

Data Explorer Team Details

Group Name: Data Explorer

Name: Mohammad Tohin Bapari

Email: tohin@gmx.de

Country: Germany

University: Bergische Universität Wuppertal

Specialization: Data Science

Problem Description

Pharmaceutical companies need to understand the persistency of drug usage as per physician prescriptions to improve patient adherence. ABC Pharma aims to automate the process of identifying factors impacting persistency. The goal is to build a classification model to predict whether a patient will be persistent or non-persistent based on various demographic, clinical, and treatment-related factors.

Github Link

<https://github.com/iamtohin/Data-Analytst-Internship-at-Data-Glacier/tree/main/Week%2010>

```
In [4]: # Importing Libraries
import pandas as pd
import numpy as np
from sklearn.impute import SimpleImputer
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns
from IPython.display import display
```

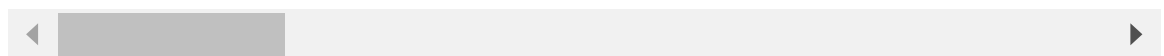
```
In [5]: # Load the dataset
file_path = 'Cleaned_Healthcare_Dataset.xlsx'
data = pd.read_excel(file_path, 'Dataset')
```

```
In [6]: # Display the first few rows of the dataset to understand its structure
data.head()
```

Out[6]:

	Ptid	Persistency_Flag	Gender	Race	Ethnicity	Region	Age_Bucket	Ntr
0	P1	Persistent	Male	Caucasian	Not Hispanic	West	>75	PR/
1	P2	Non-Persistent	Male	Asian	Not Hispanic	West	55-65	PR/
2	P3	Non-Persistent	Female	Other/Unknown	Hispanic	Midwest	65-75	PR/
3	P4	Non-Persistent	Female	Caucasian	Not Hispanic	Midwest	>75	PR/
4	P5	Non-Persistent	Female	Caucasian	Not Hispanic	Midwest	>75	PR/

5 rows × 69 columns



Descriptive analysis

```
In [8]: # Statistical Summary of Numerical Variables
numerical_summary = data.describe()
```

```
In [9]: # Frequency Counts for Categorical Variables
categorical_columns = ['Persistency_Flag', 'Gender', 'Race', 'Ethnicity', 'Region']
categorical_summary = ({col: data[col].value_counts() for col in categorical_columns})
```

Descriptive Analysis Results

Overview of Numerical Variables

Numerical Summary:

> Dexa_Freq_During_Rx and Count_Of_Risks are the numerical variables present in the dataset.

> Below are the descriptive statistics for these variables:

```
In [12]: display(numerical_summary)
```

	Dexa_Freq_During_Rx	Count_Of_Risks
count	2850.000000	2850.000000
mean	0.582456	1.191579
std	1.586748	1.041673
min	0.000000	0.000000
25%	0.000000	0.000000
50%	0.000000	1.000000
75%	0.000000	2.000000
max	6.000000	4.000000

Frequency Counts for Categorical Variables

Persistency Flag:

Non-Persistent: 2034 (71.4%)

Persistent: 816 (28.6%)

Gender:

Female: 2681 (94.1%)

Male: 169 (5.9%)

Race:

Caucasian: 2611 (91.6%)

African American: 92 (3.2%)

Other/Unknown: 78 (2.7%)

Asian: 69 (2.4%)

Ethnicity:

Not Hispanic: 2698 (94.7%)

Hispanic: 77 (2.7%)

Unknown: 75 (2.6%)

Region:

Midwest: 1163 (40.8%)

South: 1077 (37.8%)

West: 372 (13.1%)

Northeast: 191 (6.7%)

Other/Unknown: 47 (1.6%)

Age Bucket:

75: 1221 (42.8%)

65-75: 881 (30.9%)

55-65: 606 (21.3%)

<55: 142 (5.0%)

Ntm Speciality:

General Practitioner: 1306 (45.8%)

Rheumatology: 494 (17.3%)

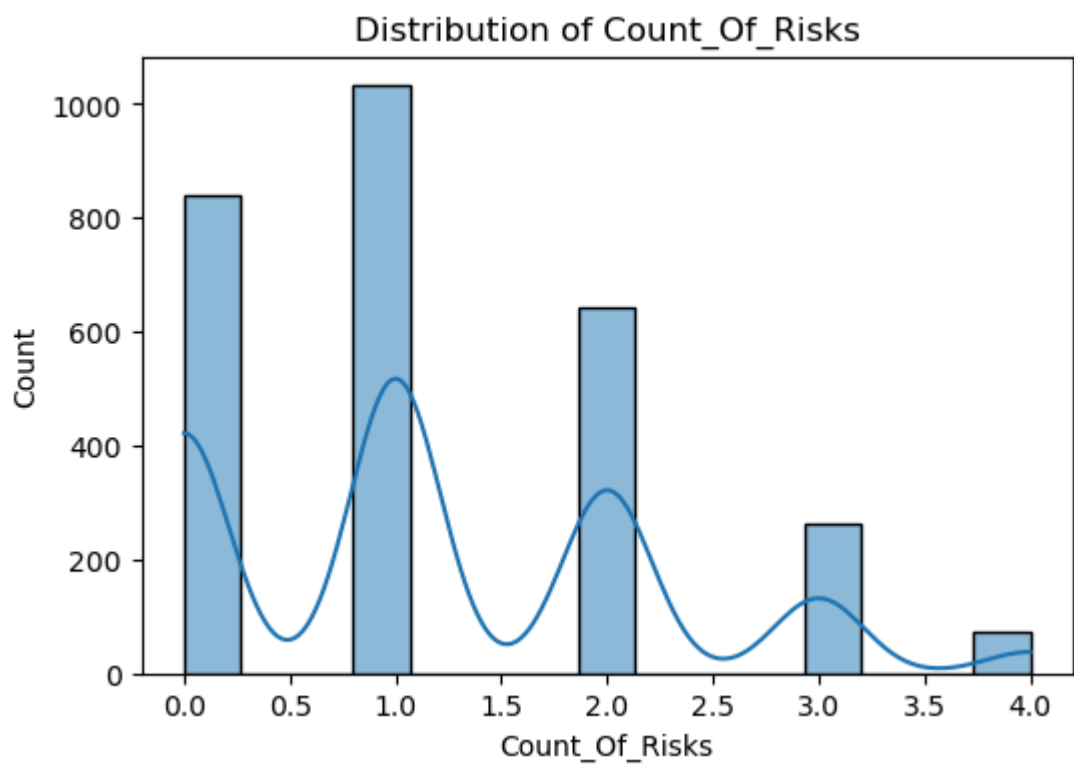
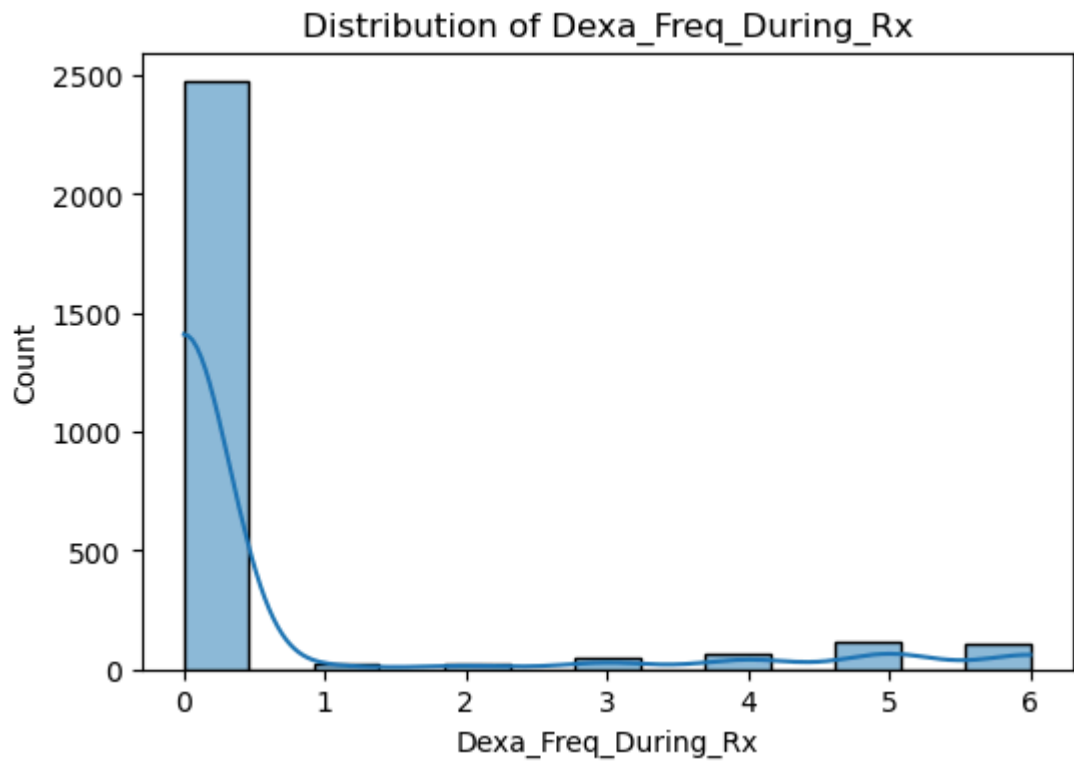
Endocrinology: 345 (12.1%)

Unknown: 282 (9.9%)

Others: Remaining values

Other Categorical Variables: Similar detailed frequency counts have been generated for each categorical variable.

```
In [14]: # Histograms for numerical variables
numerical_columns = data.select_dtypes(include=['int64', 'float64']).columns
for col in numerical_columns:
    plt.figure(figsize=(6, 4))
    sns.histplot(data[col], kde=True)
    plt.title(f'Distribution of {col}')
    plt.show()
```



Correlation Analysis Plan

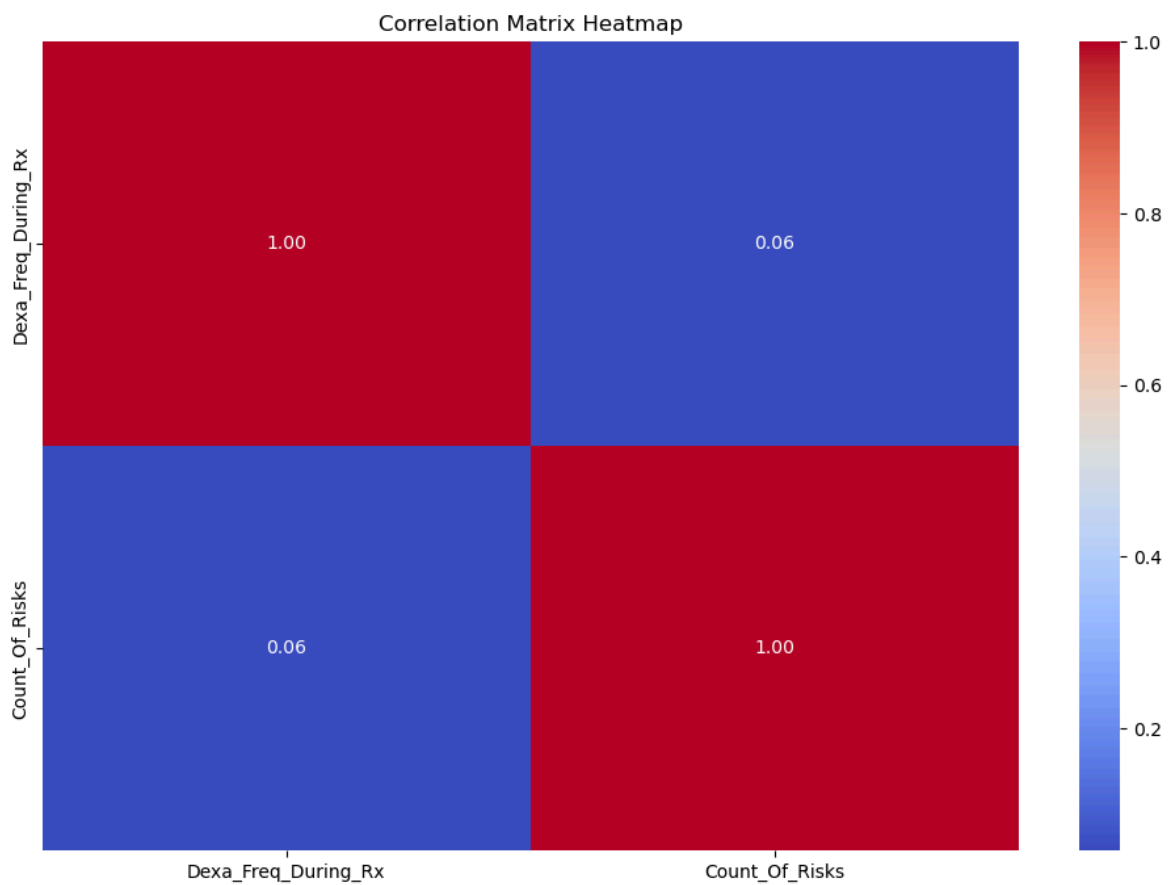
- Compute the correlation matrix.
- Display the correlation matrix.
- Visualize the correlation matrix using a heatmap.

```
In [30]: # Compute the correlation matrix
correlation_matrix = data.corr(numeric_only=True)
print("Correlation Matrix:")
print(correlation_matrix)
```

Correlation Matrix:

	Dexa_Freq_During_Rx	Count_Of_Risks
Dexa_Freq_During_Rx	1.000000	0.057757
Count_Of_Risks	0.057757	1.000000

```
In [32]: # Visualize the correlation matrix using a heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix Heatmap')
plt.show()
```



In []: