Performance Assessment:Exploratory Data Analysis (OEM2)

by Bader Ale

For this Performance Assessment, I will be using the medical data contained in the D207 Definitions and Datafile directory.

Part 1: Research Question and Variables

The research question for this analysis is: **Is there a significant difference in readmission based on the number of office visits?**

The first thing we have to do is import the original CSV file that contains our data. To do this, we must first import the necessary packages.

In [ ]:

```
# Importing Libraries
import pandas as pd
from scipy import stats
from scipy.stats import ttest_1samp
import seaborn as sns
import matplotlib.pyplot as plt
from IPython.core.interactiveshell import InteractiveShell # Importing so we can run
multiple lines in one cell
InteractiveShell.ast_node_interactivity = "all" # Code so multiple lines in one cell
can be ran simultaenously

import warnings
warnings.filterwarnings('ignore')
```

In [ ]:

```
# Reading in the original CSV file
df = pd.read_csv(r'F:\GitHub Repos\WGU_MSDA\D207_Exploratory Data
Analysis\medical_clean.csv', index_col = 'ReAdmis') # Using ReAdmis as index
```

After importing our CSV file, we will see the first 5 records of our dataframe and see the overall shape/size.

In [ ]:

```
# Returning first 5 records of dataframe
df.head(5)
```

Out[ ]:

	CaseOrder	Customer_id	Interaction	UID	City	State	County	Zip	
ReAdmis									
No	1	C412403	8cd49b13-f45a-4b47-a2bd-173ffa932c2f	3a83ddb66e2ae73798bdf1d705dc0932	Eva	AL	Morgan	35621	34
No	2	Z919181	d2450b70-0337-4406-bdbb-bc1037f1734c	176354c5eef714957d486009feabf195	Marianna	FL	Jackson	32446	3(
No	3	F995323	a2057123-abf5-4a2c-abad-8ffe33512562	e19a0fa00aeda885b8a436757e889bc9	Sioux Falls	SD	Minnehaha	57110	43
			1dec528d-						

```
                                eb34-4079-                                        New
 No    4   A879973                   adce-  cd17d7b6d152cb6f23957346d11c3f07  Richland  MN   Waseca    56072  43
                              0d7a40e82205

                                 5885f56b-
                             d6da-43a3-                                        West          King
 No    5   C544523                   8760-  d2f0425877b10ed6bb381f3e2579424a  Point     VA   William   23181  37
                              83583af94266

5 rows × 49 columns
```

For our research question, the pertinent variables are *ReAdmis* and *Doc-visits*. We will now decalre the variables that we will be using.

Our first variable is ReAdmis - this variable is categorical with two levels (Yes/No).

Our second variable is Doc_visits - this variable is continuous.

In [ ]:

```
# Declaring variables used in t-test

# Readmin_yes dataframe: ReAdmis = No and Doc_visits
readmis_yes = df.loc[['Yes'], ['Doc_visits']]
readmis_yes


# Readmin_no dataframe: ReAdmis = Yes and Doc_visits
readmis_no = df.loc[['No'], ['Doc_visits']]
readmis_no
```

Out[ ]:

```
         Doc_visits
ReAdmis
    Yes           5
    Yes           4
    Yes           4
    Yes           6
    Yes           6
    ...         ...
    Yes           3
    Yes           5
    Yes           4
    Yes           5
    Yes           5

3669 rows × 1 columns
```

Out[ ]:

```
         Doc_visits
ReAdmis
     No           6
```

No	4
No	4
No	4
No	5
...	...
No	6
No	6
No	4
No	4
No	4

6331 rows × 1 columns

Two Sample t-Test

Since we have two variables (one categorical and one integer), we will use a two-sample t-test.

In [ ]:

```
# Performing two-sample t-test
stats.ttest_ind(readmis_yes, readmis_no, equal_var = False)
```

Out[ ]:

```
Ttest_indResult(statistic=array([0.02452273]), pvalue=array([0.9804363]))
```

We can see that our p-value is 98.04 and the t-statistic is 0.0245 - what these values say is that there is NO statistical significance between the number of visits from a doctor during a patient's admission and that patient's readmission.

In other words, the doctor visits did not affect in any statistical significant way readmission rates.

Univariate Analysis

We will be performing univariate analysis on two variables, 2 continuous and 2 categorical 1) Continuous Variables - Income and VitD_levels 2) Categorical Variables - Gender and Area

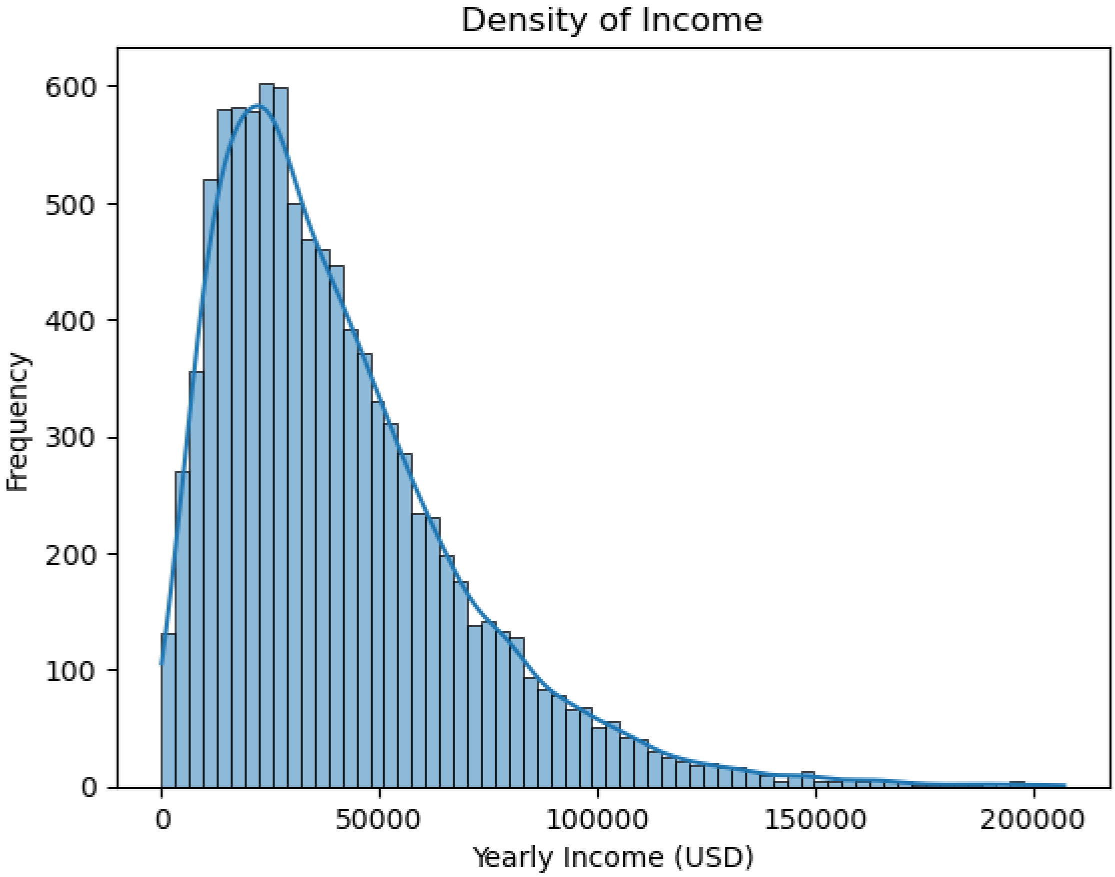
We will be using density plots to visualize the distribution.

In [ ]:

```
# Creating new series from original dataframe with only Income and VitD_level variables
#a = df['Income'].reset_index(drop=True)
#b = df['VitD_levels'].reset_index(drop=True)
#c = df['Gender'].reset_index(drop=True)
d = df['Area'].reset_index(drop=True)
```

In [ ]:

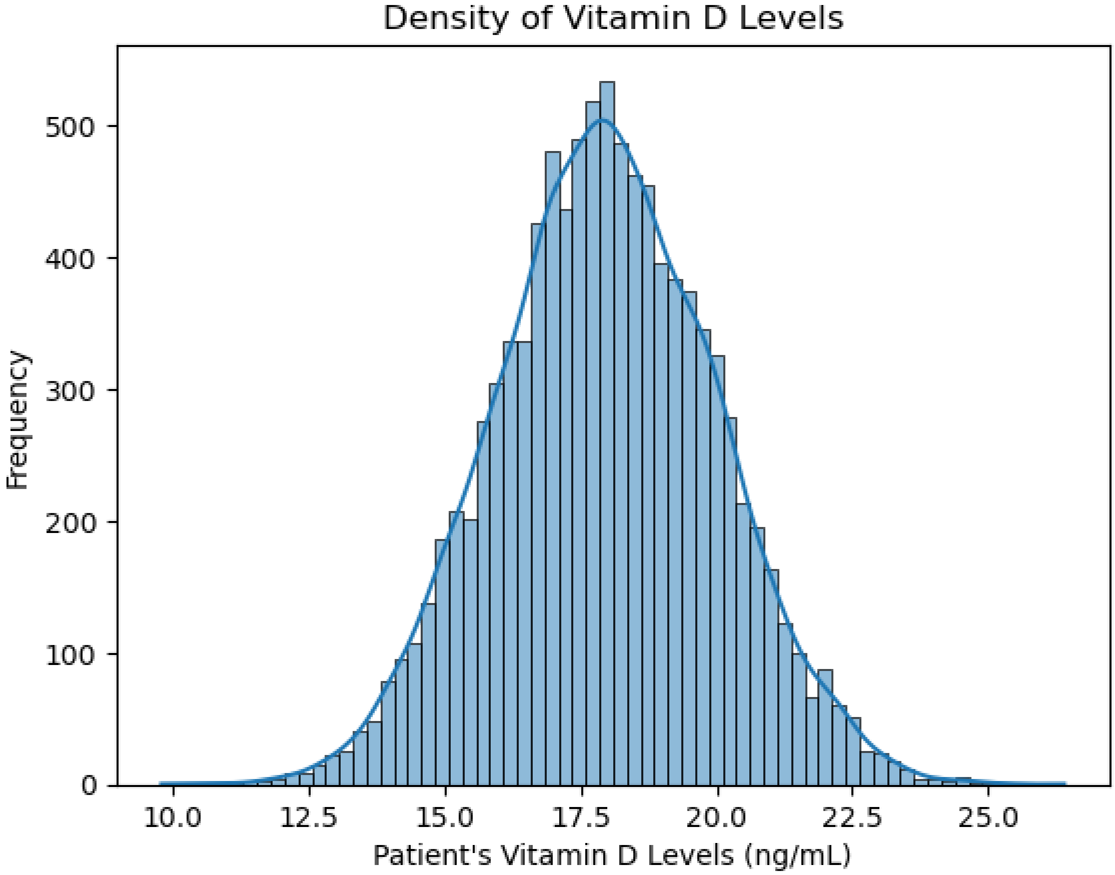
```
# Using Seaborn to plot distplot of Income
sns.histplot(data = df['Income'], kde=True)
plt.title('Density of Income')
plt.xlabel('Yearly Income (USD)')
plt.ylabel('Frequency');
```



The above figure shows that the Income variable displays a positively skewed (right-skewed) distribution

In [ ]:

```
# Using Seaborn to plot distplot of Vitamin D Levels
sns.histplot(data = df['VitD_levels'], kde=True)
plt.title('Density of Vitamin D Levels')
plt.xlabel("Patient's Vitamin D Levels (ng/mL)")
plt.ylabel('Frequency');
```

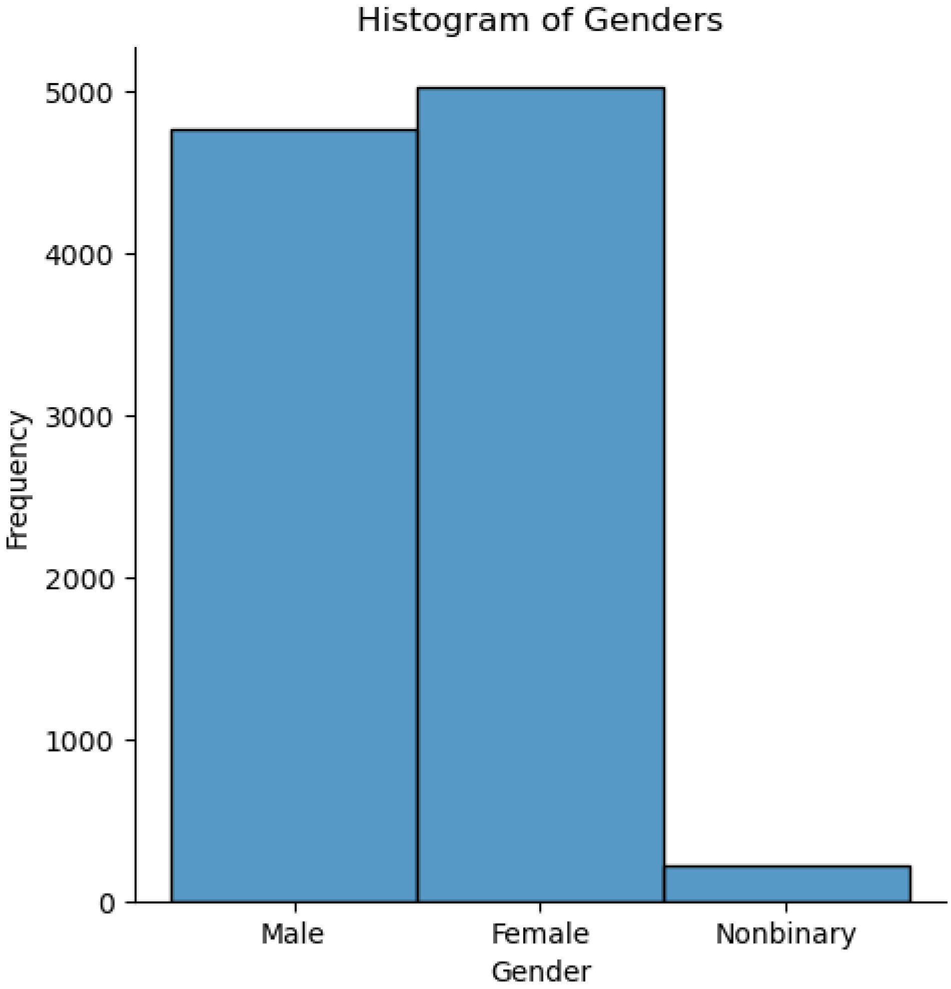


The above figure shows that the Vitamin D level variable displays a normal distribution curve

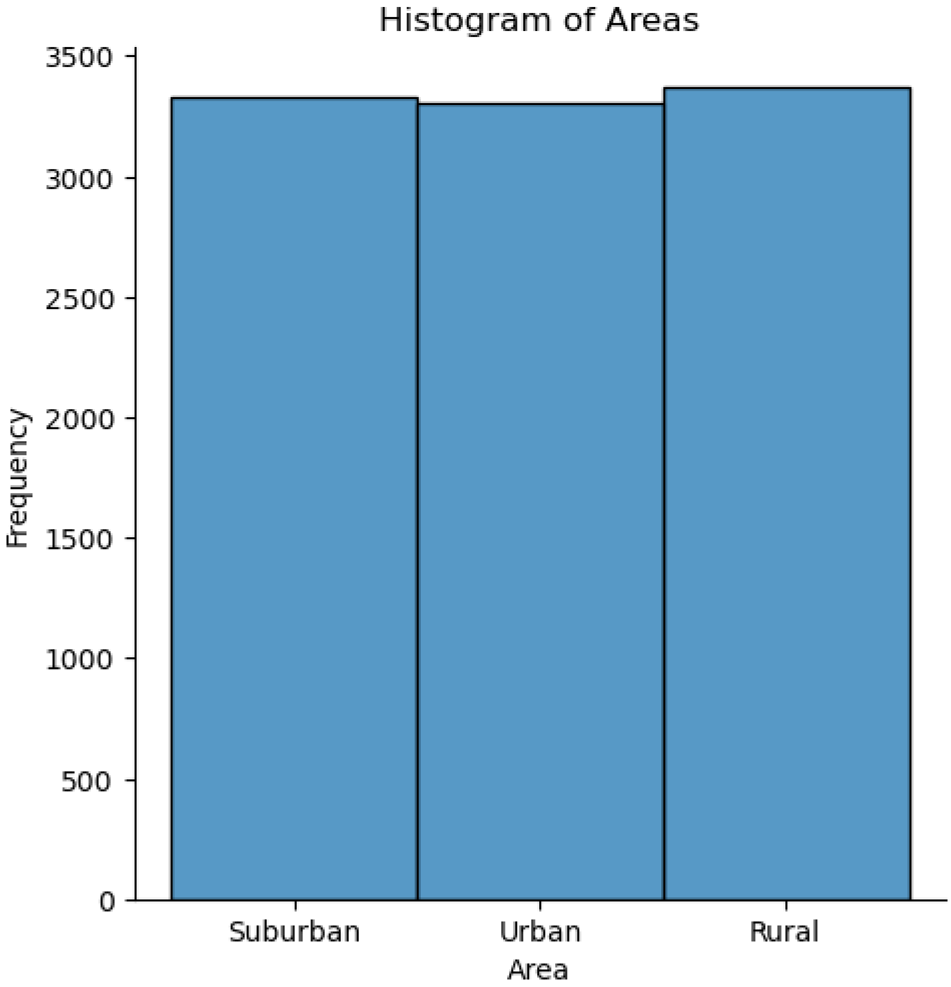
In [ ]:
```
df['Gender'].value_counts()
```

Out[ ]:
```
Female       5018
Male         4768
Nonbinary     214
Name: Gender, dtype: int64
```

In [ ]:
```
# Using Seaborn to plot histogram of Gender
sns.displot(df['Gender'], kind='hist')
plt.title('Histogram of Genders')
plt.xlabel('Gender')
plt.ylabel('Frequency');
```



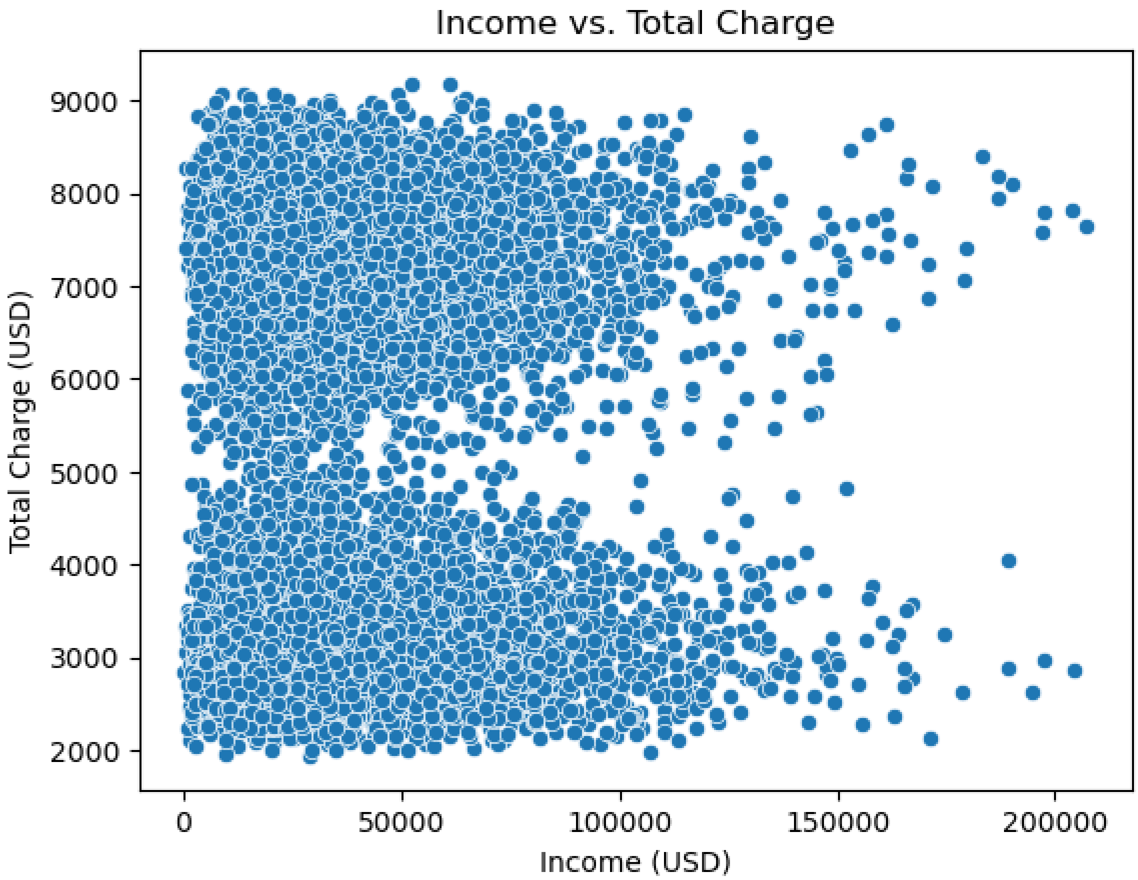
In [ ]:
```
# Using Seaborn to plot histogram of Area
sns.displot(df['Area'], kind='hist')
plt.title('Histogram of Areas')
plt.xlabel('Area')
plt.ylabel('Frequency');
```



Bivariate Analysis

We will now be performing bivariate analysis on two continuous variables - in this case, Income and Total-Charge was selected

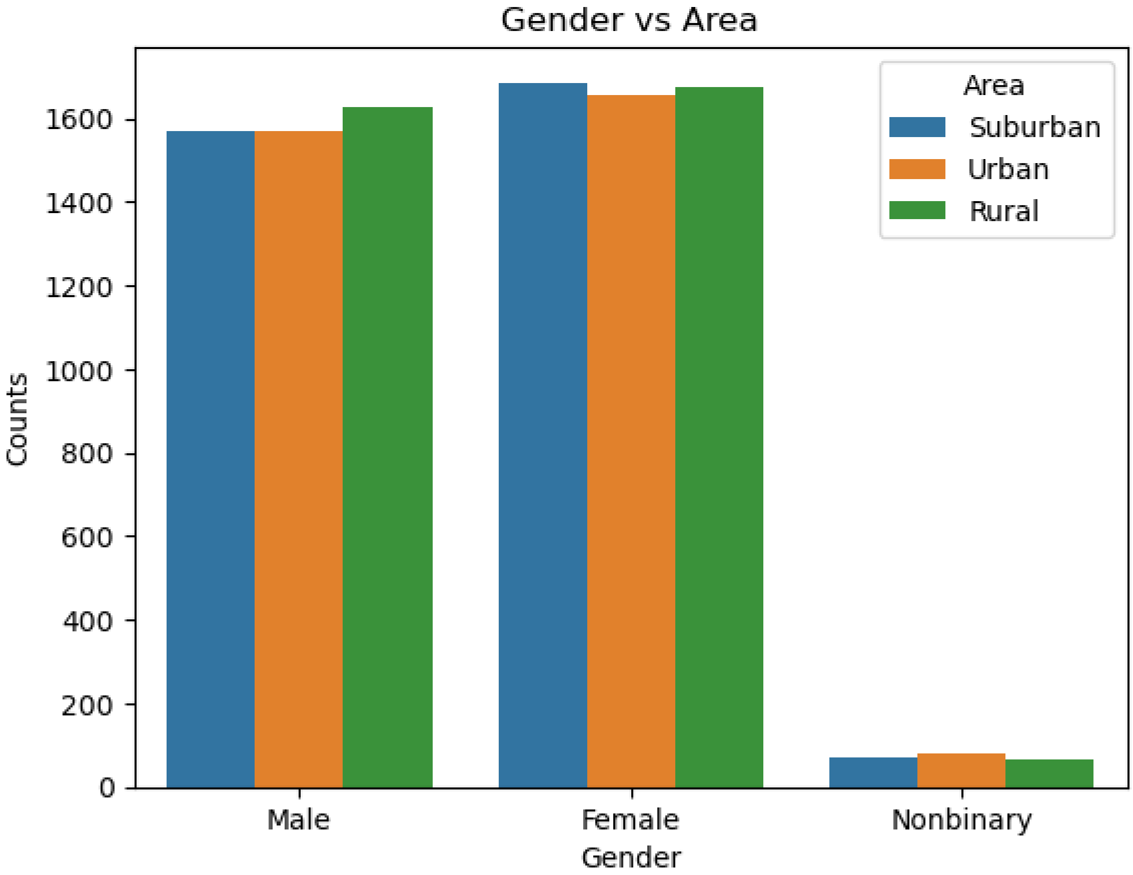
In [ ]:
```
# Using Searbon scatterplot for continuous variables
sns.scatterplot(data=df, x='Income', y='TotalCharge')
plt.title('Income vs. Total Charge')
plt.xlabel('Income (USD)')
plt.ylabel('Total Charge (USD)');
```



The same will be applied to categorical variables - in this case, Gender and Area were selected.

In [ ]:

```
# Plotting Gender vs Area
sns.countplot(data=df, x='Gender', hue='Area')
plt.title('Gender vs Area')
plt.xlabel('Gender')
plt.ylabel('Counts');
```


Gender vs Area
Counts
0
200
400
600
800
1000
1200
1400
1600
Male
Female
Nonbinary
Gender
Area
Suburban
Urban
Rural