

```
In [48]: #imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os

In [49]: #allows plots to appear directly in the notebook
%matplotlib inline

In [50]: pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)

In [51]: #upload data
df = pd.read_excel(r"C:\Users\Sierra\Documents\los_and_readmission.xlsx")

In [53]: #shape of data frame
df.shape

Out[53]: (146606, 40)

In [54]: #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 [55]: #examine first few rows of data
df.head()

Out[55]:
   ENCOUNTER_KEY  PATIENT_NUMBER  gender  race_cd  PatientAge  Diagnosis_Group  icd9_target  DRG_APR_CODE  DRG_APR_DESC  DRG_APR_SEVERITY
0      105240011      992190011      F      White           87              CHF              1           194  HEART FAILURE              2.C
1      105240017      992190017      M      White           67              CHF              1           194  HEART FAILURE              1.C
2      105240019      992190019      M      White           68              CHF              1           194  HEART FAILURE              1.C
3      105240021      992190021      F      White           72              AMI              1           139      OTHER PNEUMONIA              2.C
4      105240029      992190029      F      White           75              CHF              1           194  HEART FAILURE              2.C

In [56]: #examine descriptive statistics
df[['ENCOUNTER_KEY', 'PATIENT_NUMBER', 'PatientAge', 'icd9_target', 'DRG_APR_SEVERITY']].describe()

Out[56]:
   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     4.247078e+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      0.000000      2.000000
50%     1.053278e+08      9.921988e+09      76.000000      0.000000      3.000000
75%     1.053644e+08      9.922024e+09      83.000000      0.000000      3.000000
max     1.054011e+08      9.922061e+09     101.000000      0.000000      4.000000

In [57]: #examine descriptive statistics
df[['DIAGNOSIS_SUBCAT_CODE', 'PROCEDURE_SUBCAT_CODE', 'DOCTOR', 'LENGTH_OF_STAY']].describe()

Out[57]:
   DIAGNOSIS_SUBCAT_CODE  PROCEDURE_SUBCAT_CODE  DOCTOR  LENGTH_OF_STAY
count  146606.000000      99875.000000      146606.000000  146606.000000
mean    446.938972          76.542148      208245.745208      5.428925
std     33.820447          24.928839      3713131.6974      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      34581.000000      32.000000

In [58]: #examine descriptive statistics
df[['ICU_DAYS', 'op_visits6', 'Num_Chronic_Cond', 'Disch_Nurse_ID', 'admit_month']].describe()

Out[58]:
   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      2.454674          1.019550  157374.126925      5.916504
std     3.820778      2.965829          0.937125  158107.906869      3.636043
min      0.000000      0.000000          0.000000    1001.000000      1.000000
25%     0.000000      0.000000          0.000000   32005.000000      3.000000
50%     2.000000      1.000000          1.000000  100202.000000      5.000000
75%     4.000000      4.000000          1.000000  100812.000000     10.000000
max     29.000000     31.000000          4.000000  827296.000000     12.000000

In [59]: #examine descriptive statistics
df[['readmit_month', 'order_set_used', 'order_total_charges', 'readmit_number', 'operationcount', 'ZIP']].describe()

Out[59]:
   readmit_month  order_set_used  order_total_charges  readmit_number  operationcount  ZIP
count  22642.000000  146606.000000  146606.000000  146606.000000  146606.000000  146606.000000
mean     6.142523      0.802511    28723.934020      0.154441      0.787962  55306.679195
std     3.470503      0.398105     8474.330612      0.361372      0.806079  1141.573432
min      1.000000      0.000000     -2104.000000      0.000000      0.000000  50644.000000
25%     3.000000      1.000000     22306.000000      0.000000      0.000000  55406.000000
50%     6.000000      1.000000     27839.000000      0.000000      1.000000  55406.000000
75%     9.000000      1.000000     34368.000000      0.000000      1.000000  55437.000000
max     12.000000      1.000000     67671.000000      1.000000      6.000000  58072.000000

In [60]: #drop useless 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 [61]: #check for linear relationships among columns
#note strong correlation between LENGTH_OF_STAY and ICU_DAYS
df.corr()

Out[61]:
           PatientAge  DRG_APR_SEVERITY  DIAGNOSIS_SUBCAT_CODE  DIAGNOSIS_ICD_CODE  PROCEDURE_SUBCAT_CODE  PROCEDURE
PatientAge      1.000000              0.019316              -0.023399              -0.023311              -0.019313
DRG_APR_SEVERITY  0.019316              1.000000              -0.005991              -0.005804              -0.187127
DIAGNOSIS_SUBCAT_CODE -0.023399              -0.005991              1.000000              0.999981              -0.196078
DIAGNOSIS_ICD_CODE  -0.023311              -0.005804              0.999981              1.000000              -0.196403
PROCEDURE_SUBCAT_CODE  0.019313              -0.187127              -0.196078              -0.196403              1.000000
PROCEDURE_ICD_CODE   0.014747              -0.186313              -0.199526              -0.199529              0.999950
LENGTH_OF_STAY     -0.005383              0.234020              -0.005533              -0.005299              -0.127479
ICU_DAYS             0.007687              0.229224              0.018182              0.018440              -0.139828
op_visits6           0.013638              0.023032              -0.010198              -0.010270              -0.005006
Num_Chronic_Cond     -0.027101              -0.013396              -0.064415              -0.064712              0.021335
admit_month          -0.023886              -0.050489              -0.014885              -0.014845              -0.010995
order_set_used        0.027166              -0.001746              -0.031455              -0.031316              0.004383
operationcount        -0.007998              -0.007982              -0.040662              -0.040702              -0.011674

In [62]: #printed: strong correlation between LENGTH_OF_STAY and ICU_DAYS
df[['LENGTH_OF_STAY', 'ICU_DAYS']].corr(df[['ICU_DAYS']])

Out[62]: 0.9877336804958825

In [63]: #check values and counts of column
df['gender'].value_counts().to_dict()

Out[63]: {'F': 83802, 'M': 63604}

In [64]: #check values and counts of column
df['race_cd'].value_counts().to_dict()

Out[64]: {'White': 125231, 'Black': 13849, 'Others': 7526}

In [65]: #check values and counts of column
df['Diagnosis_Group'].value_counts().to_dict()

Out[65]: {'CHF': 95270, 'AMI': 40143, 'COPD': 11193}

In [66]: #check values and counts of column
df['DIAGNOSIS_SUBCAT_DESC'].value_counts().to_dict()

Out[66]: {'HEART FAILURE': 87828,
'PNEUMONIA ORGANISM UNSP': 32531,
'CHRONIC BRONCHITIS': 7761,
'OTHER BACTERIAL PNEUMONIA': 3763,
'PNEUMOCOCCAL PNEUMONIA': 2985,
'HYPERTENSIVE HEART AND C': 2984,
'HYPERTENSIVE HEART DISEA': 1744,
'OTHER RHEUMATIC HEART DI': 1635,
'ASTHMA': 1159,
'BRONCHIECTASIS': 1016,
'EMPHYSEMA': 820,
'VIRAL PNEUMONIA': 576,
'ACUTE MYOCARDIAL INFARCT': 481,
'CHRONIC AIRWAY OBSTRUCT': 388,
'CHRONIC Hosp 48 HEART': 251,
'PNEUMONIA DUE TO OTHER S': 192,
'SYMPTOMS CONCERNING NUTR': 98,
'SEPTICEMIA': 97,
'OTHER AND UNSPECIFIED DI': 97,
'BRONCHOPNEUMONIA ORGANI': 96,
'DISORDERS OF FLUID ELEC': 95,
'DISEASES DUE TO OTHER NY': 49}

In [67]: #notice DIAGNOSIS_SUBCAT_DESC and DIAGNOSIS_SUBCAT_CODE have same number of unique values
len(df['DIAGNOSIS_SUBCAT_DESC'].unique())

Out[67]: 22

In [68]: len(df['DIAGNOSIS_SUBCAT_CODE'].unique())

Out[68]: 22

In [69]: #check values and counts of column
df['DRG_APR_DESC'].value_counts().to_dict()

Out[69]: {'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': 399,
'HIV W ONE SIGNIF HIV COND OR W/O SIGNIF RELATED COND': 389,
'OTHER CIRCULATORY SYSTEM DIAGNOSES': 291,
'OTHER RESPIRATORY & CHEST PROCEDURES': 290,
'NOTATA': 193,
'TRACHEOSTOMY W LONG TERM MECHANICAL VENTILATION W/O EXTENSIVE PROCEDUR': 193,
'CARDIAC CATHETERIZATION W CIRC DISORD EXC ISCHEMIC HEART DISEASE': 193,
'INDEPENDENCE MEDICAL CENTER': 187,
'MALNUTRITION, FAILURE TO THRIVE & OTHER NUTRITIONAL DISORDERS': 98,
'EXTENSIVE PROCEDURE UNRELATED TO PRINCIPAL DIAGNOSIS': 98,
'BRONCHOLITIS & RSV PNEUMONIA': 97,
'CONNECTIVE TISSUE DISORDERS': 97,
'MAJOR RESPIRATORY & CHEST PROCEDURES': 97,
'MODERATELY EXTENSIVE PROCEDURE UNRELATED TO PRINCIPAL DIAGNOSIS': 97,
'SEPTICEMIA & DISSEMINATED INFECTIONS': 97,
'TRACHEOSTOMY W LONG TERM MECHANICAL VENTILATION W EXTENSIVE PROCEDURE': 96,
'ELECTROLYTE DISORDERS EXCEPT HYPOVOLEMIA RELATED': 95}

In [70]: #notice DIAGNOSIS_SUBCAT_DESC and DIAGNOSIS_SUBCAT_CODE have same number of unique values
len(df['DRG_APR_DESC'].unique())

Out[70]: 24

In [71]: len(df['DRG_APR_CODE'].unique())

Out[71]: 24

In [72]: #notice DRG_APR_CODE is not stored as numeric due to 000000
#no need to fix since it is a duplicate of DRG_APR_DESC
df['DRG_APR_CODE'].value_counts().to_dict()

Out[72]: {'134': 93725,
'130': 34636,
'140': 9611,
'137': 2271,
'131': 2012,
'130': 961,
'130': 481,
'892': 389,
'894': 389,
'207': 291,
'121': 290,
'191': 193,
'089': 193,
'00000': 193,
'421': 98,
'132': 98,
'980': 98,
'951': 97,
'120': 97,
'138': 97,
'720': 97,
'348': 97,
'084': 96,
'425': 96}

In [73]: #check values and counts of column
df['PROCEDURE_SUBCAT_DESC'].value_counts().to_dict()

Out[73]: {'OTHER DIAGNOSTIC RADIOLO': 47926,
'OTHER NONOPERATIVE PROC': 17120,
'INCISION, EXCISION, AND': 13635,
'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 INTERVEN': 192,
'INCISION AND EXCISION OF': 146,
'REPLACEMENT AND REMOVAL': 98,
'OPERATIONS ON RECTUM REC': 97,
'OPERATIONS ON LYMPHATIC': 97,
'OTHER OPERATIONS ON TEE': 96,
'OPERATIONS ON ANUS': 96,
'OPERATIONS ON LIVER': 94}

In [74]: #notice PROCEDURE_SUBCAT_DESC and PROCEDURE_SUBCAT_CODE have almost same number of unique values
len(df['PROCEDURE_SUBCAT_CODE'].unique())

Out[74]: 27

In [75]: len(df['PROCEDURE_SUBCAT_CODE'].unique())

Out[75]: 28

In [76]: #notice ADMIT_DATE has too many unique values to check
len(df['ADMIT_DATE'].unique())

Out[76]: 317

In [77]: #check values and counts for column
df['Standard_Orders_Used'].value_counts().to_dict()

Out[77]: {'Y': 117653, 'N': 28953}

In [78]: #check values and counts for column
df['HOSPITAL'].value_counts().to_dict()

Out[78]: {'St. Anthony Medical Center': 69577,
'Mercy Hospital': 34840,
'Hilding-Long Memorial Hospital': 18109,
'Oxbow Regional Hospital': 5133,
'Independence Medical Center': 5787,
'Superior-Parkland Hospital': 5113,
'Valley City Regional Hospital': 2601,
'Delaware County Hospital': 1446}

In [79]: #check values and counts for column
df['STATECODE'].value_counts().to_dict()

Out[79]: {'MN': 122526, 'WI': 14246, 'IA': 7233, 'ND': 2601}

In [80]: #check values and counts for column
df['City'].value_counts().to_dict()

Out[80]: {'Minneapolis': 69577,
'Bloomington': 34840,
'Park Rapids': 18109,
'Eau Claire': 9133,
'Waterloo': 5787,
'Parkland': 5113,
'Valley City': 2601,
'Manchester': 1446}

In [81]: #check values and counts for column
df['County_name'].value_counts().to_dict()

Out[81]: {'Hennepin': 104417,
'Hubbard': 18109,
'Eau Claire': 9133,
'Black Hawk': 5787,
'Douglas': 5113,
'Barnes': 2601,
'Delaware': 1446}

In [82]: #drop columns with data leakage into the future
#drop redundant and useless columns
df.drop(['Diagnosis_Group', 'DRG_APR_CODE', 'DRG_APR_DESC', 'DIAGNOSIS_SUBCAT_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 [83]: #null value check
df.isnull().sum()/len(df)*100

Out[83]:
gender              0.000000
race_cd             0.000000
PatientAge          0.000000
DRG_APR_SEVERITY    0.131645
PROCEDURE_SUBCAT_DESC 31.875238
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 [84]: #drop null values for DRG_APR_SEVERITY
df = df[df['DRG_APR_SEVERITY'].notna()]

In [85]: #drop null values for Num_Chronic_Cond
df = df[df['Num_Chronic_Cond'].notna()]

In [86]: #impute null values of PROCEDURE_SUBCAT_DESC with No Procedure
df['PROCEDURE_SUBCAT_DESC'].fillna('No Procedure', inplace = True)

In [87]: #null value check
df.isnull().sum()/len(df)*100

Out[87]:
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 [88]: #get dummy variables and drop first column
gender = pd.get_dummies(df['gender'], drop_first=True)

In [89]: #get dummy variables and drop first column
race_cd = pd.get_dummies(df['race_cd'], drop_first=True)

In [90]: #get dummy variables and drop first column
PROCEDURE_SUBCAT_DESC = pd.get_dummies(df['PROCEDURE_SUBCAT_DESC'], drop_first=True)

In [91]: #get dummy variables and drop first column
Standard_Orders_Used = pd.get_dummies(df['Standard_Orders_Used'], drop_first=True)

In [92]: #get dummy variables and drop first column
admit_month = pd.get_dummies(df['admit_month'], drop_first=True)

In [93]: #get dummy variables and drop first column
HOSPITAL = pd.get_dummies(df['HOSPITAL'], drop_first=True)

In [94]: #concatenate dummies with data frame
df = pd.concat([df, gender, race_cd, PROCEDURE_SUBCAT_DESC, Standard_Orders_Used, admit_month, HOSPITAL], axis=1)

In [95]: #drop original dummy variables from data frame
df.drop(['gender', 'race_cd', 'PROCEDURE_SUBCAT_DESC', 'Standard_Orders_Used', 'admit_month', 'HOSPITAL'], inplace = True, axis = 1)

In [96]: #export as csv
df.to_csv(r"C:\Users\Sierra\Documents\Preprocessed_LOS_READM.csv")

In [ ]: #Project paused
```