Anomaly Detection

Data collection and exploration

```
# Import necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Read dataset
data=pd.read_csv('/content/drive/MyDrive/AI-Inernship/Healthcare Providers.csv')
data.head()
```

	index	National Provider Identifier	Last Name/Organization Name of the Provider	First Name of the Provider	Middle Initial of the Provider	Credentials of the Provider	Gende of th Provide
0	8774979	1891106191	UPADHYAYULA	SATYASREE	NaN	M.D.	
1	3354385	1346202256	JONES	WENDY	P	M.D.	
2	3001884	1306820956	DUROCHER	RICHARD	W	DPM	I
3	7594822	1770523540	FULLARD	JASPER	NaN	MD	I
4	746159	1073627758	PERROTTI	ANTHONY	E	DO	I

5 rows × 27 columns

```
data.info()  # information about the data such as entries, datatypes, no of rows and columns, coun
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 27 columns):
 #   Column                                                    Non-Null Count   Dtype
---  ------                                                    --------------   -----
 0   index                                                     100000 non-null  int64
 1   National Provider Identifier                              100000 non-null  int64
 2   Last Name/Organization Name of the Provider               100000 non-null  object
 3   First Name of the Provider                                95745 non-null   object
 4   Middle Initial of the Provider                            70669 non-null   object
 5   Credentials of the Provider                               92791 non-null   object
 6   Gender of the Provider                                    95746 non-null   object
 7   Entity Type of the Provider                               100000 non-null  object
 8   Street Address 1 of the Provider                          100000 non-null  object
 9   Street Address 2 of the Provider                          40637 non-null   object
 10  City of the Provider                                      100000 non-null  object
 11  Zip Code of the Provider                                  100000 non-null  float64
 12  State Code of the Provider                                100000 non-null  object
 13  Country Code of the Provider                              100000 non-null  object
 14  Provider Type                                             100000 non-null  object
 15  Medicare Participation Indicator                          100000 non-null  object
 16  Place of Service                                          100000 non-null  object
 17  HCPCS Code                                                100000 non-null  object
 18  HCPCS Description                                         100000 non-null  object
 19  HCPCS Drug Indicator                                      100000 non-null  object
 20  Number of Services                                        100000 non-null  object
 21  Number of Medicare Beneficiaries                          100000 non-null  object
 22  Number of Distinct Medicare Beneficiary/Per Day Services  100000 non-null  object
 23  Average Medicare Allowed Amount                           100000 non-null  object
 24  Average Submitted Charge Amount                           100000 non-null  object
 25  Average Medicare Payment Amount                           100000 non-null  object
 26  Average Medicare Standardized Amount                      100000 non-null  object
dtypes: float64(1), int64(2), object(24)
memory usage: 20.6+ MB
```

Data Preprocessing

- The very first step of preprocessing is starting from droping the index columns.
- As index column does not have any significance for EDA

```
df=data.drop('index',axis=1)
df.head()
```

	National Provider Identifier	Last Name/Organization Name of the Provider	First Name of the Provider	Middle Initial of the Provider	Credentials of the Provider	Gender of the Provider	Enti Type t Provid
0	1891106191	UPADHYAYULA	SATYASREE	NaN	M.D.	F	
1	1346202256	JONES	WENDY	P	M.D.	F	
2	1306820956	DUROCHER	RICHARD	W	DPM	M	
3	1770523540	FULLARD	JASPER	NaN	MD	M	
4	1073627758	PERROTTI	ANTHONY	E	DO	M	

5 rows × 26 columns

```
# Printing all the columns in the dataset
df.columns
```

```
Index(['National Provider Identifier',
       'Last Name/Organization Name of the Provider',
       'First Name of the Provider', 'Middle Initial of the Provider',
       'Credentials of the Provider', 'Gender of the Provider',
       'Entity Type of the Provider', 'Street Address 1 of the Provider',
       'Street Address 2 of the Provider', 'City of the Provider',
       'Zip Code of the Provider', 'State Code of the Provider',
       'Country Code of the Provider', 'Provider Type',
       'Medicare Participation Indicator', 'Place of Service', 'HCPCS Code',
       'HCPCS Description', 'HCPCS Drug Indicator', 'Number of Services',
       'Number of Medicare Beneficiaries',
       'Number of Distinct Medicare Beneficiary/Per Day Services',
       'Average Medicare Allowed Amount', 'Average Submitted Charge Amount',
       'Average Medicare Payment Amount',
       'Average Medicare Standardized Amount'],
      dtype='object')
```

Handling of name columns

- As we can see there are three columns in the dataset for name of the provider.

- we can create a separate column for the name of the provider as **'Name of The Provider'** by combining given three columns.

```
# creating a new column for name of the provider by joining existing three columns
df['First Name of the Provider'] = df['First Name of the Provider'].fillna("")
df['Middle Initial of the Provider'] = df['Middle Initial of the Provider'].fillna("")
df['First Name of the Provider']=df['First Name of the Provider'] + df['Middle Initial of the Prov
df.rename(columns={'First Name of the Provider':'Name of the Provider'},inplace=True)
df['Name of the Provider'] = df['Name of the Provider'].str.strip()
```

```
# We are droping unnecessary columns
drop_cols=['Middle Initial of the Provider',
           'Last Name/Organization Name of the Provider']
df=df.drop(drop_cols,axis=1)
```

- Similarly there are two columns for the address of the provider.
- We can combine both of them as a single column name **'Street Address of the Provider'**.

```
# joining of street addresses
df['Street Address 2 of the Provider'] = df['Street Address 2 of the Provider'].fillna("")
df['Street Address 1 of the Provider']=df['Street Address 1 of the Provider'] + df['Street Address
df.drop(['Street Address 2 of the Provider'],axis=1,inplace=True)
df.rename(columns={'Street Address 1 of the Provider':'Street Address of the Provider'},inplace=Tr
```

```
# converting the values like M.D. as MD As it has the same meaning
df['Credentials of the Provider']=df['Credentials of the Provider'].str.replace(".","")
```

Preprocessing for numerical columns

```
# @title Preprocessing for numerical columns
df.iloc[:,16:]=df.iloc[:,16:].apply(lambda x: x.str.replace(',', ''))
```

Converting columns of 'object' datatypes as 'Float'

```
# @title Converting columns of 'object' datatypes as 'Float'
object_cols = df.iloc[:,16:].columns
df[object_cols] = df[object_cols].apply(lambda x: x.astype(float))
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 23 columns):
 #   Column                                                    Non-Null Count   Dtype
---  ------                                                    --------------   -----
 0   National Provider Identifier                              100000 non-null  int64
 1   Name of the Provider                                      100000 non-null  object
 2   Credentials of the Provider                               92791 non-null   object
 3   Gender of the Provider                                    95746 non-null   object
 4   Entity Type of the Provider                               100000 non-null  object
 5   Street Address of the Provider                            100000 non-null  object
 6   City of the Provider                                      100000 non-null  object
 7   Zip Code of the Provider                                  100000 non-null  float64
 8   State Code of the Provider                                100000 non-null  object
 9   Country Code of the Provider                              100000 non-null  object
 10  Provider Type                                             100000 non-null  object
 11  Medicare Participation Indicator                          100000 non-null  object
 12  Place of Service                                          100000 non-null  object
 13  HCPCS Code                                                100000 non-null  object 
 14  HCPCS Description                                         100000 non-null  object 
 15  HCPCS Drug Indicator                                      100000 non-null  object 
 16  Number of Services                                        100000 non-null  float64
 17  Number of Medicare Beneficiaries                          100000 non-null  float64
 18  Number of Distinct Medicare Beneficiary/Per Day Services  100000 non-null  float64
 19  Average Medicare Allowed Amount                           100000 non-null  float64
 20  Average Submitted Charge Amount                           100000 non-null  float64
 21  Average Medicare Payment Amount                           100000 non-null  float64
 22  Average Medicare Standardized Amount                      100000 non-null  float64
dtypes: float64(8), int64(1), object(14)
memory usage: 17.5+ MB
```

inference

- As we can see above, Now as a result last seven columns have float datatype.
- These columns are one which are having numerical values.

Handling Missing values in the dataset

```
# @title Handling Missing values in the dataset
# check for missing values
df.isnull().sum()
```

```
National Provider Identifier                                   0
Name of the Provider                                           0
Credentials of the Provider                                 7209
Gender of the Provider                                      4254
Entity Type of the Provider                                    0
Street Address of the Provider                                 0
City of the Provider                                           0
Zip Code of the Provider                                       0
State Code of the Provider                                     0
Country Code of the Provider                                   0
Provider Type                                                  0
Medicare Participation Indicator                               0
Place of Service                                               0
HCPCS Code                                                     0
HCPCS Description                                              0
HCPCS Drug Indicator                                           0
Number of Services                                             0
Number of Medicare Beneficiaries                               0
Number of Distinct Medicare Beneficiary/Per Day Services       0
Average Medicare Allowed Amount                                0
Average Submitted Charge Amount                                0
Average Medicare Payment Amount                                0
Average Medicare Standardized Amount                           0
dtype: int64
```

```
df["Credentials of the Provider"] = df["Credentials of the Provider"].fillna(df["Credentials of th
df["Gender of the Provider"] = df["Gender of the Provider"].fillna(df["Gender of the Provider"].mc
```

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

```
National Provider Identifier                                0
Name of the Provider                                        0
Credentials of the Provider                                 0
Gender of the Provider                                      0
Entity Type of the Provider                                 0
Street Address of the Provider                              0
City of the Provider                                        0
Zip Code of the Provider                                    0
State Code of the Provider                                  0
Country Code of the Provider                                0
Provider Type                                               0
Medicare Participation Indicator                            0
Place of Service                                            0
HCPCS Code                                                  0
HCPCS Description                                           0
HCPCS Drug Indicator                                        0
Number of Services                                          0
Number of Medicare Beneficiaries                            0
Number of Distinct Medicare Beneficiary/Per Day Services    0
Average Medicare Allowed Amount                             0
Average Submitted Charge Amount                             0
Average Medicare Payment Amount                             0
Average Medicare Standardized Amount                    0
dtype: int64
```

Inference

- Now there is no missing value in each feature.
- To handle this we have use fillna() function and filled the null values with the mode of the perticular feature.

Exploratory data analysis

Visualization of categorical features - Univariate Analysis

```
df.head()
```

	National Provider Identifier	Name of the Provider	Credentials of the Provider	Gender of the Provider	Entity Type of the Provider	Street Address of the Provider	City of t Provic
0	1891106191	SATYASREE UPADHYAYULA	MD	F	I	1402 S GRAND BLVDFDT 14TH FLOOR	SAINT LO
1	1346202256	WENDYP JONES	MD	F	I	2950 VILLAGE DR	FAYETTEVIL
2	1306820956	RICHARDW DUROCHER	DPM	M	I	20 WASHINGTON AVESTE 212	NORTH HAV
3	1770523540	JASPER FULLARD	MD	M	I	5746 N BROADWAY ST	KANSAS C
4	1073627758	ANTHONYE PERROTTI	DO	M	I	875 MILITARY TRLSUITE 200	JUPIT

5 rows × 23 columns

```
df.columns
```

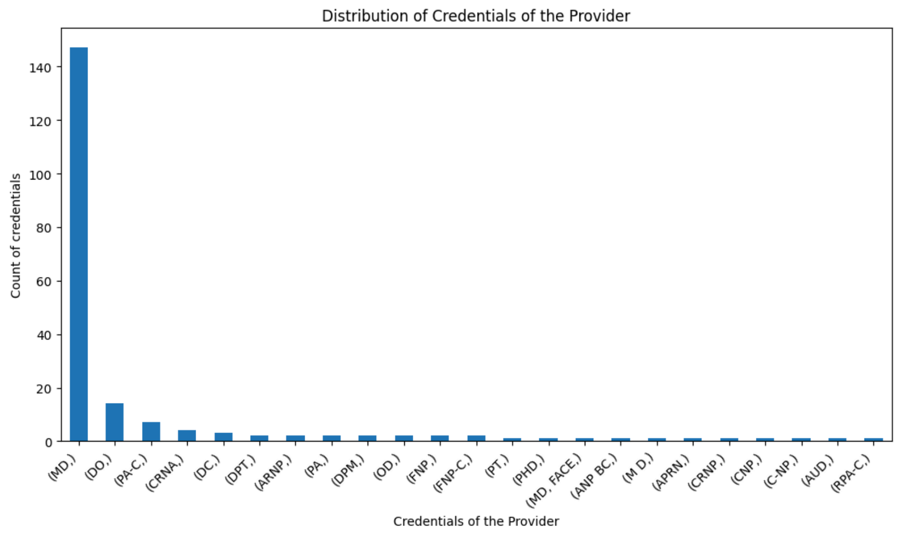
```
Index(['National Provider Identifier', 'Name of the Provider',
       'Credentials of the Provider', 'Gender of the Provider',
       'Entity Type of the Provider', 'Street Address of the Provider',
       'City of the Provider', 'Zip Code of the Provider',
       'State Code of the Provider', 'Country Code of the Provider',
       'Provider Type', 'Medicare Participation Indicator', 'Place of Service',
       'HCPCS Code', 'HCPCS Description', 'HCPCS Drug Indicator',
       'Number of Services', 'Number of Medicare Beneficiaries',
       'Number of Distinct Medicare Beneficiary/Per Day Services',
       'Average Medicare Allowed Amount', 'Average Submitted Charge Amount',
       'Average Medicare Payment Amount',
       'Average Medicare Standardized Amount'],
      dtype='object')
```

Credential of the provider

```
# @title Credential of the provider

Credential_counts = df.iloc[:200,2:3].value_counts()

# Plot the bar chart
Credential_counts.plot(kind='bar', figsize=(12, 6))
plt.title('Distribution of Credentials of the Provider')
plt.xlabel('Credentials of the Provider')
plt.ylabel('Count of credentials')
_ = plt.xticks(rotation=45, ha='right')
```



Inference

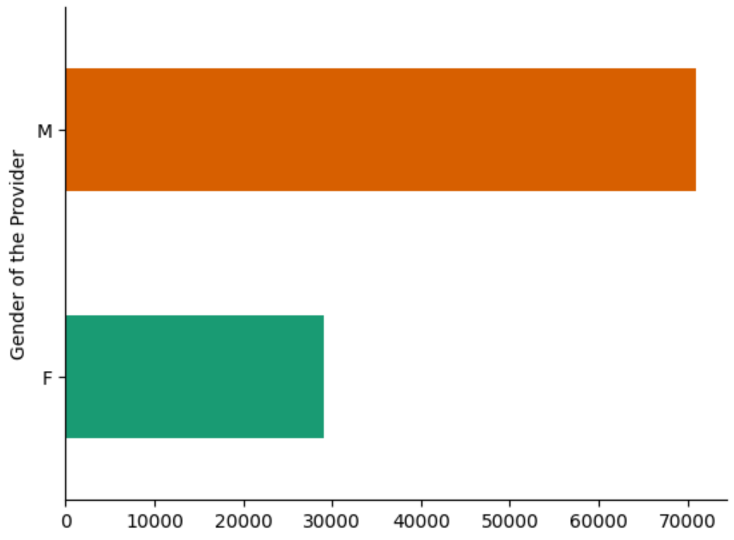
- We have plot this graph using top 200 rows of the feature.
- WE found that 'MD' credential has the highest count.

Gender of the Provider

```
# @title Gender of the Provider
print(df.groupby('Gender of the Provider').size())

df.groupby('Gender of the Provider').size().plot(kind='barh', color=sns.palettes.mpl_palette('Dark
plt.gca().spines[['top', 'right',]].set_visible(False)
```

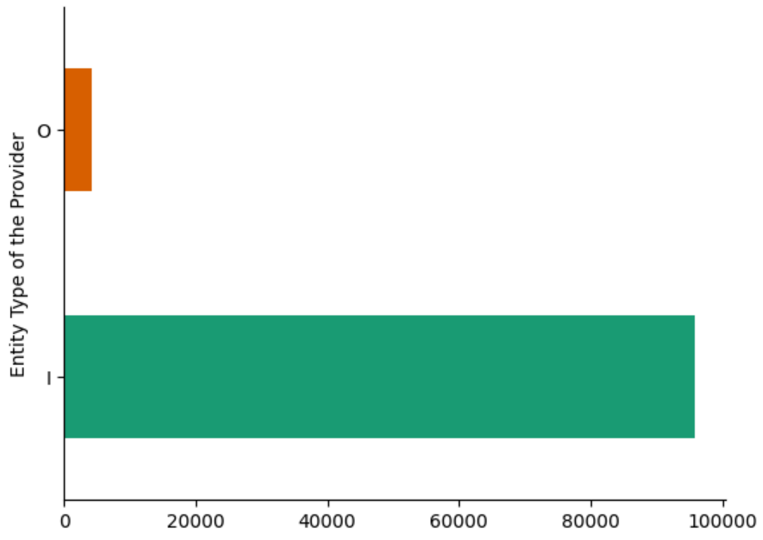
```
Gender of the Provider
F    29105
M    70895
dtype: int64
```



Entity Type of the Provider

```
# @title Entity Type of the Provider
print(df.groupby('Entity Type of the Provider').size())
df.groupby('Entity Type of the Provider').size().plot(kind='barh', color=sns.palettes.mpl_palette(
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
Entity Type of the Provider
I    95746
O     4254
dtype: int64
```



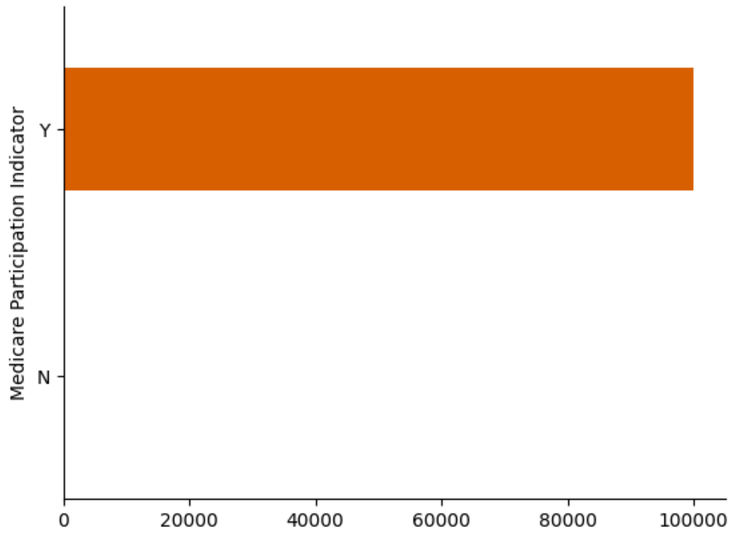
- This graph shows that individual entity of the providers are more than the the organizational providers.
- The above plot shows that the male providers is more than the female providers.

Medicare Participation Indicator

```
# @title Medicare Participation Indicator
print(df.groupby('Medicare Participation Indicator').size())

df.groupby('Medicare Participation Indicator').size().plot(kind='barh', color=sns.palettes.mpl_pal
plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
Medicare Participation Indicator
N       31
Y    99969
dtype: int64
```



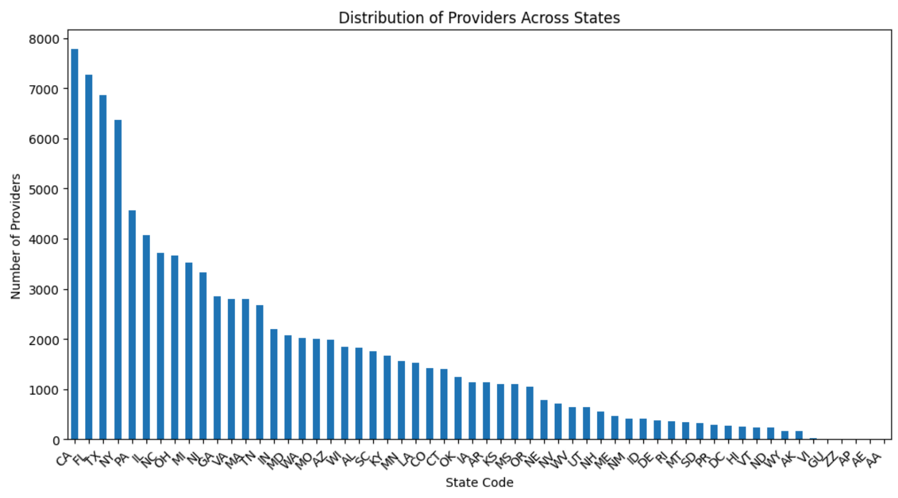
- **medicare_participation_indicator** - Identifies whether the provider participates in Medicare and/or accepts the assigned assignment of Medicare allowed amounts.
- According to the graph there is a less quantity of providers which does not participate in medicare.

Distribution of Providers Across States

```
# @title Distribution of Providers Across States

# Count the number of providers in each state
state_counts = df['State Code of the Provider'].value_counts()

# Plot the bar chart
state_counts.plot(kind='bar', figsize=(12, 6))
plt.title('Distribution of Providers Across States')
plt.xlabel('State Code')
plt.ylabel('Number of Providers')
_ = plt.xticks(rotation=45, ha='right')
```

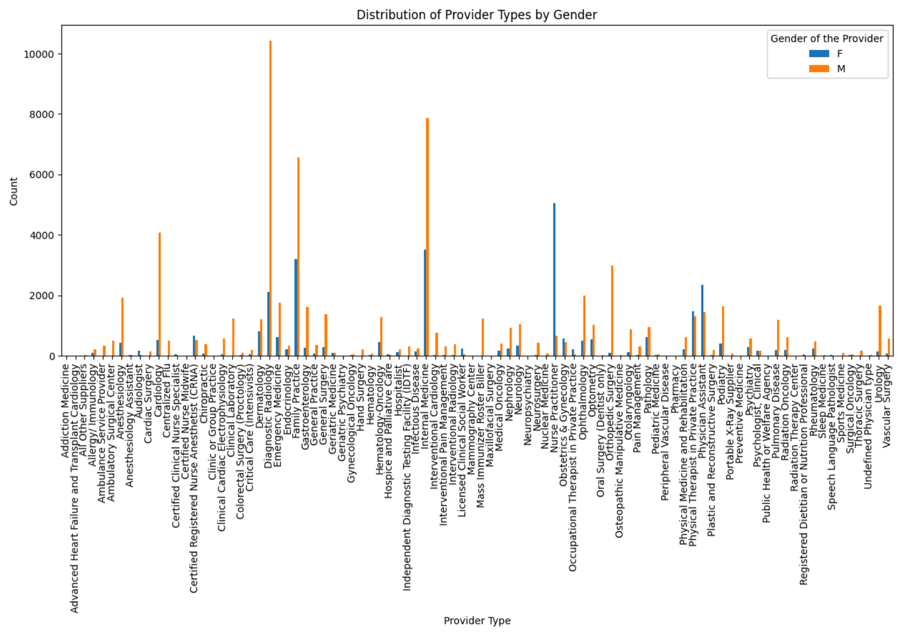


Bivariate Analysis

Distribution of Provider Types by Gender

```
# @title Distribution of Provider Types by Gender
# Group the data by 'Provider Type' and 'Gender of the Provider'
provider_gender = df.groupby(['Provider Type', 'Gender of the Provider'])['National Provider Ident

# Plot the grouped bar chart
provider_gender.plot(kind='bar', figsize=(15, 6))
plt.title('Distribution of Provider Types by Gender')
plt.xlabel('Provider Type')
plt.ylabel('Count')
_ = plt.xticks(rotation=90)
```



```
df['Provider Type'].nunique()
```

```
90
```

Inference

- Top 3 Male providers are **Diagnostic Radiology, Family Practice and Interna Medicine**.
- Top 3 Female providers are **Nurse Practitioner, Internal Medicine and Family practice**.
- There are 90 unique Providers in the dataset.

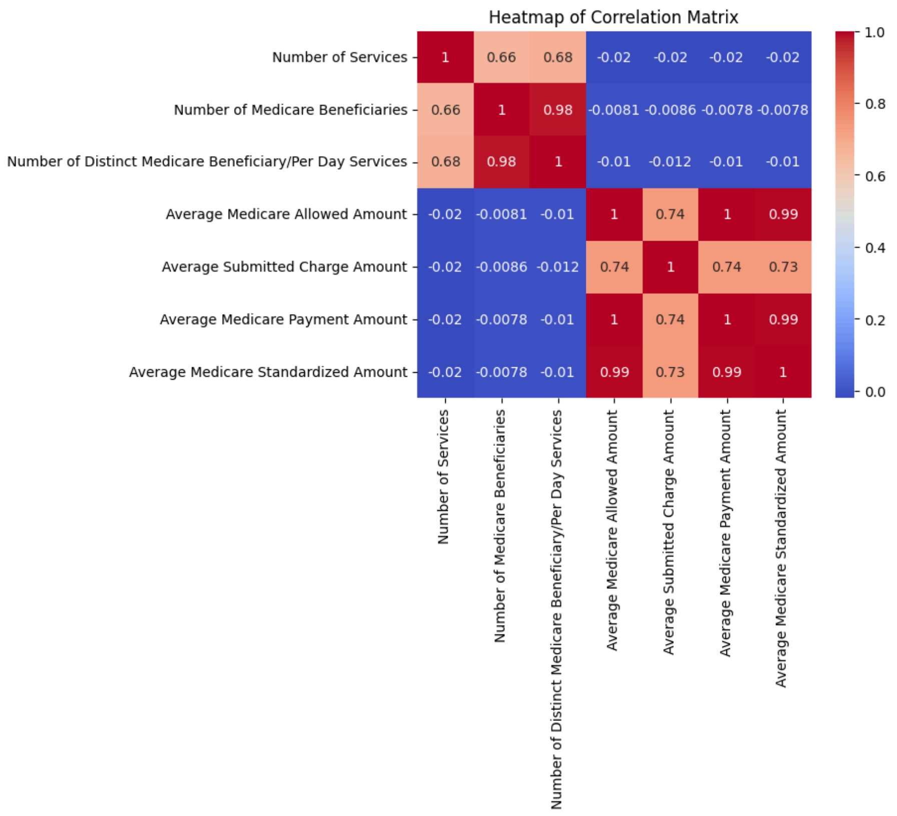
Visualization for numerical columns

```
# @title Visualization for numerical columns
```

Heatmap for the numerical columns

```
# @title Heatmap for the numerical columns
corr_matrix = df[['Number of Services', 'Number of Medicare Beneficiaries',
       'Number of Distinct Medicare Beneficiary/Per Day Services',
       'Average Medicare Allowed Amount', 'Average Submitted Charge Amount',
       'Average Medicare Payment Amount',
       'Average Medicare Standardized Amount']].corr()

# Heatmap
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Heatmap of Correlation Matrix')
plt.show()
```



	Number of Services	Number of Medicare Beneficiaries	Number of Distinct Medicare Beneficiary/Per Day Services	Average Medicare Allowed Amount	Average Submitted Charge Amount	Average Medicare Payment Amount	Average Medicare Standardized Amount
Number of Services	1	0.66	0.68	-0.02	-0.02	-0.02	-0.02
Number of Medicare Beneficiaries	0.66	1	0.98	-0.0081	-0.0086	-0.0078	-0.0078
Number of Distinct Medicare Beneficiary/Per Day Services	0.68	0.98	1	-0.01	-0.012	-0.01	-0.01
Average Medicare Allowed Amount	-0.02	-0.0081	-0.01	1	0.74	1	0.99
Average Submitted Charge Amount	-0.02	-0.0086	-0.012	0.74	1	0.74	0.73
Average Medicare Payment Amount	-0.02	-0.0078	-0.01	1	0.74	1	0.99
Average Medicare Standardized Amount	-0.02	-0.0078	-0.01	0.99	0.73	0.99	1

Inference

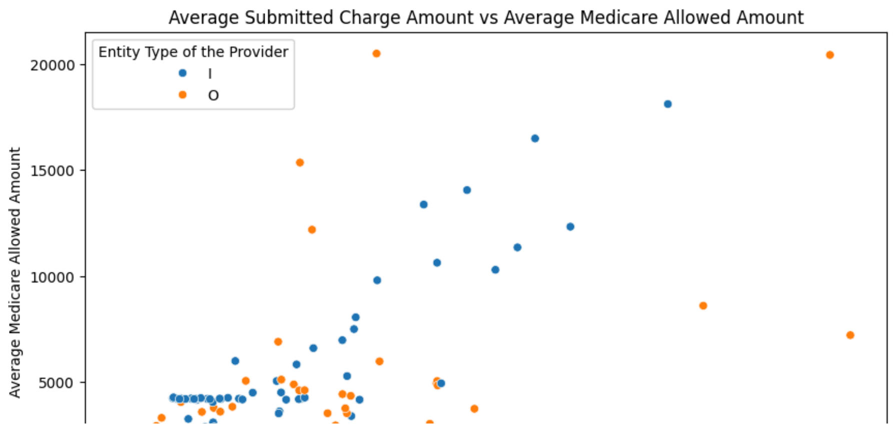
- Above heatmap is plotted using the numerical columns.
- This shows the correlation between different features.
- Some of the features are highly correlated and many of them are slightly correlated.

```
df.columns
```

```
Index(['National Provider Identifier', 'Name of the Provider',
       'Credentials of the Provider', 'Gender of the Provider',
       'Entity Type of the Provider', 'Street Address of the Provider',
       'City of the Provider', 'Zip Code of the Provider',
       'State Code of the Provider', 'Country Code of the Provider',
       'Provider Type', 'Medicare Participation Indicator', 'Place of Service',
       'HCPCS Code', 'HCPCS Description', 'HCPCS Drug Indicator',
       'Number of Services', 'Number of Medicare Beneficiaries',
       'Number of Distinct Medicare Beneficiary/Per Day Services',
       'Average Medicare Allowed Amount', 'Average Submitted Charge Amount',
       'Average Medicare Payment Amount',
       'Average Medicare Standardized Amount'],
      dtype='object')
```

```
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df,x='Average Submitted Charge Amount',y='Average Medicare Allowed Amount',
                hue='Entity Type of the Provider')
plt.title('Average Submitted Charge Amount vs Average Medicare Allowed Amount' )
```

```
Text(0.5, 1.0, 'Average Submitted Charge Amount vs Average Medicare Allowed Amount')
```


Average Submitted Charge Amount vs Average Medicare Allowed Amount
Entity Type of the Provider
● I
● O
Average Medicare Allowed Amount