D207 Performance Assessment

Alfred Conrad Santos

Western Governors University

Exploratory Data Analysis - D207

Kesselly Kamara

22 APR 2024

A1: Research Question

I will be utilizing the same dataset employed in my previous research for D206, although my focus will now be on a distinct research question. Specifically, I aim to explore the correlation between patients diagnosed with hypertension (high blood pressure) and their likelihood of hospital readmission. This topic holds personal significance due to a recurring health concern within my family's medical history. My revised research question is as follows: 'Do patients with hypertension exhibit a higher rate of hospital readmission compared to individuals without hypertension?' This inquiry will be investigated in relation to the null hypothesis.

$$H_0: readmission_{hypertension} = readmission_{population}$$

$$H_1: readmission_{hypertension} \neq readmission_{population}$$

A2: Explain how stakeholders in the organization could benefit from an analysis of the data.

The state of one's well-being is often closely intertwined with their level of mobility, and this correlation likely extends to their capacity to recuperate and bounce back following a hospital stay. Hypertension significantly elevates the risk of severe and potentially life-threatening conditions such as heart attacks and strokes, thereby diminishing overall wellness. This decline in well-being and recovery potential could lead to adverse outcomes following an initial hospitalization, potentially resulting in extended stays or subsequent readmissions. Identifying such a relationship could underscore the importance not only of

addressing the primary ailment responsible for the hospitalization but also of managing hypertension to enhance recovery and mitigate the likelihood of readmission. While this is evidently crucial for individuals undergoing hospitalization, it also holds significant implications for hospitals themselves. As hospitals strive to lower their readmission rates, the financial implications, including penalties for high readmission rates and decreased efficiency due to the need for repeated treatments, underscore the importance of addressing hypertension comprehensively.

```
import pandas as pd
from pandas.api.types import CategoricalDtype
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
import seaborn as sns
from scipy import stats

# Importing the dataset with the first column as an index, and Pandas
may duplicate this index creating an additional column unless
specified with 'index_col=0'
df = pd.read_csv('./medical_clean.csv', index_col=0)
# Verifying data types, quantity of entries, and the overall size of
the dataframe
df.info()
```

```
# Visually inspect dataframe to facilitate exploration, spot problems
pd.set_option("display.max_columns", None)
df.head(5)
```

```
# Convert integer column to string and front-fill with 0's to ensure
5 characters
df['Zip'] = df['Zip'].astype("str").str.zfill(5)
# Convert string column to categorical
df["Area"] = df["Area"].astype("category")
# Replace city-specific time zones with their respective regions
df.TimeZone.replace({
      # Puerto Rico remains on Atlantic Standard Time
      "America/Puerto_Rico" : "US - Puerto Rico",
      # Eastern Time Zone observes DST
      "America/New_York": "US - Eastern",
      "America/Detroit" : "US - Eastern",
      "America/Indiana/Indianapolis" : "US - Eastern",
      "America/Indiana/Vevay" : "US - Eastern",
      "America/Indiana/Vincennes" : "US - Eastern",
      "America/Kentucky/Louisville" : "US - Eastern",
      "America/Toronto" : "US - Eastern",
      "America/Indiana/Marengo" : "US - Eastern",
      "America/Indiana/Winamac" : "US - Eastern",
      # Central Time Zone observes DST
      "America/Chicago" : "US - Central",
      "America/Menominee" : "US - Central",
      "America/Indiana/Knox" : "US - Central",
      "America/Indiana/Tell_City" : "US - Central",
      "America/North_Dakota/Beulah" : "US - Central",
      "America/North_Dakota/New_Salem" : "US - Central",
      # Mountain Time Zone observes DST
      "America/Denver" : "US - Mountain",
      "America/Boise" : "US - Mountain",
      # Arizona stays on Mountain Standard Time
      "America/Phoenix" : "US - Arizona",
      # Pacific Time Zone observes DST
      "America/Los_Angeles" : "US - Pacific",
      # Alaskan Time Zone observes DST
      "America/Nome" : "US - Alaskan",
      "America/Anchorage" : "US - Alaskan",
      "America/Sitka" : "US - Alaskan",
      "America/Yakutat" : "US - Alaskan",
      # Aleutian Time Zone observes DST
```

```
        "America/Adak" : "US - Aleutian",
        # Hawaii remains on Hawaii Standard Time
        "Pacific/Honolulu" : "US - Hawaiian"
        }, inplace=True)
# Convert string column to categorical
df["TimeZone"] = df["TimeZone"].astype("category")
# Convert string column to categorical
df["Marital"] = df["Marital"].astype("category")
# Convert string column to categorical
df["Gender"] = df["Gender"].astype("category")
# Map "Yes" and "No" to True and False for boolean conversion
bool_mapping = {"Yes" : True, "No" : False}
# Convert string column to boolean
df["ReAdmis"] = df["ReAdmis"].map(bool_mapping)
# Convert string column to boolean
df["Soft_drink"] = df["Soft_drink"].map(bool_mapping)
# Convert string column to categorical
df["Initial_admin"] = df["Initial_admin"].astype("category")
# Convert string column to boolean
df["HighBlood"] = df["HighBlood"].map(bool_mapping)
# Convert string column to boolean
df["Stroke"] = df["Stroke"].map(bool_mapping)
# Convert string column to categorical
df["Complication_risk"] = df["Complication_risk"].astype("category")
# Convert string column to boolean
df["Overweight"] = df["Overweight"].map(bool_mapping)
# Convert string column to boolean
df["Arthritis"] = df["Arthritis"].map(bool_mapping)
# Convert string column to boolean
df["Diabetes"] = df["Diabetes"].map(bool_mapping)
# Convert string column to boolean
df["Hyperlipidemia"] = df["Hyperlipidemia"].map(bool_mapping)
# Convert string column to boolean
df["BackPain"] = df["BackPain"].map(bool_mapping)
# Convert string column to boolean
df["Anxiety"] = df["Anxiety"].map(bool_mapping)
# Convert string column to boolean
df["Allergic_rhinitis"] = df["Allergic_rhinitis"].map(bool_mapping)
# Convert string column to boolean
```

```
df["Reflux_esophagitis"] = df["Reflux_esophagitis"].map(bool_mapping)
# Convert string column to boolean
df["Asthma"] = df["Asthma"].map(bool_mapping)
# Convert string column to categorical
df["Services"] = df["Services"].astype("category")
# Establish ordered categorical datatype structure for survey
response columns
survey_scores = CategoricalDtype(categories=["8", "7", "6", "5", "4",
"3", "2", "1"], ordered=True)
# Map integers to strings for categorical conversion
df["Item1"] = df["Item1"].map(str)
df["Item2"] = df["Item2"].map(str)
df["Item3"] = df["Item3"].map(str)
df["Item4"] = df["Item4"].map(str)
df["Item5"] = df["Item5"].map(str)
df["Item6"] = df["Item6"].map(str)
df["Item7"] = df["Item7"].map(str)
df["Item8"] = df["Item8"].map(str)
# Convert string columns to ordered categorical
df["Item1"] = df["Item1"].astype(survey_scores)
df["Item2"] = df["Item2"].astype(survey_scores)
df["Item3"] = df["Item3"].astype(survey_scores)
df["Item4"] = df["Item4"].astype(survey_scores)
df["Item5"] = df["Item5"].astype(survey_scores)
df["Item6"] = df["Item6"].astype(survey_scores)
df["Item7"] = df["Item7"].astype(survey_scores)
df["Item8"] = df["Item8"].astype(survey_scores)
# Rename 'HighBlood' column to 'Hypertension'
df = df.rename(columns={'HighBlood': 'Hypertension'})
```

A3: Identify *all* of the data in your data set that are relevant to answering your question in part A1

The pertinent variables for the research inquiry include:

1. **ReAdmis (Categorical)**: For instance, in row 1, it's represented as "True". This binary variable denotes whether the patient was readmitted within one month post-discharge.
2. **Hypertension (Categorical)**: For example, in row 1, it's recorded as "True". This binary variable indicates whether the patient suffers from hypertension, commonly known as high blood pressure.

B1: Using one of the following techniques, write code (in either Python or R) to run the analysis of the data set

Multiple chi-square test variants are available for analyzing this dataset, with the chi-square test for independence being one of the most relevant options for assessing proportions across two distinct groups (as outlined in WGU Courseware Resources). According to ScyPi documentation, the chi2_contingency function stands out as a tool for conducting the Chi-square test for independence, a method aimed at exploring associations between categorical variables in contingency tables.

```
# Generate a table with ratios of Hypertension/no Hypertension
patients vs ReAdmits/non-ReAdmits patients utiliizing crosstab
table = pd.crosstab(df.ReAdmis, df.Hypertension)
print(table)
```

```
Hypertension  False  True
ReAdmis
False          3747   2584
True           2163   1506
```

```
# Calculate the chi-square test statistic and associated p-value for independence
chi_result = stats.chi2_contingency(table)

# Print the chi-square test result
print(chi_result)

# Print the p-value, which represents the probability under the null hypothesis of obtaining an observation as extreme as the one observed
print(f"The calculated p-value, representing the probability under the null hypothesis of obtaining an observation as extreme as the one observed, is {chi_result[1]:.3}.")
```

```
Chi2ContingencyResult(statistic=0.04239657973011679,
pvalue=0.8368656684578771, dof=1, expected_freq=array([[3741.621,
2589.379],
       [2168.379, 1500.621]]))
The calculated p-value, representing the probability under the null hypothesis of obtaining an observation as extreme as the one observed, is 0.837.
```

B2: Provide the output and the results of *any* calculations from the analysis you performed.

The chi-square test was conducted to explore the relationship between hypertension and readmitted patients, yielding a p-value of approximately 0.837. This value indicates the probability of observing the data or more extreme results if there were truly no association between hypertension and readmission. With an alpha level of 0.05, commonly used for significance testing, the obtained p-value exceeds this threshold.

Consequently, failing to reject the null hypothesis suggests that there is no statistically significant relationship between hypertension and readmission. This means that, based on this analysis, we cannot conclude that hypertension has a meaningful impact on the likelihood of patient readmission.

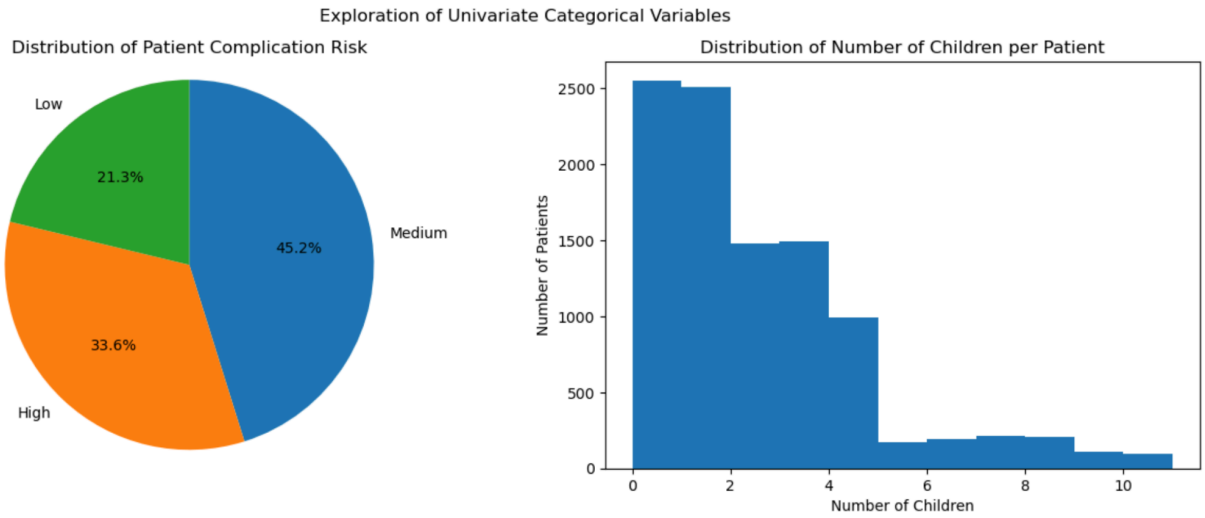
B3: Justify why you chose this analysis technique.

The decision to employ the chi-square test for independence stemmed from the unique characteristics of the dataset under scrutiny and the specific inquiry at hand. The primary objective was to discern whether there exists a statistically significant disparity in the proportion of patients readmitted with hypertension compared to those without hypertension. Utilizing a chi-square test was deemed fitting due to its applicability in assessing the independence of two proportions, aligning with the analytical requirements of this investigation. While alternative statistical methods could have been considered for similar analyses, the chi-square test was selected due to the discrete and categorical nature of the dataset, where precise measurement was unattainable.

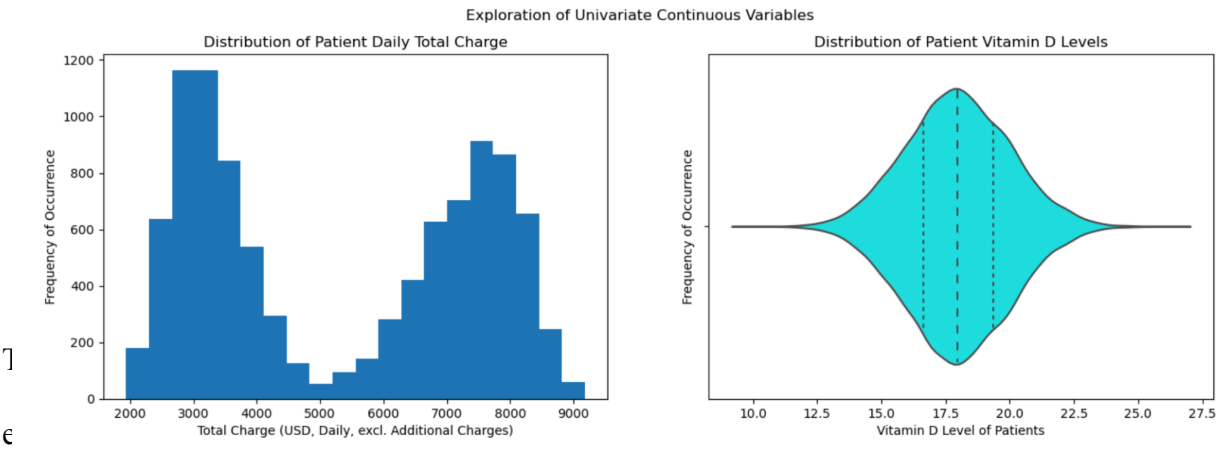
C1: Identify the distribution of two continuous variables and two categorical variables using univariate statistics from your cleaned and prepared data. Represent your findings in Part C, visually as part of your submission.

```
import matplotlib.pyplot as plt  # Importing the matplotlib library
for plotting
import numpy as np  # Importing numpy library for numerical
operations
# Create a figure with a size of 16x5
plt.figure(figsize=[16, 5])
# Title for the entire figure
```

```
plt.suptitle("Exploration of Univariate Categorical Variables")
# Plot on the left: Distribution of Patient Complication Risk
plt.subplot(1, 2, 1)  # Creating a subplot, 1 row, 2 columns, and
plot 1
plt.title("Distribution of Patient Complication Risk")  # Title for
the subplot
risk_counts = df["Complication_risk"].value_counts()  # Counting the
occurrences of each risk level
risk_labels = ["Medium", "High", "Low"]  # Labels for the risk levels
plt.pie(risk_counts, labels=risk_labels, autopct='%1.1f%%',
startangle=90, counterclock=False)  # Creating a pie chart
plt.axis('square')  # Ensuring the pie chart is circular
# Plot on the right: Distribution of the Number of Children per
Patient
plt.subplot(1, 2, 2)  # Creating a subplot, 1 row, 2 columns, and
plot 2
plt.title("Distribution of Number of Children per Patient")  # Title
for the subplot
bins = np.arange(0, df["Children"].max() + 2, 1)  # Defining the bins
for histogram
plt.hist(data=df, x="Children", bins=bins)  # Creating a histogram
for the number of children
plt.xlabel("Number of Children")  # Label for x-axis
plt.ylabel("Number of Patients")  # Label for y-axis
plt.show()  # Displaying the plot
```



```
import matplotlib.pyplot as plt  # Importing matplotlib for plotting
import seaborn as sns  # Importing seaborn for advanced visualization
# Create a figure with a size of 16x5
plt.figure(figsize=[16, 5])
# Title for the entire figure
plt.suptitle("Exploration of Univariate Continuous Variables")
# Plot on the left: Distribution of Total Charge
plt.subplot(1, 2, 1)  # Creating a subplot, 1 row, 2 columns, and
plot 1
plt.title('Distribution of Patient Daily Total Charge')  # Title for
the subplot
plt.hist(data=df, x="TotalCharge", bins=20)  # Creating a histogram
for TotalCharge
plt.xlabel('Total Charge (USD, Daily, excl. Additional Charges)')  #
Label for x-axis
plt.ylabel("Frequency of Occurrence")  # Label for y-axis
# Plot on the right: Distribution of Vitamin D Levels
plt.subplot(1, 2, 2)  # Creating a subplot, 1 row, 2 columns, and
plot 2
plt.title("Distribution of Patient Vitamin D Levels")  # Title for
the subplot
sns.violinplot(data=df, x="VitD_levels", color="cyan",
inner="quartile")  # Creating a violin plot for VitD_levels
plt.xlabel("Vitamin D Level of Patients")  # Label for x-axis
plt.ylabel("Frequency of Occurrence")  # Label for y-axis
plt.show()  # Displaying the plots
```



- Complication_risk (categorical)
- Children (categorical)
- TotalCharge (continuous)
- VitD_level (continuous)

Each plot illustrates the distribution pattern of its respective variable, facilitating a comprehensive exploration of the dataset's characteristics and potential trends.

These are the code blocks i used to verify each respective variable:(univariate stats)

```
# Count the occurrences of different levels of complication risk in the DataFrame column "Complication_risk"
df.Complication_risk.value_counts()
```

```
Medium    4517
High      3358
Low       2125
Name: Complication_risk, dtype: int64
```

```
# Count the occurrences of different numbers of children in the DataFrame column "Children" and sort them by index
df.Children.value_counts().sort_index()
```

```
0     2548
1     2509
2     1475
3     1489
4      995
5      169
6      191
7      213
8      209
9      108
10      94
Name: Children, dtype: int64
```

```
# Generate descriptive statistics for the continuous variable
"TotalCharge" in the DataFrame
df.TotalCharge.describe()

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

# Compute descriptive statistics for the continuous variable
"VitD_levels" in the DataFrame
df.VitD_levels.describe()

count    10000.000000
mean        17.964262
std          2.017231
min          9.806483
25%         16.626439
50%         17.951122
75%         19.347963
max         26.394449
Name: VitD_levels, dtype: float64
```

One notable observation pertains to the distribution of the patient's number of children. Contrary to expectations of a gradual decline with increasing numbers of children, the data reveals four distinct plateaus: at 0 and 1 children, followed by 2 and 3 children, then 4 children, and finally, 5 to 10 children, each with comparable frequencies. While the discreteness of the data and the histogram's format likely contribute to this pattern, the absence of a smooth downward slope

beyond 2 or 3 children is surprising. Further investigation into this unexpected observation is warranted to understand its implications for the study's outcomes fully.

Another striking observation concerns the variable labeled "TotalCharge." Despite its suggestive name, this variable does not accurately reflect the total hospitalization charges. Instead, it represents the total charges to the patient, excluding any additional costs, averaged on a per-day basis throughout their stay. While one might anticipate a distribution with a long tail due to outlier hospitalizations accruing substantial costs, the actual distribution presents with two distinct peaks, which is unexpected. Consulting with domain experts could offer insights into the underlying reasons for this unexpected pattern and its potential impact on the study's conclusions.

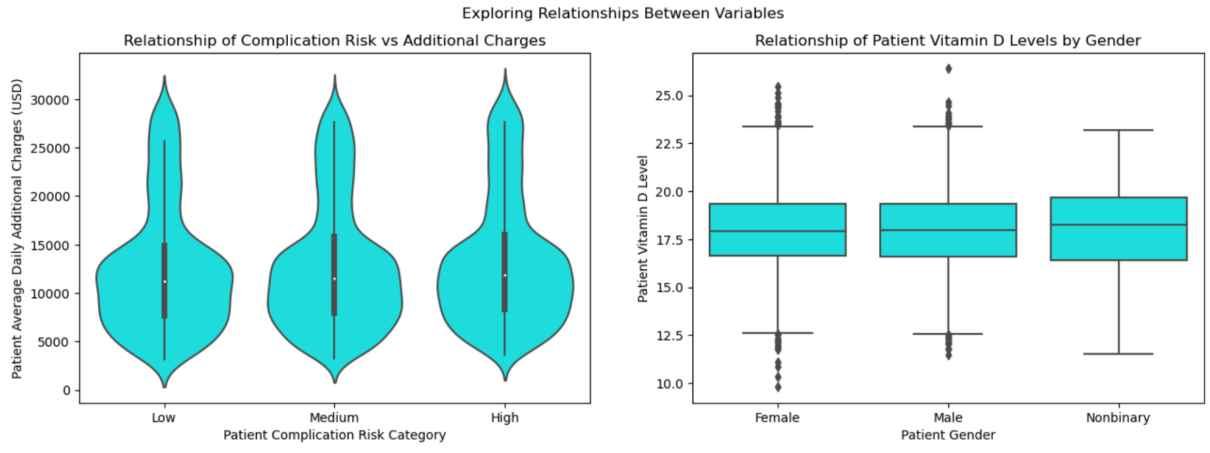
In summary, while the provided data offers valuable insights, a more comprehensive description of the distributions of vitamin D levels and complication risk, along with further exploration of the unexpected patterns observed in the patient's number of children and total charges, is necessary for a thorough understanding of the study's findings.

D1: Identify the distribution of two continuous variables and two categorical variables using bivariate statistics from your cleaned and prepared data. Represent your findings in Part D, visually as part of your submission.

```
# Create a figure with a size of 16x5 for bivariate exploration
plt.figure(figsize=[16, 5])
# Add a super title for the entire figure
plt.suptitle("Exploring Relationships Between Variables")
```

```
# Plot on the left: Bivariate exploration of Complication_risk and
Additional_charges
plt.subplot(1, 2, 1)  # Create a subplot, 1 row, 2 columns, and plot
1
plt.title('Relationship of Complication Risk vs Additional Charges')
# Title for the subplot
plot_order = ["Low", "Medium", "High"]  # Define order for x-axis
categories
sns.violinplot(data=df, x="Complication_risk",
y="Additional_charges", color="cyan", order=plot_order)  # Create
violin plot
plt.xlabel('Patient Complication Risk Category')  # Label for x-axis
plt.ylabel("Patient Average Daily Additional Charges (USD)")  # Label
for y-axis
# Plot on the right: Bivariate exploration of Gender and VitD_levels
plt.subplot(1, 2, 2)  # Create a subplot, 1 row, 2 columns, and plot
2
plt.title("Relationship of Patient Vitamin D Levels by Gender")  #
Title for the subplot
sns.boxplot(data=df, x="Gender", y="VitD_levels", color="cyan")  #
Create box plot
plt.xlabel("Patient Gender")  # Label for x-axis
plt.ylabel("Patient Vitamin D Level")  # Label for y-axis

plt.show()  # Display the plots
```



These are the code blocks i used to verify each respective variable:

```
# Generate descriptive statistics for the continuous variable
"Additional_charges" in the DataFrame
df.Additional_charges.describe()
count    10000.000000
mean     12934.528587
std       6542.601544
min       3125.703000
25%       7986.487755
50%      11573.977735
75%      15626.490000
max      30566.070000
Name: Additional_charges, dtype: float64

# Calculate correlation between VitD_levels and Additional_charges
correlation = df['VitD_levels'].corr(df['Additional_charges'])
print("Correlation between VitD_levels and Additional_charges:",
correlation)

Correlation between VitD_levels and Additional_charges:
0.008289992581332683


# Define the bins for categorizing Additional_charges
bins = [3125.703, 7986.487755, 11573.977735, 15626.490000,
30566.070000]
labels = ['Low', 'Medium', 'High', 'Very High']

from scipy.stats import chi2_contingency


# Categorize Additional_charges
df['Additional_charges_category'] = pd.cut(df['Additional_charges'],
bins=bins, labels=labels)

# Create the contingency table
contingency_table = pd.crosstab(df['Complication_risk'],
df['Additional_charges_category'])

# Perform the chi-square test
chi2, p, dof, expected = chi2_contingency(contingency_table)
```

```
# Print the contingency table
print("Contingency Table:")
print(contingency_table)
Contingency Table:
Additional_charges_category   Low  Medium  High  Very High
Complication_risk
High                           759     845   886        868
Low                            588     516   534        486
Medium                        1152    1139  1080       1146

# Print the chi-square test results with clarification
print("\nChi-Square Test Results (for association between
Complication_risk and Additional_charges_category):")
print(f"Chi-square value: {chi2}")
print(f"P-value: {p}")
Chi-Square Test Results (for association between Complication_risk and
Additional_charges_category):
Chi-square value: 24.61312982182842
P-value: 0.0004026007048223903

Mean Vitamin D levels for Female: 17.98585338779992
Mean Vitamin D levels for Male: 17.936360199517615
Mean Vitamin D levels for Nonbinary: 18.07962064182243

Bivariate Stats Interpretation:
1. Correlation between VitD_levels and Additional_charges indicates
the strength and direction of their relationship.
2. Chi-Square Test Results assess the association between
Complication_risk and Additional_charges_category.
3. Mean Vitamin D levels are compared across gender groups.
```

Overall, while there are slight differences in the mean Vitamin D levels between the gender groups, these differences are relatively small. The similarities lie in the closeness of mean levels between Female and Male, while the difference with Nonbinary individuals is somewhat higher.

In the left-hand plot depicting complication risk versus additional charges, subtle variations in the shapes of the violin plots are observable. Higher complication risks appear to exhibit a slightly greater amount of additional charges, evidenced by the slightly fuller top of the high-risk plot compared to the medium- and low-risk plots. Moreover, there is a slight elevation in the beginning and ending of the high-risk plot relative to the medium- and low-risk plots, although this distinction may not be readily discernible.

On the right-hand side, the plot illustrating patient gender versus Vitamin D levels suggests that female patients generally have a broader range of Vitamin D levels compared to male patients,

although this discrepancy is primarily notable in the outliers. The broader range observed for nonbinary patients, both in terms of overall distribution and interquartile range, likely stems from the smaller sample size of nonbinary patients relative to both male and female patients.

Bivariate Stats Overall:

1. **Contingency Table**: This table shows the observed counts of each combination of "Complication_risk" and "Additional_charges_category." For example, in the "High" complication risk category, there are 759 observations in the "Low" Additional_charges category, 845 in the "Medium" category, 886 in the "High" category, and 868 in the "Very High" category.
2. **Chi-Square Test Results**:
    - **Chi-square value**: This value indicates the strength of the association between the two categorical variables ("Complication_risk" and "Additional_charges_category"). A higher chi-square value suggests a stronger association.
    - **P-value**: This value represents the probability of observing the data if the null hypothesis (no association between the variables) were true. A smaller p-value indicates stronger evidence against the null hypothesis.
3. With a p-value of 0.0004026, which is less than the typical significance level of 0.05, we reject the null hypothesis. Therefore, we conclude that there is evidence to support the alternative hypothesis that there is an association between "Complication_risk" and "Additional_charges_category."

These results suggest that there is a statistically significant relationship between the level of complication risk and the amount of additional charges, as measured by the chi-square test.

The analysis of bivariate statistics reveals insights into the relationships between variables portrayed in two plots. Meanwhile, the comparison of mean Vitamin D levels across gender groups demonstrates relatively small differences. Females and males exhibit similar mean levels of Vitamin D, with nonbinary individuals showing slightly higher means. The broader range of Vitamin D levels observed in females, particularly in outliers, suggests a potential variability within this group. Similarly, nonbinary individuals also exhibit a wider range, likely attributed to their smaller sample size compared to males and females. These analyses provide valuable insights into the relationships between variables and potential influencing factors within the dataset, corroborating the visual patterns observed in the plots.

E1. Discuss the results of the hypothesis test.

$H_0: readmission_{hypertension} = readmission_{population}$

$H_1: readmission_{hypertension} \neq readmission_{population}$

As previously indicated, we employed an alpha level of 0.05 (ensuring 95% certainty) to evaluate the null hypothesis, positing no discrepancy in readmission rates between patients with chronic back pain and the rest of the population, against the alternative hypothesis suggesting otherwise. The resulting p-value from the chi-square test was around 0.837, surpassing the alpha value of 0.05. Consequently, we do not reject the null hypothesis, indicating no statistically significant difference in readmission rates between these two groups. It's worth noting the reference to hypertension later in the explanation, which might require clarification for coherence.

E2. Discuss the limitations of your data analysis.

In the analysis conducted, the dataset comprised a total of 10,000 records, segmented into a 2x2 contingency table based on patients' readmission status and whether they had hypertension. These four groups ranged from 1506 patients (hypertensive and readmitted) to 3747 patients (non-hypertensive and not readmitted), summing up to the total patient count of 10,000. While this sample size might appear modest, especially within a nationwide network of hospitals, a larger and more comprehensive dataset could potentially yield different outcomes by reducing the standard deviation of readmission rate distributions.

Initially, the assumption was that patients with hypertension, known for its association with stroke and heart attacks, might experience increased hospital readmissions due to other potential health complications. However, this hypothesis was not supported by the data. One possible explanation could be that since all hospitalized patients have underlying health issues requiring medical attention, hypertension may not significantly contribute to readmission rates within this context. It's plausible that hypertension, while a notable health concern, might be overshadowed by other more pressing medical conditions, such as stroke and heart attacks, among hospitalized patients. This speculation warrants further investigation and cannot be inferred solely from this dataset. Nonetheless, it presents a plausible avenue for future research and analysis.

E3. Recommend a course of action based on your results.

As the analysis resulted in the acceptance of the null hypothesis, there are no immediate recommendations concerning healthcare outcomes or the reduction of readmission rates. While it

could be suggested that this outcome might justify discontinuing the collection and recording of this particular data, such a conclusion assumes a narrow focus solely on hypertension versus readmission. However, hypertension could be intertwined with other factors or combined with additional variables to more significantly influence readmission rates. Thus, I advocate for the ongoing collection and recording of this data, as its potential benefits are valuable for further exploration and analysis.

F. Provide a Panopto video recording that includes a demonstration of the functionality of the code used for the analysis and a summary of the tool(s) used.

This will be provided with the submission

G. Reference the web sources used to acquire segments of third-party code to support the analysis.

The SciPy chi2_contingency function proved instrumental in leveraging the capabilities of the SciPy package and conducting the Chi-square test for independence within the Python environment.

H. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized

WGU Courseware Resources and Data Camp