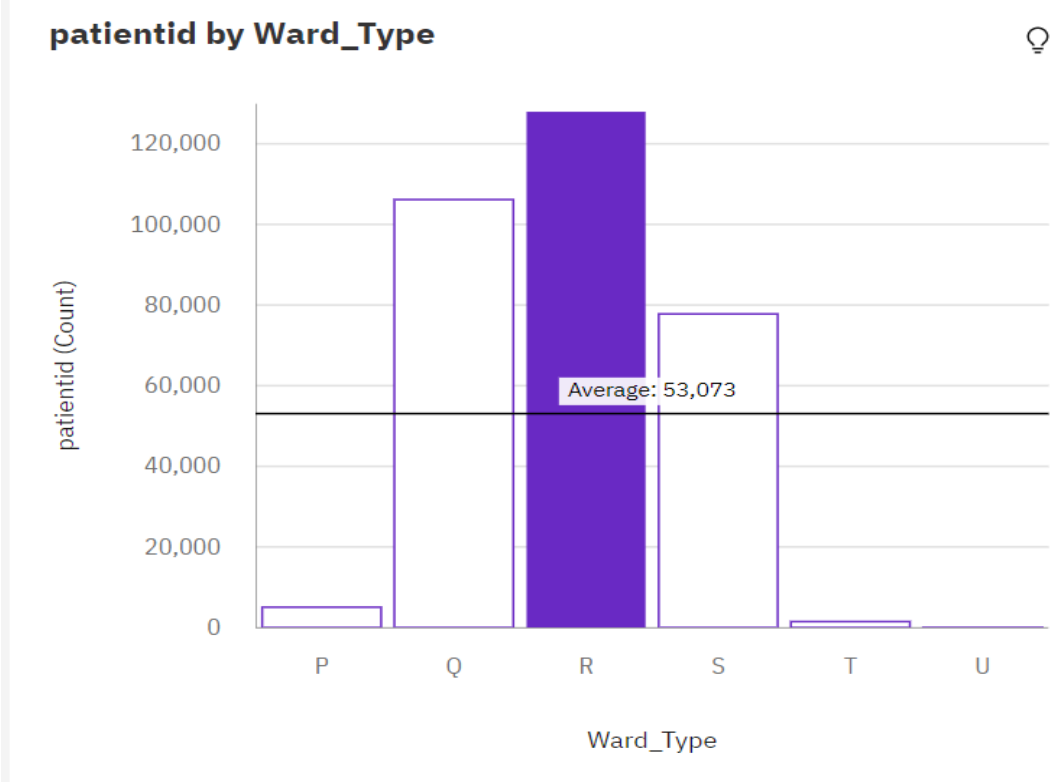
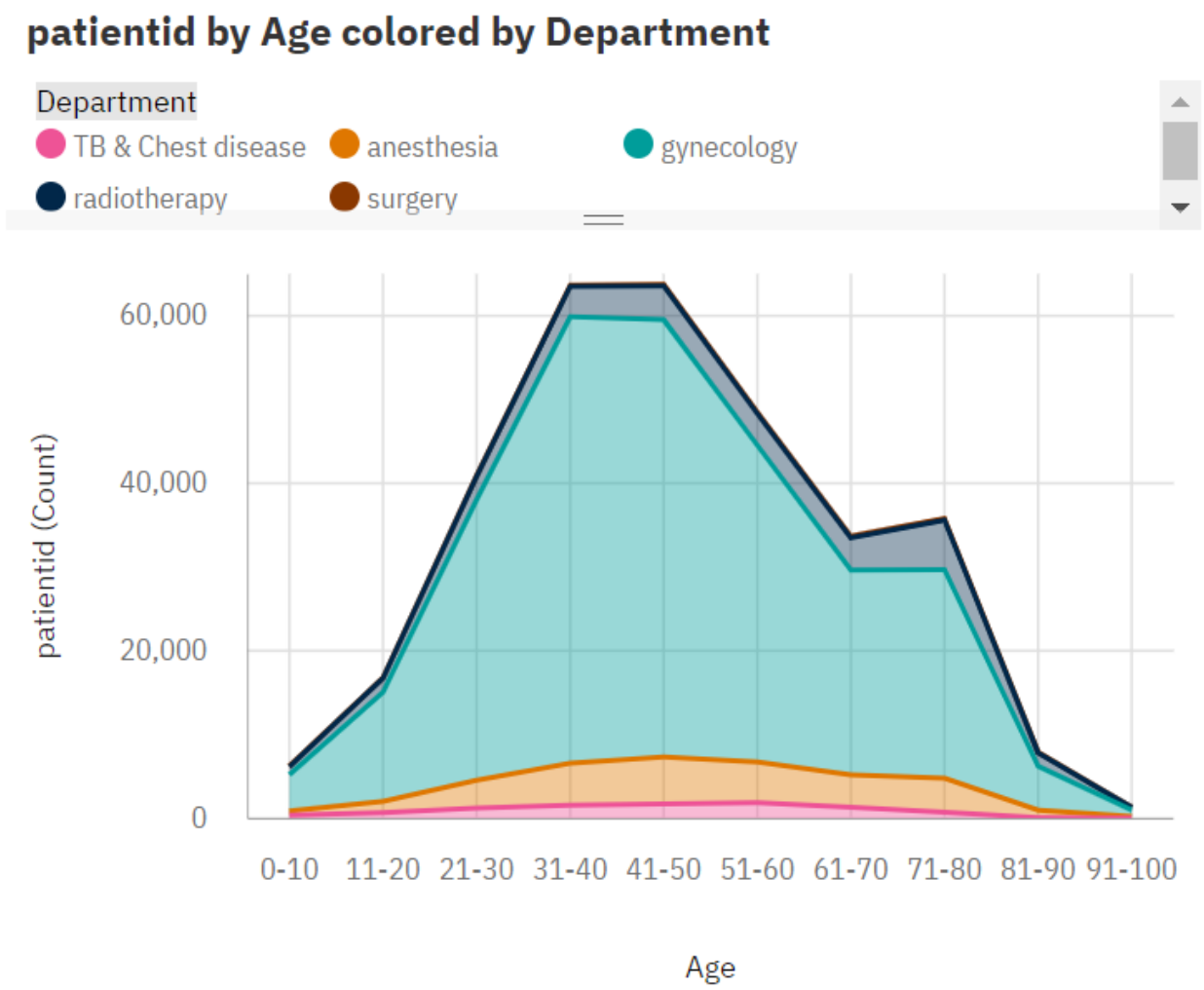
Sprint -1

Data Visualization

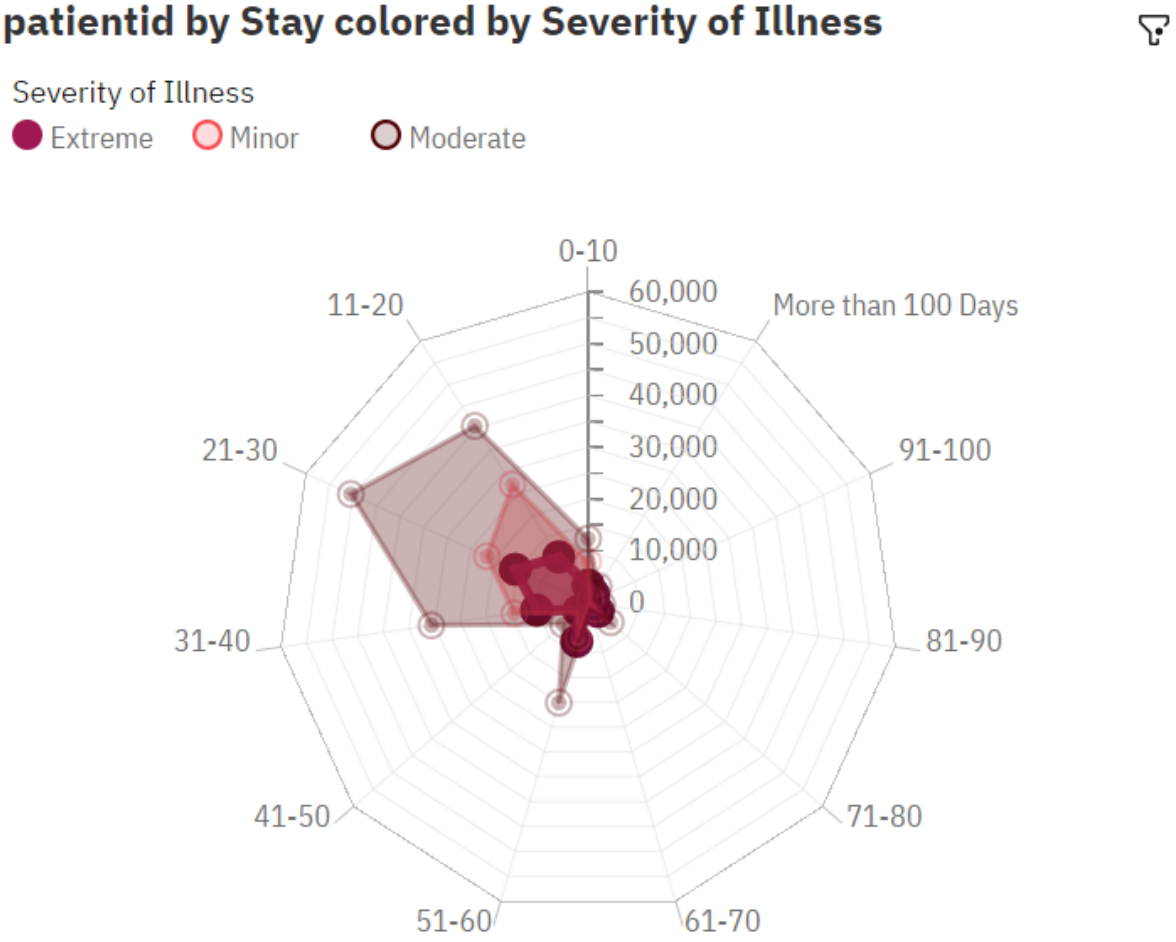
No of Patients in each ward



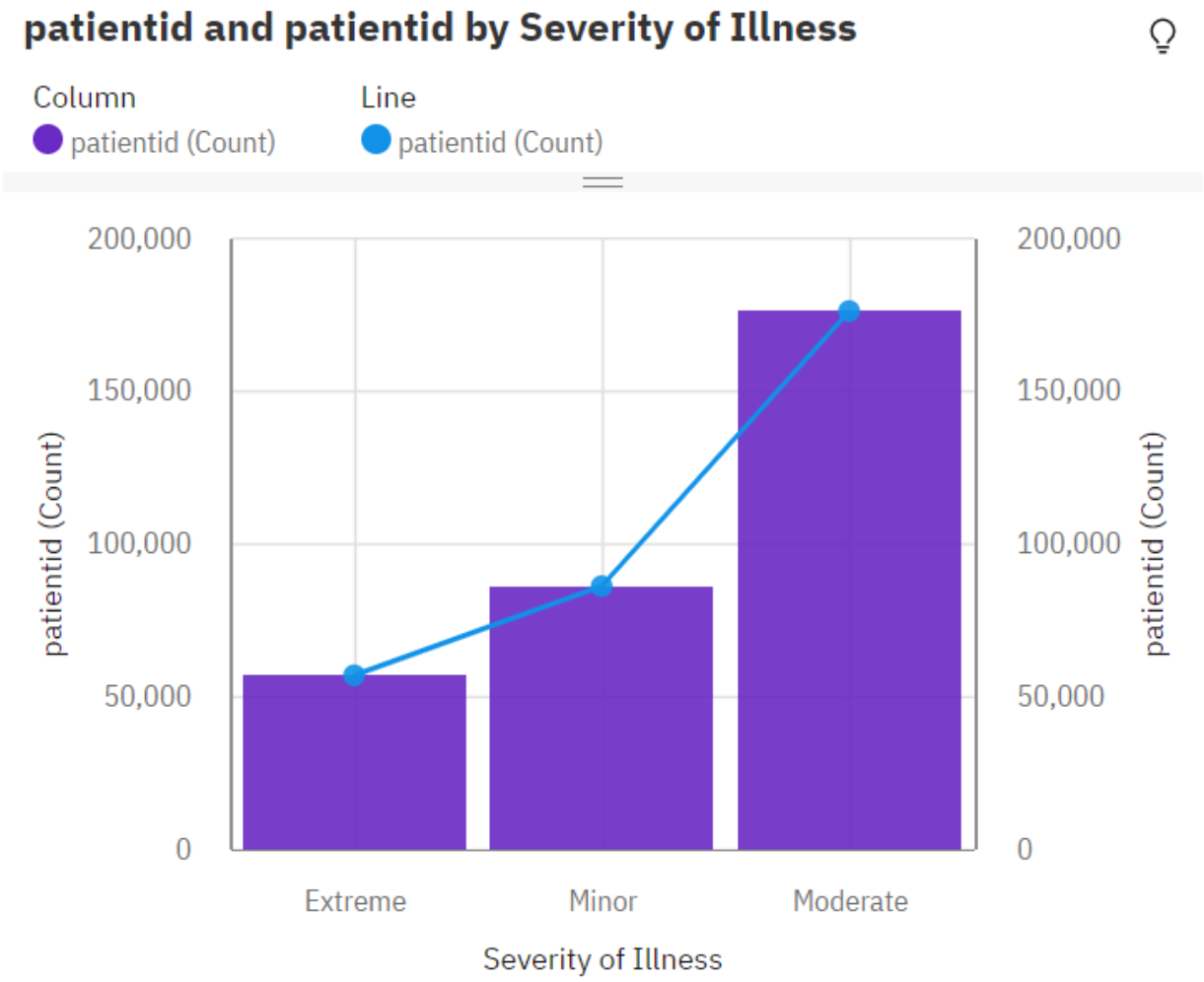
No of patients by age colored by department



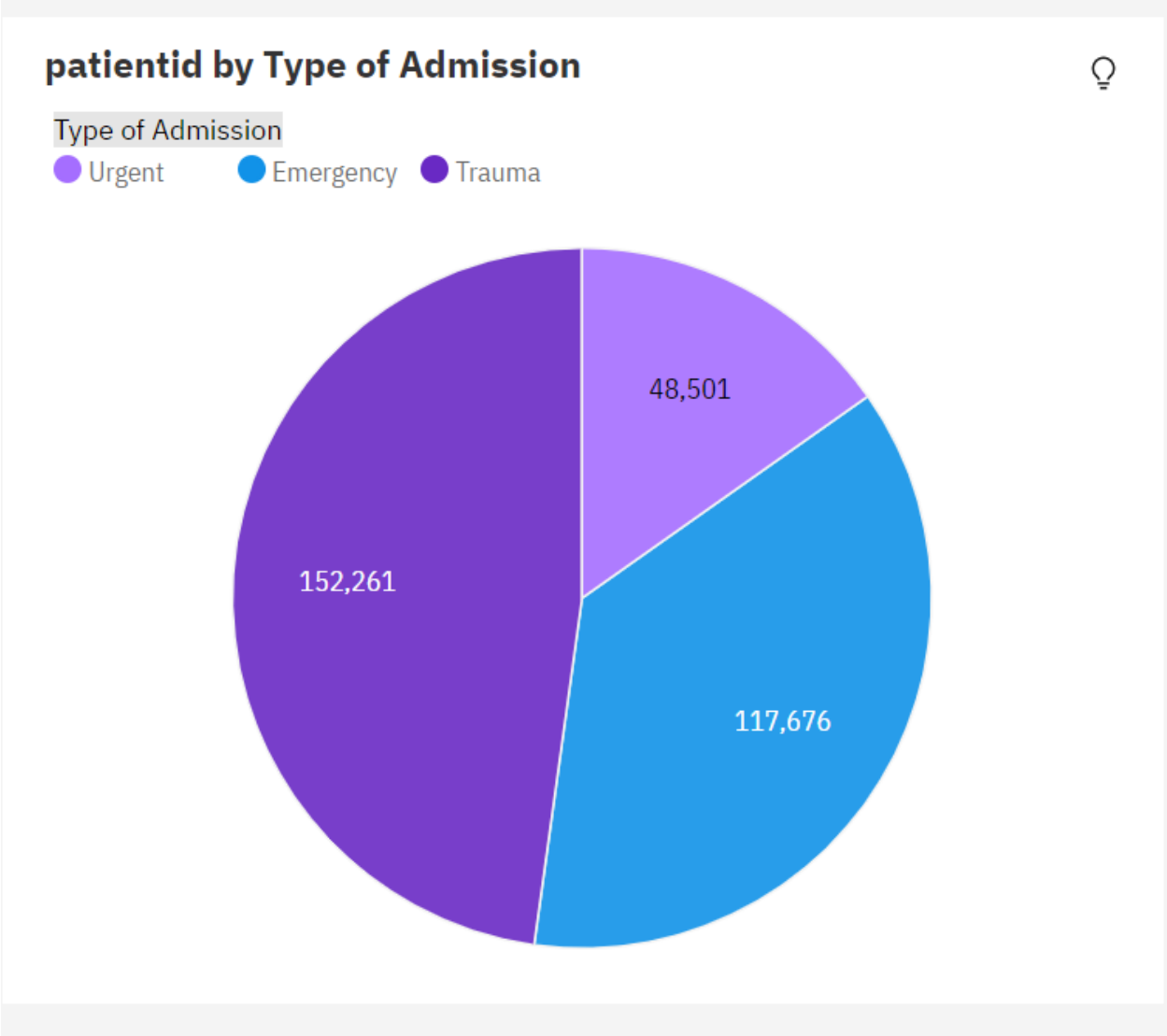
No of patients by no of days in stay colored by severity of Illness



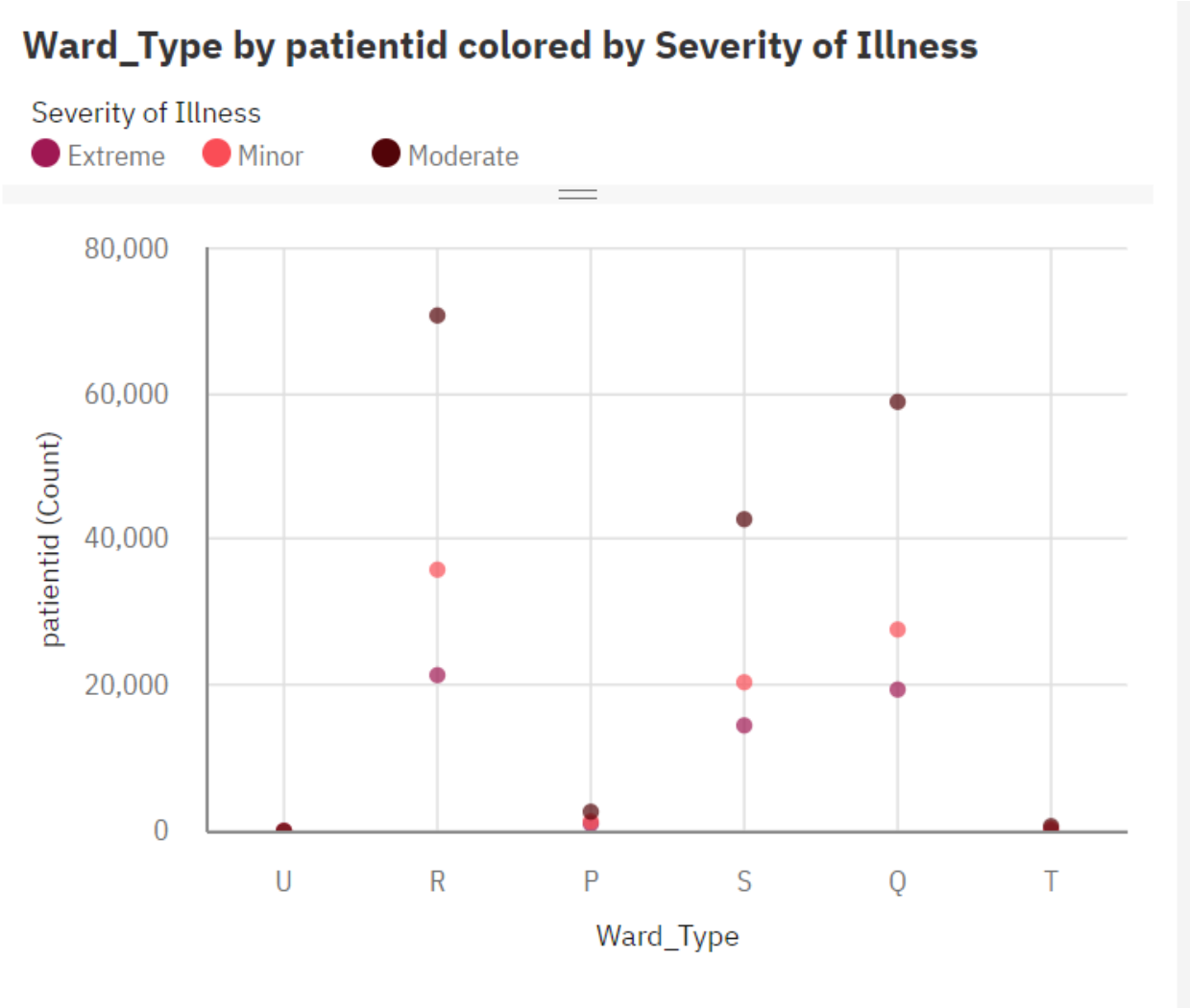
No of patients by severity of disease



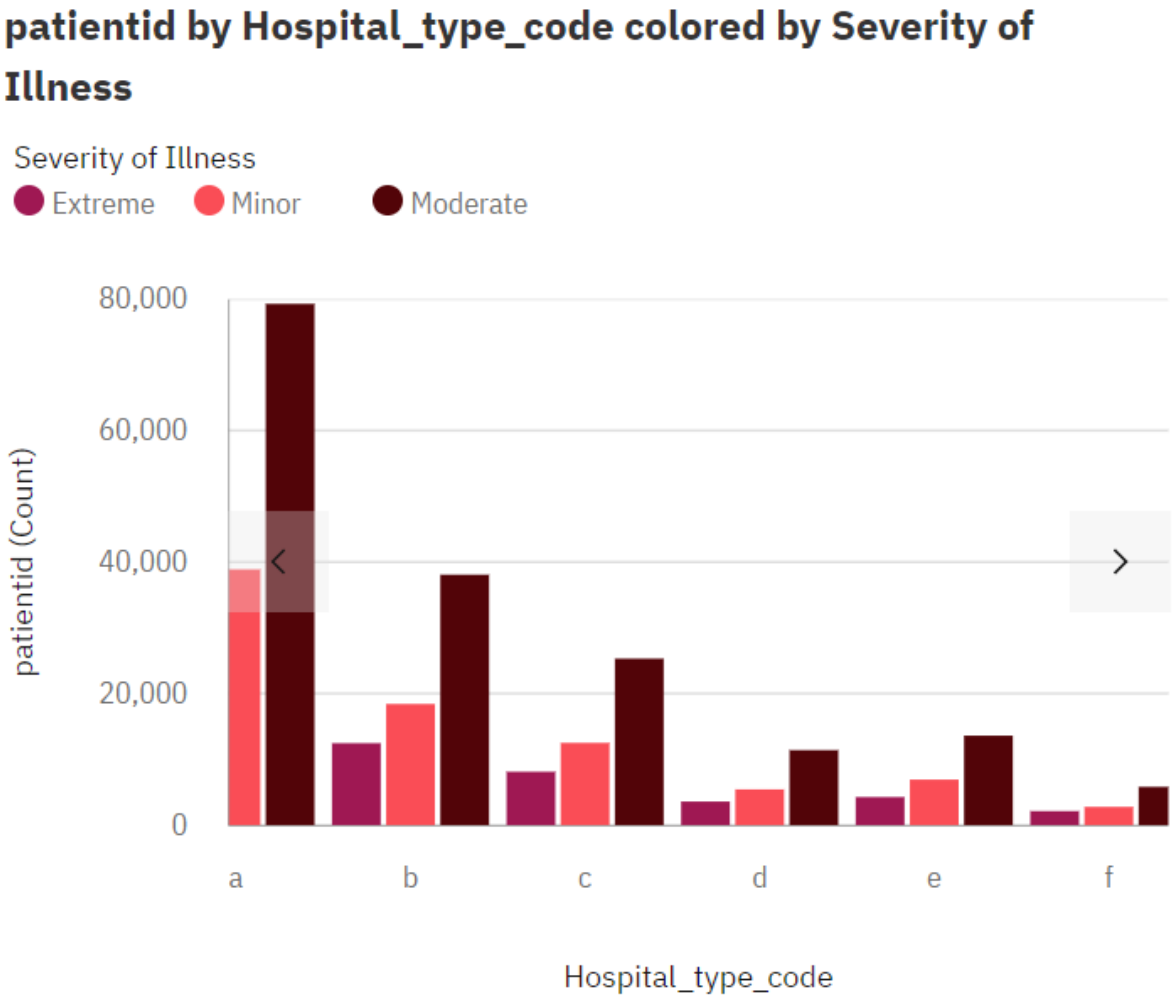
No of patients and type of admission:



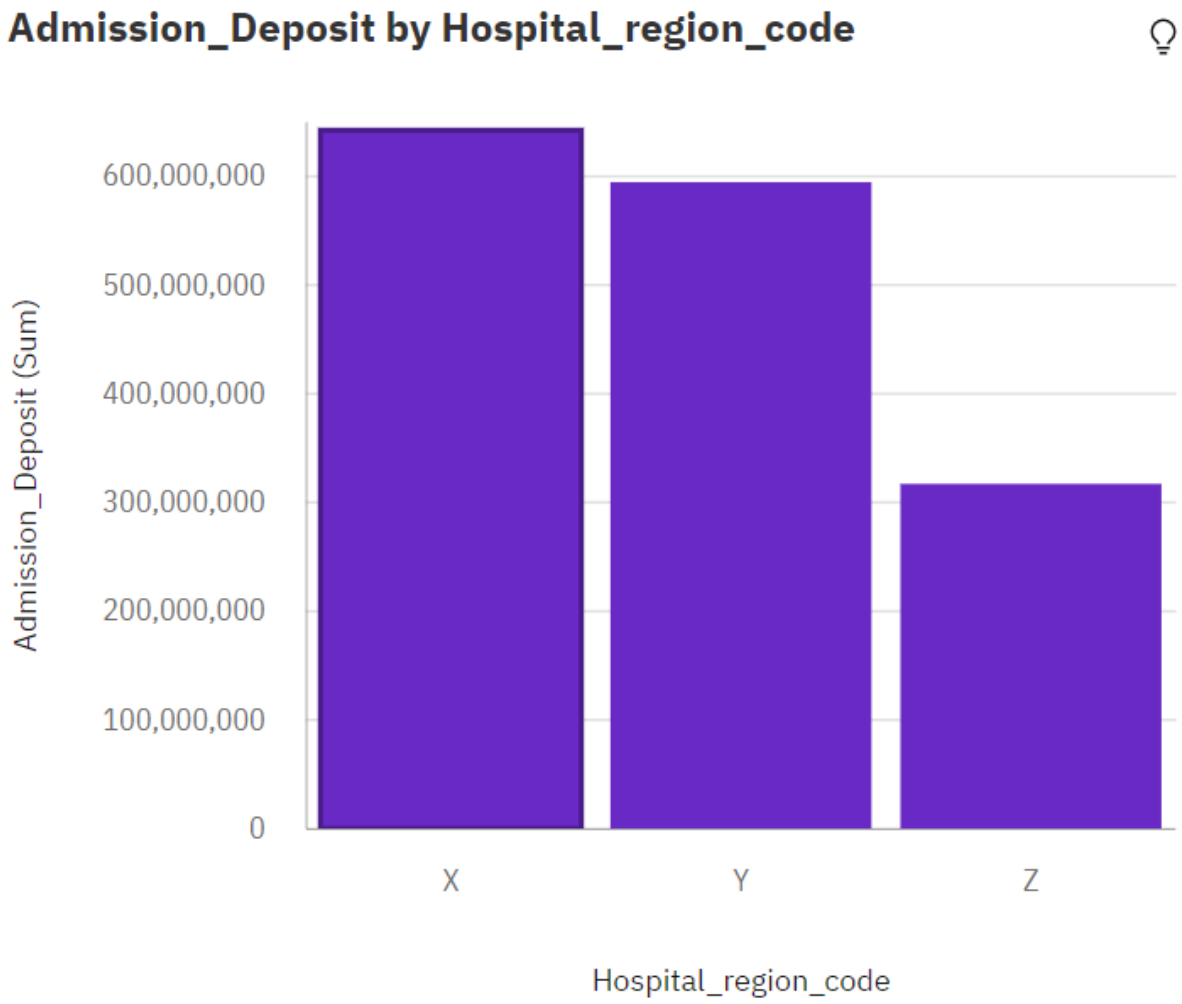
No of patients by ward type colored by severity of illness



No of patients in a hospital type colored by Severity of Illness



Admission Deposit by Hospital region code:



No of patients by city code hospital colored by age

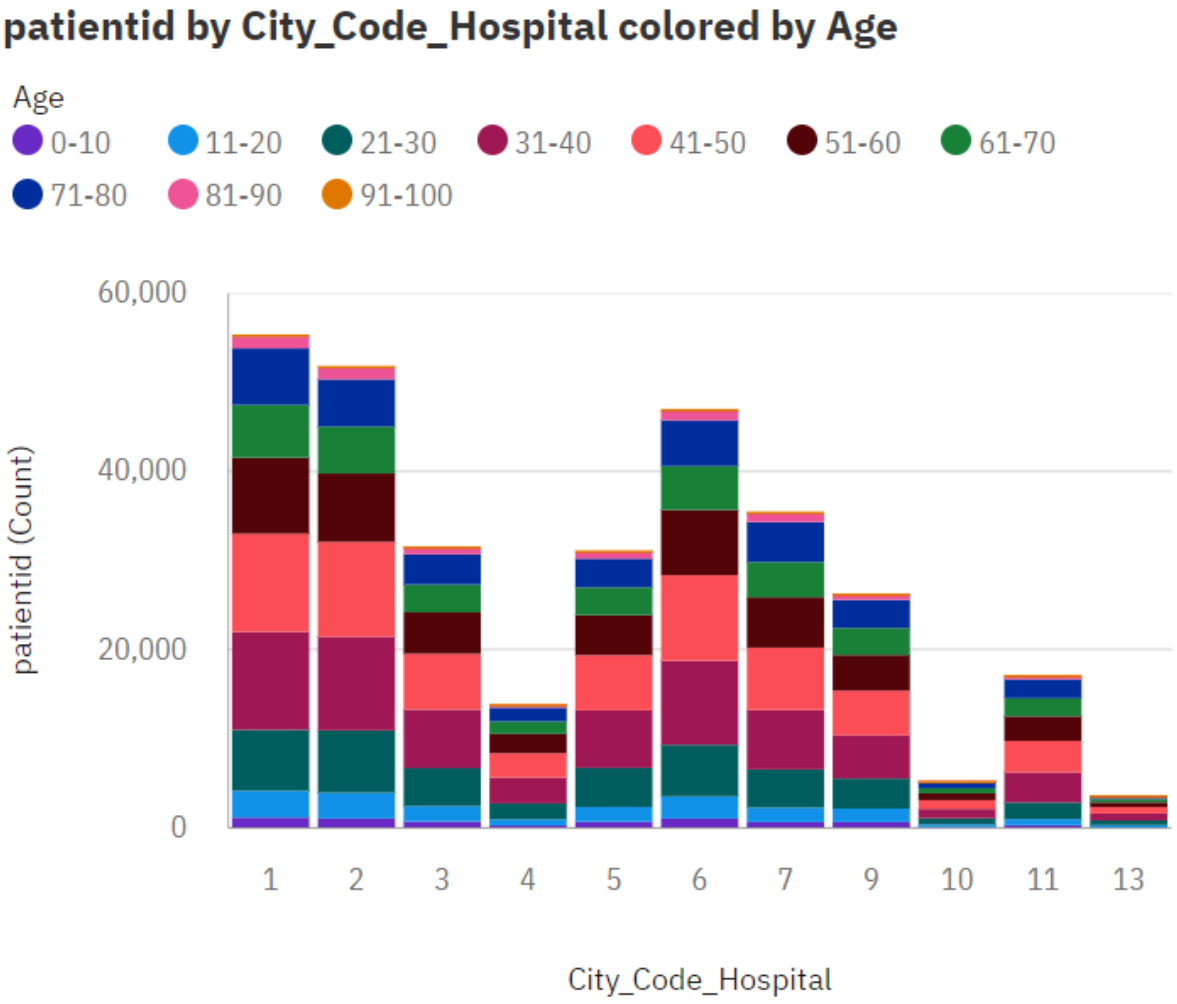
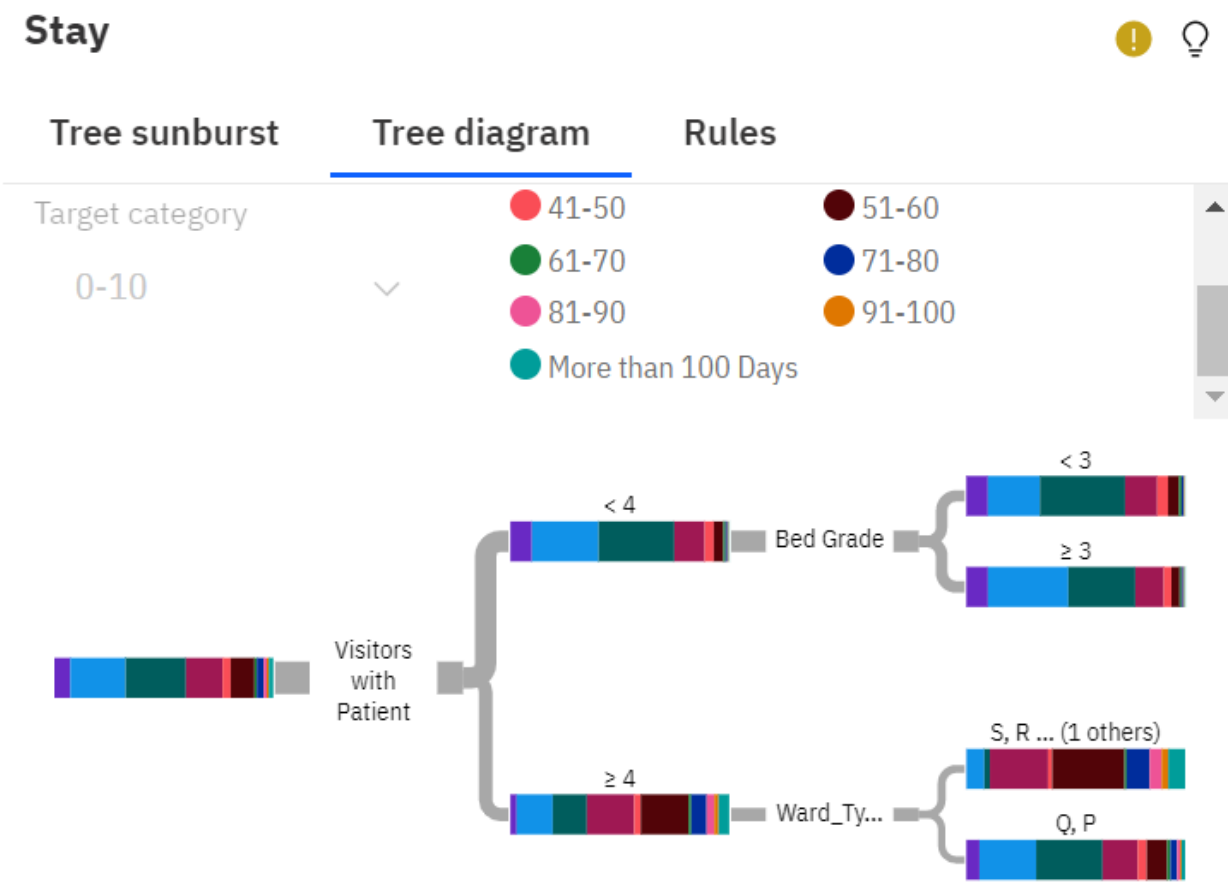


Table for age and no of patients in each age category:

Age and Hospital_region_code

Hospital_region_code	Age	Age
X	0-10	2,571
	11-20	6,900
	21-30	16,298
	31-40	26,716
	41-50	26,897
	51-60	20,786
	61-70	14,246
	71-80	15,167
	81-90	3,236
	91-100	519
Summary		133,336

Decision Tree:



Data Cleaning

DATA CLEANING - HANDLING NULL VALUES

[9]
```
df["Bed Grade"].value_counts()
```

```
2.0    123671
3.0    110583
4.0     57566
1.0     26505
Name: Bed Grade, dtype: int64
```

[10]
```
df["Bed Grade"].fillna(2.0, inplace=True)
```

[11]
```
df["Bed Grade"].isna().sum()
```

```
0
```

[12]
```
df["City_Code_Hospital"].value_counts()
```

```
1     55351
2     51809
6     46991
7     35463
3     31569
5     31105
9     26277
11    17137
4     13857
10     5249
13     3630
Name: City_Code_Hospital, dtype: int64
```

[13]
```
df["City_Code_Patient"].fillna(1, inplace=True)
```

```
df.isna().sum()
```

```
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                           0
City_Code_Patient                   0
Type of Admission                   0
Severity of Illness                 0
Visitors with Patient               0
Age                                 0
Admission_Deposit                   0
Stay                                0
dtype: int64
```

Dropping unnecessary attributes

[8]

```
df.drop(['case_id', 'patientid'], axis=1, inplace=True)
```

Correlations in data

```
df.corr()
```

[7]

Visualize

	case_id float64 -0.045972258824...	Hospital_code fl... -0.059638371212...	City_Code_Hosp... -0.049309083896...	Available Extra ... -0.143739099539...	Bed Grade float -0.115867687
case_id	1.0	-0.043022506716140094	-0.011351672166076155	0.04258049297268307	0.0137019181
Hospital_code	-0.043022506716140094	1.0	0.12829362483898515	-0.059638371212683096	-0.0137389593
City_Code_Ho...	-0.011351672166076155	0.12829362483898515	1.0	-0.04577097039417465	-0.0493090830
Available Extr...	0.04258049297268307	-0.059638371212683096	-0.04577097039417465	1.0	-0.115867682
Bed Grade	0.013701912168819724	-0.013738959637931984	-0.049309083896303674	-0.11586768762021678	
patientid	-0.004149891023962355	0.0022906150968089894	0.0007503734599816436	0.0009209169929943411	0.00164487639
City_Code_Pa...	0.06519606662981017	-0.01552984445298396	-0.023988370312 9584	-0.009680986398408104	-0.0081054475
Visitors with...	0.0013089431184283259	-0.028500291008472723	0.018184441177590032	0.09671435368821796	0.088945364
Admissi	-0.045972258824	0.045445524001	-0.034455292791	-0.143739099539	0.07383255

Ready

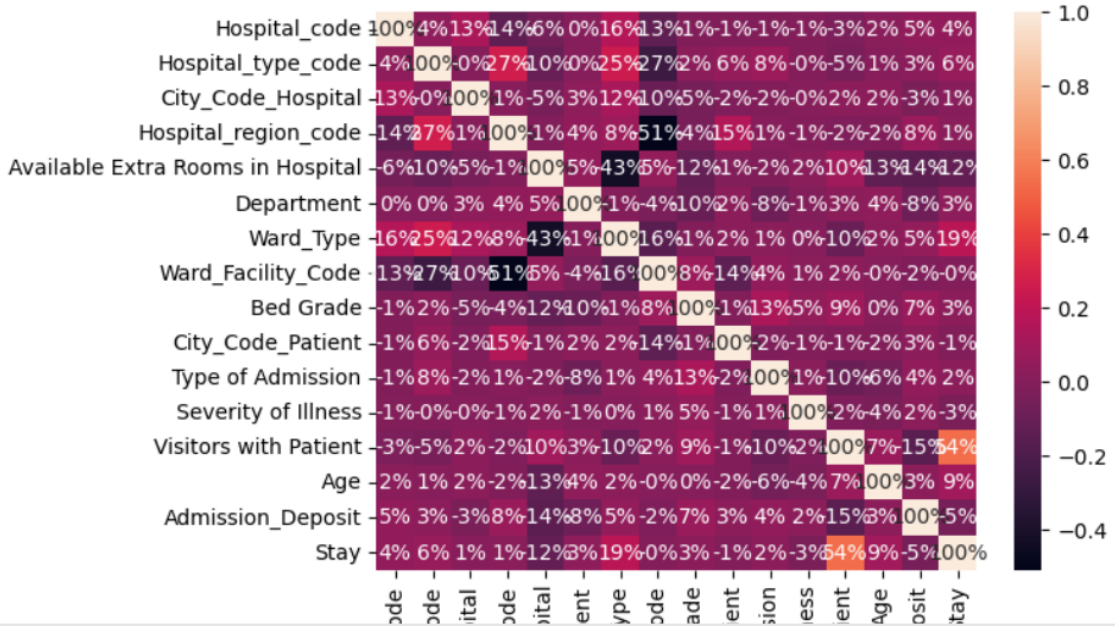
Run notebook

[42]

```
import seaborn

seaborn.heatmap(df.corr(), annot=True, fmt='.0%')
```

```
<AxesSubplot: >
```



DATA TRANSFORMATION

[16]
```
from sklearn import preprocessing
```

[17]
```
le = preprocessing.LabelEncoder()
```

[18]
```
le.fit(df["Stay"])
```

```
LabelEncoder()
```

[19]
```
transformed = le.transform(df["Stay"])
```

[20]
```
df["Stay"] = transformed
```

[21]

```
le.fit(df["Age"])
```

LabelEncoder

LabelEncoder()

[22]

```
age_transformed = le.transform(df["Age"])
```

[23]

```
df["Age"] = age_transformed
```

[24]

```
le.fit(df["Hospital_region_code"])
hrc_transformed = le.transform(df["Hospital_region_code"])
df["Hospital_region_code"] = hrc_transformed
```

[25]

```
le.fit(df["Department"])
dept_transformed = le.transform(df["Department"])
df["Department"] = dept_transformed
```

[24]
```
le.fit(df["Hospital_region_code"])
hrc_transformed = le.transform(df["Hospital_region_code"])
df["Hospital_region_code"] = hrc_transformed
```

[25]
```
le.fit(df["Department"])
dept_transformed = le.transform(df["Department"])
df["Department"] = dept_transformed
```

[26]
```
le.fit(df["Ward_Type"])
wt_transformed = le.transform(df["Ward_Type"])
df["Ward_Type"] = wt_transformed
```

[27]
```
le.fit(df["Ward_Facility_Code"])
wfc_transformed = le.transform(df["Ward_Facility_Code"])
df["Ward_Facility_Code"] = wfc_transformed
```

Transformed data

[31]
```
df.head()
```

Visualize

	Hospital_code in...	Hospital_type_c...	City_Code_Hosp...	Hospital_region...	Available Extra
0	8	2	3	2	
1	2	2	5	2	
2	10	4	1	0	
3	26	1	2	1	
4	26	1	2	1	

5 rows, showing 10 per page Page 1 of 1