In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
beneficiary_data = pd.read_csv('D:/CMS_DS/Health Claims/Train_Beneficiarydata-15428
inpatient_data = pd.read_csv('D:/CMS_DS/Health Claims/Train_Inpatientdata-154286562
outpatient_data = pd.read_csv('D:/CMS_DS/Health Claims/Train_Outpatientdata-1542865
claims_data = pd.read_csv('D:/CMS_DS/Health Claims/Train-1542865627584.csv')
```

In [4]:
```
# Display the first few rows of the beneficiary data
print("Beneficiary Data")
display(beneficiary_data.head())

# Display the first few rows of the inpatient data
print("\nInpatient Data")
display(inpatient_data.head())

# Display the first few rows of the outpatient data
print("\nOutpatient Data")
display(outpatient_data.head())

# Display the first few rows of the claims data
print("\nClaims Data")
display(claims_data.head())
```

```
Beneficiary Data
```

	BeneID	DOB	DOD	Gender	Race	RenalDiseaseIndicator	State	County	NoOfMonthl
0	BENE11001	1943-01-01	NaN	1	1	0	39	230	
1	BENE11002	1936-09-01	NaN	2	1	0	39	280	
2	BENE11003	1936-08-01	NaN	1	1	0	52	590	
3	BENE11004	1922-07-01	NaN	1	1	0	39	270	
4	BENE11005	1935-09-01	NaN	1	1	0	24	680	

5 rows × 25 columns

```
Inpatient Data
```

	BeneID	ClaimID	ClaimStartDt	ClaimEndDt	Provider	InscClaimAmtReimbursed	At
0	BENE11001	CLM46614	2009-04-12	2009-04-18	PRV55912	26000	
1	BENE11001	CLM66048	2009-08-31	2009-09-02	PRV55907	5000	
2	BENE11001	CLM68358	2009-09-17	2009-09-20	PRV56046	5000	
3	BENE11011	CLM38412	2009-02-14	2009-02-22	PRV52405	5000	
4	BENE11014	CLM63689	2009-08-13	2009-08-30	PRV56614	10000	

5 rows × 30 columns

Outpatient Data

	BeneID	ClaimID	ClaimStartDt	ClaimEndDt	Provider	InscClaimAmtReimbursed	A
0	BENE11002	CLM624349	2009-10-11	2009-10-11	PRV56011	30	
1	BENE11003	CLM189947	2009-02-12	2009-02-12	PRV57610	80	
2	BENE11003	CLM438021	2009-06-27	2009-06-27	PRV57595	10	
3	BENE11004	CLM121801	2009-01-06	2009-01-06	PRV56011	40	
4	BENE11004	CLM150998	2009-01-22	2009-01-22	PRV56011	200	

5 rows × 27 columns

Claims Data

	Provider	PotentialFraud
0	PRV51001	No
1	PRV51003	Yes
2	PRV51004	No
3	PRV51005	Yes
4	PRV51007	No

In [5]:
```
# Check for missing values and data types in the beneficiary data
print("Beneficiary Data Info")
print(beneficiary_data.info())

# Check for missing values and data types in the inpatient data
print("\nInpatient Data Info")
print(inpatient_data.info())

# Check for missing values and data types in the outpatient data
print("\nOutpatient Data Info")
print(outpatient_data.info())

# Check for missing values and data types in the claims data
```

```
print("\nClaims Data Info")
print(claims_data.info())
```

```
Beneficiary Data Info
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 138556 entries, 0 to 138555
Data columns (total 25 columns):
 #   Column                            Non-Null Count   Dtype
---  ------                            --------------   -----
 0   BeneID                            138556 non-null  object
 1   DOB                               138556 non-null  object
 2   DOD                               1421 non-null    object
 3   Gender                            138556 non-null  int64
 4   Race                              138556 non-null  int64
 5   RenalDiseaseIndicator             138556 non-null  object
 6   State                             138556 non-null  int64
 7   County                            138556 non-null  int64
 8   NoOfMonths_PartACov               138556 non-null  int64
 9   NoOfMonths_PartBCov               138556 non-null  int64
 10  ChronicCond_Alzheimer             138556 non-null  int64
 11  ChronicCond_Heartfailure          138556 non-null  int64
 12  ChronicCond_KidneyDisease         138556 non-null  int64
 13  ChronicCond_Cancer                138556 non-null  int64
 14  ChronicCond_ObstrPulmonary        138556 non-null  int64
 15  ChronicCond_Depression            138556 non-null  int64
 16  ChronicCond_Diabetes              138556 non-null  int64
 17  ChronicCond_IschemicHeart         138556 non-null  int64
 18  ChronicCond_Osteoporasis          138556 non-null  int64
 19  ChronicCond_rheumatoidarthritis   138556 non-null  int64
 20  ChronicCond_stroke                138556 non-null  int64
 21  IPAnnualReimbursementAmt          138556 non-null  int64
 22  IPAnnualDeductibleAmt             138556 non-null  int64
 23  OPAnnualReimbursementAmt          138556 non-null  int64
 24  OPAnnualDeductibleAmt             138556 non-null  int64
dtypes: int64(21), object(4)
memory usage: 26.4+ MB
None

Inpatient Data Info
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40474 entries, 0 to 40473
Data columns (total 30 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   BeneID                   40474 non-null  object
 1   ClaimID                  40474 non-null  object
 2   ClaimStartDt             40474 non-null  object
 3   ClaimEndDt               40474 non-null  object
 4   Provider                 40474 non-null  object
 5   InscClaimAmtReimbursed   40474 non-null  int64
 6   AttendingPhysician       40362 non-null  object
 7   OperatingPhysician       23830 non-null  object
 8   OtherPhysician           4690 non-null   object
 9   AdmissionDt              40474 non-null  object
 10  ClmAdmitDiagnosisCode    40474 non-null  object
 11  DeductibleAmtPaid        39575 non-null  float64
 12  DischargeDt              40474 non-null  object
 13  DiagnosisGroupCode       40474 non-null  object
 14  ClmDiagnosisCode_1       40474 non-null  object
```

```
 15  ClmDiagnosisCode_2      40248 non-null  object
 16  ClmDiagnosisCode_3      39798 non-null  object
 17  ClmDiagnosisCode_4      38940 non-null  object
 18  ClmDiagnosisCode_5      37580 non-null  object
 19  ClmDiagnosisCode_6      35636 non-null  object
 20  ClmDiagnosisCode_7      33216 non-null  object
 21  ClmDiagnosisCode_8      30532 non-null  object
 22  ClmDiagnosisCode_9      26977 non-null  object
 23  ClmDiagnosisCode_10     3927 non-null   object
 24  ClmProcedureCode_1      23148 non-null  float64
 25  ClmProcedureCode_2      5454 non-null   float64
 26  ClmProcedureCode_3      965 non-null    float64
 27  ClmProcedureCode_4      116 non-null    float64
 28  ClmProcedureCode_5      9 non-null      float64
 29  ClmProcedureCode_6      0 non-null      float64
dtypes: float64(7), int64(1), object(22)
memory usage: 9.3+ MB
None

Outpatient Data Info
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 517737 entries, 0 to 517736
Data columns (total 27 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   BeneID                  517737 non-null  object
 1   ClaimID                 517737 non-null  object
 2   ClaimStartDt            517737 non-null  object
 3   ClaimEndDt              517737 non-null  object
 4   Provider                517737 non-null  object
 5   InscClaimAmtReimbursed  517737 non-null  int64
 6   AttendingPhysician      516341 non-null  object
 7   OperatingPhysician      90617 non-null   object
 8   OtherPhysician          195046 non-null  object
 9   ClmDiagnosisCode_1      507284 non-null  object
 10  ClmDiagnosisCode_2      322357 non-null  object
 11  ClmDiagnosisCode_3      203257 non-null  object
 12  ClmDiagnosisCode_4      125596 non-null  object
 13  ClmDiagnosisCode_5      74344 non-null   object
 14  ClmDiagnosisCode_6      48756 non-null   object
 15  ClmDiagnosisCode_7      32961 non-null   object
 16  ClmDiagnosisCode_8      22912 non-null   object
 17  ClmDiagnosisCode_9      14838 non-null   object
 18  ClmDiagnosisCode_10     1083 non-null    object
 19  ClmProcedureCode_1      162 non-null     float64
 20  ClmProcedureCode_2      36 non-null      float64
 21  ClmProcedureCode_3      4 non-null       float64
 22  ClmProcedureCode_4      2 non-null       float64
 23  ClmProcedureCode_5      0 non-null       float64
 24  ClmProcedureCode_6      0 non-null       float64
 25  DeductibleAmtPaid       517737 non-null  int64
 26  ClmAdmitDiagnosisCode   105425 non-null  object
dtypes: float64(6), int64(2), object(19)
memory usage: 106.7+ MB
None
```

```
Claims Data Info
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5410 entries, 0 to 5409
Data columns (total 2 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Provider        5410 non-null   object
 1   PotentialFraud  5410 non-null   object
dtypes: object(2)
memory usage: 84.7+ KB
None
```

In [6]:
```
# Descriptive statistics for the beneficiary data
print("Beneficiary Data Description")
display(beneficiary_data.describe())

# Descriptive statistics for the inpatient data
print("\nInpatient Data Description")
display(inpatient_data.describe())

# Descriptive statistics for the outpatient data
print("\nOutpatient Data Description")
display(outpatient_data.describe())

# Descriptive statistics for the claims data
print("\nClaims Data Description")
display(claims_data.describe())
```

```
Beneficiary Data Description
```

	Gender	Race	State	County	NoOfMonths_PartACov
count	138556.000000	138556.000000	138556.000000	138556.000000	138556.000000
mean	1.570932	1.254511	25.666734	374.424745	11.907727
std	0.494945	0.717007	15.223443	266.277581	1.032332
min	1.000000	1.000000	1.000000	0.000000	0.000000
25%	1.000000	1.000000	11.000000	141.000000	12.000000
50%	2.000000	1.000000	25.000000	340.000000	12.000000
75%	2.000000	1.000000	39.000000	570.000000	12.000000
max	2.000000	5.000000	54.000000	999.000000	12.000000

8 rows × 21 columns

```
Inpatient Data Description
```

	InscClaimAmtReimbursed	DeductibleAmtPaid	ClmProcedureCode_1	ClmProcedureCod
count	40474.000000	39575.0	23148.000000	5454.000
mean	10087.884074	1068.0	5894.611759	4103.738
std	10303.099402	0.0	3049.304400	2028.182
min	0.000000	1068.0	11.000000	42.000
25%	4000.000000	1068.0	3848.000000	2724.000
50%	7000.000000	1068.0	5369.000000	4019.000
75%	12000.000000	1068.0	8666.250000	4439.000
max	125000.000000	1068.0	9999.000000	9999.000

Outpatient Data Description

	InscClaimAmtReimbursed	ClmProcedureCode_1	ClmProcedureCode_2	ClmProcedureCo
count	517737.000000	162.000000	36.000000	4.00
mean	286.334799	6116.611111	4503.277778	2959.00
std	694.034343	3217.719258	2504.015000	1863.45
min	0.000000	51.000000	412.000000	412.00
25%	40.000000	3893.000000	2724.000000	2146.00
50%	80.000000	5244.500000	4019.000000	3511.50
75%	200.000000	9421.500000	5849.000000	4324.50
max	102500.000000	9999.000000	9982.000000	4401.00

Claims Data Description

	Provider	PotentialFraud
count	5410	5410
unique	5410	2
top	PRV51001	No
freq	1	4904

In [7]:
```
beneficiary_data.columns
```

Out[7]:
```
Index(['BeneID', 'DOB', 'DOD', 'Gender', 'Race', 'RenalDiseaseIndicator',
       'State', 'County', 'NoOfMonths_PartACov', 'NoOfMonths_PartBCov',
       'ChronicCond_Alzheimer', 'ChronicCond_Heartfailure',
       'ChronicCond_KidneyDisease', 'ChronicCond_Cancer',
       'ChronicCond_ObstrPulmonary', 'ChronicCond_Depression',
       'ChronicCond_Diabetes', 'ChronicCond_IschemicHeart',
       'ChronicCond_Osteoporasis', 'ChronicCond_rheumatoidarthritis',
       'ChronicCond_stroke', 'IPAnnualReimbursementAmt',
       'IPAnnualDeductibleAmt', 'OPAnnualReimbursementAmt',
       'OPAnnualDeductibleAmt'],
      dtype='object')
```

In [8]:
```
inpatient_data.columns
```

Out[8]:
```
Index(['BeneID', 'ClaimID', 'ClaimStartDt', 'ClaimEndDt', 'Provider',
       'InscClaimAmtReimbursed', 'AttendingPhysician', 'OperatingPhysician',
       'OtherPhysician', 'AdmissionDt', 'ClmAdmitDiagnosisCode',
       'DeductibleAmtPaid', 'DischargeDt', 'DiagnosisGroupCode',
       'ClmDiagnosisCode_1', 'ClmDiagnosisCode_2', 'ClmDiagnosisCode_3',
       'ClmDiagnosisCode_4', 'ClmDiagnosisCode_5', 'ClmDiagnosisCode_6',
       'ClmDiagnosisCode_7', 'ClmDiagnosisCode_8', 'ClmDiagnosisCode_9',
       'ClmDiagnosisCode_10', 'ClmProcedureCode_1', 'ClmProcedureCode_2',
       'ClmProcedureCode_3', 'ClmProcedureCode_4', 'ClmProcedureCode_5',
       'ClmProcedureCode_6'],
      dtype='object')
```

In [9]:
```
outpatient_data.columns
```

Out[9]:
```
Index(['BeneID', 'ClaimID', 'ClaimStartDt', 'ClaimEndDt', 'Provider',
       'InscClaimAmtReimbursed', 'AttendingPhysician', 'OperatingPhysician',
       'OtherPhysician', 'ClmDiagnosisCode_1', 'ClmDiagnosisCode_2',
       'ClmDiagnosisCode_3', 'ClmDiagnosisCode_4', 'ClmDiagnosisCode_5',
       'ClmDiagnosisCode_6', 'ClmDiagnosisCode_7', 'ClmDiagnosisCode_8',
       'ClmDiagnosisCode_9', 'ClmDiagnosisCode_10', 'ClmProcedureCode_1',
       'ClmProcedureCode_2', 'ClmProcedureCode_3', 'ClmProcedureCode_4',
       'ClmProcedureCode_5', 'ClmProcedureCode_6', 'DeductibleAmtPaid',
       'ClmAdmitDiagnosisCode'],
      dtype='object')
```

In [10]:
```
claims_data.columns
```

Out[10]:
```
Index(['Provider', 'PotentialFraud'], dtype='object')
```

In [11]:
```
beneficiary_data.isnull().sum()
```

Out[11]:
```
BeneID                              0
DOB                                 0
DOD                            137135
Gender                              0
Race                                0
RenalDiseaseIndicator               0
State                               0
County                              0
NoOfMonths_PartACov                 0
NoOfMonths_PartBCov                 0
ChronicCond_Alzheimer               0
ChronicCond_Heartfailure            0
ChronicCond_KidneyDisease           0
ChronicCond_Cancer                  0
ChronicCond_ObstrPulmonary          0
ChronicCond_Depression              0
ChronicCond_Diabetes                0
ChronicCond_IschemicHeart           0
ChronicCond_Osteoporasis            0
ChronicCond_rheumatoidarthritis     0
ChronicCond_stroke                  0
IPAnnualReimbursementAmt            0
IPAnnualDeductibleAmt               0
OPAnnualReimbursementAmt            0
OPAnnualDeductibleAmt               0
dtype: int64
```

In [12]:
```
inpatient_data.isnull().sum()
```

Out[12]:
```
BeneID                        0
ClaimID                       0
ClaimStartDt                  0
ClaimEndDt                    0
Provider                      0
InscClaimAmtReimbursed        0
AttendingPhysician          112
OperatingPhysician        16644
OtherPhysician            35784
AdmissionDt                   0
ClmAdmitDiagnosisCode         0
DeductibleAmtPaid           899
DischargeDt                   0
DiagnosisGroupCode            0
ClmDiagnosisCode_1            0
ClmDiagnosisCode_2          226
ClmDiagnosisCode_3          676
ClmDiagnosisCode_4         1534
ClmDiagnosisCode_5         2894
ClmDiagnosisCode_6         4838
ClmDiagnosisCode_7         7258
ClmDiagnosisCode_8         9942
ClmDiagnosisCode_9        13497
ClmDiagnosisCode_10       36547
ClmProcedureCode_1        17326
ClmProcedureCode_2        35020
ClmProcedureCode_3        39509
ClmProcedureCode_4        40358
ClmProcedureCode_5        40465
ClmProcedureCode_6        40474
dtype: int64
```

In [13]:
```
outpatient_data.isnull().sum()
```

Out[13]:
```
BeneID                        0
ClaimID                       0
ClaimStartDt                  0
ClaimEndDt                    0
Provider                      0
InscClaimAmtReimbursed        0
AttendingPhysician         1396
OperatingPhysician       427120
OtherPhysician           322691
ClmDiagnosisCode_1        10453
ClmDiagnosisCode_2       195380
ClmDiagnosisCode_3       314480
ClmDiagnosisCode_4       392141
ClmDiagnosisCode_5       443393
ClmDiagnosisCode_6       468981
ClmDiagnosisCode_7       484776
ClmDiagnosisCode_8       494825
ClmDiagnosisCode_9       502899
ClmDiagnosisCode_10      516654
ClmProcedureCode_1       517575
ClmProcedureCode_2       517701
ClmProcedureCode_3       517733
ClmProcedureCode_4       517735
ClmProcedureCode_5       517737
ClmProcedureCode_6       517737
DeductibleAmtPaid             0
ClmAdmitDiagnosisCode    412312
dtype: int64
```

In [14]:
```
claims_data.isnull().sum()
```

Out[14]:
```
Provider          0
PotentialFraud    0
dtype: int64
```

In [15]:
```
[features for features in beneficiary_data.columns if beneficiary_data[features].is
```

Out[15]:
```
['DOD']
```

In [16]:
```
[features for features in inpatient_data.columns if inpatient_data[features].isnull
```

Out[16]:
```
['AttendingPhysician',
 'OperatingPhysician',
 'OtherPhysician',
 'DeductibleAmtPaid',
 'ClmDiagnosisCode_2',
 'ClmDiagnosisCode_3',
 'ClmDiagnosisCode_4',
 'ClmDiagnosisCode_5',
 'ClmDiagnosisCode_6',
 'ClmDiagnosisCode_7',
 'ClmDiagnosisCode_8',
 'ClmDiagnosisCode_9',
 'ClmDiagnosisCode_10',
 'ClmProcedureCode_1',
 'ClmProcedureCode_2',
 'ClmProcedureCode_3',
 'ClmProcedureCode_4',
 'ClmProcedureCode_5',
 'ClmProcedureCode_6']
```

In [17]:
```
[features for features in outpatient_data.columns if outpatient_data[features].isnu
```

Out[17]:
```
['AttendingPhysician',
 'OperatingPhysician',
 'OtherPhysician',
 'ClmDiagnosisCode_1',
 'ClmDiagnosisCode_2',
 'ClmDiagnosisCode_3',
 'ClmDiagnosisCode_4',
 'ClmDiagnosisCode_5',
 'ClmDiagnosisCode_6',
 'ClmDiagnosisCode_7',
 'ClmDiagnosisCode_8',
 'ClmDiagnosisCode_9',
 'ClmDiagnosisCode_10',
 'ClmProcedureCode_1',
 'ClmProcedureCode_2',
 'ClmProcedureCode_3',
 'ClmProcedureCode_4',
 'ClmProcedureCode_5',
 'ClmProcedureCode_6',
 'ClmAdmitDiagnosisCode']
```

In [18]:
```
[features for features in claims_data.columns if claims_data[features].isnull().sum
```

Out[18]:
```
[]
```

In [19]:
```
inpatient_data.shape
```

Out[19]:
```
(40474, 30)
```

In [20]:
```
outpatient_data.shape
```

Out[20]:
```
(517737, 27)
```

In [21]:
```
beneficiary_data.shape
```

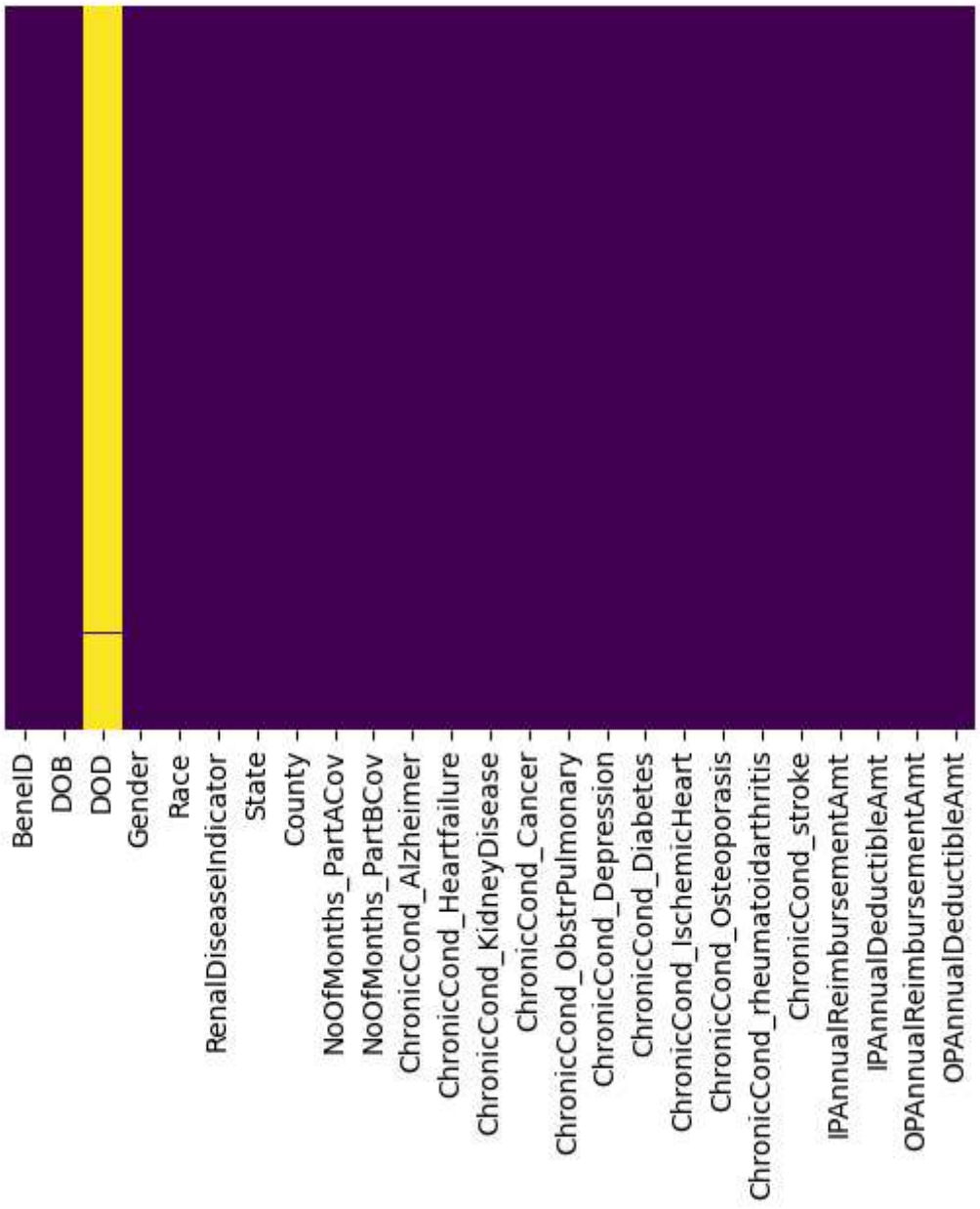
Out[21]: (138556, 25)

In [22]:
```
claims_data.shape
```

Out[22]: (5410, 2)

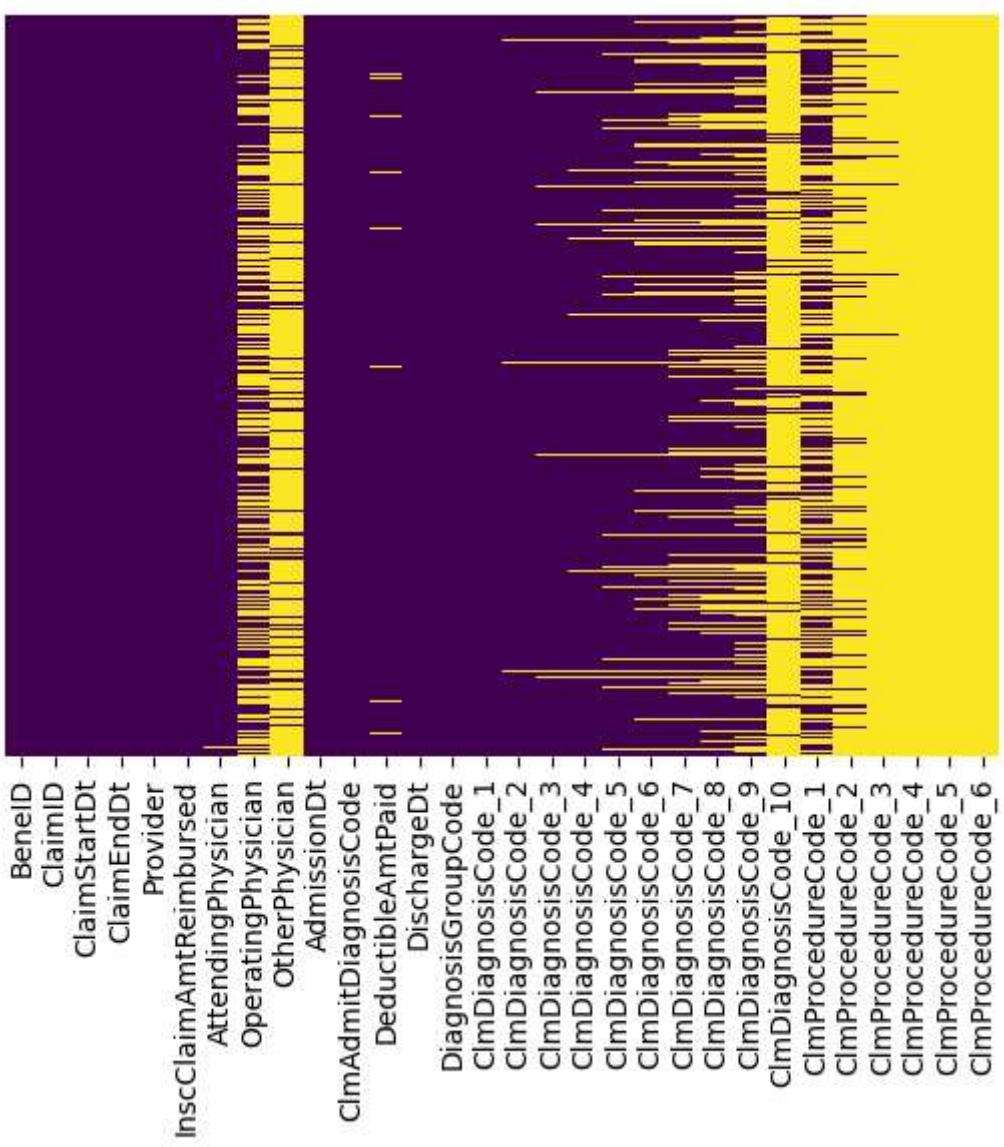
In [23]:
```
sns.heatmap(beneficiary_data.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[23]: <Axes: >



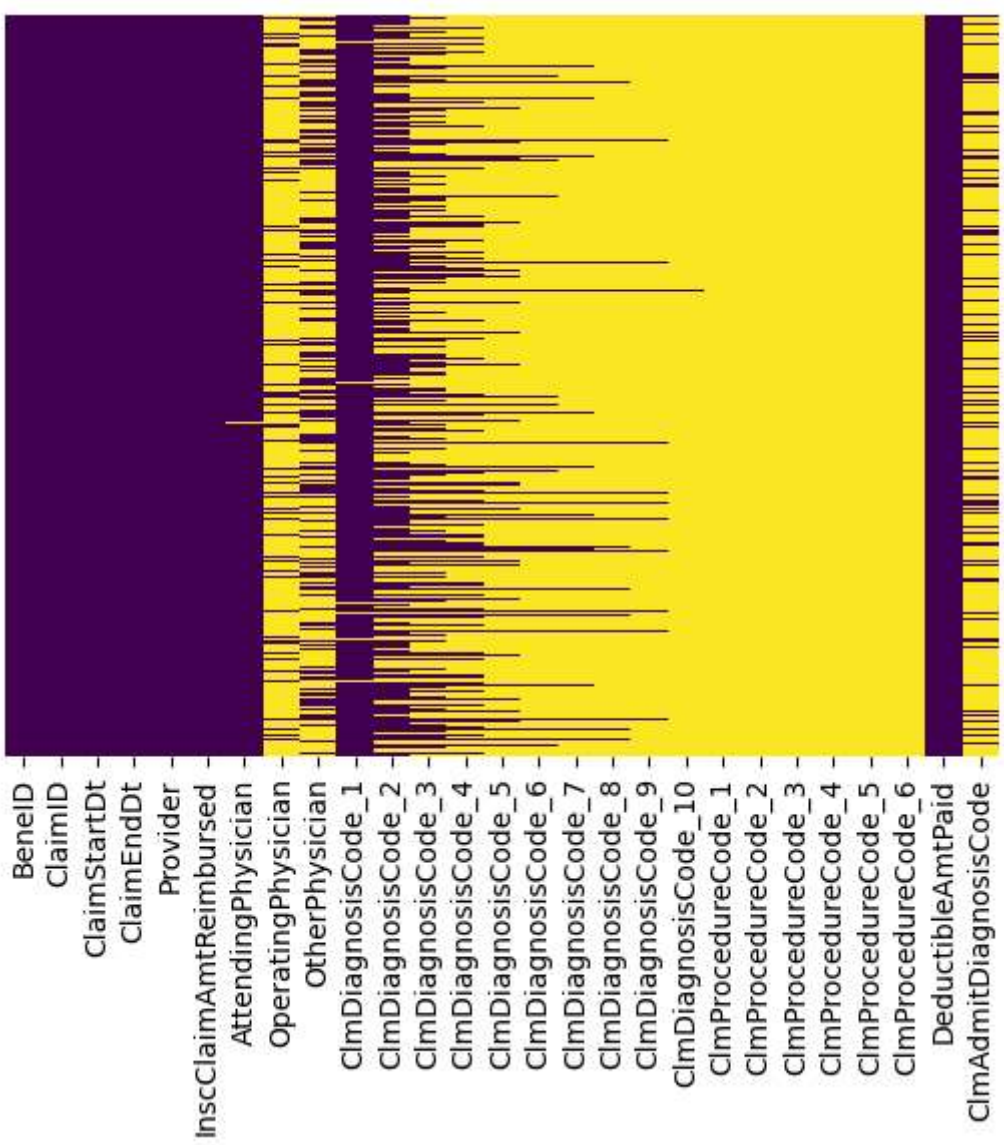
In [24]:
```
sns.heatmap(inpatient_data.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[24]: <Axes: >



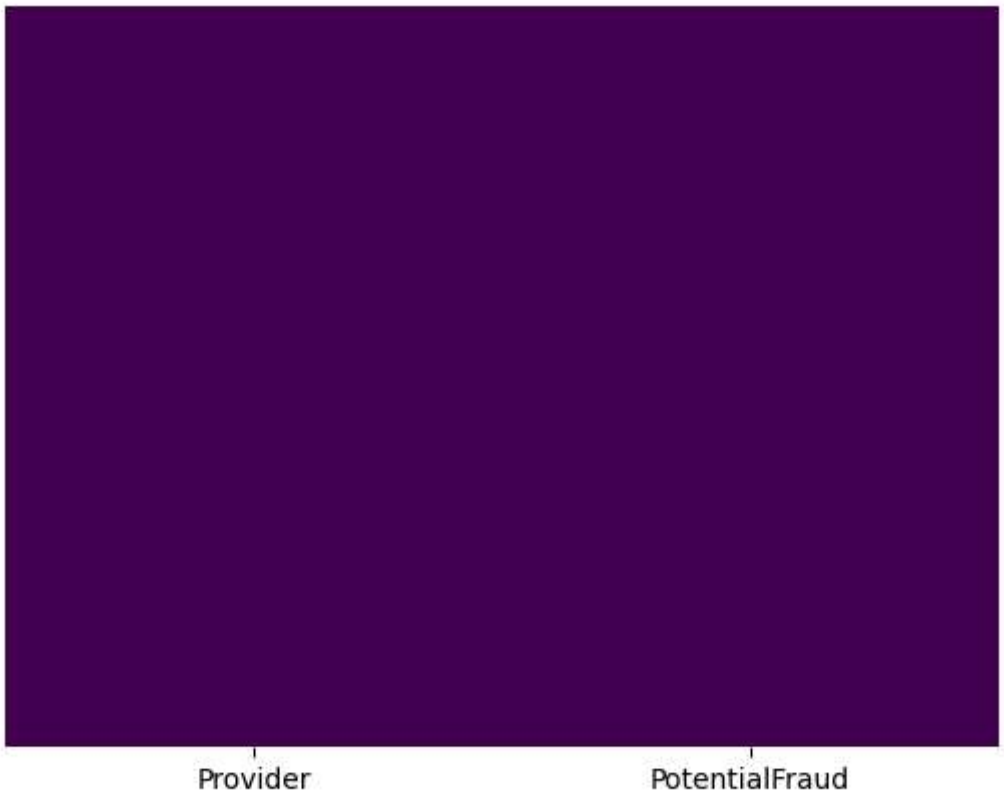
In [25]:
```
sns.heatmap(outpatient_data.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[25]: <Axes: >



In [26]:
```
sns.heatmap(claims_data.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[26]: <Axes: >



In [27]:
```
# Merging beneficiary data with inpatient and outpatient data on 'BeneID'
inpatient_merged = pd.merge(inpatient_data, beneficiary_data, on='BeneID', how='lef
outpatient_merged = pd.merge(outpatient_data, beneficiary_data, on='BeneID', how='l

# Merging the inpatient and outpatient data with train data on 'Provider'
# First, we combine inpatient and outpatient data
combined_in_out_patient = pd.concat([inpatient_merged, outpatient_merged], axis=0)

# Then merge with the claims data on 'Provider'
full_merged_data = pd.merge(combined_in_out_patient, claims_data, on='Provider', ho

# Checking the merged data structure
full_merged_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 558211 entries, 0 to 558210
Data columns (total 55 columns):
 #   Column                            Non-Null Count   Dtype  
---  ------                            --------------   -----  
 0   BeneID                            558211 non-null  object 
 1   ClaimID                           558211 non-null  object 
 2   ClaimStartDt                      558211 non-null  object 
 3   ClaimEndDt                        558211 non-null  object 
 4   Provider                          558211 non-null  object 
 5   InscClaimAmtReimbursed            558211 non-null  int64  
 6   AttendingPhysician                556703 non-null  object 
 7   OperatingPhysician                114447 non-null  object 
 8   OtherPhysician                    199736 non-null  object 
 9   AdmissionDt                       40474 non-null   object 
 10  ClmAdmitDiagnosisCode             145899 non-null  object 
 11  DeductibleAmtPaid                 557312 non-null  float64
 12  DischargeDt                       40474 non-null   object 
 13  DiagnosisGroupCode                40474 non-null   object 
 14  ClmDiagnosisCode_1                547758 non-null  object 
 15  ClmDiagnosisCode_2                362605 non-null  object 
 16  ClmDiagnosisCode_3                243055 non-null  object 
 17  ClmDiagnosisCode_4                164536 non-null  object 
 18  ClmDiagnosisCode_5                111924 non-null  object 
 19  ClmDiagnosisCode_6                84392 non-null   object 
 20  ClmDiagnosisCode_7                66177 non-null   object 
 21  ClmDiagnosisCode_8                53444 non-null   object 
 22  ClmDiagnosisCode_9                41815 non-null   object 
 23  ClmDiagnosisCode_10               5010 non-null    object 
 24  ClmProcedureCode_1                23310 non-null   float64
 25  ClmProcedureCode_2                5490 non-null    float64
 26  ClmProcedureCode_3                969 non-null     float64
 27  ClmProcedureCode_4                118 non-null     float64
 28  ClmProcedureCode_5                9 non-null       float64
 29  ClmProcedureCode_6                0 non-null       float64
 30  DOB                               558211 non-null  object 
 31  DOD                               4131 non-null    object 
 32  Gender                            558211 non-null  int64  
 33  Race                              558211 non-null  int64  
 34  RenalDiseaseIndicator             558211 non-null  object 
 35  State                             558211 non-null  int64  
 36  County                            558211 non-null  int64  
 37  NoOfMonths_PartACov               558211 non-null  int64  
 38  NoOfMonths_PartBCov               558211 non-null  int64  
 39  ChronicCond_Alzheimer             558211 non-null  int64  
 40  ChronicCond_Heartfailure          558211 non-null  int64  
 41  ChronicCond_KidneyDisease         558211 non-null  int64  
 42  ChronicCond_Cancer                558211 non-null  int64  
 43  ChronicCond_ObstrPulmonary        558211 non-null  int64  
 44  ChronicCond_Depression            558211 non-null  int64  
 45  ChronicCond_Diabetes              558211 non-null  int64  
 46  ChronicCond_IschemicHeart         558211 non-null  int64  
 47  ChronicCond_Osteoporasis          558211 non-null  int64  
 48  ChronicCond_rheumatoidarthritis   558211 non-null  int64  
 49  ChronicCond_stroke                558211 non-null  int64  
 50  IPAnnualReimbursementAmt          558211 non-null  int64  
```

```
 51  IPAnnualDeductibleAmt            558211 non-null  int64
 52  OPAnnualReimbursementAmt         558211 non-null  int64
 53  OPAnnualDeductibleAmt            558211 non-null  int64
 54  PotentialFraud                   558211 non-null  object
dtypes: float64(7), int64(22), object(26)
memory usage: 234.2+ MB
```

In [28]:
```
full_merged_data.head()
```

Out[28]:

	BeneID	ClaimID	ClaimStartDt	ClaimEndDt	Provider	InscClaimAmtReimbursed
0	BENE11001	CLM46614	2009-04-12	2009-04-18	PRV55912	26000
1	BENE11001	CLM66048	2009-08-31	2009-09-02	PRV55907	5000
2	BENE11001	CLM68358	2009-09-17	2009-09-20	PRV56046	5000
3	BENE11011	CLM38412	2009-02-14	2009-02-22	PRV52405	5000
4	BENE11014	CLM63689	2009-08-13	2009-08-30	PRV56614	10000

5 rows × 55 columns

In [29]:
```
full_merged_data.to_csv('full_merged_data.csv', index=False)
```

In [30]:
```
full_merged_data.describe()
```

Out[30]:

	InscClaimAmtReimbursed	DeductibleAmtPaid	ClmProcedureCode_1	ClmProcedureCo
count	558211.000000	557312.000000	23310.000000	5490.0
mean	997.012133	78.421085	5896.154612	4106.3
std	3821.534891	274.016812	3050.489933	2031.6
min	0.000000	0.000000	11.000000	42.0
25%	40.000000	0.000000	3848.000000	2724.0
50%	80.000000	0.000000	5363.000000	4019.0
75%	300.000000	0.000000	8669.000000	4439.0
max	125000.000000	1068.000000	9999.000000	9999.0

8 rows × 29 columns

In [31]:
```
full_merged_data.nunique()
```

Out[31]:

BeneID	138556
ClaimID	558211
ClaimStartDt	398
ClaimEndDt	366
Provider	5410
InscClaimAmtReimbursed	438
AttendingPhysician	82063
OperatingPhysician	35315
OtherPhysician	46457
AdmissionDt	398
ClmAdmitDiagnosisCode	4098
DeductibleAmtPaid	17
DischargeDt	365
DiagnosisGroupCode	736
ClmDiagnosisCode_1	10450
ClmDiagnosisCode_2	5300
ClmDiagnosisCode_3	4756
ClmDiagnosisCode_4	4359
ClmDiagnosisCode_5	3970
ClmDiagnosisCode_6	3607
ClmDiagnosisCode_7	3388
ClmDiagnosisCode_8	3070
ClmDiagnosisCode_9	2774
ClmDiagnosisCode_10	1158
ClmProcedureCode_1	1117
ClmProcedureCode_2	300
ClmProcedureCode_3	154
ClmProcedureCode_4	48
ClmProcedureCode_5	6
ClmProcedureCode_6	0
DOB	900
DOD	11
Gender	2
Race	4
RenalDiseaseIndicator	2
State	52
County	314
NoOfMonths_PartACov	13
NoOfMonths_PartBCov	13
ChronicCond_Alzheimer	2
ChronicCond_Heartfailure	2
ChronicCond_KidneyDisease	2
ChronicCond_Cancer	2
ChronicCond_ObstrPulmonary	2
ChronicCond_Depression	2
ChronicCond_Diabetes	2
ChronicCond_IschemicHeart	2
ChronicCond_Osteoporasis	2
ChronicCond_rheumatoidarthritis	2
ChronicCond_stroke	2
IPAnnualReimbursementAmt	3004
IPAnnualDeductibleAmt	147
OPAnnualReimbursementAmt	2078
OPAnnualDeductibleAmt	789
PotentialFraud	2

dtype: int64

In [32]:
```
full_merged_data.isnull().sum()
```

Out[32]:

BeneID	0
ClaimID	0
ClaimStartDt	0
ClaimEndDt	0
Provider	0
InscClaimAmtReimbursed	0
AttendingPhysician	1508
OperatingPhysician	443764
OtherPhysician	358475
AdmissionDt	517737
ClmAdmitDiagnosisCode	412312
DeductibleAmtPaid	899
DischargeDt	517737
DiagnosisGroupCode	517737
ClmDiagnosisCode_1	10453
ClmDiagnosisCode_2	195606
ClmDiagnosisCode_3	315156
ClmDiagnosisCode_4	393675
ClmDiagnosisCode_5	446287
ClmDiagnosisCode_6	473819
ClmDiagnosisCode_7	492034
ClmDiagnosisCode_8	504767
ClmDiagnosisCode_9	516396
ClmDiagnosisCode_10	553201
ClmProcedureCode_1	534901
ClmProcedureCode_2	552721
ClmProcedureCode_3	557242
ClmProcedureCode_4	558093
ClmProcedureCode_5	558202
ClmProcedureCode_6	558211
DOB	0
DOD	554080
Gender	0
Race	0
RenalDiseaseIndicator	0
State	0
County	0
NoOfMonths_PartACov	0
NoOfMonths_PartBCov	0
ChronicCond_Alzheimer	0
ChronicCond_Heartfailure	0
ChronicCond_KidneyDisease	0
ChronicCond_Cancer	0
ChronicCond_ObstrPulmonary	0
ChronicCond_Depression	0
ChronicCond_Diabetes	0
ChronicCond_IschemicHeart	0
ChronicCond_Osteoporasis	0
ChronicCond_rheumatoidarthritis	0
ChronicCond_stroke	0
IPAnnualReimbursementAmt	0
IPAnnualDeductibleAmt	0
OPAnnualReimbursementAmt	0
OPAnnualDeductibleAmt	0
PotentialFraud	0

dtype: int64

In [33]:
```
(full_merged_data.isnull().sum()/(len(full_merged_data)))*100
```

Out[33]:

BeneID	0.000000
ClaimID	0.000000
ClaimStartDt	0.000000
ClaimEndDt	0.000000
Provider	0.000000
InscClaimAmtReimbursed	0.000000
AttendingPhysician	0.270149
OperatingPhysician	79.497538
OtherPhysician	64.218548
AdmissionDt	92.749337
ClmAdmitDiagnosisCode	73.863109
DeductibleAmtPaid	0.161050
DischargeDt	92.749337
DiagnosisGroupCode	92.749337
ClmDiagnosisCode_1	1.872589
ClmDiagnosisCode_2	35.041588
ClmDiagnosisCode_3	56.458221
ClmDiagnosisCode_4	70.524407
ClmDiagnosisCode_5	79.949517
ClmDiagnosisCode_6	84.881702
ClmDiagnosisCode_7	88.144805
ClmDiagnosisCode_8	90.425843
ClmDiagnosisCode_9	92.509105
ClmDiagnosisCode_10	99.102490
ClmProcedureCode_1	95.824160
ClmProcedureCode_2	99.016501
ClmProcedureCode_3	99.826410
ClmProcedureCode_4	99.978861
ClmProcedureCode_5	99.998388
ClmProcedureCode_6	100.000000
DOB	0.000000
DOD	99.259957
Gender	0.000000
Race	0.000000
RenalDiseaseIndicator	0.000000
State	0.000000
County	0.000000
NoOfMonths_PartACov	0.000000
NoOfMonths_PartBCov	0.000000
ChronicCond_Alzheimer	0.000000
ChronicCond_Heartfailure	0.000000
ChronicCond_KidneyDisease	0.000000
ChronicCond_Cancer	0.000000
ChronicCond_ObstrPulmonary	0.000000
ChronicCond_Depression	0.000000
ChronicCond_Diabetes	0.000000
ChronicCond_IschemicHeart	0.000000
ChronicCond_Osteoporasis	0.000000
ChronicCond_rheumatoidarthritis	0.000000
ChronicCond_stroke	0.000000
IPAnnualReimbursementAmt	0.000000
IPAnnualDeductibleAmt	0.000000
OPAnnualReimbursementAmt	0.000000
OPAnnualDeductibleAmt	0.000000
PotentialFraud	0.000000

dtype: float64

In [34]:
```
# Check the first few rows to understand the data structure
print(full_merged_data[['BeneID', 'ClaimEndDt', 'DOD']].head())

# Inspect missing DOD values
print(full_merged_data[full_merged_data['DOD'].isnull()][['BeneID', 'ClaimEndDt', '
```

```
      BeneID  ClaimEndDt  DOD
0  BENE11001  2009-04-18  NaN
1  BENE11001  2009-09-02  NaN
2  BENE11001  2009-09-20  NaN
3  BENE11011  2009-02-22  NaN
4  BENE11014  2009-08-30  NaN
      BeneID  ClaimEndDt  DOD
0  BENE11001  2009-04-18  NaN
1  BENE11001  2009-09-02  NaN
2  BENE11001  2009-09-20  NaN
3  BENE11011  2009-02-22  NaN
4  BENE11014  2009-08-30  NaN
```

In [35]:
```
# Fill missing DOD with ClaimEndDt
full_merged_data.loc[full_merged_data['DOD'].isnull(), 'DOD'] = full_merged_data['C
```

In [36]:
```
# Verify the changes
print(full_merged_data[['BeneID', 'ClaimEndDt', 'DOD']].head())
```

```
      BeneID  ClaimEndDt         DOD
0  BENE11001  2009-04-18  2009-04-18
1  BENE11001  2009-09-02  2009-09-02
2  BENE11001  2009-09-20  2009-09-20
3  BENE11011  2009-02-22  2009-02-22
4  BENE11014  2009-08-30  2009-08-30
```

In [37]:
```
full_merged_data.to_csv('full_merged_data.csv', index=False)
```

In [38]:
```
# Ensure that admission and discharge dates are in datetime format
full_merged_data['AdmissionDt'] = pd.to_datetime(full_merged_data['AdmissionDt'])
full_merged_data['DischargeDt'] = pd.to_datetime(full_merged_data['DischargeDt'])

# Calculate Length of Stay
full_merged_data['LengthOfStay'] = (full_merged_data['DischargeDt'] - full_merged_d
```

In [39]:
```
full_merged_data.head()
```

Out[39]:

	BeneID	ClaimID	ClaimStartDt	ClaimEndDt	Provider	InscClaimAmtReimbursed
0	BENE11001	CLM46614	2009-04-12	2009-04-18	PRV55912	26000
1	BENE11001	CLM66048	2009-08-31	2009-09-02	PRV55907	5000
2	BENE11001	CLM68358	2009-09-17	2009-09-20	PRV56046	5000
3	BENE11011	CLM38412	2009-02-14	2009-02-22	PRV52405	5000
4	BENE11014	CLM63689	2009-08-13	2009-08-30	PRV56614	10000

5 rows × 56 columns

In [40]:
```
print(full_merged_data[['BeneID', 'AdmissionDt', 'DischargeDt', 'LengthOfStay']].he
```

```
      BeneID AdmissionDt DischargeDt  LengthOfStay
0  BENE11001  2009-04-12  2009-04-18           6.0
1  BENE11001  2009-08-31  2009-09-02           2.0
2  BENE11001  2009-09-17  2009-09-20           3.0
3  BENE11011  2009-02-14  2009-02-22           8.0
4  BENE11014  2009-08-13  2009-08-30          17.0
```

In [41]:
```
beneficiary_data.shape
```

Out[41]: (138556, 25)

In [42]:
```
inpatient_data.shape
```

Out[42]: (40474, 30)

In [43]:
```
outpatient_data.shape
```

Out[43]: (517737, 27)

In [44]:
```
claims_data.shape
```

Out[44]: (5410, 2)

In [45]:
```
print(inpatient_data['BeneID'].is_unique)
print(outpatient_data['BeneID'].is_unique)
print(beneficiary_data['BeneID'].is_unique)
print(claims_data['Provider'].is_unique)
```

```
False
False
True
True
```

In [46]:
```
# Merge inpatient data with an indicator
inpatient_check = pd.merge(inpatient_data, beneficiary_data, on='BeneID', how='left

# Check which BeneID in inpatient_data did not have a match in beneficiary_data
inpatient_unmatched = inpatient_check[inpatient_check['_merge'] == 'left_only']
```

```
# Do the same for outpatient data
outpatient_check = pd.merge(outpatient_data, beneficiary_data, on='BeneID', how='le
outpatient_unmatched = outpatient_check[outpatient_check['_merge'] == 'left_only']

# Now, inpatient_unmatched and outpatient_unmatched contain the rows from inpatient
# Let's see the counts of unmatched BeneID
unmatched_inpatient_count = inpatient_unmatched['BeneID'].nunique()
unmatched_outpatient_count = outpatient_unmatched['BeneID'].nunique()

(unmatched_inpatient_count, unmatched_outpatient_count)
```

Out[46]: (0, 0)

In [47]:
```
# Step 1: Concatenate inpatient and outpatient data
combined_claims = pd.concat([inpatient_data, outpatient_data], axis=0)

# Step 2: Merge the combined claims with beneficiary data
combined_claims_beneficiary = pd.merge(combined_claims, beneficiary_data, on='BeneI

# Step 3: Merge the resulting data with claims data
full_merged_data = pd.merge(combined_claims_beneficiary, claims_data, on='Provider'
```

In [48]:
```
full_merged_data.to_csv('full_merged_data.csv', index=False)
```

In [49]:
```
combined_claims_beneficiary.to_csv('combined_claims_beneficiary.csv', index=False)
```

In [50]:
```
full_merged_data.shape
```

Out[50]: (558211, 55)

In [51]:
```
(full_merged_data.isnull().sum()/(len(full_merged_data)))*100
```

Out[51]:

BeneID	0.000000
ClaimID	0.000000
ClaimStartDt	0.000000
ClaimEndDt	0.000000
Provider	0.000000
InscClaimAmtReimbursed	0.000000
AttendingPhysician	0.270149
OperatingPhysician	79.497538
OtherPhysician	64.218548
AdmissionDt	92.749337
ClmAdmitDiagnosisCode	73.863109
DeductibleAmtPaid	0.161050
DischargeDt	92.749337
DiagnosisGroupCode	92.749337
ClmDiagnosisCode_1	1.872589
ClmDiagnosisCode_2	35.041588
ClmDiagnosisCode_3	56.458221
ClmDiagnosisCode_4	70.524407
ClmDiagnosisCode_5	79.949517
ClmDiagnosisCode_6	84.881702
ClmDiagnosisCode_7	88.144805
ClmDiagnosisCode_8	90.425843
ClmDiagnosisCode_9	92.509105
ClmDiagnosisCode_10	99.102490
ClmProcedureCode_1	95.824160
ClmProcedureCode_2	99.016501
ClmProcedureCode_3	99.826410
ClmProcedureCode_4	99.978861
ClmProcedureCode_5	99.998388
ClmProcedureCode_6	100.000000
DOB	0.000000
DOD	99.259957
Gender	0.000000
Race	0.000000
RenalDiseaseIndicator	0.000000
State	0.000000
County	0.000000
NoOfMonths_PartACov	0.000000
NoOfMonths_PartBCov	0.000000
ChronicCond_Alzheimer	0.000000
ChronicCond_Heartfailure	0.000000
ChronicCond_KidneyDisease	0.000000
ChronicCond_Cancer	0.000000
ChronicCond_ObstrPulmonary	0.000000
ChronicCond_Depression	0.000000
ChronicCond_Diabetes	0.000000
ChronicCond_IschemicHeart	0.000000
ChronicCond_Osteoporasis	0.000000
ChronicCond_rheumatoidarthritis	0.000000
ChronicCond_stroke	0.000000
IPAnnualReimbursementAmt	0.000000
IPAnnualDeductibleAmt	0.000000
OPAnnualReimbursementAmt	0.000000
OPAnnualDeductibleAmt	0.000000
PotentialFraud	0.000000

dtype: float64

In [52]:
```
# Check the first few rows to understand the data structure
print(full_merged_data[['BeneID', 'ClaimEndDt', 'DOD']].head())

# Inspect missing DOD values
print(full_merged_data[full_merged_data['DOD'].isnull()][['BeneID', 'ClaimEndDt', '
```

```
      BeneID  ClaimEndDt  DOD
0  BENE11001  2009-04-18  NaN
1  BENE11001  2009-09-02  NaN
2  BENE11001  2009-09-20  NaN
3  BENE11011  2009-02-22  NaN
4  BENE11014  2009-08-30  NaN
      BeneID  ClaimEndDt  DOD
0  BENE11001  2009-04-18  NaN
1  BENE11001  2009-09-02  NaN
2  BENE11001  2009-09-20  NaN
3  BENE11011  2009-02-22  NaN
4  BENE11014  2009-08-30  NaN
```

In [53]:
```
# Fill missing DOD with ClaimEndDt
full_merged_data.loc[full_merged_data['DOD'].isnull(), 'DOD'] = full_merged_data['C
```

In [54]:
```
# Verify the changes
print(full_merged_data[['BeneID', 'ClaimEndDt', 'DOD']].head())
```

```
      BeneID  ClaimEndDt         DOD
0  BENE11001  2009-04-18  2009-04-18
1  BENE11001  2009-09-02  2009-09-02
2  BENE11001  2009-09-20  2009-09-20
3  BENE11011  2009-02-22  2009-02-22
4  BENE11014  2009-08-30  2009-08-30
```

In [55]:
```
# Ensure that admission and discharge dates are in datetime format
full_merged_data['AdmissionDt'] = pd.to_datetime(full_merged_data['AdmissionDt'])
full_merged_data['DischargeDt'] = pd.to_datetime(full_merged_data['DischargeDt'])

# Calculate Length of Stay
full_merged_data['LengthOfStay'] = (full_merged_data['DischargeDt'] - full_merged_d
```

In [56]:
```
print(full_merged_data[['BeneID', 'AdmissionDt', 'DischargeDt', 'LengthOfStay']].he
```

```
      BeneID AdmissionDt DischargeDt  LengthOfStay
0  BENE11001  2009-04-12  2009-04-18           6.0
1  BENE11001  2009-08-31  2009-09-02           2.0
2  BENE11001  2009-09-17  2009-09-20           3.0
3  BENE11011  2009-02-14  2009-02-22           8.0
4  BENE11014  2009-08-13  2009-08-30          17.0
```

In [57]:
```
print(full_merged_data['BeneID'].is_unique)
```

```
False
```

In [58]:
```
# Convert ClaimEndDt to datetime if it's not already
full_merged_data['ClaimEndDt'] = pd.to_datetime(full_merged_data['ClaimEndDt'])

# Group by 'BeneID' and use 'transform' to assign the max ClaimEndDt to each entry
```

```
full_merged_data['MostRecentClaimEndDt'] = full_merged_data.groupby('BeneID')['Clai

# Now, set this date as the DOD for each BeneID
full_merged_data['DOD'] = full_merged_data['MostRecentClaimEndDt']

# If you want to only update DOD where it's missing
full_merged_data.loc[full_merged_data['DOD'].isnull(), 'DOD'] = full_merged_data['M
```

In [59]:
```
print(full_merged_data[['BeneID', 'DOB', 'DOD']].head())
```

```
     BeneID         DOB        DOD
0  BENE11001  1943-01-01 2009-09-20
1  BENE11001  1943-01-01 2009-09-20
2  BENE11001  1943-01-01 2009-09-20
3  BENE11011  1914-03-01 2009-08-04
4  BENE11014  1938-04-01 2009-08-30
```

In [60]:
```
full_merged_data.to_csv('full_merged_data.csv', index=False)
```

In [61]:
```
(full_merged_data.isnull().sum()/(len(full_merged_data)))*100
```

Out[61]:

BeneID	0.000000
ClaimID	0.000000
ClaimStartDt	0.000000
ClaimEndDt	0.000000
Provider	0.000000
InscClaimAmtReimbursed	0.000000
AttendingPhysician	0.270149
OperatingPhysician	79.497538
OtherPhysician	64.218548
AdmissionDt	92.749337
ClmAdmitDiagnosisCode	73.863109
DeductibleAmtPaid	0.161050
DischargeDt	92.749337
DiagnosisGroupCode	92.749337
ClmDiagnosisCode_1	1.872589
ClmDiagnosisCode_2	35.041588
ClmDiagnosisCode_3	56.458221
ClmDiagnosisCode_4	70.524407
ClmDiagnosisCode_5	79.949517
ClmDiagnosisCode_6	84.881702
ClmDiagnosisCode_7	88.144805
ClmDiagnosisCode_8	90.425843
ClmDiagnosisCode_9	92.509105
ClmDiagnosisCode_10	99.102490
ClmProcedureCode_1	95.824160
ClmProcedureCode_2	99.016501
ClmProcedureCode_3	99.826410
ClmProcedureCode_4	99.978861
ClmProcedureCode_5	99.998388
ClmProcedureCode_6	100.000000
DOB	0.000000
DOD	0.000000
Gender	0.000000
Race	0.000000
RenalDiseaseIndicator	0.000000
State	0.000000
County	0.000000
NoOfMonths_PartACov	0.000000
NoOfMonths_PartBCov	0.000000
ChronicCond_Alzheimer	0.000000
ChronicCond_Heartfailure	0.000000
ChronicCond_KidneyDisease	0.000000
ChronicCond_Cancer	0.000000
ChronicCond_ObstrPulmonary	0.000000
ChronicCond_Depression	0.000000
ChronicCond_Diabetes	0.000000
ChronicCond_IschemicHeart	0.000000
ChronicCond_Osteoporasis	0.000000
ChronicCond_rheumatoidarthritis	0.000000
ChronicCond_stroke	0.000000
IPAnnualReimbursementAmt	0.000000
IPAnnualDeductibleAmt	0.000000
OPAnnualReimbursementAmt	0.000000
OPAnnualDeductibleAmt	0.000000
PotentialFraud	0.000000
LengthOfStay	92.749337

```
MostRecentClaimEndDt                0.000000
dtype: float64
```

In [62]:
```
(inpatient_data.isnull().sum()/(len(inpatient_data)))*100
```

Out[62]:
```
BeneID                       0.000000
ClaimID                      0.000000
ClaimStartDt                 0.000000
ClaimEndDt                   0.000000
Provider                     0.000000
InscClaimAmtReimbursed       0.000000
AttendingPhysician           0.276721
OperatingPhysician          41.122696
OtherPhysician              88.412314
AdmissionDt                  0.000000
ClmAdmitDiagnosisCode        0.000000
DeductibleAmtPaid            2.221179
DischargeDt                  0.000000
DiagnosisGroupCode           0.000000
ClmDiagnosisCode_1           0.000000
ClmDiagnosisCode_2           0.558383
ClmDiagnosisCode_3           1.670208
ClmDiagnosisCode_4           3.790087
ClmDiagnosisCode_5           7.150269
ClmDiagnosisCode_6          11.953353
ClmDiagnosisCode_7          17.932500
ClmDiagnosisCode_8          24.563918
ClmDiagnosisCode_9          33.347334
ClmDiagnosisCode_10         90.297475
ClmProcedureCode_1          42.807728
ClmProcedureCode_2          86.524683
ClmProcedureCode_3          97.615753
ClmProcedureCode_4          99.713396
ClmProcedureCode_5          99.977764
ClmProcedureCode_6         100.000000
dtype: float64
```

In [63]:
```
(outpatient_data.isnull().sum()/(len(outpatient_data)))*100
```

Out[63]:
```
BeneID                       0.000000
ClaimID                      0.000000
ClaimStartDt                 0.000000
ClaimEndDt                   0.000000
Provider                     0.000000
InscClaimAmtReimbursed       0.000000
AttendingPhysician           0.269635
OperatingPhysician          82.497484
OtherPhysician              62.327205
ClmDiagnosisCode_1           2.018979
ClmDiagnosisCode_2          37.737307
ClmDiagnosisCode_3          60.741264
ClmDiagnosisCode_4          75.741351
ClmDiagnosisCode_5          85.640586
ClmDiagnosisCode_6          90.582864
ClmDiagnosisCode_7          93.633640
ClmDiagnosisCode_8          95.574587
ClmDiagnosisCode_9          97.134066
ClmDiagnosisCode_10         99.790820
ClmProcedureCode_1          99.968710
ClmProcedureCode_2          99.993047
ClmProcedureCode_3          99.999227
ClmProcedureCode_4          99.999614
ClmProcedureCode_5         100.000000
ClmProcedureCode_6         100.000000
DeductibleAmtPaid            0.000000
ClmAdmitDiagnosisCode       79.637345
dtype: float64
```

In [64]:
```
full_merged_data['ClaimEndDt'] = pd.to_datetime(full_merged_data['ClaimEndDt'])
full_merged_data['ClaimStartDt'] = pd.to_datetime(full_merged_data['ClaimStartDt'])

# Calculate claim duration
full_merged_data['ClaimDuration'] = (full_merged_data['ClaimEndDt'] - full_merged_d
```

In [65]:
```
(full_merged_data.isnull().sum()/(len(full_merged_data)))*100
```

Out[65]:

BeneID	0.000000
ClaimID	0.000000
ClaimStartDt	0.000000
ClaimEndDt	0.000000
Provider	0.000000
InscClaimAmtReimbursed	0.000000
AttendingPhysician	0.270149
OperatingPhysician	79.497538
OtherPhysician	64.218548
AdmissionDt	92.749337
ClmAdmitDiagnosisCode	73.863109
DeductibleAmtPaid	0.161050
DischargeDt	92.749337
DiagnosisGroupCode	92.749337
ClmDiagnosisCode_1	1.872589
ClmDiagnosisCode_2	35.041588
ClmDiagnosisCode_3	56.458221
ClmDiagnosisCode_4	70.524407
ClmDiagnosisCode_5	79.949517
ClmDiagnosisCode_6	84.881702
ClmDiagnosisCode_7	88.144805
ClmDiagnosisCode_8	90.425843
ClmDiagnosisCode_9	92.509105
ClmDiagnosisCode_10	99.102490
ClmProcedureCode_1	95.824160
ClmProcedureCode_2	99.016501
ClmProcedureCode_3	99.826410
ClmProcedureCode_4	99.978861
ClmProcedureCode_5	99.998388
ClmProcedureCode_6	100.000000
DOB	0.000000
DOD	0.000000
Gender	0.000000
Race	0.000000
RenalDiseaseIndicator	0.000000
State	0.000000
County	0.000000
NoOfMonths_PartACov	0.000000
NoOfMonths_PartBCov	0.000000
ChronicCond_Alzheimer	0.000000
ChronicCond_Heartfailure	0.000000
ChronicCond_KidneyDisease	0.000000
ChronicCond_Cancer	0.000000
ChronicCond_ObstrPulmonary	0.000000
ChronicCond_Depression	0.000000
ChronicCond_Diabetes	0.000000
ChronicCond_IschemicHeart	0.000000
ChronicCond_Osteoporasis	0.000000
ChronicCond_rheumatoidarthritis	0.000000
ChronicCond_stroke	0.000000
IPAnnualReimbursementAmt	0.000000
IPAnnualDeductibleAmt	0.000000
OPAnnualReimbursementAmt	0.000000
OPAnnualDeductibleAmt	0.000000
PotentialFraud	0.000000
LengthOfStay	92.749337

```
MostRecentClaimEndDt                0.000000
ClaimDuration                       0.000000
dtype: float64
```

In [66]:
```
full_merged_data.head()
```

Out[66]:

	BeneID	ClaimID	ClaimStartDt	ClaimEndDt	Provider	InscClaimAmtReimbursed
0	BENE11001	CLM46614	2009-04-12	2009-04-18	PRV55912	26000
1	BENE11001	CLM66048	2009-08-31	2009-09-02	PRV55907	5000
2	BENE11001	CLM68358	2009-09-17	2009-09-20	PRV56046	5000
3	BENE11011	CLM38412	2009-02-14	2009-02-22	PRV52405	5000
4	BENE11014	CLM63689	2009-08-13	2009-08-30	PRV56614	10000

5 rows × 58 columns

In [67]:
```
full_merged_data.describe()
```

Out[67]:

	ClaimStartDt	ClaimEndDt	InscClaimAmtReimbursed	AdmissionDt
count	558211	558211	558211.000000	40474
mean	2009-06-24 23:39:21.603766528	2009-06-26 17:07:35.601913600	997.012133	2009-06-19 17:38:12.493946880
min	2008-11-27 00:00:00	2008-12-28 00:00:00	0.000000	2008-11-27 00:00:00
25%	2009-03-27 00:00:00	2009-03-29 00:00:00	40.000000	2009-03-20 00:00:00
50%	2009-06-23 00:00:00	2009-06-24 00:00:00	80.000000	2009-06-16 00:00:00
75%	2009-09-22 00:00:00	2009-09-23 00:00:00	300.000000	2009-09-17 00:00:00
max	2009-12-31 00:00:00	2009-12-31 00:00:00	125000.000000	2009-12-31 00:00:00
std	NaN	NaN	3821.534891	NaN

8 rows × 37 columns

In [68]:
```
full_merged_data['DOB'] = pd.to_datetime(full_merged_data['DOB'])
full_merged_data['DOD'] = pd.to_datetime(full_merged_data['DOD'])

# Calculate claim duration
full_merged_data['Age'] = (full_merged_data['DOD'] - full_merged_data['DOB']).dt.da
```

In [69]:
```
full_merged_data.head()
```

Out[69]:

	BeneID	ClaimID	ClaimStartDt	ClaimEndDt	Provider	InscClaimAmtReimbursed
0	BENE11001	CLM46614	2009-04-12	2009-04-18	PRV55912	26000
1	BENE11001	CLM66048	2009-08-31	2009-09-02	PRV55907	5000
2	BENE11001	CLM68358	2009-09-17	2009-09-20	PRV56046	5000
3	BENE11011	CLM38412	2009-02-14	2009-02-22	PRV52405	5000
4	BENE11014	CLM63689	2009-08-13	2009-08-30	PRV56614	10000

5 rows × 59 columns

In [70]:
```
full_merged_data.describe()
```

Out[70]:

	ClaimStartDt	ClaimEndDt	InscClaimAmtReimbursed	AdmissionDt
count	558211	558211	558211.000000	40474
mean	2009-06-24 23:39:21.603766528	2009-06-26 17:07:35.601913600	997.012133	2009-06-19 17:38:12.493946880
min	2008-11-27 00:00:00	2008-12-28 00:00:00	0.000000	2008-11-27 00:00:00
25%	2009-03-27 00:00:00	2009-03-29 00:00:00	40.000000	2009-03-20 00:00:00
50%	2009-06-23 00:00:00	2009-06-24 00:00:00	80.000000	2009-06-16 00:00:00
75%	2009-09-22 00:00:00	2009-09-23 00:00:00	300.000000	2009-09-17 00:00:00
max	2009-12-31 00:00:00	2009-12-31 00:00:00	125000.000000	2009-12-31 00:00:00
std	NaN	NaN	3821.534891	NaN

8 rows × 39 columns

In [71]:
```
(full_merged_data.isnull().sum()/(len(full_merged_data)))*100
```

Out[71]:

BeneID	0.000000
ClaimID	0.000000
ClaimStartDt	0.000000
ClaimEndDt	0.000000
Provider	0.000000
InscClaimAmtReimbursed	0.000000
AttendingPhysician	0.270149
OperatingPhysician	79.497538
OtherPhysician	64.218548
AdmissionDt	92.749337
ClmAdmitDiagnosisCode	73.863109
DeductibleAmtPaid	0.161050
DischargeDt	92.749337
DiagnosisGroupCode	92.749337
ClmDiagnosisCode_1	1.872589
ClmDiagnosisCode_2	35.041588
ClmDiagnosisCode_3	56.458221
ClmDiagnosisCode_4	70.524407
ClmDiagnosisCode_5	79.949517
ClmDiagnosisCode_6	84.881702
ClmDiagnosisCode_7	88.144805
ClmDiagnosisCode_8	90.425843
ClmDiagnosisCode_9	92.509105
ClmDiagnosisCode_10	99.102490
ClmProcedureCode_1	95.824160
ClmProcedureCode_2	99.016501
ClmProcedureCode_3	99.826410
ClmProcedureCode_4	99.978861
ClmProcedureCode_5	99.998388
ClmProcedureCode_6	100.000000
DOB	0.000000
DOD	0.000000
Gender	0.000000
Race	0.000000
RenalDiseaseIndicator	0.000000
State	0.000000
County	0.000000
NoOfMonths_PartACov	0.000000
NoOfMonths_PartBCov	0.000000
ChronicCond_Alzheimer	0.000000
ChronicCond_Heartfailure	0.000000
ChronicCond_KidneyDisease	0.000000
ChronicCond_Cancer	0.000000
ChronicCond_ObstrPulmonary	0.000000
ChronicCond_Depression	0.000000
ChronicCond_Diabetes	0.000000
ChronicCond_IschemicHeart	0.000000
ChronicCond_Osteoporasis	0.000000
ChronicCond_rheumatoidarthritis	0.000000
ChronicCond_stroke	0.000000
IPAnnualReimbursementAmt	0.000000
IPAnnualDeductibleAmt	0.000000
OPAnnualReimbursementAmt	0.000000
OPAnnualDeductibleAmt	0.000000
PotentialFraud	0.000000
LengthOfStay	92.749337

```
MostRecentClaimEndDt          0.000000
ClaimDuration                 0.000000
Age                           0.000000
dtype: float64
```

In [72]:
```
full_merged_data['LengthOfStay'] = full_merged_data['LengthOfStay'].fillna(0)
```

In [73]:
```
(full_merged_data.isnull().sum()/(len(full_merged_data)))*100
```

Out[73]:

BeneID	0.000000
ClaimID	0.000000
ClaimStartDt	0.000000
ClaimEndDt	0.000000
Provider	0.000000
InscClaimAmtReimbursed	0.000000
AttendingPhysician	0.270149
OperatingPhysician	79.497538
OtherPhysician	64.218548
AdmissionDt	92.749337
ClmAdmitDiagnosisCode	73.863109
DeductibleAmtPaid	0.161050
DischargeDt	92.749337
DiagnosisGroupCode	92.749337
ClmDiagnosisCode_1	1.872589
ClmDiagnosisCode_2	35.041588
ClmDiagnosisCode_3	56.458221
ClmDiagnosisCode_4	70.524407
ClmDiagnosisCode_5	79.949517
ClmDiagnosisCode_6	84.881702
ClmDiagnosisCode_7	88.144805
ClmDiagnosisCode_8	90.425843
ClmDiagnosisCode_9	92.509105
ClmDiagnosisCode_10	99.102490
ClmProcedureCode_1	95.824160
ClmProcedureCode_2	99.016501
ClmProcedureCode_3	99.826410
ClmProcedureCode_4	99.978861
ClmProcedureCode_5	99.998388
ClmProcedureCode_6	100.000000
DOB	0.000000
DOD	0.000000
Gender	0.000000
Race	0.000000
RenalDiseaseIndicator	0.000000
State	0.000000
County	0.000000
NoOfMonths_PartACov	0.000000
NoOfMonths_PartBCov	0.000000
ChronicCond_Alzheimer	0.000000
ChronicCond_Heartfailure	0.000000
ChronicCond_KidneyDisease	0.000000
ChronicCond_Cancer	0.000000
ChronicCond_ObstrPulmonary	0.000000
ChronicCond_Depression	0.000000
ChronicCond_Diabetes	0.000000
ChronicCond_IschemicHeart	0.000000
ChronicCond_Osteoporasis	0.000000
ChronicCond_rheumatoidarthritis	0.000000
ChronicCond_stroke	0.000000
IPAnnualReimbursementAmt	0.000000
IPAnnualDeductibleAmt	0.000000
OPAnnualReimbursementAmt	0.000000
OPAnnualDeductibleAmt	0.000000
PotentialFraud	0.000000
LengthOfStay	0.000000

```
MostRecentClaimEndDt              0.000000
ClaimDuration                     0.000000
Age                               0.000000
dtype: float64
```

In [74]:
```
full_merged_data['DischargeDt'] = full_merged_data['DischargeDt'].fillna(0)
full_merged_data['AdmissionDt'] = full_merged_data['AdmissionDt'].fillna(0)
```

In [75]:
```
(full_merged_data.isnull().sum()/(len(full_merged_data)))*100
```

Out[75]:

BeneID	0.000000
ClaimID	0.000000
ClaimStartDt	0.000000
ClaimEndDt	0.000000
Provider	0.000000
InscClaimAmtReimbursed	0.000000
AttendingPhysician	0.270149
OperatingPhysician	79.497538
OtherPhysician	64.218548
AdmissionDt	0.000000
ClmAdmitDiagnosisCode	73.863109
DeductibleAmtPaid	0.161050
DischargeDt	0.000000
DiagnosisGroupCode	92.749337
ClmDiagnosisCode_1	1.872589
ClmDiagnosisCode_2	35.041588
ClmDiagnosisCode_3	56.458221
ClmDiagnosisCode_4	70.524407
ClmDiagnosisCode_5	79.949517
ClmDiagnosisCode_6	84.881702
ClmDiagnosisCode_7	88.144805
ClmDiagnosisCode_8	90.425843
ClmDiagnosisCode_9	92.509105
ClmDiagnosisCode_10	99.102490
ClmProcedureCode_1	95.824160
ClmProcedureCode_2	99.016501
ClmProcedureCode_3	99.826410
ClmProcedureCode_4	99.978861
ClmProcedureCode_5	99.998388
ClmProcedureCode_6	100.000000
DOB	0.000000
DOD	0.000000
Gender	0.000000
Race	0.000000
RenalDiseaseIndicator	0.000000
State	0.000000
County	0.000000
NoOfMonths_PartACov	0.000000
NoOfMonths_PartBCov	0.000000
ChronicCond_Alzheimer	0.000000
ChronicCond_Heartfailure	0.000000
ChronicCond_KidneyDisease	0.000000
ChronicCond_Cancer	0.000000
ChronicCond_ObstrPulmonary	0.000000
ChronicCond_Depression	0.000000
ChronicCond_Diabetes	0.000000
ChronicCond_IschemicHeart	0.000000
ChronicCond_Osteoporasis	0.000000
ChronicCond_rheumatoidarthritis	0.000000
ChronicCond_stroke	0.000000
IPAnnualReimbursementAmt	0.000000
IPAnnualDeductibleAmt	0.000000
OPAnnualReimbursementAmt	0.000000
OPAnnualDeductibleAmt	0.000000
PotentialFraud	0.000000
LengthOfStay	0.000000

```
MostRecentClaimEndDt              0.000000
ClaimDuration                     0.000000
Age                               0.000000
dtype: float64
```

In [76]:
```
full_merged_data.head()
```

Out[76]:

	BeneID	ClaimID	ClaimStartDt	ClaimEndDt	Provider	InscClaimAmtReimbursed
0	BENE11001	CLM46614	2009-04-12	2009-04-18	PRV55912	26000
1	BENE11001	CLM66048	2009-08-31	2009-09-02	PRV55907	5000
2	BENE11001	CLM68358	2009-09-17	2009-09-20	PRV56046	5000
3	BENE11011	CLM38412	2009-02-14	2009-02-22	PRV52405	5000
4	BENE11014	CLM63689	2009-08-13	2009-08-30	PRV56614	10000

5 rows × 59 columns

In [77]:
```
# Count the number of non-null diagnosis codes for each claim
full_merged_data['NumDiagnosisCodes'] = full_merged_data[['ClmDiagnosisCode_1', 'Cl

# Count the number of non-null procedure codes for each claim
full_merged_data['NumProcedureCodes'] = full_merged_data[['ClmProcedureCode_1', 'Cl
```

In [78]:
```
full_merged_data.head()
```

Out[78]:

	BeneID	ClaimID	ClaimStartDt	ClaimEndDt	Provider	InscClaimAmtReimbursed
0	BENE11001	CLM46614	2009-04-12	2009-04-18	PRV55912	26000
1	BENE11001	CLM66048	2009-08-31	2009-09-02	PRV55907	5000
2	BENE11001	CLM68358	2009-09-17	2009-09-20	PRV56046	5000
3	BENE11011	CLM38412	2009-02-14	2009-02-22	PRV52405	5000
4	BENE11014	CLM63689	2009-08-13	2009-08-30	PRV56614	10000

5 rows × 61 columns

In [79]:
```
full_merged_data.describe()
```

Out[79]:

	ClaimStartDt	ClaimEndDt	InscClaimAmtReimbursed	DeductibleAmtPaid
count	558211	558211	558211.000000	557312.000000
mean	2009-06-24 23:39:21.603766528	2009-06-26 17:07:35.601913600	997.012133	78.421085
min	2008-11-27 00:00:00	2008-12-28 00:00:00	0.000000	0.000000
25%	2009-03-27 00:00:00	2009-03-29 00:00:00	40.000000	0.000000
50%	2009-06-23 00:00:00	2009-06-24 00:00:00	80.000000	0.000000
75%	2009-09-22 00:00:00	2009-09-23 00:00:00	300.000000	0.000000
max	2009-12-31 00:00:00	2009-12-31 00:00:00	125000.000000	1068.000000
std	NaN	NaN	3821.534891	274.016812

8 rows × 39 columns

In [80]:
```
full_merged_data.to_csv('full_merged_data.csv', index=False)
```

In [81]:
```
# Separate the numerical and categorical columns
num_cols = full_merged_data.select_dtypes(include=['int64', 'float64']).columns.tol
cat_cols = full_merged_data.select_dtypes(exclude=['int64', 'float64', 'datetime64[

print("Categorical Variables:")
print(cat_cols)
print("Numerical Variables:")
print(num_cols)
```

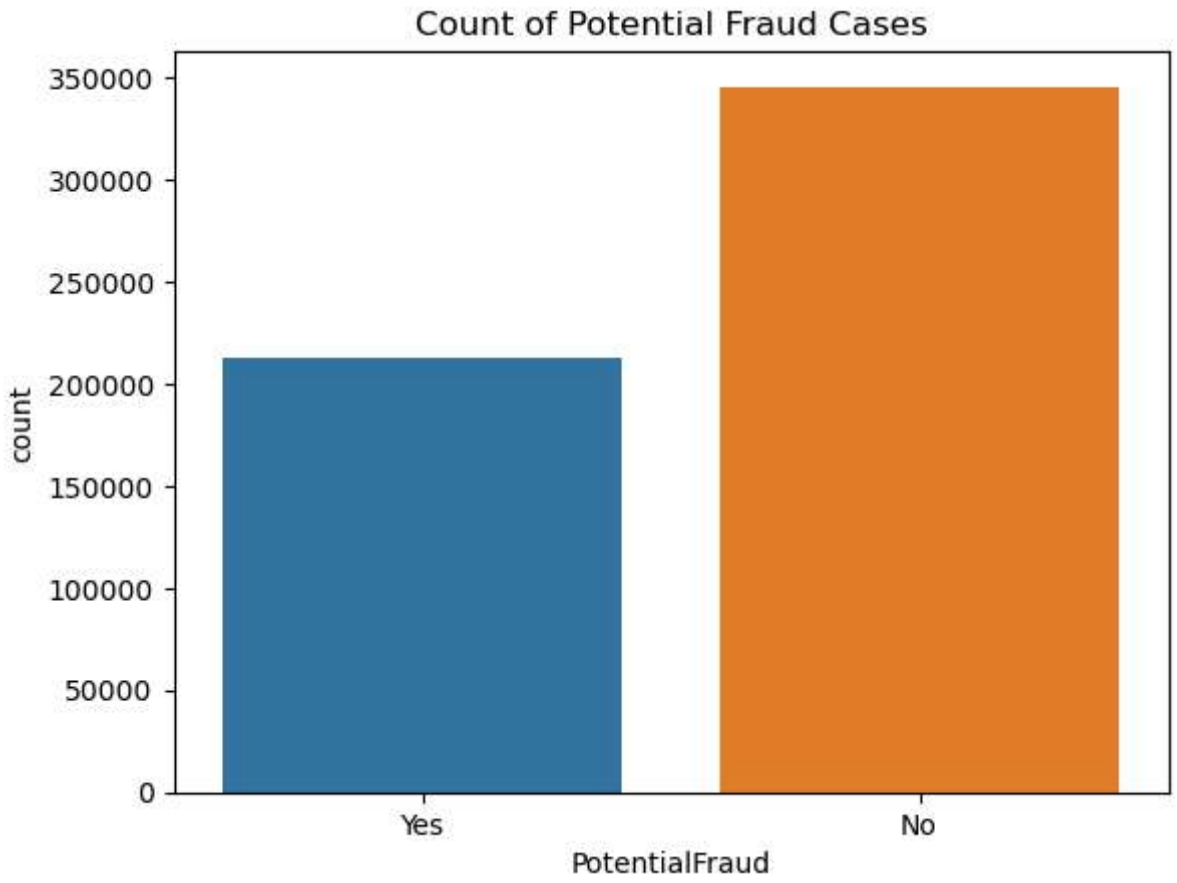
```
Categorical Variables:
['BeneID', 'ClaimID', 'Provider', 'AttendingPhysician', 'OperatingPhysician', 'Other
Physician', 'AdmissionDt', 'ClmAdmitDiagnosisCode', 'DischargeDt', 'DiagnosisGroupCo
de', 'ClmDiagnosisCode_1', 'ClmDiagnosisCode_2', 'ClmDiagnosisCode_3', 'ClmDiagnosis
Code_4', 'ClmDiagnosisCode_5', 'ClmDiagnosisCode_6', 'ClmDiagnosisCode_7', 'ClmDiagn
osisCode_8', 'ClmDiagnosisCode_9', 'ClmDiagnosisCode_10', 'RenalDiseaseIndicator',
'PotentialFraud']
Numerical Variables:
['InscClaimAmtReimbursed', 'DeductibleAmtPaid', 'ClmProcedureCode_1', 'ClmProcedureC
ode_2', 'ClmProcedureCode_3', 'ClmProcedureCode_4', 'ClmProcedureCode_5', 'ClmProced
ureCode_6', 'Gender', 'Race', 'State', 'County', 'NoOfMonths_PartACov', 'NoOfMonths_
PartBCov', 'ChronicCond_Alzheimer', 'ChronicCond_Heartfailure', 'ChronicCond_KidneyD
isease', 'ChronicCond_Cancer', 'ChronicCond_ObstrPulmonary', 'ChronicCond_Depressio
n', 'ChronicCond_Diabetes', 'ChronicCond_IschemicHeart', 'ChronicCond_Osteoporasis',
'ChronicCond_rheumatoidarthritis', 'ChronicCond_stroke', 'IPAnnualReimbursementAmt',
'IPAnnualDeductibleAmt', 'OPAnnualReimbursementAmt', 'OPAnnualDeductibleAmt', 'Lengt
hOfStay', 'ClaimDuration', 'Age', 'NumDiagnosisCodes', 'NumProcedureCodes']
```

In [82]:
```
full_merged_data.info()
```

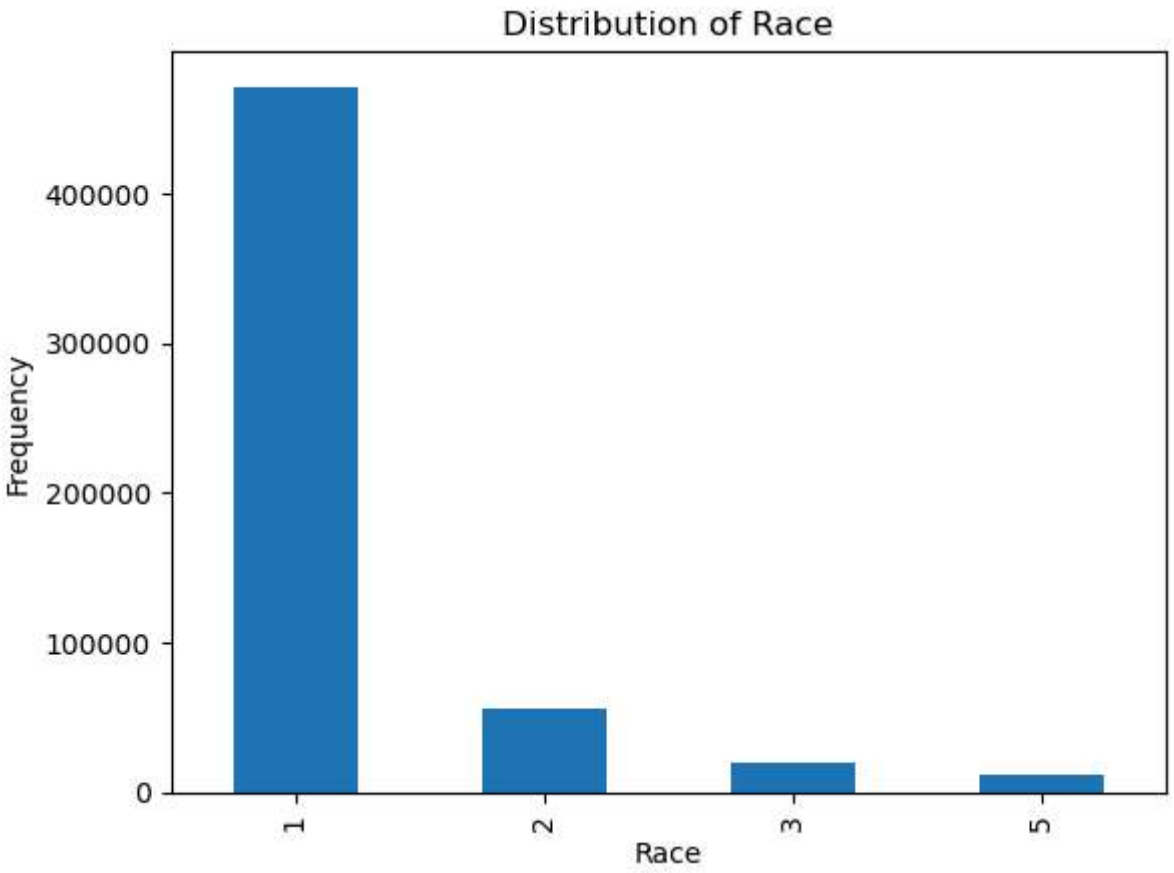
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 558211 entries, 0 to 558210
Data columns (total 61 columns):
 #   Column                            Non-Null Count   Dtype
---  ------                            --------------   -----
 0   BeneID                            558211 non-null  object
 1   ClaimID                           558211 non-null  object
 2   ClaimStartDt                      558211 non-null  datetime64[ns]
 3   ClaimEndDt                        558211 non-null  datetime64[ns]
 4   Provider                          558211 non-null  object
 5   InscClaimAmtReimbursed            558211 non-null  int64
 6   AttendingPhysician                556703 non-null  object
 7   OperatingPhysician                114447 non-null  object
 8   OtherPhysician                    199736 non-null  object
 9   AdmissionDt                       558211 non-null  object
 10  ClmAdmitDiagnosisCode             145899 non-null  object
 11  DeductibleAmtPaid                 557312 non-null  float64
 12  DischargeDt                       558211 non-null  object
 13  DiagnosisGroupCode                40474 non-null   object
 14  ClmDiagnosisCode_1                547758 non-null  object
 15  ClmDiagnosisCode_2                362605 non-null  object
 16  ClmDiagnosisCode_3                243055 non-null  object
 17  ClmDiagnosisCode_4                164536 non-null  object
 18  ClmDiagnosisCode_5                111924 non-null  object
 19  ClmDiagnosisCode_6                84392 non-null   object
 20  ClmDiagnosisCode_7                66177 non-null   object
 21  ClmDiagnosisCode_8                53444 non-null   object
 22  ClmDiagnosisCode_9                41815 non-null   object
 23  ClmDiagnosisCode_10               5010 non-null    object
 24  ClmProcedureCode_1                23310 non-null   float64
 25  ClmProcedureCode_2                5490 non-null    float64
 26  ClmProcedureCode_3                969 non-null     float64
 27  ClmProcedureCode_4                118 non-null     float64
 28  ClmProcedureCode_5                9 non-null       float64
 29  ClmProcedureCode_6                0 non-null       float64
 30  DOB                               558211 non-null  datetime64[ns]
 31  DOD                               558211 non-null  datetime64[ns]
 32  Gender                            558211 non-null  int64
 33  Race                              558211 non-null  int64
 34  RenalDiseaseIndicator             558211 non-null  object
 35  State                             558211 non-null  int64
 36  County                            558211 non-null  int64
 37  NoOfMonths_PartACov               558211 non-null  int64
 38  NoOfMonths_PartBCov               558211 non-null  int64
 39  ChronicCond_Alzheimer             558211 non-null  int64
 40  ChronicCond_Heartfailure          558211 non-null  int64
 41  ChronicCond_KidneyDisease         558211 non-null  int64
 42  ChronicCond_Cancer                558211 non-null  int64
 43  ChronicCond_ObstrPulmonary        558211 non-null  int64
 44  ChronicCond_Depression            558211 non-null  int64
 45  ChronicCond_Diabetes              558211 non-null  int64
 46  ChronicCond_IschemicHeart         558211 non-null  int64
 47  ChronicCond_Osteoporasis          558211 non-null  int64
 48  ChronicCond_rheumatoidarthritis   558211 non-null  int64
 49  ChronicCond_stroke                558211 non-null  int64
 50  IPAnnualReimbursementAmt          558211 non-null  int64
```

```
 51  IPAnnualDeductibleAmt           558211 non-null  int64
 52  OPAnnualReimbursementAmt        558211 non-null  int64
 53  OPAnnualDeductibleAmt           558211 non-null  int64
 54  PotentialFraud                  558211 non-null  object
 55  LengthOfStay                    558211 non-null  float64
 56  MostRecentClaimEndDt            558211 non-null  datetime64[ns]
 57  ClaimDuration                   558211 non-null  int64
 58  Age                             558211 non-null  int64
 59  NumDiagnosisCodes               558211 non-null  int64
 60  NumProcedureCodes               558211 non-null  int64
dtypes: datetime64[ns](5), float64(8), int64(26), object(22)
memory usage: 259.8+ MB
```

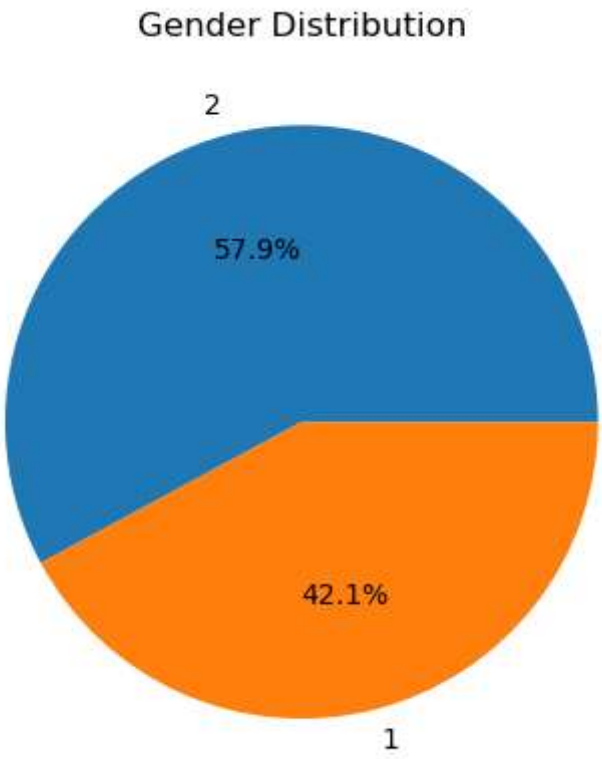
In [83]:
```
sns.countplot(x='PotentialFraud', data=full_merged_data)
plt.title('Count of Potential Fraud Cases')
plt.show()
```



In [84]:
```
full_merged_data['Race'].value_counts().plot(kind='bar')
plt.title('Distribution of Race')
plt.xlabel('Race')
plt.ylabel('Frequency')
plt.show()
```

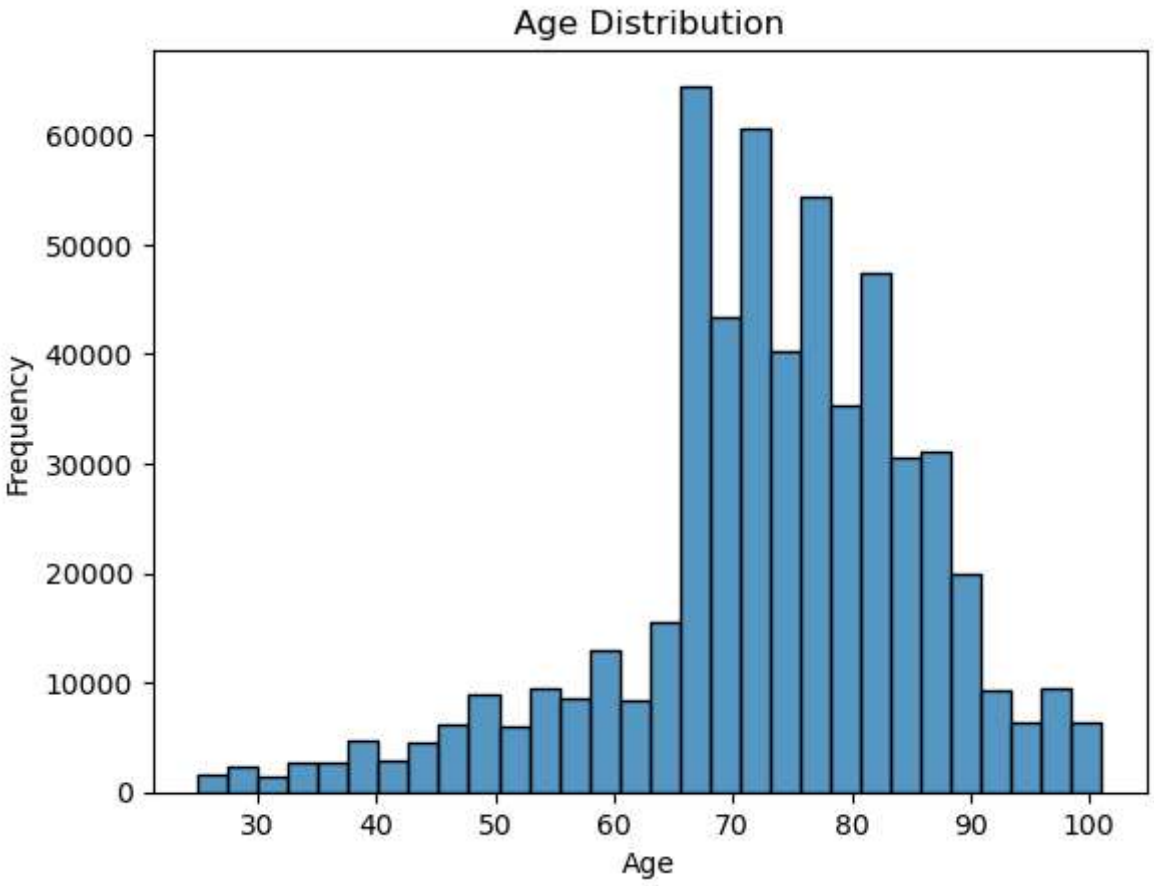


In [85]:
```
full_merged_data['Gender'].value_counts().plot(kind='pie', autopct='%1.1f%%')
plt.title('Gender Distribution')
plt.ylabel('')
plt.show()
```

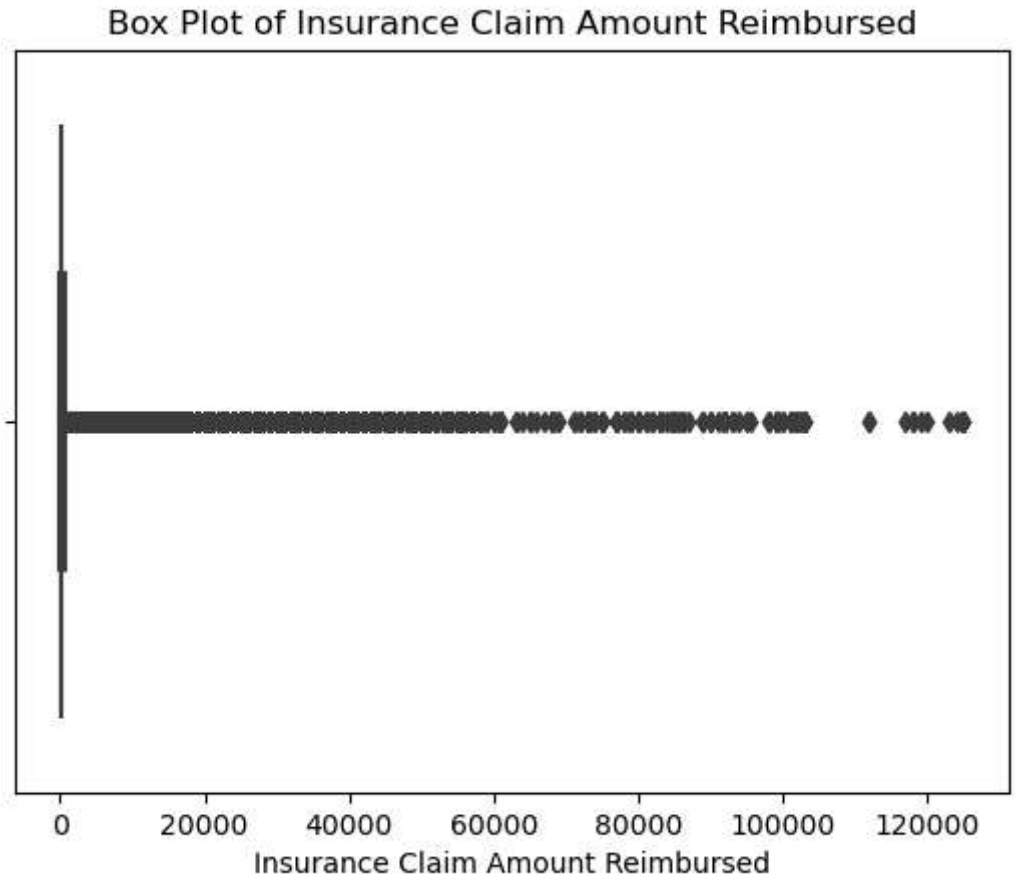


In [86]:
```
sns.histplot(full_merged_data['Age'], bins=30)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```

```
C:\Users\asus\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: u
se_inf_as_na option is deprecated and will be removed in a future version. Convert i
nf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

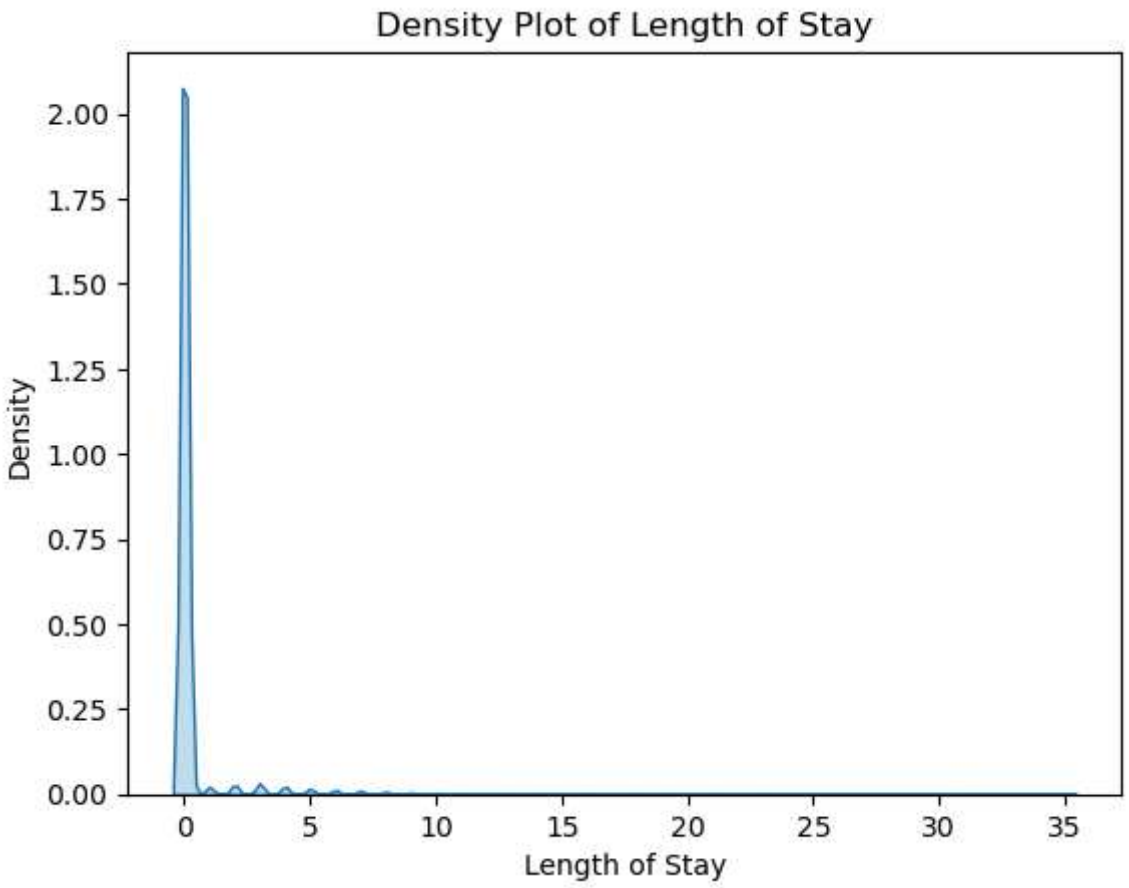


In [87]:
```
sns.boxplot(x=full_merged_data['InscClaimAmtReimbursed'])
plt.title('Box Plot of Insurance Claim Amount Reimbursed')
plt.xlabel('Insurance Claim Amount Reimbursed')
plt.show()
```

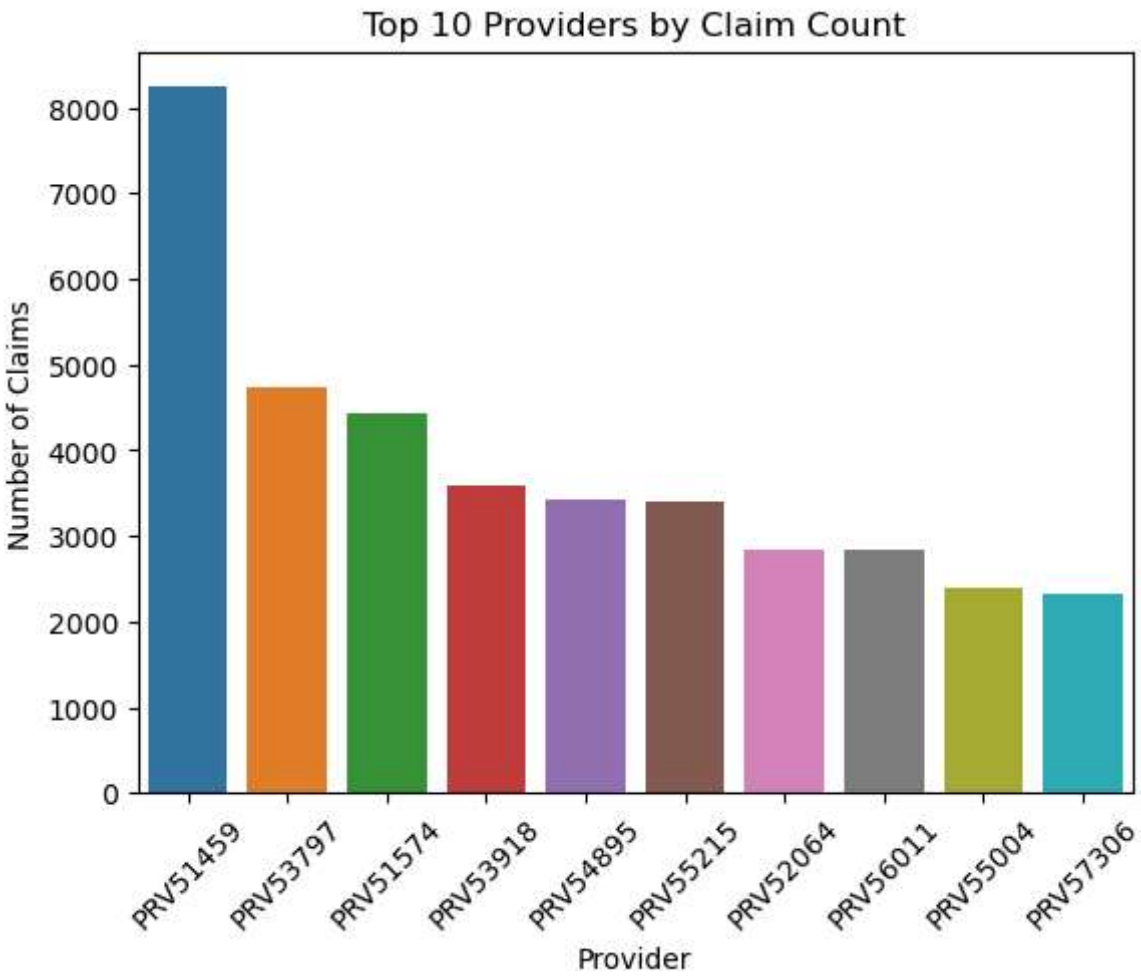


In [88]:
```
sns.kdeplot(full_merged_data['LengthOfStay'], fill=True)
plt.title('Density Plot of Length of Stay')
plt.xlabel('Length of Stay')
plt.show()
```

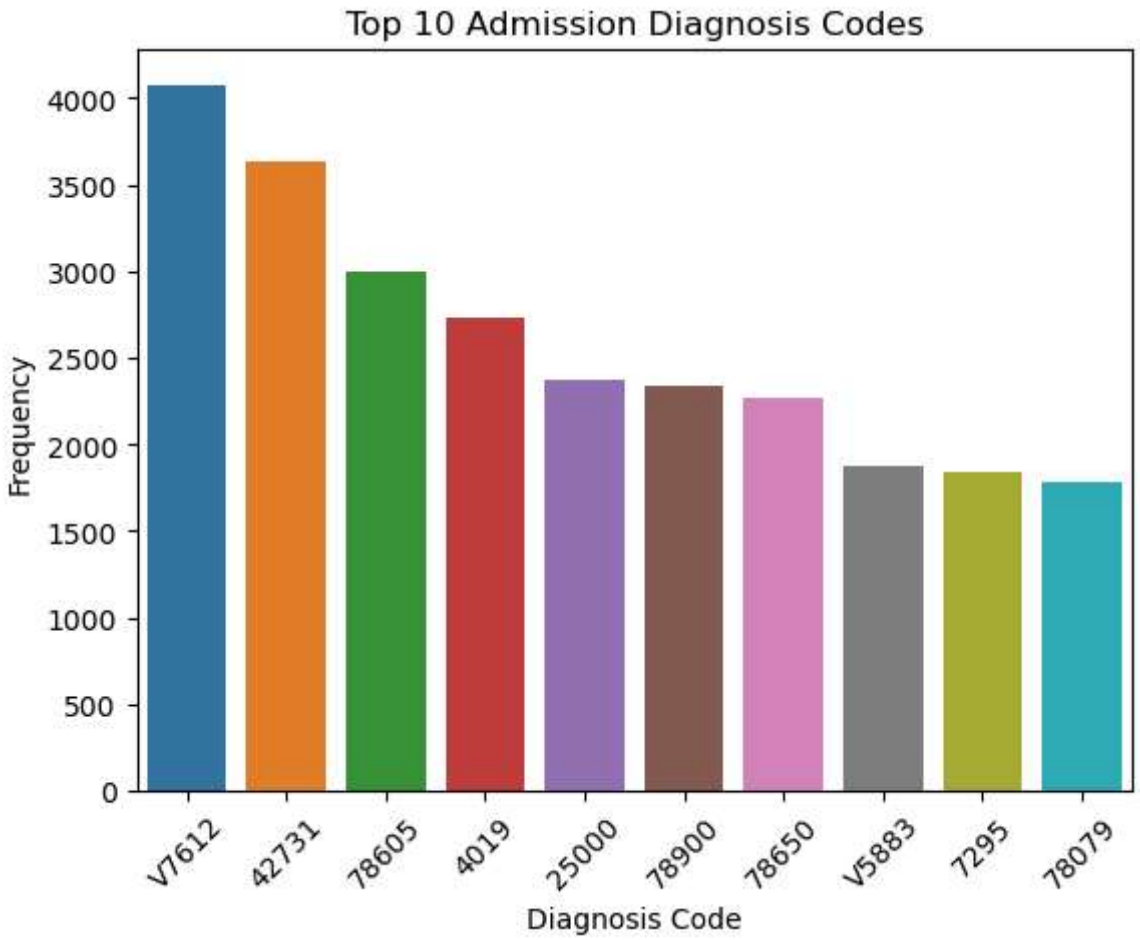
```
C:\Users\asus\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: u
se_inf_as_na option is deprecated and will be removed in a future version. Convert i
nf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



In [89]:
```
top_providers = full_merged_data['Provider'].value_counts().head(10)  # Top 10 prov
sns.barplot(x=top_providers.index, y=top_providers.values)
plt.title('Top 10 Providers by Claim Count')
plt.xlabel('Provider')
plt.ylabel('Number of Claims')
plt.xticks(rotation=45)
plt.show()
```

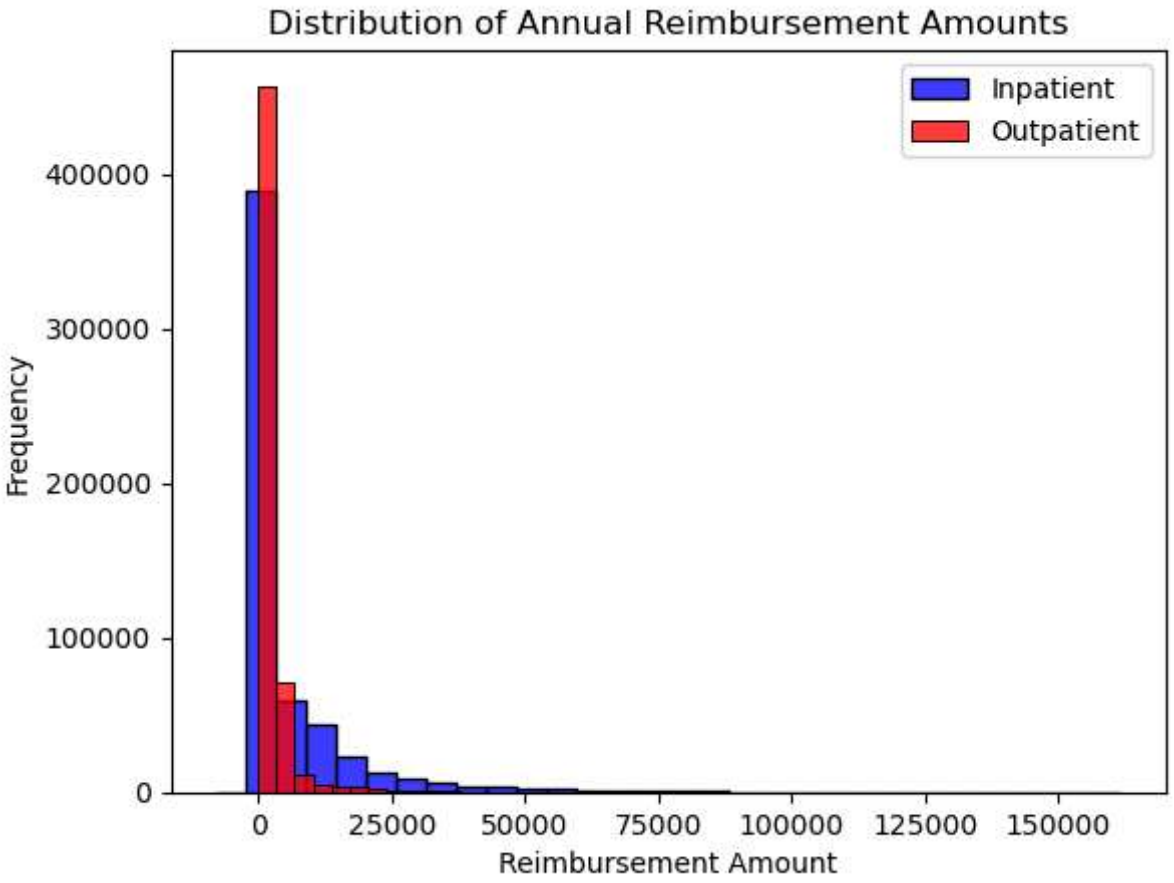


In [90]:
```
top_diagnosis_codes = full_merged_data['ClmAdmitDiagnosisCode'].value_counts().head
sns.barplot(x=top_diagnosis_codes.index, y=top_diagnosis_codes.values)
plt.title('Top 10 Admission Diagnosis Codes')
plt.xlabel('Diagnosis Code')
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.show()
```



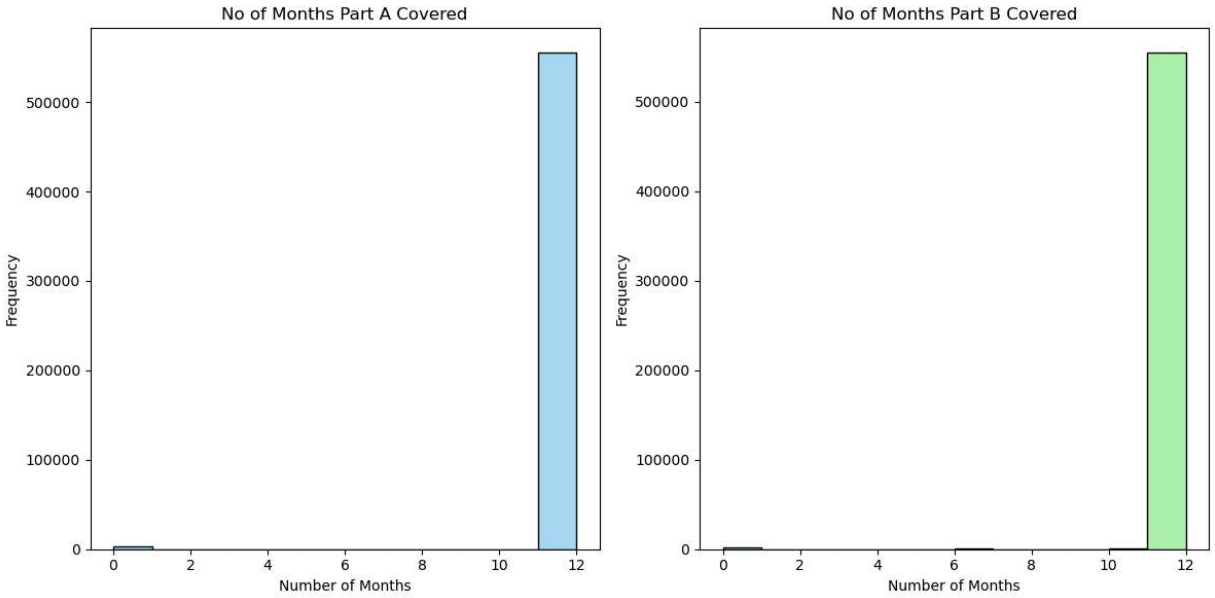
In [91]:
```
sns.histplot(full_merged_data['IPAnnualReimbursementAmt'], bins=30, color='blue', l
sns.histplot(full_merged_data['OPAnnualReimbursementAmt'], bins=30, color='red', la
plt.title('Distribution of Annual Reimbursement Amounts')
plt.xlabel('Reimbursement Amount')
plt.ylabel('Frequency')
plt.legend()
plt.show()
```

```
C:\Users\asus\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: u
se_inf_as_na option is deprecated and will be removed in a future version. Convert i
nf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\asus\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: u
se_inf_as_na option is deprecated and will be removed in a future version. Convert i
nf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

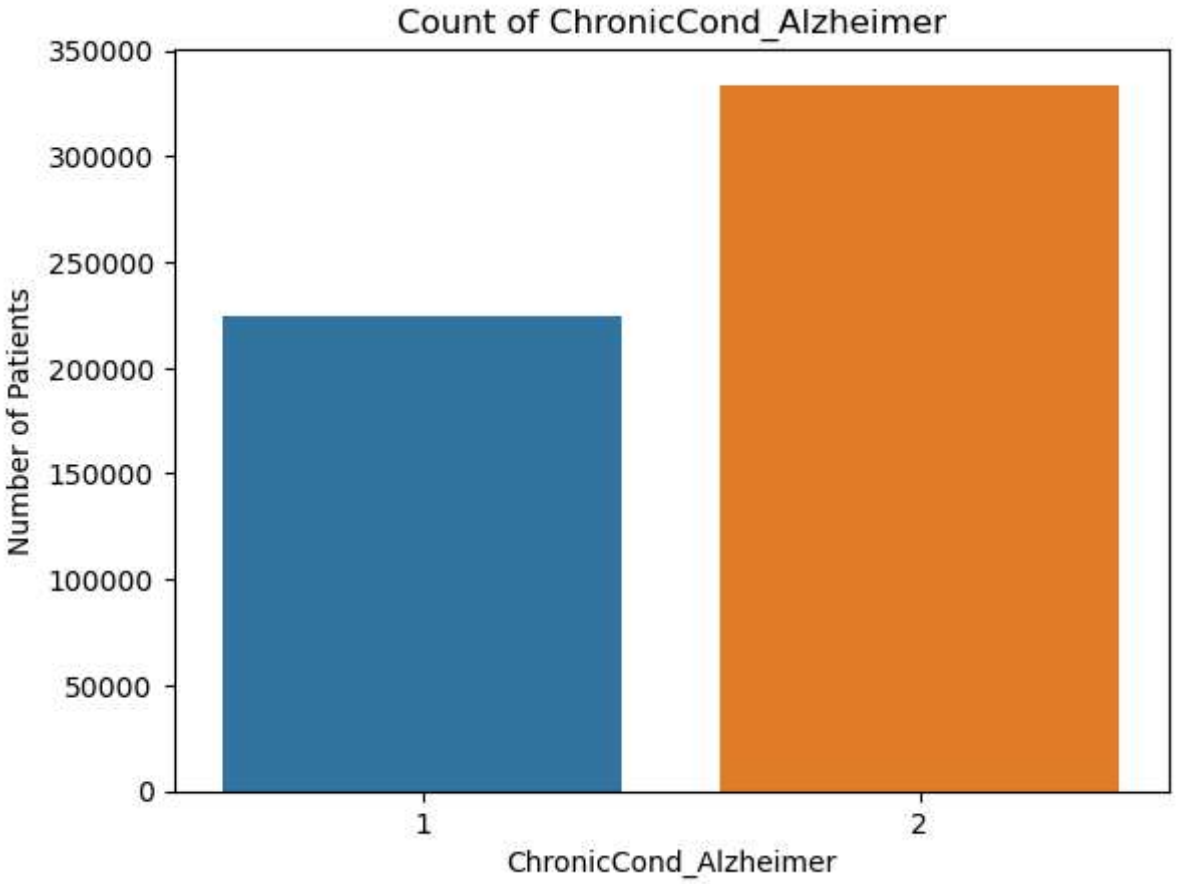


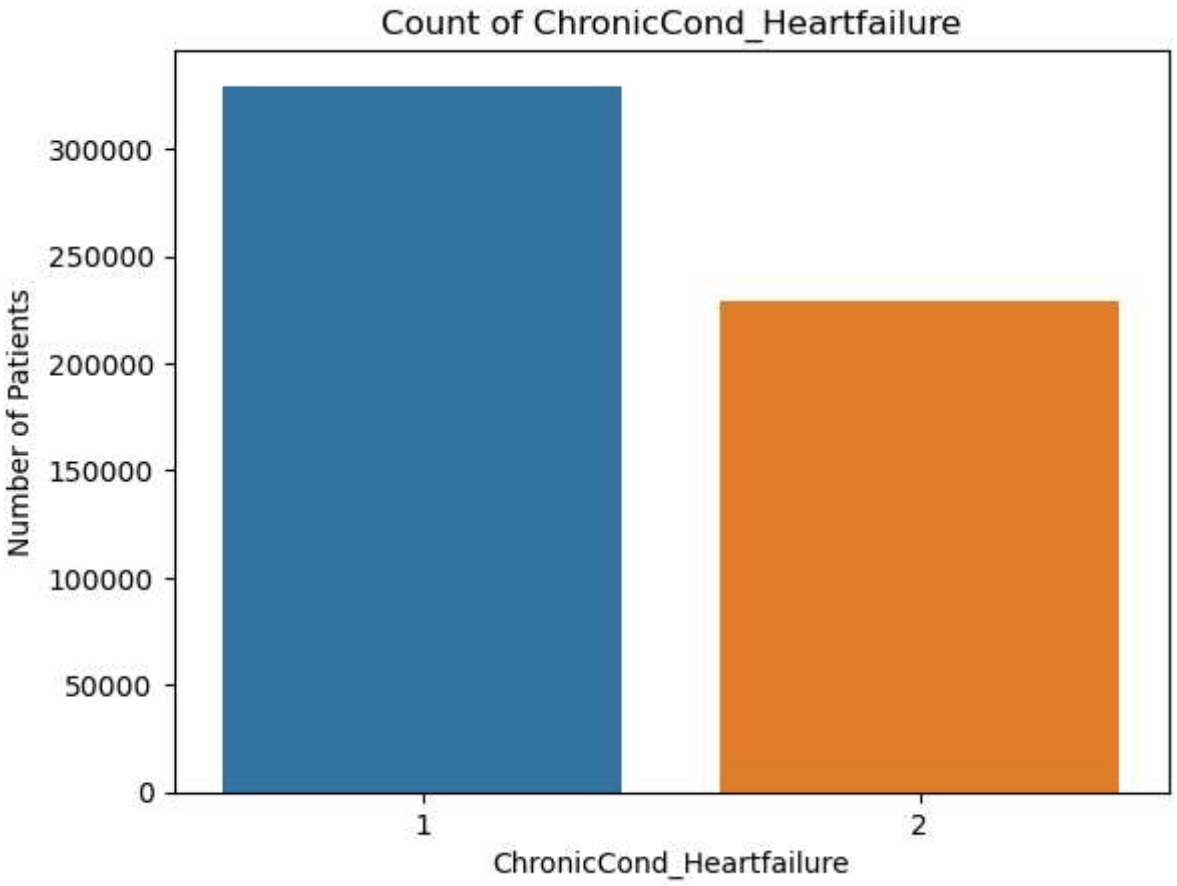
In [92]:
```
fig, ax = plt.subplots(1, 2, figsize=(12, 6))

sns.histplot(full_merged_data['NoOfMonths_PartACov'], bins=12, ax=ax[0], color='sky
ax[0].set_title('No of Months Part A Covered')
ax[0].set_xlabel('Number of Months')
ax[0].set_ylabel('Frequency')

sns.histplot(full_merged_data['NoOfMonths_PartBCov'], bins=12, ax=ax[1], color='lig
ax[1].set_title('No of Months Part B Covered')
ax[1].set_xlabel('Number of Months')
ax[1].set_ylabel('Frequency')

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

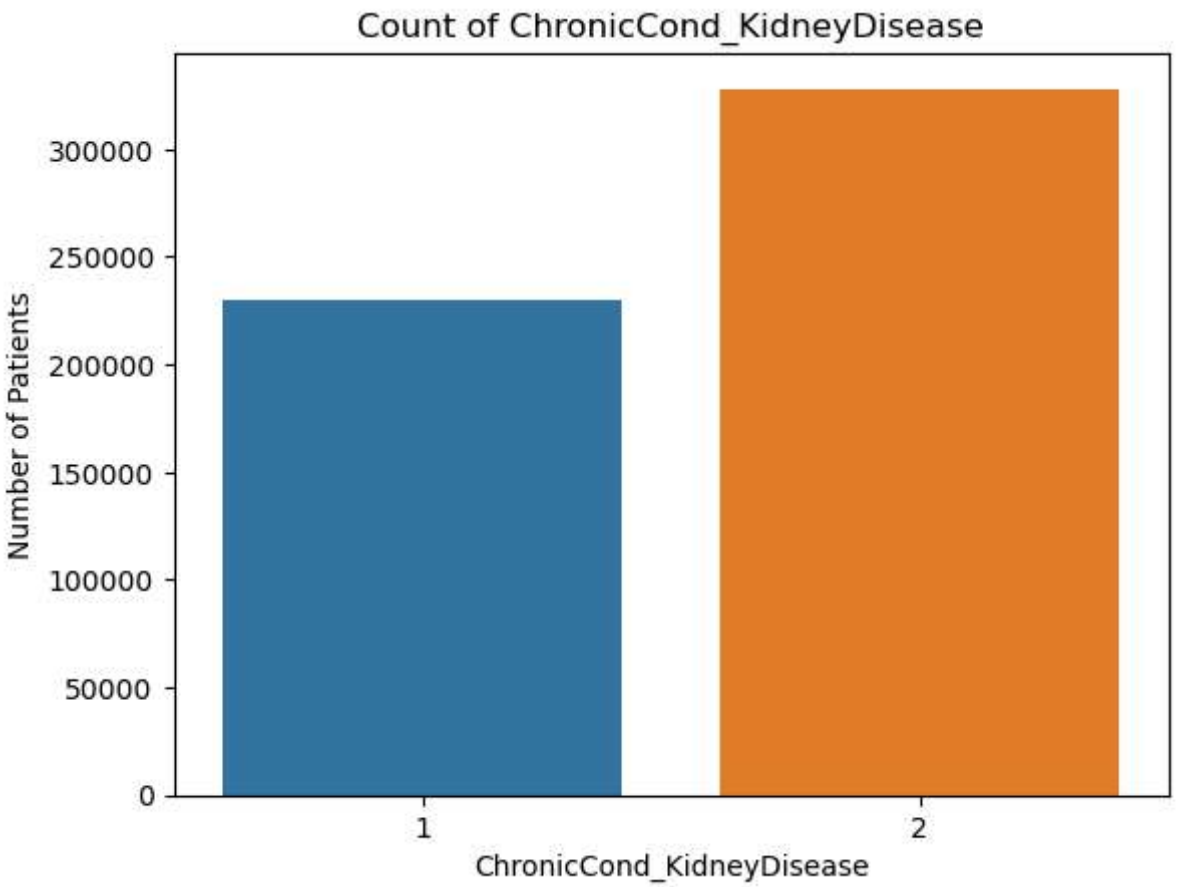
```
C:\Users\asus\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: u
se_inf_as_na option is deprecated and will be removed in a future version. Convert i
nf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\asus\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: u
se_inf_as_na option is deprecated and will be removed in a future version. Convert i
nf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

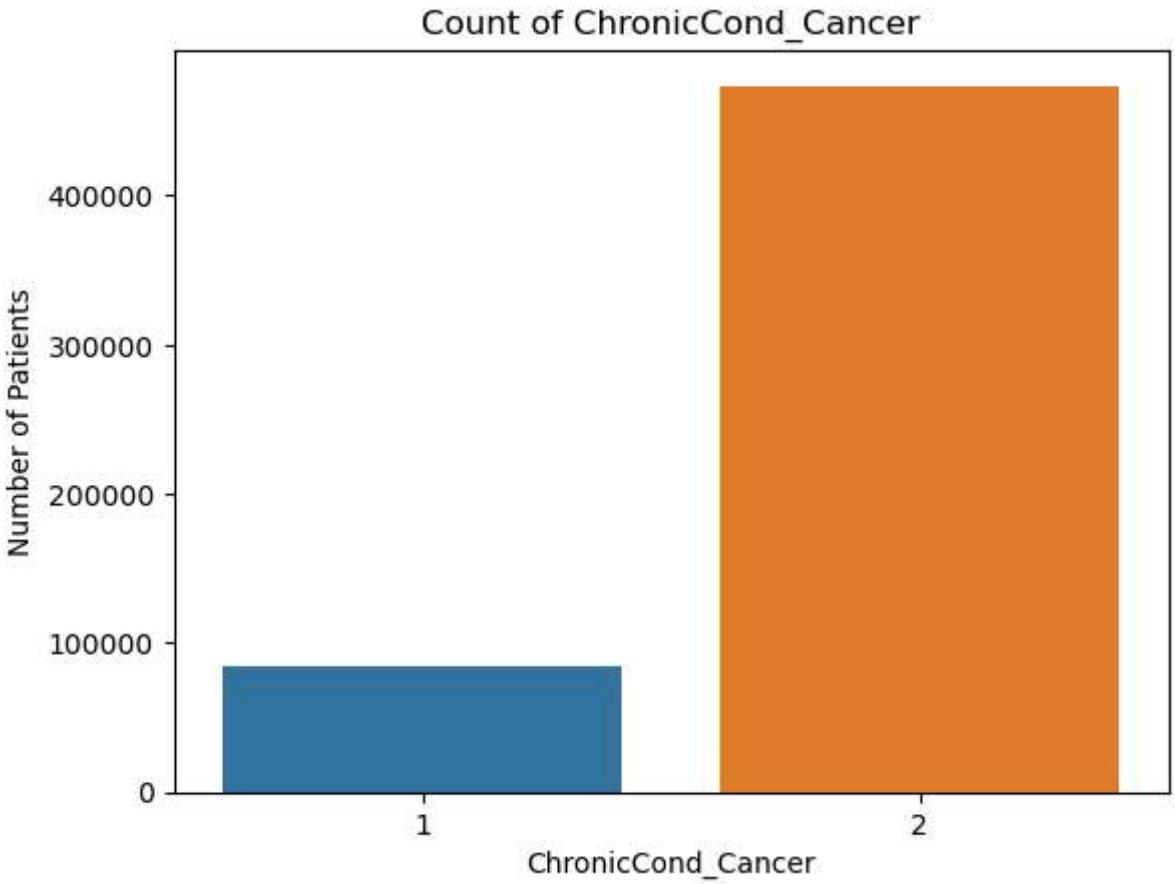


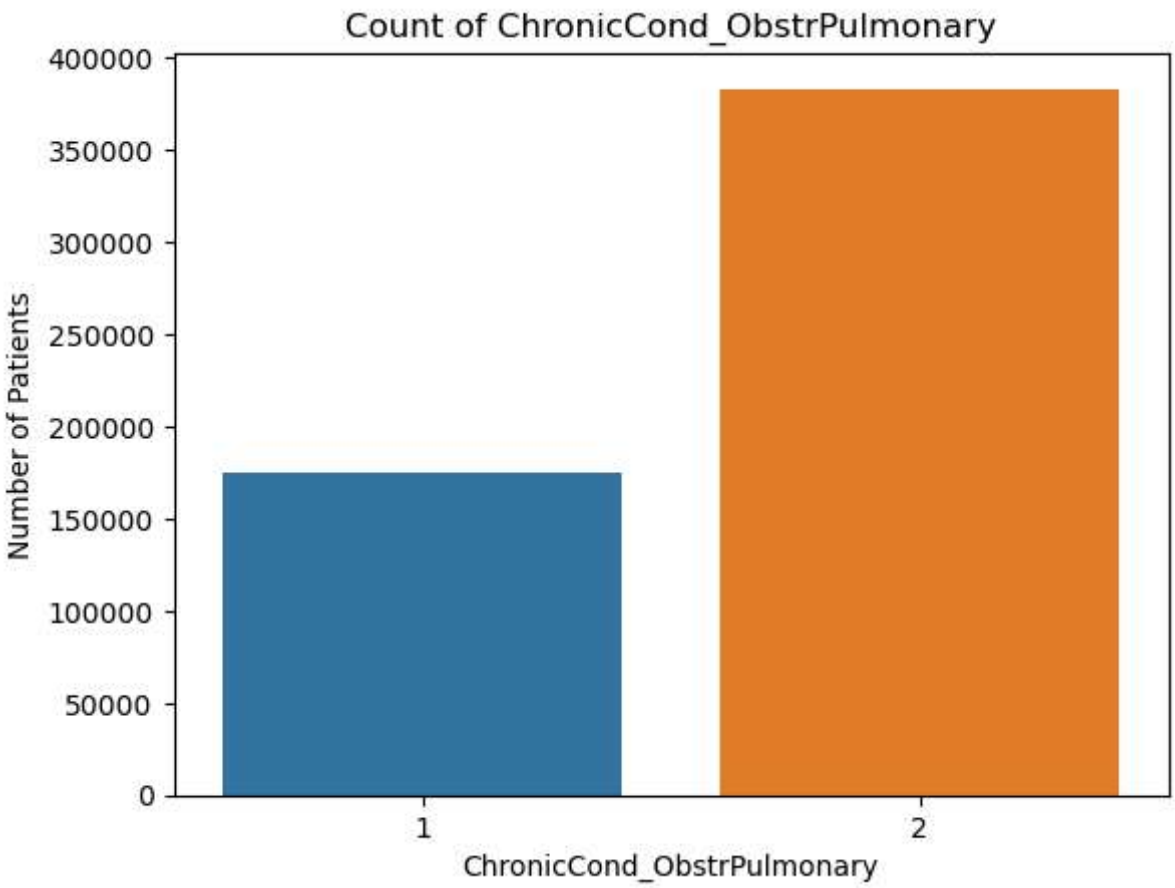
In [93]:
```
chronic_conditions = ['ChronicCond_Alzheimer', 'ChronicCond_Heartfailure', 'Chronic
for condition in chronic_conditions:
    sns.countplot(x=full_merged_data[condition])
    plt.title(f'Count of {condition}')
    plt.xlabel(condition)
    plt.ylabel('Number of Patients')
    plt.show()
```

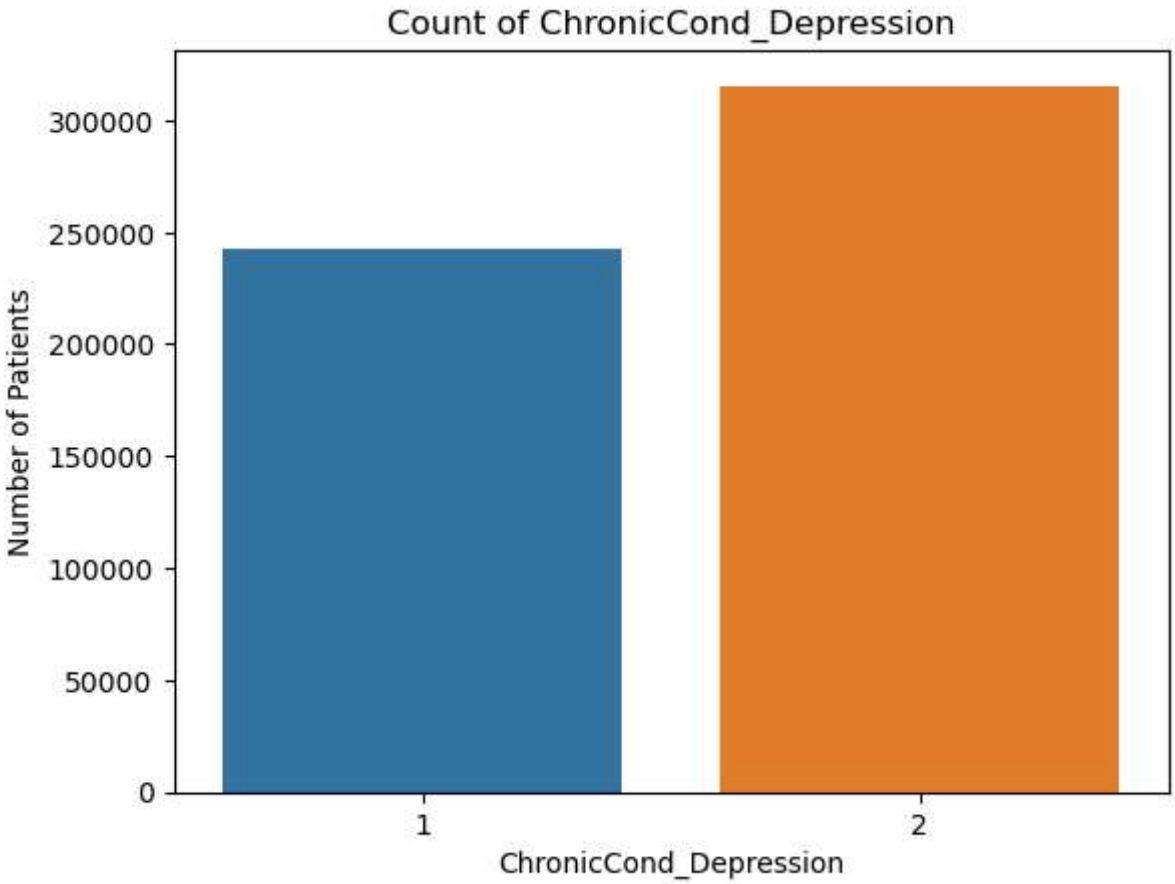


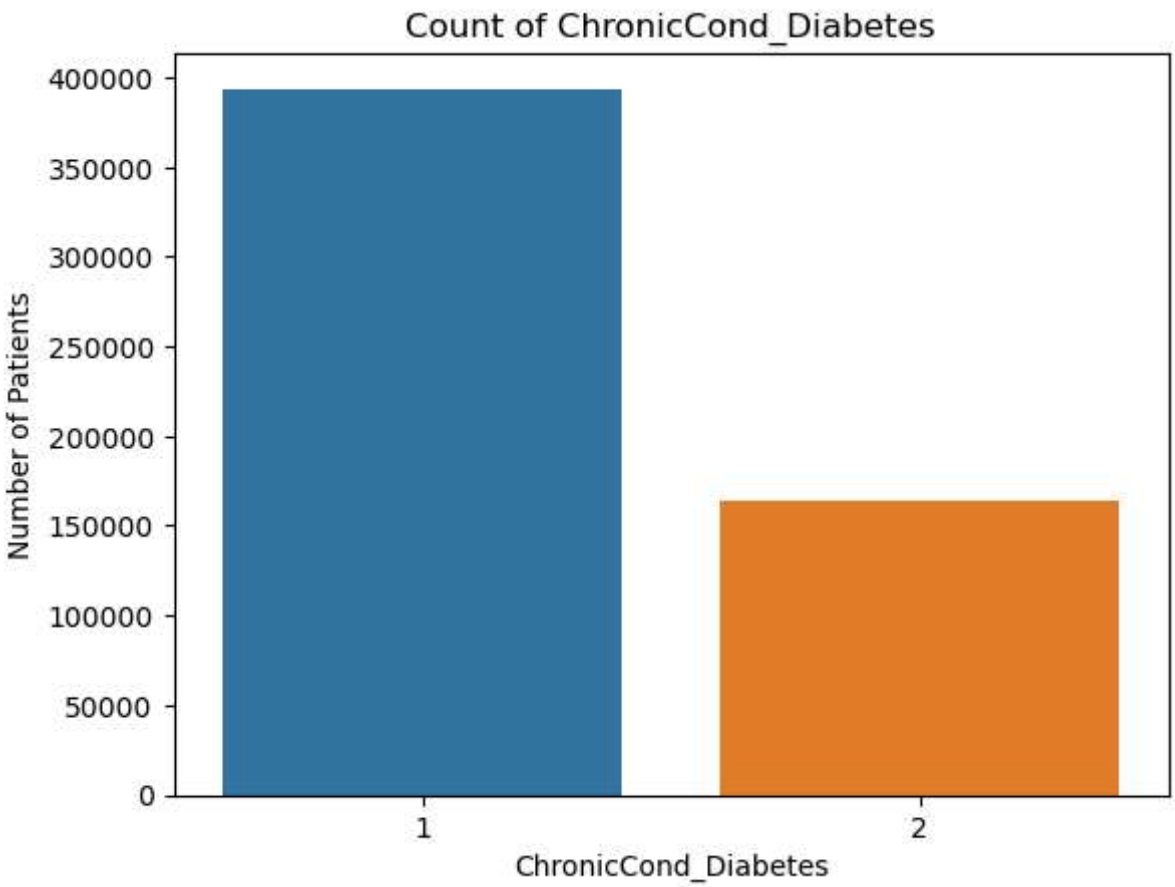

Count of ChronicCond_Heartfailure
Number of Patients
300000
250000
200000
150000
100000
50000
0
1
2
ChronicCond_Heartfailure

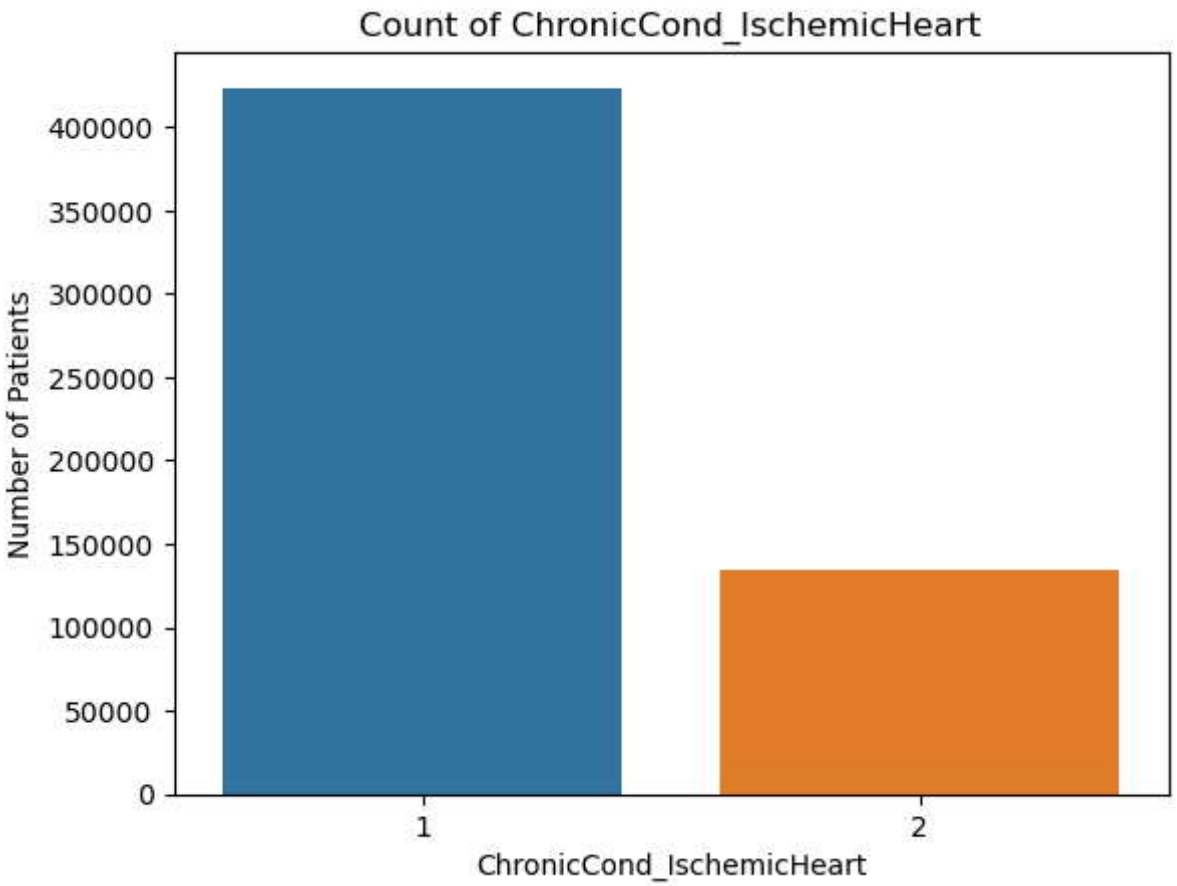

Count of ChronicCond_KidneyDisease
Number of Patients
300000
250000
200000
150000
100000
50000
0
1
2
ChronicCond_KidneyDisease

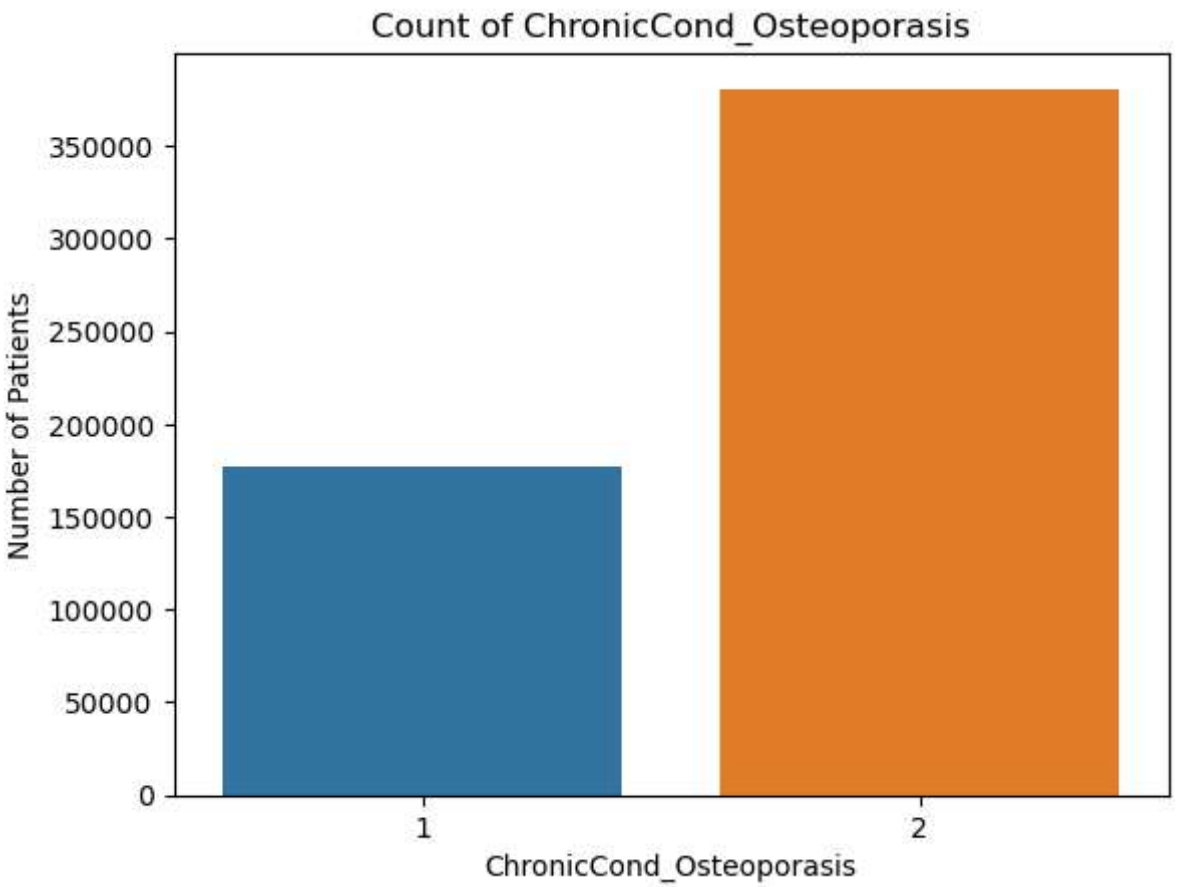

Count of ChronicCond_Cancer
Number of Patients
400000
300000
200000
100000
0
1
2
ChronicCond_Cancer

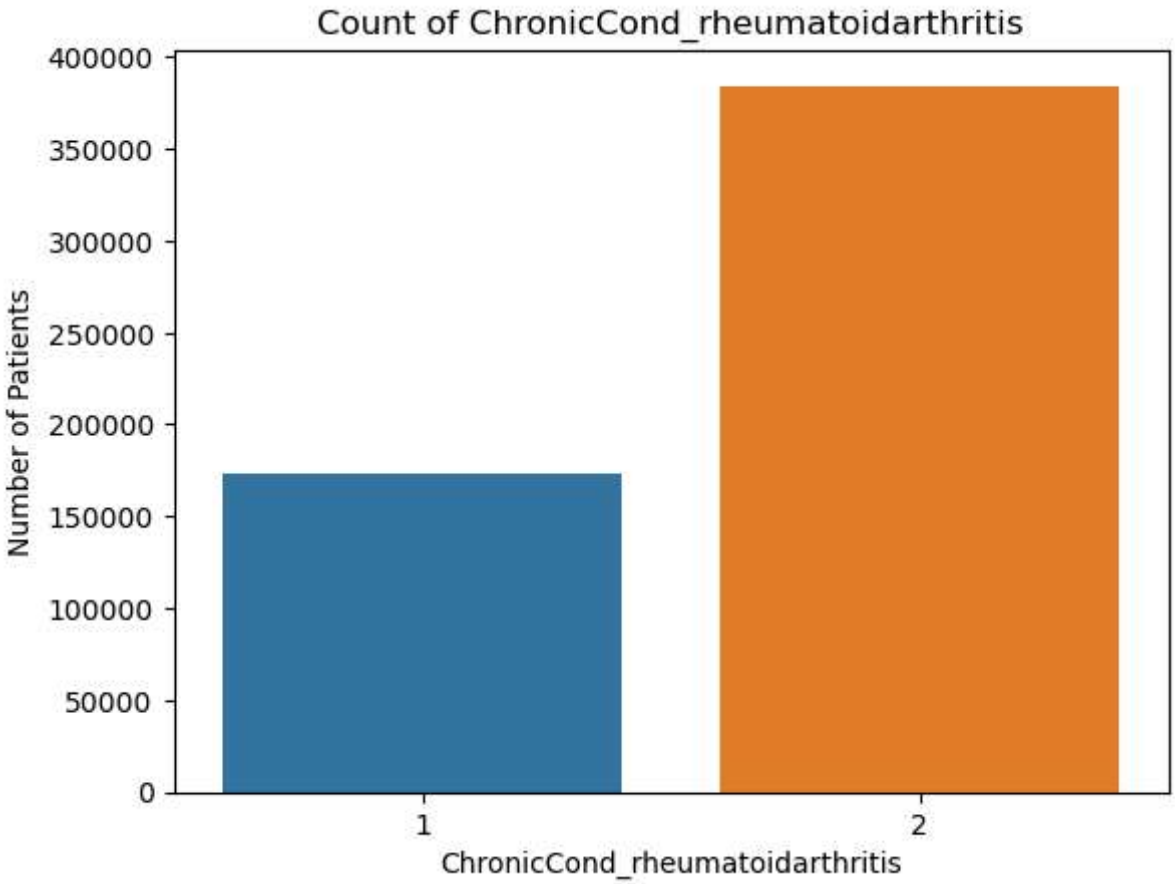

Count of ChronicCond_ObstrPulmonary
Number of Patients
400000
350000
300000
250000
200000
150000
100000
50000
0
1
2
ChronicCond_ObstrPulmonary

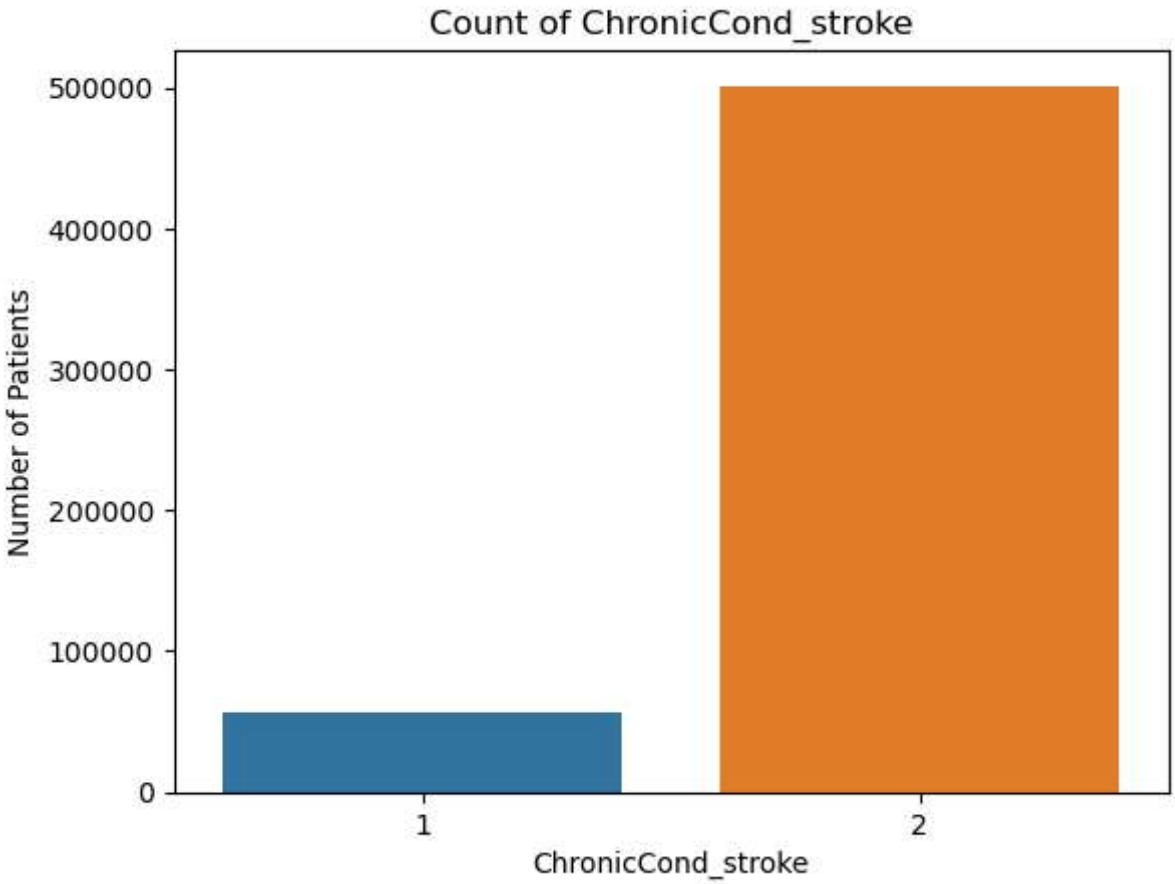

Count of ChronicCond_Depression
Number of Patients
300000
250000
200000
150000
100000
50000
0
1
2
ChronicCond_Depression


Count of ChronicCond_Diabetes
Number of Patients
400000
350000
300000
250000
200000
150000
100000
50000
0
1
2
ChronicCond_Diabetes


Count of ChronicCond_IschemicHeart
Number of Patients
400000
350000
300000
250000
200000
150000
100000
50000
0
1
2
ChronicCond_IschemicHeart

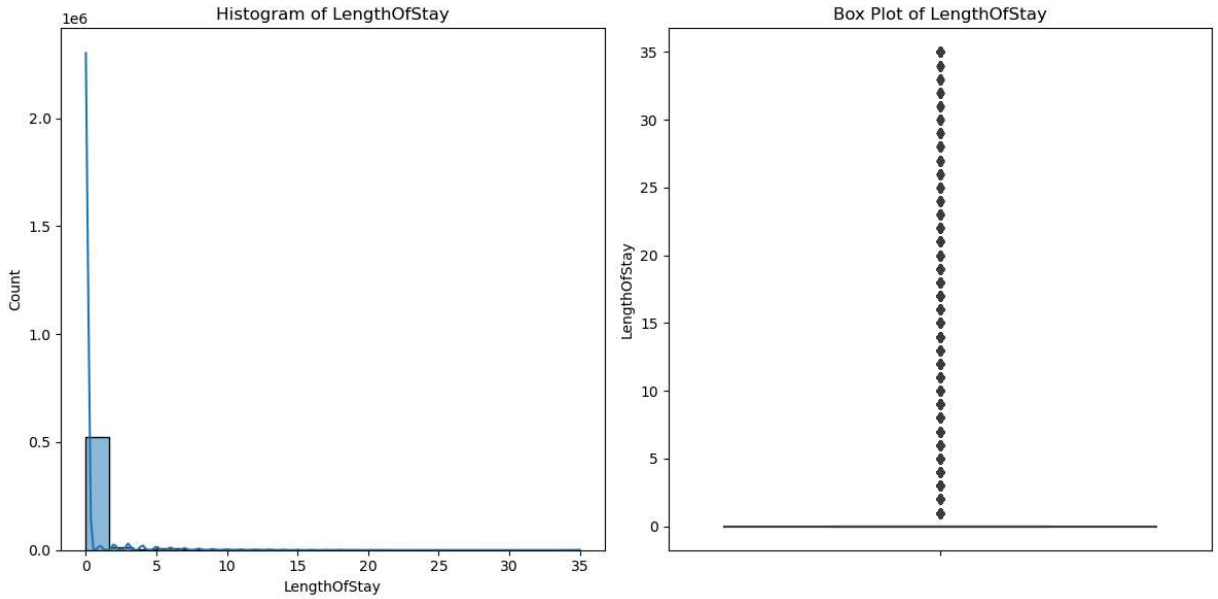

Count of ChronicCond_Osteoporasis
Number of Patients
350000
300000
250000
200000
150000
100000
50000
0
1
2
ChronicCond_Osteoporasis


Count of ChronicCond_rheumatoidarthritis
Number of Patients
400000
350000
300000
250000
200000
150000
100000
50000
0
1
2
ChronicCond_rheumatoidarthritis

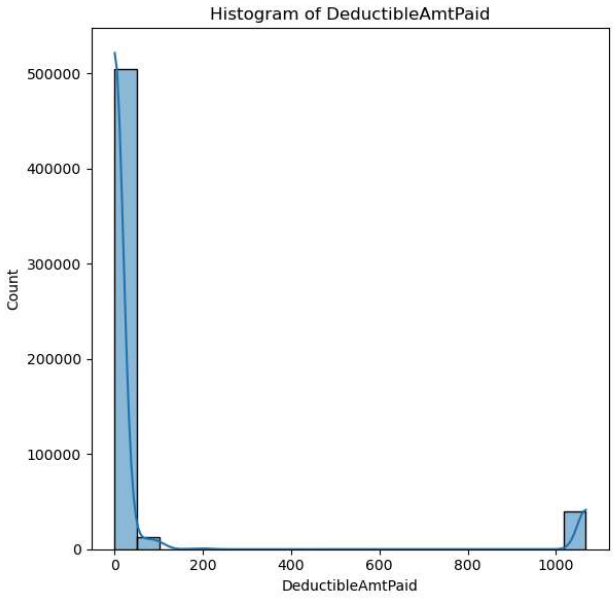

Count of ChronicCond_stroke
Number of Patients
500000
400000
300000
200000
100000
0
1
2
ChronicCond_stroke

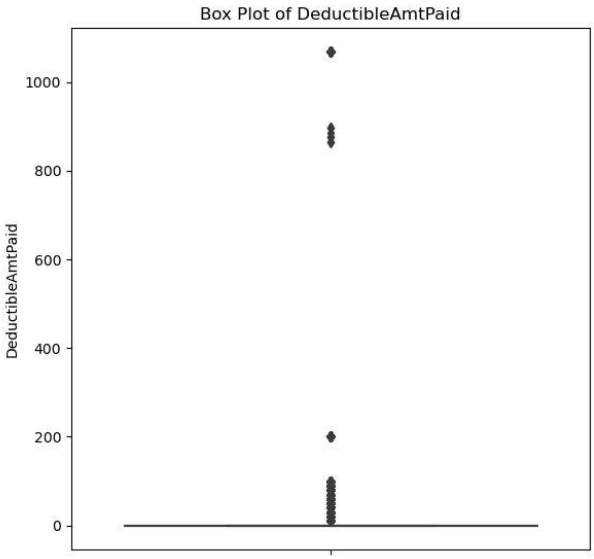
In [94]:
```
# List of numerical variables to check for skewness
variables_to_check = ['LengthOfStay', 'DeductibleAmtPaid', 'Age', 'NoOfMonths_PartA

for var in variables_to_check:
    plt.figure(figsize=(12, 6))

    # Histogram
    plt.subplot(1, 2, 1)
    sns.histplot(full_merged_data[var], kde=True)
    plt.title(f'Histogram of {var}')

    # Boxplot
    plt.subplot(1, 2, 2)
    sns.boxplot(y=full_merged_data[var])
    plt.title(f'Box Plot of {var}')

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

```
C:\Users\asus\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: u
se_inf_as_na option is deprecated and will be removed in a future version. Convert i
nf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

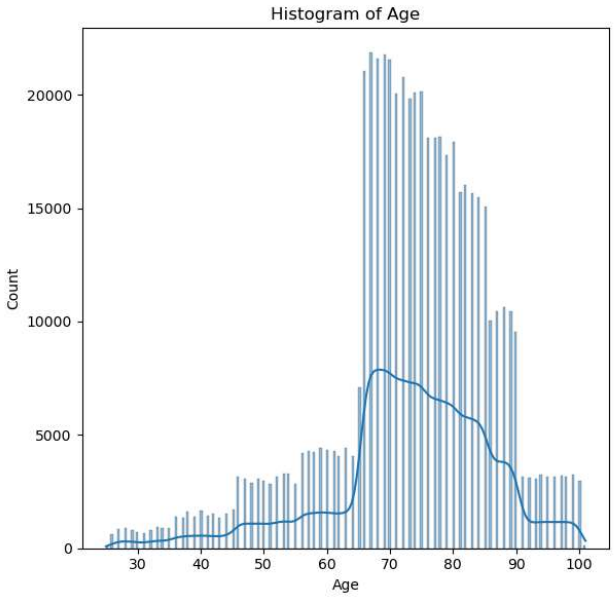


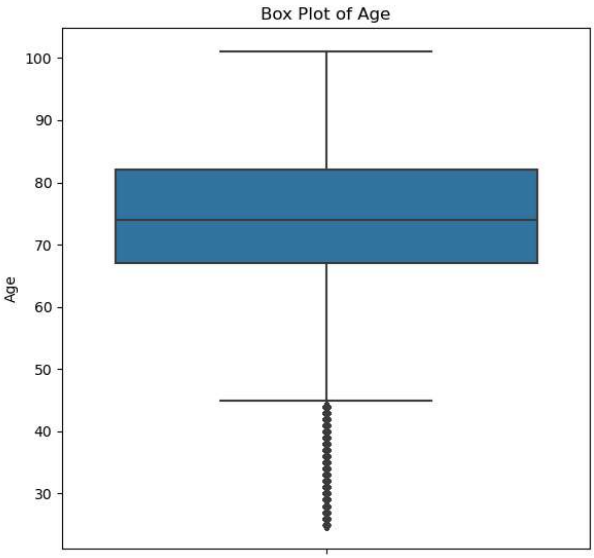

```
C:\Users\asus\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: u
se_inf_as_na option is deprecated and will be removed in a future version. Convert i
nf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



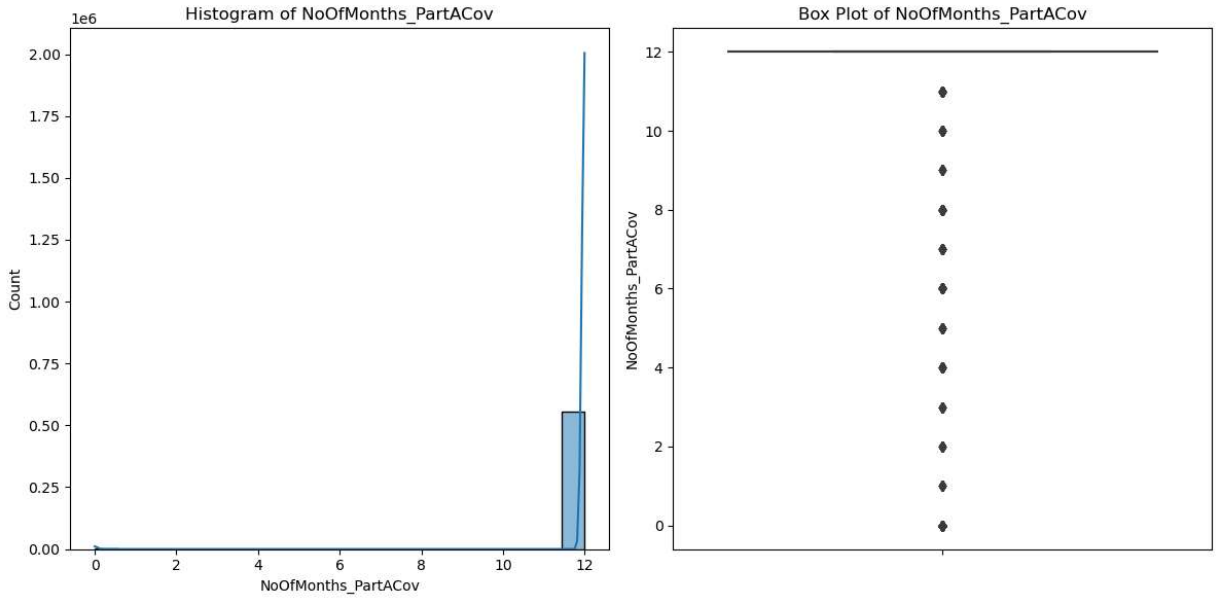



```
C:\Users\asus\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: u
se_inf_as_na option is deprecated and will be removed in a future version. Convert i
nf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

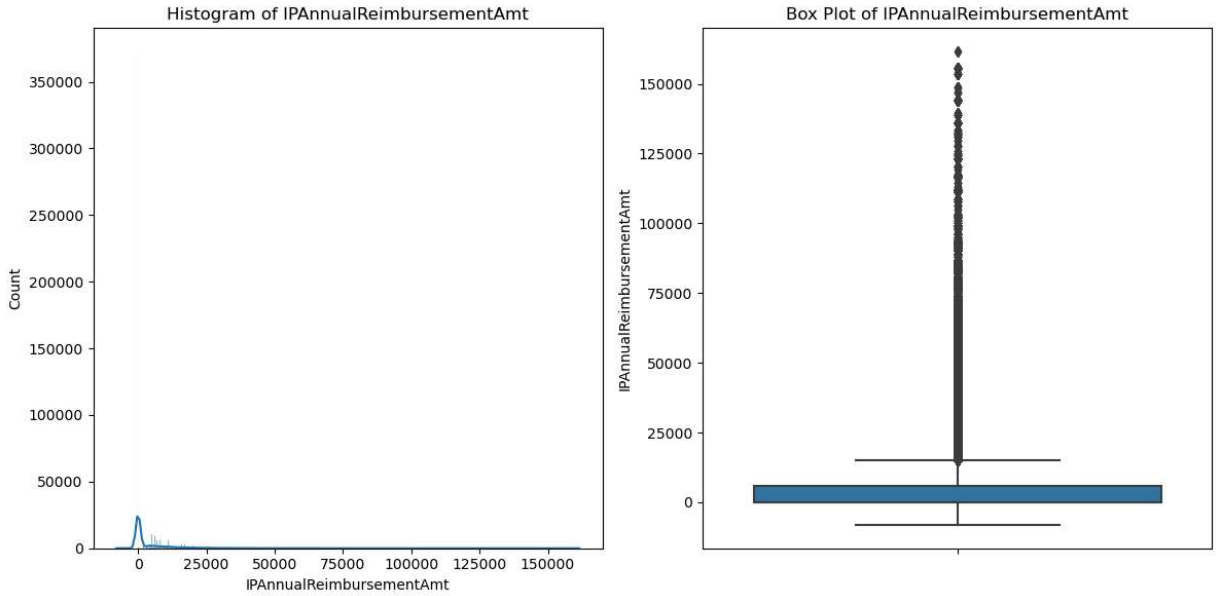




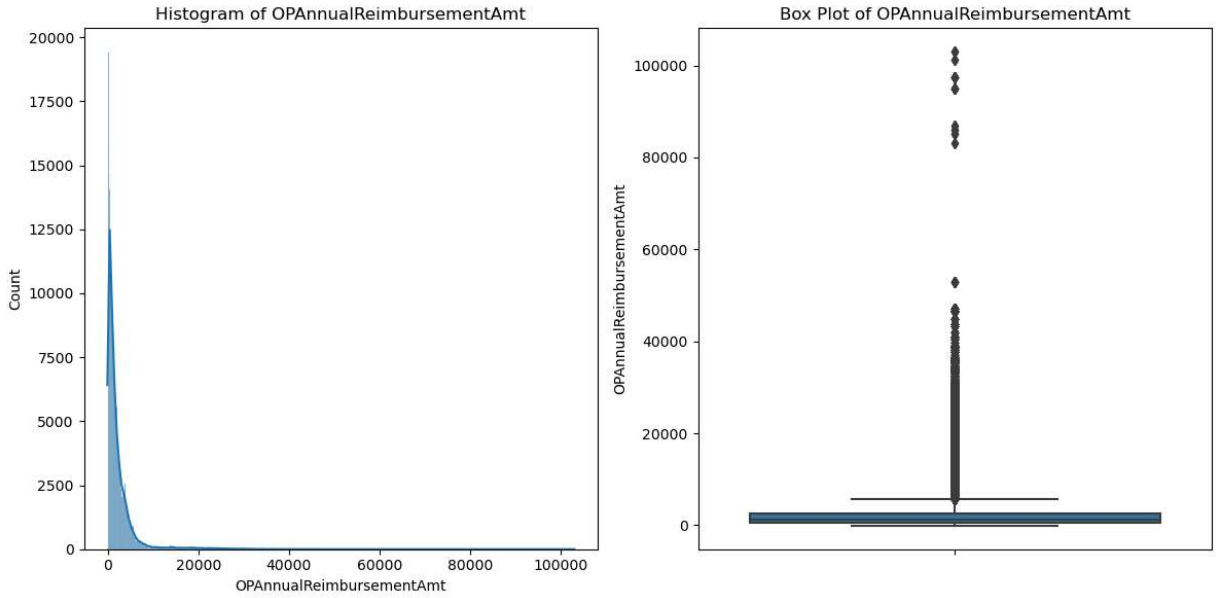

```
C:\Users\asus\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: u
se_inf_as_na option is deprecated and will be removed in a future version. Convert i
nf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```




```
C:\Users\asus\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: u
se_inf_as_na option is deprecated and will be removed in a future version. Convert i
nf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```




```
C:\Users\asus\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: u
se_inf_as_na option is deprecated and will be removed in a future version. Convert i
nf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

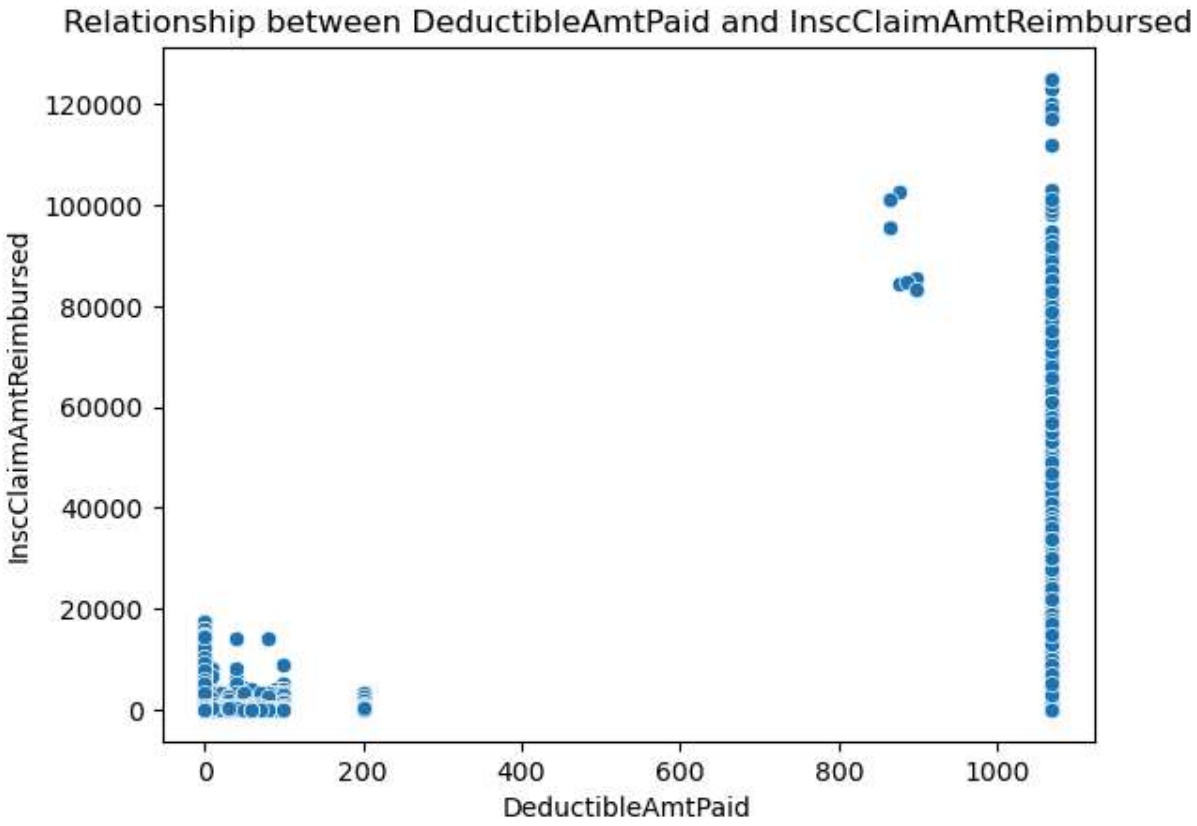


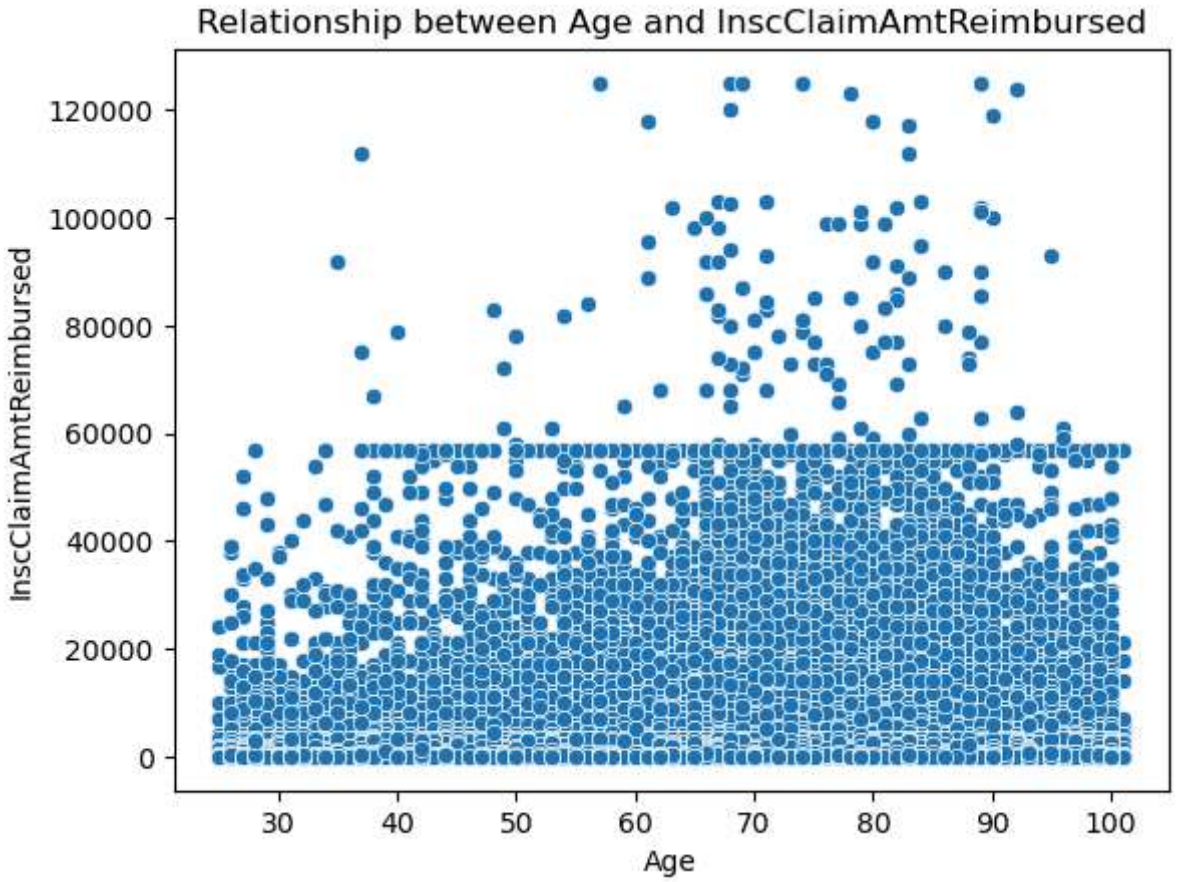
In [95]:
```
# Mapping 'Yes' and 'No' to 1 and 0 in the 'PotentialFraud' column
full_merged_data['PotentialFraud'] = full_merged_data['PotentialFraud'].map({'Yes':

# Mapping 'Y' and '0' to 1 and 0 in the 'RenalDiseaseIndicator' column
full_merged_data['RenalDiseaseIndicator'] = full_merged_data['RenalDiseaseIndicator

# Display the first few entries of the dataframe to verify the changes
full_merged_data[['PotentialFraud', 'RenalDiseaseIndicator']].head()
```

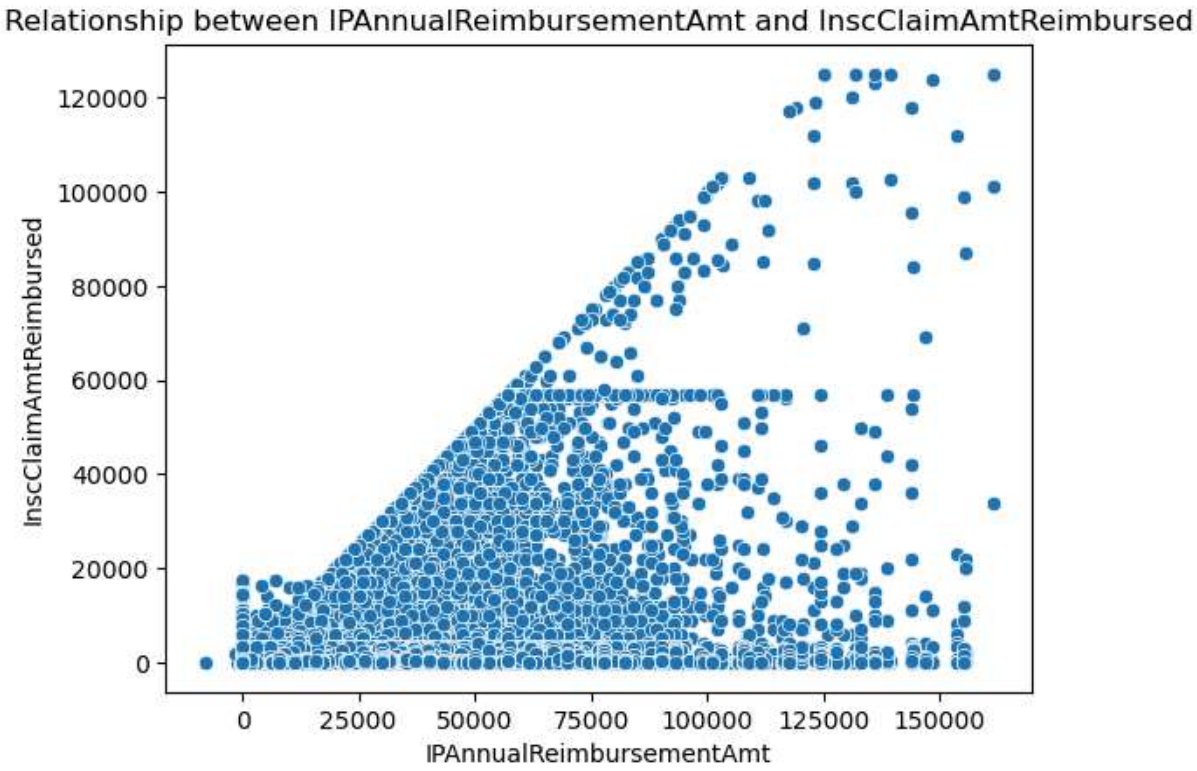
Out[95]:

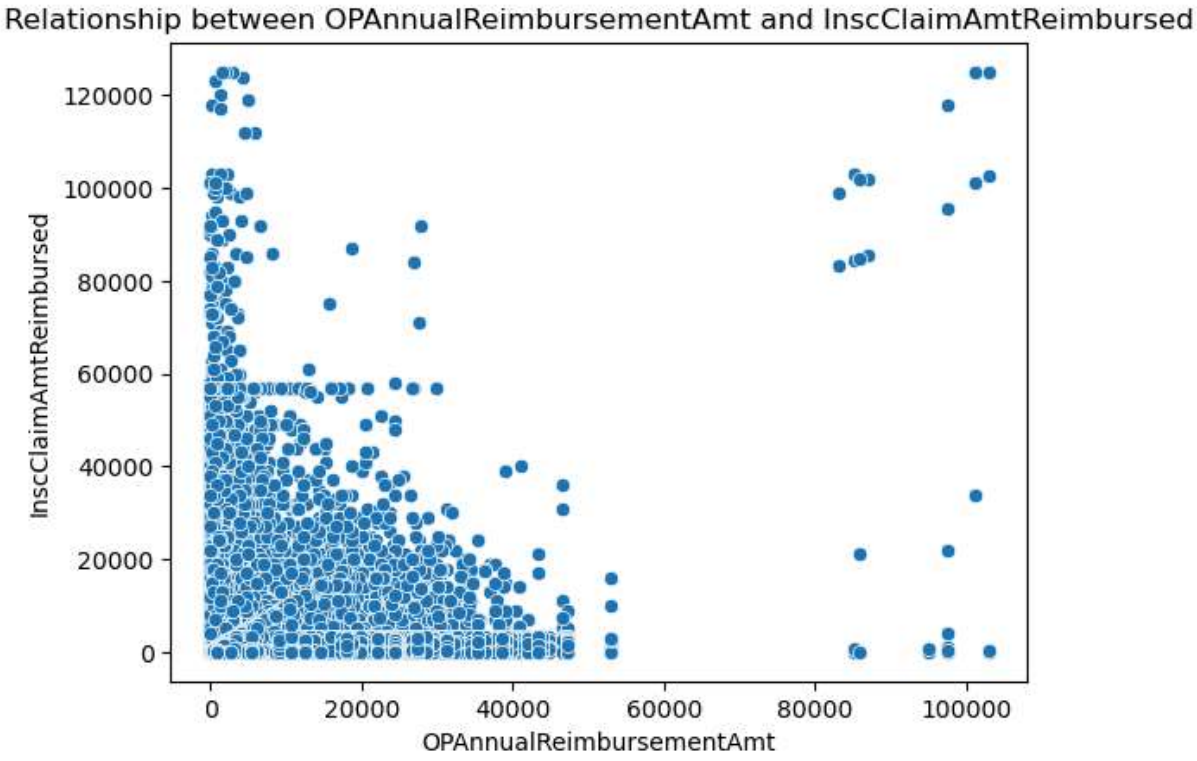
	PotentialFraud	RenalDiseaseIndicator
0	1	0
1	0	0
2	0	0
3	0	0
4	0	1

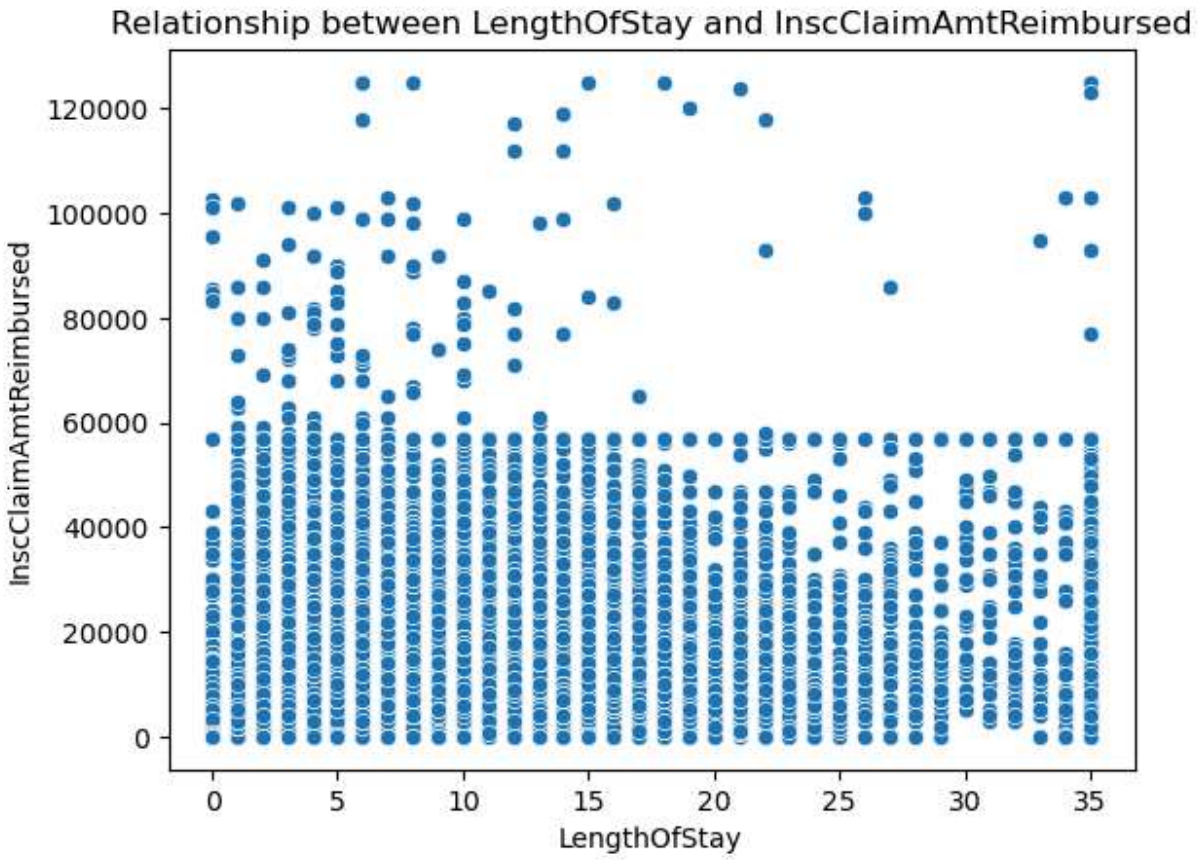
In [96]:
```
numerical_variables = ['DeductibleAmtPaid', 'Age', 'IPAnnualReimbursementAmt', 'OPA

for var in numerical_variables:
    sns.scatterplot(data=full_merged_data, x=var, y='InscClaimAmtReimbursed')
    plt.title(f'Relationship between {var} and InscClaimAmtReimbursed')
    plt.show()
```

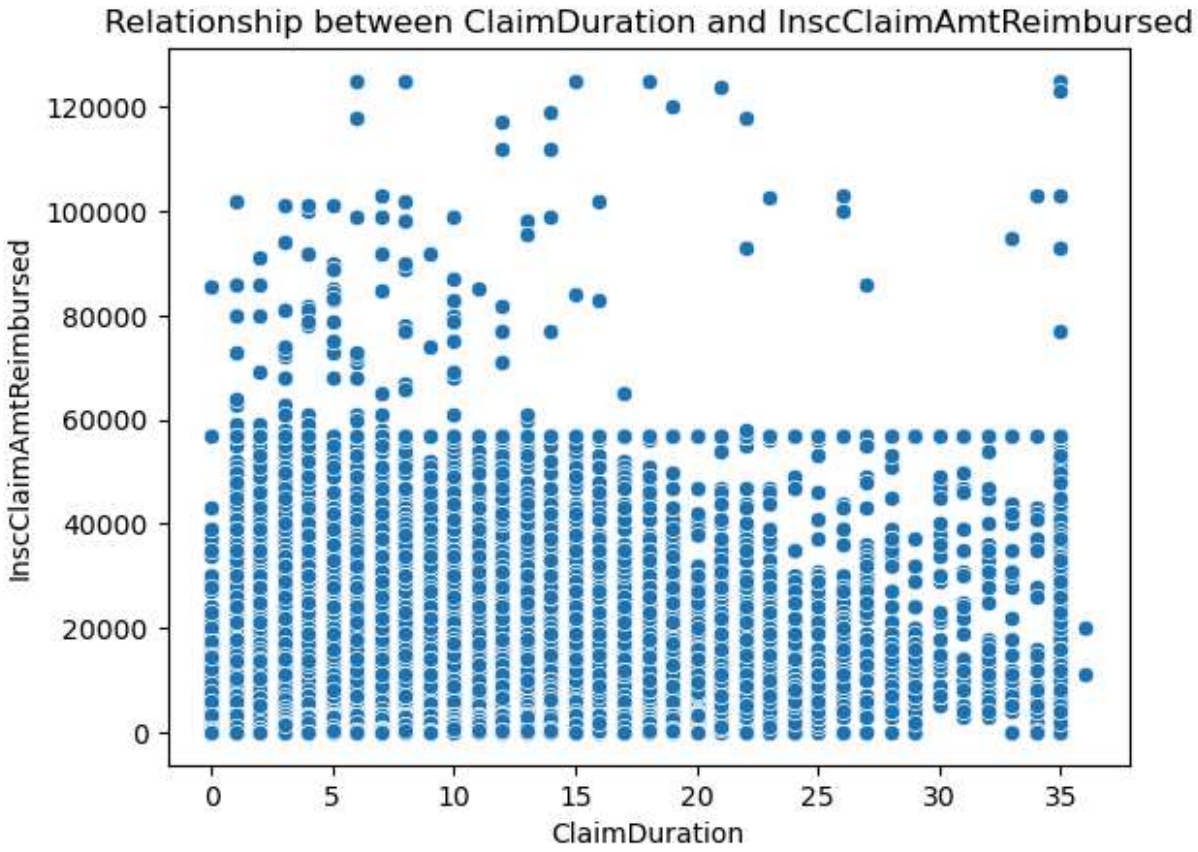

Relationship between DeductibleAmtPaid and InscClaimAmtReimbursed
InscClaimAmtReimbursed
DeductibleAmtPaid
0
20000
40000
60000
80000
100000
120000
0
200
400
600
800
1000


Relationship between Age and InscClaimAmtReimbursed
InscClaimAmtReimbursed
Age
0
20000
40000
60000
80000
100000
120000
30
40
50
60
70
80
90
100


Relationship between IPAnnualReimbursementAmt and InscClaimAmtReimbursed
InscClaimAmtReimbursed
IPAnnualReimbursementAmt
0
20000
40000
60000
80000
100000
120000
0
25000
50000
75000
100000
125000
150000

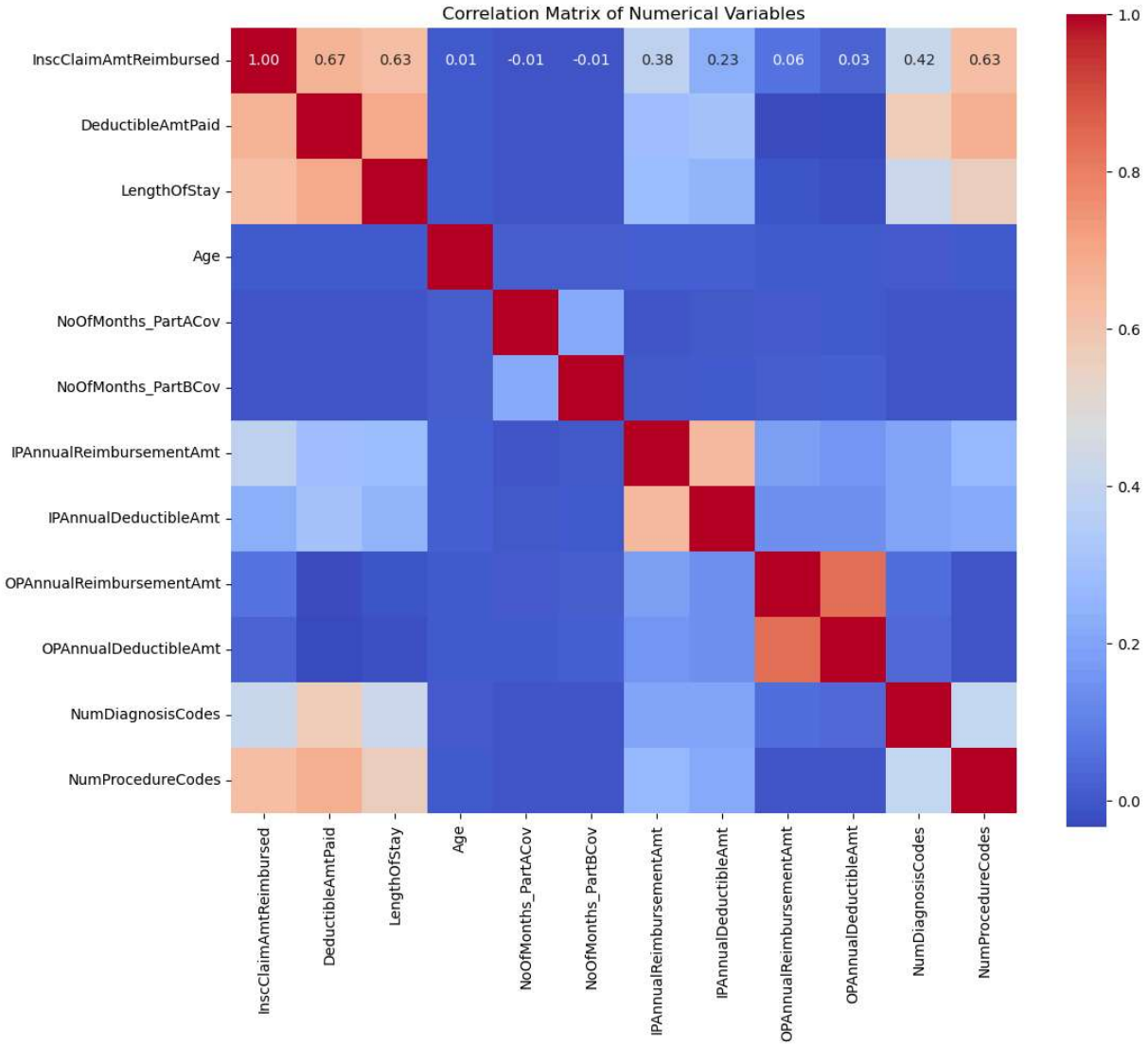

Relationship between OPAnnualReimbursementAmt and InscClaimAmtReimbursed
InscClaimAmtReimbursed
OPAnnualReimbursementAmt
0
20000
40000
60000
80000
100000
120000
0
20000
40000
60000
80000
100000





In [97]:
```
# Select numerical columns for correlation analysis
numerical_cols = ['InscClaimAmtReimbursed', 'DeductibleAmtPaid', 'LengthOfStay',
                  'Age', 'NoOfMonths_PartACov', 'NoOfMonths_PartBCov',
```

```
                  'IPAnnualReimbursementAmt', 'IPAnnualDeductibleAmt',
                  'OPAnnualReimbursementAmt', 'OPAnnualDeductibleAmt',
                  'NumDiagnosisCodes', 'NumProcedureCodes']  # Add any other numeri

# Calculate the correlation matrix
correlation_matrix = full_merged_data[numerical_cols].corr()

# Plot the correlation matrix using seaborn
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, fmt='.2f', cmap='coolwarm', square=True
plt.title('Correlation Matrix of Numerical Variables')
plt.show()
```



In [98]:
```
# Define the columns that we want to drop
columns_to_drop = [
    'BeneID', 'ClaimID', 'ClaimStartDt', 'ClaimEndDt', 'Provider',
    'AttendingPhysician', 'OperatingPhysician', 'OtherPhysician',
    'ClmAdmitDiagnosisCode', 'AdmissionDt','DischargeDt',
    'DiagnosisGroupCode', 'ClmDiagnosisCode_1', 'ClmDiagnosisCode_2', 'ClmDiagnosis
    'ClmDiagnosisCode_4', 'ClmDiagnosisCode_5', 'ClmDiagnosisCode_6', 'ClmDiagnosis
    'ClmDiagnosisCode_8', 'ClmDiagnosisCode_9', 'ClmDiagnosisCode_10',
    'ClmProcedureCode_1', 'ClmProcedureCode_2', 'ClmProcedureCode_3',
```

```
    'ClmProcedureCode_4', 'ClmProcedureCode_5', 'ClmProcedureCode_6',
    'DOB', 'DOD', 'Gender', 'Race', 'State', 'County', 'MostRecentClaimEndDt', 'Pot
]

# Drop the specified columns
data_dropped = full_merged_data.drop(columns=columns_to_drop)

# Display the first few entries of the dataframe to verify the changes
data_dropped.head()
```

Out[98]:

	InscClaimAmtReimbursed	DeductibleAmtPaid	RenalDiseaseIndicator	NoOfMonths_PartA
0	26000	1068.0	0	
1	5000	1068.0	0	
2	5000	1068.0	0	
3	5000	1068.0	0	
4	10000	1068.0	1	

5 rows × 25 columns

In [99]:
```
data_dropped.to_csv('full_merged_dataXGB.csv', index=False)
```

In [100…
```
!conda install -c conda-forge xgboost -y
```

```
Channels:
 - conda-forge
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): ...working... done
Solving environment: ...working... done

# All requested packages already installed.
```

In [101…
```
from sklearn.model_selection import train_test_split
import xgboost as xgb
from sklearn.metrics import mean_squared_error, r2_score

# Define features and target variable
X = data_dropped.drop(['InscClaimAmtReimbursed'], axis=1)
y = data_dropped['InscClaimAmtReimbursed']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

# Initialize the XGBoost regressor with default parameters
xg_reg = xgb.XGBRegressor(objective='reg:squarederror', random_state=42)

# Fit the regressor to the training set
xg_reg.fit(X_train, y_train)
```

```
# Predict on the training set
y_train_pred = xg_reg.predict(X_train)

# Calculate RMSE and R-squared for the training set
rmse_train = np.sqrt(mean_squared_error(y_train, y_train_pred))
r2_train = r2_score(y_train, y_train_pred)

# Print training set metrics
print(f'Training RMSE: {rmse_train:.2f}')
print(f'Training R-squared: {r2_train:.2f}')

# Predict on the test set
y_test_pred = xg_reg.predict(X_test)

# Calculate RMSE and R-squared for the test set
rmse_test = np.sqrt(mean_squared_error(y_test, y_test_pred))
r2_test = r2_score(y_test, y_test_pred)

# Print test set metrics
print(f'Test RMSE: {rmse_test:.2f}')
print(f'Test R-squared: {r2_test:.2f}')
```

```
Training RMSE: 1097.05
Training R-squared: 0.92
Test RMSE: 1653.88
Test R-squared: 0.81
```

In [102…

```
from scipy import stats

# Calculate the Z-scores of each column
z_scores = stats.zscore(data_dropped[numerical_cols])

# Convert to absolute values for comparison
abs_z_scores = np.abs(z_scores)

# Determine a threshold
threshold = 3

# Identify outliers
outliers = (abs_z_scores > threshold).any(axis=1)

# Remove outliers
data_cleaned = data_dropped[~outliers]

# Now data_cleaned contains the data without outliers based on Z-scores
```

In [103…

```
data_cleaned.head()
```

Out[103…

	InscClaimAmtReimbursed	DeductibleAmtPaid	RenalDiseaseIndicator	NoOfMonths_PartA
2	5000	1068.0	0	
8	7000	1068.0	0	
13	6000	1068.0	0	
16	7000	1068.0	0	
17	4000	1068.0	1	

5 rows × 25 columns

In [104…

```
from sklearn.model_selection import train_test_split
import xgboost as xgb
from sklearn.metrics import mean_squared_error, r2_score

# Define features and target variable
X = data_cleaned.drop(['InscClaimAmtReimbursed'], axis=1)
y = data_cleaned['InscClaimAmtReimbursed']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

# Initialize the XGBoost regressor with default parameters
xg_reg = xgb.XGBRegressor(objective='reg:squarederror', random_state=42)

# Fit the regressor to the training set
xg_reg.fit(X_train, y_train)

# Predict on the training set
y_train_pred = xg_reg.predict(X_train)

# Calculate RMSE and R-squared for the training set
rmse_train = np.sqrt(mean_squared_error(y_train, y_train_pred))
r2_train = r2_score(y_train, y_train_pred)

# Print training set metrics
print(f'Training RMSE: {rmse_train:.2f}')
print(f'Training R-squared: {r2_train:.2f}')

# Predict on the test set
y_test_pred = xg_reg.predict(X_test)

# Calculate RMSE and R-squared for the test set
rmse_test = np.sqrt(mean_squared_error(y_test, y_test_pred))
r2_test = r2_score(y_test, y_test_pred)

# Print test set metrics
print(f'Test RMSE: {rmse_test:.2f}')
print(f'Test R-squared: {r2_test:.2f}')
```

```
Training RMSE: 484.07
Training R-squared: 0.76
Test RMSE: 527.90
Test R-squared: 0.70
```

In [105…
```python
import pickle

# Serialize the trained XGBoost model using pickle
with open('xgboost_claim_prediction_model.pkl', 'wb') as file:
    pickle.dump(xg_reg, file)
```

In [106…
```python
data_cleaned.to_csv('Clean_full_merged_dataXGB.csv', index=False)
```

In [107…
```python
data_cleaned.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 491555 entries, 2 to 558210
Data columns (total 25 columns):
 #   Column                          Non-Null Count   Dtype  
---  ------                          --------------   -----  
 0   InscClaimAmtReimbursed          491555 non-null  int64  
 1   DeductibleAmtPaid               491383 non-null  float64
 2   RenalDiseaseIndicator           491555 non-null  int64  
 3   NoOfMonths_PartACov             491555 non-null  int64  
 4   NoOfMonths_PartBCov             491555 non-null  int64  
 5   ChronicCond_Alzheimer           491555 non-null  int64  
 6   ChronicCond_Heartfailure        491555 non-null  int64  
 7   ChronicCond_KidneyDisease       491555 non-null  int64  
 8   ChronicCond_Cancer              491555 non-null  int64  
 9   ChronicCond_ObstrPulmonary      491555 non-null  int64  
 10  ChronicCond_Depression          491555 non-null  int64  
 11  ChronicCond_Diabetes            491555 non-null  int64  
 12  ChronicCond_IschemicHeart       491555 non-null  int64  
 13  ChronicCond_Osteoporasis        491555 non-null  int64  
 14  ChronicCond_rheumatoidarthritis 491555 non-null  int64  
 15  ChronicCond_stroke              491555 non-null  int64  
 16  IPAnnualReimbursementAmt        491555 non-null  int64  
 17  IPAnnualDeductibleAmt           491555 non-null  int64  
 18  OPAnnualReimbursementAmt        491555 non-null  int64  
 19  OPAnnualDeductibleAmt           491555 non-null  int64  
 20  LengthOfStay                    491555 non-null  float64
 21  ClaimDuration                   491555 non-null  int64  
 22  Age                             491555 non-null  int64  
 23  NumDiagnosisCodes               491555 non-null  int64  
 24  NumProcedureCodes               491555 non-null  int64  
dtypes: float64(2), int64(23)
memory usage: 97.5 MB
```

In [108…
```python
X_train.dtypes
```

Out[108…

```
DeductibleAmtPaid                float64
RenalDiseaseIndicator              int64
NoOfMonths_PartACov                int64
NoOfMonths_PartBCov                int64
ChronicCond_Alzheimer              int64
ChronicCond_Heartfailure           int64
ChronicCond_KidneyDisease          int64
ChronicCond_Cancer                 int64
ChronicCond_ObstrPulmonary         int64
ChronicCond_Depression             int64
ChronicCond_Diabetes               int64
ChronicCond_IschemicHeart          int64
ChronicCond_Osteoporasis           int64
ChronicCond_rheumatoidarthritis    int64
ChronicCond_stroke                 int64
IPAnnualReimbursementAmt           int64
IPAnnualDeductibleAmt              int64
OPAnnualReimbursementAmt           int64
OPAnnualDeductibleAmt              int64
LengthOfStay                     float64
ClaimDuration                      int64
Age                                int64
NumDiagnosisCodes                  int64
NumProcedureCodes                  int64
dtype: object
```

In [ ]: