In [5]:

```
import pandas as pd
import numpy as np
import matplotlib as mp
import seaborn as snb
import math
import matplotlib.pyplot as plt
from datetime import date

!pip install -U pandasql
from pandasql import sqldf
mysql=lambda q: sqldf(q,globals())

Patients = pd.read_excel("HospitalDatabase .xlsx","Patients")
EDVisits = pd.read_excel("HospitalDatabase .xlsx","EDVisits")
AmbulatoryVisits = pd.read_excel("HospitalDatabase .xlsx","Ambulatory
ReAdmissionRegistry = pd.read_excel("HospitalDatabase .xlsx","ReAdmi
Discharges = pd.read_excel("HospitalDatabase .xlsx","Discharges")
Providers = pd.read_excel("HospitalDatabase .xlsx","Providers")

EDUnique = pd.read_excel("HospitalDatabase .xlsx","EDUnique")
Merged = pd.merge(Patients, Discharges, on="PatientID")
```

```
Merged1= pd.merge(Patients, ReAdmissionRegistry, on="PatientID", how=
```

```
Requirement already satisfied: pandasql in c:\users\gb
haskaran\anaconda3\lib\site-packages (0.7.3)
Requirement already satisfied: sqlalchemy in c:\users
\gbhaskaran\anaconda3\lib\site-packages (from pandasq
l) (1.4.32)
Requirement already satisfied: numpy in c:\users\gbhas
karan\anaconda3\lib\site-packages (from pandasql) (1.2
1.5)
Requirement already satisfied: pandas in c:\users\gbha
skaran\anaconda3\lib\site-packages (from pandasql) (1.
4.2)
Requirement already satisfied: pytz>=2020.1 in c:\user
s\gbhaskaran\anaconda3\lib\site-packages (from pandas-
>pandasql) (2021.3)
Requirement already satisfied: python-dateutil>=2.8.1
in c:\users\gbhaskaran\anaconda3\lib\site-packages (fr
om pandas->pandasql) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\gb
```

```
haskaran\anaconda3\lib\site-packages (from python-date
util>=2.8.1->pandas->pandasql) (1.16.0)
Requirement already satisfied: greenlet!=0.4.17 in
c:\users\gbhaskaran\anaconda3\lib\site-packages (from
sqlalchemy->pandasql) (1.1.1)
```

In [12]:

```
#####58. Display total count of patients service wise based on gender
df = Merged.groupby(['Service','Gender']).size().unstack(fill_value=0
df.head()
```

Out[12]:

Gender	Female	Male
Service		
Cardiology	41	54
General Medicine	114	149
Hospitalist	35	66
ICU	65	86
Neurology	34	31

In [30]:

```
##42.Using loc , get the details of the providers where providerId is

##Providers.iloc [11:21]
##df1 = Providers[(Providers["ProviderSpecialty"] == 'Surgery')]
df1 = Providers.loc[(Providers["ProviderSpecialty"] == 'Surgery')]
```

```
Number of Rows count is: 7
```

arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10], [11, 12, 13, 14, 15], [16, 17, 18, 19, 20], [21, 22, 23, 24, 25], [26, 27, 28, 29, 30]]) print(arr[2, 0:2], [3, 0:2])

In [71]:

```
#####34. np.arange(1,31).reshape(6,5) Find the array slicing to get t
arr=np.arange(1,31).reshape(6,5)
print(arr[2:4,0:2])
```

```
[[11 12]
 [16 17]]
```

In [81]:

```
arr=np.arange(1,31).reshape(6,5)
print(arr[0::1,1::3])
```

```
[[ 2  5]
 [ 7 10]
 [12 15]
 [17 20]
 [22 25]
 [27 30]]
```

In [94]:

```
###12. Connect to sql and write a query to get list of Provider names
mysql = lambda q: sqldf(q,globals())
mysql("Select ProviderName from Providers where ProviderName LIKE 'T%
```

Out[94]:

	ProviderName
0	Ted Texas
1	Ted Green
2	Ted Black
3	Tyler Conner
4	Tony Creed
5	Trent Tye

In [96]:

```
##64. "Using numpy functions, multiply the following arrays a=np.aran

a=np.arange(6).reshape(2,3)
b=np.arange(6).reshape(3,2)
res = np.dot(a,b)
print(res)
```

```
[[10 13]
 [28 40]]
```

In [167]:

```
##69Details of the patients whose firstname or lastname contains str

mysql = lambda q: sqldf(q,globals())
mysql("Select FirstName,LastName from Patients where FirstName LIKE (
```

Out[167]:

	FirstName	LastName
0	Lauren	Gaskal
1	Lauren	Foort
2	Zulauf	Ellingham
3	Zulauf	LLC
4	Zulauf	Alvar
5	Zulauf	Manske
6	Zulauf	Bitcheno
7	Zulauf	O'Shavlan

	FirstName	LastName
8	Lemmy	Klausen
9	Jerrilyn	Klausen
10	Zulauf	Orbine

In [183]:

```
irstName, LastName, DateOfBirth of the Patients whose reason for visi
 q: sqldf(q,globals())

 FirstName, LastName, DateOfBirth from Patients p inner join EDUnique
```

Out[183]:

	FirstName	LastName	DateOfBirth
0	Zonnya	Ab	1963-05-23 10:04:33.074000
1	Gan	Yu	1970-06-02 06:22:54.675000
2	Devlin	Michael	1976-04-15 02:52:09.762000
3	Joesph	Long	1979-12-04 16:45:56.080000
4	Gabriel	Joseph	1986-05-31 09:36:05.716000
...	...	...	...
111	Hauck	Rubbens	1963-11-16 03:31:38.929000

	FirstName	LastName	DateOfBirth
112	Barrows	Coupland	1979-10-31 18:28:35.483000
113	Knox	Group	1975-08-25 22:27:50.177000
114	Kuvalis	Coupland	1986-05-23 19:23:27.752000
115	Daniel	Shakesby	1980-11-28 17:57:03.702000

116 rows × 3 columns

In [185]:

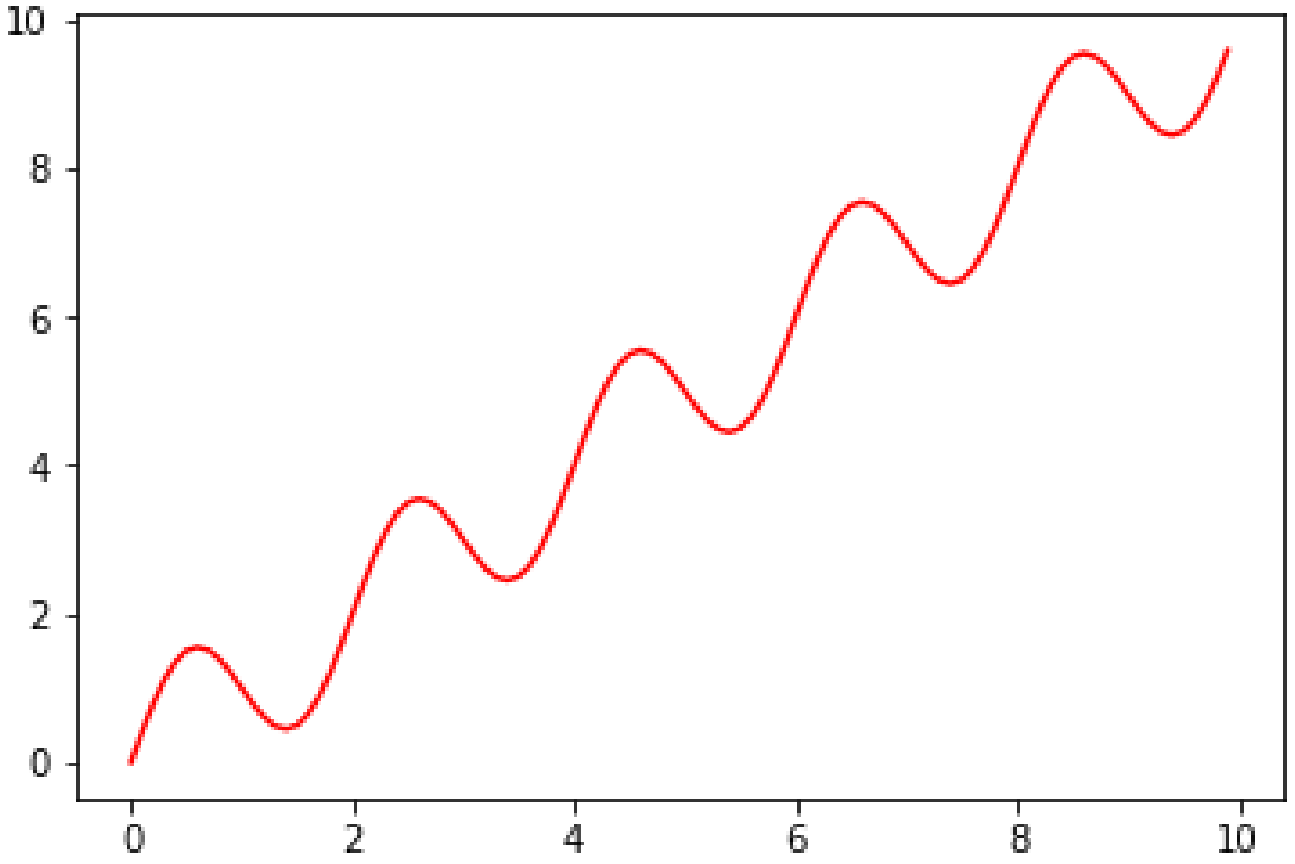
```
##29. Calculate average LOS.
Discharges["ExpectedLOS"].mean()
```

Out[185]:

```
8.82458915915616
```

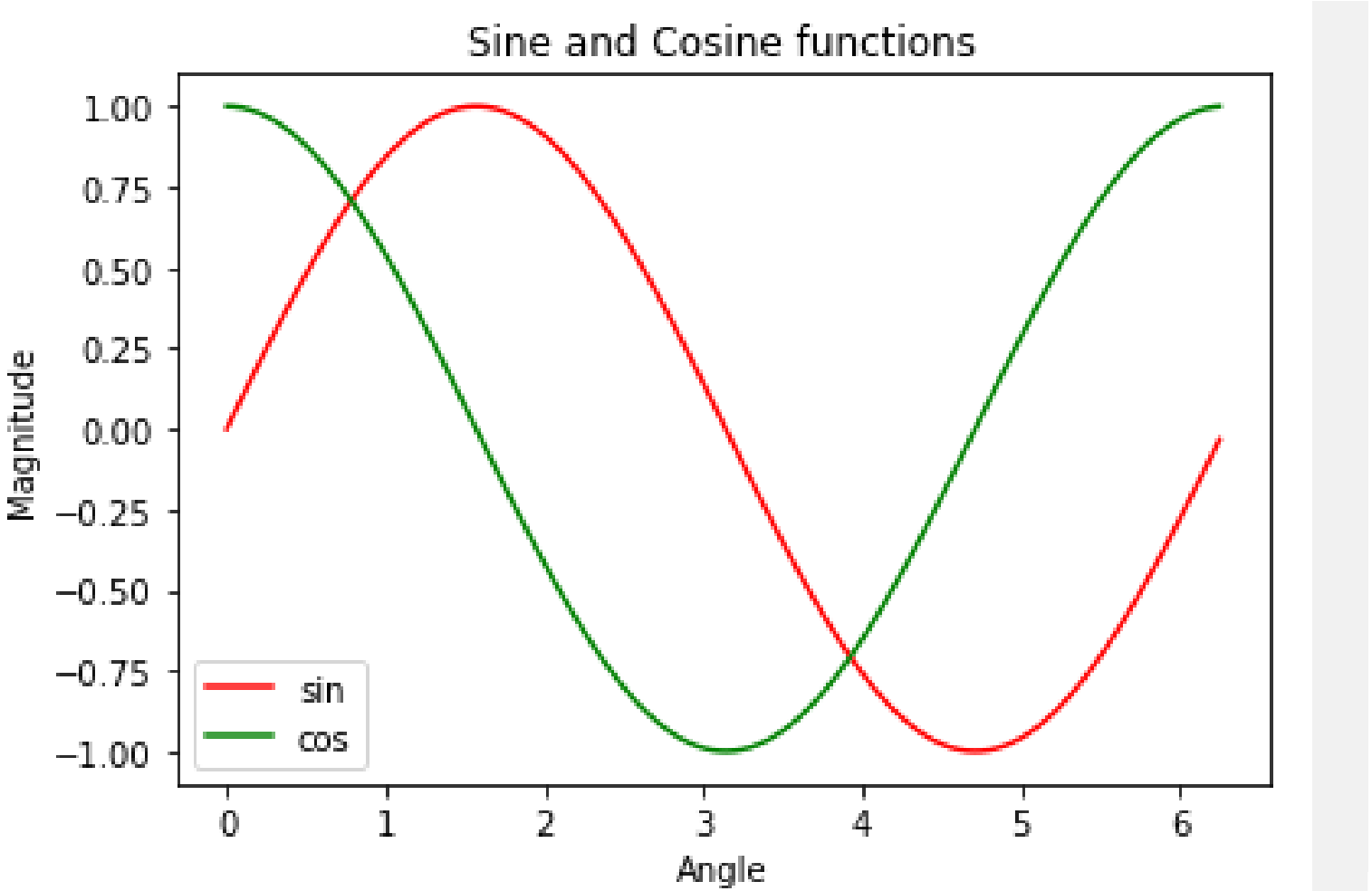
In [250]:

```
##13. Create a subplot on x = np.arange(0, 10, 0.1) , y = np.sin(np.p
X = x = np.arange(0, 10, 0.1)
y = np.sin(np.pi * x) + x
plt.plot(X, y, color='r', label='sin')
plt.show()
```


10
8
6
4
2
0
0
2
4
6
8
10

In [252]:

```
###61. Plot a graph by multiplotting on the same canvas  (Take any se
X = np.arange(0, math.pi*2, 0.05)
y = np.sin(X)
z = np.cos(X)
plt.plot(X, y, color='r', label='sin')
plt.plot(X, z, color='g', label='cos')
plt.xlabel("Angle")
plt.ylabel("Magnitude")
plt.title("Sine and Cosine functions")
plt.legend()
plt.show()
```


Sine and Cosine functions
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Magnitude
0
1
2
3
4
5
6
Angle
sin
cos

In [259]:

```
##4. Display data by splitting age in 4 quartiles and labeling the qu
Patients["Age"] = (pd.to_datetime("today").year-pd.to_datetime(Patier
df = pd.qcut(Patients['Age'], 4)
display (df)
```

```
0          (56.0, 62.0]
1        (34.999, 42.0]
2          (42.0, 49.0]
3          (49.0, 56.0]
4          (56.0, 62.0]
              ...
940      (34.999, 42.0]
941        (49.0, 56.0]
942      (34.999, 42.0]
943        (56.0, 62.0]
944        (49.0, 56.0]
Name: Age, Length: 945, dtype: category
```

```
Categories (4, interval[float64, right]): [(34.999,
42.0] < (42.0, 49.0] < (49.0, 56.0] < (56.0, 62.0]]
```

In [275]:

```
##80. Write a code snippet to print different ProviderSpecialty ( use

#display (Providers['ProviderSpecialty'])
df = Providers.groupby("ProviderSpecialty").count()
df.head()
```

Out[275]:

ProviderSpecialty	ProviderID	ProviderName	ProviderDateOnStaff
Cardiology	8	8	8
Pediatrics	9	9	9
PrimaryCare	16	16	16
Surgery	7	7	7

In [69]:

```
##78. Find reasonForVisit with highest count of acuity 5 patients.
mysql = lambda q: sqldf(q,globals())

mysql("Select ReasonForVisit from EDVisits where Acuity=5")
```

Out[69]:

	ReasonForVisit
0	Car Accident
1	Chest Pain
2	Chest Pain
3	Chest Pain
4	Shortness of Breath
...	...
204	Shortness of Breath

	ReasonForVisit
205	Shortness of Breath
206	Shortness of Breath
207	Shortness of Breath
208	Shortness of Breath

209 rows × 1 columns

In [303]:

```
##28 Which reason of visit has maximum mortality rate.

mysql = lambda q: sqldf(q,globals())
mysql("Select a.ReasonForVisit, b.ExpectedMortality from EDUnique a j
```

Out[303]:

	ReasonForVisit	ExpectedMortality
0	Accident	0.325386
1	Bleeding	0.027476
2	Car Accident	0.525589
3	Chest Pain	0.646007
4	Fever	0.622168
5	Gun Shot	0.526509
6	Intoxication	0.967396

	ReasonForVisit	ExpectedMortality
7	Laceration	0.426426
8	Migraine	0.671473
9	Pneumonia	0.086662
10	Shortness of Breath	0.467549
11	Stomach Ache	0.010046

In [13]:

```
##80Write a code snippet to print different ProviderSpecialty ( use g

Providers[['first_name','last_name']] = Providers['ProviderName'].loc
Providers
##Providers.loc[Providers['ProviderName'].str.split().str.len() == 2,
##Providers
```

Out[13]:

	ProviderID	ProviderName	ProviderSpecialty	ProviderDateOn
0	1	Sally Sue	Pediatrics	1993-0 00:00:00
1	2	Mike Myers	Pediatrics	1993-0 00:00:17
2	3	Jordan Michael	Pediatrics	1993-0 21:31:46

	ProviderID	ProviderName	ProviderSpecialty	ProviderDateOn
3	4	Ted Texas	Pediatrics	1993-1 21:33:52
4	5	Ala Bama	Pediatrics	1995-0 02:53:47
5	6	Harry Kane	Pediatrics	1995-0 03:49:03
6	7	Barry Bar	Pediatrics	1995-1 19:30:38
7	8	Ted Green	Pediatrics	1996-0 06:14:52
8	9	Ted Black	Pediatrics	1997-0 08:01:10
9	10	Fred Man	Surgery	1998-0 01:47:18
10	11	Kim Kimberly	Surgery	1998-0 14:47:29
11	12	Sarah Ab	Surgery	1998-0 05:22:40

	ProviderID	ProviderName	ProviderSpecialty	ProviderDateOn
12	13	Abigail Marriot	Surgery	1998-1 12:25:45
13	14	Dave Yu	Surgery	1999-0 16:18:57
14	15	Christian Saint	Surgery	2000-0 00:12:40
15	16	Perry Pardon	Surgery	2001-0 11:49:17
16	17	Kent Kendall	Cardiology	2001-1 11:18:32
17	18	Ryan Kevin	Cardiology	2003-0 21:16:34
18	19	Tyler Conner	Cardiology	2003-0 02:41:40
19	20	Bailey Barret	Cardiology	2003-0 16:05:50
20	21	Megan Bonco	Cardiology	2004-0 21:22:30

	ProviderID	ProviderName	ProviderSpecialty	ProviderDateOn
21	22	Joesph Walter	Cardiology	2005-0 11:40:44
22	23	Walter King	Cardiology	2006-0 22:11:58
23	24	Luke Long	Cardiology	2006-0 00:16:36
24	25	Justin Time	PrimaryCare	2006-0 04:46:36
25	26	Mike Joseph	PrimaryCare	2006-1 16:26:41
26	27	Bridget Brenda	PrimaryCare	2007-0 05:53:46
27	28	Brenda Bing	PrimaryCare	2007-1 22:33:55
28	29	Chandler Bing	PrimaryCare	2007-1 23:50:10
29	30	Joesph Ross	PrimaryCare	2008-1 18:06:32

	ProviderID	ProviderName	ProviderSpecialty	ProviderDateOn
30	31	Dwight Scott	PrimaryCare	2009-0 19:15:44
31	32	Michael Halpert	PrimaryCare	2009-0 00:14:41
32	33	Pamela Ding	PrimaryCare	2009-1 03:59:49
33	34	Tony Creed	PrimaryCare	2010-0 14:11:26
34	35	Phyllis Stanley	PrimaryCare	2011-0 16:06:03
35	36	Holly Hue	PrimaryCare	2012-0 06:55:22
36	37	Trent Tye	PrimaryCare	2013-0 15:32:21
37	38	Kimberly Cone	PrimaryCare	2013-0 05:17:19
38	39	Harry West	PrimaryCare	2013-0 00:22:03

ProviderID	ProviderName	ProviderSpecialty	ProviderDateOn

In [85]:

```
###18 Add column 'Age' in Patient table.
df["Age"] = (pd.to_datetime("today").year-pd.to_datetime(df["DateOfB:
df.head(945)
```

Out[85]:

	PatientID	FirstName	LastName	DateOfBirth	Gender
0	1	Lanni	Sue	1960-01-01 00:00:00.000	Male
1	2	Far	Myers	1985-11-15 02:08:42.090	Male
2	3	Devlin	Michael	1976-04-15 02:52:09.762	Male
3	4	Carmine	Texas	1968-10-15 03:32:13.635	Male
4	5	Tann	Bama	1962-05-01 19:12:58.950	Male
...	...	...	...	...	...

	PatientID	FirstName	LastName	DateOfBirth	Gender	
940	941	Wat	Fideler	1986-05-26 00:01:19.761	Male	Black/A Am
941	942	Wandie	Baythrop	1970-06-10 21:41:03.814	Male	Black/A Am
942	943	Diahann	Smeeton	1983-01-08 21:49:27.884	Male	Black/A Am
943	944	Panchito	Sharple	1963-06-05 07:57:05.569	Male	Black/A Am
944	945	Walsh	Calvie	1972-08-06 03:40:03.454	Male	Black/A Am

945 rows × 8 columns

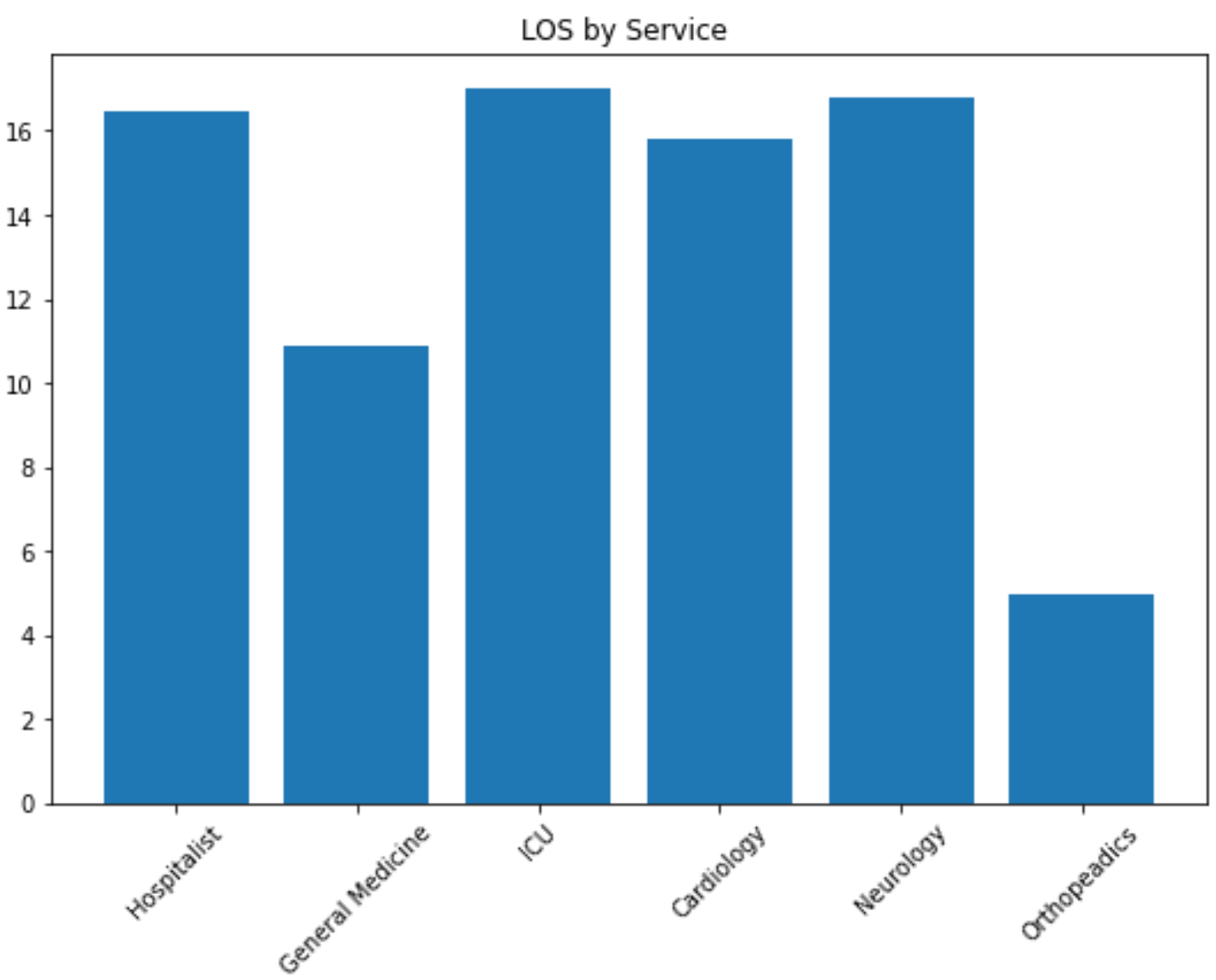
In [86]:

```
##55Create a bar chart on service & expected length of stay.

df = pd.read_excel("HospitalDatabase .xlsx","ReAdmissionRegistry")
plt.figure(figsize=(9,6))
plt.bar(x=df['Service'],
height=df['ExpectedLOS'])
plt.xticks(rotation=45)
plt.title('LOS by Service ')
```

Out[86]:

```
Text(0.5, 1.0, 'LOS by Service ')
```


LOS by Service
0
2
4
6
8
10
12
14
16
Hospitalist
General Medicine
ICU
Cardiology
Neurology
Orthopeadics

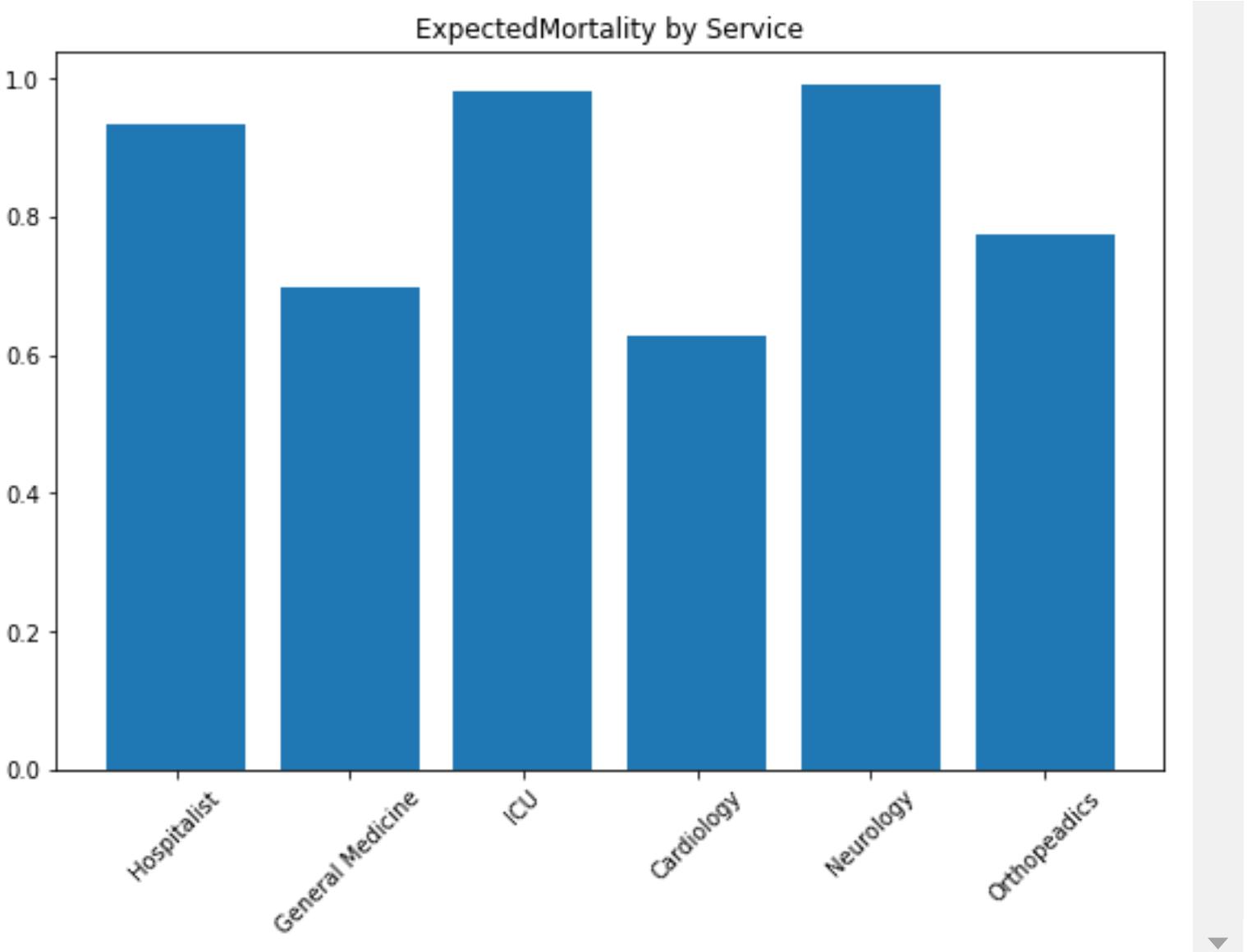
In [87]:

```
##33.Using a bar chart, which Service had the lowest count of Expecte


df = pd.read_excel("HospitalDatabase .xlsx","ReAdmissionRegistry")
rows_count = df1.count()
plt.figure(figsize=(9,6))
plt.bar(x=df['Service'],
height=df['ExpectedMortality'])
plt.xticks(rotation=45)
plt.title('ExpectedMortality by Service')
```

Out[87]:

```
Text(0.5, 1.0, 'ExpectedMortality by Service')
```


ExpectedMortality by Service
1.0
0.8
0.6
0.4
0.2
0.0
Hospitalist
General Medicine
ICU
Cardiology
Neurology
Orthopeadics

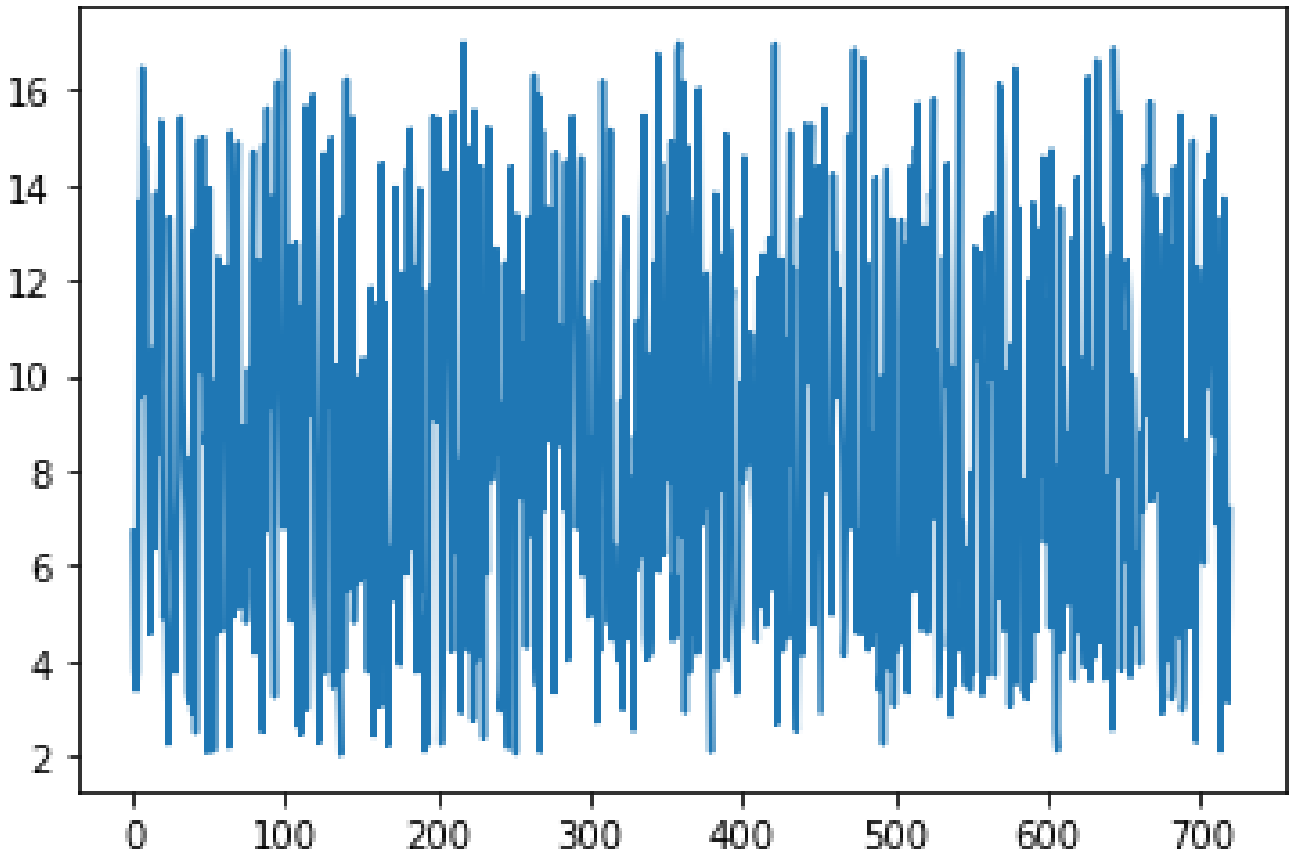
In [88]:

```
##16 Plot a graph to show the distribution of expected length of stay

df = pd.read_excel("HospitalDatabase .xlsx","ReAdmissionRegistry")
df = df["ExpectedLOS"]
plt.plot(df)
```

Out[88]:

```
[<matplotlib.lines.Line2D at 0x1526268ea00>]
```


16
14
12
10
8
6
4
2
0
100
200
300
400
500
600
700

In [89]:

```
###5.Display full name of patients who are born in 1986.

df = pd.read_excel("HospitalDatabase .xlsx","Patients")
df['FullName'] = df['FirstName'] + ' ' + df["LastName"]
df["DateOfBirthYear"] = pd.to_datetime(df["DateOfBirth"]).dt.year
df[df["DateOfBirthYear"] == 1986]
```

Out[89]:

	PatientID	FirstName	LastName	DateOfBirth	Gender
23	24	Gabriel	Joseph	1986-05-31 09:36:05.716	Male
24	25	Lincoln	Brenda	1986-07-24 17:36:00.791	Male
29	30	Ala	Halpert	1986-11-26 10:44:22.628	Female
72	74	Lolita	Darci	1986-01-08 02:34:04.596	Female

	PatientID	FirstName	LastName	DateOfBirth	Gender	
164	165	Fadel	Bernardt	1986-05-29 00:35:58.694	Male	Bl
238	239	Bentley	Kippax	1986-02-11 06:22:40.734	Male	Bl
367	368	Vale	Olanda	1986-04-15 00:49:58.690	Female	
373	374	Britt	Dureden	1986-11-19 23:50:47.955	Female	
386	387	Cristabel	Chatel	1986-07-10 16:27:30.640	Male	
415	416	Constantia	Group	1986-03-26 06:37:22.525	Female	
452	453	Morgan	Scrowston	1986-12-19 21:34:24.472	Female	
561	562	Dom	Baglow	1986-07-26 14:43:49.240	Male	Bl
568	569	Ignazio	Melling	1986-12-25 03:40:43.884	Male	Bl

	PatientID	FirstName	LastName	DateOfBirth	Gender	
595	596	Homenick	Rings	1986-02-24 15:08:30.404	Male	Bl
639	640	Hashim	Slark	1986-10-13 10:40:43.596	Female	Bl
643	644	Ellie	Ramsbotham	1986-05-21 16:35:52.711	Female	Bl
674	675	Llewellyn	Group	1986-11-09 17:15:11.196	Female	
714	715	Niles	Shaw	1986-12-15 06:31:44.358	Male	
733	734	Yvette	Inc	1986-11-23 02:32:12.587	Female	
777	778	Siouxie	Group	1986-04-28 03:25:01.182	Male	
852	853	Carly	Group	1986-11-21 02:44:49.632	Female	
868	869	Worth	Pickering	1986-12-22 08:21:02.691	Female	

	PatientID	FirstName	LastName	DateOfBirth	Gender	
877	878	Arni	Baldack	1986-04-10 13:18:15.354	Male	
909	910	Kuvalis	Coupland	1986-05-23 19:23:27.752	Male	Bl
922	923	Rebbecca	Rollingson	1986-09-29 11:58:56.083	Male	Bl

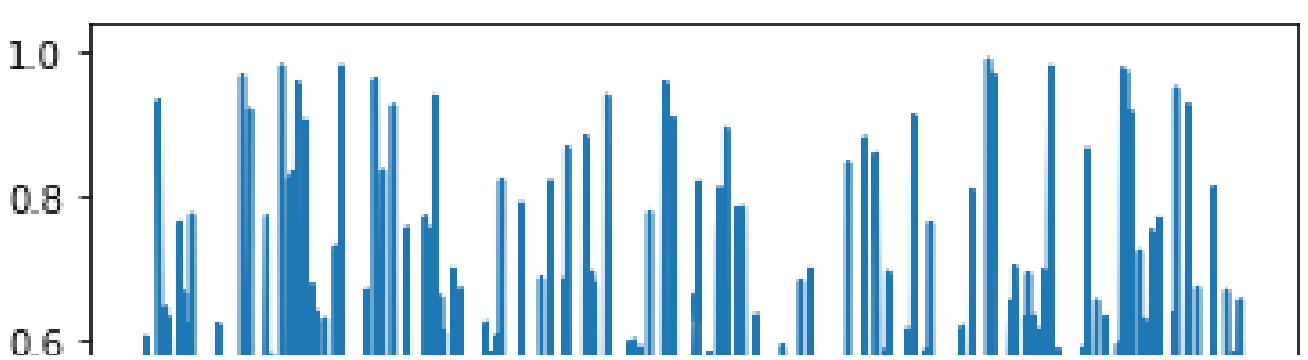
In [90]:

```
##70.Plot a graph to show the distribution of expected mortality.

df = pd.read_excel("HospitalDatabase .xlsx","ReAdmissionRegistry")
df = df["ExpectedMortality"]
plt.plot(df)
```

Out[90]:

```
[<matplotlib.lines.Line2D at 0x152636aecd0>]
```



In [91]:

```
##56.Count of canceled status.


df = pd.read_excel("HospitalDatabase .xlsx","AmbulatoryVisits")
df1 = df[ (df["VisitStatus"] == 'Canceled')]
rows_count = df1.count()[0]
print('Number of Rows count is:', rows_count )
```

```
Number of Rows count is: 60
```

In [ ]: