In [92]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
```

In [93]:
```
%matplotlib inline
```

In [94]:
```
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
```

In [138]:
```
#upload data
df = pd.read_excel(r"C:\Users\Sierra\Documents\los_and_readmission.xlsx"
)
```

In [140]:
```
#examine columns in data set
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 146606 entries, 0 to 146605
Data columns (total 40 columns):
ENCOUNTER_KEY               146606 non-null int64
PATIENT_NUMBER              146606 non-null int64
gender                      146606 non-null object
race_cd                     146606 non-null object
PatientAge                  146606 non-null int64
Diagnosis_Group             146606 non-null object
icd9_target                 146606 non-null int64
DRG_APR_CODE                146606 non-null object
DRG_APR_DESC                146606 non-null object
DRG_APR_SEVERITY            146413 non-null float64
DIAGNOSIS_SUBCAT_CODE       146606 non-null int64
DIAGNOSIS_SUBCAT_DESC       146606 non-null object
DIAGNOSIS_ICD_CODE          146606 non-null float64
PROCEDURE_SUBCAT_CODE       99875 non-null float64
PROCEDURE_SUBCAT_DESC       99875 non-null object
PROCEDURE_ICD_CODE          99875 non-null float64
DOCTOR                      146606 non-null int64
ADMIT_DATE                  146606 non-null datetime64[ns]
DISCHARGE_DATE              146606 non-null datetime64[ns]
readmit_date                22642 non-null datetime64[ns]
readmit_discharge_date      22642 non-null datetime64[ns]
readmit_days                22642 non-null object
LENGTH_OF_STAY              146606 non-null int64
ICU_DAYS                    146606 non-null int64
DISCHARGED_TO               146606 non-null object
op_visits6                  146606 non-null int64
Standard_Orders_Used        146606 non-null object
Num_Chronic_Cond            146413 non-null float64
Disch_Nurse_ID              146606 non-null int64
admit_month                 146606 non-null int64
readmit_month               22642 non-null float64
order_set_used              146606 non-null int64
order_total_charges         146606 non-null int64
readmit_number              146606 non-null int64
operationcount              146606 non-null int64
HOSPITAL                    146606 non-null object
ZIP                         146606 non-null int64
STATECODE                   146606 non-null object
City                        146606 non-null object
County_name                 146606 non-null object
dtypes: datetime64[ns](4), float64(6), int64(16), object(14)
memory usage: 44.7+ MB
```

In [141]:
```
#examine first few rows of data
df.head()
```

Out[141]:

	ENCOUNTER_KEY	PATIENT_NUMBER	gender	race_cd	PatientAge	Diagnosis_Group	icd9_targ
0	105240011	9921900011	F	White	87	CHF	
1	105240017	9921900017	M	White	67	CHF	
2	105240019	9921900019	M	White	68	CHF	
3	105240021	9921900021	F	White	72	AMI	
4	105240029	9921900029	F	White	75	CHF	

In [142]:
```
#examine descriptive statistics
df[['ENCOUNTER_KEY','PATIENT_NUMBER','PatientAge','icd9_target','DRG_APR
_SEVERITY']].describe()
```

Out[142]:

	ENCOUNTER_KEY	PATIENT_NUMBER	PatientAge	icd9_target	DRG_APR_SEVERITY
count	1.466060e+05	1.466060e+05	146606.000000	146606.000000	146413.000000
mean	1.053277e+08	9.921988e+09	74.440848	0.876172	2.539945
std	4.247076e+04	4.247076e+04	13.267879	0.329387	0.746014
min	1.052400e+08	9.921900e+09	27.000000	0.000000	1.000000
25%	1.052911e+08	9.921951e+09	69.000000	1.000000	2.000000
50%	1.053278e+08	9.921988e+09	76.000000	1.000000	3.000000
75%	1.053644e+08	9.922024e+09	83.000000	1.000000	3.000000
max	1.054011e+08	9.922061e+09	101.000000	1.000000	4.000000

In [143]:
```
#examine descriptive statistics
df[['DIAGNOSIS_SUBCAT_CODE','PROCEDURE_SUBCAT_CODE','DOCTOR','LENGTH_OF_
STAY']].describe()
```

Out[143]:

	DIAGNOSIS_SUBCAT_CODE	PROCEDURE_SUBCAT_CODE	DOCTOR	LENGTH_OF_STAY
count	146606.000000	99875.000000	146606.000000	146606.000000
mean	446.938972	76.542148	268245.745208	5.428925
std	33.820447	24.928839	37313.516974	4.100839
min	31.000000	0.000000	201620.000000	1.000000
25%	428.000000	39.000000	235415.000000	3.000000
50%	428.000000	88.000000	268058.000000	4.000000
75%	486.000000	89.000000	297501.000000	7.000000
max	783.000000	99.000000	344581.000000	32.000000

In [144]:
```
#examine descriptive statistics
df[['ICU_DAYS','op_visits6','Num_Chronic_Cond','Disch_Nurse_ID','admit_m
onth']].describe()
```

Out[144]:

	ICU_DAYS	op_visits6	Num_Chronic_Cond	Disch_Nurse_ID	admit_month
count	146606.000000	146606.000000	146413.000000	146606.000000	146606.000000
mean	2.785398	2.464674	1.018550	157374.126925	5.916504
std	3.832078	2.965829	0.937125	158107.900859	3.635043
min	0.000000	0.000000	0.000000	1001.000000	1.000000
25%	0.000000	0.000000	0.000000	13005.000000	3.000000
50%	2.000000	2.000000	1.000000	100292.000000	5.000000
75%	4.000000	4.000000	1.000000	310012.000000	10.000000
max	29.000000	31.000000	4.000000	827298.000000	12.000000

In [145]:
```
#examine descriptive statistics
df[['readmit_month','order_set_used','order_total_charges','readmit_numb
er','operationcount','ZIP']].describe()
```

Out[145]:

	readmit_month	order_set_used	order_total_charges	readmit_number	operationcount	
count	22642.000000	146606.000000	146606.000000	146606.000000	146606.000000	14660
mean	6.142523	0.802511	28723.934020	0.154441	0.787962	5530
std	3.470503	0.398105	8474.330612	0.361372	0.806079	114
min	1.000000	0.000000	-2104.000000	0.000000	0.000000	5064
25%	3.000000	1.000000	22306.000000	0.000000	0.000000	5540
50%	6.000000	1.000000	27839.000000	0.000000	1.000000	5540
75%	9.000000	1.000000	34368.000000	0.000000	1.000000	5543
max	12.000000	1.000000	67671.000000	1.000000	6.000000	5807

In [146]:
```
#drop uselss and data leakage into the future numeric columns
df.drop(['ENCOUNTER_KEY','PATIENT_NUMBER','icd9_target','DOCTOR','Disch_
Nurse_ID','readmit_month',
'order_total_charges','readmit_number','ZIP'], inplace = True, axis = 1)
```

In [147]:
```
#check for linear relationships among columns
df.corr()
```

Out[147]:

	PatientAge	DRG_APR_SEVERITY	DIAGNOSIS_SUBCAT_CODE	DIAGN
PatientAge	1.000000	0.019316	-0.023399	
DRG_APR_SEVERITY	0.019316	1.000000	-0.005591	
DIAGNOSIS_SUBCAT_CODE	-0.023399	-0.005591	1.000000	
DIAGNOSIS_ICD_CODE	-0.023311	-0.005804	0.999981	
PROCEDURE_SUBCAT_CODE	0.015013	-0.187127	-0.196678	
PROCEDURE_ICD_CODE	0.014747	-0.186313	-0.199526	
LENGTH_OF_STAY	-0.005383	0.234020	-0.005533	
ICU_DAYS	-0.007687	0.229224	0.018182	
op_visits6	0.013638	0.023032	-0.010198	
Num_Chronic_Cond	-0.027101	-0.013398	-0.064415	
admit_month	-0.023886	-0.050489	-0.014885	
order_set_used	0.027166	-0.001746	-0.031455	
operationcount	-0.007898	-0.007982	-0.040662	

In [148]:
```
#check values and counts of column
df['gender'].value_counts().to_dict()
```

Out[148]:
```
{'F': 83002, 'M': 63604}
```

In [149]:
```
#check values and counts of column
df['race_cd'].value_counts().to_dict()
```

Out[149]:
```
{'White': 125231, 'Black': 13849, 'Others': 7526}
```

In [150]:
```
#check values and counts of column
df['Diagnosis_Group'].value_counts().to_dict()
```

Out[150]:
```
{'CHF': 95270, 'AMI': 40143, 'COPD': 11193}
```

In [151]:
```
#check values and counts of column
df['DIAGNOSIS_SUBCAT_DESC'].value_counts().to_dict()
```

Out[151]:
```
{'HEART FAILURE': 87828,
 'PNEUMONIA ORGANISM UNSP': 32531,
 'CHRONIC BRONCHITIS': 7761,
 'OTHER BACTERIAL PNEUMONI': 3763,
 'PNEUMOCOCCAL PNEUMONIA': 2985,
 'HYPERTENSIVE HEART AND C': 2904,
 'HYPERTENSIVE HEART DISEA': 1744,
 'OTHER RHEUMATIC HEART DI': 1635,
 'ASTHMA': 1159,
 'BRONCHIECTASIS': 1016,
 'EMPHYSEMA': 820,
 'VIRAL PNEUMONIA': 576,
 'ACUTE MYOCARDIAL INFARCT': 481,
 'CHRONIC AIRWAY OBSTRUCTI': 388,
 'CHRONIC Hosp 46 HEART': 291,
 'PNEUMONIA DUE TO OTHER S': 192,
 'SYMPTOMS CONCERNING NUTR': 98,
 'OTHER AND UNSPECIFIED DI': 97,
 'SEPTICEMIA': 97,
 'BRONCHOPNEUMONIA ORGANI': 96,
 'DISORDERS OF FLUID ELEC': 95,
 'DISEASES DUE TO OTHER MY': 49}
```

In [152]:
```
#notice DIAGNOSIS_SUBCAT_DESC and DIAGNOSIS_SUBCAT_CODE have same number
of unique values
len(df['DIAGNOSIS_SUBCAT_DESC'].unique())
```

Out[152]: 22

In [153]:
```
len(df['DIAGNOSIS_SUBCAT_CODE'].unique())
```

Out[153]: 22

In [154]:
```
#check values and counts of column
df['DRG_APR_DESC'].value_counts().to_dict()
```

Out[154]:
```
{'HEART FAILURE': 93725,
 'OTHER PNEUMONIA': 34636,
 'CHRONIC OBSTRUCTIVE Hosp 46 DISEASE': 9611,
 'MAJOR RESPIRATORY INFECTIONS & INFLAMMATIONS': 2271,
 'CYSTIC FIBROSIS - Hosp 46 DISEASE': 2012,
 'RESPIRATORY SYSTEM DIAGNOSIS W VENTILATOR SUPPORT 96+ HOURS': 961,
 'ACUTE MYOCARDIAL INFARCTION': 481,
 'HIV W MAJOR HIV RELATED CONDITION': 390,
 'HIV W ONE SIGNIF HIV COND OR W/O SIGNIF RELATED COND': 389,
 'OTHER CIRCULATORY SYSTEM DIAGNOSES': 291,
 'OTHER RESPIRATORY & CHEST PROCEDURES': 290,
 'NODATA': 193,
 'TRACHEOSTOMY W LONG TERM MECHANICAL VENTILATION W/O EXTENSIVE PROCEDU
R': 193,
 'CARDIAC CATHETERIZATION W CIRC DISORD EXC ISCHEMIC HEART DISEASE': 19
3,
 'BPD & OTH CHRONIC RESPIRATORY DISEASES ARISING IN PERINATAL PERIOD': 9
8,
 'EXTENSIVE PROCEDURE UNRELATED TO PRINCIPAL DIAGNOSIS': 98,
 'MALNUTRITION, FAILURE TO THRIVE & OTHER NUTRITIONAL DISORDERS': 98,
 'SEPTICEMIA & DISSEMINATED INFECTIONS': 97,
 'MODERATELY EXTENSIVE PROCEDURE UNRELATED TO PRINCIPAL DIAGNOSIS': 97,
 'MAJOR RESPIRATORY & CHEST PROCEDURES': 97,
 'CONNECTIVE TISSUE DISORDERS': 97,
 'BRONCHIOLITIS & RSV PNEUMONIA': 97,
 'TRACHEOSTOMY W LONG TERM MECHANICAL VENTILATION W EXTENSIVE PROCEDUR
E': 96,
 'ELECTROLYTE DISORDERS EXCEPT HYPOVOLEMIA RELATED': 95}
```

In [155]:
```
#notice DIAGNOSIS_SUBCAT_DESC and DIAGNOSIS_SUBCAT_CODE have same number
of unique values
len(df['DRG_APR_DESC'].unique())
```

Out[155]: 24

In [156]:
```
len(df['DRG_APR_CODE'].unique())
```

Out[156]: 24

In [157]:
```
#notice DRG_APR_CODE is not stored as numeric due to @@@@@
#no need to fix since it is a duplicate of DRG_APR_DESC
df['DRG_APR_CODE'].value_counts().to_dict()
```

Out[157]:
```
{'194': 93725,
 '139': 34636,
 '140': 9611,
 '137': 2271,
 '131': 2012,
 '130': 961,
 '190': 481,
 '892': 390,
 '894': 389,
 '207': 291,
 '121': 290,
 '191': 193,
 '@@@@@': 193,
 '005': 193,
 '950': 98,
 '421': 98,
 '132': 98,
 '138': 97,
 '720': 97,
 '120': 97,
 '346': 97,
 '951': 97,
 '004': 96,
 '425': 95}
```

In [158]:
```
#check values and counts of column
df['PROCEDURE_SUBCAT_DESC'].value_counts().to_dict()
```

Out[158]:
```
{'OTHER DIAGNOSTIC RADIOLO': 47026,
 'OTHER NONOPERATIVE PROCE': 17120,
 'INCISION, EXCISION, AND': 11635,
 'OTHER OPERATIONS ON VESS': 5154,
 'NONOPERATIVE INTUBATION': 5016,
 'OTHER OPERATIONS ON LUNG': 4053,
 'OPERATIONS ON CHEST WALL': 2462,
 'INTERVIEW, EVALUATION, C': 2310,
 'OTHER OPERATIONS ON HEAR': 821,
 'NUCLEAR MEDICINE': 811,
 'OPERATIONS ON SPINAL COR': 578,
 'PROCEDURES RELATED TO TH': 383,
 'OTHER OPERATIONS ON LARY': 336,
 'OTHER OPERATIONS ON ABDO': 293,
 'OPERATIONS ON BONE MARRO': 288,
 'OPERATIONS ON SKIN AND S': 285,
 'OPERATIONS ON NOSE': 195,
 'REPAIR AND PLASTIC OPERA': 193,
 'PROCEDURES AND INTERVENT': 192,
 'INCISION AND EXCISION OF': 146,
 'REPLACEMENT AND REMOVAL': 98,
 'OPERATIONS ON LYMPHATIC': 97,
 'OPERATIONS ON RECTUM REC': 97,
 'OPERATIONS ON ANUS': 96,
 'OTHER OPERATIONS ON TEET': 96,
 'OPERATIONS ON LIVER': 94}
```

In [159]:
```
#notice PROCEDURE_SUBCAT_DESC and PROCEDURE_SUBCAT_CODE have almost same
number of unique values
len(df['PROCEDURE_SUBCAT_DESC'].unique())
```

Out[159]: 27

In [160]:
```
len(df['PROCEDURE_SUBCAT_CODE'].unique())
```

Out[160]: 28

In [161]:
```
#notice ADMIT_DATE has too many unique values to examine count of each
len(df['ADMIT_DATE'].unique())
```

Out[161]: 317

In [162]:
```
#check values and counts for column
df['Standard_Orders_Used'].value_counts().to_dict()
```

Out[162]:
```
{'Y': 117653, 'N': 28953}
```

In [163]:
```
#check values and counts for column
df['HOSPITAL'].value_counts().to_dict()
```

Out[163]:
```
{'St. Anthony Medical Center': 69577,
 'Mercy Hospital': 34840,
 'Hilding-Long Memorial Hospital': 18109,
 'Oxbow Regional Hospital': 9133,
 'Independence Medical Center': 5787,
 'Superior-Parkland Hospital': 5113,
 'Valley City Regional Hospital': 2601,
 'Delaware County Hospital': 1446}
```

In [164]:
```
#check values and counts for column
df['STATECODE'].value_counts().to_dict()
```

Out[164]:
```
{'MN': 122526, 'WI': 14246, 'IA': 7233, 'ND': 2601}
```

In [165]:
```
#check values and counts for column
df['City'].value_counts().to_dict()
```

Out[165]:
```
{'Minneapolis': 69577,
 'Bloomington': 34840,
 'Park Rapids': 18109,
 'Eau Claire': 9133,
 'Waterloo': 5787,
 'Parkland': 5113,
 'Valley City': 2601,
 'Manchester': 1446}
```

In [166]:
```
#check values and counts for column
df['County_name'].value_counts().to_dict()
```

Out[166]:
```
{'Hennepin': 104417,
 'Hubbard': 18109,
 'Eau Claire': 9133,
 'Black Hawk': 5787,
 'Douglas': 5113,
 'Barnes': 2601,
 'Delaware': 1446}
```

In [167]:
```
#drop columns with data leakage into the future
#drop redundant and useless columns
df.drop(['Diagnosis_Group','DRG_APR_CODE','DRG_APR_DESC','DIAGNOSIS_SUBC
AT_DESC','DIAGNOSIS_SUBCAT_CODE','DIAGNOSIS_ICD_CODE','PROCEDURE_SUBCAT_
CODE','PROCEDURE_ICD_CODE','STATECODE','City','County_name','ADMIT_DATE'
,'DISCHARGE_DATE','readmit_date','readmit_discharge_date','readmit_days'
,'DISCHARGED_TO'], inplace = True, axis = 1)
```

In [168]:
```
#null value check
df.isnull().sum()/len(df)*100
```

Out[168]:
```
gender                   0.000000
race_cd                  0.000000
PatientAge               0.000000
DRG_APR_SEVERITY         0.131645
PROCEDURE_SUBCAT_DESC   31.875230
LENGTH_OF_STAY           0.000000
ICU_DAYS                 0.000000
op_visits6               0.000000
Standard_Orders_Used     0.000000
Num_Chronic_Cond         0.131645
admit_month              0.000000
order_set_used           0.000000
operationcount           0.000000
HOSPITAL                 0.000000
dtype: float64
```

In [169]:
```
#drop null values for DRG_APR_SEVERITY
df = df[df['DRG_APR_SEVERITY'].notna()]
```

In [170]:
```
#drop null values for Num_Chronic_Cond
df = df[df['Num_Chronic_Cond'].notna()]
```

In [171]:
```
#impute null values of PROCEDURE_SUBCAT_DESCn with No Procedure
df['PROCEDURE_SUBCAT_DESC'].fillna('No Procedure', inplace = True)
```

In [172]:
```
#null value check
df.isnull().sum()/len(df)*100
```

Out[172]:
```
gender                  0.0
race_cd                 0.0
PatientAge              0.0
DRG_APR_SEVERITY        0.0
PROCEDURE_SUBCAT_DESC   0.0
LENGTH_OF_STAY          0.0
ICU_DAYS                0.0
op_visits6              0.0
Standard_Orders_Used    0.0
Num_Chronic_Cond        0.0
admit_month             0.0
order_set_used          0.0
operationcount          0.0
HOSPITAL                0.0
dtype: float64
```

In [173]:
```
#get dummy variables and drop first column
gender = pd.get_dummies(df['gender'],drop_first=True)
```

In [174]:
```
#get dummy variables and drop first column
race_cd = pd.get_dummies(df['race_cd'],drop_first=True)
```

In [175]:
```
#get dummy variables and drop first column
PROCEDURE_SUBCAT_DESC = pd.get_dummies(df['PROCEDURE_SUBCAT_DESC'],drop_
first=True)
```

In [176]:
```
#get dummy variables and drop first column
Standard_Orders_Used = pd.get_dummies(df['Standard_Orders_Used'],drop_fi
rst=True)
```

In [177]:
```
#get dummy variables and drop first column
admit_month = pd.get_dummies(df['admit_month'],drop_first=True)
```

In [178]:
```
#get dummy variables and drop first column
HOSPITAL = pd.get_dummies(df['HOSPITAL'],drop_first=True)
```

In [179]:
```
#concatentate dummies with data frame
df = pd.concat([df,gender,race_cd,PROCEDURE_SUBCAT_DESC,Standard_Orders_
Used,admit_month,HOSPITAL],axis=1)
```

In [180]:
```
#drop original dummy variables from data frame
df.drop(['gender','race_cd','PROCEDURE_SUBCAT_DESC','Standard_Orders_Use
d','admit_month','HOSPITAL'], inplace = True, axis = 1)
```

In [181]:
```
#export as csv
df.to_csv(r'C:\Users\Sierra\Documents\Preprocessed_LOS_READM.csv')
```

In [ ]:
```
#Project paused
```

In [ ]:
```
#Project paused
```

In [ ]:
```
#Project paused
```