```
#import the librarys needed include the Panda for opening the file and seaborn and matplotlib.pyplot for plotting
#and chi2_contingency from scipy.stats for hypothesis testing
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

#Setting the option to display all columns for preview
pd.set_option("display.max.columns", None)

#Open the dataset and print the variables info
inpatient = pd.read_csv("Hospital_Inpatient_Discharges__SPARCS_De-Identified___2021_20231012.csv", low_memory=False)

#Examine the dataset
inpatient.info()

#fill null values
for col in inpatient.columns:
    inpatient[col] = inpatient[col].fillna(inpatient[col].mode()[0])

#Selected needed columns only
```

```
inpatient = inpatient[['Facility Name', 'Age Group', 'Gender' , 'Race', 'Ethnicity', 'Length of Stay', 'Type of Admission',
                       'Patient Disposition', 'CCSR Diagnosis Description', 'APR DRG Description',
                       'APR Severity of Illness Description', 'APR Medical Surgical Description', 'Payment Typology 1',
                           'Payment Typology 2', 'Payment Typology 3']]

    #create a new column called Insured
    inpatient["Insured"] = inpatient['Payment Typology 1'] == "Self-Pay"

    #display the first 20 records
    inpatient.head(20)

    #made the values of 120+ 121 and change the datatype of Length of Stay to numbers
    inpatient["Length of Stay"] = inpatient["Length of Stay"].replace('120 +', 121)
    inpatient["Length of Stay"] = pd.to_numeric(inpatient["Length of Stay"])


    #remove all records of death people
    inpatient = inpatient[inpatient['Patient Disposition'] != "Expired"]

    #show the summary descriptive values of the column Length of stay
```

```
inpatient['Length of Stay'].describe()

# check the distribution of the Length of stay variable
plt.figure(figsize=(10, 6))
sns.histplot(data=inpatient, x='Length of Stay', bins = 50)
plt.title('Distribution of Length of Stay')
plt.ylabel('Count')
plt.grid(True)
plt.show()

#Distribution of the length of stay by payment typology
ax=sns.barplot(x="Payment Typology 1", y="Length of Stay", data=inpatient,
errorbar=None)
ax.bar_label(ax.containers[0], fontsize=10);
plt.xticks(rotation=90)
plt.title('Length of Stay vs. Primary Payment')
plt.show()

#creating a crosstab table
inpatient_tab = pd.crosstab(inpatient['Length of Stay'], inpatient['Insured'], margins =
True)
inpatient_tab
```

```
# label encoding
from sklearn.preprocessing import LabelEncoder
for col in inpatient.columns:
    inpatient_le = LabelEncoder()
    inpatient[col] = inpatient_le.fit_transform(inpatient[col])
inpatient.head()

from sklearn.feature_selection import chi2
x = inpatient.drop(columns=['Length of Stay'], axis=1)
y = inpatient['Length of Stay']
chi_scores = chi2(x, y)
chi_scores

pd.options.plotting.backend = "plotly"
#a plot of the chi_scores,  higher the chi value, higher the importance
chi_values = pd.Series(chi_scores[0], index=x.columns)
chi_values.sort_values(ascending=False, inplace=True)
fig = chi_values.plot.bar()
fig.update_yaxes(tickformat=".0s").show() # show number as is

import pandas as pd
#pd.options.plotting.backend = "plotly"
# if p-value > 0.05, lower the importance
```

```
p_values = pd.Series(chi_scores[1], index=x.columns)

p_values.sort_values(ascending=False, inplace=True)


p_values.plot.bar()

fig.update_yaxes(tickformat=".0s").show() # show number as is
```