

✓ PROJECT - **Unsupervised Anomaly Detection**

DATASET - **Healthcare Providers Data For Anomaly Detection**

NAME - **Shrikar Gaikar**

Mount Google Drive

```
# Mounting Google Drive to access the dataset
from google.colab import drive
drive.mount('/content/drive')
```



Mounted at /content/drive

Import Libraries

```
# Importing necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Load Dataset

```
# Loading the dataset
data = pd.read_csv("/content/drive/MyDrive/Datasets/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	Gender of the Provider	Entity Type of the Provider	Street Address 1 of the Provider	Street Address 2 of the Provider	...
0	8774979	1891106191	UPADHYAYULA	SATYASREE	NaN	M.D.	F	I	1402 S GRAND BLVD	FDT 14TH FLOOR	...
1	3354385	1346202256	JONES	WENDY	P	M.D.	F	I	2950 VILLAGE DR	NaN	...
2	3001884	1306820956	DUROCHER	RICHARD	W	DPM	M	I	20 WASHINGTON AVE	STE 212	...
3	7594822	1770523540	FULLARD	JASPER	NaN	MD	M	I	5746 N BROADWAY ST	NaN	...
4	746159	1073627758	PERROTTI	ANTHONY	E	DO	M	I	875 MILITARY TRL	SUITE 200	...

5 rows × 27 columns

Descriptive Statistics

```
# Displaying descriptive statistics
data.describe()
```



	index	National Provider Identifier	Zip Code of the Provider
count	1.000000e+05	1.000000e+05	1.000000e+05
mean	4.907646e+06	1.498227e+09	4.163820e+08
std	2.839633e+06	2.874125e+08	3.082566e+08
min	2.090000e+02	1.003001e+09	6.010000e+02
25%	2.458791e+06	1.245669e+09	1.426300e+08
50%	4.901266e+06	1.497847e+09	3.633025e+08
75%	7.349450e+06	1.740374e+09	6.819881e+08
max	9.847440e+06	1.993000e+09	9.990166e+08



Dataset Information

```
# Displaying information about the dataset
data.info()
```



```
<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

Convert Object Columns to Numeric Type

```
# Converting object columns to numeric type where necessary
numeric_columns = [
    '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'
]
```

```
for column in numeric_columns:
    data[column] = pd.to_numeric(data[column], errors='coerce')
```

```
# Verifying data types after conversion
data.info()
```



```
<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                        97347 non-null   float64
21   Number of Medicare Beneficiaries          99595 non-null   float64
22   Number of Distinct Medicare Beneficiary/Per Day Services  98500 non-null   float64
23   Average Medicare Allowed Amount           99255 non-null   float64
24   Average Submitted Charge Amount           93277 non-null   float64
25   Average Medicare Payment Amount           99534 non-null   float64
26   Average Medicare Standardized Amount      99530 non-null   float64
dtypes: float64(8), int64(2), object(17)
memory usage: 20.6+ MB
```

Handle Missing Values

```
# Checking for missing values and imputing them with the mean
```

```
print(data.isnull().sum())
```

```
# Imputation of missing values with mean
```

```
data[numeric_columns] = data[numeric_columns].fillna(data[numeric_columns].mean())
```

```
# Checking for missing values post imputation
```

```
print(data.isnull().sum())
```



index	0
National Provider Identifier	0
Last Name/Organization Name of the Provider	0
First Name of the Provider	4255
Middle Initial of the Provider	29331
Credentials of the Provider	7209
Gender of the Provider	4254
Entity Type of the Provider	0
Street Address 1 of the Provider	0
Street Address 2 of the Provider	59363
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	2653
Number of Medicare Beneficiaries	405
Number of Distinct Medicare Beneficiary/Per Day Services	1500
Average Medicare Allowed Amount	745
Average Submitted Charge Amount	6723
Average Medicare Payment Amount	466
Average Medicare Standardized Amount	470
dtype: int64	
index	0
National Provider Identifier	0
Last Name/Organization Name of the Provider	0
First Name of the Provider	4255
Middle Initial of the Provider	29331
Credentials of the Provider	7209
Gender of the Provider	4254
Entity Type of the Provider	0
Street Address 1 of the Provider	0
Street Address 2 of the Provider	59363
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
```

Check for Duplicate Values

```
# Checking for duplicate values
print(data.duplicated().sum())
```

 0

Merge Name Columns

```
# Merging the name columns into a single column
data['Full Name'] = data['First Name of the Provider'].fillna('') + ' ' + \
    data['Middle Initial of the Provider'].fillna('') + ' ' + \
    data['Last Name/Organization Name of the Provider'].fillna('')
data['Full Name'] = data['Full Name'].str.strip()

# Dropping the original name columns
data = data.drop(columns=['Last Name/Organization Name of the Provider',
    'First Name of the Provider',
    'Middle Initial of the Provider'])

data.head()
```



	index	National Provider Identifier	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	...	Des
0	8774979	1891106191	M.D.	F	I	1402 S GRAND BLVD	FDT 14TH FLOOR	SAINT LOUIS	631041004.0	MO	...	Initi inpa typi
1	3354385	1346202256	M.D.	F	I	2950 VILLAGE DR	NaN	FAYETTEVILLE	283043815.0	NC	...	mami b vie
2	3001884	1306820956	DPM	M	I	20 WASHINGTON AVE	STE 212	NORTH HAVEN	64732343.0	CT	...	E pat visi
3	7594822	1770523540	MD	M	I	5746 N BROADWAY ST	NaN	KANSAS CITY	641183998.0	MO	...	I m
4	746159	1073627758	DO	M	I	875 MILITARY TRL	SUITE 200	JUPITER	334585700.0	FL	...	be s mu

5 rows × 25 columns

Merge Address Columns

```
# Merging the address columns into a single column
data['Full Address'] = data['Street Address 1 of the Provider'].fillna('') + ' ' + \
                        data['Street Address 2 of the Provider'].fillna('')
data['Full Address'] = data['Full Address'].str.strip()

# Dropping the original address columns
data = data.drop(columns=['Street Address 1 of the Provider',
                        'Street Address 2 of the Provider'])

data.head()
```



	index	National Provider Identifier	Credentials of the Provider	Gender of the Provider	Entity Type of the Provider	City of the Provider	Zip Code of the Provider	State Code of the Provider	Country Code of the Provider	Provider Type	...	HCP Dr Indicati
0	8774979	1891106191	M.D.	F	I	SAINT LOUIS	631041004.0	MO	US	Internal Medicine	...	
1	3354385	1346202256	M.D.	F	I	FAYETTEVILLE	283043815.0	NC	US	Obstetrics & Gynecology	...	
2	3001884	1306820956	DPM	M	I	NORTH HAVEN	64732343.0	CT	US	Podiatry	...	
3	7594822	1770523540	MD	M	I	KANSAS CITY	641183998.0	MO	US	Internal Medicine	...	
4	746159	1073627758	DO	M	I	JUPITER	334585700.0	FL	US	Internal Medicine	...	

5 rows × 24 columns

Standardize Credentials Column

```
# Standardizing the credentials column
data['Credentials of the Provider'] = data['Credentials of the Provider'].str.replace(r'\.', '', regex=True).str.upper()

data.head()
```




	index	National Provider Identifier	Credentials of the Provider	Gender of the Provider	Entity Type of the Provider	City of the Provider	Zip Code of the Provider	State Code of the Provider	Country Code of the Provider	Provider Type	...	HCP Dr Indicati
0	8774979	1891106191	MD	F	I	SAINT LOUIS	631041004.0	MO	US	Internal Medicine	...	
1	3354385	1346202256	MD	F	I	FAYETTEVILLE	283043815.0	NC	US	Obstetrics & Gynecology	...	
2	3001884	1306820956	DPM	M	I	NORTH HAVEN	64732343.0	CT	US	Podiatry	...	
3	7594822	1770523540	MD	M	I	KANSAS CITY	641183998.0	MO	US	Internal Medicine	...	
4	746159	1073627758	DO	M	I	JUPITER	334585700.0	FL	US	Internal Medicine	...	

5 rows × 24 columns

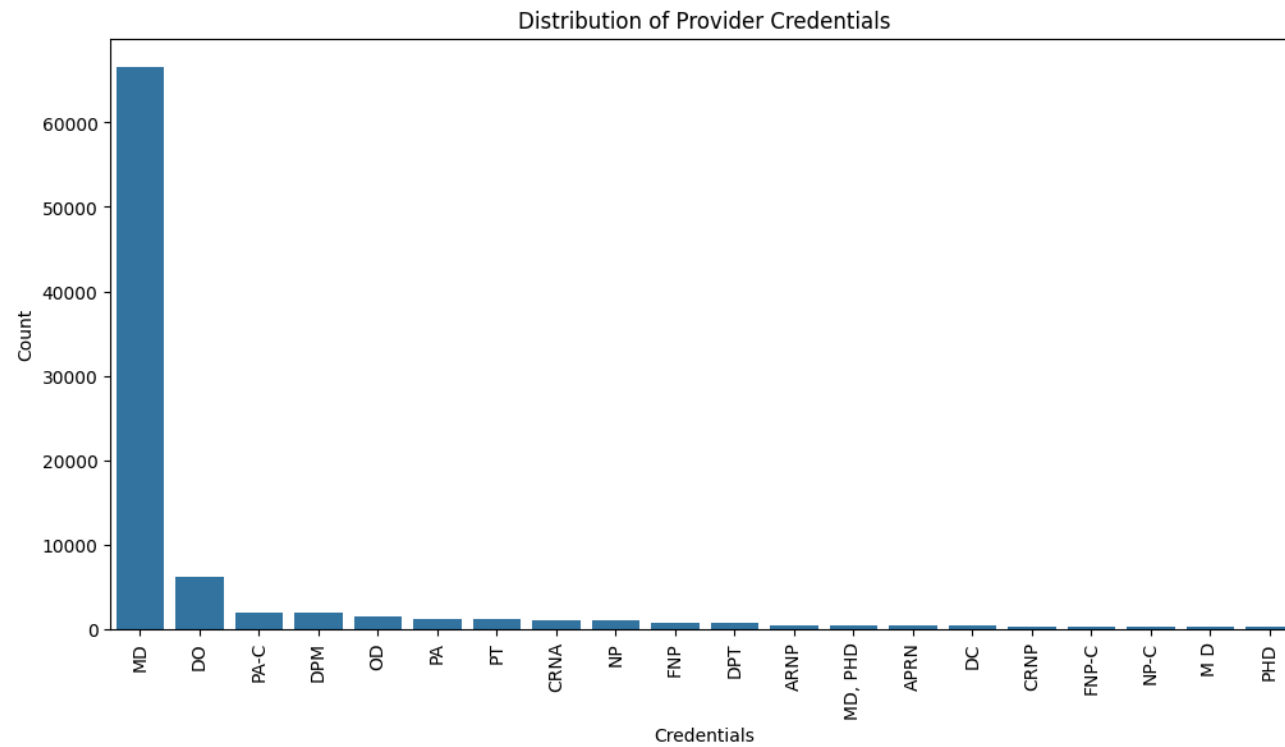
▼ Exploratory Data Analysis (EDA)

▼ 1. Univariate Analysis

Distribution of Provider Credentials

```
# Plotting the distribution of provider credentials
credentials_counts = data['Credentials of the Provider'].value_counts().head(20)

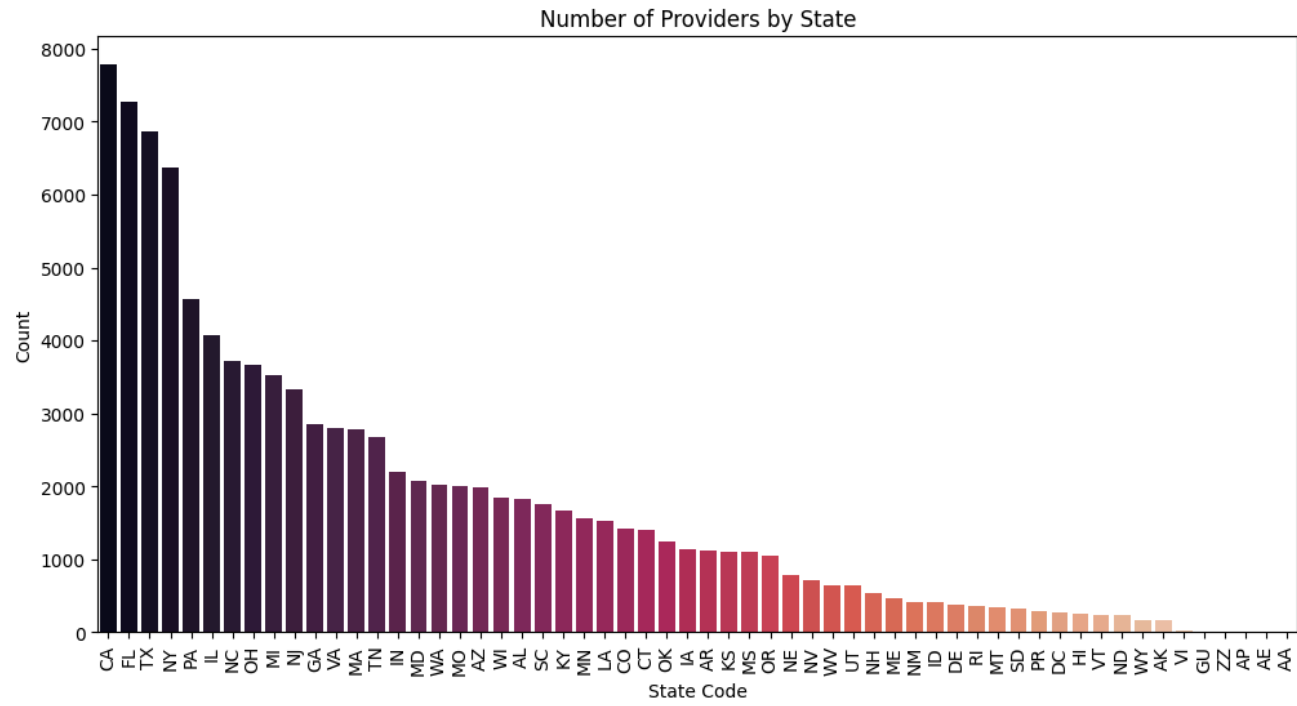
plt.figure(figsize=(12, 6))
sns.barplot(x=credentials_counts.index, y=credentials_counts.values)
plt.title('Distribution of Provider Credentials')
plt.xlabel('Credentials')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```



Number of Providers by State

```
# Plotting the number of providers by state
state_counts = data['State Code of the Provider'].value_counts()

plt.figure(figsize=(12, 6))
sns.barplot(x=state_counts.index, y=state_counts.values, palette='rocket')
plt.title('Number of Providers by State')
plt.xlabel('State Code')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```



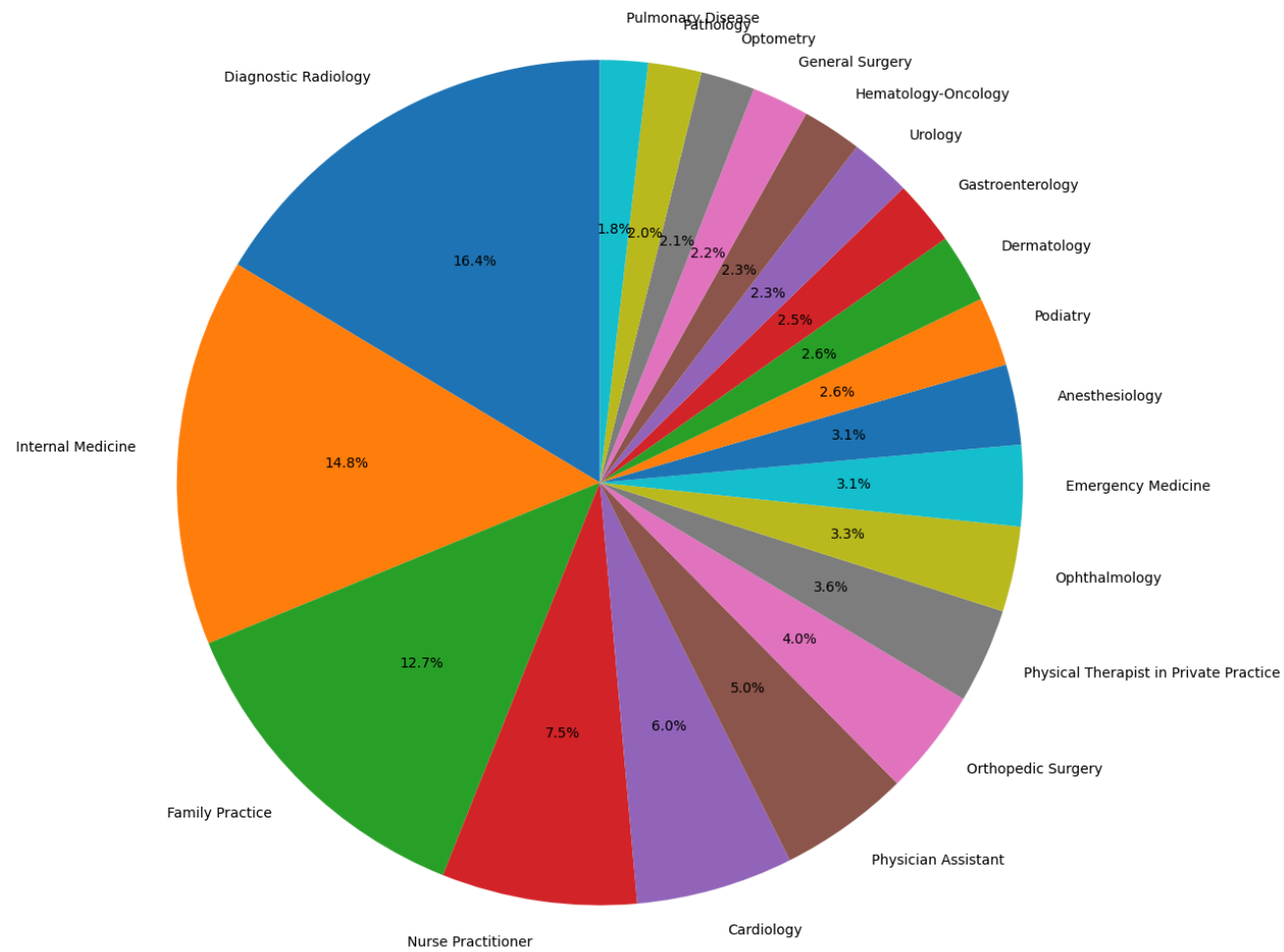
Distribution of Provider Types

```
# Plotting the distribution of provider types
provider_type_counts = data['Provider Type'].value_counts().head(20)

plt.figure(figsize=(12, 14))
plt.pie(provider_type_counts, labels=provider_type_counts.index, autopct='%1.1f%%', startangle=90)
plt.title('Distribution of Provider Types')
plt.axis('equal')
plt.show()
```



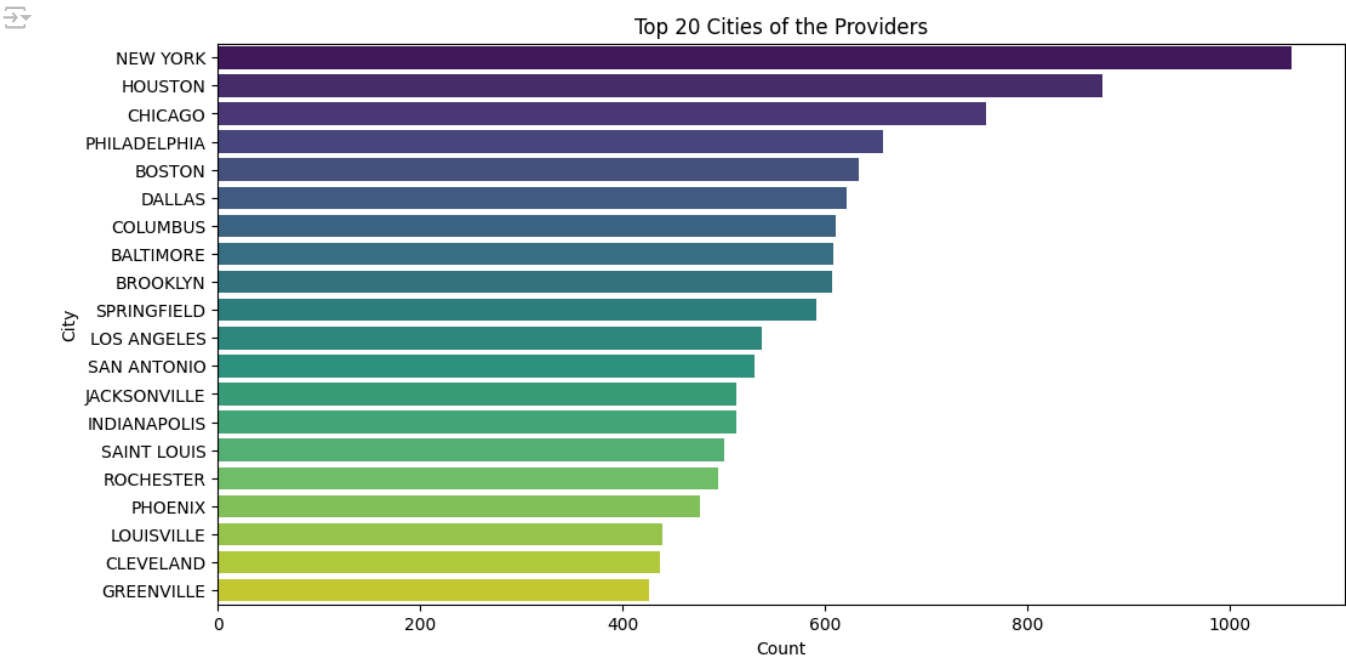
Distribution of Provider Types



Double-click (or enter) to edit

```
# Plotting the top 20 cities of the providers
city_counts = data['City of the Provider'].value_counts().head(20)

plt.figure(figsize=(12, 6))
sns.barplot(x=city_counts.values, y=city_counts.index, palette='viridis')
plt.title('Top 20 Cities of the Providers')
plt.xlabel('Count')
plt.ylabel('City')
plt.show()
```



Distribution of Numeric Columns

```
# Plotting the distribution of numeric columns
numeric_columns = [
    'Number of Services',
    'Average Medicare Allowed Amount',
    'Average Submitted Charge Amount',
    'Average Medicare Payment Amount'
]

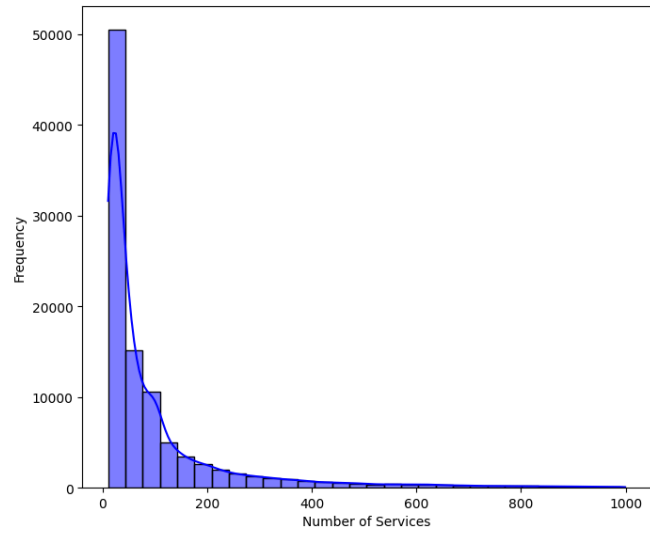
plt.figure(figsize=(14, 12))

for i, column in enumerate(numeric_columns, 1):
    plt.subplot(2, 2, i)
    sns.histplot(data[column].dropna(), bins=30, kde=True, color='blue')
    plt.title(f'Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')

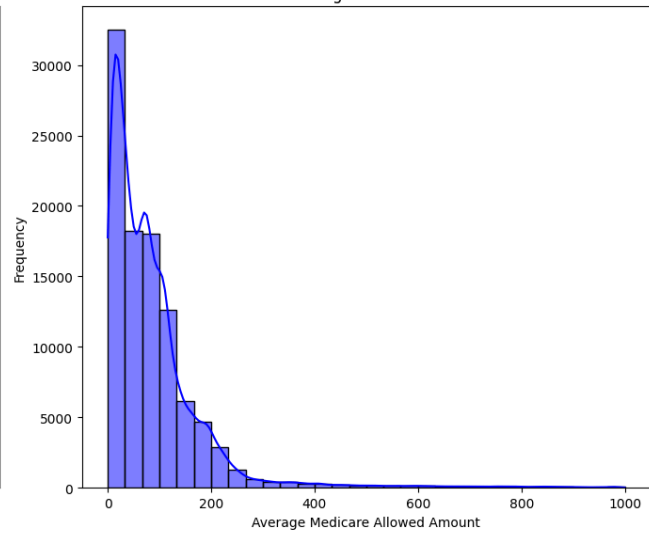
plt.tight_layout()
plt.show()
```



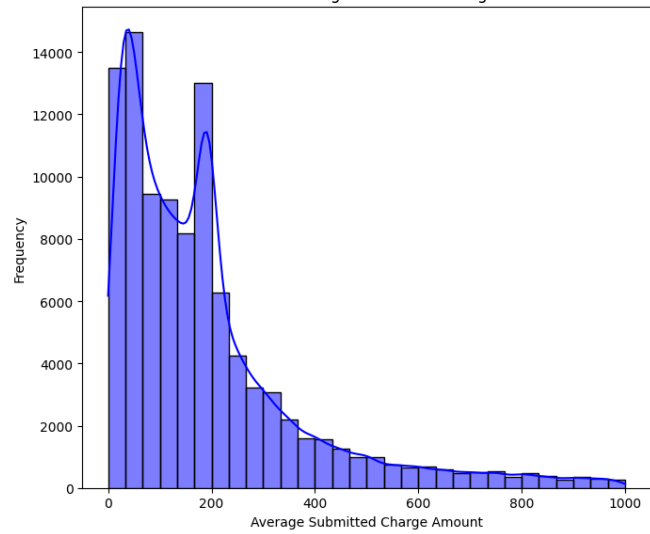
Distribution of Number of Services



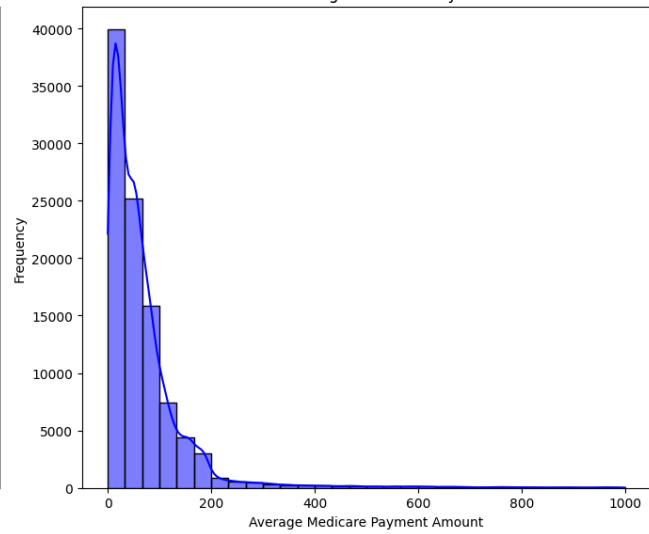
Distribution of Average Medicare Allowed Amount



Distribution of Average Submitted Charge Amount



Distribution of Average Medicare Payment Amount



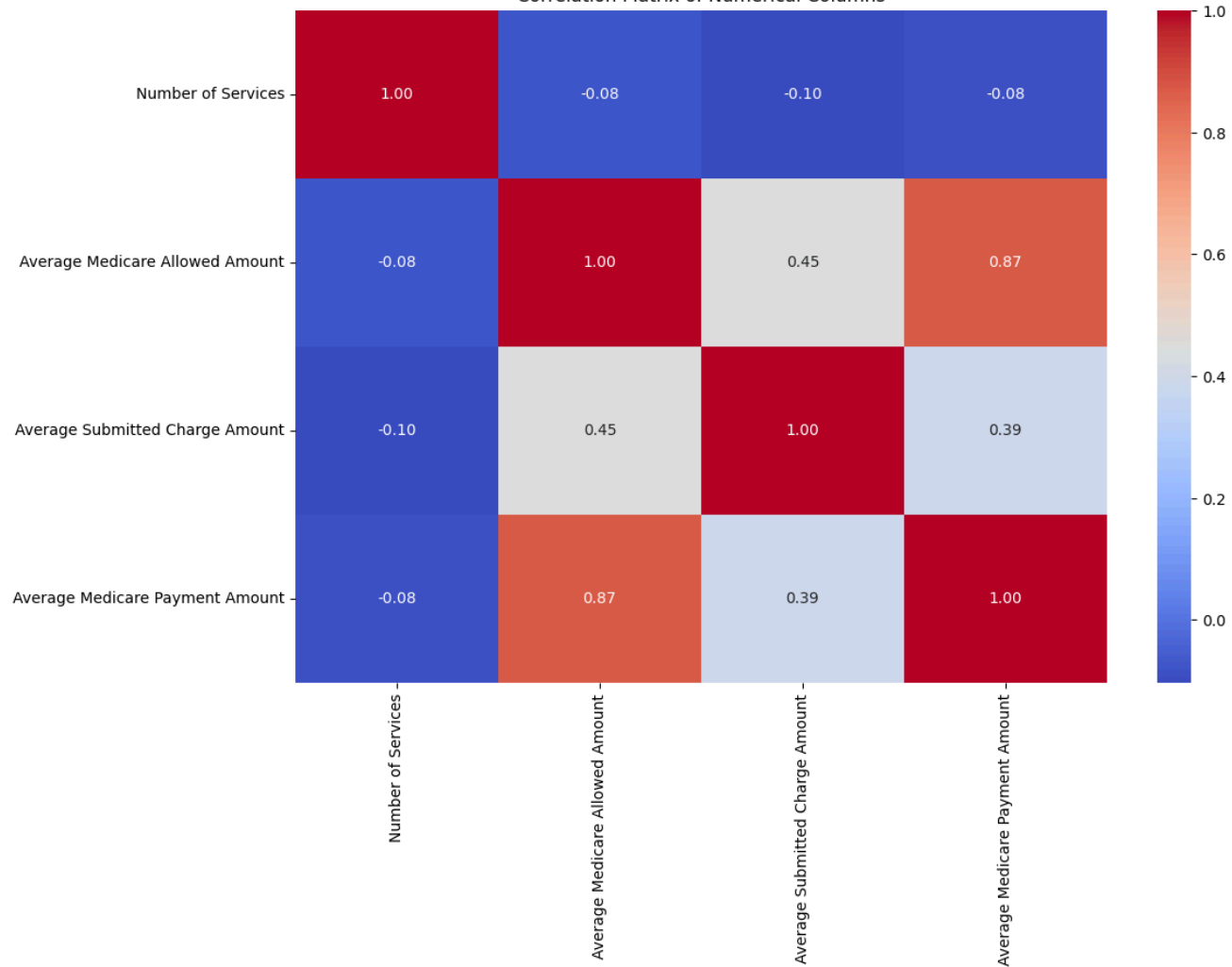
Correlation Matrix

```
# Plotting the correlation matrix of numerical columns
corr_matrix = data[numeric_columns].corr()

plt.figure(figsize=(12, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix of Numerical Columns')
plt.show()
```




Correlation Matrix of Numerical Columns

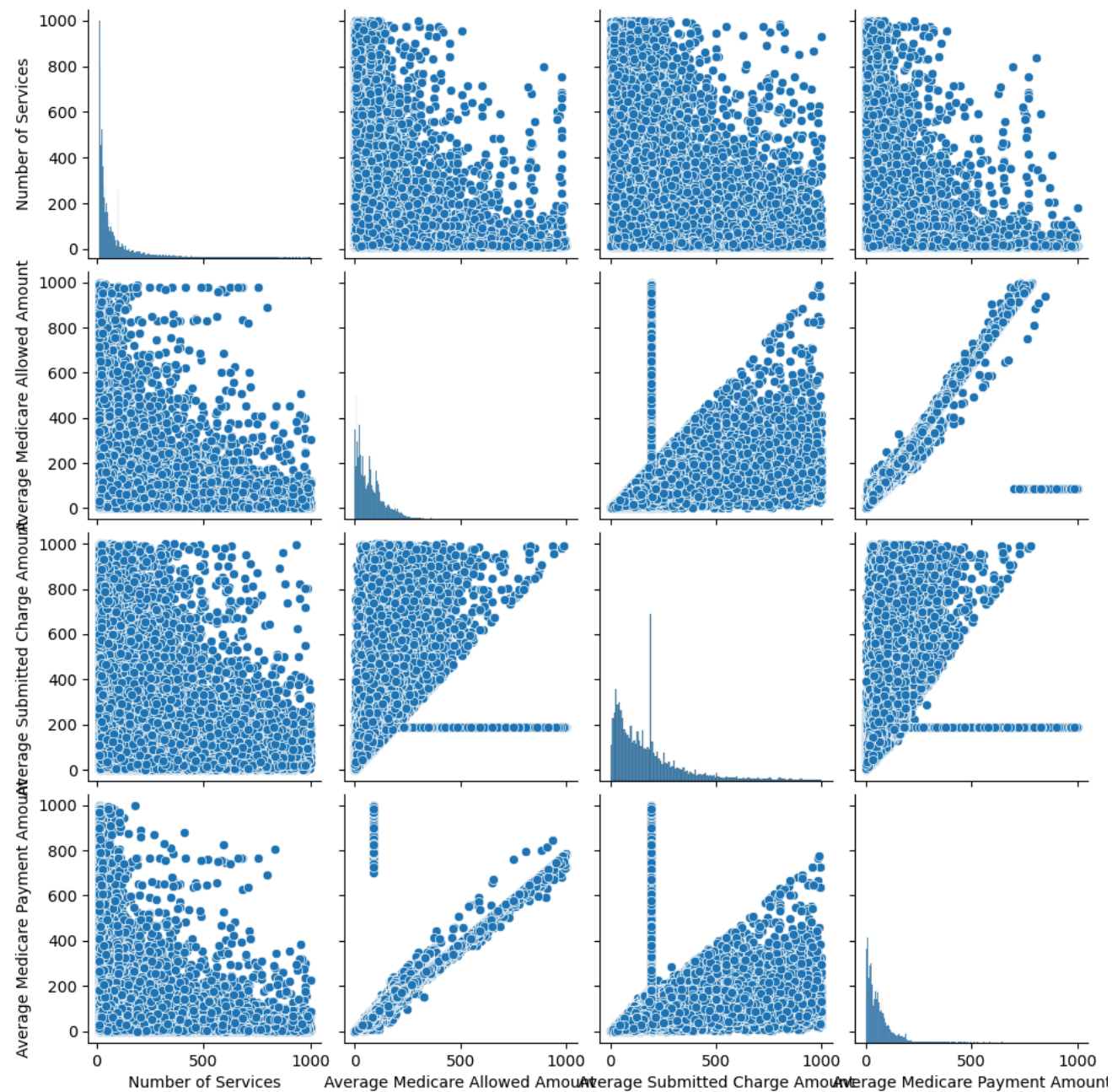


Pairplot of Numerical Variables

```
# Plotting the pairplot of numerical variables
sns.pairplot(data[numeric_columns])
plt.suptitle('Pairplot of Numerical Variables', y=1.02)
plt.show()
```



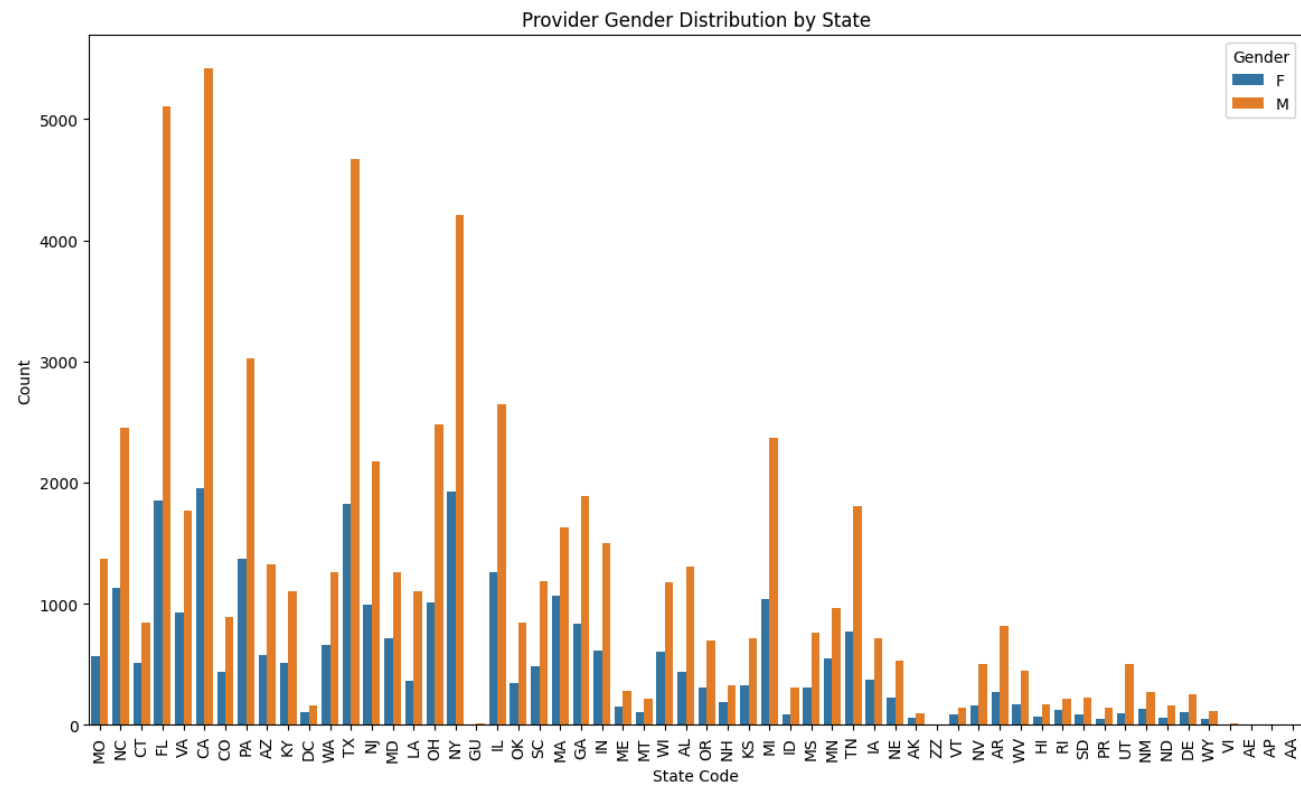
Pairplot of Numerical Variables



2. Bivariate Analysis

Provider Gender Distribution by State

```
# Countplot of provider gender distribution by state
plt.figure(figsize=(14, 8))
sns.countplot(x='State Code of the Provider', hue='Gender of the Provider', data=data)
plt.title('Provider Gender Distribution by State')
plt.xlabel('State Code')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.legend(title='Gender')
plt.show()
```

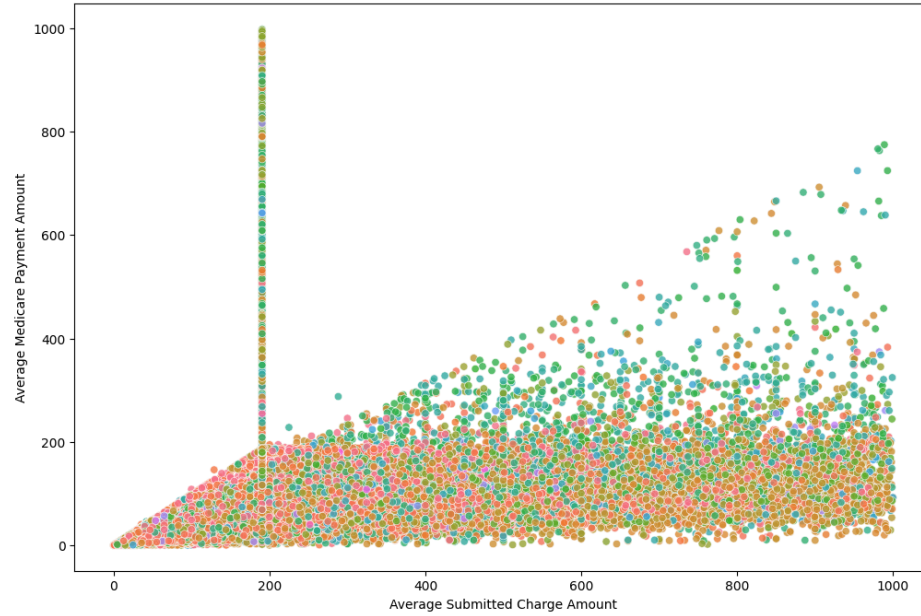


Average Submitted Charge vs. Average Payment

```
# Scatter plot of average submitted charge vs. average payment
plt.figure(figsize=(12, 8))
sns.scatterplot(x='Average Submitted Charge Amount', y='Average Medicare Payment Amount', data=data, hue='Provider Type', alpha=0.8)
plt.title('Average Submitted Charge Amount vs. Average Medicare Payment Amount')
plt.xlabel('Average Submitted Charge Amount')
plt.ylabel('Average Medicare Payment Amount')
plt.legend(title='Provider Type', loc='upper left', bbox_to_anchor=(1, 1))
plt.show()
```



Average Submitted Charge Amount vs. Average Medicare Payment Amount



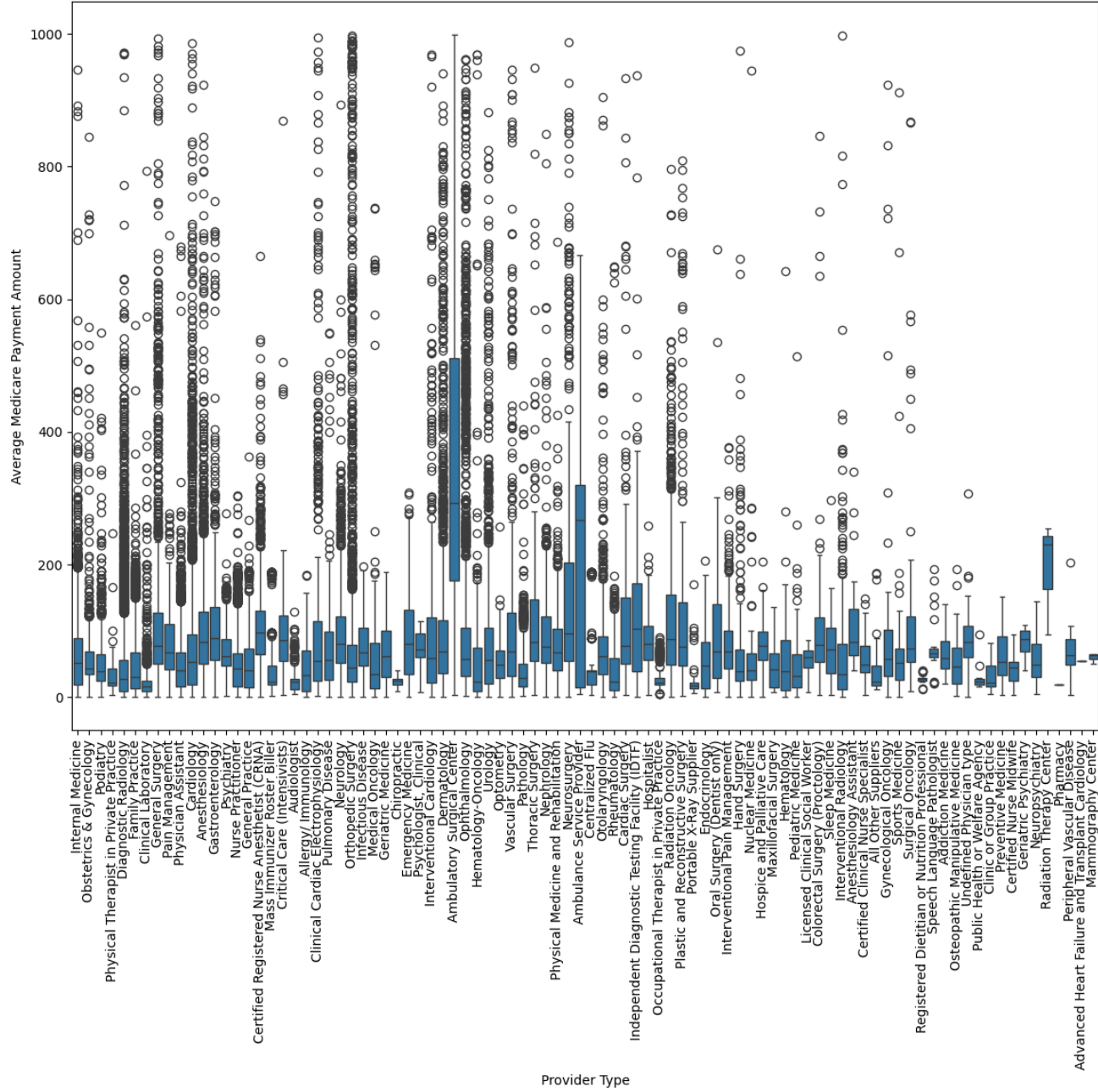
- Provider Type
- Internal Medicine
 - Obstetrics & Gynecology
 - Podiatry
 - Physical Therapist in Private Practice
 - Diagnostic Radiology
 - Family Practice
 - Clinical Laboratory
 - General Surgery
 - Pain Management
 - Physician Assistant
 - Cardiology
 - Anesthesiology
 - Gastroenterology
 - Psychiatry
 - Nurse Practitioner
 - General Practice
 - Certified Registered Nurse Anesthetist (CRNA)
 - Mass Immunizer Roster Biller
 - Critical Care (Intensivists)
 - Audiologist
 - Allergy/Immunology
 - Clinical Cardiac Electrophysiology
 - Pulmonary Disease
 - Neurology
 - Orthopedic Surgery
 - Infectious Disease
 - Medical Oncology
 - Geriatric Medicine
 - Chiropractic
 - Emergency Medicine
 - Psychologist, Clinical
 - Interventional Cardiology
 - Dermatology
 - Ambulatory Surgical Center
 - Ophthalmology
 - Hematology-Oncology
 - Urology
 - Optometry
 - Vascular Surgery
 - Pathology
 - Thoracic Surgery
 - Nephrology
 - Physical Medicine and Rehabilitation
 - Neurosurgery
 - Ambulance Service Provider
 - Centralized Flu
 - Otolaryngology
 - Rheumatology
 - Cardiac Surgery
 - Independent Diagnostic Testing Facility (IDTF)
 - Hospitalist
 - Occupational Therapist in Private Practice
 - Radiation Oncology
 - Plastic and Reconstructive Surgery
 - Portable X-Ray Supplier
 - Endocrinology
 - Oral Surgery (Dentist only)
 - Interventional Pain Management
 - Hand Surgery
 - Nuclear Medicine
 - Hospice and Palliative Care
 - Maxillofacial Surgery
 - Hematology
 - Pediatric Medicine
 - Licensed Clinical Social Worker
 - Colorectal Surgery (Proctology)
 - Sleep Medicine
 - Interventional Radiology
 - Anesthesiology Assistant
 - Certified Clinical Nurse Specialist
 - All Other Suppliers

- Gynecological Oncology
- Sports Medicine
- Surgical Oncology
- Registered Dietitian or Nutrition Professional
- Speech Language Pathologist
- Addiction Medicine
- Osteopathic Manipulative Medicine
- Undefined Physician type
- Public Health or Welfare Agency
- Clinic or Group Practice
- Preventive Medicine
- Certified Nurse Midwife
- Geriatric Psychiatry
- Neuropsychiatry
- Radiation Therapy Center
- Pharmacy
- Peripheral Vascular Disease
- Advanced Heart Failure and Transplant Cardiology
- Mammography Center

Average Medicare Payment Amount by Provider Type

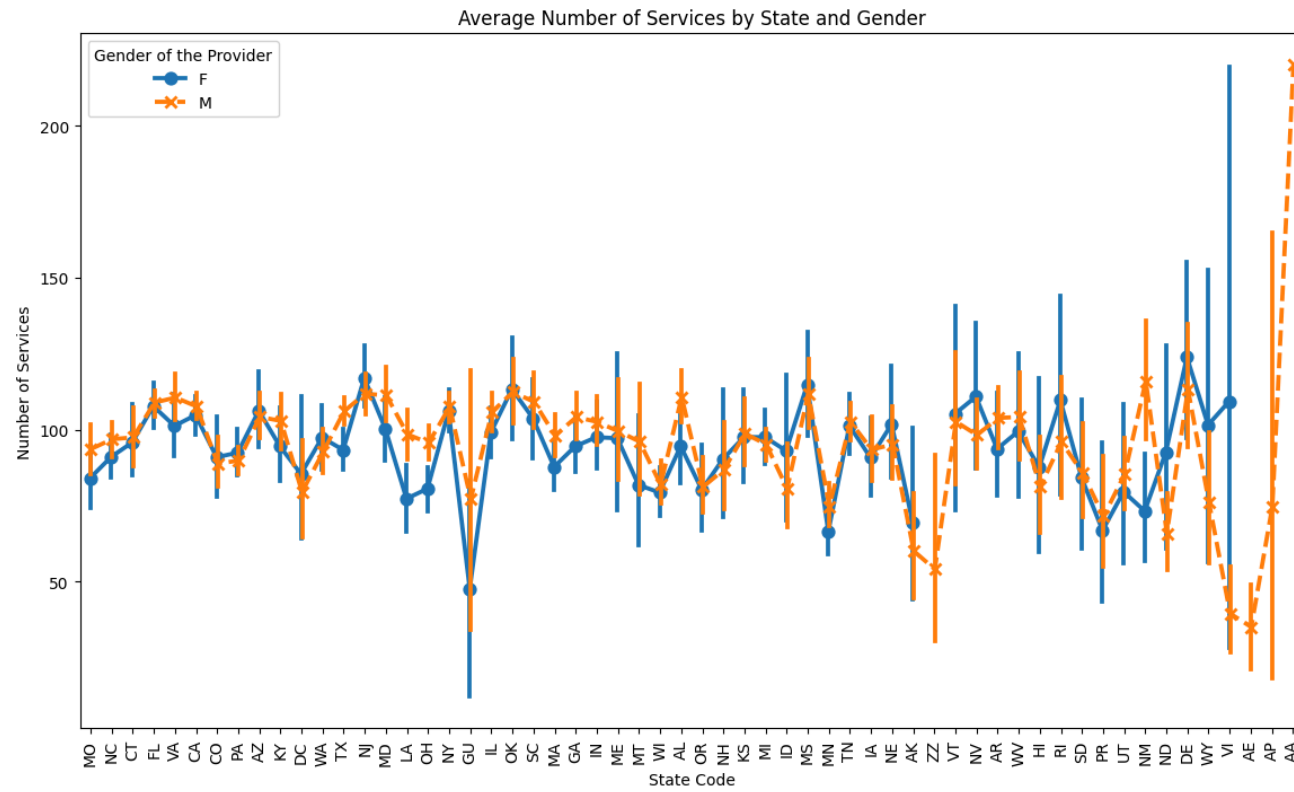
```
# Boxplot of average Medicare payment amount by provider type
plt.figure(figsize=(14, 10))
sns.boxplot(x='Provider Type', y='Average Medicare Payment Amount', data=data)
plt.title('Distribution of Average Medicare Payment Amount by Provider Type')
plt.xlabel('Provider Type')
plt.ylabel('Average Medicare Payment Amount')
plt.xticks(rotation=90)
plt.show()
```

Distribution of Average Medicare Payment Amount by Provider Type



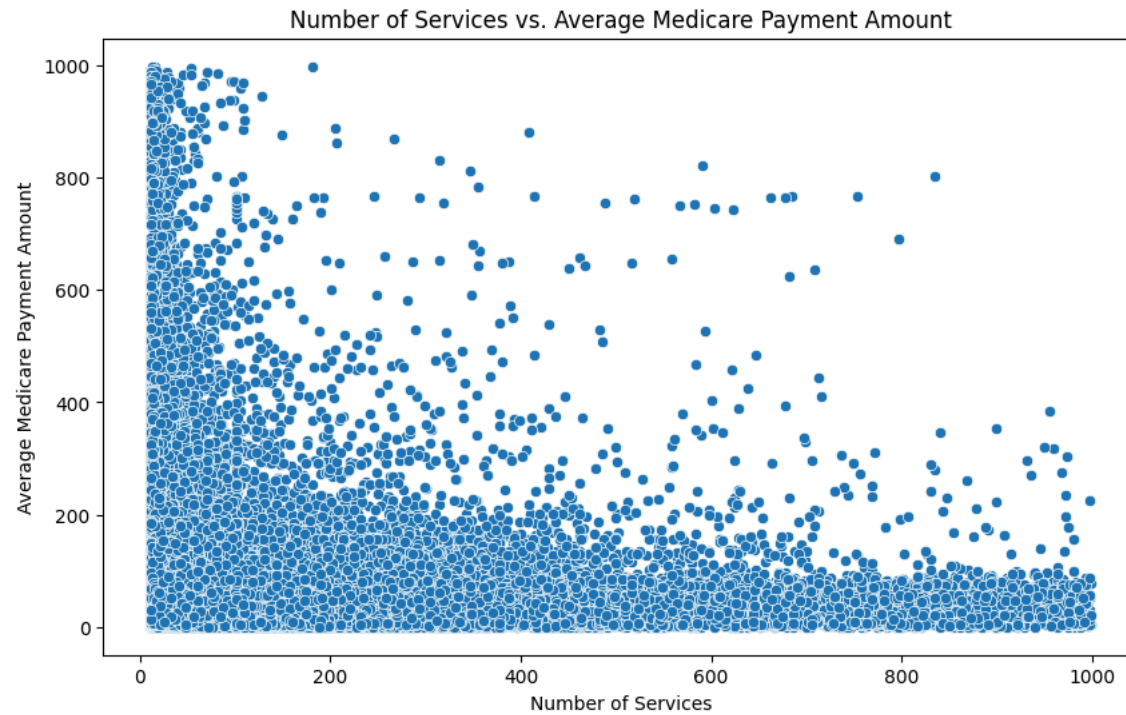
Average Number of Services by State and Gender

```
# Point plot of average number of services by state and gender
plt.figure(figsize=(14, 8))
sns.pointplot(x='State Code of the Provider', y='Number of Services', hue='Gender of the Provider', data=data, dodge=True, markers=["o", "x"], linestyle=["-", "--"])
plt.title('Average Number of Services by State and Gender')
plt.xlabel('State Code')
plt.ylabel('Number of Services')
plt.xticks(rotation=90)
plt.show()
```



Correlation between Number of Services and Payment Amounts

```
# Scatter plot of number of services vs. average Medicare payment amount
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Number of Services', y='Average Medicare Payment Amount', data=data)
plt.title('Number of Services vs. Average Medicare Payment Amount')
plt.xlabel('Number of Services')
plt.ylabel('Average Medicare Payment Amount')
plt.show()
```



Average Submitted Charge Amount by City

```
# Bar graph of average submitted charge amount by city
top_20_cities = data['City of the Provider'].value_counts().head(20).index
filtered_city_data = data[data['City of the Provider'].isin(top_20_cities)]

plt.figure(figsize=(14, 8))
sns.barplot(x='Average Submitted Charge Amount', y='City of the Provider', data=filtered_city_data, estimator=sum)
plt.title('Average Submitted Charge Amount by Top 20 Cities')
plt.xlabel('Average Submitted Charge Amount')
plt.ylabel('City')
plt.show()
```



Average Submitted Charge Amount by Top 20 Cities

SAINT LOUIS

