DSC 630-T302
5.2 Course Project: Milestone 3 -- Preliminary Analysis
Project Group 1

Abstract

In the healthcare industry, emergency room (ER) visits represent one of the highest cost medical services. Every year, many patients have to file bankruptcy mainly due to increasing hospital and medical bills mostly made up of ER visits which lead to hospitalizations. Although most ER visits are warranted and have saved countless lives, a considerable amount of ER visits are avoidable and could be addressed with a visit to a primary physician or even an urgent care facility, most of which have lesser cost involved. It is likely individuals who utilize the emergency room more than 4 times per year could be doing so unnecessarily. Several healthcare focused entities from hospitals to insurance providers have had to implement measures to help prevent unnecessary emergency room visits. Not only do these unnecessary visits cost patients thousands of dollars out of pocket, but a significant portion is paid by healthcare insurance providers. These companies are now focusing on developing clinical outreach programs to get to patients in time in order to avoid these unnecessary costs.

This paper focuses on the application of a machine learning model with the goal to predict patients who are likely to over-utilize the emergency room or have more than 4 ER visits in a year. It could be a step toward reaching these patients ahead of time to help them avoid these unnecessary costs. This application leverages historical data on patients, from demographic to clinical history that an algorithm can learn from and accurately predict those members at high risk to have several needless ER visits. The

resulting predictive model would then allow care managers to prioritize their outreach to members that are more likely to benefit from their programs and target the right patients. The specific application of this model is to flag anyone with a probability score above 50% as being at a high risk and direct focus at them specifically. Nurse care managers will then engage with patients that would have a high probability score from this model to help educate and provide care to them even in their home if needed. This approach prevents or delays disease progression that could lead to a visit to the ER and reduces cost since nurse home visits are less costly.

Background

The healthcare industry continually reviews efficiencies and balances stakeholder value with appropriate and effective care to patients. Readmissions are used to evaluate the quality of healthcare services provided by institutions (1). A readmission is defined as, “an episode when a patient who had been discharged from a hospital is admitted again with a specified time interval (2)." Additional requirements imposed by Medicare mandate hospitals to implement a Hospital Readmissions Reduction Program (HRRP)(3). An HRRP program focuses on improving communication, coordination, and ultimately the healthcare received. Through this Federal requirement, hospitals are evaluated relative to other institutions' readmission rates. Additional requirements are established by the Emergency Medical Treatment & Labor Act (EMTALA) which ensures public access to emergency services regardless of ability to pay (4). Emergency Rooms play an integral role as an immediate response service for healthcare emergencies as they account for half of all hospital admissions (5). Understanding the relationship between multiple ER visits as a potential to reduce readmission rates for longer term care should be evaluated.

Problem Statement

The problem to be evaluated will focus specifically on Emergency Room (ER) utilization data to help build a predictive model to understand the likelihood of a patient returning to the ER more than 4 times per year.

Scope

The data, model development, and deployment of the results of this project will focus explicitly on repeated ERs visits. The dataset leveraged was obtained from a leading government sponsored health care provider in the United States of America and contains demographics, various metrics, and associated categorical information of healthcare patients who visited the ER over the course of a 12 month period. Consideration of the time period, demographic information, specific to the geographical location, and primary activities of ER visits define the boundaries and application of the model, interpretation of the results, and subsequent deployment. Limitations of the data and model prohibit the use for predicting the likelihood of more than 4 ER visits per year outside of the USA or for other healthcare services. Additionally, pandemic conditions must be considered as there has been a material decrease in ER visits during the COVID-19 pandemic, primarily due to potential patients avoiding the risk of exposure to the virus (6).

Literature Review

Documentation and literature review performed in preparation for this project centered on multiple resource constraints healthcare providers' face (7). Unique to the healthcare industry is patient well-being and ethical duty to provide medical services. However, healthcare providers face challenges similar to other industries such as balancing economic efficiencies with stakeholder value.

As stated previously, over half of all hospital admissions are now entering the healthcare system from ER visits (5). As a de facto front door to a hospital, emergency department activities and readmissions have been reviewed extensively. It has also been shown the ER accounts as the primary source of admissions for elderly patients as well (8). The problem statement and objective of the review is supported by continued research on analyzing and reducing readmissions to the emergency room (9, 10).

Methods

Technical Approach

A machine learning model with the goal to predict patients who are likely to over-utilize the emergency room or have more than 4 ER visits in a year could be a step toward reaching these patients ahead of time to help them avoid unnecessary costs. This could be translated into a machine learning classification solution where algorithms such as a logistic regression, a gradient boosted decision tree, and others can be fit to the data to help determine the model with the best performance.

The programming languages Python (see Appendix 1) and R (see Appendix 2) were chosen for this project due to their ease of use, modeling capability, and visualizations. The JupyterNotebook and RStudio IDEs were chosen because of the open source nature of the software and supportive community of specialists.

Data Overview

To help build the model, we've acquired healthcare data from the leading government sponsored healthcare provider in the U.S. The available attributes include medical and pharmacy claims as well as demographic variables such as gender, age, and location. Overall, the dataset includes 69K records on patients over the previous 12 months, containing 46 features, with "MORE_THAN_4_ER_VISITS" identified as the target.

Handling Null/Missing Values

The dataset contains 20 Numerical features which contain either NaN or missing values. Additionally, there are 15 categorical variables in the dataset. We used LabelEncoder to transform categorical features into numeric values. After review, there are 11 features with an average 65 null values along with "Member_Months_Pre" with 2 and "ORCA_SCORE" being highest with 3400 null values in it. We have decided to replace null values with their median values instead of deleting the records completely.

Data Exploration

Data exploration started with looking into population and distribution of target feature i.e. “More_Than_4_Er_Visits”. Of the total 69K observations, 32K records indicated patients who had more than 4 ER visits versus 37K with less than 4 ER visits.

Outlier detection

The calculation of a Z score, or how many standard deviations a number is away from the mean, was used to detect outliers in the dataset. As a standard practice our threshold value was “3” standard deviations. Any record with 3+ Z score was marked as an outlier and replaced with the median value of that feature.

Feature selection

We considered “correlation” to identify most suitable features for modeling. We took 37 top correlated features into consideration with scores starting from -0.27 to 0.56.

Model Preparation

Model preparation was done with “More than 4 Er” being a target variable and the remaining 36 being dependent features. The entire dataset had previously been converted in numeric format during preprocessing using Label Encoder and was ready for modeling.

Logistic Regression

The problem statement is focused on predicting whether a patient will either have 4 or more visits to an ER or not. As this is a binary outcome, a Logistic Regression algorithm was chosen for modeling

Revisiting Model

After running the defined model with 100% population, a summary of the model output was used to further fine tune on the basis of p-Value score. Two features: Country_Clean & Reg_Region_Desc were removed from feature list due to significantly higher p_value score.

Results

After fine tuning the model, the following results were found.

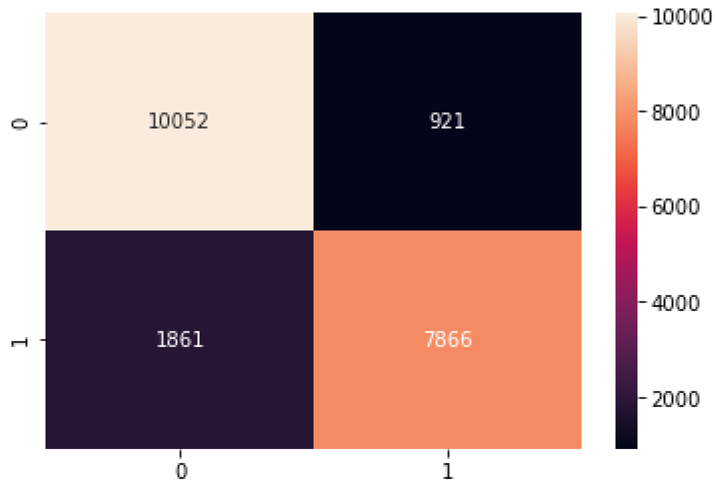
Accuracy

In order to ensure that the Logistic Regression model performs well on new data, a portion of the initial dataset of 30% was set aside to serve as the testing sample. The remaining 70% of the dataset was used for training purposes. All iterations of the Logistic Regression based on the attributes and methods documented above showed 87% accuracy

```
Accuracy of logistic regression classifier on test set: 0.87
```

Confusion Matrix

Confusion matrix and heatmap visualization were generated to indicate efficiency of the model with the number of false positives, false negatives, and true negatives.

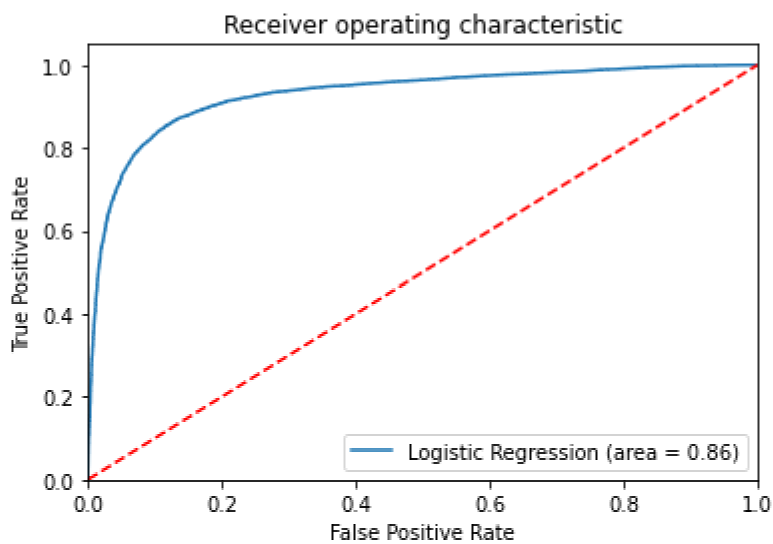


Classification Report and ROC Curve

To support performance evaluation done in previous steps, a classification report was utilized for further evaluation of forward key performance indicators: precision, recall and f1-score. All metrics indicated strong values (near 1), see below.

	precision	recall	f1-score	support
0	0.84	0.92	0.88	10973
1	0.90	0.81	0.85	9727
accuracy			0.87	20700
macro avg	0.87	0.86	0.86	20700
weighted avg	0.87	0.87	0.86	20700

Additionally, an ROC Curve was plotted as a part of visual performance indicator. The spatial distance from the Logistic Regression indicates a strong metric for performance. See below.



Discussion and Conclusion

Overall, the model developed showed favorable accuracy in the testing and training processes with the dataset available. Other metrics, such as precision, recall and f1 scores, also produced optimistic results towards the capability and potential applicability of the mode. Based on these results, this model could be used to predict the probability patients visiting the ER could return more than 4 times in a year and then potentially become readmitted to the hospital system.

When deployed, healthcare practitioners may input the same information and determine what level of care and remediation steps should be applied on a situational basis to reduce repeat visits and consequently limit impacts to the healthcare system. The only constraints would be on healthcare practitioners ability to collect the information used to create the model in addition to the scope limitations noted above.

Furthermore, after the model is deployed, ongoing monitoring should be put in place to ensure that the level of performance seen at training continues to hold true. This could require tracking actual outcome (or lack thereof) for a certain period of time and then compare these to the predictions made at the time. This will allow the project team to decide when it’s time to revisit the model and potentially re-train it if performance starts to degrade.

Acknowledgments
TBD

References

1. Brennan, J. J., Chan, T. C., Killeen, J. P., & Castillo, E. M. (2015). Inpatient Readmissions and Emergency Department Visits within 30 Days of a Hospital Admission. The western journal of emergency medicine, 16(7), 1025–1029. https://doi.org/10.5811/westjem.2015.8.26157

2. Wikimedia Foundation. (2021, June 8). Hospital readmission. Wikipedia. Retrieved September 10, 2021, from https://en.wikipedia.org/wiki/Hospital_readmission.

3. Hospital readmissions reduction Program (HRRP). CMS. (n.d.). Retrieved September 10, 2021, from https://www.cms.gov/Medicare/Medicare-Fee-for-Service-Payment/AcuteInpatientPPS/Readmissions-Reduction-Program.

4. Emergency medical Treatment & Labor act (EMTALA). CMS. (n.d.). Retrieved September 10, 2021, from https://www.cms.gov/Regulations-and-Guidance/Legislation/EMTALA.

5. Morganti, K. G., Bauhoff, S., Blanchard, J. C., Abir, M., Iyer, N., Smith, A., Vesely, J. V., Okeke, E. N., & Kellermann, A. L. (2013). The Evolving Role of Emergency Departments in the United States. Rand health quarterly, 3(2), 3.

6. Hartnett, K. P., Kite-Powell, A., DeVies, J., Coletta, M. A., Boehmer, T. K., Adjemian, J., Gundlapalli, A. V., & National Syndromic Surveillance Program Community of Practice (2020). Impact of the COVID-19 Pandemic on Emergency Department Visits - United States, January 1, 2019-May 30, 2020. MMWR. Morbidity and mortality weekly report, 69(23), 699–704. https://doi.org/10.15585/mmwr.mm6923e1

7. van Baal, P., Morton, A., & Severens, J. L. (2018). Health care input constraints and cost effectiveness analysis decision rules. Social science & medicine (2018), 200, 59–64. https://doi.org/10.1016/j.socscimed.2018.01.026

8. Greenwald, P. W., Estevez, R. M., Clark, S., Stern, M. E., Rosen, T., & Flomenbaum, N. (2016). The ed as the primary source of hospital admission for older (but Not YOUNGER) adults. The American Journal of Emergency Medicine, 34(6), 943–947. https://doi.org/10.1016/j.ajem.2015.05.041

9. Tsai, M. H., Xirasagar, S., Carroll, S., Bryan, C. S., Gallagher, P. J., Davis, K., & Jauch, E. C. (2018). Reducing High-Users' Visits to the Emergency Department by a Primary Care Intervention for the Uninsured: A Retrospective Study. Inquiry : a journal of medical care organization, provision and financing, 55, 46958018763917. https://doi.org/10.1177/0046958018763917

10. Kacprzyk, A., Stefura, T., Chłopaś, K. et al. “Analysis of readmissions to the emergency department among patients presenting with abdominal pain”. BMC Emerg Med 20, 37 (2020). https://doi.org/10.1186/s12873-020-00334-x

Appendix 1 - Python

Week 5

Name : Ayachit Madhukar

Course : DSC630

Instructor : Fadi Alsaleem

Date : 25 Sep 2021

Import

In [185…

```
# Importing required libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

import pandas_profiling as pp
```

In [73]:

```
import sys
# installing pandas-profiing
#!{sys.executable} -m pip install pandas-profiling
```

Data

In [74]:

```
# Load Source Data
datafile='Data/er_data.txt'
df = pd.read_csv(datafile,sep="|")
df.head()
```

Out[74]:

	AGE	SEX	RACE_ETHNICITY	PLAN_TYPE	STATE_CODE	PLAN_REGION	COMPLEXCARE_IND	MMP_
0	38.0	F	White	MARKETPLACE	FL	SOUTHEAST	0	
1	81.0	M	White	MEDICAID	NY	NORTHEAST	1	
2	30.0	F	White	MARKETPLACE	TX	SOUTHWEST	0	
3	88.0	F	White	MEDICARE	TX	SOUTHWEST	0	
4	1.0	F	Hispanic	MEDICAID	NE	MIDDLESTATES	0	

5 rows × 46 columns

In [75]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 69000 entries, 0 to 68999
Data columns (total 46 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   AGE                      69000 non-null  float64
 1   SEX                      69000 non-null  object 
 2   RACE_ETHNICITY           69000 non-null  object 
 3   PLAN_TYPE                69000 non-null  object 
 4   STATE_CODE               69000 non-null  object 
 5   PLAN_REGION              69000 non-null  object 
 6   COMPLEXCARE_IND          69000 non-null  int64  
 7   MMP_DUAL_IND             69000 non-null  int64  
 8   DUAL_PRODUCT_IND         69000 non-null  int64  
 9   LTC_IND                  69000 non-null  int64  
 10  MEDICAID_ELIGIBLE        69000 non-null  int64  
 11  MEDICARE_ELIGIBLE        69000 non-null  int64  
 12  BEHAVIORAL_ELIGIBLE      69000 non-null  int64  
 13  COMMERCIAL_ELIGIBLE      69000 non-null  int64  
 14  OTHER_ELIGIBLE           69000 non-null  int64  
 15  RISK_TYPE_DESC           6891 non-null   object 
 16  MEMBER_MONTHS_PRE        68998 non-null  float64
 17  ADD_STATE                68113 non-null  object 
 18  COUNTY_CLEAN             50817 non-null  object 
 19  REG_REGION_DESC          69000 non-null  object 
 20  RISK_SCORE               68935 non-null  float64
 21  PRIOR_TOTAL_COSTS_ANNUAL 68935 non-null  float64
 22  PRIOR_RX_COSTS_ANNUAL    68935 non-null  float64
 23  ANNUAL_IP_COSTS          68935 non-null  float64
 24  ANNUAL_ER_COSTS          68935 non-null  float64
 25  ANNUAL_OTHER_COSTS       68935 non-null  float64
 26  FUTURE_RISK_INPATIENT    68935 non-null  float64
 27  BH_RISK_SCORE            68935 non-null  float64
 28  RX_RISK_SCORE            68935 non-null  float64
 29  ER_RISK_SCORE            68935 non-null  float64
 30  ORCA_SCORE               65600 non-null  float64
 31  ORCA_RISK_GROUP          65600 non-null  object 
 32  SUD_SEG_VALUE            68935 non-null  float64
 33  SUD_SEG_DEF              68935 non-null  object 
 34  ENG_SCORE                68935 non-null  float64
 35  POPHEALTHCAT_GROUPED     69000 non-null  object 
 36  INTERVENABLE_IND         69000 non-null  int64  
 37  SHORT_DESC               68935 non-null  object 
 38  SHORT_DESC_2             69000 non-null  object 
 39  RISK_CAT_RECODE          68935 non-null  object 
 40  MEDICAID_CLAIMS          69000 non-null  int64  
 41  MEDICARE_CLAIMS          69000 non-null  int64  
 42  BEHAVIORAL_CLAIMS        69000 non-null  int64  
 43  COMMERCIAL_CLAIMS        69000 non-null  int64  
 44  OTHER_CLAIMS             69000 non-null  int64  
 45  MORE_THAN_4_ER_VISITS    69000 non-null  int64  
dtypes: float64(15), int64(16), object(15)
memory usage: 24.2+ MB
```

VARIABLE DEFINITION

AGE
The age of the patient at the time the data was gathered

SEX
The Gender of the patient (Male or Female)

RACE_ETHNICITY
The race or ethnicity of the patient

PLAN_TYPE
The type of plan or benefit the patient is on such as medicaid, medicare, marketplace (ObamaCare) or Commerical Insurance

STATE_CODE
The State in which the patient gets benefits from

PLAN_REGION
The region of the U.S the patient lives in: Midwest, Southwest....

COMPLEXCARE_IND
Specify whether the patient is deemed to require complex care services

MMP_DUAL_IND
Specify whether the patient has both medicare and medicaid coverage

DUAL_PRODUCT_IND
Specify whether the patient has more than one public benefit, such social security, TANF, food stamps...

LTC_IND
Specify whether the patient has long term care needs

MEDICAID_ELIGIBLE Specify whether the patient is eligible for medicaid

MEDICARE_ELIGIBLE
specify whether the patient is eligible for medicare

BEHAVIORAL_ELIGIBLE
specify whether the patient is eligibile for behavioral health services

COMMERCIAL_ELIGIBLE
specify whether the patient is eligible for health coverage through an employer

OTHER_ELIGIBLE
Specify whether the patient has some other type of medical coverages

RISK_TYPE_DESC
he type of risk that the patient represent to their health plan, specify whether the insurer takes on the full risk, or share the risk

MEMBER_MONTHS_PRE
The total number of months the member has coverage during the previous 12 months

ADD_STATE
The state in which the patient lives

COUNTY_CLEAN
The county in which the patient lives if available

REG_REGION_DESC
The regio in which the patient lives

RISK_SCORE
The overall health risk score attributed to the patient. The higher the score the worse the patient

PRIOR_TOTAL_COSTS_ANNUAL
The total medical or healthcare cost incurred by the patients during the prior year

PRIOR_RX_COSTS_ANNUAL
The total Pharmacy or drugs cost incurred by the patients during the prior year

ANNUAL_IP_COSTS
The total inpatient or hospitalization cost incurred by the patients during the prior year

ANNUAL_ER_COSTS
The total emergency room (ER) cost incurred by the patients during the prior year

ANNUAL_OTHER_COSTS
All other medical services cost incurred by the patients during the prior year

FUTURE_RISK_INPATIENT
A score that's designed to be predictive of the future risk of hospitalization of the patient

BH_RISK_SCORE A score that's designed to be predictive of the future risk of behavioral health needs of the patient

RX_RISK_SCORE A score that's designed to be predictive of the future medication needs of the patient

ER_RISK_SCORE A score that's designed to be predictive of the future emergency care needs of the patient

ORCA_SCORE Opioid risk classification algorithm/ The likelihood of the patient abusing opioid

ORCA_RISK_GROUP A grouping of the patient based on the ORCA score

SUD_SEG_VALUE The substance use disorder segment that the member belongs to

SUD_SEG_DEF A definition of the SUD_SEG_VALUE

ENG_SCORE The likelihood of the member successfully completing a care management program

POPHEALTHCAT_GROUPED

The population health category that the patient belongs to based on their medical history

INTERVENABLE_IND Specify whether the patient is likely to benefit from an intervention

SHORT_DESC Description of the condition(s) that the patient might be suffering from

SHORT_DESC_2 Description of the condition(s) that the patient might be suffering from

RISK_CAT_RECODE A grouping of the type of healthcare needs the patient requires

MEDICAID_CLAIMS The total number of healthcare or medical claims that the patients incurred using medicaid

MEDICARE_CLAIMS The total number of healthcare or medical claims that the patients incurred using medicare

BEHAVIORAL_CLAIMS The total number of healthcare or medical claims that the patients incurred using behavioral health coverage

COMMERCIAL_CLAIMS The total number of healthcare or medical claims that the patients incurred using commercial or employer coverage

OTHER_CLAIMS The total number of all other healthcare or medical claims that the patients incurred

***MORE_THAN_4_ER_VISITS** Specify whether or not the patient has had 4 or more ER visits previously **(This is the target to predict)**.

In [77]:

```
df.shape
```

Out[77]:

```
(69000, 46)
```

Identifying and Handling Non Numerical data

In [76]:

```
df.describe(include="O").columns
```

Out[76]:

```
Index(['SEX', 'RACE_ETHNICITY', 'PLAN_TYPE', 'STATE_CODE', 'PLAN_REGION',
       'RISK_TYPE_DESC', 'ADD_STATE', 'COUNTY_CLEAN', 'REG_REGION_DESC',
       'ORCA_RISK_GROUP', 'SUD_SEG_DEF', 'POPHEALTHCAT_GROUPED', 'SHORT_DESC',
       'SHORT_DESC_2', 'RISK_CAT_RECODE'],
      dtype='object')
```

In [179…

```
object_columns=['SEX', 'RACE_ETHNICITY', 'PLAN_TYPE', 'STATE_CODE', 'PLAN_REGION',
         'RISK_TYPE_DESC', 'ADD_STATE', 'COUNTY_CLEAN', 'REG_REGION_DESC',
         'ORCA_RISK_GROUP', 'SUD_SEG_DEF', 'POPHEALTHCAT_GROUPED', 'SHORT_DESC',
         'SHORT_DESC_2', 'RISK_CAT_RECODE']
```

In [79]:

```
###  Handling Non Numerical data using Label Encoder

from sklearn import preprocessing
labelencoder = preprocessing.LabelEncoder()
cleaned_df=df
for c in object_columns:
    cleaned_df[c]=labelencoder.fit_transform(cleaned_df[c])

cleaned_df
```

Out[79]:

	AGE	SEX	RACE_ETHNICITY	PLAN_TYPE	STATE_CODE	PLAN_REGION	COMPLEXCARE_IND	MM
0	38.0	0	6	4	6	3	0	
1	81.0	1	6	5	26	1	1	
2	30.0	0	6	4	33	4	0	
3	88.0	0	6	6	33	4	0	
4	1.0	0	3	5	21	0	0	
...	...	...	...	...	...	...	...	
68995	0.0	1	6	5	13	3	1	
68996	0.0	1	5	5	9	0	0	
68997	0.0	1	5	5	10	0	1	
68998	0.0	1	6	5	27	0	0	
68999	0.0	1	5	5	10	0	0	

69000 rows × 46 columns

In [80]:

```
cleaned_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 69000 entries, 0 to 68999
Data columns (total 46 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   AGE                       69000 non-null  float64
 1   SEX                       69000 non-null  int64
 2   RACE_ETHNICITY            69000 non-null  int64
 3   PLAN_TYPE                 69000 non-null  int64
 4   STATE_CODE                69000 non-null  int64
 5   PLAN_REGION               69000 non-null  int64
 6   COMPLEXCARE_IND           69000 non-null  int64
 7   MMP_DUAL_IND              69000 non-null  int64
 8   DUAL_PRODUCT_IND          69000 non-null  int64
 9   LTC_IND                   69000 non-null  int64
 10  MEDICAID_ELIGIBLE         69000 non-null  int64
 11  MEDICARE_ELIGIBLE         69000 non-null  int64
 12  BEHAVIORAL_ELIGIBLE       69000 non-null  int64
 13  COMMERCIAL_ELIGIBLE       69000 non-null  int64
 14  OTHER_ELIGIBLE            69000 non-null  int64
 15  RISK_TYPE_DESC            69000 non-null  int64
 16  MEMBER_MONTHS_PRE         68998 non-null  float64
 17  ADD_STATE                 69000 non-null  int64
```

```
 18  COUNTY_CLEAN             69000 non-null  int64
 19  REG_REGION_DESC          69000 non-null  int64
 20  RISK_SCORE               68935 non-null  float64
 21  PRIOR_TOTAL_COSTS_ANNUAL 68935 non-null  float64
 22  PRIOR_RX_COSTS_ANNUAL    68935 non-null  float64
 23  ANNUAL_IP_COSTS          68935 non-null  float64
 24  ANNUAL_ER_COSTS          68935 non-null  float64
 25  ANNUAL_OTHER_COSTS       68935 non-null  float64
 26  FUTURE_RISK_INPATIENT    68935 non-null  float64
 27  BH_RISK_SCORE            68935 non-null  float64
 28  RX_RISK_SCORE            68935 non-null  float64
 29  ER_RISK_SCORE            68935 non-null  float64
 30  ORCA_SCORE               65600 non-null  float64
 31  ORCA_RISK_GROUP          69000 non-null  int64
 32  SUD_SEG_VALUE            68935 non-null  float64
 33  SUD_SEG_DEF              69000 non-null  int64
 34  ENG_SCORE                68935 non-null  float64
 35  POPHEALTHCAT_GROUPED     69000 non-null  int64
 36  INTERVENABLE_IND         69000 non-null  int64
 37  SHORT_DESC               69000 non-null  int64
 38  SHORT_DESC_2             69000 non-null  int64
 39  RISK_CAT_RECODE          69000 non-null  int64
 40  MEDICAID_CLAIMS          69000 non-null  int64
 41  MEDICARE_CLAIMS          69000 non-null  int64
 42  BEHAVIORAL_CLAIMS        69000 non-null  int64
 43  COMMERCIAL_CLAIMS        69000 non-null  int64
 44  OTHER_CLAIMS             69000 non-null  int64
 45  MORE_THAN_4_ER_VISITS    69000 non-null  int64
dtypes: float64(15), int64(31)
memory usage: 24.2 MB
```

Identifying Null values and replacing it with median

In [92]:

```
#looking for null values
s=cleaned_df.isnull().sum()
s=s[s!=0]
s
```

Out[92]:

```
MEMBER_MONTHS_PRE              2
RISK_SCORE                    65
PRIOR_TOTAL_COSTS_ANNUAL      65
PRIOR_RX_COSTS_ANNUAL         65
ANNUAL_IP_COSTS               65
ANNUAL_ER_COSTS               65
ANNUAL_OTHER_COSTS            65
FUTURE_RISK_INPATIENT         65
BH_RISK_SCORE                 65
RX_RISK_SCORE                 65
ER_RISK_SCORE                 65
ORCA_SCORE                  3400
SUD_SEG_VALUE                 65
ENG_SCORE                     65
dtype: int64
```

In [180…

```
# replacing null with median value
Null_columns=['MEMBER_MONTHS_PRE','RISK_SCORE','PRIOR_TOTAL_COSTS_ANNUAL','PRIOR_RX_COS
for c in Null_columns:
    median = cleaned_df[c].median()
    cleaned_df[c].fillna(median, inplace=True)

cleaned_df.isnull().sum()
```

Out[180…

```
AGE                          0
SEX                          0
RACE_ETHNICITY               0
PLAN_TYPE                    0
STATE_CODE                   0
PLAN_REGION                  0
COMPLEXCARE_IND              0
MMP_DUAL_IND                 0
DUAL_PRODUCT_IND             0
LTC_IND                      0
MEDICAID_ELIGIBLE            0
MEDICARE_ELIGIBLE            0
BEHAVIORAL_ELIGIBLE          0
COMMERCIAL_ELIGIBLE          0
OTHER_ELIGIBLE               0
RISK_TYPE_DESC               0
MEMBER_MONTHS_PRE            0
ADD_STATE                    0
COUNTY_CLEAN                 0
REG_REGION_DESC              0
RISK_SCORE                   0
PRIOR_TOTAL_COSTS_ANNUAL     0
PRIOR_RX_COSTS_ANNUAL        0
ANNUAL_IP_COSTS              0
ANNUAL_ER_COSTS              0
ANNUAL_OTHER_COSTS           0
FUTURE_RISK_INPATIENT        0
BH_RISK_SCORE                0
RX_RISK_SCORE                0
ER_RISK_SCORE                0
ORCA_SCORE                   0
ORCA_RISK_GROUP              0
SUD_SEG_VALUE                0
SUD_SEG_DEF                  0
ENG_SCORE                    0
POPHEALTHCAT_GROUPED         0
INTERVENABLE_IND             0
SHORT_DESC                   0
SHORT_DESC_2                 0
RISK_CAT_RECODE              0
MEDICAID_CLAIMS              0
MEDICARE_CLAIMS              0
BEHAVIORAL_CLAIMS            0
COMMERCIAL_CLAIMS            0
OTHER_CLAIMS                 0
MORE_THAN_4_ER_VISITS        0
dtype: int64
```

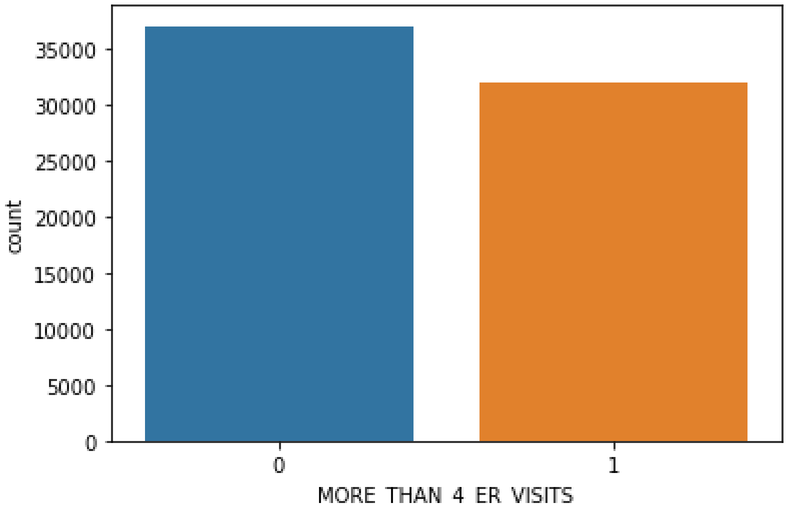
Exploration

In [246…

```
# exiting count breakup  of 4+ ER Visits

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

ax = sns.countplot(cleaned_df.MORE_THAN_4_ER_VISITS,label="Count")
print(y.value_counts())
```

```
0    37000
```

```
1    32000
Name: MORE_THAN_4_ER_VISITS, dtype: int64
/Users/madhukarayachit/opt/anaconda3/lib/python3.8/site-packages/seaborn/_decorators.py:
36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



Outlier Detection and cleaning

In [247…

```
# Outlier detection
import scipy.stats as stats
#Internally studentized method (z-score)
def z_score_method(df, variable_name):
    #Takes two parameters: dataframe & variable of interest as string
    columns = df.columns
    z = np.abs(stats.zscore(df))
    threshold = 3
    outlier = []
    index=0
    for item in range(len(columns)):
        if columns[item] == variable_name:
            index = item
    for i, v in enumerate(z[:, index]):
        if v > threshold:
            outlier.append(i)
        else:
            continue
    return outlier

outlier_z = z_score_method(cleaned_df, 'AGE')
for c in cleaned_df.columns:
    outlier_z = z_score_method(cleaned_df, c)

    if (len(outlier_z)>0):
        print (len(outlier_z) , ' outliers in ' , c)
        print(cleaned_df[c].iloc[outlier_z])

        # replacing outlier with median value
        median =cleaned_df[c].median()
        cleaned_df[c].iloc[outlier_z] = np.nan
        cleaned_df.fillna(median,inplace=True)
```

```
/Users/madhukarayachit/opt/anaconda3/lib/python3.8/site-packages/scipy/stats/stats.py:25
00: RuntimeWarning: invalid value encountered in true_divide
  return (a - mns) / sstd
31  outliers in  PLAN_TYPE
37       3.0
110      3.0
457      3.0
488      3.0
1222     3.0
1691     3.0
2056     3.0
2621     3.0
3366     3.0
5790     3.0
6354     3.0
6857     3.0
7351     3.0
7799     3.0
7822     3.0
9005     3.0
9278     3.0
10319    3.0
11494    3.0
12533    3.0
16542    3.0
17146    3.0
22581    3.0
25822    3.0
26634    3.0
26786    3.0
26927    3.0
31305    3.0
50998    3.0
51010    3.0
51063    3.0
Name: PLAN_TYPE, dtype: float64
/Users/madhukarayachit/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.p
y:670: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  iloc._setitem_with_indexer(indexer, value)
3514  outliers in  RISK_TYPE_DESC
17       2.0
25       3.0
30       2.0
55       3.0
61       3.0
        ...
68836    3.0
68864    3.0
68879    2.0
68884    3.0
68922    2.0
Name: RISK_TYPE_DESC, Length: 3514, dtype: float64
1753  outliers in  RISK_SCORE
24       15.4592
195      15.4639
326      18.4911
356      16.5013
458      15.5998
          ...
68788    20.9538
```

```
68877    21.2681
68951    15.8066
68975    14.6363
68985    15.0308
Name: RISK_SCORE, Length: 1753, dtype: float64
2077  outliers in  PRIOR_TOTAL_COSTS_ANNUAL
13        84033.83
24        73343.52
183      137381.44
200      100995.35
276       83579.71
           ...
68771     73661.84
68788     71369.40
68902    117157.42
68926     70520.05
68998    121874.55
Name: PRIOR_TOTAL_COSTS_ANNUAL, Length: 2077, dtype: float64
1655  outliers in  PRIOR_RX_COSTS_ANNUAL
127      56771.11
162      23609.76
195      54173.52
200      19538.45
260      26276.60
           ...
68109    29542.89
68113    19338.72
68238    34305.35
68711    42646.47
68877    19945.49
Name: PRIOR_RX_COSTS_ANNUAL, Length: 1655, dtype: float64
1843  outliers in  ANNUAL_IP_COSTS
25       22955.83
28       61275.36
207      36366.40
244      59605.56
329      51881.23
           ...
68926    23370.55
68975    57147.82
68981    22823.47
68982    26982.38
68998    36512.64
Name: ANNUAL_IP_COSTS, Length: 1843, dtype: float64
1777  outliers in  ANNUAL_ER_COSTS
819      6715.20
1123     6397.73
1300     4627.05
1547     4837.70
2146     5797.05
          ...
68695    4958.63
68788    6101.40
68822    4430.98
68823    4596.71
68914    4273.62
Name: ANNUAL_ER_COSTS, Length: 1777, dtype: float64
1995  outliers in  ANNUAL_OTHER_COSTS
323      40842.62
413      62465.00
519      50696.49
584      40204.66
620      43321.34
           ...
68743    58468.65
```

```
68788    51178.95
68877    45421.84
68926    46607.97
68934    48342.40
Name: ANNUAL_OTHER_COSTS, Length: 1995, dtype: float64
2569  outliers in  FUTURE_RISK_INPATIENT
50       17.6432
200      23.8625
224      19.7088
286      16.0005
374      20.9119
          ...
67929    18.7353
67957    16.4742
68113    20.7151
68474    22.6004
68794    22.8755
Name: FUTURE_RISK_INPATIENT, Length: 2569, dtype: float64
2280  outliers in  BH_RISK_SCORE
93       28.568
131      34.478
407      31.145
553      25.013
570      34.642
          ...
67840    31.442
67957    35.285
67973    27.400
68065    32.369
68872    33.550
Name: BH_RISK_SCORE, Length: 2280, dtype: float64
1749  outliers in  RX_RISK_SCORE
195      12.9797
234      11.0840
268      17.2066
286      13.4253
318      11.2428
          ...
68531    11.7058
68542    11.1915
68649    12.7739
68757    11.8065
68951    15.1222
Name: RX_RISK_SCORE, Length: 1749, dtype: float64
1679  outliers in  ER_RISK_SCORE
891      22.8315
1052     23.1255
1234     22.8801
1689     25.2189
2082     23.2170
          ...
67973    23.7476
68045    26.1962
68335    22.8101
68572    23.0872
68984    24.6212
Name: ER_RISK_SCORE, Length: 1679, dtype: float64
1921  outliers in  SUD_SEG_VALUE
13       2.0
78       2.0
2481     2.0
2546     2.0
2565     2.0
        ...
67581    2.0
```

```
67599    2.0
67700    2.0
67957    2.0
68579    2.0
Name: SUD_SEG_VALUE, Length: 1921, dtype: float64
4232  outliers in  SUD_SEG_DEF
66       2.0
135      2.0
161      2.0
217      2.0
247      2.0
        ...
68093    2.0
68105    2.0
68794    2.0
68872    2.0
68917    2.0
Name: SUD_SEG_DEF, Length: 4232, dtype: float64
```

In [102…

```
columns = np.full((cleaned_df.corr().shape[0],), True, dtype=bool)
for i in range(cleaned_df.corr().shape[0]):
    for j in range(i+1, cleaned_df.corr().shape[0]):
        if cleaned_df.corr().iloc[i,j] >= 0.9:
            if columns[j]:
                columns[j] = False
selected_columns = cleaned_df.columns[columns]
data = cleaned_df[selected_columns]
data
```

Out[102…

	AGE	SEX	RACE_ETHNICITY	PLAN_TYPE	STATE_CODE	PLAN_REGION	COMPLEXCARE_IND	MM
0	38.0	0	6	4.0	6	3	0	
1	81.0	1	6	5.0	26	1	1	
2	30.0	0	6	4.0	33	4	0	
3	88.0	0	6	6.0	33	4	0	
4	1.0	0	3	5.0	21	0	0	
...	...	...	...	...	...	...	...	
68995	0.0	1	6	5.0	13	3	1	
68996	0.0	1	5	5.0	9	0	0	
68997	0.0	1	5	5.0	10	0	1	
68998	0.0	1	6	5.0	27	0	0	
68999	0.0	1	5	5.0	10	0	0	

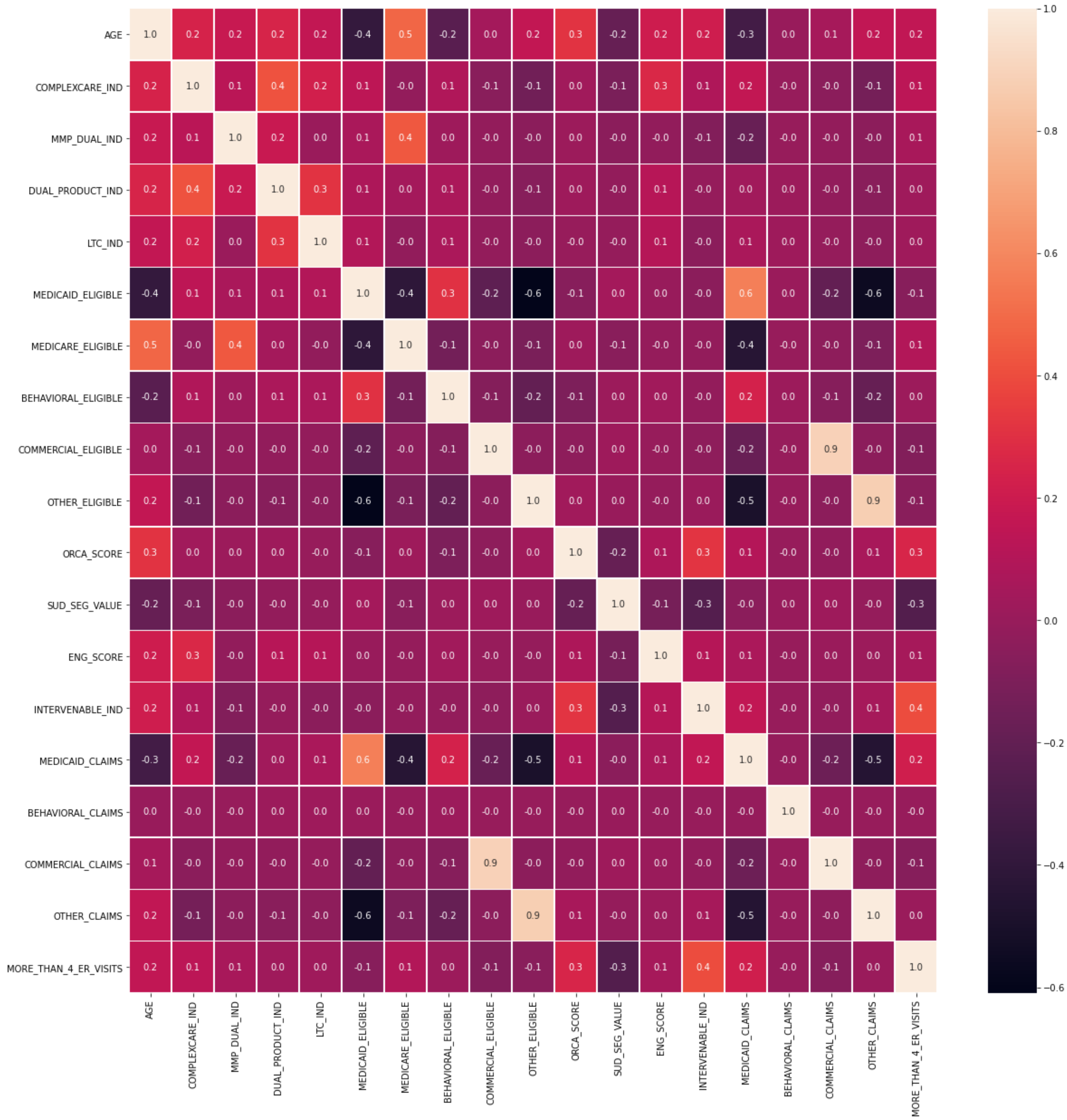
69000 rows × 46 columns

Feature selection using corelation

In [32]:

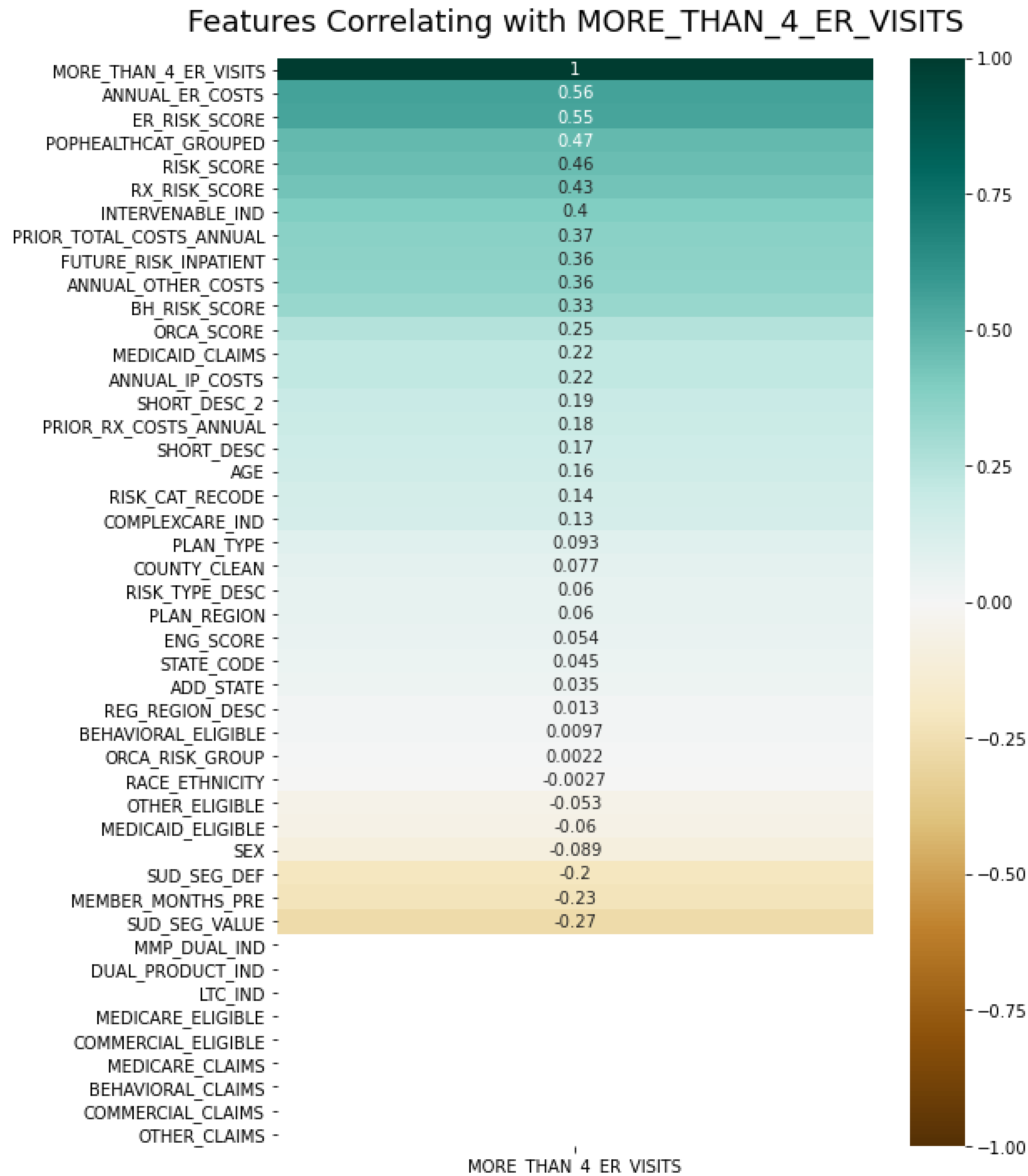
```
# Corelation map
f,ax = plt.subplots(figsize=(20, 20))
sns.heatmap(data.corr(), annot=True, linewidths=.5, fmt= '.1f',ax=ax)
```

Out[32]: <AxesSubplot:>



In [103…

```
# corelation with target variable
plt.figure(figsize=(8, 12))
heatmap = sns.heatmap(data.corr()[['MORE_THAN_4_ER_VISITS']].sort_values(by='MORE_THAN_
heatmap.set_title('Features Correlating with MORE_THAN_4_ER_VISITS', fontdict={'fontsiz
```



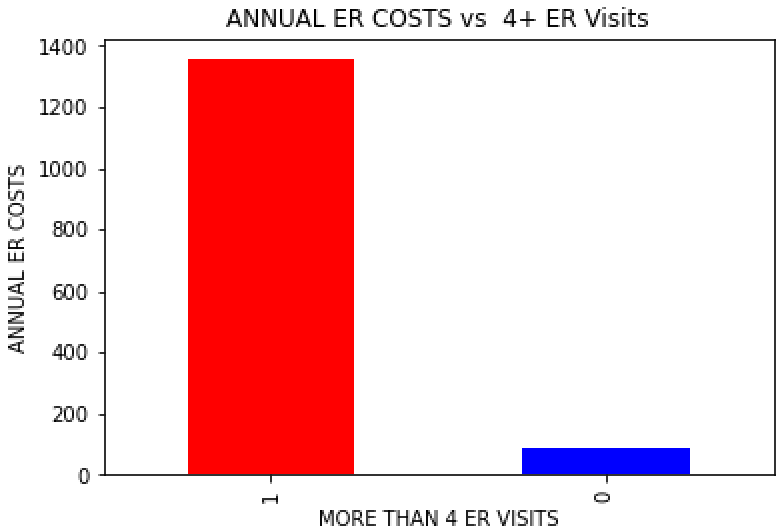
Bar graph for top 3 corelations

In [249…

```
erdata=data.groupby("MORE_THAN_4_ER_VISITS")['ANNUAL_ER_COSTS'].describe().sort_values(
erdata["mean"].plot(kind='bar',color=['red',  'blue', ])
plt.xlabel('MORE THAN 4 ER VISITS')
plt.ylabel("ANNUAL ER COSTS")
plt.title("ANNUAL ER COSTS vs  4+ ER Visits")
```

Out[249…

```
Text(0.5, 1.0, 'ANNUAL ER COSTS vs  4+ ER Visits')
```

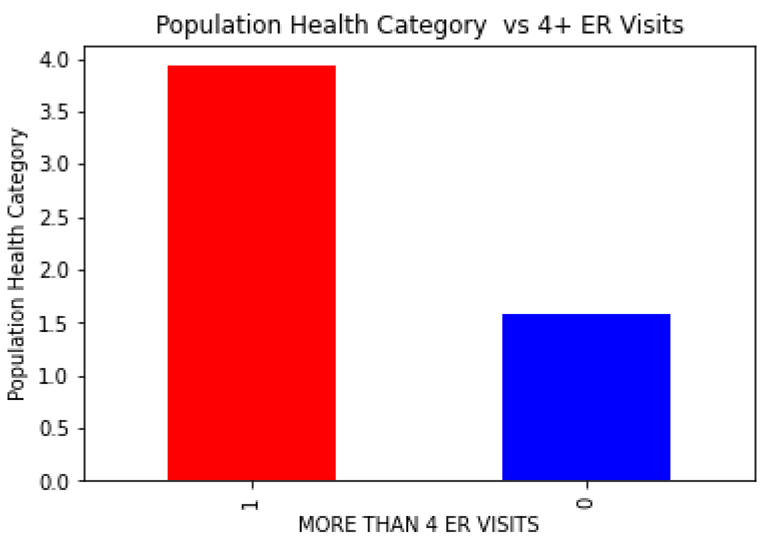


In [250…

```
erdata=data.groupby("MORE_THAN_4_ER_VISITS")['POPHEALTHCAT_GROUPED'].describe().sort_va
erdata["mean"].plot(kind='bar',color=['red',  'blue', ])
plt.xlabel('MORE THAN 4 ER VISITS')
plt.ylabel("Population Health Category")
plt.title("Population Health Category  vs 4+ ER Visits")
```

Out[250…

```
Text(0.5, 1.0, 'Population Health Category  vs 4+ ER Visits')
```

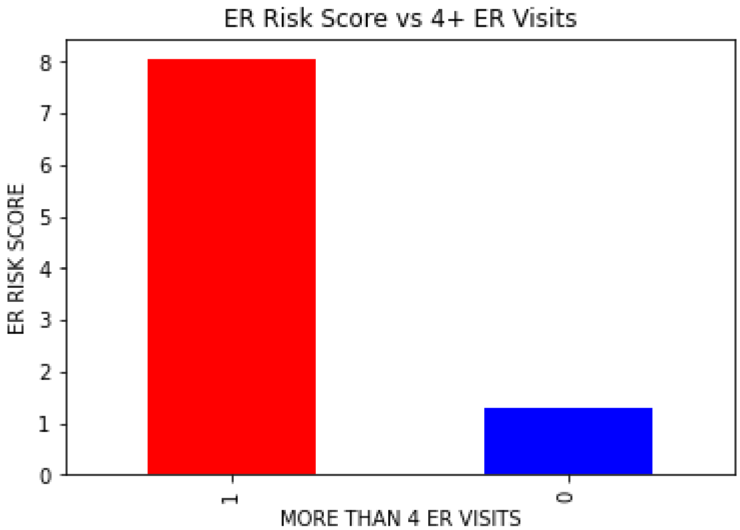


In [251…

```
erdata=data.groupby("MORE_THAN_4_ER_VISITS")['ER_RISK_SCORE'].describe().sort_values('m
erdata["mean"].plot(kind='bar',color=['red',  'blue', ])
plt.xlabel('MORE THAN 4 ER VISITS')
plt.ylabel("ER RISK SCORE")
plt.title("ER Risk Score vs 4+ ER Visits")
```

Out[251…

```
Text(0.5, 1.0, 'ER Risk Score vs 4+ ER Visits')
```



Preparing data for model

In [252…

```
# Preparing model data
selected_columns=['MORE_THAN_4_ER_VISITS',
                  'ANNUAL_ER_COSTS',
                  'ER_RISK_SCORE',
                  'POPHEALTHCAT_GROUPED',
                  'RISK_SCORE',
                  'RX_RISK_SCORE',
                  'INTERVENABLE_IND',
                  'PRIOR_TOTAL_COSTS_ANNUAL',
                  'FUTURE_RISK_INPATIENT',
                  'ANNUAL_OTHER_COSTS',
                  'BH_RISK_SCORE',
                  'ORCA_SCORE',
                  'MEDICAID_CLAIMS',
                  'ANNUAL_IP_COSTS',
                  'SHORT_DESC_2',
                  'PRIOR_RX_COSTS_ANNUAL',
                  'SHORT_DESC',
                  'AGE',
                  'RISK_CAT_RECODE',
                  'COMPLEXCARE_IND',
                  'PLAN_TYPE',
                  'COUNTY_CLEAN',
                  'RISK_TYPE_DESC',
                  'PLAN_REGION',
                  'ENG_SCORE',
                  'STATE_CODE',
                  'ADD_STATE',
                  'REG_REGION_DESC',
                  'BEHAVIORAL_ELIGIBLE',
                  'ORCA_RISK_GROUP',
                  'RACE_ETHNICITY',
                  'OTHER_ELIGIBLE',
                  'MEDICAID_ELIGIBLE',
                  'SEX',
                  'SUD_SEG_DEF',
                  'MEMBER_MONTHS_PRE',
                  'SUD_SEG_VALUE']
```

```
model_data=data[selected_columns]
model_data
```

Out[252...

	MORE_THAN_4_ER_VISITS	ANNUAL_ER_COSTS	ER_RISK_SCORE	POPHEALTHCAT_GROUPED	RISK_
0	0	0.00	1.9154	5	
1	0	0.00	8.3131	4	
2	0	0.00	0.6467	0	
3	0	0.00	2.0956	2	
4	0	0.00	0.9482	0	
...	...	...	...	...	
68995	1	777.07	7.5670	4	1
68996	1	2763.28	16.0033	1	
68997	1	941.15	2.6039	1	
68998	1	1079.95	4.9284	4	
68999	1	104.98	1.5763	1	

69000 rows × 37 columns

In [168...

```
import statsmodels.api as sm

y=model_data.MORE_THAN_4_ER_VISITS
X=model_data.drop("MORE_THAN_4_ER_VISITS",axis=1)


X.describe()
```

Out[168...

	ANNUAL_ER_COSTS	ER_RISK_SCORE	POPHEALTHCAT_GROUPED	RISK_SCORE	RX_RISK_SCORE	I
count	69000.000000	69000.000000	69000.000000	69000.000000	69000.000000	
mean	673.549401	4.417997	2.672145	2.507399	2.268842	
std	1133.082080	6.084783	2.472848	3.567680	2.916019	
min	0.000000	0.289600	0.000000	0.100000	0.134700	
25%	0.000000	0.667100	0.000000	0.354300	0.471700	
50%	121.940000	1.525200	2.000000	1.064300	0.974900	
75%	940.637500	5.366350	4.000000	3.206975	2.905525	
max	7676.760000	26.544900	10.000000	24.902200	17.221500	

8 rows × 36 columns

Modeling

In [169…

```
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary2())
```

```
Optimization terminated successfully.
         Current function value: 0.199834
         Iterations 9
                             Results: Logit
=========================================================================
Model:                   Logit                 Pseudo R-squared:   0.711
Dependent Variable:      MORE_THAN_4_ER_VISITS AIC:                27649.0715
Date:                    2021-09-25 21:54      BIC:                27978.1785
No. Observations:        69000                 Log-Likelihood:     -13789.
Df Model:                35                    LL-Null:            -47646.
Df Residuals:            68964                 LLR p-value:        0.0000
Converged:               1.0000                Scale:              1.0000
No. Iterations:          9.0000
-------------------------------------------------------------------------
                           Coef.  Std.Err.    z     P>|z|   [0.025  0.975]
-------------------------------------------------------------------------
ANNUAL_ER_COSTS            0.0031   0.0000  73.2131 0.0000  0.0030  0.0032
ER_RISK_SCORE              0.2968   0.0063  46.9788 0.0000  0.2844  0.3092
POPHEALTHCAT_GROUPED       0.1720   0.0094  18.3388 0.0000  0.1536  0.1904
RISK_SCORE                 0.1687   0.0085  19.7526 0.0000  0.1520  0.1854
RX_RISK_SCORE              0.0607   0.0092   6.6293 0.0000  0.0428  0.0787
INTERVENABLE_IND          -0.2281   0.0419  -5.4463 0.0000 -0.3103 -0.1460
PRIOR_TOTAL_COSTS_ANNUAL   0.0000   0.0000   2.1624 0.0306  0.0000  0.0000
FUTURE_RISK_INPATIENT      0.0232   0.0061   3.8301 0.0001  0.0113  0.0351
ANNUAL_OTHER_COSTS         0.0000   0.0000   6.6473 0.0000  0.0000  0.0000
BH_RISK_SCORE              0.0110   0.0033   3.3061 0.0009  0.0045  0.0175
ORCA_SCORE                 0.0060   0.0005  12.9559 0.0000  0.0051  0.0070
MEDICAID_CLAIMS            3.4167   0.0758  45.0908 0.0000  3.2682  3.5652
ANNUAL_IP_COSTS            0.0000   0.0000   4.9325 0.0000  0.0000  0.0000
SHORT_DESC_2              -0.0085   0.0011  -7.6187 0.0000 -0.0107 -0.0063
PRIOR_RX_COSTS_ANNUAL     -0.0000   0.0000 -10.9507 0.0000 -0.0000 -0.0000
SHORT_DESC                 0.0118   0.0010  11.6581 0.0000  0.0098  0.0138
AGE                       -0.0083   0.0011  -7.7053 0.0000 -0.0104 -0.0062
RISK_CAT_RECODE            0.0177   0.0019   9.2522 0.0000  0.0139  0.0214
COMPLEXCARE_IND            0.1299   0.0522   2.4884 0.0128  0.0276  0.2321
PLAN_TYPE                  0.5660   0.0442  12.8033 0.0000  0.4794  0.6526
COUNTY_CLEAN              -0.0000   0.0000  -0.5012 0.6163 -0.0001  0.0001
RISK_TYPE_DESC            -0.7328   0.0504 -14.5268 0.0000 -0.8317 -0.6339
PLAN_REGION                0.0552   0.0123   4.4880 0.0000  0.0311  0.0794
ENG_SCORE                 -0.0042   0.0006  -7.4727 0.0000 -0.0053 -0.0031
STATE_CODE                -0.0098   0.0028  -3.4307 0.0006 -0.0154 -0.0042
ADD_STATE                 -0.0034   0.0019  -1.7580 0.0787 -0.0072  0.0004
REG_REGION_DESC            0.0001   0.0003   0.2288 0.8190 -0.0006  0.0007
BEHAVIORAL_ELIGIBLE        0.1212   0.0437   2.7712 0.0056  0.0355  0.2069
ORCA_RISK_GROUP           -0.0299   0.0229  -1.3032 0.1925 -0.0749  0.0151
RACE_ETHNICITY            -0.0593   0.0091  -6.5488 0.0000 -0.0771 -0.0416
OTHER_ELIGIBLE             1.2677   0.0944  13.4316 0.0000  1.0827  1.4527
MEDICAID_ELIGIBLE         -1.4501   0.0694 -20.9078 0.0000 -1.5860 -1.3141
SEX                        0.0484   0.0324   1.4951 0.1349 -0.0150  0.1119
SUD_SEG_DEF               -0.0491   0.0369  -1.3314 0.1831 -0.1214  0.0232
MEMBER_MONTHS_PRE         -0.4464   0.0050 -89.6315 0.0000 -0.4562 -0.4367
SUD_SEG_VALUE             -0.0619   0.0334  -1.8533 0.0638 -0.1273  0.0036
=========================================================================
```

Modeling

Removing variables with higher p-values

In [170…

```
model_data=model_data.drop("COUNTY_CLEAN",axis=1)
```

In [171…

```
model_data=model_data.drop("REG_REGION_DESC",axis=1)
```

In [172…

```
y=model_data.MORE_THAN_4_ER_VISITS
X=model_data.drop("MORE_THAN_4_ER_VISITS",axis=1)
X.describe()
```

Out[172…

	ANNUAL_ER_COSTS	ER_RISK_SCORE	POPHEALTHCAT_GROUPED	RISK_SCORE	RX_RISK_SCORE	I
count	69000.000000	69000.000000	69000.000000	69000.000000	69000.000000	
mean	673.549401	4.417997	2.672145	2.507399	2.268842	
std	1133.082080	6.084783	2.472848	3.567680	2.916019	
min	0.000000	0.289600	0.000000	0.100000	0.134700	
25%	0.000000	0.667100	0.000000	0.354300	0.471700	
50%	121.940000	1.525200	2.000000	1.064300	0.974900	
75%	940.637500	5.366350	4.000000	3.206975	2.905525	
max	7676.760000	26.544900	10.000000	24.902200	17.221500	

8 rows × 34 columns

In [173…

```
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary2())
```

```
Optimization terminated successfully.
         Current function value: 0.199837
         Iterations 9
                         Results: Logit
=========================================================================
Model:              Logit                  Pseudo R-squared:  0.711
Dependent Variable: MORE_THAN_4_ER_VISITS  AIC:               27645.4561
Date:               2021-09-25 21:54       BIC:               27956.2794
No. Observations:   69000                  Log-Likelihood:    -13789.
Df Model:           33                     LL-Null:           -47646.
Df Residuals:       68966                  LLR p-value:       0.0000
Converged:          1.0000                 Scale:             1.0000
No. Iterations:     9.0000
-------------------------------------------------------------------------
                          Coef.  Std.Err.    z     P>|z|   [0.025  0.975]
-------------------------------------------------------------------------
ANNUAL_ER_COSTS          0.0031   0.0000  73.2627 0.0000  0.0030  0.0032
ER_RISK_SCORE            0.2969   0.0063  47.0237 0.0000  0.2846  0.3093
POPHEALTHCAT_GROUPED     0.1716   0.0094  18.3424 0.0000  0.1533  0.1899
RISK_SCORE               0.1687   0.0085  19.7535 0.0000  0.1520  0.1855
RX_RISK_SCORE            0.0607   0.0092   6.6242 0.0000  0.0427  0.0786
INTERVENABLE_IND        -0.2272   0.0419  -5.4267 0.0000 -0.3092 -0.1451
PRIOR_TOTAL_COSTS_ANNUAL 0.0000   0.0000   2.1592 0.0308  0.0000  0.0000
FUTURE_RISK_INPATIENT    0.0233   0.0061   3.8368 0.0001  0.0114  0.0351
ANNUAL_OTHER_COSTS       0.0000   0.0000   6.6355 0.0000  0.0000  0.0000
```

```
BH_RISK_SCORE              0.0110   0.0033   3.3034 0.0010  0.0045  0.0175
ORCA_SCORE                 0.0060   0.0005  12.9573 0.0000  0.0051  0.0070
MEDICAID_CLAIMS            3.4163   0.0755  45.2450 0.0000  3.2683  3.5643
ANNUAL_IP_COSTS            0.0000   0.0000   4.9778 0.0000  0.0000  0.0000
SHORT_DESC_2              -0.0085   0.0011  -7.6168 0.0000 -0.0107 -0.0063
PRIOR_RX_COSTS_ANNUAL     -0.0000   0.0000 -10.9592 0.0000 -0.0000 -0.0000
SHORT_DESC                 0.0118   0.0010  11.6571 0.0000  0.0098  0.0138
AGE                       -0.0082   0.0011  -7.6957 0.0000 -0.0103 -0.0061
RISK_CAT_RECODE            0.0177   0.0019   9.2560 0.0000  0.0139  0.0214
COMPLEXCARE_IND            0.1317   0.0520   2.5306 0.0114  0.0297  0.2337
PLAN_TYPE                  0.5644   0.0441  12.7943 0.0000  0.4779  0.6508
RISK_TYPE_DESC            -0.7354   0.0493 -14.9034 0.0000 -0.8321 -0.6387
PLAN_REGION                0.0547   0.0122   4.4631 0.0000  0.0307  0.0787
ENG_SCORE                 -0.0042   0.0006  -7.4787 0.0000 -0.0053 -0.0031
STATE_CODE                -0.0096   0.0028  -3.4119 0.0006 -0.0151 -0.0041
ADD_STATE                 -0.0035   0.0019  -1.8209 0.0686 -0.0073  0.0003
BEHAVIORAL_ELIGIBLE        0.1302   0.0406   3.2089 0.0013  0.0507  0.2097
ORCA_RISK_GROUP           -0.0297   0.0229  -1.2958 0.1951 -0.0747  0.0152
RACE_ETHNICITY            -0.0596   0.0090  -6.5879 0.0000 -0.0773 -0.0418
OTHER_ELIGIBLE             1.2701   0.0940  13.5124 0.0000  1.0859  1.4544
MEDICAID_ELIGIBLE         -1.4446   0.0686 -21.0638 0.0000 -1.5790 -1.3102
SEX                        0.0487   0.0324   1.5034 0.1327 -0.0148  0.1121
SUD_SEG_DEF               -0.0488   0.0369  -1.3230 0.1858 -0.1211  0.0235
MEMBER_MONTHS_PRE         -0.4464   0.0050 -89.9396 0.0000 -0.4561 -0.4367
SUD_SEG_VALUE             -0.0621   0.0334  -1.8610 0.0627 -0.1276  0.0033
=========================================================================
```

Accuracy

In [175…

```
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
y_pred = logreg.predict(X_test)
print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(logreg.sc
```

```
Accuracy of logistic regression classifier on test set: 0.87
/Users/madhukarayachit/opt/anaconda3/lib/python3.8/site-packages/sklearn/linear_model/_l
ogistic.py:763: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Confusion Matrix

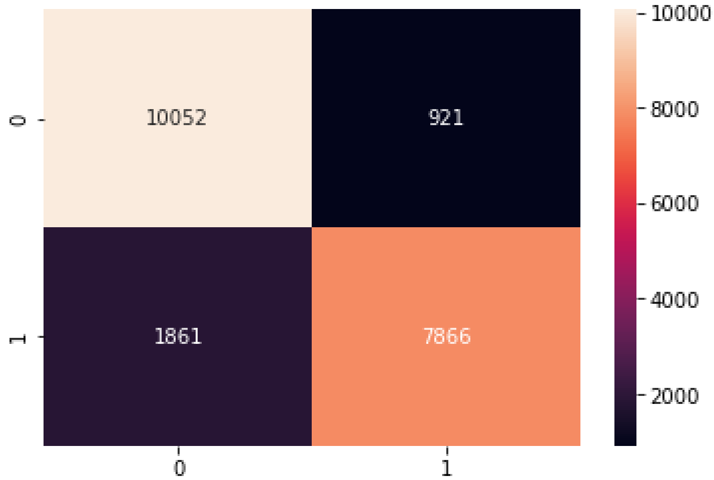
In [176…

```
from sklearn.metrics import confusion_matrix
confusion_matrix = confusion_matrix(y_test, y_pred)
print(confusion_matrix)
sns.heatmap(confusion_matrix,annot=True,fmt="d")
```

```
[[10052   921]
 [ 1861  7866]]
```

Out[176…

```
<AxesSubplot:>
```



Clasification Report

In [177…

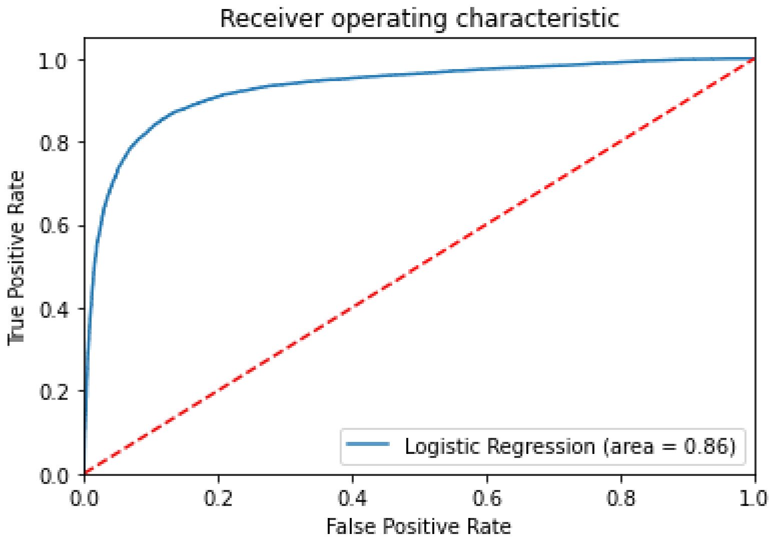
```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.84      0.92      0.88     10973
           1       0.90      0.81      0.85      9727

    accuracy                           0.87     20700
   macro avg       0.87      0.86      0.86     20700
weighted avg       0.87      0.87      0.86     20700
```

ROC Curve

In [178…

```
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
logit_roc_auc = roc_auc_score(y_test, logreg.predict(X_test))
fpr, tpr, thresholds = roc_curve(y_test, logreg.predict_proba(X_test)[:,1])
plt.figure()
plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' % logit_roc_auc)
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.savefig('Log_ROC')
plt.show()
```


Receiver operating characteristic
True Positive Rate
False Positive Rate
Logistic Regression (area = 0.86)
0.0
0.2
0.4
0.6
0.8
1.0

Appendix 2 - R

Process data

Load data

In [1]:
```
# required to pre-load for piping
library('magrittr')

# set plot size
 options(plot.height=3, plot.width=3)

# load dodgers data file
er_data <- read.csv('ER_DATA.csv',sep='|')
```

Exploratory Data Analysis

In [2]:
```
# check data sample
head(er_data)
```

	AGE	SEX	RACE_ETHNICITY	PLAN_TYPE	STATE_CODE	PLAN_REGION	COMPLEXCARE_IND	MI
	<dbl>	<fct>	<fct>	<fct>	<fct>	<fct>	<int>	
1	38	F	White	MARKETPLACE	FL	SOUTHEAST	0	
2	81	M	White	MEDICAID	NY	NORTHEAST	1	
3	30	F	White	MARKETPLACE	TX	SOUTHWEST	0	
4	88	F	White	MEDICARE	TX	SOUTHWEST	0	
5	1	F	Hispanic	MEDICAID	NE	MIDDLESTATES	0	
6	59	M	Asian	MARKETPLACE	NY	NORTHEAST	0	

In [3]:
```
#View the structure of the data and the data types
str(er_data)
```

```
'data.frame':	69000 obs. of  46 variables:
 $ AGE                         : num  38 81 30 88 1 59 61 38 78 102 ...
 $ SEX                         : Factor w/ 2 levels "F","M": 1 2 1 1 1 2 2 2 2 1 ...
 $ RACE_ETHNICITY              : Factor w/ 7 levels "American Indian or A",..: 7 7 7 7 4 2 4
4 3 3 ...
 $ PLAN_TYPE                   : Factor w/ 7 levels "BEHAVIORAL","COMMERCIAL",..: 5 6 5 7 6
5 6 6 7 6 ...
```

```
 $ STATE_CODE               : Factor w/ 37 levels "AL","AR","AZ",..: 7 27 34 34 22 27 34
34 8 4 ...
 $ PLAN_REGION              : Factor w/ 5 levels "MIDDLESTATES",..: 4 2 5 5 1 2 5 5 4 3
...
 $ COMPLEXCARE_IND          : int  0 1 0 0 0 0 1 0 0 1 ...
 $ MMP_DUAL_IND             : int  0 0 0 0 0 0 0 0 0 0 ...
 $ DUAL_PRODUCT_IND         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ LTC_IND                  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ MEDICAID_ELIGIBLE        : int  0 1 0 0 1 0 1 1 0 1 ...
 $ MEDICARE_ELIGIBLE        : int  0 0 0 1 0 0 0 0 2 0 ...
 $ BEHAVIORAL_ELIGIBLE      : int  0 0 0 0 0 0 1 1 0 0 ...
 $ COMMERCIAL_ELIGIBLE      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ OTHER_ELIGIBLE           : int  1 0 1 0 0 1 0 0 0 0 ...
 $ RISK_TYPE_DESC           : Factor w/ 5 levels "","DUAL RISK",..: 1 1 1 1 1 1 1 1 1 3
...
 $ MEMBER_MONTHS_PRE        : num  12 12 1.94 4.93 12 ...
 $ ADD_STATE                : Factor w/ 73 levels "","12","13","15",..: 34 59 68 68 54 59
68 68 35 30 ...
 $ COUNTY_CLEAN             : Factor w/ 1221 levels ""," ","ABBEVILLE",..: 840 763 1190 1
973 750 354 648 1 642 ...
 $ REG_REGION_DESC          : Factor w/ 192 levels "** NO MATCH FOUND **",..: 5 2 22 2 12
4 2 52 91 2 2 ...
 $ RISK_SCORE               : num  2.744 3.18 0.46 0.58 0.249 ...
 $ PRIOR_TOTAL_COSTS_ANNUAL: num  2394 20920 417 521 1392 ...
 $ PRIOR_RX_COSTS_ANNUAL    : num  509.1 2091.12 416.59 433.81 3.63 ...
 $ ANNUAL_IP_COSTS          : num  0 13366 0 0 0 ...
 $ ANNUAL_ER_COSTS          : num  0 0 0 0 0 ...
 $ ANNUAL_OTHER_COSTS       : num  1885.4 5462.7 0 87.6 1388 ...
 $ FUTURE_RISK_INPATIENT    : num  1.267 5.192 0.617 2.255 0.7 ...
 $ BH_RISK_SCORE            : num  0.507 0.725 0.268 1.55 0.195 ...
 $ RX_RISK_SCORE            : num  1.518 5.802 0.636 3.03 0.424 ...
 $ ER_RISK_SCORE            : num  1.915 8.313 0.647 2.096 0.948 ...
 $ ORCA_SCORE               : int  93 2 1 17 0 83 32 4 87 1 ...
 $ ORCA_RISK_GROUP          : Factor w/ 4 levels "","HIGH","LOW",..: 4 3 3 3 3 4 3 3 4 3
...
 $ SUD_SEG_VALUE            : int  6 6 6 6 6 6 6 6 6 6 ...
 $ SUD_SEG_DEF              : Factor w/ 8 levels "","01: High Cost SUD Member - No Treatm
ent",..: 7 7 7 7 7 7 7 7 7 7 ...
 $ ENG_SCORE                : num  40 78 56 90 38 77 83 26 66 96 ...
 $ POPHEALTHCAT_GROUPED     : Factor w/ 11 levels "01: Healthy",..: 6 5 1 3 1 4 9 1 9 1
...
 $ INTERVENABLE_IND         : int  0 1 0 0 0 0 0 0 0 0 ...
 $ SHORT_DESC               : Factor w/ 94 levels "","Acute and chronic renal failur
e",..: 64 36 68 56 64 36 77 23 63 23 ...
 $ SHORT_DESC_2             : Factor w/ 86 levels "Abdominal Infection/Pain",..: 23 44 35
59 23 44 64 22 45 22 ...
 $ RISK_CAT_RECODE          : Factor w/ 31 levels "","AIDS/HIV",..: 9 19 16 3 9 19 24 8 5
8 ...
 $ MEDICAID_CLAIMS          : int  0 1 0 0 1 0 1 0 0 1 ...
 $ MEDICARE_CLAIMS          : int  0 0 0 1 0 0 0 0 1 0 ...
 $ BEHAVIORAL_CLAIMS        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ COMMERCIAL_CLAIMS        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ OTHER_CLAIMS             : int  1 0 0 0 0 1 0 0 0 0 ...
 $ MORE_THAN_4_ER_VISITS    : int  0 0 0 0 0 0 0 0 0 0 ...
```

In [4]:

```
# Check data quality and basic statistics
psych::describe(er_data)
```

A psych: 46 x

	vars	n	mean	sd	median	trimmed
	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>

	vars	n	mean	sd	median	trimmed	
	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
AGE	1	69000	3.202965e+01	2.226675e+01	29.0000	3.078391e+01	2
SEX*	2	69000	1.421623e+00	4.938224e-01	1.0000	1.402029e+00	
RACE_ETHNICITY*	3	69000	5.212435e+00	1.809818e+00	6.0000	5.327772e+00	
PLAN_TYPE*	4	69000	5.907913e+00	6.870157e-01	6.0000	5.969004e+00	
STATE_CODE*	5	69000	1.632291e+01	1.070050e+01	14.0000	1.567174e+01	1
PLAN_REGION*	6	69000	3.116449e+00	1.360601e+00	3.0000	3.145562e+00	
COMPLEXCARE_IND	7	69000	1.485217e-01	3.556190e-01	0.0000	6.065217e-02	
MMP_DUAL_IND	8	69000	1.802899e-02	1.330571e-01	0.0000	0.000000e+00	
DUAL_PRODUCT_IND	9	69000	2.978261e-02	1.699883e-01	0.0000	0.000000e+00	
LTC_IND	10	69000	1.128986e-02	1.056530e-01	0.0000	0.000000e+00	
MEDICAID_ELIGIBLE	11	69000	8.598116e-01	4.825234e-01	1.0000	8.802174e-01	
MEDICARE_ELIGIBLE	12	69000	9.028986e-02	2.895168e-01	0.0000	0.000000e+00	
BEHAVIORAL_ELIGIBLE	13	69000	2.700580e-01	4.524962e-01	0.0000	2.078442e-01	
COMMERCIAL_ELIGIBLE	14	69000	1.531884e-02	1.236418e-01	0.0000	0.000000e+00	
OTHER_ELIGIBLE	15	69000	1.138116e-01	3.223856e-01	0.0000	1.536232e-02	
RISK_TYPE_DESC*	16	69000	1.265072e+00	8.848860e-01	1.0000	1.000000e+00	
MEMBER_MONTHS_PRE	17	68998	9.855747e+00	3.584698e+00	12.0000	1.058881e+01	
ADD_STATE*	18	69000	4.341897e+01	1.565407e+01	40.0000	4.341953e+01	1
COUNTY_CLEAN*	19	69000	4.429531e+02	3.879715e+02	396.0000	4.138443e+02	58
REG_REGION_DESC*	20	69000	4.988143e+01	5.949340e+01	10.0000	4.119237e+01	1
RISK_SCORE	21	68935	3.293383e+00	7.210057e+00	1.0643	1.840817e+00	
PRIOR_TOTAL_COSTS_ANNUAL	22	68935	1.430178e+04	4.219008e+04	2511.1300	5.905007e+03	370
PRIOR_RX_COSTS_ANNUAL	23	68935	2.893580e+03	1.941766e+04	80.0000	4.996784e+02	11
ANNUAL_IP_COSTS	24	68935	3.286617e+03	1.975213e+04	0.0000	1.464175e+02	
ANNUAL_ER_COSTS	25	68935	8.081706e+02	2.291147e+03	121.9400	4.348911e+02	18
ANNUAL_OTHER_COSTS	26	68935	7.313408e+03	2.335150e+04	1309.2400	2.879420e+03	194
FUTURE_RISK_INPATIENT	27	68935	3.848280e+00	7.122357e+00	0.8823	1.850348e+00	
BH_RISK_SCORE	28	68935	5.133888e+00	1.093811e+01	0.7240	2.394595e+00	
RX_RISK_SCORE	29	68935	2.759930e+00	4.826812e+00	0.9749	1.740806e+00	
ER_RISK_SCORE	30	68935	5.087040e+00	7.156474e+00	1.5252	3.429224e+00	
ORCA_SCORE	31	65600	3.676759e+01	4.116319e+01	11.0000	3.355610e+01	1
ORCA_RISK_GROUP*	32	69000	2.876000e+00	6.863000e-01	3.0000	2.906594e+00	

	vars	n	mean	sd	median	trimmed
	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
SUD_SEG_VALUE	33	68935	5.449801e+00	1.321688e+00	6.0000	5.821393e+00
SUD_SEG_DEF*	34	69000	6.543072e+00	1.175051e+00	7.0000	6.871123e+00
ENG_SCORE	35	68935	5.111443e+01	3.022647e+01	51.0000	5.135368e+01
POPHEALTHCAT_GROUPED*	36	69000	3.672145e+00	2.472848e+00	3.0000	3.323841e+00
INTERVENABLE_IND	37	69000	2.938986e-01	4.555493e-01	0.0000	2.423732e-01
SHORT_DESC*	38	69000	5.014596e+01	2.633632e+01	60.0000	5.053147e+01
SHORT_DESC_2*	39	69000	4.187752e+01	2.439846e+01	35.0000	4.164670e+01
RISK_CAT_RECODE*	40	69000	1.481843e+01	9.223851e+00	14.0000	1.444676e+01
MEDICAID_CLAIMS	41	69000	6.745797e-01	4.685351e-01	1.0000	7.182246e-01
MEDICARE_CLAIMS	42	69000	8.349275e-02	2.766276e-01	0.0000	0.000000e+00
BEHAVIORAL_CLAIMS	43	69000	2.898551e-05	5.383780e-03	0.0000	0.000000e+00
COMMERCIAL_CLAIMS	44	69000	1.223188e-02	1.099202e-01	0.0000	0.000000e+00
OTHER_CLAIMS	45	69000	8.957971e-02	2.855808e-01	0.0000	0.000000e+00
MORE_THAN_4_ER_VISITS	46	69000	4.637681e-01	4.986891e-01	0.0000	4.547101e-01

Check categorical columns

In [5]:

```
# examine the data through descriptive statistics
Hmisc::describe(er_data)
```

```
er_data

 46  Variables      69000  Observations
--------------------------------------------------------------------------------
AGE
       n  missing distinct     Info     Mean      Gmd      .05      .10
   69000        0      105        1    32.03    25.38        1        4
     .25      .50      .75      .90      .95
      14       29       49       63       71

lowest :   0   1   2   3   4, highest: 100 101 102 103 105
--------------------------------------------------------------------------------
SEX
       n  missing distinct
   69000        0        2

Value          F     M
Frequency  39908 29092
Proportion 0.578 0.422
--------------------------------------------------------------------------------
RACE_ETHNICITY
       n  missing distinct
   69000        0        7

lowest : American Indian or A Asian                Black or African Ame Hispanic
Native Hawaiian and
highest: Black or African Ame Hispanic             Native Hawaiian and  Unknown
```

```
White

American Indian or A (310, 0.004), Asian (2815, 0.041), Black or African Ame
(14842, 0.215), Hispanic (13209, 0.191), Native Hawaiian and (92, 0.001),
Unknown (8228, 0.119), White (29504, 0.428)
--------------------------------------------------------------------------------
PLAN_TYPE
       n  missing distinct
   69000        0        7

lowest : BEHAVIORAL   COMMERCIAL   CORRECTIONAL DUALS        MARKETPLACE
highest: CORRECTIONAL DUALS        MARKETPLACE  MEDICAID     MEDICARE

Value        BEHAVIORAL   COMMERCIAL CORRECTIONAL        DUALS  MARKETPLACE
Frequency            10         1035          325           31         7210
Proportion        0.000        0.015        0.005        0.000        0.104

Value          MEDICAID     MEDICARE
Frequency         54306         6083
Proportion        0.787        0.088
--------------------------------------------------------------------------------
STATE_CODE
       n  missing distinct
   69000        0       37

lowest : AL AR AZ CA CT, highest: TN TX VT WA WI
--------------------------------------------------------------------------------
PLAN_REGION
       n  missing distinct
   69000        0        5

lowest : MIDDLESTATES NORTHEAST    PACIFIC      SOUTHEAST    SOUTHWEST
highest: MIDDLESTATES NORTHEAST    PACIFIC      SOUTHEAST    SOUTHWEST

Value      MIDDLESTATES    NORTHEAST      PACIFIC    SOUTHEAST    SOUTHWEST
Frequency         13600         8992        12706        23177        10525
Proportion        0.197        0.130        0.184        0.336        0.153
--------------------------------------------------------------------------------
COMPLEXCARE_IND
       n  missing distinct     Info      Sum     Mean      Gmd
   69000        0        2    0.379    10248   0.1485   0.2529

--------------------------------------------------------------------------------
MMP_DUAL_IND
       n  missing distinct     Info      Sum     Mean      Gmd
   69000        0        2    0.053     1244  0.01803  0.03541

--------------------------------------------------------------------------------
DUAL_PRODUCT_IND
       n  missing distinct     Info      Sum     Mean      Gmd
   69000        0        2    0.087     2055  0.02978  0.05779

--------------------------------------------------------------------------------
LTC_IND
       n  missing distinct     Info      Sum     Mean      Gmd
   69000        0        2    0.033      779  0.01129  0.02233

--------------------------------------------------------------------------------
MEDICAID_ELIGIBLE
       n  missing distinct     Info     Mean      Gmd
   69000        0        5    0.572   0.8598   0.4201

lowest : 0 1 2 3 5, highest: 0 1 2 3 5

Value          0     1     2     3     5
```

```
Frequency  13512 51681  3777    29     1
Proportion 0.196 0.749 0.055 0.000 0.000
--------------------------------------------------------------------------------
MEDICARE_ELIGIBLE
       n  missing distinct     Info     Mean      Gmd
   69000        0        3    0.244  0.09029   0.1646

Value          0     1     2
Frequency  62828  6114    58
Proportion 0.911 0.089 0.001
--------------------------------------------------------------------------------
BEHAVIORAL_ELIGIBLE
       n  missing distinct     Info     Mean      Gmd
   69000        0        4    0.587   0.2701   0.3983

Value          0     1     2     3
Frequency  50627 18114   257     2
Proportion 0.734 0.263 0.004 0.000
--------------------------------------------------------------------------------
COMMERCIAL_ELIGIBLE
       n  missing distinct     Info     Mean      Gmd
   69000        0        3    0.045  0.01532  0.03017

Value          0     1     2
Frequency  67950  1043     7
Proportion 0.985 0.015 0.000
--------------------------------------------------------------------------------
OTHER_ELIGIBLE
       n  missing distinct     Info     Mean      Gmd
   69000        0        4    0.299   0.1138   0.2024

Value          0     1     2     3
Frequency  61252  7644   103     1
Proportion 0.888 0.111 0.001 0.000
--------------------------------------------------------------------------------
RISK_TYPE_DESC
       n  missing distinct
   69000        0        5

lowest :                  DUAL RISK       FEE FOR SERVICE FULL RISK       SHARED RISK
highest:                  DUAL RISK       FEE FOR SERVICE FULL RISK       SHARED RISK

Value                              DUAL RISK FEE FOR SERVICE       FULL RISK
Frequency            62109            1696            1681             824
Proportion           0.900           0.025           0.024           0.012

Value          SHARED RISK
Frequency             2690
Proportion           0.039
--------------------------------------------------------------------------------
MEMBER_MONTHS_PRE
       n  missing distinct     Info     Mean      Gmd      .05      .10
   68998        2      362    0.707    9.856    3.278    1.941    2.928
     .25      .50      .75      .90      .95
   7.895   12.000   12.000   12.000   12.000

lowest :  0.0658  0.0987  0.1316  0.1645  0.1974
highest: 11.8750 11.9079 11.9408 11.9737 12.0000
--------------------------------------------------------------------------------
ADD_STATE
       n  missing distinct
   69000        0       73

lowest :    12 13 15 17, highest: UT VA VT WA WI
--------------------------------------------------------------------------------
```

```
COUNTY_CLEAN
       n  missing distinct 
   69000        0     1221 

lowest :                     ABBEVILLE ACADIA    ACCOMACK 
highest: YOUNG     YUBA      YUMA      ZAPATA    ZAVALA   
--------------------------------------------------------------------------------
REG_REGION_DESC 
       n  missing distinct 
   69000        0      192 

lowest : ** NO MATCH FOUND **       ** NOT PROVIDED   **       AD                        
All AZ Regions             All FL Regions            
highest: WEST CENTRAL INDIANA       West CFC - Cinci Child     West CFC - Non Cinci Chil
d WESTERN                    Wisconsin                 
--------------------------------------------------------------------------------
RISK_SCORE 
       n  missing distinct     Info     Mean      Gmd      .05      .10 
   68935       65    35194        1    3.293    4.595   0.1000   0.1358 
     .25      .50      .75      .90      .95 
  0.3543   1.0643   3.4656   7.6434  11.9494 

lowest :   0.1000   0.1016   0.1074   0.1155   0.1241
highest:  99.1961  99.6001 100.3859 101.8011 105.0436
--------------------------------------------------------------------------------
PRIOR_TOTAL_COSTS_ANNUAL 
       n  missing distinct     Info     Mean      Gmd      .05      .10 
   68935       65    58730    0.999    14302    22748      0.0      0.0 
     .25      .50      .75      .90      .95 
   387.7   2511.1  10858.3  35598.6  65995.7 

lowest :       0.00       0.15       0.20       0.24       0.27
highest: 1272214.85 1664410.65 1724977.95 2105454.98 2116494.01
--------------------------------------------------------------------------------
PRIOR_RX_COSTS_ANNUAL 
       n  missing distinct     Info     Mean      Gmd      .05      .10 
   68935       65    37449    0.981     2894     5265      0.0      0.0 
     .25      .50      .75      .90      .95 
     0.0     80.0    763.1   5007.3  11636.1 

lowest :       0.00       0.01       0.02       0.03       0.04
highest:  834213.58  913792.63 1203595.10 2076825.97 2103466.22
--------------------------------------------------------------------------------
ANNUAL_IP_COSTS 
       n  missing distinct     Info     Mean      Gmd      .05      .10 
   68935       65     9017    0.382     3287     6243        0        0 
     .25      .50      .75      .90      .95 
       0        0        0     4695    15488 

lowest :       0.00       4.82      13.73      34.24      50.22
highest:  728725.33  804726.76  844176.59 1335664.48 1452105.77
--------------------------------------------------------------------------------
ANNUAL_ER_COSTS 
       n  missing distinct     Info     Mean      Gmd      .05      .10 
   68935       65    32405    0.896    808.2     1235      0.0      0.0 
     .25      .50      .75      .90      .95 
     0.0    121.9    989.0   2199.9   3349.0 

lowest :      0.00      0.24      0.30      1.24      1.27
highest:  75902.95  81748.40  97316.47 117117.70 312528.31

Value           0   5000  10000  15000  20000  25000  30000  35000  40000
Frequency   63306   4929    491    100     55     14     10      9      8
Proportion  0.918  0.072  0.007  0.001  0.001  0.000  0.000  0.000  0.000
```

```
Value       45000  55000  60000  70000  75000  80000  95000 115000 315000
Frequency       3      1      1      3      1      1      1      1      1
Proportion  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000

For the frequency table, variable is rounded to the nearest 5000
--------------------------------------------------------------------------------
ANNUAL_OTHER_COSTS
       n  missing distinct     Info     Mean      Gmd      .05      .10
   68935       65    54845    0.996     7313    11973      0.0      0.0
     .25      .50      .75      .90      .95
   206.8   1309.2   5421.9  16276.1  32059.1

lowest : -418331.1 -337918.3 -187334.9 -159611.5 -149616.9
highest:  594933.1  636623.9  660672.7  994105.1 1239582.3
--------------------------------------------------------------------------------
FUTURE_RISK_INPATIENT
       n  missing distinct     Info     Mean      Gmd      .05      .10
   68935       65    24523        1    3.848     5.31   0.5879   0.6075
     .25      .50      .75      .90      .95
  0.6386   0.8823   2.5404  11.9109  22.3316

lowest :  0.5858  0.5879  0.5910  0.5932  0.5939
highest: 36.6167 36.6188 36.6259 36.6287 36.6300
--------------------------------------------------------------------------------
BH_RISK_SCORE
       n  missing distinct     Info     Mean      Gmd      .05      .10
   68935       65    16146    0.999    5.134    7.887    0.131    0.195
     .25      .50      .75      .90      .95
   0.270    0.724    4.239   15.304   26.658

lowest :   0.100   0.101   0.102   0.103   0.104
highest: 130.419 130.538 133.453 135.587 139.763
--------------------------------------------------------------------------------
RX_RISK_SCORE
       n  missing distinct     Info     Mean      Gmd      .05      .10
   68935       65    28740        1     2.76    3.555   0.2046   0.2852
     .25      .50      .75      .90      .95
  0.4717   0.9749   3.1694   6.9881  10.1655

lowest :  0.1347  0.1742  0.1794  0.1816  0.1846
highest: 63.8118 65.6683 66.6049 71.9618 74.5865
--------------------------------------------------------------------------------
ER_RISK_SCORE
       n  missing distinct     Info     Mean      Gmd      .05      .10
   68935       65    28667        1    5.087    6.586   0.3620   0.4464
     .25      .50      .75      .90      .95
  0.6667   1.5252   6.2923  16.6256  22.4544

lowest :  0.2896  0.3320  0.3374  0.3469  0.3473
highest: 33.5977 33.6080 33.6198 33.6532 33.7343
--------------------------------------------------------------------------------
ORCA_SCORE
       n  missing distinct     Info     Mean      Gmd      .05      .10
   65600     3400       99    0.943    36.77    43.96        0        0
     .25      .50      .75      .90      .95
       0       11       85       98       99

lowest :   0   1   2   3   4, highest:  96  97  98  99 100
--------------------------------------------------------------------------------
ORCA_RISK_GROUP
       n  missing distinct
   69000        0        4

Value              HIGH    LOW MEDIUM
Frequency    3400  10858  45640   9102
```

```
Proportion  0.049  0.157  0.661  0.132
--------------------------------------------------------------------------------
SUD_SEG_VALUE
       n  missing distinct     Info     Mean      Gmd 
   68935       65        7    0.483     5.45   0.9522 

lowest : 0 1 2 3 4, highest: 2 3 4 5 6
                                                    
Value          0     1     2     3     4     5     6
Frequency    970  1284  1921  4232    67  5174 55287
Proportion 0.014 0.019 0.028 0.061 0.001 0.075 0.802
--------------------------------------------------------------------------------
SUD_SEG_DEF 
       n  missing distinct 
   69000        0        8 

lowest :                                           01: High Cost SUD Member - No Treatme
nt   02: High Cost SUD Member - Some Treatment 03: Not High Cost SUD Member
04: Harmful Use Rx Only (No SUD Dx)
highest: 03: Not High Cost SUD Member              04: Harmful Use Rx Only (No SUD Dx)
05: Nicotine Only (Not in segments 1-4)   06: No known SUD behavior                 New
--------------------------------------------------------------------------------
ENG_SCORE 
       n  missing distinct     Info     Mean      Gmd      .05      .10 
   68935       65      101        1    51.11    34.88        4        9 
     .25      .50      .75      .90      .95 
      24       51       79       92       96 

lowest :   0   1   2   3   4, highest:  96  97  98  99 100
--------------------------------------------------------------------------------
POPHEALTHCAT_GROUPED 
       n  missing distinct 
   69000        0       11 

lowest : 01: Healthy                        02: Acute Episodic                  03: Chr
onic Stable PH/BH           04: Health Coaching                05: Chronic Interventio
nal PH/BH CM
highest: 07: Catastrophic Conditions        08: Dementia and Custodial Care     09: LTS
S                           10: End of Life Care               99: Unclassified
--------------------------------------------------------------------------------
INTERVENABLE_IND 
       n  missing distinct     Info      Sum     Mean      Gmd 
   69000        0        2    0.623    20279   0.2939   0.4151 

--------------------------------------------------------------------------------
SHORT_DESC 
       n  missing distinct 
   69000        0       94 

lowest :                                               Acute and chronic renal failure
Acute bronchitis                              Agents used to treat enzyme deficiency sta
tes AIDS/HIV
highest: Septicemia                                     Sickle-cell anemia
Substance Abuse                               Ulcers, gastritis/duodenitis
Valvular disorders
--------------------------------------------------------------------------------
SHORT_DESC_2 
       n  missing distinct 
   69000        0       86 

lowest : Abdominal Infection/Pain        Acute And Chronic Renal Failure Acute Bronchiti
s                Adhd/Idd/Autism                 Aids/Hiv
highest: Substance Abuse                 Ulcers, Gastritis/Duodenitis    Upper Gi Inflam
mation/Infection Urology                         Valvular Disorders
--------------------------------------------------------------------------------
```

```
RISK_CAT_RECODE
       n  missing distinct 
   69000        0       31 

lowest :                              AIDS/HIV                     Behavioral Health (In
c. SUD) Cancer                       Cardiology
highest: Orthopedics                  Other Medical                Pulmonology
Rheumatology                 Urology
--------------------------------------------------------------------------------
MEDICAID_CLAIMS
       n  missing distinct     Info      Sum     Mean      Gmd 
   69000        0        2    0.659    46546   0.6746   0.4391 

--------------------------------------------------------------------------------
MEDICARE_CLAIMS
       n  missing distinct     Info      Sum     Mean      Gmd 
   69000        0        2     0.23     5761  0.08349    0.153 

--------------------------------------------------------------------------------
BEHAVIORAL_CLAIMS
        n   missing  distinct      Info       Sum      Mean       Gmd 
    69000         0         2         0         2 2.899e-05 5.797e-05 

--------------------------------------------------------------------------------
COMMERCIAL_CLAIMS
       n  missing distinct     Info      Sum     Mean      Gmd 
   69000        0        2    0.036      844  0.01223  0.02416 

--------------------------------------------------------------------------------
OTHER_CLAIMS
       n  missing distinct     Info      Sum     Mean      Gmd 
   69000        0        2    0.245     6181  0.08958   0.1631 

--------------------------------------------------------------------------------
MORE_THAN_4_ER_VISITS
       n  missing distinct     Info      Sum     Mean      Gmd 
   69000        0        2    0.746    32000   0.4638   0.4974 

--------------------------------------------------------------------------------
```

Therefore, the observations are:

- The outcome variable is fairly balanced, as such there will not be a need to oversample or downsample the data
- The data is fairly clean with minimimal missing records.
- The majority of patients are female at fifthy eight percent compared to male are forthy two percent
- The data is made up of sixty nine thousand records and forthy six variables

Box plot

ER Risk Score by Race & Ethnicity

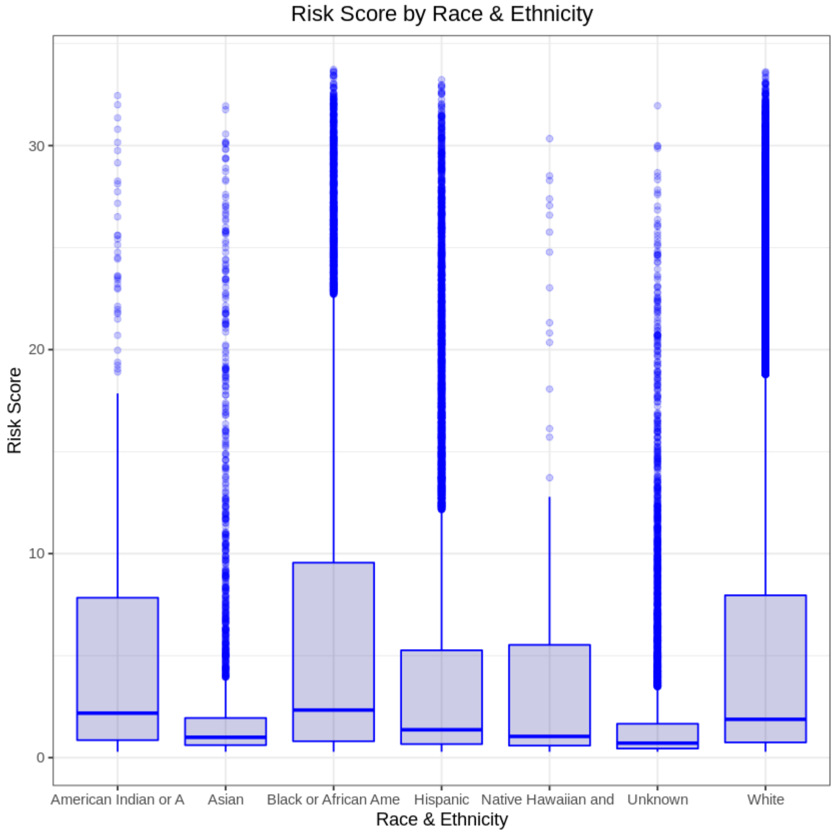
In [6]:

```
# set plot size
options(plot.height=3, plot.width=3)
```

In [7]:

```
ggplot2::ggplot(er_data, ggplot2::aes(x=RACE_ETHNICITY, y=ER_RISK_SCORE)) +
    ggplot2::geom_boxplot(color="blue", fill="darkblue", alpha=0.2) +
    ggplot2::labs(title='Risk Score by Race & Ethnicity',
                  x='Race & Ethnicity',
                  y='Risk Score')+
    ggplot2::theme_bw() +
    ggplot2::theme(plot.title = ggplot2::element_text(hjust = 0.5))
```

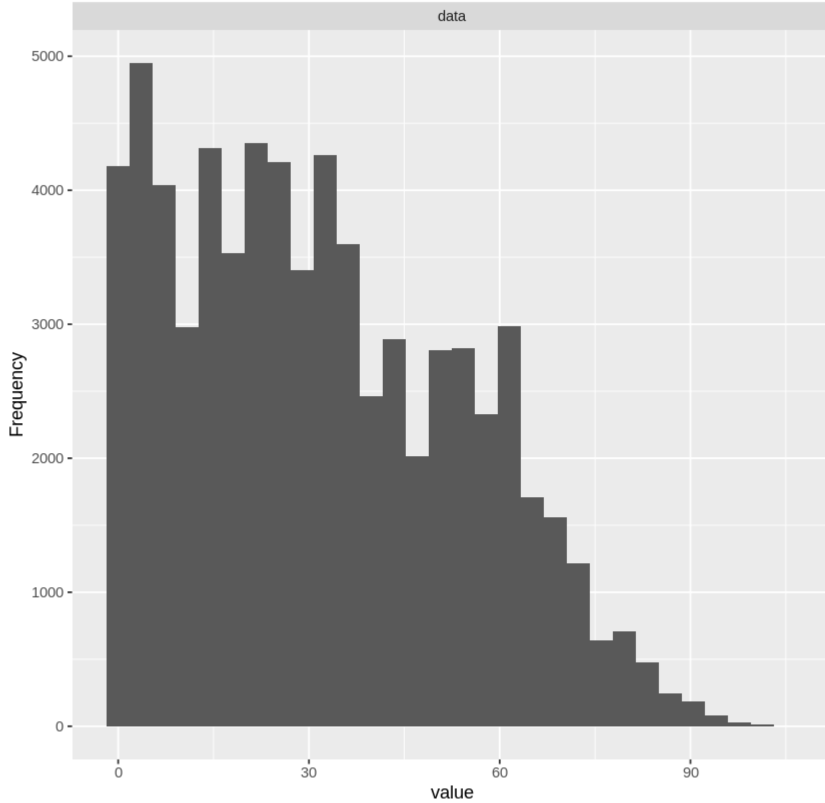
```
Warning message:
“Removed 65 rows containing non-finite values (stat_boxplot).”
```



Based on the plot, african american patients seem to have higher risk scores compared to other race & ethnicity

In [8]:

```
# Histogram of the AGE variable
DataExplorer::plot_histogram(er_data$AGE)
```

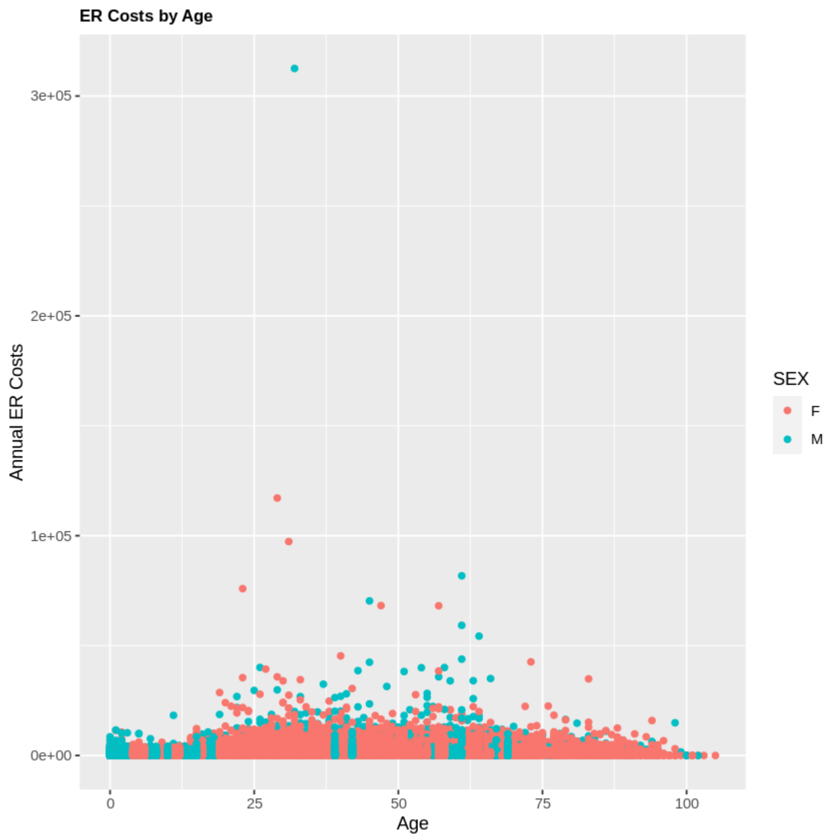


The data is highly skewed and patients are relatively younger with the majority of the population being 80 years old or younger as you would expect.

In [9]:

```
#Strip Plot of Attendance by opponent or visiting team
ggplot2::ggplot(er_data, ggplot2::aes(x=AGE, y=ANNUAL_ER_COSTS, color = SEX)) +
        ggplot2::geom_point() +
        ggplot2::labs(title = 'ER Costs by Age',
                      x='Age',
                      y='Annual ER Costs') +
        ggplot2::theme(plot.title = ggplot2::element_text(lineheight=3, face="bold", co
```

```
Warning message:
“Removed 65 rows containing missing values (geom_point).”
```



This plot further confirms that the data is fairly contained with relatively very few outliers.