D207 Data Exploration: Performance Assessment

Arjun Gupta

Student ID: 012296064

MSDA, Western Governors University

A. Question

1. Since I used the same data set from the previous class, I will also use a similar question. The question that I will be asking in this PA is, "Is there a relationship between Diabetes and consuming over three or more sodas a day?".

2. This is an important question to ask due to the magnitude of people affected by diabetes. This would allow doctors to find preventative measures to reduce the risk of diabetes and educate patients on the dangers of consuming that much soda a day. I want to see the relationship between these two variables and see if they are that similar to each other or not.

3. I will be using the medical_clean.csv file provided to me for the PA and within this data set, I will be using the Diabetes and the Soft_drink variables to answer this question.

B. Describe the data

1. I will be using the chi-square technique.

In [18]:
```
#import the libraries
import numpy as np
import pandas as pd
from scipy import stats
from scipy.stats import chi2_contingency
import plotnine as p9
```

In [2]:
```
#Importing the medical data file
df = pd.read_csv(r"C:\Users\arjun\OneDrive\Desktop\WGU\D207\medical_clean.csv")
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 50 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   CaseOrder           10000 non-null  int64  
 1   Customer_id         10000 non-null  object 
 2   Interaction         10000 non-null  object 
 3   UID                 10000 non-null  object 
 4   City                10000 non-null  object 
 5   State               10000 non-null  object 
 6   County              10000 non-null  object 
 7   Zip                 10000 non-null  int64  
 8   Lat                 10000 non-null  float64
 9   Lng                 10000 non-null  float64
 10  Population          10000 non-null  int64  
 11  Area                10000 non-null  object 
 12  TimeZone            10000 non-null  object 
 13  Job                 10000 non-null  object 
 14  Children            10000 non-null  int64  
 15  Age                 10000 non-null  int64  
 16  Income              10000 non-null  float64
 17  Marital             10000 non-null  object 
 18  Gender              10000 non-null  object 
 19  ReAdmis             10000 non-null  object 
 20  VitD_levels         10000 non-null  float64
 21  Doc_visits          10000 non-null  int64  
 22  Full_meals_eaten    10000 non-null  int64  
 23  vitD_supp           10000 non-null  int64  
 24  Soft_drink          10000 non-null  object 
 25  Initial_admin       10000 non-null  object 
 26  HighBlood           10000 non-null  object 
 27  Stroke              10000 non-null  object 
 28  Complication_risk   10000 non-null  object 
 29  Overweight          10000 non-null  object 
 30  Arthritis           10000 non-null  object 
 31  Diabetes            10000 non-null  object 
 32  Hyperlipidemia      10000 non-null  object 
 33  BackPain            10000 non-null  object 
 34  Anxiety             10000 non-null  object 
 35  Allergic_rhinitis   10000 non-null  object 
 36  Reflux_esophagitis  10000 non-null  object 
 37  Asthma              10000 non-null  object 
 38  Services            10000 non-null  object 
 39  Initial_days        10000 non-null  float64
 40  TotalCharge         10000 non-null  float64
 41  Additional_charges  10000 non-null  float64
 42  Item1               10000 non-null  int64  
 43  Item2               10000 non-null  int64  
 44  Item3               10000 non-null  int64  
 45  Item4               10000 non-null  int64  
 46  Item5               10000 non-null  int64  
 47  Item6               10000 non-null  int64  
 48  Item7               10000 non-null  int64  
 49  Item8               10000 non-null  int64  
```

```
dtypes: float64(7), int64(16), object(27)
memory usage: 3.8+ MB
```

In [3]:
```
#creating the contingency table
cont_tbl = pd.crosstab(df['Diabetes'], df['Soft_drink'])
print(cont_tbl)
```

```
Soft_drink    No   Yes
Diabetes
No          5425  1837
Yes         2000   738
```

In [4]:
```
#Perform the chi-squared test (values are in order- chi-squared statistic, p-value,
chi_stat, p_val, deg_free, expect = chi2_contingency(cont_tbl)
```

In [5]:
```
#printing the results of the Chi-squared test
print("Chi-squared test statistic:", chi_stat)
print("p-value:", p_val)
print("Degrees of freedom:", deg_free)
print("Expected frequencies: \n", expect)
```

```
Chi-squared test statistic: 2.7724736974606583
p-value: 0.09589785708737551
Degrees of freedom: 1
Expected frequencies:
 [[5392.035 1869.965]
 [2032.965  705.035]]
```

2. The results of the analysis performed are:

- The chi-squared test statistic: 2.7724736974606583
- p-value: 0.09589785708737551
- Degrees of freedom: 1
- Expected frequencies:
  - [[5392.035 1869.965]
    [2032.965 705.035]]

3. I chose this technique because the chi-squared test of independence is used to find the relationship between two categorical variables (Soft_drinks and Diabetes).

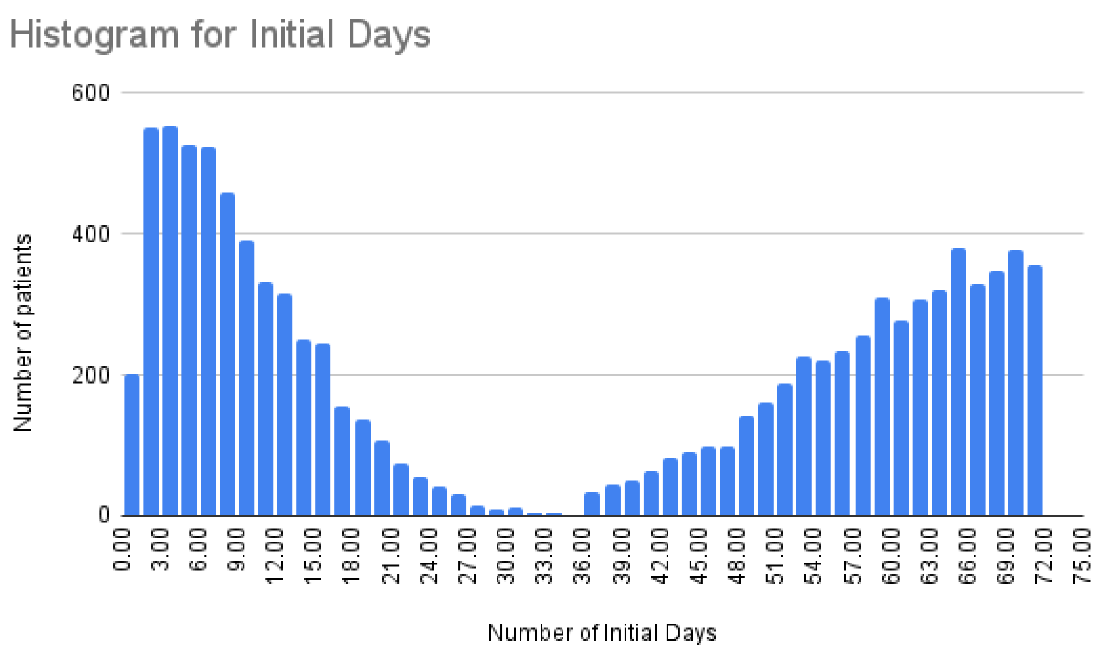
C. Distribution of 2 continuous and 2 categorical variables (Univariate)

The variables that I will be using are-

- Continuous
  - Initial_days. This variable will be represented using a Histogram.

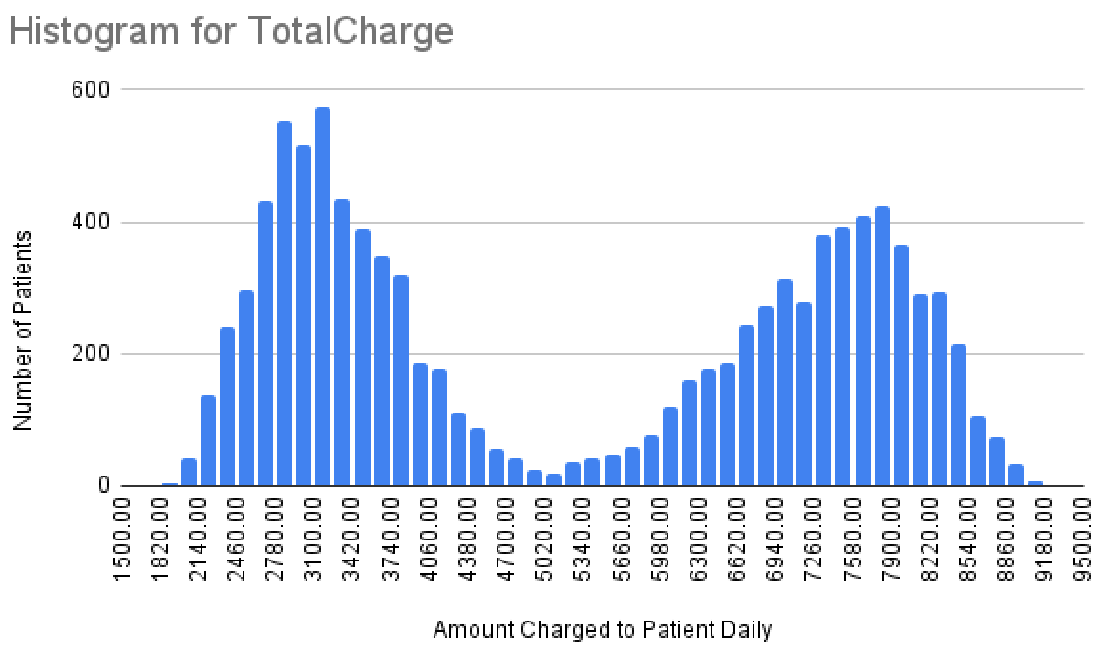
    - TotalCharge. This variable will be represented using a Histogram.
- Categorical
    - Initial_admin. This variable will be represented using a bar chart.
    - Area. This variable will be represented using a bar chart

1. I will be representing Initial_days and TotalCharge on a histogram using Excel.



In [10]:
```
#Initial days univariate statistics
df.Initial_days.describe()
```

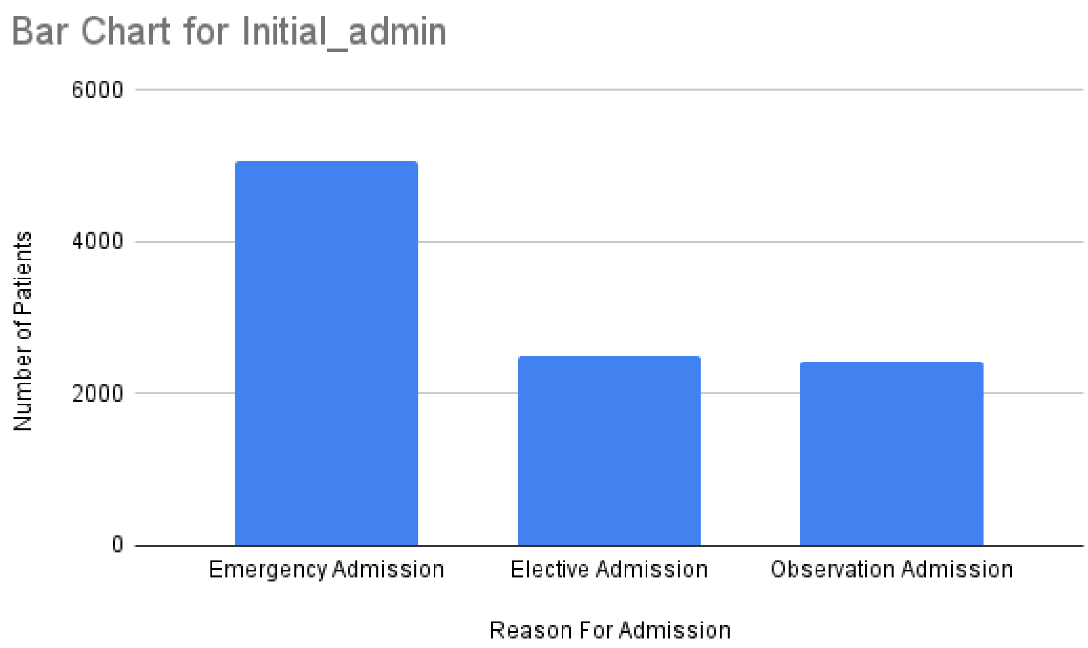
Out[10]:
```
count    10000.000000
mean        34.455299
std         26.309341
min          1.001981
25%          7.896215
50%         35.836244
75%         61.161020
max         71.981490
Name: Initial_days, dtype: float64
```



In [11]:
```
#TotalCharge univariate statistics
df.TotalCharge.describe()
```

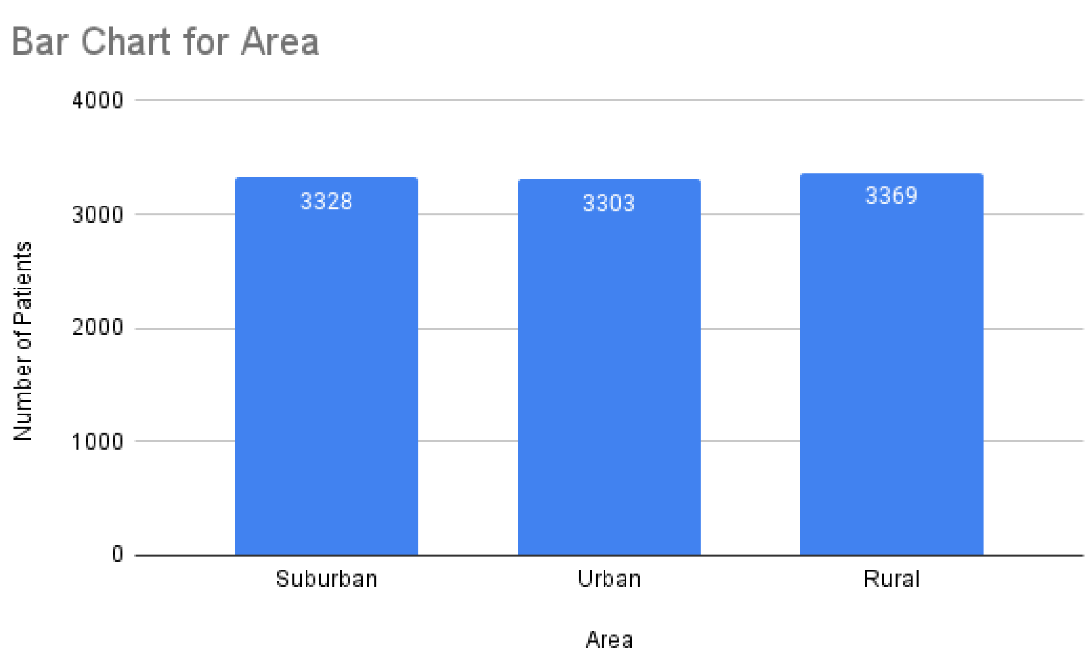
Out[11]:
```
count    10000.000000
mean      5312.172769
std       2180.393838
min       1938.312067
25%       3179.374015
50%       5213.952000
75%       7459.699750
max       9180.728000
Name: TotalCharge, dtype: float64
```

I will be representing Initial_admin and Area on a Bar Chart using Excel.



In [14]:
```
#Initial_admin univariate statistics
df.Initial_admin.value_counts()
```

Out[14]:
```
Initial_admin
Emergency Admission      5060
Elective Admission       2504
Observation Admission    2436
Name: count, dtype: int64
```



In [15]:
```
#Area univariate statistics
df.Area.value_counts()
```

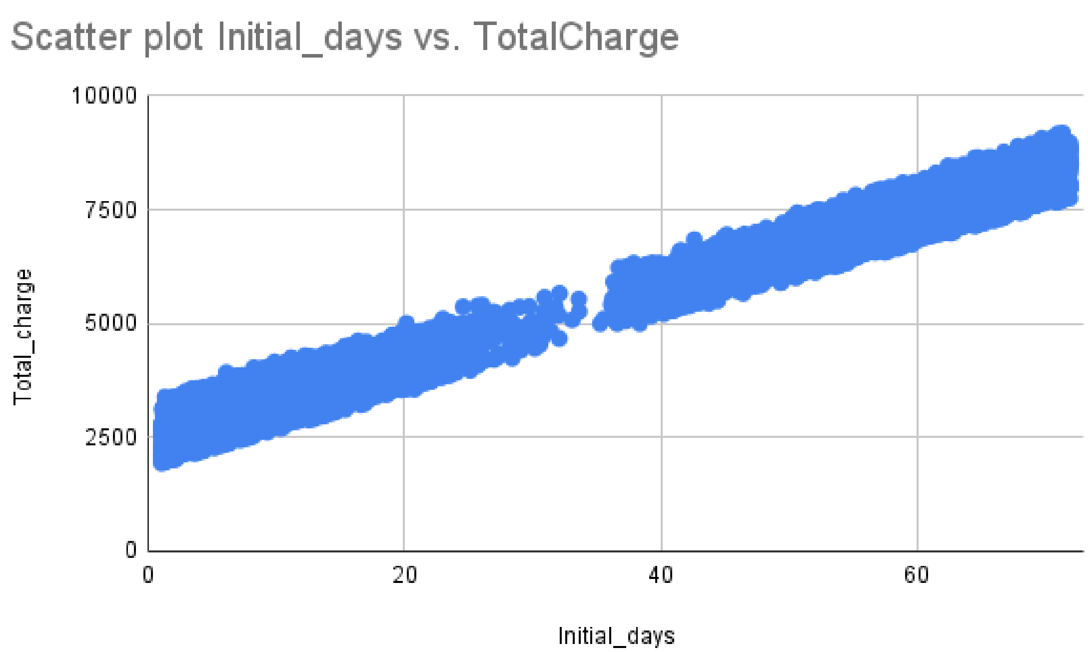
Out[15]:
```
Area
Rural       3369
Suburban    3328
Urban       3303
Name: count, dtype: int64
```

D. Distribution of 2 continuous and 2 categorical variables (Bivariate)

The variables that I will be using are-

- Continuous
  - Initial_days.
  - TotalCharge. These variables will be represented using a scatter plot.
- Categorical
  - Initial_admin.
  - Area. These variables will be represented using a stacked bar chart

1. I will be representing Initial_days and TotalCharge on a Scatter Plot using Excel.

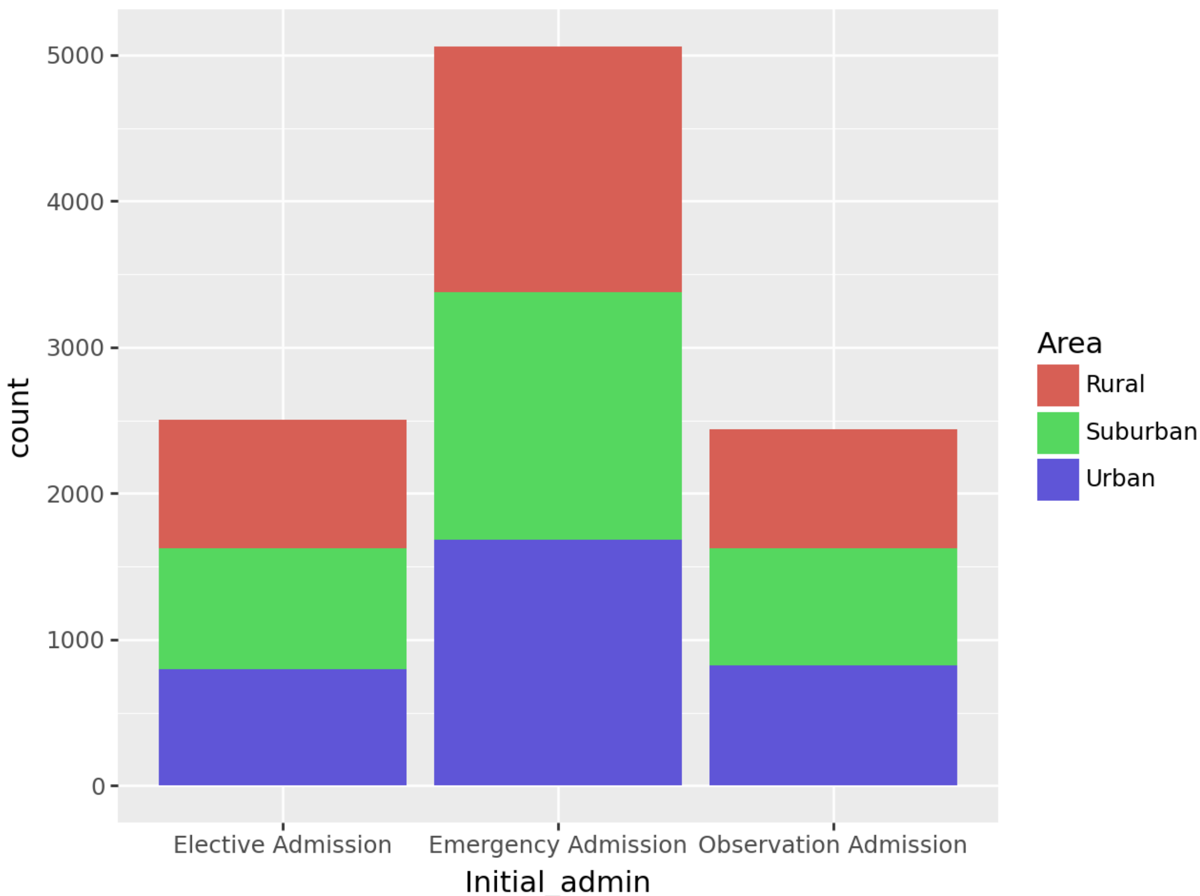


In [19]:
```python
#linear regression on Initial_days vs TotalCharge
slope, intercept, r_value, p_value, std_err = stats.linregress(df['Initial_days'],
print(f'Slope: {slope}')
print(f'Intercept: {intercept}')
print(f'R-squared: {r_value**2}')
print(f'P-value: {p_value}')
print(f'Standard error: {std_err}')
```

```
Slope: 81.85095641319026
Intercept: 2491.973570329295
R-squared: 0.97543329411556
P-value: 0.0
Standard error: 0.12990978615408147
```

I will be representing Initial_admin and Area on a Bar Chart using plotnine.

In [7]:
```
# code for bar chart for Initial_admin and Area
(p9.ggplot(df)+p9.aes('Initial_admin', fill ='Area') + p9.geom_bar())
```



In [43]:
```
#Creating a contingency table for Bivariate Statistics
contingency_table = pd.crosstab(df['Initial_admin'], df['Area'])
print("Contingency Table:\n", contingency_table)
print('')
#Chi-squared statistics
chi2, p, dof, expected = chi2_contingency(contingency_table)
print(f"Chi-square statistic: {chi2}")
print(f"P-value: {p}")
print(f"Degrees of freedom: {dof}")
print("Expected frequencies:\n", expected)
```

```
Contingency Table:
 Area                  Rural  Suburban  Urban
Initial_admin
Elective Admission       877       830    797
Emergency Admission     1682      1692   1686
Observation Admission    810       806    820

Chi-square statistic: 3.35997005945306
P-value: 0.4994869431163934
Degrees of freedom: 4
Expected frequencies:
 [[ 843.5976  833.3312  827.0712]
 [1704.714  1683.968  1671.318 ]
 [ 820.6884  810.7008  804.6108]]
```

E. Implications of the analysis

1. Hypothesis test

- H_0 = Diabetes and Soft_drinks are independent on each other
- H_1 = Diabetes and Soft_drinks are dependent on each other

Since I am using the chi-squared test for independence, I will use $\alpha = 0.05$. This means that if the p-value is lower than this alpha variable, it would mean that there is a statistically significant difference between the relationship between Diabetes and Soft_drinks. I calculated a p-value of 0.0959, higher than the alpha value. This means that we fail to reject the null hypothesis. There is no relationship between Diabetes and Soft_drinks, and the two variables are independent. This is useful for the hospital as now they would not tell the patients who drink a lot of sodas to avoid it to help prevent Diabetes. This also allows doctors not to spread misinformation as most people believe that soda is a leading cause of Diabetes.

2. Limitations

One limitation that should be taken into account is that although the data set provided to me was already cleaned, I did not look into it myself and cleaned it how I deemed to be appropriate, so there may be some duplicates, outliers, or even other issues that have not been addressed in the data set provided to me.

This is also the only method for checking the relationship between categorical variables, so it may not be the best model. Still, regarding my knowledge of the material and the instructions on the PA mention, the chi-squared test is the best method for finding the answer to my question initially stated at the start.

There may also be other hidden/ latent variables which may be influencing the results which are not provided to us such as amount of sugar or the size of each soda being drank.

3. Recommendations

It can be concluded that there is no relationship between Diabetes and consuming 3 or more soft drinks a day.

My recommendation to the hospital is to correctly educate their patients on the matter. Make sure you mention that there is no relationship found between Diabetes and consuming over 3 sodas a day. I would also reeducate the doctors on the relationship between the two variables and how they can teach their patients the truth found in the data.

I also recommend running the test using the data through another test for categorical data to see if there are any discrepancies between the two tests.

G. Web references

1. NovoStats. “Chi Squared Test in Python.” YouTube, YouTube, 19 Dec. 2021, www.youtube.com/watch?v=VqopW3zfguA&ab_channel=NovoStats.
2. Kibirige, Hassan. “A Grammar of Graphics for Python.” Plotnine 0.13.6, MIT, plotnine.org/. Accessed 1 Aug. 2024.

H. No sources used