D207: Exploratory Data Analysis Performance Assessment

Shawn Wheeler

000364855

Table of Contents

[A1: Question for Analysis 3](#_Toc164944408)

[A2: Benefits from Analysis 3](#_Toc164944409)

[A3: Data Identification 3](#_Toc164944410)

[B1: Code 6](#_Toc164944411)

[B2: Output 6](#_Toc164944412)

[B3: Justification 7](#_Toc164944413)

[C: Univariate Statistics 8](#_Toc164944414)

[C1: Visual of Findings 8](#_Toc164944415)

[D: Bivariate Statistics 11](#_Toc164944416)

[D1: Visual of Findings 12](#_Toc164944417)

[E1: Results of Analysis 14](#_Toc164944418)

[E2: Limitations of Analysis 14](#_Toc164944419)

[E3: Recommended Course of Action 14](#_Toc164944420)

[F: Video 14](#_Toc164944421)

[G: Source for Third-Party Code 15](#_Toc164944422)

[H: Sources 15](#_Toc164944423)

## A1: Question for Analysis

The question proposed for this assessment is “does a significant relationship between the Services variable and the ReAdmis variable exist?” More broadly, my question would refer to finding which variables are worthy of further investigation with regards to readmissions, but for the scope of this project, I will only be using the Services variable in a Chi-Square test of independence along with ReAdmis. As such, H0 : P1 != P2. The null hypothesis is that there is no association between ReAdmis and Services based on an alpha level of .05. This means that the alternative hypothesis states that there is, in fact, an association between ReAdmis and Services represented as H1 : P1 = P2.

## A2: Benefits from Analysis

This analysis could benefit shareholders and the company by providing insight into what should be focused on in order to increase or decrease the number of readmissions, in this case. A company solely focused on profits and thus most beneficial to shareholders might be more interested in determining methods of increasing the readmission rate of patients if they find that to be most profitable. Oppositely, one might want to utilize the various significances to lower readmissions in order to engage in a broader outreach for new patients as there could be more rooms, beds, or available staff to accommodate them.

## A3: Data Identification

*See attached code:* d207complete.ipynb

The data claims to have been cleaned for this project, but I will be applying additional data cleaning code from my d206 project to the medical\_clean data set shown at the end of topic A3. I’ll also include a brief summary of the tasks performed. After this cleaning, the data set remained 10,000 rows and 50 columns. All data was considered for cleaning purposes and no duplicates were detected. Many alterations were implemented in order to adhere more closely to the data dictionary.

Missing data was identified within the data set and all were replaced with appropriate values to ensure the total rows of 10,000 for all variables with more usable data. Some data types were altered as needed to suit the usage of data as it pertains to this project.

Income had its null values and outliers replaced with median values for this project via univariate imputation. Additional outliers aside from Income were discovered and remain in place for the purposes of this project. The logic being applied is that all values were within acceptable and reasonable ranges even if outside of the normal range of 3 standard deviations and were irrelevant to the questions to be answered within this project.

The variables that will be focused on for the purposes of this project’s question are ReAdmis and Services as shown below with relevant data. Both variables are categorical. Performing the Chi-Square test for independence will show the relationship of dependent or independent between them. These variables further meet the criteria for a Chi-Square test because the observations are independent, their categorical levels are mutually exclusive, and data provided will allow an answer for the original question.

*Table Describing ReAdmis and Services:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Name** | **Data Type** | **Description** | **Possible Values (after cleaning)** |
| ReAdmis | Qualitative  (categorical) | Within a month of visit, patient returned | no, yes |
| Services | Qualitative  (categorical) | Service received by each patient | CT scan, intravenous, blood work, MRI |

*Additional copy of data cleaning code below:*

import pandas as pd

import numpy as np

from scipy.stats import chi2\_contingency

import seaborn

import matplotlib.pyplot as plt

import statistics

%matplotlib inline

# create dataframe, open file for manipulation

df = pd.read\_csv("C:/Users/Owner/medical\_clean.csv")

# minimum was far too low as zip must have 5 digits

df['Zip'] = df['Zip'].astype("str").str.zfill(5)

# Area Replacing capitalized letters using map in pandas

df['Area'] = df['Area'].map({'Rural': 'rural', 'Urban': 'urban', 'Suburban': 'suburban'})

# Timezone renamed by using rename in pandas

df.rename(columns={'Timezone': 'TimeZone'}, inplace=True)

# Children Data mismatch - Had to fill in NAs with 0s before changing float64 to Int64

# Children Missing data - Simply replaced with 0

df['Children'] = df['Children'].fillna(0).astype(np.int64)

# Age data type mismatch changing to Int64

# Missing values Int64 allows dealing with NA

df['Age'] = df['Age'].astype("Int64")

df['Age'].dropna(inplace=True)

# Setting outliers to NaN

df['Income'] = np.where(df['Income'] > 127029, np.nan, df['Income'])

# Setting NaN and outliers to Median value

df['Income'].fillna(df['Income'].median(), inplace=True)

# Changed data type

df['Income'] = np.int64(df['Income'])

# Change Prefer not to answer to nonbinary as per data dictionary

df['Gender'] = df['Gender'].map({'Male': 'male', 'Female': 'female', 'Nonbinary': 'nonbinary'})

# ReAdmis Yes/No to yes/no

df['ReAdmis'] = df['ReAdmis'].map({'Yes':'yes', 'No':'no'})

# Null values rows filled with No to become no

df.Soft\_drink.fillna('No', inplace=True)

# Soft drink changed from Yes / No to yes/no per data dictionary

df['Soft\_drink'] = df['Soft\_drink'].map({'Yes': 'yes', 'No': 'no'})

# Changing words to match data dictionary

df['Initial\_admin'] = df['Initial\_admin'].map({'Emergency Admission': 'emergency admission', 'Elective Admission': 'elective admission', 'Observation Admission': 'observation'})

# HighBlood Yes/No to yes/no

df['HighBlood'] = df['HighBlood'].map({'Yes': 'yes', 'No': 'no'})

# Stroke Changing Yes/No to yes/no

df['Stroke'] = df['Stroke'].map({'Yes': 'yes', 'No': 'no'})

# Change to lowercase

df['Complication\_risk'] = df['Complication\_risk'].map({'High': 'high', 'Medium': 'medium', 'Low':'low'})

# Replace null values with No

# Changed 1,0s to yes,no

df['Overweight'] = df['Overweight'].fillna(0)

df['Overweight'] = df['Overweight'].astype(str)

df.Overweight.replace({'1':'yes', '0':'no'}, inplace=True)

# Yes/No replaced with yes/no

df['Arthritis'] = df['Arthritis'].map({'Yes': 'yes', 'No': 'no'})

# Converting Yes/No to yes/no

df['Diabetes'] = df['Diabetes'].map({'Yes': 'yes', 'No': 'no'})

# Yes/No replacement

df['Hyperlipidemia'] = df['Hyperlipidemia'].map({'Yes': 'yes', 'No': 'no'})

# replacing Yes/No

df['BackPain'] = df['BackPain'].map({'Yes': 'yes', 'No': 'no'})

# Replace null values with 0

df['Anxiety'] = df['Anxiety'].fillna(0)

# Remap values and replace Yes / No for yes/no

df['Anxiety'] = df['Anxiety'].astype(str)

df.Anxiety.replace({'1':'yes', '0':'no'}, inplace=True)

# Changed Yes/No to yes/no

df['Allergic\_rhinitis'] = df['Allergic\_rhinitis'].map({'Yes': 'yes', 'No': 'no'})

# Changed Yes/No to yes/no

df['Reflux\_esophagitis'] = df['Reflux\_esophagitis'].map({'Yes': 'yes', 'No': 'no'})

# Yes/No to yes/no

df['Asthma'] = df['Asthma'].map({'Yes': 'yes', 'No': 'no'})

# replacing Blood Work and Intravenous with lower case for data dictionary as well as the s in CT Scan

df['Services'] = df['Services'].map({'Blood Work': 'blood work', 'Intravenous': 'intravenous', 'CT Scan': 'CT scan', 'MRI': 'MRI'})

# Dropping null values rows

df['Initial\_days'] = np.where(df['Initial\_days'] == '', np.nan, df['Initial\_days'])

# Filling null values with median

df['Initial\_days'].fillna(df['Initial\_days'].median(), inplace=True)

## B1: Code

*See attached code:* d207complete.ipynb

*Additional copy of Chi-Square test below:*

# Following course material for chi-square test, creating table (Course material, n.d.)

chi = pd.crosstab(df['ReAdmis'], df['Services'])

chi

# Heatmap to visualize relationship (Course material, n.d.)

plt.figure(figsize=(8,6))

seaborn.heatmap(chi, annot=True, cmap="PuBuGn")

# Chi-square test for independence

c, p, dof, expected = chi2\_contingency(chi)

# Print the p-value

print('P-value is: ', p)

# interpret p-value

alpha = 0.05

if p <= alpha:

print('Null is rejected')

else:

print('Null can not be rejected')

## B2: Output

*Crosstab of Services and ReAdmis:*

A screenshot of a graph

Description automatically generated

*Heatmap of Services and ReAdmis:*

A screenshot of a color chart

Description automatically generated

*P-value output and if statement message:*

 

The p-value given is approximately .03 and “Null is rejected” was also returned from the if statement. Given that the alpha value was set at .05 and .03 is less than that, this proves with 95% certainty the alternative hypothesis ,H1 : P1 = P2, is accurate and the null hypothesis must be rejected. The heatmap provided is for visual representation showing the strength of the relationship between ReAdmis and Services. The two variables are dependent upon each other.

## B3: Justification

I opted to perform the Chi-Square test of independence as I felt it best suited the needs of the question itself and the data to be used. The project question essentially asks to perform a hypothesis test that is non-parametric based on the ReAdmis and Services variables. The method used to accomplish this was chi2\_contingency() from the scipy package. A t-test was not chosen because it seeks to determine a null hypothesis about two means to see if they are equal which is not what the original question is asking (Course material, n.d.). ANOVA was also disregarded as an option immediately as it requires more than two variables. A Chi-Square test also does not require testing for normality as other methods may. Using Chi-Square test of independence will allow us to answer the project question of whether ReAdmis and Services have a significant relationship.

## C: Univariate Statistics

Using univariate statistics, I analyzed distribution for four variables. Initial\_days and Additional\_charges were the two continuous variables while Services and Area were considered for the categorical variables. The seaborn and matplotlib packages were used in the creation of all graphs included.

As seen below, Initial\_days is visualized using a violinplot which shows distribution of data while also including density for the underlying distribution (Waskom, 2024). Looking at this, we can see the highest concentration around the 1st and 3rd quartile (around 46% and 37% of values respectively based on a bin size of four) meaning that it is bimodally distributed. The Additional\_charges variable is represented with a histogram which shows a strong right skew meaning the mean, 12,934, is greater than the median, 11,573.

With the categorical variables, I chose to represent Services with a horizontal bar graph and Area with a pie graph. As shown in the bar graph, the data for Services is skewed heavily with blood work being provided over 52% of the time while the other three options combine for the remaining 48%. Looking at the pie chart, Area seems to be very normally distributed with less than a 1% difference between the frequencies of categories overall.

## C1: Visual of Findings

*See attached code:* d207complete.ipynb

*Copy of univariate statistics code:*

# univariates

# 2 continuous - Initial\_days, Additional\_charges

# 2 categorical - Initial\_admin, Area

# Initial\_days - violin plot

plt.title("Distribution of Days of Initial Admission")

seaborn.violinplot(data=df, x="Initial\_days", cut=3, density\_norm="count", width=1, color="coral", inner="quartile")

plt.xlabel("Days of Initial Admission")

plt.ylabel("Frequency")

df.Initial\_days.describe()

df.Initial\_days.value\_counts(normalize=True, bins=4)

# Additional\_charges - Histogram / tables of values / describe

plt.title("Distribution of Additional Charges")

plt.hist(data=df, x="Additional\_charges", bins=500, color="turquoise")

plt.xlabel("Additional Charges")

plt.ylabel("Frequency")

df.Additional\_charges.describe()

statistics.median(df.Additional\_charges)

# Services - horizontal bar chart histogram

plt.subplot(2,1,1)

plt.title("Distribution of Service Provided")

plt.hist(data=df, x="Services", orientation="horizontal", color="brown")

plt.xlabel("Frequency")

plt.ylabel("Service Provided")

df.Services.value\_counts()

# Area - Pie chart (Training,2023)

area\_counts = df["Area"].value\_counts()

area\_labels = ["Rural", "Suburban", "Urban"]

explode = [0.01, 0.01, 0.01]

plt.title("Distribution of Area")

plt.pie(area\_counts, labels=area\_labels, explode=explode, autopct='%.01f%%', )

df.Area.value\_counts(normalize=True)

*Violin Plot for Initial\_days:*

A diagram of a number of days

Description automatically generated

*Counts and Percentages in Four Bins for Initial\_days:*

A white background with black numbers

Description automatically generated A number of numbers and letters

Description automatically generated with medium confidence

*Histogram for Additional\_charges:*

A chart of a distribution of charge

Description automatically generated

*Counts, mean, median for Additional\_charges:*

A screenshot of a computer

Description automatically generated 

*Horizontal Bar Graph for Distribution of Services:*

A bar graph with numbers and a number of red bars

Description automatically generated with medium confidence

*Services Counts:*

A white background with black text

Description automatically generated

*Pie Chart for Distribution of Area:*

A pie chart with numbers and a few different colored circles

Description automatically generated

*Normalized Counts for Area:*

A close-up of numbers

Description automatically generated

## D: Bivariate Statistics

For the bivariate analysis section of this project, two sets of variables were examined. One pair of variables, Gender and Marital, are categorical, while the other set, Income and VitD\_levels, are continuous. The nature of the relationship between Gender and Marital is shown below as a vertical bar graph grouped by Gender. This shows a very normal distribution for Gender and Marital. VitD\_levels and Income’s relationship is presented below in a scatterplot. This, also, shows a very normal distribution.

## D1: Visual of Findings

*See attached code:*  d207complete.ipynb

*Copy of attached bivariate statistics code:*

# bivariates

# categorical - Gender, Marital

# continuous - VitD\_levels, Income

# Categorical - Gender, Marital - bar chart same as above (Waskom, 2012)

ax = seaborn.countplot(data=df, x='Gender', hue='Marital', order=['male', 'female', 'nonbinary'], hue\_order=['Married', 'Divorced', 'Never Married', 'Widowed'], palette={'Married': 'green', 'Divorced':'blue', 'Never Married':'Red', 'Widowed':'orange'})

plt.title("Relationship of Gender and Marital Status")

plt.ylabel("Frequency")

gender\_marital\_cross = pd.crosstab(df.Gender, df.Marital, margins=True)

print(gender\_marital\_cross)

# Continuous - VitD\_levels, Income - scatterplot

seaborn.set\_theme(style='whitegrid')

plt.title("Relationship between Income and Vitamin D Levels")

seaborn.scatterplot(data=df, x="VitD\_levels", y="Income")

plt.xlabel("Vitamin D Levels")

df.Income.describe()

df.VitD\_levels.describe()

*Bivariate Countplot Bar Chart for Gender and Marital Status:*

A graph of a person and person

Description automatically generated

*Crosstab for Gender and Marital:*

A white background with black text

Description automatically generated

*Scatter Plot for Income and VitD\_levels:*

A diagram of a number of blue dots

Description automatically generated

*Counts for Income and VitD\_levels:*

A screenshot of a computer

Description automatically generatedA screenshot of a computer code

Description automatically generated

## E1: Results of Analysis

The null hypothesis was rejected. The alternative, H1 : P1 = P2, is true and the null, H0 : P1 != P2, is proven false. There is an association between Services and ReAdmis. As shown before, the p-value calculated was below the alpha threshold of .05 revealing a significant relationship at over 95% certainty. Both variables, ReAdmis and Services, are also therefore dependent variables with regards to one another. Further investigation is warranted in this case.

## E2: Limitations of Analysis

Although the information presented is quite useful, there are limitations of this analysis. Even though the data supports a strong relationship between ReAdmis and Services, it’s not necessarily analysis that would prove useful in a business setting. Perhaps the broad nature of the services listed are not the actual root of the issue causing Services to act more as a red herring showing correlation and causation are not always bound together. Or, even more simply, there may be no useful application of the data itself as there may not be a way to determine a way to manipulate the ReAdmis and Services variables in one’s favor.

## 

## E3: Recommended Course of Action

The original question purported was “does a significant relationship between the Services variable and the ReAdmis variable exist?” After analysis, the answer is yes, a significant relationship does exist between Services and ReAdmis. I would absolutely recommend further investigation into the connection between readmissions and services provided. Since the null hypothesis was rejected, this is the logical course of action with above a 95 percent certainty based on our p-value. Furthermore, there are many other potential candidates within the data set that would be worth testing in the same or similar ways to see if similar results could be found elsewhere in the data. Of course, I would lastly recommend continuing collecting this data to increase the sample size over time and thus accuracy and usability while also considering a new set of variables related to ReAdmis and Services to gather data for future, more precise analysis.

## F: Video

*Link attached : https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=ccaa2c33-83bc-46ef-84dc-b15f01665299*

## G: Source for Third-Party Code

Course Material

Training, P. (2023, May 18). *Seaborn Pie Chart: A Tutorial for Data Visualization*. Pierian Training. https://pieriantraining.com/seaborn-pie-chart-a-tutorial-for-data-visualization/

Waskom, M. (2012). *seaborn.countplot — seaborn 0.9.0 documentation*. Pydata.org. https://seaborn.pydata.org/generated/seaborn.countplot.html

## H: Sources

Course Material

Waskom, M. (2024). *seaborn.violinplot — seaborn 0.10.1 documentation*. (n.d.). Seaborn.pydata.org. https://seaborn.pydata.org/generated/seaborn.violinplot.html