultation Revenue	Doctor Type	Financial Class	Patient Type	Entry Time	Consulta
0	2019-11-04	1183.22	10.0	20.17	ANCHOR	HMO	OUTPATIENT	2021-01-08 08:35:45	2021-0 09:
1	2019-11-06	738.48	0.0	15.00	ANCHOR	INSURANCE	OUTPATIENT	2021-01-08 19:19:16	2021-0 21:
2	2019-11-02	660.00	0.0	21.17	ANCHOR	HMO	OUTPATIENT	2021-01-08 10:46:52	2021-0 11:
3	2019-11-06	600.00	0.0	0.00	ANCHOR	MEDICARE	OUTPATIENT	2021-01-08 09:38:34	2021-0 10:
4	2019-11-01	591.60	0.0	12.00	ANCHOR	INSURANCE	OUTPATIENT	2021-01-08 11:16:21	2021-0 12:

In [21]:

```
Most_Visitors = df.groupby(df['Entry Hour'])['Patient ID'].count()
Most_Visitors
```

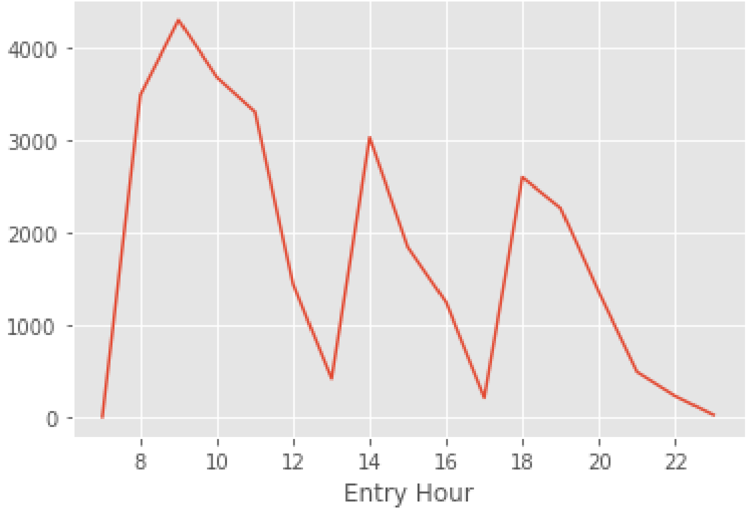
Out[21]:

```
Entry Hour
7       12
8     3489
9     4297
10    3680
11    3306
12    1446
13     426
14    3030
15    1847
16    1249
17     219
18    2600
```

```
19    2269
20    1358
21     500
22     238
23      33
Name: Patient ID, dtype: int64
```

In [22]:

```
plt.style.use('ggplot')
Most_Visitors = df.groupby(df['Entry Hour'])['Patient ID'].count().plot
()
```

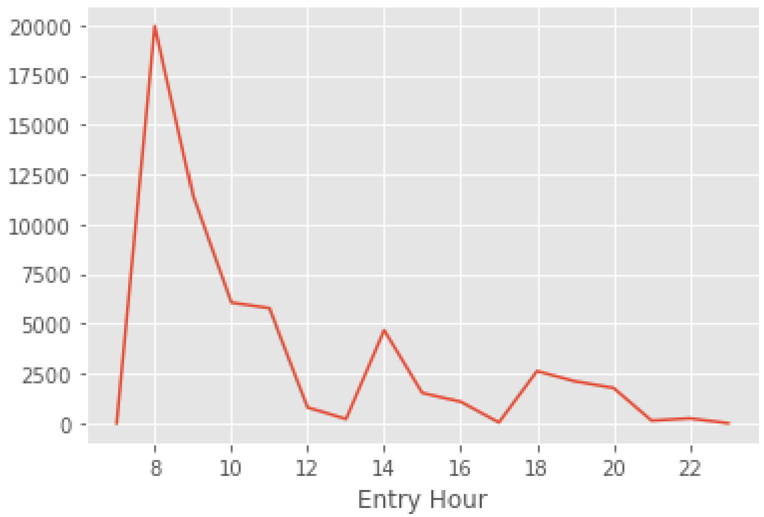


By visualizing,we can conclude that the maximum number of patients visiting the clinic is around 9AM(i.e when the clinic opens).This implies that there are around 4300 visitor patients.This figure drops at the 5PM.

Does the Total Revenue corelate with the Total Visitors?

In [23]:

```
RV=df.groupby(df['Entry Hour'])['Total Revenue'].sum().plot()
```



In [24]:

```
case1=df.groupby(df['Entry Hour']).sum()
case2=df.groupby(df['Entry Hour']).count()
merge = case1.merge(case2, on='Entry Hour')
merge
```

Out[24]:

	Medication Revenue_x	Lab Cost_x	Consultation Revenue_x	Total Revenue_x	Date	Medication Revenue_y	Lab Cost_y	Consultation Revenue_y	D
Entry Hour									
7	350.66	520.00	198.63	0.00	12	12	12	12	
8	83404.62	22475.27	51180.15	19959.28	3489	3489	3489	3489	
9	96195.61	14879.28	60216.56	11429.42	4297	4297	4297	4297	
10	75239.66	6688.98	52072.67	6060.60	3680	3680	3680	3680	
11	72675.83	6860.23	48192.03	5780.90	3306	3306	3306	3306	
12	25755.63	2614.32	21251.02	797.19	1446	1446	1446	1446	
13	7773.26	1125.83	7633.83	213.32	426	426	426	426	

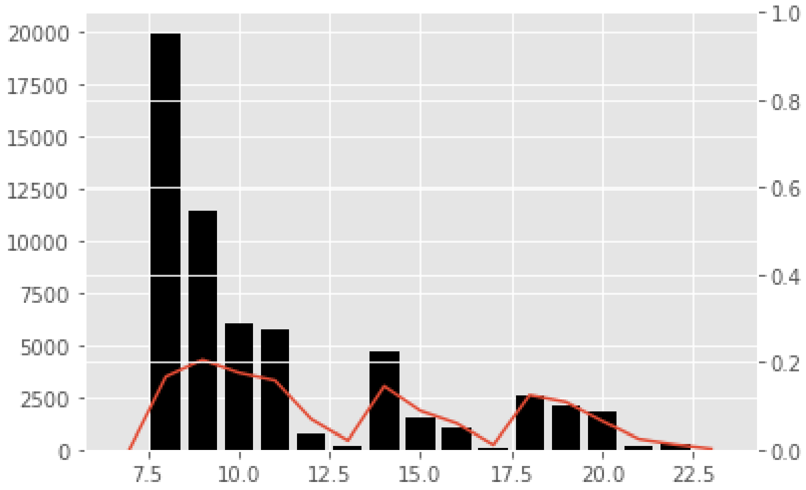
	Medication Revenue_x	Lab Cost_x	Consultation Revenue_x	Total Revenue_x	Date	Medication Revenue_y	Lab Cost_y	Consultation Revenue_y
Entry Hour								
14	59035.21	4946.53	42690.42	4696.92	3030	3030	3030	3030
15	37054.80	1959.22	26703.60	1531.58	1847	1847	1847	1847
16	23029.02	803.09	19004.51	1082.93	1249	1249	1249	1249
17	3729.36	192.21	3673.13	40.10	219	219	219	219
18	51938.32	1377.79	43156.58	2629.96	2600	2600	2600	2600
19	46518.20	1337.78	37812.06	2115.09	2269	2269	2269	2269
20	25264.62	747.12	23424.94	1779.10	1358	1358	1358	1358
21	7387.60	250.58	11693.07	133.15	500	500	500	500
22	3284.33	92.00	7822.08	246.74	238	238	238	238
23	423.31	0.00	1291.45	0.00	33	33	33	33

In [25]:

```
fig, ax1=plt.subplots()
ax2=ax1.twinx()
ax1.bar(merge.index,merge['Total Revenue_x'],color='black')
ax1.plot(merge.index,merge['Patient ID'])
```

Out[25]:

```
[<matplotlib.lines.Line2D at 0x1cea5bb9f70>]
```

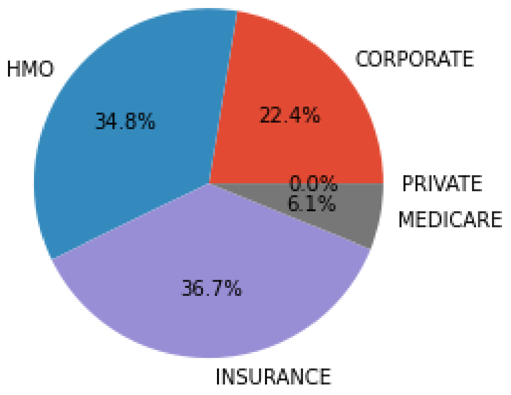


By this we can say that the Total Revenue get increased at the entry hour of the clinic and as well at around 2PM.The graph explains that the Total Revenue and the Total Visitors correlate well with eachother.

What percentage of revenue comes from different Insurance Types ?

In [26]:

```
pivot=df.groupby('Financial Class').sum()
plt.pie(pivot['Total Revenue'],labels=pivot.index,autopct='%1.1f%%');
```



The figure tells us that more revenue was generated through INSURANCE which constitutes around 36.7%, whereas HMO and CORPORATE generated around 34.8% and 22.4%. The lowest revenue is from MEDICARE.

```
In [ ]:
```