

```
!pip install pandas numpy matplotlib seaborn
```

Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
 Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (2.0.2)
 Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (3.
 Requirement already satisfied: seaborn in /usr/local/lib/python3.11/dist-packages (0.13.
 Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-
 Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (
 Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages
 Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packag
 Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (
 Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packa
 Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packa
 Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-package
 Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (fro
 Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packag
 Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from

```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.moun

```
import pandas as pd
```

```
file_path = '/content/drive/My Drive/insurance_data.txt' # Update if in a subf
df = pd.read_csv(file_path, sep='|', encoding='latin1')
print("First 5 rows:\n", df.head())
print("\nColumns:", df.columns.tolist())
```



rTerm', 'ExcessSelected', 'CoverCategory', 'CoverType', 'CoverGroup', 'Section', 'Product'

Double-click (or enter) to edit

```
df.rename(columns={
    'make': 'Make',
    'mmcode': 'Mmcode',
    'cubiccapacity': 'Cubiccapacity',
    'kilowatts': 'Kilowatts',
    'bodytype': 'Bodytype'
}, inplace=True)
print("Updated Columns:", df.columns.tolist())
```

Updated Columns: ['UnderwrittenCoverID', 'PolicyID', 'TransactionMonth', 'IsVATRegistered']

```
import matplotlib.pyplot as plt
import seaborn as sns
province_lr = df.groupby('Province')['LossRatio'].mean()
print("Loss Ratio by Province:\n", province_lr)
```

```
plt.figure(figsize=(8, 5))
sns.barplot(x=province_lr.index, y=province_lr.values, palette='viridis')
plt.title('Average Loss Ratio by Province')
plt.ylabel('Loss Ratio')
plt.xticks(rotation=45)
plt.show()
```



Loss Ratio by Province:

Province

Eastern Cape 0.235589

Free State 0.106209

Gauteng 0.428889

KwaZulu-Natal 0.264746

Limpopo 0.348712

Mpumalanga 0.392698

North West 0.285348

Northern Cape 0.203831

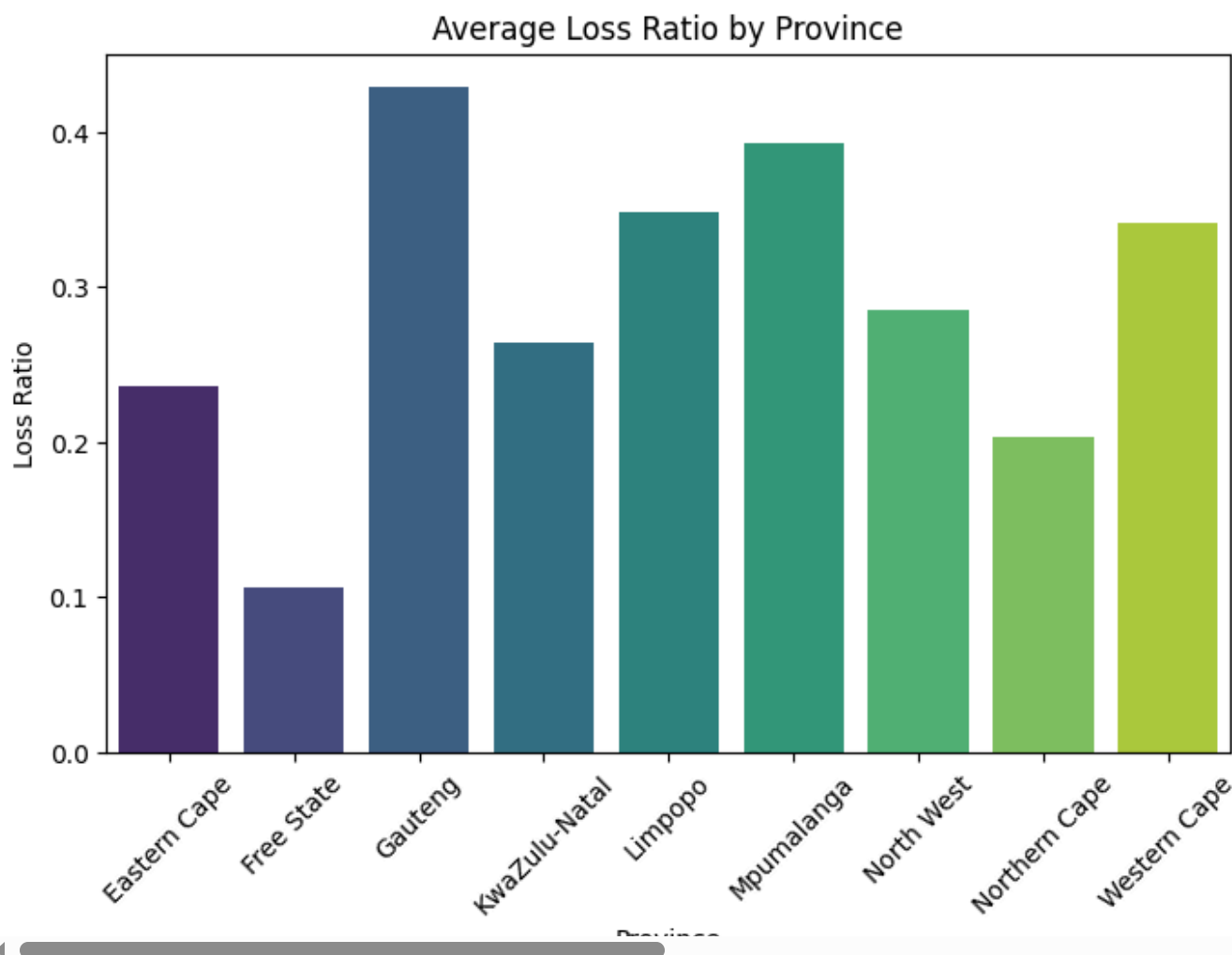
Western Cape 0.34175

Name: LossRatio, dtype: object

<ipython-input-24-3430488082>:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

```
sns.barplot(x=province_lr.index, y=province_lr.values, palette='viridis')
```



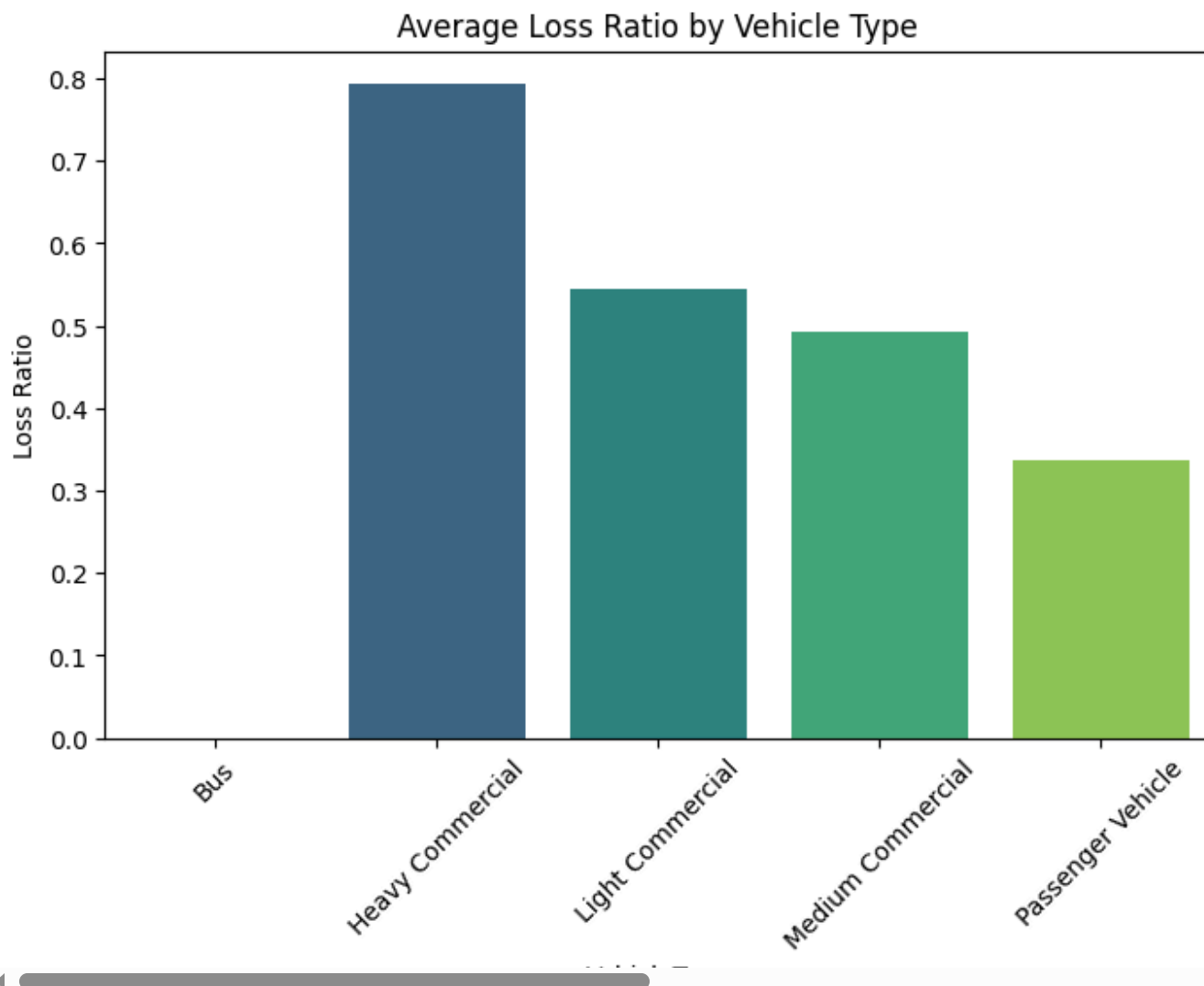
```
vehicle_lr = df.groupby('VehicleType')['LossRatio'].mean()
print("Loss Ratio by VehicleType:\n", vehicle_lr)
```

```
plt.figure(figsize=(8, 5))
sns.barplot(x=vehicle_lr.index, y=vehicle_lr.values, palette='viridis')
plt.title('Average Loss Ratio by Vehicle Type')
plt.ylabel('Loss Ratio')
plt.xticks(rotation=45)
plt.show()
```

```
➞ Loss Ratio by VehicleType:
VehicleType
Bus                0.0
Heavy Commercial   0.793617
Light Commercial   0.543917
Medium Commercial  0.493356
Passenger Vehicle  0.337368
Name: LossRatio, dtype: object
<ipython-input-17-3189275291>:5: FutureWarning:
```

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

```
sns.barplot(x=vehicle_lr.index, y=vehicle_lr.values, palette='viridis')
```



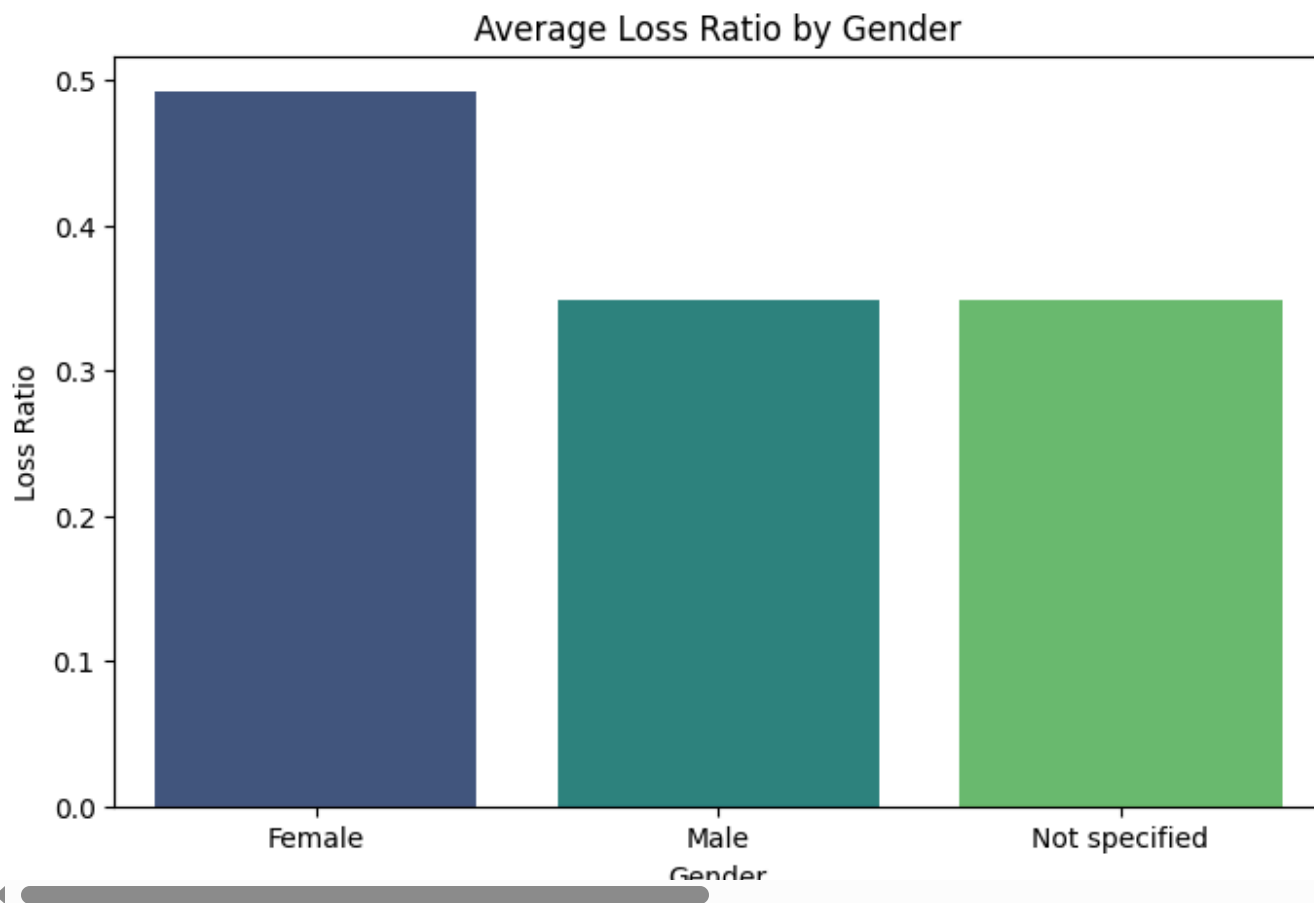
```
gender_lr = df.groupby('Gender')['LossRatio'].mean()
print("Loss Ratio by Gender:\n", gender_lr)
```

```
plt.figure(figsize=(8, 5))
sns.barplot(x=gender_lr.index, y=gender_lr.values, palette='viridis')
plt.title('Average Loss Ratio by Gender')
plt.ylabel('Loss Ratio')
plt.show()
```

```
➡ Loss Ratio by Gender:
Gender
Female      0.491979
Male        0.348852
Not specified 0.347904
Name: LossRatio, dtype: object
<ipython-input-18-285509109>:5: FutureWarning:
```

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

```
sns.barplot(x=gender_lr.index, y=gender_lr.values, palette='viridis')
```



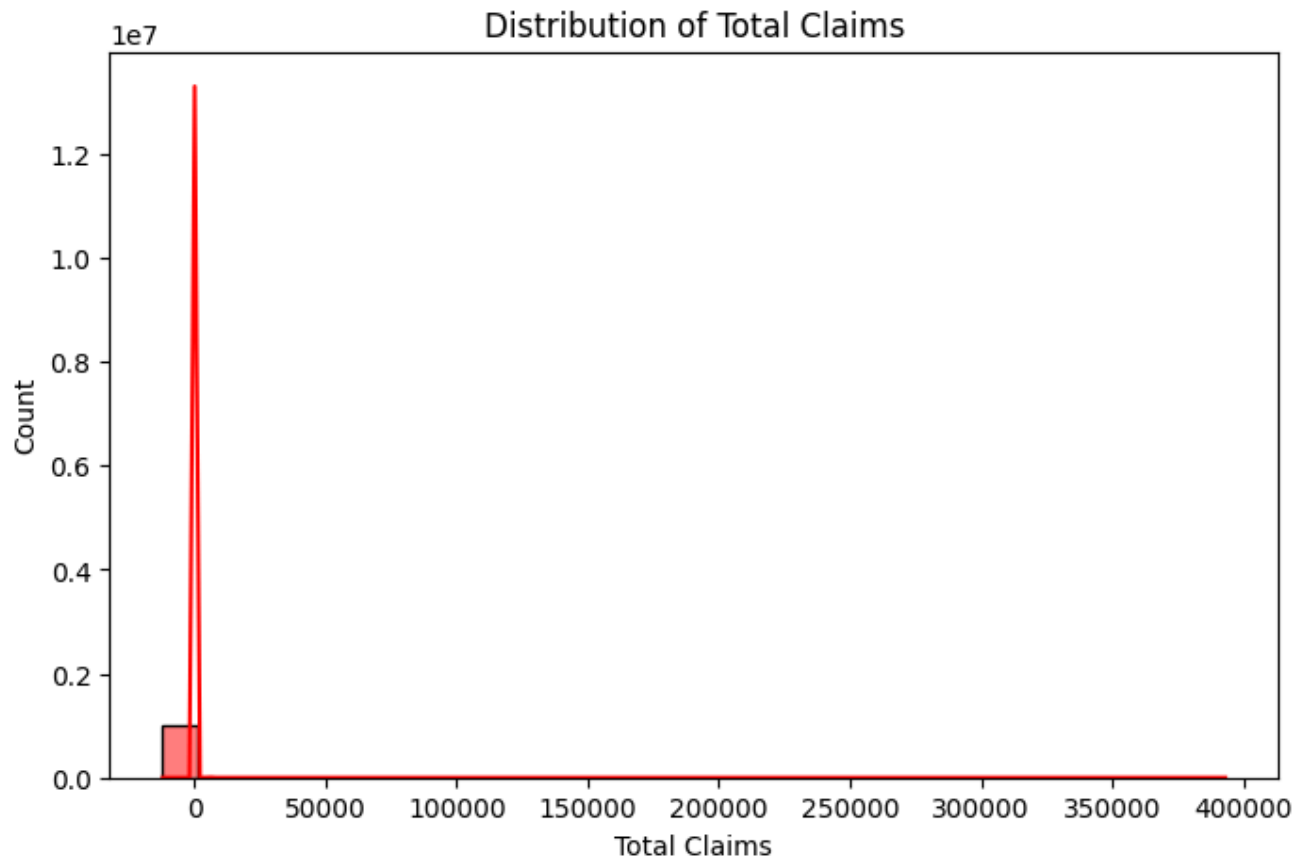
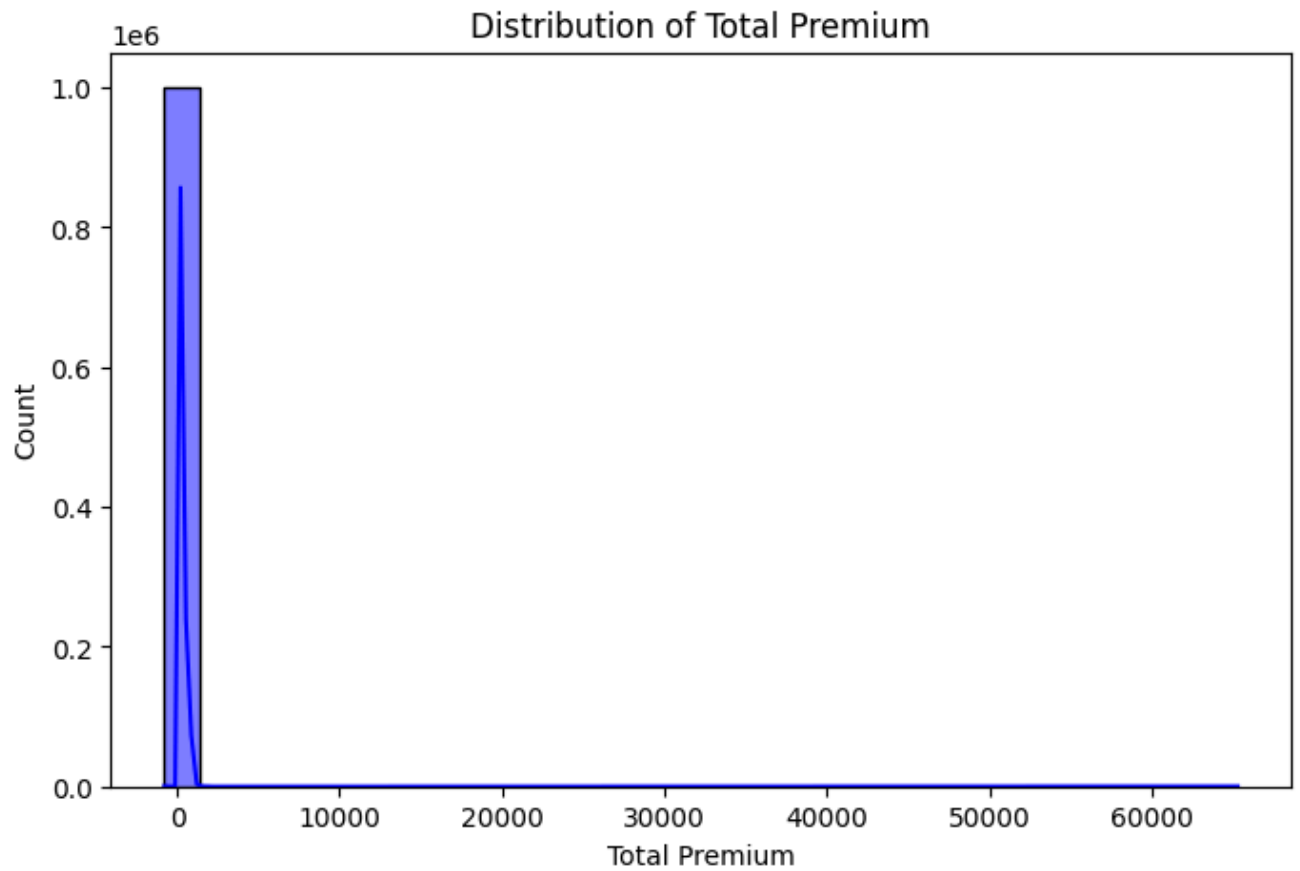
```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
plt.figure(figsize=(8, 5))
sns.histplot(df['TotalPremium'], bins=30, kde=True, color='blue')
```

```
plt.title('Distribution of Total Premium')
plt.xlabel('Total Premium')
plt.show()

plt.figure(figsize=(8, 5))
sns.histplot(df['TotalClaims'], bins=30, kde=True, color='red')
plt.title('Distribution of Total Claims')
plt.xlabel('Total Claims')
plt.show()

print("TotalPremium Stats:\n", df['TotalPremium'].describe())
print("TotalClaims Stats:\n", df['TotalClaims'].describe())
```



TotalPremium Stats:

```
count    1.000098e+06
mean      6.190550e+01
std       2.302845e+02
min      -7.825768e+02
```

```

25%      0.000000e+00
50%      2.178333e+00
75%      2.192982e+01
max       6.528260e+04
Name: TotalPremium, dtype: float64
TotalClaims Stats:
  count      1.000098e+06
  mean       6.486119e+01
  std        2.384075e+03
  min       -1.200241e+04
  25%        0.000000e+00
  50%        0.000000e+00

```

```
# Temporal Trends
```

```

df['TransactionMonth'] = pd.to_datetime(df['TransactionMonth'], errors='coerce')
monthly_trends = df.groupby(df['TransactionMonth'].dt.to_period('M')).agg({
    'TotalClaims': ['mean', 'count']
}).reset_index()
monthly_trends.columns = ['Month', 'AvgClaims', 'ClaimCount']
monthly_trends['Month'] = monthly_trends['Month'].dt.to_timestamp() # Convert

```

```

plt.figure(figsize=(10, 5))
sns.lineplot(x='Month', y='AvgClaims', data=monthly_trends, label='Average Clai
plt.title('Temporal Trends in Claims')
plt.xticks(rotation=45)
plt.show()

```

```

plt.figure(figsize=(10, 5))
sns.lineplot(x='Month', y='ClaimCount', data=monthly_trends, label='Claim Count
plt.title('Temporal Trends in Claim Frequency')
plt.xticks(rotation=45)
plt.show()

```




Temporal Trends in Claims

