In [186]:
```
import pandas as pd
```

In [187]:
```
data = pd.read_csv("train_data1.csv", engine= "python")
```

In [188]:
```
data.head(10)
```

Out[188]:

	case_id	Hospital_code	Hospital_type_code	City_Code_Hospital	Hospital_region_code	Avai Re Hos
0	1	8	c	3	Z	
1	2	2	c	5	Z	
2	3	10	e	1	X	
3	4	26	b	2	Y	
4	5	26	b	2	Y	
5	6	23	a	6	X	
6	7	32	f	9	Y	
7	8	23	a	6	X	
8	9	1	d	10	Y	
9	10	10	e	1	X	

In [189]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 318438 entries, 0 to 318437
Data columns (total 18 columns):
 #   Column                             Non-Null Count   Dtype  
---  ------                             --------------   -----  
 0   case_id                            318438 non-null  int64  
 1   Hospital_code                      318438 non-null  int64  
 2   Hospital_type_code                 318438 non-null  object 
 3   City_Code_Hospital                 318438 non-null  int64  
 4   Hospital_region_code               318438 non-null  object 
 5   Available Extra Rooms in Hospital  318438 non-null  int64  
 6   Department                         318438 non-null  object 
 7   Ward_Type                          318438 non-null  object 
 8   Ward_Facility_Code                 318438 non-null  object 
 9   Bed Grade                          318325 non-null  float64
 10  patientid                          318438 non-null  int64  
 11  City_Code_Patient                  313906 non-null  float64
 12  Type of Admission                  318438 non-null  object 
 13  Severity of Illness                318438 non-null  object 
 14  Visitors with Patient              318438 non-null  int64  
 15  Age                                318438 non-null  object 
 16  Admission_Deposit                  318438 non-null  float64
 17  Stay                               318438 non-null  object 
dtypes: float64(3), int64(6), object(9)
memory usage: 43.7+ MB
```

In [190]:
```
pip install plotly_express==0.4.1
```

```
Requirement already satisfied: plotly_express==0.4.1 in c:\users\moiez\anac
onda3\lib\site-packages (0.4.1)
Requirement already satisfied: scipy>=0.18 in c:\users\moiez\anaconda3\lib
\site-packages (from plotly_express==0.4.1) (1.5.0)
Requirement already satisfied: patsy>=0.5 in c:\users\moiez\anaconda3\lib\s
ite-packages (from plotly_express==0.4.1) (0.5.1)
Requirement already satisfied: plotly>=4.1.0 in c:\users\moiez\anaconda3\li
b\site-packages (from plotly_express==0.4.1) (4.12.0)
Requirement already satisfied: statsmodels>=0.9.0 in c:\users\moiez\anacond
a3\lib\site-packages (from plotly_express==0.4.1) (0.11.1)
Requirement already satisfied: pandas>=0.20.0 in c:\users\moiez\anaconda3\l
ib\site-packages (from plotly_express==0.4.1) (1.0.5)
Requirement already satisfied: numpy>=1.11 in c:\users\moiez\anaconda3\lib
\site-packages (from plotly_express==0.4.1) (1.18.5)
Requirement already satisfied: six in c:\users\moiez\anaconda3\lib\site-pac
kages (from patsy>=0.5->plotly_express==0.4.1) (1.15.0)
Requirement already satisfied: retrying>=1.3.3 in c:\users\moiez\anaconda3
\lib\site-packages (from plotly>=4.1.0->plotly_express==0.4.1) (1.3.3)
Requirement already satisfied: pytz>=2017.2 in c:\users\moiez\anaconda3\lib
\site-packages (from pandas>=0.20.0->plotly_express==0.4.1) (2020.1)
Requirement already satisfied: python-dateutil>=2.6.1 in c:\users\moiez\ana
conda3\lib\site-packages (from pandas>=0.20.0->plotly_express==0.4.1) (2.8.
1)
Note: you may need to restart the kernel to use updated packages.
```

In [191]:
```
data.isnull().sum()
```

Out[191]:
```
case_id                                0
Hospital_code                          0
Hospital_type_code                     0
City_Code_Hospital                     0
Hospital_region_code                   0
Available Extra Rooms in Hospital      0
Department                             0
Ward_Type                              0
Ward_Facility_Code                     0
Bed Grade                            113
patientid                              0
City_Code_Patient                   4532
Type of Admission                      0
Severity of Illness                    0
Visitors with Patient                  0
Age                                    0
Admission_Deposit                      0
Stay                                   0
dtype: int64
```

In [192]:
```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
```

In [193]:
```

age_lst = data["Age"].unique()

age_lst.sort()
age_dict = dict(zip(age_lst, range(len(age_lst))))
data["new_age"]=data["Age"].replace(age_dict)
print(age_dict)

```

```
{'0-10': 0, '11-20': 1, '21-30': 2, '31-40': 3, '41-50': 4, '51-60': 5, '61
-70': 6, '71-80': 7, '81-90': 8, '91-100': 9}
```

In [194]:
```
data.head()
```

Out[194]:

	case_id	Hospital_code	Hospital_type_code	City_Code_Hospital	Hospital_region_code	Avai R Hos
0	1	8	c	3	Z	
1	2	2	c	5	Z	
2	3	10	e	1	X	
3	4	26	b	2	Y	
4	5	26	b	2	Y	

In [195]:
```
stay_list = data["Stay"].unique()
stay_list.sort()
dept_Stay = dict(zip(stay_list, range(len(stay_list))))
data["new_stay"]= data["Stay"].replace(dept_Stay)
print(dept_Stay)
```

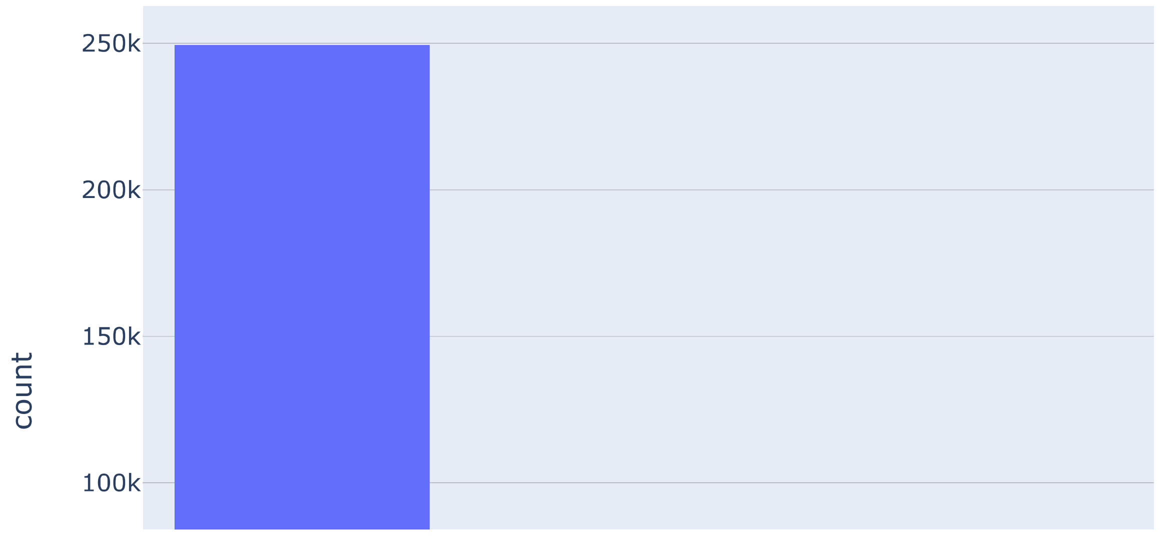
```
{'0-10': 0, '11-20': 1, '21-30': 2, '31-40': 3, '41-50': 4, '51-60': 5, '61
-70': 6, '71-80': 7, '81-90': 8, '91-100': 9, 'More than 100 Days': 10}
```

In [196]:
```
data.head()
```

Out[196]:

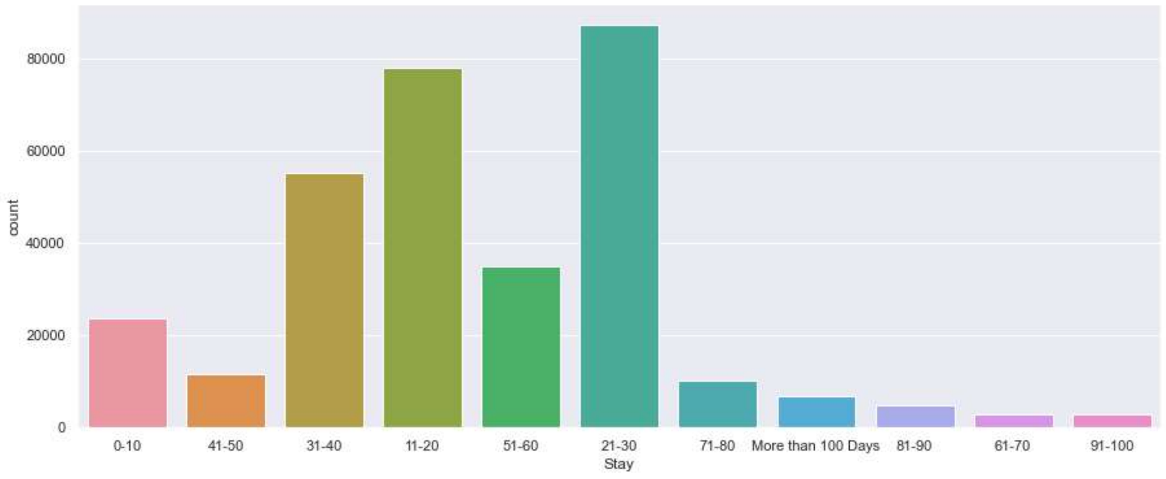
	case_id	Hospital_code	Hospital_type_code	City_Code_Hospital	Hospital_region_code	Avai Ro Hos
0	1	8	c	3	Z	
1	2	2	c	5	Z	
2	3	10	e	1	X	
3	4	26	b	2	Y	
4	5	26	b	2	Y	

```
In [197]:  import plotly.express as px
           fig = px.histogram(data, x="Department").update_xaxes(categoryorder="total de
           fig.show()
```



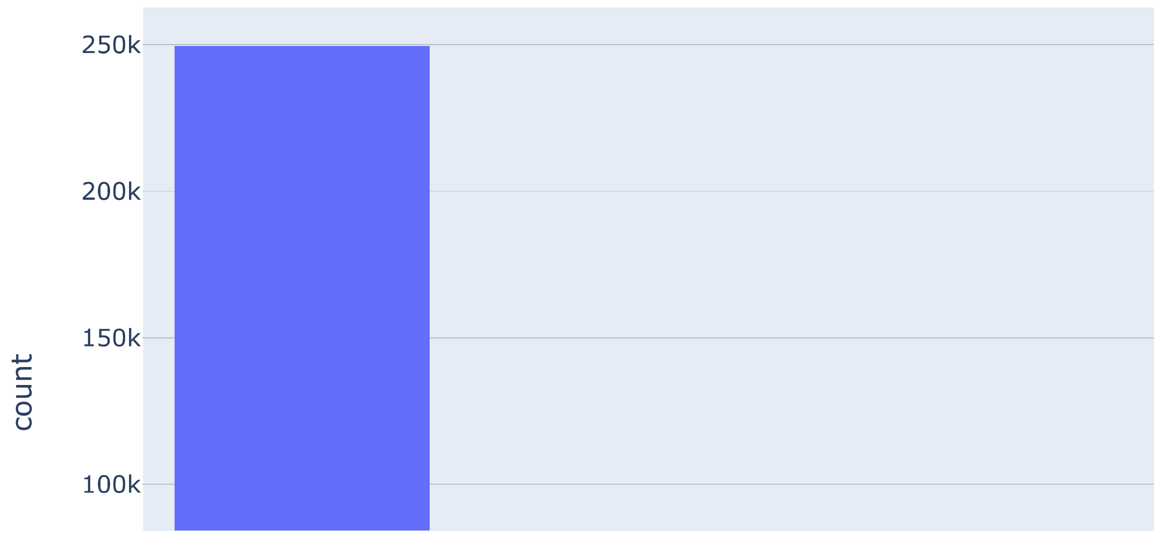
In [198]:
```
plt.figure(figsize=(15, 6))
sns.countplot(data.Stay)
```

Out[198]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a603f83340>
```

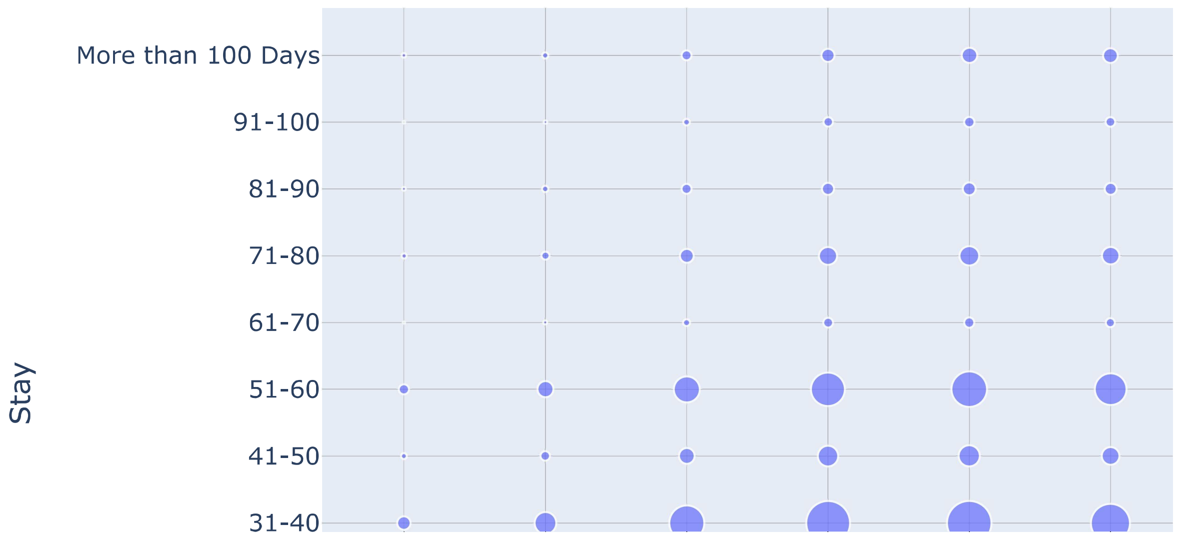


In [199]:

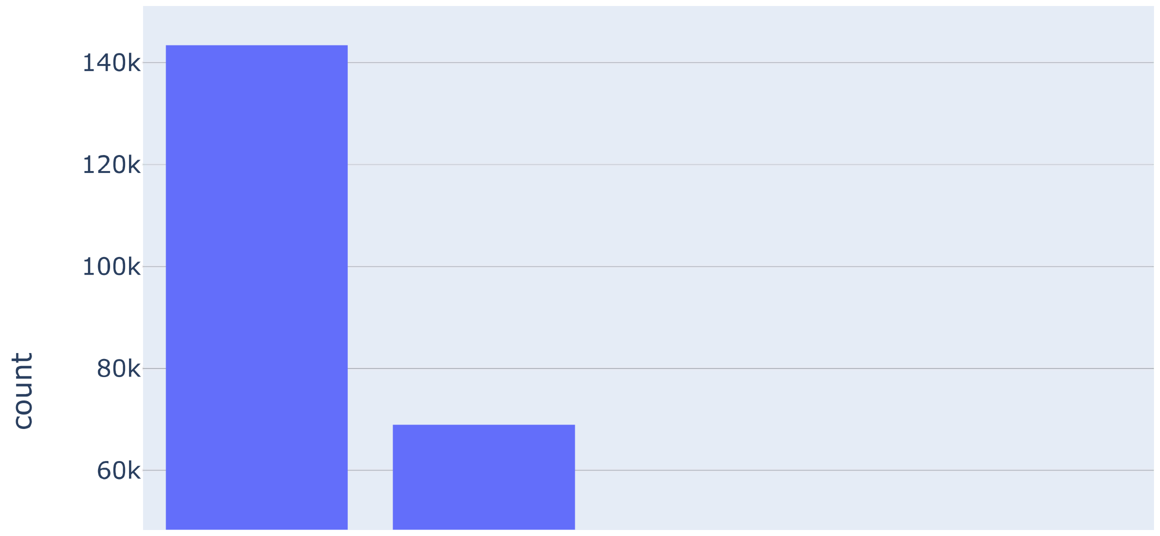
```
import plotly.express as px
fig = px.histogram(data, x="Department",).update_xaxes(categoryorder="total c

fig.update_layout(uniformtext_minsize=8, uniformtext_mode='hide')
fig.show()
```



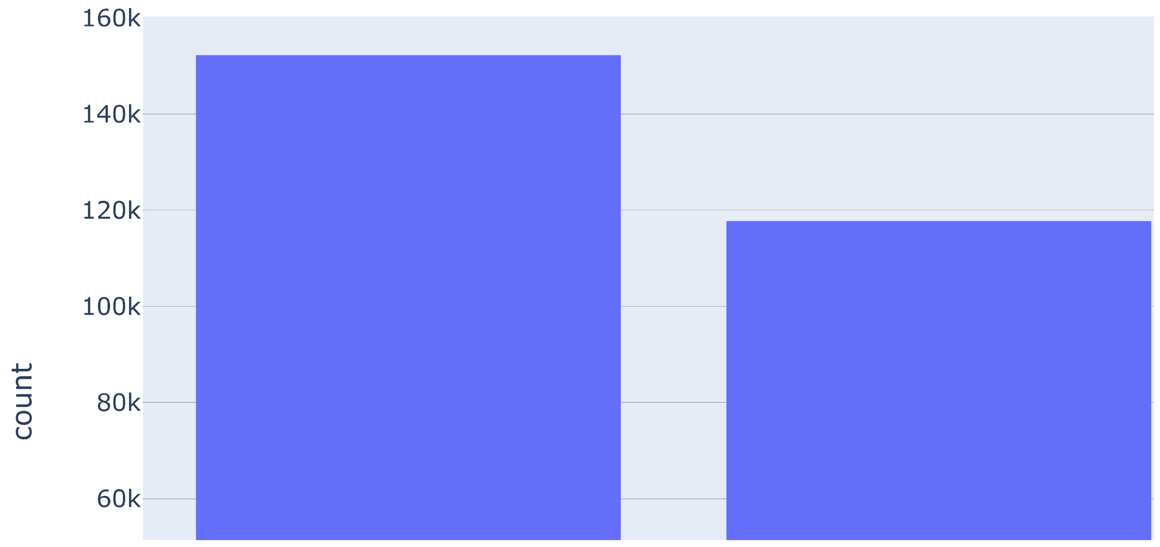
In [200]:
```
scat1 = data.groupby(['Age','Stay']).count().reset_index()
scat1['count']=y_val['Hospital_code']

import plotly.express as px
df = px.data.gapminder()

fig = px.scatter(scat1, x="Age", y="Stay",
      size="count")
fig.show()
```



In [201]:
```
fig2 = px.histogram(data, x="Hospital_type_code").update_xaxes(categoryorder=
fig2.show()
```

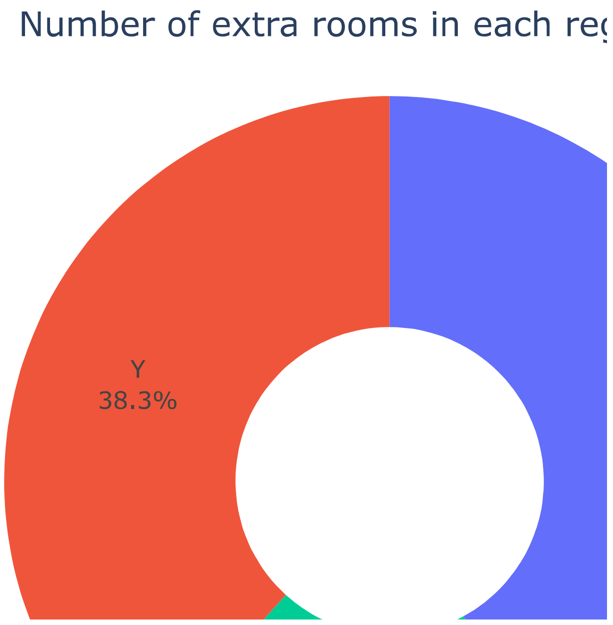


In [202]:
```
fig3 = px.histogram(data, x="Type of Admission").update_xaxes(categoryorder="
fig3.show()
```



In [206]:

```
beds = data.groupby('Hospital_region_code')['Available Extra Rooms in Hospita
fig4=px.pie(beds,values='Available Extra Rooms in Hospital',names='Hospital_r
fig4.update_layout(title='Number of extra rooms in each region code',title_x=
fig4.update_traces(textinfo='percent+label')
```



In [207]:

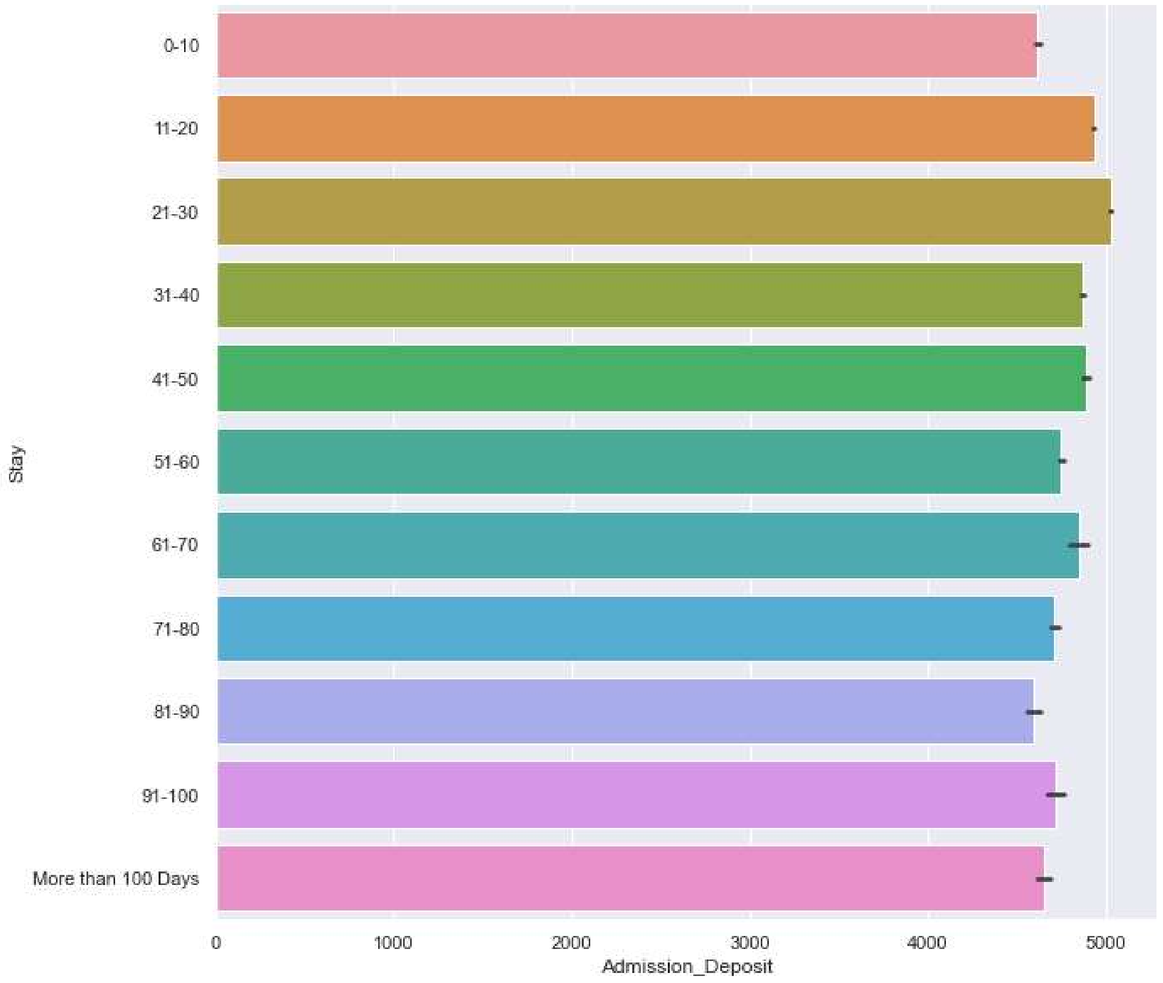
```
data['Stay'].value_counts()
```

Out[207]:

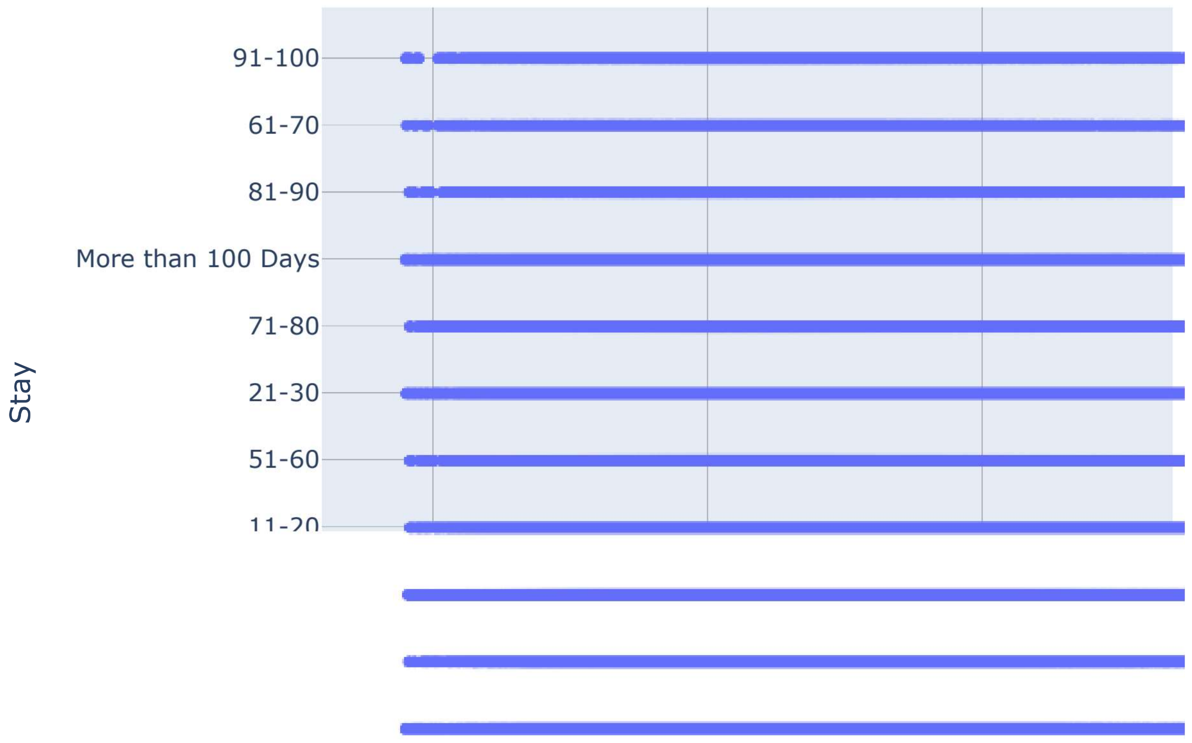
```
21-30                 87491
11-20                 78139
31-40                 55159
51-60                 35018
0-10                  23604
41-50                 11743
71-80                 10254
More than 100 Days     6683
81-90                  4838
91-100                 2765
61-70                  2744
Name: Stay, dtype: int64
```

In [208]:

```
import seaborn as sns
sns.set(style="darkgrid")

g=sns.barplot(y="Stay", x="Admission_Deposit", data=data,order=['0-10','11-20
```

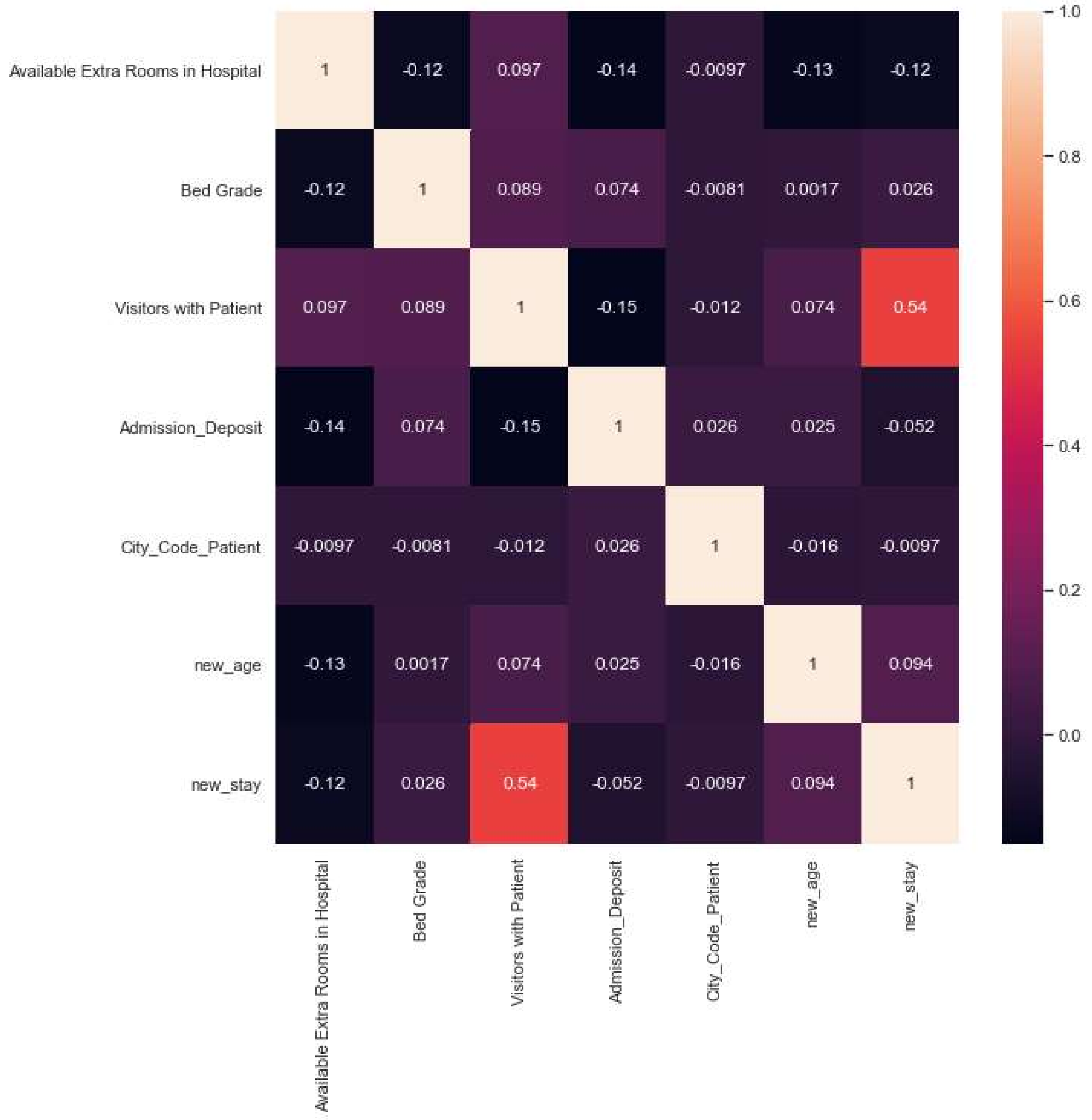


In [209]:
```
import plotly.express as px
df = px.data.gapminder()

fig4 = px.scatter(data, x="Admission_Deposit", y="Stay")
fig4.show()
```



In [210]:

```
heatmapdata = data[['Available Extra Rooms in Hospital', 'Bed Grade', 'Visito
               , 'Admission_Deposit','City_Code_Patient','new_age','new_stay']]

correlation_graph = heatmapdata.corr()

sns.set(rc={'figure.figsize':(10,10)})
sns.heatmap(data = correlation_graph,annot=True)
plt.show()
```



	Available Extra Rooms in Hospital	Bed Grade	Visitors with Patient	Admission_Deposit	City_Code_Patient	new_age	new_stay
Available Extra Rooms in Hospital	1	-0.12	0.097	-0.14	-0.0097	-0.13	-0.12
Bed Grade	-0.12	1	0.089	0.074	-0.0081	0.0017	0.026
Visitors with Patient	0.097	0.089	1	-0.15	-0.012	0.074	0.54
Admission_Deposit	-0.14	0.074	-0.15	1	0.026	0.025	-0.052
City_Code_Patient	-0.0097	-0.0081	-0.012	0.026	1	-0.016	-0.0097
new_age	-0.13	0.0017	0.074	0.025	-0.016	1	0.094
new_stay	-0.12	0.026	0.54	-0.052	-0.0097	0.094	1

In [211]:

```
cormat = heatmapdata.corr()
cormat
```

Out[211]:

	Available Extra Rooms in Hospital	Bed Grade	Visitors with Patient	Admission_Deposit	City_Code_Patient	ne
Available Extra Rooms in Hospital	1.000000	-0.115868	0.096714	-0.143739	-0.009681	-0.
Bed Grade	-0.115868	1.000000	0.088945	0.073833	-0.008105	0.
Visitors with Patient	0.096714	0.088945	1.000000	-0.150358	-0.012074	0.
Admission_Deposit	-0.143739	0.073833	-0.150358	1.000000	0.025837	0.
City_Code_Patient	-0.009681	-0.008105	-0.012074	0.025837	1.000000	-0.
new_age	-0.133491	0.001732	0.073795	0.025182	-0.016406	1.
new_stay	-0.121120	0.025741	0.537537	-0.052077	-0.009704	0.

In [212]:

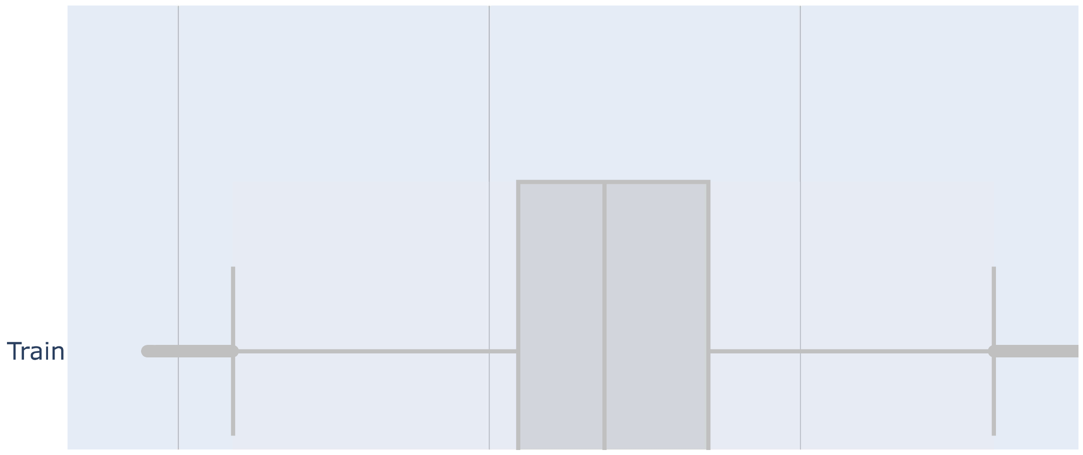
```
import numpy as np
import scipy.stats
x = data['new_age']
y = data['new_stay']
m = scipy.stats.pearsonr(x, y)    # Pearson's r
print( 'pearsons value is {} '.format(m) )
n = scipy.stats.spearmanr(x, y)   # Spearman's rho
print(n)
k = scipy.stats.kendalltau(x, y)
k
```

```
pearsons value is (0.09416326795751304, 0.0)
SpearmanrResult(correlation=0.09031087349478524, pvalue=0.0)
```

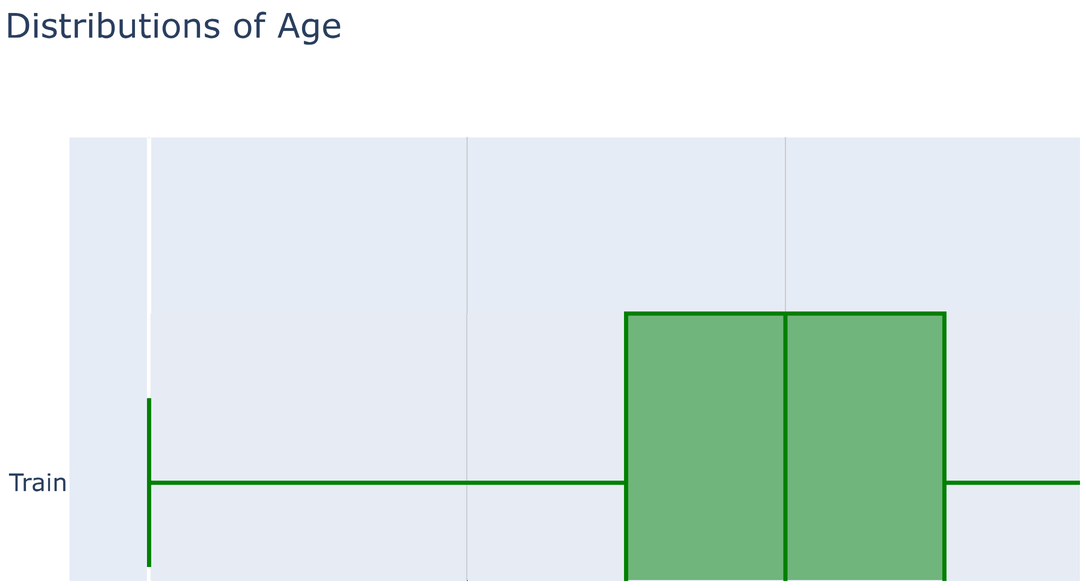
Out[212]:

```
KendalltauResult(correlation=0.07007718319174233, pvalue=0.0)
```

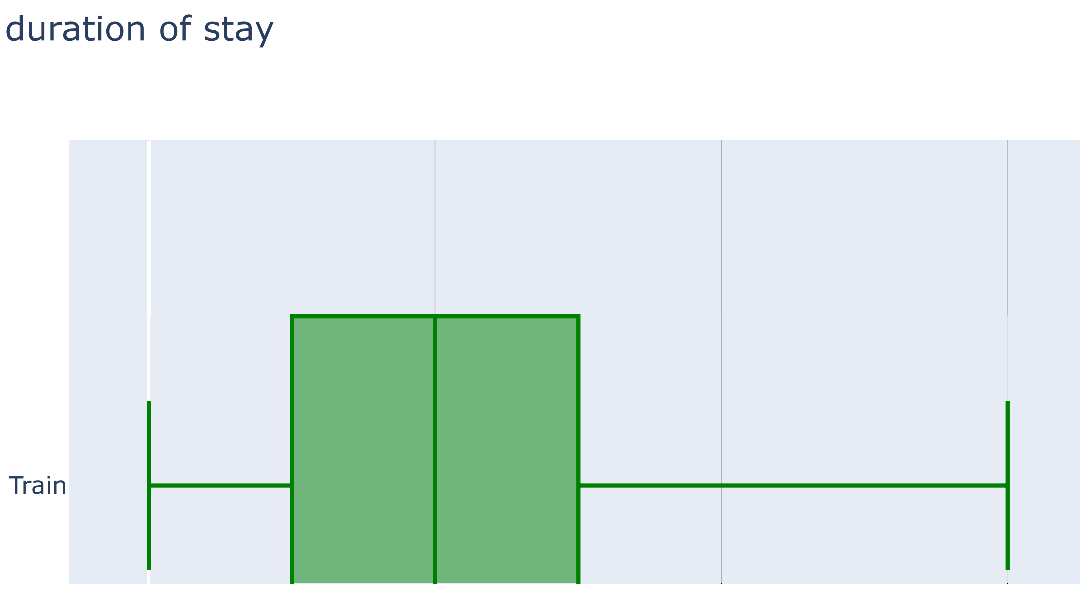
In [213]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go

from sklearn.preprocessing import LabelEncoder
fig = go.Figure()
fig.add_trace(go.Box(x=data['Admission_Deposit'],
                     marker_color="silver",
                     name="Train"))
```



```
In [214]:  fig = go.Figure()
           fig.add_trace(go.Box(x=data['new_age'],
                                marker_color="green",
                                name="Train"))
           fig.update_layout(title="Distributions of Age")
           fig.show()
```



In [215]:
```
fig = go.Figure()
fig.add_trace(go.Box(x=data['new_stay'],
                     marker_color="green",
                     name="Train"))
fig.update_layout(title="duration of stay")
fig.show()
```



In [216]:
```
p = data.groupby('Department')['Available Extra Rooms in Hospital'].agg('cour
q = data.groupby('Type of Admission')['Available Extra Rooms in Hospital'].ag
p,q
```

Out[216]:
```
(Department
 TB & Chest disease      9586
 anesthesia             29649
 gynecology            249486
 radiotherapy           28516
 surgery                 1201
 Name: Available Extra Rooms in Hospital, dtype: int64,
 Type of Admission
 Emergency    117676
 Trauma       152261
 Urgent        48501
 Name: Available Extra Rooms in Hospital, dtype: int64)
```

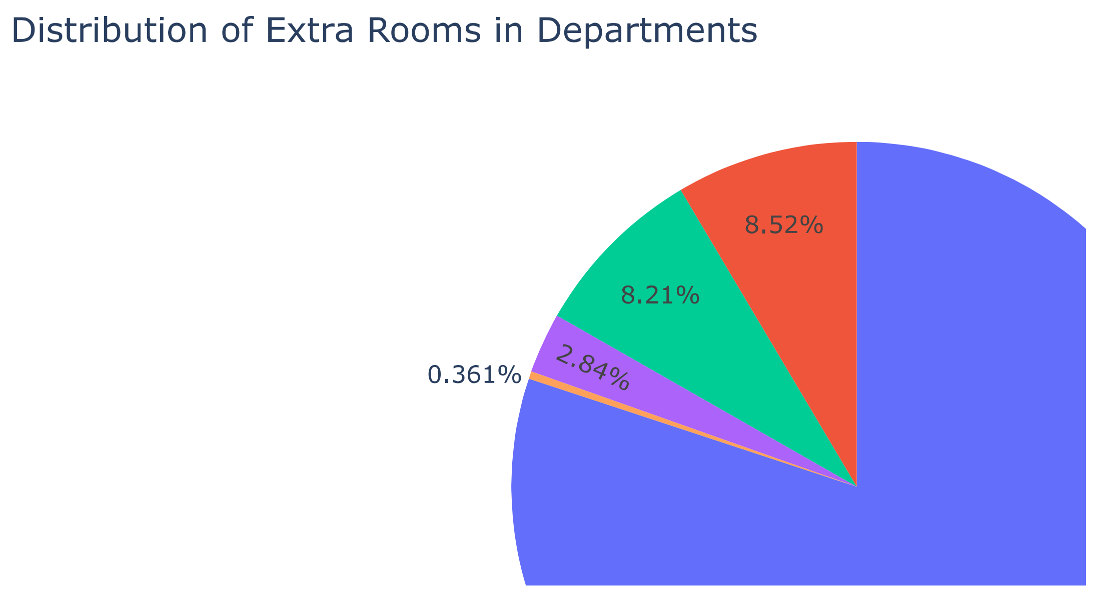
In [217]:
```
data.groupby('Severity of Illness')['Available Extra Rooms in Hospital'].agg(
```

Out[217]:
```
Severity of Illness
Extreme      56723
Minor        85872
Moderate    175843
Name: Available Extra Rooms in Hospital, dtype: int64
```

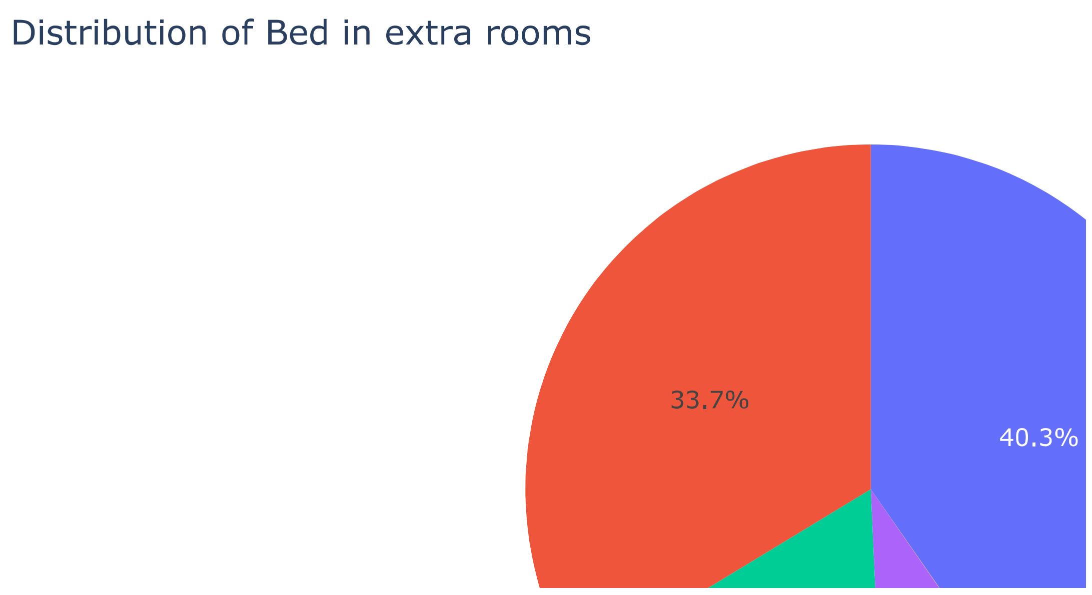
In [218]:
```
data.groupby('Severity of Illness')['Bed Grade'].agg('mean')
```

Out[218]:
```
Severity of Illness
Extreme     2.254702
Minor       2.991415
Moderate    2.566917
Name: Bed Grade, dtype: float64
```

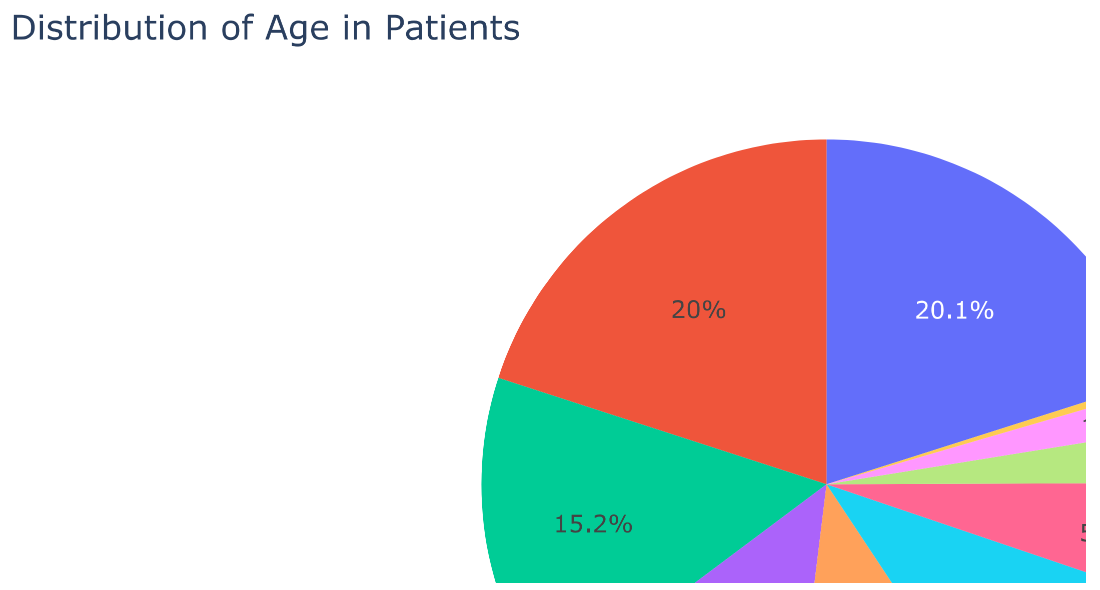
In [219]:
```
px.pie(data,values='Available Extra Rooms in Hospital',names='Department',tit
```



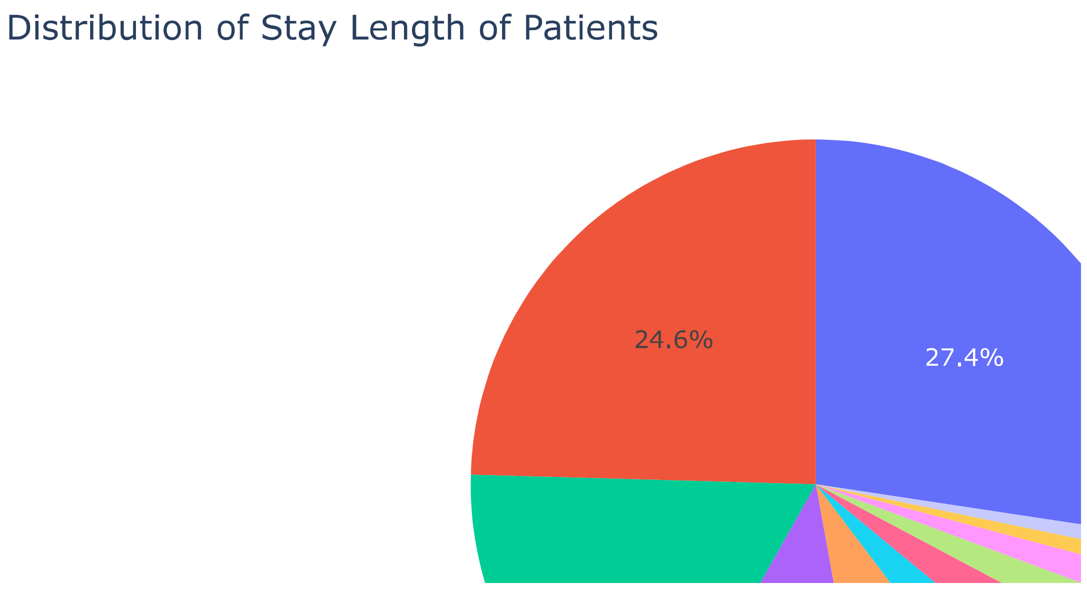
```
In [220]:  px.pie(data,values='Available Extra Rooms in Hospital',names='Bed Grade',titl
```



```
In [221]: px.pie(data,values='patientid',names='Age',title='Distribution of Age in Pati
```



In [222]:
```
px.pie(data,values='patientid',names='Stay',title='Distribution of Stay Lengt
```

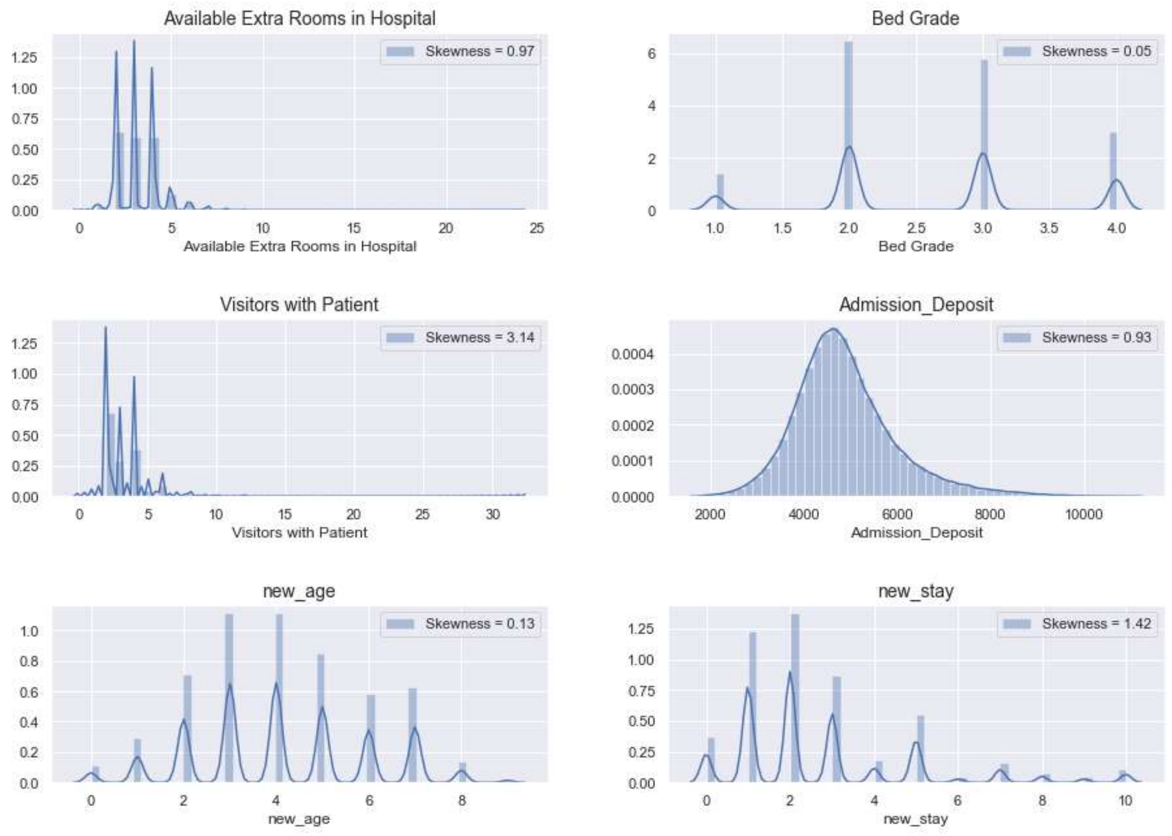


In [223]:
```
data.to_csv('new_mz.csv')
```

In [224]:
```
numerical_data = data[['Available Extra Rooms in Hospital', 'Bed Grade', 'Vis
                 , 'Admission_Deposit', 'new_age','new_stay']]
```

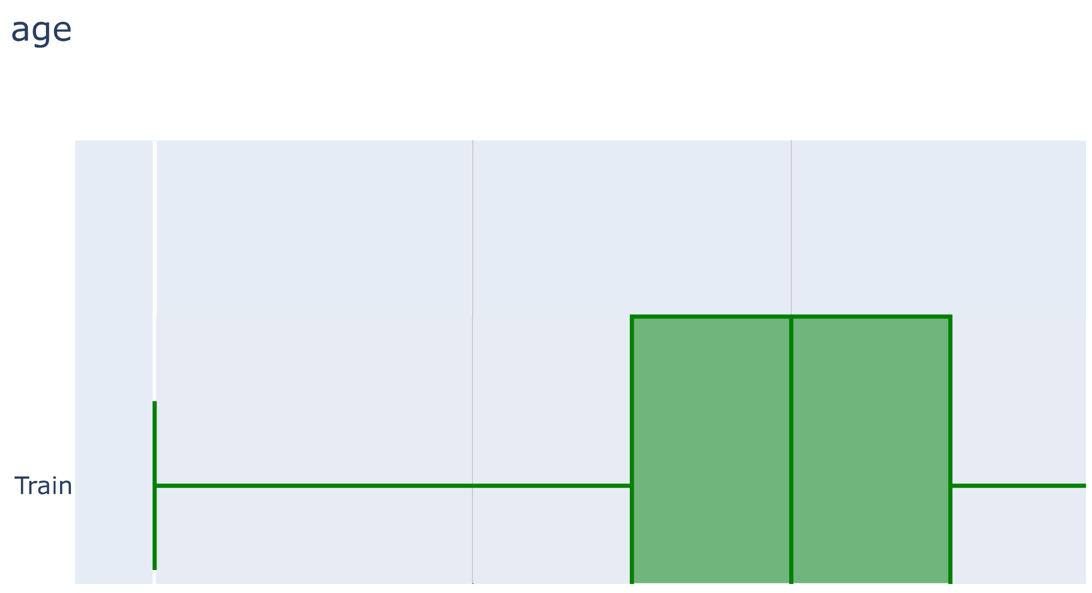
In [225]:

```
fig, new_plot =plt.subplots(3,2, figsize=(14,10))
fig.tight_layout(pad=5.0)

for new_plot, n in zip(new_plot.flatten(), numerical_data.columns.tolist()):
    sns.distplot(ax=new_plot, a=numerical_data[n], label="Skewness = %.2f"%(r
    new_plot.set_title(n, fontsize = 14)
    new_plot.legend(loc = 'best')
```



In [226]:

```
fig = go.Figure()
fig.add_trace(go.Box(x=numerical_data['new_age'],
                     marker_color="green",
                     name="Train"))
fig.update_layout(title="age")
fig.show()
```



In [227]:

```
from scipy import stats
import numpy as np
z = np.abs(stats.zscore(numerical_data))
print(z)
```

```
[[0.16917678        nan 0.72792324 0.02783522 0.46160017 1.25509783]
 [1.02521686        nan 0.72792324 0.98755589 0.46160017 0.60530517]
 [1.02521686        nan 0.72792324 0.12491035 0.46160017 0.14020442]
 ...
 [0.16917678        nan 0.16104865 0.59418892 1.52004528 0.78999708]
 [0.16917678        nan 0.97270052 1.03034194 1.65529005 0.78999708]
 [1.5429034         nan 0.72792324 0.11846927 1.65529005 1.25509783]]
```

In [228]:
```
threshold = 3
print (np.where(z > 3))
```

```
(array([    13,     15,     20, ..., 318384, 318409, 318432], dtype=int64),
array([3, 3, 3, ..., 0, 3, 3], dtype=int64))

<ipython-input-228-ec5b025f7bd6>:2: RuntimeWarning:

invalid value encountered in greater
```

In [ ]:

In [229]:
```
print('Train columns :\n',data.columns)
print('Train shape : ', data.shape)
print('\n')
```

```
Train columns :
 Index(['case_id', 'Hospital_code', 'Hospital_type_code', 'City_Code_Hospit
al',
       'Hospital_region_code', 'Available Extra Rooms in Hospital',
       'Department', 'Ward_Type', 'Ward_Facility_Code', 'Bed Grade',
       'patientid', 'City_Code_Patient', 'Type of Admission',
       'Severity of Illness', 'Visitors with Patient', 'Age',
       'Admission_Deposit', 'Stay', 'new_age', 'new_stay'],
      dtype='object')
Train shape :  (318438, 20)
```

In [230]:
```
new1 = data.drop(['case_id', 'Hospital_code','Age', 'City_Code_Hospital', 'Ci
            , 'Hospital_type_code', 'Hospital_region_code','Stay', 'Ward_Type
           , axis = 1)
```

In [231]:
```


    TOA_lst = new1["Type of Admission"].unique()
    TOA_lst.sort()
    TOA_dict = dict(zip(TOA_lst, range(len(TOA_lst))))
    new1["Type of Admission"].replace(TOA_dict, inplace=True)
    print(TOA_dict)
```

```
{'Emergency': 0, 'Trauma': 1, 'Urgent': 2}
```

In [232]:
```
new1
```

Out[232]:
```
        Available
            Extra                  Bed                Type of  Severity  Visitors
            Rooms   Department   Grade  patientid  Admission        of      with  Admission_De
               in                                              Illness   Patient
         Hospital
     0          3 radiotherapy     2.0      31397          0   Extreme         2            49
     1          2 radiotherapy     2.0      31397          1   Extreme         2            59
     2          2   anesthesia     2.0      31397          1   Extreme         2            47
     3          2 radiotherapy     2.0      31397          1   Extreme         2            72
     4          2 radiotherapy     2.0      31397          1   Extreme         2            55
   ...        ...          ...     ...        ...        ...       ...       ...
318433          3 radiotherapy     4.0      86499          0  Moderate         3            41
318434          2   anesthesia     4.0        325          2  Moderate         4            66
318435          3   gynecology     4.0     125235          0     Minor         3            42
318436          3   anesthesia     3.0      91081          1     Minor         5            37
318437          5   gynecology     2.0      21641          0     Minor         2            47

318438 rows × 10 columns
```

In [233]:
```
new2= pd.get_dummies(new1,columns=['Department','Severity of Illness'],drop_f
```

In [234]:
```
new3 = new2.dropna()
```

In [235]:
```
new3.head()
```

Out[235]:

	Available Extra Rooms in Hospital	Bed Grade	patientid	Type of Admission	Visitors with Patient	Admission_Deposit	new_age	new_stay	D
0	3	2.0	31397	0	2	4911.0	5	0	
1	2	2.0	31397	1	2	5954.0	5	4	
2	2	2.0	31397	1	2	4745.0	5	3	
3	2	2.0	31397	1	2	7272.0	5	4	
4	2	2.0	31397	1	2	5558.0	5	4	

In [ ]:

In [237]:
```

column_names = ['new_stay','new_age','Available Extra Rooms in Hospital','Bed
                 'patientid','Type of Admission','Visitors with Patient','Admi
                 'Department_anesthesia','Department_gynecology',
                 'Department_radiotherapy','Department_surgery','Severity of I
                ]

new3 = new3.reindex(columns=column_names)
```

In [238]:
```
new3.head()
```

Out[238]:

	new_stay	new_age	Available Extra Rooms in Hospital	Bed Grade	patientid	Type of Admission	Visitors with Patient	Admission_Deposit	D
0	0	5	3	2.0	31397	0	2	4911.0	
1	4	5	2	2.0	31397	1	2	5954.0	
2	3	5	2	2.0	31397	1	2	4745.0	
3	4	5	2	2.0	31397	1	2	7272.0	
4	4	5	2	2.0	31397	1	2	5558.0	

In [239]:
```
x_train = new3.iloc[:, 1:].values
y_train = new3.iloc[:, 0].values
```

In [240]:
```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score


x_train_split, x_val_split, y_train_split, y_val_split = train_test_split(x_t
clf = RandomForestClassifier(n_estimators=300, max_depth = 20, min_samples_le
clf.fit(x_train_split, y_train_split)
y_pred = clf.predict(x_val_split)
accuracy = accuracy_score(y_pred, y_val_split)
print('Accuracy :',accuracy)
```

```
Accuracy : 0.37147374814132234
```

In [241]:
```
# Fit the model into the whole data train
clf.fit(x_train, y_train)
```

Out[241]:
```
RandomForestClassifier(max_depth=20, max_features=0.5, min_samples_leaf=10,
                       n_estimators=300)
```