Install libraries

In [12]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
import statsmodels.api as sm
from scipy.stats import pearsonr
```

Loading Data

In [3]:

```
data = pd.read_csv('data.csv')
data.head()
```

Out[3]:

	Company	File No.	Opened	Closed	Coverage	SubCoverage	Reason	S
0	Anthem Health Plans, Inc	7045593	05/31/2022	06/02/2022	Group	Health Only	Claim Handling	
1	Anthem Health Plans, Inc	7043381	02/28/2022	06/02/2022	Group	Health Only	Claim Handling	
2	Anthem Health Plans, Inc	7044860	05/03/2022	06/02/2022	A & H	Health Only	Claim Handling	
3	Anthem Health Plans, Inc	7043381	02/28/2022	06/02/2022	Group	A & H	Claim Handling	
4	Anthem Health Plans, Inc	7052007	02/23/2023	03/17/2023	A & H	A & H	Marketing & Sales	

Descriptive Statistics & Summary Measures

Calculate mean, median, standard deviation, quartiles, and range for numerical variables

In [4]:

```
data.info()

# File No is an id
# Recovery is useful

recovery = data['Recovery']
```

```
recovery_mean = np.mean(recovery)
recovery_median = np.median(recovery)
recovery_std = np.std(recovery)
recovery_min = np.min(recovery)
recovery_max = np.max(recovery)
recovery_range = recovery_max - recovery_min
recovery_quartiles = np.percentile(recovery, [25, 50, 75, 100])

print("\n")
print(f'Mean: {recovery_mean:.2f}')
print(f'Median: {recovery_median:.2f}')
print(f'Standard Deviation: {recovery_std:.2f}')
print(f'Range: {recovery_range:.2f}')
print(f'Quartiles: {recovery_quartiles}')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 68469 entries, 0 to 68468
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Company      68469 non-null  object 
 1   File No.     68469 non-null  int64  
 2   Opened       68469 non-null  object 
 3   Closed       66992 non-null  object 
 4   Coverage     65130 non-null  object 
 5   SubCoverage  55020 non-null  object 
 6   Reason       65057 non-null  object 
 7   SubReason    65057 non-null  object 
 8   Disposition  41476 non-null  object 
 9   Conclusion   42809 non-null  object 
 10  Recovery     68469 non-null  float64
 11  Status       68469 non-null  object 
dtypes: float64(1), int64(1), object(10)
memory usage: 6.3+ MB


Mean: 1723.51
Median: 0.00
Standard Deviation: 13989.96
Range: 843825.85
Quartiles: [     0.          0.          0.     843825.85]
```

Missing Value Analysis

Identify missing values, analyze their patterns (random or systematic), and calculate the percentage of missing entries for each column.

In [5]:

```
# Missing values
missing_values = data.isnull().sum()
print("\nTotal Number of Missing Values:")
print(missing_values)

# Percentage of Missing Entries
```

```python
total_entries = len(data)
missing_percentage = (missing_values / total_entries) * 100
print("\nPercentage of Missing Entries:")
print(missing_percentage)
```

```
Total Number of Missing Values:
Company            0
File No.           0
Opened             0
Closed          1477
Coverage        3339
SubCoverage    13449
Reason          3412
SubReason       3412
Disposition    26993
Conclusion     25660
Recovery           0
Status             0
dtype: int64

Percentage of Missing Entries:
Company         0.000000
File No.        0.000000
Opened          0.000000
Closed          2.157181
Coverage        4.876660
SubCoverage    19.642466
Reason          4.983277
SubReason       4.983277
Disposition    39.423681
Conclusion     37.476814
Recovery        0.000000
Status          0.000000
dtype: float64
```

Univariate Distribution Visualization

Create histograms, box plots, or density plots for individual variables.

In [ ]:

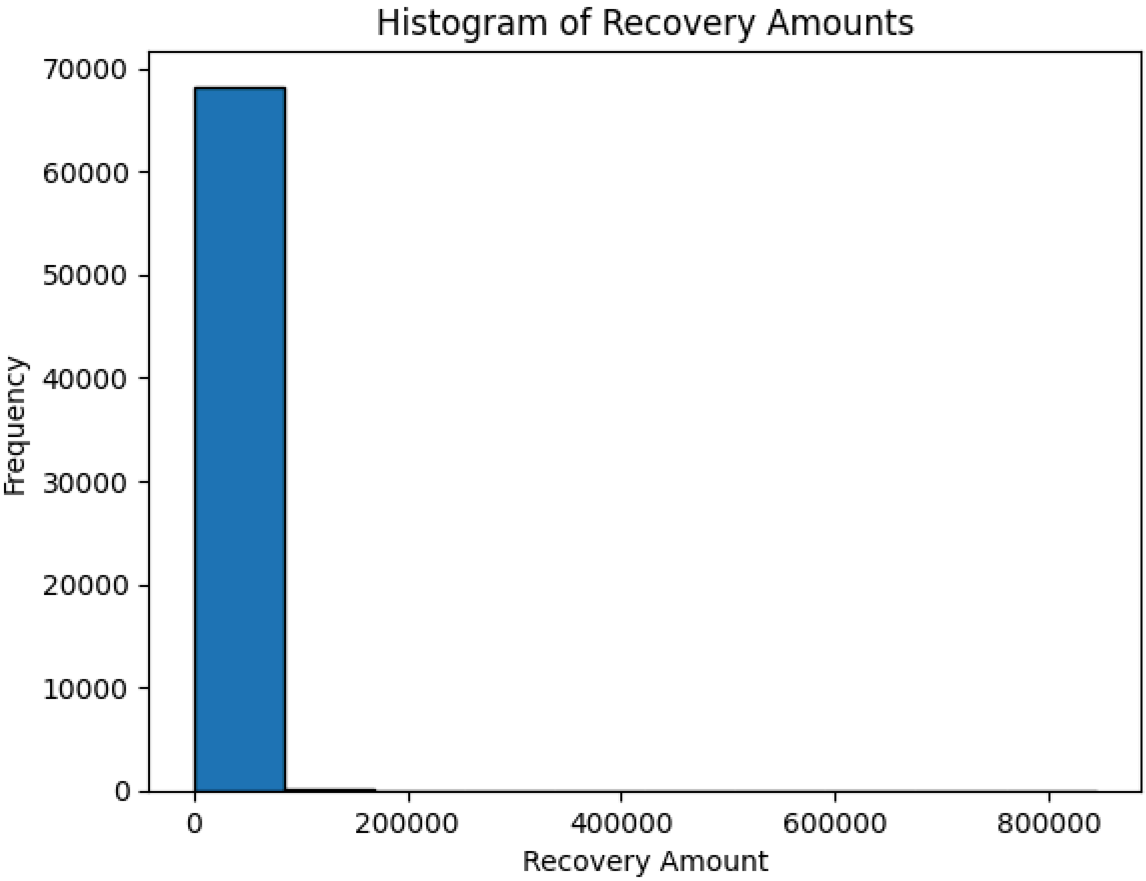
```python
recovery_max = np.max(recovery)
print(recovery_max)
```

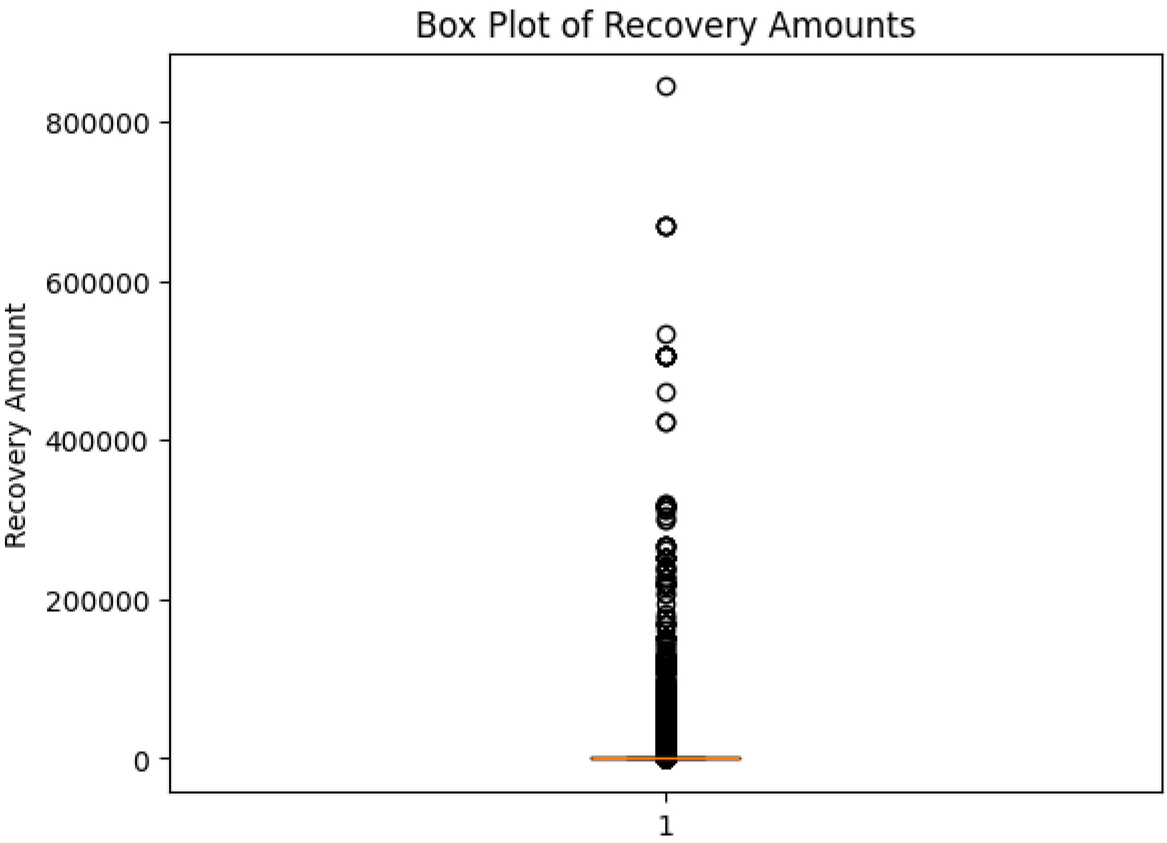
```
843825.85
```

In [6]:

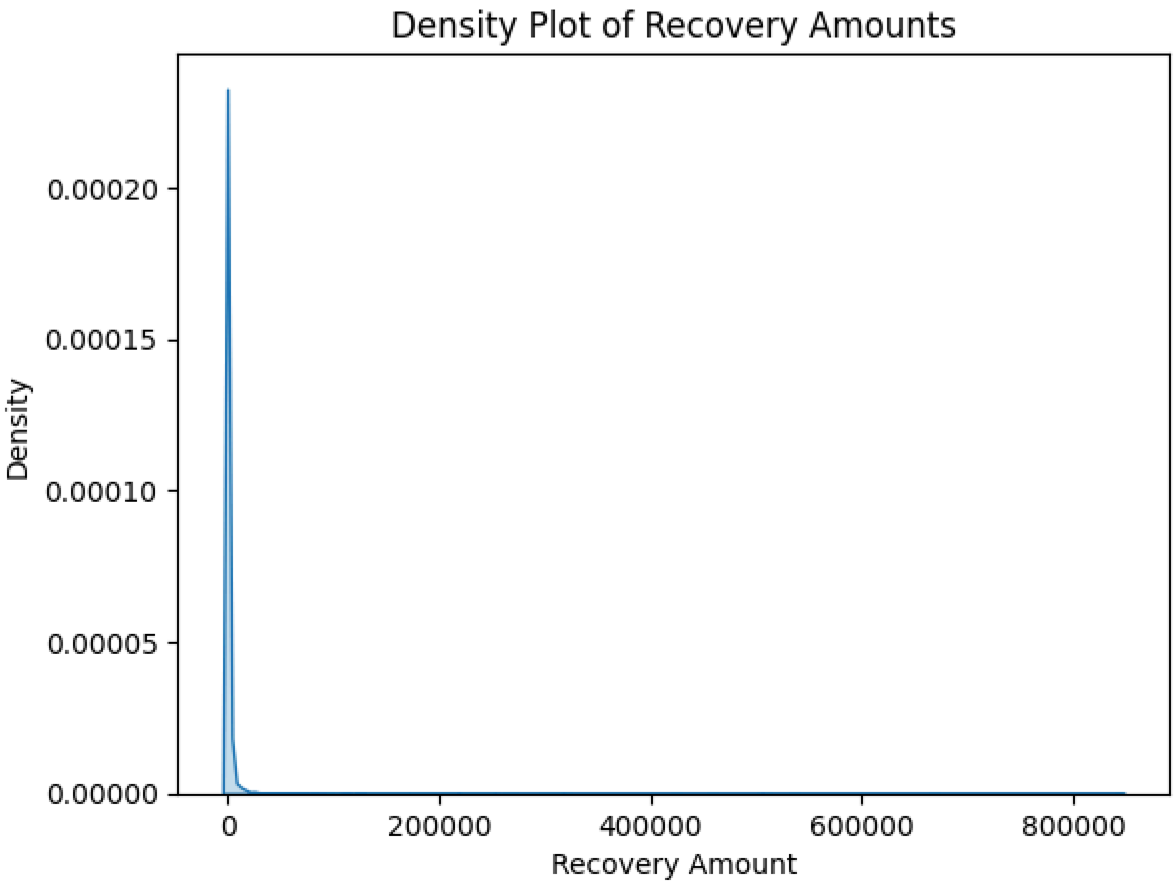
```python
# histogram:
plt.hist(data['Recovery'], edgecolor='black')
plt.title('Histogram of Recovery Amounts')
plt.xlabel('Recovery Amount')
plt.ylabel('Frequency')
plt.show()

# box plot:
```

```
plt.boxplot(data['Recovery'])
plt.title('Box Plot of Recovery Amounts')
plt.ylabel('Recovery Amount')
plt.show()

# density plot:
sns.kdeplot(data['Recovery'], shade=True)
plt.title('Density Plot of Recovery Amounts')
plt.xlabel('Recovery Amount')
plt.ylabel('Density')
plt.show()
```





```
<ipython-input-6-0b29ca802637>:15: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(data['Recovery'], shade=True)
```



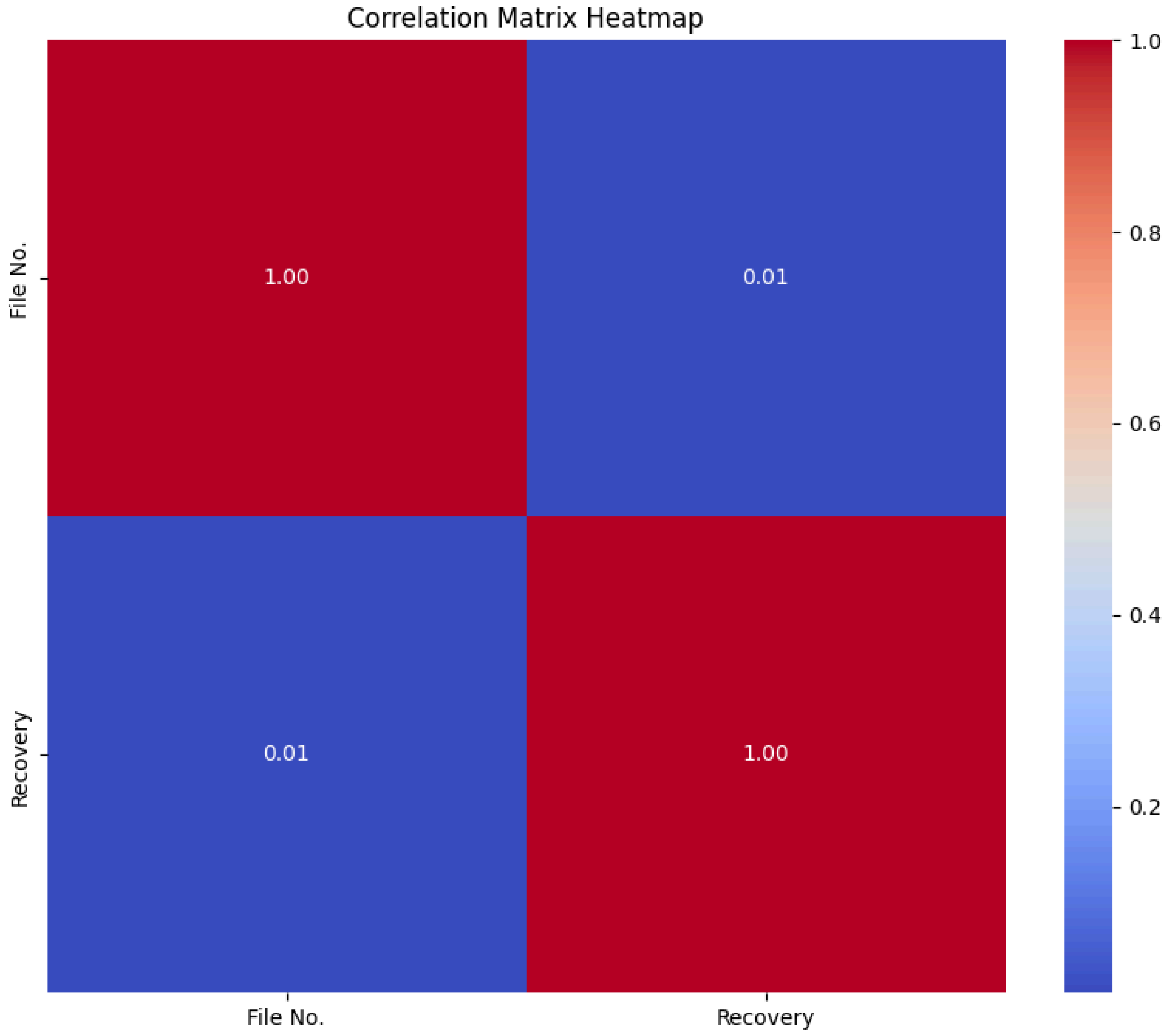
Correlation and Relationship Analysis

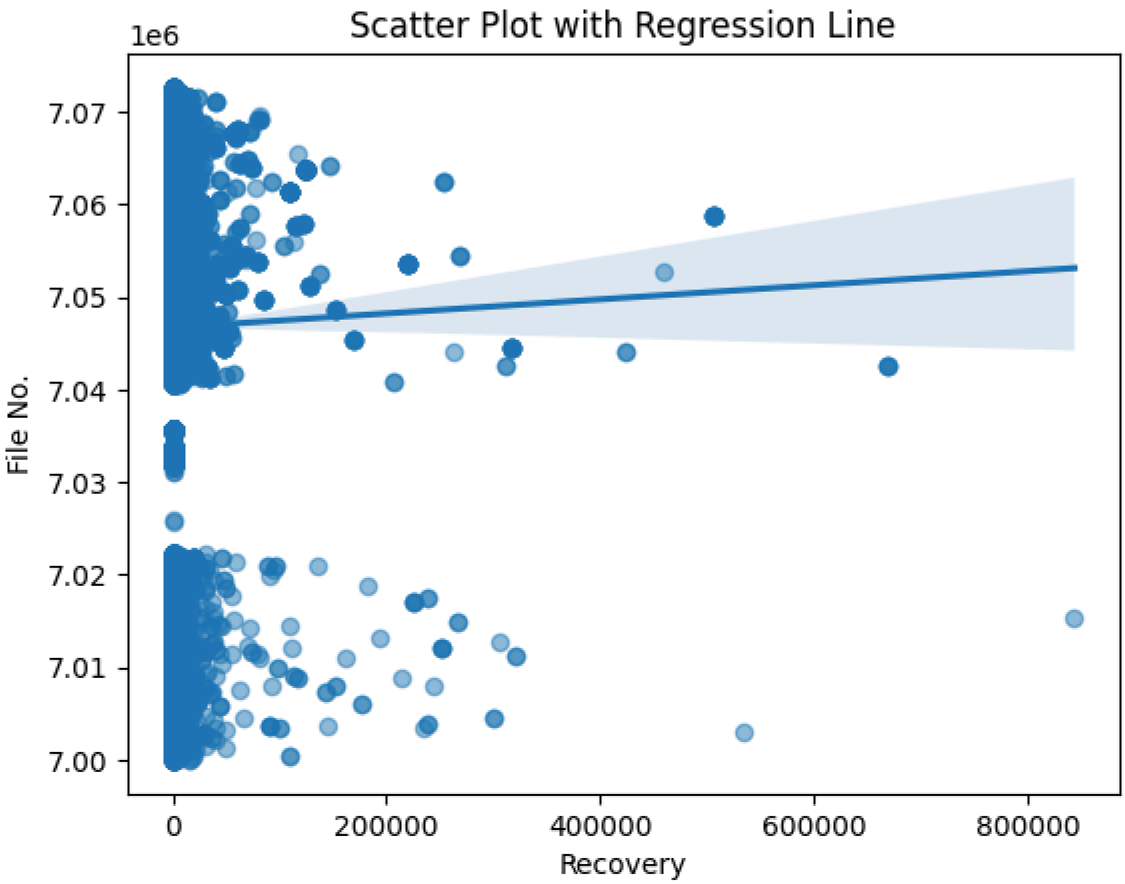
In [13]:

```
# Compute the correlation matrix (select only numeric columns)
numerical_data = data.select_dtypes(include=np.number)
corr_matrix = numerical_data.corr()

# Display the correlation matrix
print(corr_matrix)

# Set the figure size for better readability
plt.figure(figsize=(10, 8))
# Create a heatmap with correlation coefficients annotated
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
# Add a title and display the plot
plt.title("Correlation Matrix Heatmap")
plt.show()

# Create a scatter plot with regression line using the actual column names
sns.regplot(x='Recovery', y='File No.', data=data, scatter_kws={'alpha': 0.5
plt.title("Scatter Plot with Regression Line")
plt.xlabel("Recovery")
plt.ylabel("File No.")
plt.show()

# Compute Pearson correlation coefficient and p-value using 'Recovery' and '
corr_coef, p_value = pearsonr(data['Recovery'], data['File No.'])
```

```
print(f"Pearson correlation coefficient: {corr_coef:.3f}")
print(f"P-value: {p_value:.3f}")

# Define the independent variable(s) and the dependent variable
X = data[['Recovery']]  # Independent variable(s)
y = data['File No.']     # Dependent variable

# Add a constant to the independent variables (the intercept)
X = sm.add_constant(X)

# Fit the Ordinary Least Squares (OLS) model
model = sm.OLS(y, X).fit()

# Print the regression summary
print(model.summary())
```

```
          File No.  Recovery
File No.  1.000000  0.005019
Recovery  0.005019  1.000000
```




Scatter Plot with Regression Line
1e6
7.07
7.06
7.05
7.04
7.03
7.02
7.01
7.00
File No.
0
200000
400000
600000
800000
Recovery

```
Pearson correlation coefficient: 0.005
P-value: 0.189
                            OLS Regression Results                            
============================================================================
==
Dep. Variable:               File No.   R-squared:                       0.0
00
Model:                            OLS   Adj. R-squared:                  0.0
00
Method:                 Least Squares   F-statistic:                     1.7
25
Date:                Tue, 04 Mar 2025   Prob (F-statistic):              0.1
89
Time:                        22:38:47   Log-Likelihood:            -7.7929e+
05
No. Observations:               68469   AIC:                         1.559e+
06
Df Residuals:                   68467   BIC:                         1.559e+
06
Df Model:                           1                                         
Covariance Type:            nonrobust                                         
============================================================================
==
                 coef    std err          t      P>|t|      [0.025      0.97
5]
----------------------------------------------------------------------------
--
const       7.047e+06     81.717   8.62e+04      0.000    7.05e+06    7.05e+
06
Recovery       0.0076      0.006      1.313      0.189      -0.004       0.0
19
============================================================================
==
Omnibus:                     8470.043   Durbin-Watson:                   1.0
49
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             9471.0
78
Skew:                          -0.864   Prob(JB):                         0.
00
Kurtosis:                       2.420   Cond. No.                     1.42e+
04
============================================================================
==

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is corre
ctly specified.
[2] The condition number is large, 1.42e+04. This might indicate that there
are
strong multicollinearity or other numerical problems.
```

Analyzing High Cardinality and Grouping Rare Categories

In [8]:

```
def analyze_high_cardinality(df, threshold=0.02):
    """
    Identifies high-cardinality categorical columns and groups rare categori

    Parameters:
    df (pd.DataFrame): The input DataFrame.
    threshold (float): The minimum frequency a category must have to not be

    Returns:
    pd.DataFrame: A modified DataFrame with rare categories grouped.
    """
    categorical_cols = df.select_dtypes(include=['object']).columns

    for col in categorical_cols:
        value_counts = df[col].value_counts(normalize=True)

        # Print unique value distribution before modification
        print(f"\nColumn: {col}")
        print(value_counts)

        rare_categories = value_counts[value_counts < threshold].index
        if len(rare_categories) > 0:
            print(f"\nGrouping {len(rare_categories)} rare categories in {co
            df[col] = df[col].replace(rare_categories, 'Other')

    return df

# Load CSV file
df = pd.read_csv("data.csv")

# Analyze and modify high-cardinality categorical columns
df_modified = analyze_high_cardinality(df, threshold=0.02)

# Save the modified DataFrame
df_modified.to_csv("modified_data.csv", index=False)

print("High-cardinality analysis complete. Modified data saved.")
```

```
Column: Company
Company
Anthem Health Plans, Inc                                      0.213308
ConnectiCare Benefits, Inc.                                   0.053572
UnitedHealthcare Insurance Company                            0.049847
Cigna Health and Life Insurance Company                       0.046532
ConnectiCare Insurance Company, Inc                           0.033796
                                                                ...   
Plaza Insurance Company                                       0.000015
Trenwick America Reinsurance Corporation                      0.000015
StarStone Specialty Insurance Company                         0.000015
Continental Life Insurance Company of Brentwood, Tennessee    0.000015
ARAG Insurance Company                                        0.000015
Name: proportion, Length: 843, dtype: float64

Grouping 837 rare categories in Company under 'Other'


Column: Opened
Opened
01/03/2025    0.003023
09/18/2024    0.002746
02/29/2024    0.002731
02/28/2024    0.002658
02/07/2024    0.002527
                ...   
11/02/2015    0.000015
02/18/2023    0.000015
09/08/2017    0.000015
11/06/2017    0.000015
12/26/2021    0.000015
Name: proportion, Length: 2139, dtype: float64

Grouping 2139 rare categories in Opened under 'Other'


Column: Closed
Closed
02/27/2024    0.003374
01/06/2025    0.003299
02/23/2023    0.003299
04/16/2024    0.003284
10/09/2024    0.003030
                ...   
10/02/2020    0.000015
06/01/2024    0.000015
01/05/2018    0.000015
12/28/2018    0.000015
08/30/2021    0.000015
Name: proportion, Length: 1765, dtype: float64

Grouping 1765 rare categories in Closed under 'Other'


Column: Coverage
Coverage
```

```
A & H                                    0.233671
Individual Private Passenger             0.225672
Group                                    0.193382
Individual                               0.133226
Homeowners                               0.100077
Individual Life                          0.027253
Commercial Multi-Peril                   0.013619
Travel                                   0.010226
Commercial                               0.010195
Individual Annuities                     0.008875
Condo/Townhome                           0.005988
General                                  0.005881
Group Life                               0.004069
Workers' Compensation                    0.003654
Renter/Tenants                           0.003485
Pet Insurance                            0.002948
Other [Enter Coverage]                   0.002518
Extended Warranty & Service Contracts    0.002073
Dwelling Fire                            0.001520
Credit Accident & Health                 0.001336
Umbrella                                 0.001198
Inland Marine                            0.001044
Fire, Allied Lines                       0.000829
Title                                    0.000691
Watercraft                               0.000645
Motorcycle                               0.000614
Professional/E&O                         0.000583
Unknown                                  0.000568
Motorhome                                0.000522
Mobile Homeowner                         0.000522
Portable Electronics Ins                 0.000415
Rental                                   0.000384
Federal Flood                            0.000322
Farm owner/Ranch owner                   0.000230
Group Annuities                          0.000230
Life and Annuity                         0.000215
Fidelity & Surety                        0.000215
Group Private Passenger                  0.000138
Extended Warranty                        0.000138
Portable Electronics                     0.000092
Directors & Officers                     0.000077
Ocean Marine                             0.000061
Business Interruption                    0.000061
Federal Programs                         0.000061
Group Homeowners                         0.000061
Credit Life                              0.000061
Motorsport                               0.000046
Credit Property                          0.000046
Crop/Hail                                0.000046
Aircraft                                 0.000031
GAP Ins                                  0.000031
IRA                                      0.000031
Surplus Lines                            0.000031
Builder's Risk                           0.000031
Auto Warranty                            0.000031
Products                                 0.000015
```

```
Accelerated Benefits                     0.000015
Name: proportion, dtype: float64

Grouping 51 rare categories in Coverage under 'Other'


Column: SubCoverage
SubCoverage
Health Only                                0.267884
A & H                                      0.202163
Liability                                  0.135260
Homeowners                                 0.045674
Collision                                  0.044257
                                             ...
Residual Market/Joint Underwriting Assn    0.000018
COBRA                                      0.000018
Medicare Supplement Plan K                 0.000018
Excess Loss                                0.000018
Comprehensive Personal Liability           0.000018
Name: proportion, Length: 110, dtype: float64

Grouping 99 rare categories in SubCoverage under 'Other'


Column: Reason
Reason
Claim Handling          0.765313
PolicyHolder Service    0.107321
Underwriting            0.099036
Marketing & Sales       0.028329
Name: proportion, dtype: float64

Column: SubReason
SubReason
Claim Denial                       0.113085
Claim Delay                        0.097130
Unsatisfactory Settlement/Offer    0.087462
Medical Necessity Denial           0.057012
Premium & Rating                   0.038658
                                     ...
Grace Period                       0.000015
Wellness Program                   0.000015
1035 Exchange                      0.000015
Endorsement Rider                  0.000015
Nonforfeiture                      0.000015
Name: proportion, Length: 198, dtype: float64

Grouping 184 rare categories in SubReason under 'Other'


Column: Disposition
Disposition
Company Position Substantiated                                 0.440616
Claim Settled                                                  0.244503
Question of Fact/Contract/Provision/Legal Issue                0.106158
Company Position Overturned                                    0.057841
```

```
Compromised Settlement/Resolution                              0.046895
No Action Requested/Required                                   0.044411
No Jurisdiction                                                0.034261
Insufficient Information                                       0.012103
Complaint Withdrawn                                            0.005521
Referred to Outside Agency/Dept                                0.002724
Referred to Another State’s Dept of Insurance                  0.001929
Claim Reopened                                                 0.001423
Referred to Other Division for Possible Disciplinary Action    0.001230
Fine Assessed                                                  0.000386
Name: proportion, dtype: float64

Grouping 7 rare categories in Disposition under 'Other'


Column: Conclusion
Conclusion
Claim Paid                    0.128968
Company Position Upheld       0.106753
Justified                     0.085146
Furnished Information         0.083137
Contract Provision            0.081525
Corrective Action             0.073232
Provider Issue                0.072882
Coverage Granted              0.042701
Refer-Judicial/Attorney       0.036488
Claim Paid With Interest      0.033264
Coverage Denied               0.023336
Unjustified                   0.020510
Voluntary Reconsideration     0.019295
Premium Refund                0.018734
Policy Restored/Reinstated    0.018407
Enter Arbitration             0.017052
Rate Increase Explained       0.014600
Satisfactory Explanation      0.013946
Additional Money Received     0.013081
Questionable                  0.011983
Refer To Appraisal            0.010815
Satisfied                     0.009951
External Review Info Sent     0.009834
No Action Necessary           0.006798
No Authority                  0.005466
Record Only                   0.004415
Non-Renewal Upheld            0.004345
No Cause For Action           0.003901
Non-Renewal Rescinded         0.003854
Cancellation Upheld           0.003784
Policy not written in CT      0.003738
Refer To Agency               0.002359
Federal                       0.001962
Policy Issued                 0.001939
Contract Violation            0.001822
Cancellation Withdrawn        0.001495
Rate Problem Solved           0.001402
Policy Not In Force           0.001308
Other [Enter Disposition]     0.001075
```

```
Coverage Extended            0.000888
Deductible Recovered         0.000584
Fees Returned                0.000561
Insured Retained Attorney    0.000420
Underwriting Guidelines      0.000374
Interest Paid                0.000350
Policy Offered               0.000280
Complaint Form Sent          0.000280
Class Revised                0.000280
Extl Rev Info Sent/SF        0.000210
Accident in Another State    0.000187
Underwriting Discretion      0.000140
Cease and Desist             0.000047
Cross Reference Only         0.000023
Med Jurisdiction Explained   0.000023
Filed Errors&Omission Clm    0.000023
Not Insurance Related        0.000023
Name: proportion, dtype: float64

Grouping 44 rare categories in Conclusion under 'Other'


Column: Status
Status
Closed                          0.975171
Sent to Company                 0.008734
Supervisor Review               0.005258
Open                            0.003184
Full Review - Standard          0.002308
Reopened                        0.001636
Interim Letter Sent             0.001212
Incomplete Follow-up            0.000759
New Doc                         0.000716
Awaiting Decision               0.000482
Missing Information             0.000161
Extension Granted               0.000146
Legal Review                    0.000058
Preliminary Review - Standard   0.000029
Sent to Agent                   0.000029
No Response Follow-up           0.000029
New                             0.000029
In Progress                     0.000029
Recovery Pending                0.000015
Verify Situs                    0.000015
Name: proportion, dtype: float64

Grouping 19 rare categories in Status under 'Other'

High-cardinality analysis complete. Modified data saved.
```

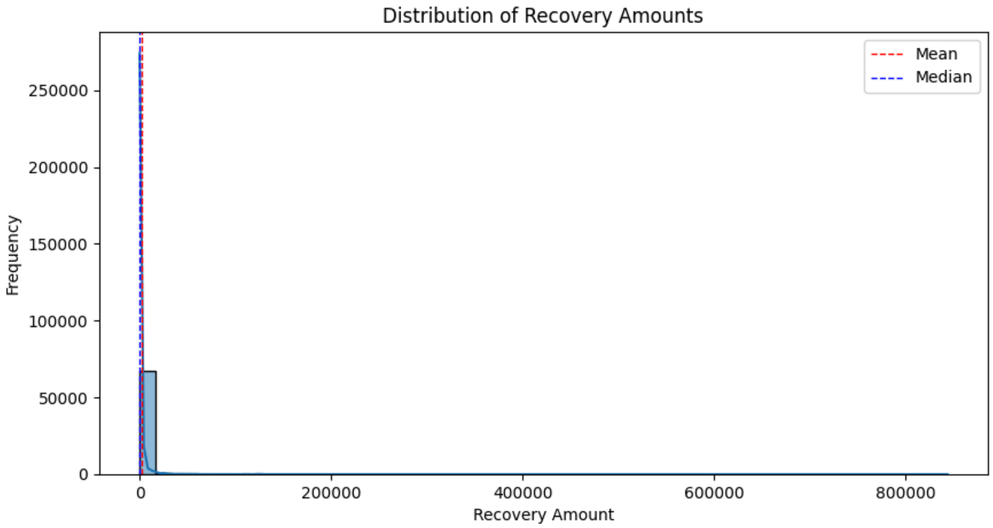
Here's what happened:

-In the Company column, 837 rare companies were grouped as "Other" since they had very low proportions in the dataset. In the Opened, Closed, Coverage, SubCoverage, SubReason, Disposition, Conclusion, and Status columns, categories that appeared infrequently were also grouped under "Other" to reduce complexity. This helps in:

-Reducing noise by merging very rare categories. Improving model performance by avoiding too many categories with little data. Making analysis and visualization clearer by focusing on the most relevant categories.

In [ ]:

```
file_path = "modified_data.csv"
df = pd.read_csv(file_path)

# Ensure 'Recovery' is numeric
df['Recovery'] = pd.to_numeric(df['Recovery'], errors='coerce')

# Define outliers using the IQR method
Q1 = df['Recovery'].quantile(0.25)
Q3 = df['Recovery'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Identify outliers
outliers = df[(df['Recovery'] < lower_bound) | (df['Recovery'] > upper_bounc
non_outliers = df[(df['Recovery'] >= lower_bound) & (df['Recovery'] <= upper

# Calculate statistics
mean_with_outliers = df['Recovery'].mean()
median_with_outliers = df['Recovery'].median()
mean_without_outliers = non_outliers['Recovery'].mean()
median_without_outliers = non_outliers['Recovery'].median()

# Print results
print(f"Mean with outliers: ${mean_with_outliers:.2f}")
print(f"Median with outliers: ${median_with_outliers:.2f}")
print(f"Mean without outliers: ${mean_without_outliers:.2f}")
print(f"Median without outliers: ${median_without_outliers:.2f}")



# Plot distribution
plt.figure(figsize=(10, 5))
sns.histplot(df['Recovery'], bins=50, kde=True)
plt.axvline(mean_with_outliers, color='red', linestyle='dashed', linewidth=1
plt.axvline(median_with_outliers, color='blue', linestyle='dashed', linewidt
plt.title('Distribution of Recovery Amounts')
plt.xlabel('Recovery Amount')
plt.ylabel('Frequency')
plt.legend()

# Save the plot
histogram_path = "recovery_histogram.png"
plt.savefig(histogram_path)
plt.close()

print(f"Histogram saved to: {histogram_path}")
```



Frequency and Counting Analysis

In [9]:

```
df = pd.read_csv('data.csv')

# Remove exact duplicate rows
df = df.drop_duplicates()

# Remove duplicate complaints based on File No.
df = df.drop_duplicates(subset=['File No.'], keep='first')

# Show the first few rows and column names
print("Preview of the dataset:")
print(df.head())
print("\nColumn names:")
print(df.columns)

# Frequency Analysis: Complaints by Company
print("\nTop 10 Companies by Complaint Count:")
top_companies = df['Company'].value_counts().head(10)
print(top_companies)

# Frequency Analysis with Percentages
def print_count_and_percentage(column_name):
    counts = df[column_name].value_counts()
    percentages = df[column_name].value_counts(normalize=True) * 100
    summary = pd.DataFrame({
        'Count': counts,
        'Percentage': percentages
    })
    print(f"\n{column_name} Distribution (Counts and Percentages):")
    print(summary)

print_count_and_percentage('Company')
print_count_and_percentage('Reason')
print_count_and_percentage('Status')
print_count_and_percentage('Conclusion')

# Detect Rare Classes
def print_rare_classes(column_name, threshold=5):
    counts = df[column_name].value_counts()
    rare_classes = counts[counts <= threshold]
    if not rare_classes.empty:
        print(f"\nRare categories in {column_name} (<= {threshold} occurrenc
        print(rare_classes)
    else:
        print(f"\nNo rare categories found in {column_name} (<= {threshold}

print_rare_classes('Company')
print_rare_classes('Reason')
print_rare_classes('Status')
print_rare_classes('Conclusion')

# Visualization: Top 10 Companies by Complaints
plt.figure(figsize=(10, 6))
sns.barplot(x=top_companies.values, y=top_companies.index, palette='Blues_r'
plt.title('Top 10 Companies by Complaint Count')
plt.xlabel('Number of Complaints')
plt.ylabel('Company')
plt.tight_layout()
plt.savefig('top_companies_barplot.png')
plt.close()

# Visualization: Top Complaint Reasons
top_reasons = df['Reason'].value_counts().head(5)
plt.figure(figsize=(10, 6))
sns.barplot(x=top_reasons.values, y=top_reasons.index, palette='muted')
plt.title('Top Complaint Reasons')
plt.xlabel('Number of Complaints')
plt.ylabel('Reason')
plt.tight_layout()
plt.savefig('top_complaint_reasons_barplot.png')
plt.close()

# Visualization: Complaint Status Distribution (Pie Chart)
plt.figure(figsize=(8, 8))
plt.pie(df['Status'].value_counts().head(6), labels=df['Status'].value_count
plt.title('Complaint Status Distribution')
plt.tight_layout()
plt.savefig('complaint_status_pie_chart.png')
plt.close()

# Visualization: Complaint Conclusion Distribution (Pie Chart)
plt.figure(figsize=(8, 8))
plt.pie(df['Conclusion'].value_counts().head(6), labels=df['Conclusion'].val
plt.title('Complaint Conclusion Distribution')
plt.tight_layout()
plt.savefig('complaint_conclusion_pie_chart.png')
plt.close()
```

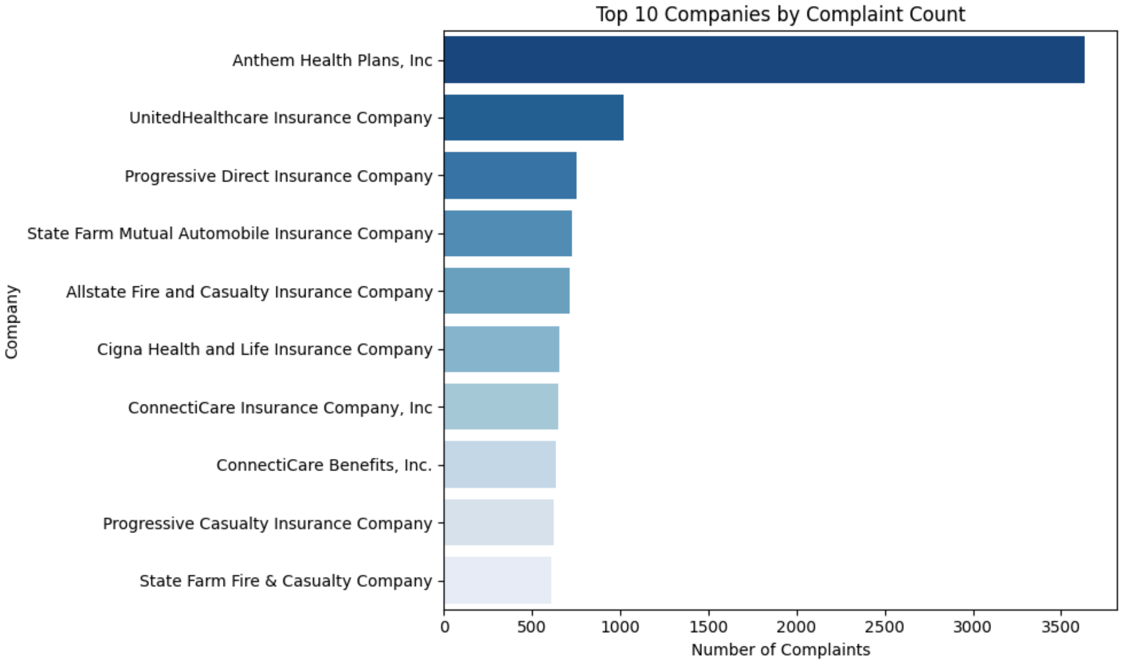
```
Preview of the dataset:
                         Company  File No.      Opened      Closed Coverage
\
0      Anthem Health Plans, Inc   7045593  05/31/2022  06/02/2022    Group
1      Anthem Health Plans, Inc   7043381  02/28/2022  06/02/2022    Group
2      Anthem Health Plans, Inc   7044860  05/03/2022  06/02/2022    A & H
4      Anthem Health Plans, Inc   7052007  02/23/2023  03/17/2023    A & H
5  Oxford Health Plans (CT), Inc   7054762  06/01/2023  08/02/2023    A & H

   SubCoverage             Reason                SubReason  \
0  Health Only     Claim Handling      Medically Necessary
1  Health Only     Claim Handling  Provider Contract Issue
2  Health Only     Claim Handling                   Denial
4        A & H  Marketing & Sales       Duplicate Coverage
5  Health Only     Claim Handling          External Review

                         Disposition              Conclusion  Recovery  \
0     Company Position Substantiated  Company Position Upheld      0.00
1                      Claim Settled                Satisfied   6467.30
2                      Claim Settled               Claim Paid    147.58
4  Compromised Settlement/Resolution           Premium Refund   2179.32
5                                NaN                      NaN      0.00

   Status
0  Closed
1  Closed
2  Closed
4  Closed
5  Closed

Column names:
Index(['Company', 'File No.', 'Opened', 'Closed', 'Coverage', 'SubCoverage',
       'Reason', 'SubReason', 'Disposition', 'Conclusion', 'Recovery',
       'Status'],
      dtype='object')

Top 10 Companies by Complaint Count:
Company
Anthem Health Plans, Inc                          3657
UnitedHealthcare Insurance Company                1024
Progressive Direct Insurance Company               755
State Farm Mutual Automobile Insurance Company     732
Allstate Fire and Casualty Insurance Company       722
Cigna Health and Life Insurance Company            663
ConnectiCare Insurance Company, Inc                651
ConnectiCare Benefits, Inc.                        643
Progressive Casualty Insurance Company             619
State Farm Fire & Casualty Company                 611
Name: count, dtype: int64

Company Distribution (Counts and Percentages):
                                                  Count  Percentage
Company
Anthem Health Plans, Inc                           3657   12.798348
UnitedHealthcare Insurance Company                 1024    3.583677
Progressive Direct Insurance Company                755    2.642262
```

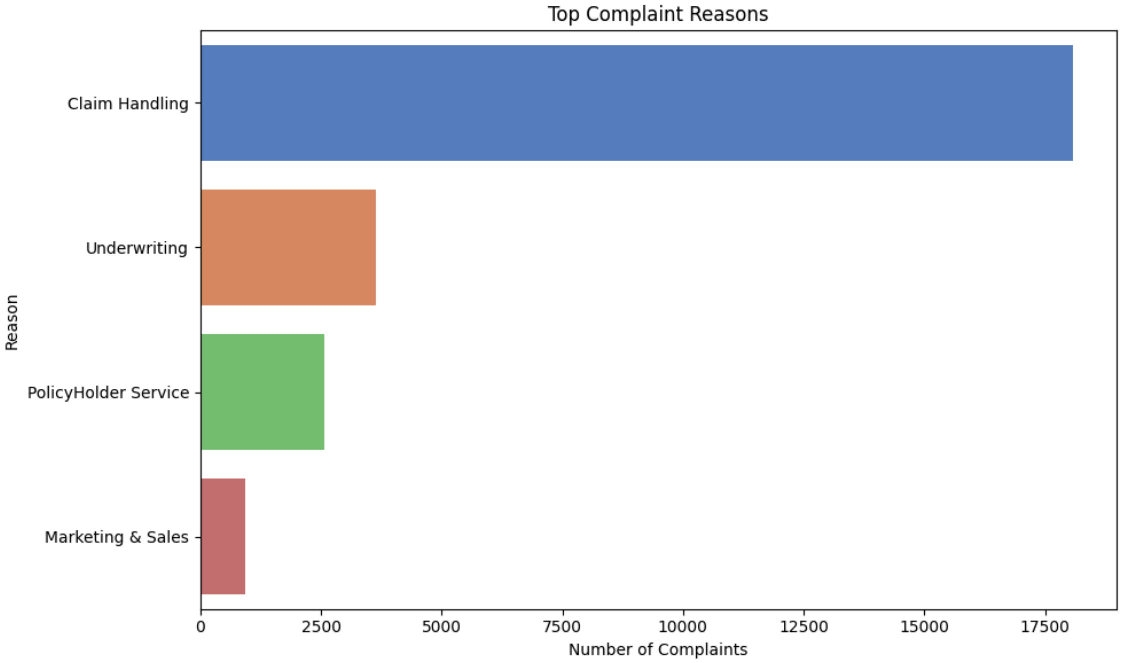
```
State Farm Mutual Automobile Insurance Company      732    2.561769
Allstate Fire and Casualty Insurance Company        722    2.526773
...                                                 ...         ...
Colony Specialty Insurance Company                    1    0.003500
First Allmerica Financial Life Insurance Company      1    0.003500
Capitol Specialty Insurance Corporation               1    0.003500
21st Century Premier Insurance Company                1    0.003500
ARAG Insurance Company                                1    0.003500

[825 rows x 2 columns]

Reason Distribution (Counts and Percentages):
                      Count  Percentage
Reason
Claim Handling        18183   71.724981
Underwriting           3655   14.417577
PolicyHolder Service   2576   10.161335
Marketing & Sales       937    3.696107

Status Distribution (Counts and Percentages):
                               Count  Percentage
Status
Closed                         28061   98.204662
Sent to Company                  187    0.654441
Open                             185    0.647442
New Doc                           33    0.115490
Supervisor Review                 25    0.087492
Full Review - Standard            20    0.069994
Reopened                          14    0.048996
Incomplete Follow-up              13    0.045496
Interim Letter Sent               11    0.038497
Awaiting Decision                  6    0.020998
Missing Information                4    0.013999
Extension Granted                  3    0.010499
Preliminary Review - Standard      2    0.006999
Sent to Agent                      2    0.006999
In Progress                        2    0.006999
New                                2    0.006999
Recovery Pending                   1    0.003500
Legal Review                       1    0.003500
Verify Situs                       1    0.003500
No Response Follow-up              1    0.003500

Conclusion Distribution (Counts and Percentages):
                            Count  Percentage
Conclusion
Justified                    1430   12.459702
Company Position Upheld      1192   10.385989
Furnished Information         988    8.608521
Claim Paid                    919    8.007319
Corrective Action             855    7.449682
Contract Provision            839    7.310273
Refer-Judicial/Attorney       826    7.197003
Unjustified                   590    5.140716
Enter Arbitration             336    2.927594
Provider Issue                298    2.596497
```

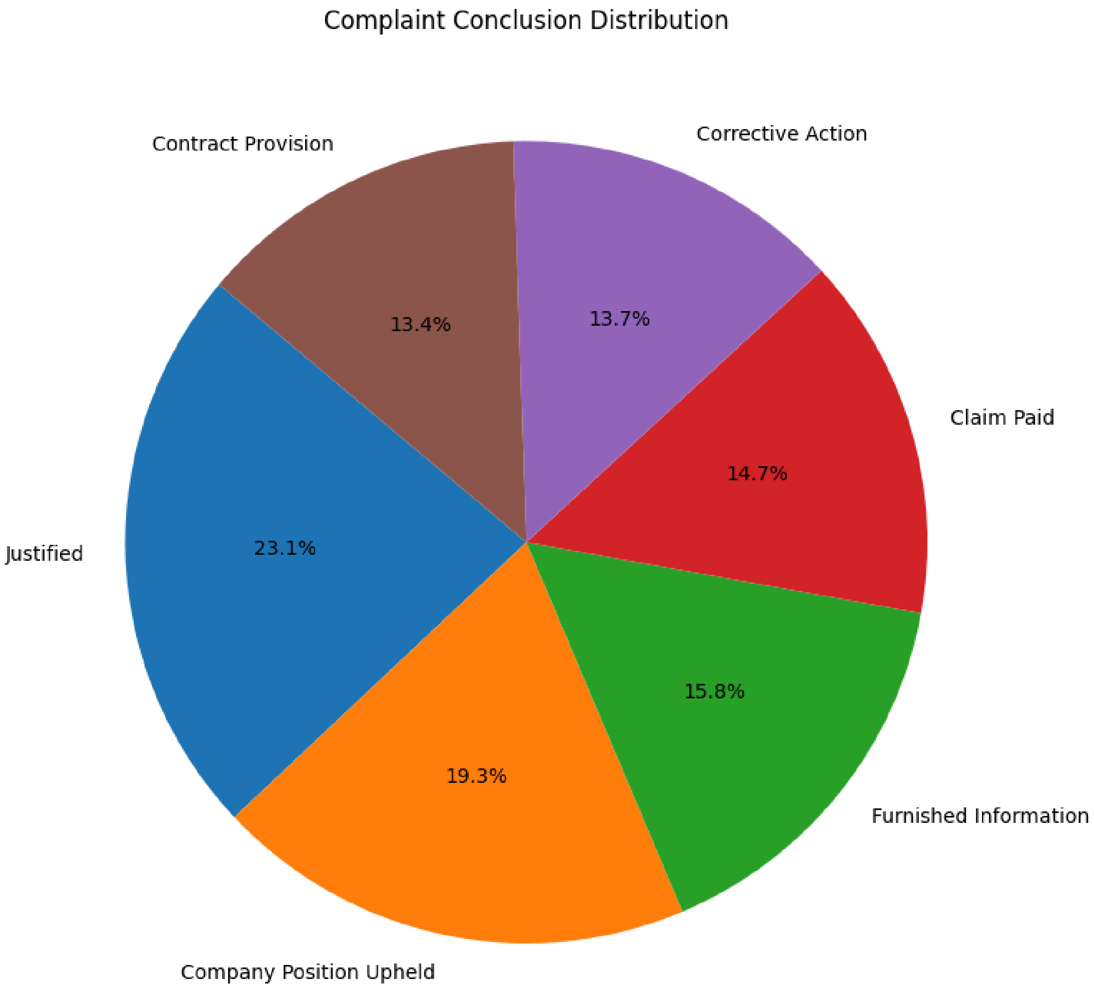
```
No Action Necessary           258    2.247974
Coverage Granted              246    2.143417
Voluntary Reconsideration     233    2.030147
Claim Paid With Interest      211    1.838460
Questionable                  207    1.803607
Refer To Appraisal            182    1.585780
Premium Refund                170    1.481223
Satisfactory Explanation      161    1.402806
Additional Money Received     159    1.385379
Coverage Denied               152    1.324388
Non-Renewal Upheld            148    1.289536
Rate Increase Explained       139    1.211118
Policy Restored/Reinstated    131    1.141413
No Cause For Action           102    0.888734
Policy not written in CT       99    0.862595
Cancellation Upheld            94    0.819029
No Authority                   87    0.758038
Record Only                    75    0.653481
Non-Renewal Rescinded          74    0.644768
External Review Info Sent      71    0.618629
Satisfied                      41    0.357236
Cancellation Withdrawn         29    0.252679
Contract Violation             20    0.174262
Refer To Agency                16    0.139409
Other [Enter Disposition]      13    0.113270
Policy Issued                  12    0.104557
Federal                        12    0.104557
Policy Not In Force            11    0.095844
Underwriting Guidelines         9    0.078418
Coverage Extended               7    0.060992
Fees Returned                   7    0.060992
Accident in Another State       5    0.043565
Underwriting Discretion         4    0.034852
Rate Problem Solved             3    0.026139
Interest Paid                   3    0.026139
Extl Rev Info Sent/SF           2    0.017426
Deductible Recovered            2    0.017426
Policy Offered                  2    0.017426
Class Revised                   1    0.008713
Complaint Form Sent             1    0.008713
Cross Reference Only            1    0.008713
Cease and Desist                1    0.008713
Med Jurisdiction Explained      1    0.008713
Filed Errors&Omission Clm       1    0.008713
Not Insurance Related           1    0.008713

Rare categories in Company (<= 5 occurrences):
Company
C.M. Life Insurance Company                         5
Northfield Insurance Company                        5
Markel Insurance Company                            5
Lincoln Life Assurance Company of Boston            5
United Financial Casualty Company                   5
                                                   ..
Colony Specialty Insurance Company                  1
First Allmerica Financial Life Insurance Company    1
```

```
Capitol Specialty Insurance Corporation             1
21st Century Premier Insurance Company              1
ARAG Insurance Company                              1
Name: count, Length: 453, dtype: int64

No rare categories found in Reason (<= 5 occurrences).

Rare categories in Status (<= 5 occurrences):
Status
Missing Information             4
Extension Granted               3
Preliminary Review - Standard   2
Sent to Agent                   2
In Progress                     2
New                             2
Recovery Pending                1
Legal Review                    1
Verify Situs                    1
No Response Follow-up           1
Name: count, dtype: int64

Rare categories in Conclusion (<= 5 occurrences):
Conclusion
Accident in Another State    5
Underwriting Discretion      4
Rate Problem Solved          3
Interest Paid                3
Extl Rev Info Sent/SF        2
Deductible Recovered         2
Policy Offered               2
Class Revised                1
Complaint Form Sent          1
Cross Reference Only         1
Cease and Desist             1
Med Jurisdiction Explained   1
Filed Errors&Omission Clm    1
Not Insurance Related        1
Name: count, dtype: int64
```

```
<ipython-input-9-3ff3d4ef68f3>:53: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the
same effect.

  sns.barplot(x=top_companies.values, y=top_companies.index, palette='Blues_
r')
<ipython-input-9-3ff3d4ef68f3>:64: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the
same effect.

  sns.barplot(x=top_reasons.values, y=top_reasons.index, palette='muted')
```

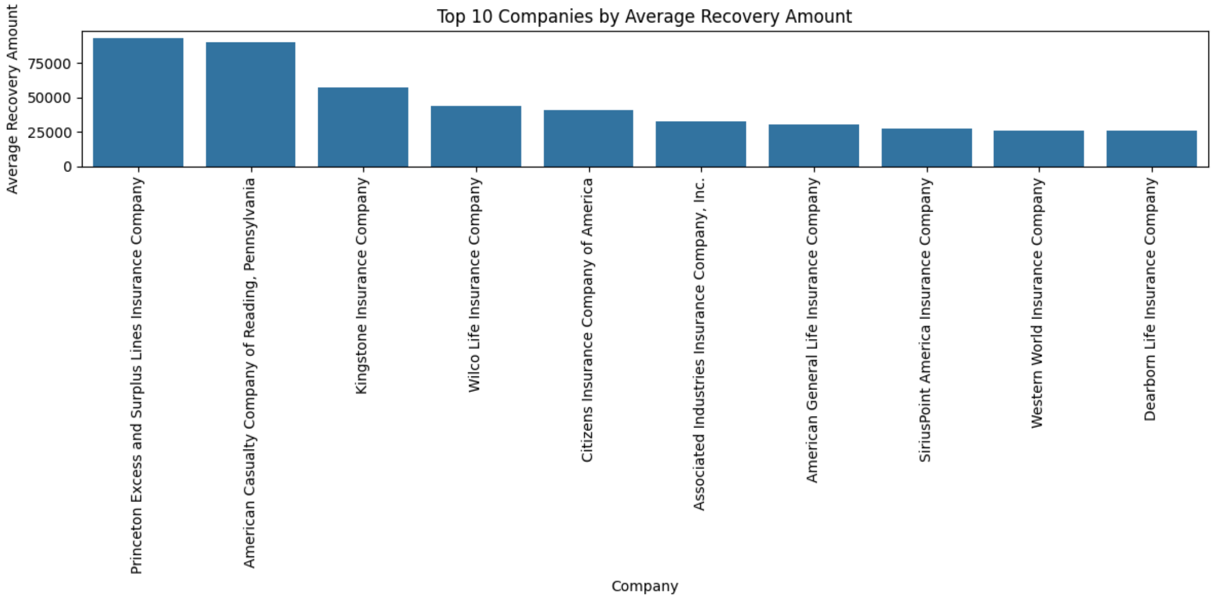

Top 10 Companies by Complaint Count
Company
Anthem Health Plans, Inc
UnitedHealthcare Insurance Company
Progressive Direct Insurance Company
State Farm Mutual Automobile Insurance Company
Allstate Fire and Casualty Insurance Company
Cigna Health and Life Insurance Company
ConnectiCare Insurance Company, Inc
ConnectiCare Benefits, Inc.
Progressive Casualty Insurance Company
State Farm Fire & Casualty Company
0
500
1000
1500
2000
2500
3000
3500
Number of Complaints


Top Complaint Reasons
Reason
Claim Handling
Underwriting
PolicyHolder Service
Marketing & Sales
0
2500
5000
7500
10000
12500
15000
17500
Number of Complaints



Extra

In [10]:
```
df = pd.read_csv('data.csv')

if 'Company' in df.columns and 'Recovery' in df.columns:

    company_recovery_avg = df.groupby('Company')['Recovery'].mean().reset_ir


    company_recovery_count = df.groupby('Company').agg(
        recovered_count=('Recovery', lambda x: (x > 0).sum()),  # Count non-
        not_recovered_count=('Recovery', lambda x: (x == 0).sum())  # Count 
    ).reset_index()


    company_recovery_stats = pd.merge(company_recovery_avg, company_recovery


    company_recovery_stats['recovery_percentage'] = (company_recovery_stats[
                                                      (company_recovery_stat
                                                       company_recovery_stat


    top_companies = company_recovery_stats.nlargest(10, 'Recovery')


    plt.figure(figsize=(12, 6))
    sns.barplot(x='Company', y='Recovery', data=top_companies)
    plt.xticks(rotation=90)
    plt.title('Top 10 Companies by Average Recovery Amount')
    plt.xlabel('Company')
    plt.ylabel('Average Recovery Amount')
    plt.tight_layout()

    # Save the graph as a PNG file
    plt.savefig('top_10_companies_vs_avg_recovery.png')

    # Optionally, display the plot
    plt.show()

    # Print the recovery stats for the top 10 companies
    print(top_companies[['Company', 'Recovery', 'recovered_count', 'not_reco

else:
    print("Necessary columns 'Company' and 'Recovery' are missing.")
```



```
                                                 Company      Recovery  \
615  Princeton Excess and Surplus Lines Insurance C...  93272.360000
57   American Casualty Company of Reading, Pennsylv...  89733.333333
426                         Kingstone Insurance Company  57435.938261
834                         Wilco Life Insurance Company  44071.456000
182                  Citizens Insurance Company of America  40780.159200
110         Associated Industries Insurance Company, Inc.  32320.254000
67                  American General Life Insurance Company  30556.362815
681                  SiriusPoint America Insurance Company  27397.000000
831                       Western World Insurance Company  26016.394167
232                       Dearborn Life Insurance Company  25961.538462

     recovered_count  not_recovered_count  recovery_percentage
615                2                    0           100.000000
57                 5                    4            55.555556
426               14                   32            30.434783
834                2                    3            40.000000
182                6                   19            24.000000
110                1                    4            20.000000
67                17                  118            12.592593
681                1                    1            50.000000
831                3                   21            12.500000
232                3                   10            23.076923
```

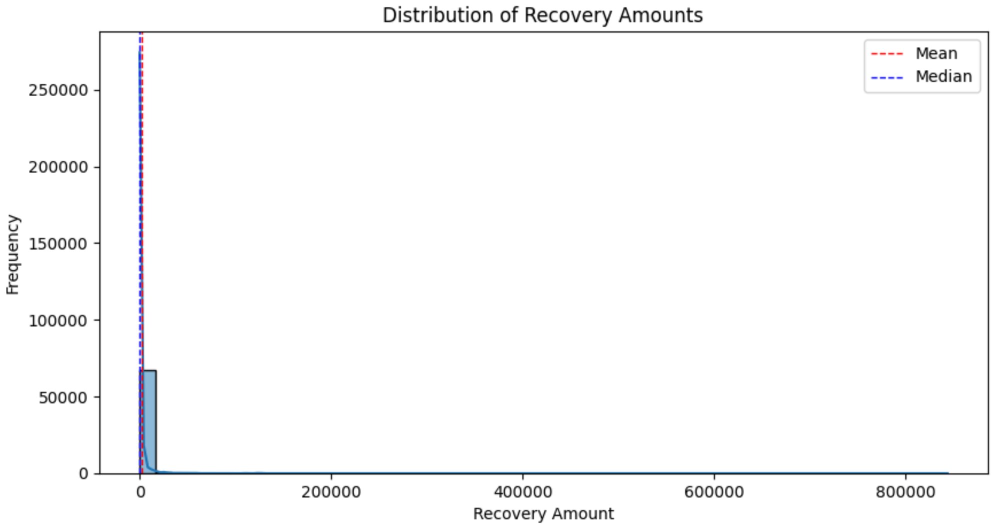
Recovery Analysis (with and without outliers)

In [ ]:

```python
file_path = "modified_data.csv"

# Ensure 'Recovery' is numeric
df['Recovery'] = pd.to_numeric(df['Recovery'], errors='coerce')

# Define outliers using the IQR method
Q1 = df['Recovery'].quantile(0.25)
Q3 = df['Recovery'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Identify outliers
outliers = df[(df['Recovery'] < lower_bound) | (df['Recovery'] > upper_bounc
non_outliers = df[(df['Recovery'] >= lower_bound) & (df['Recovery'] <= upper

# Calculate statistics
mean_with_outliers = df['Recovery'].mean()
median_with_outliers = df['Recovery'].median()
mean_without_outliers = non_outliers['Recovery'].mean()
median_without_outliers = non_outliers['Recovery'].median()

# Print results
print(f"Mean with outliers: ${mean_with_outliers:.2f}")
print(f"Median with outliers: ${median_with_outliers:.2f}")
print(f"Mean without outliers: ${mean_without_outliers:.2f}")
print(f"Median without outliers: ${median_without_outliers:.2f}")
```

```python
# Plot distribution
plt.figure(figsize=(10, 5))
sns.histplot(df['Recovery'], bins=50, kde=True)
plt.axvline(mean_with_outliers, color='red', linestyle='dashed', linewidth=1
plt.axvline(median_with_outliers, color='blue', linestyle='dashed', linewidt
plt.title('Distribution of Recovery Amounts')
plt.xlabel('Recovery Amount')
plt.ylabel('Frequency')
plt.legend()

# Save the plot
histogram_path = "recovery_histogram.png"
plt.savefig(histogram_path)
plt.close()

print(f"Histogram saved to: {histogram_path}")
```

```
Mean with outliers: $1741.72
Median with outliers: $0.00
Mean without outliers: $0.00
Median without outliers: $0.00
Histogram saved to: recovery_histogram.png
```


Distribution of Recovery Amounts
Mean
Median
Frequency
Recovery Amount