SI 624
Healthcare Data Application, Analysis, Consulting and Communication

Ashruti Tuteja

Contents

Project Scope

We are planning to explore MIMIC data, to understand the trend and prevalence of chronic diseases in the presence of various physiological conditions amongst ICU admitted patients.

INTRODUCTION

- About dataset
- Data dictionary
- Data Glossary (will repeat again)
- Data Overview
- Objective / Questions
- Stakeholders
- Application: Impact on the real world
    - Triple Aim

About Dataset
MIMIC

- The dataset is derived from MIMIC-II, the publicly-accessible critical care database.
- It contains a summary of clinical data and outcomes for 1,776 patients.
- The dataset in question was used throughout Chapter 16 (Data Analysis) by Raffa J. et al. **to investigate the effectiveness of indwelling arterial catheters in hemodynamically stable patients with respiratory failure for mortality outcomes.**

Data Dictionary

aline_flg	IAC used (binary, 1 = year, 0 = no)	censor_flg	censored or death (binary: 0 = death, 1 = censored)
icu_los_day	length of stay in ICU (days, numeric)	sepsis_flg	sepsis present (binary: 0 = no, 1 = yes -- absent (0) for all)
hospital_los_day	length of stay in hospital (days, numeric)	chf_flg	Congestive heart failure (binary: 0 = no, 1 = yes)
age	age at baseline (years, numeric)	afib_flg	Atrial fibrillation (binary: 0 = no, 1 = yes)
gender_num	patient gender (1 = male; 0=female)	renal_flg	Chronic renal disease (binary: 0 = no, 1 = yes)
weight_first	first weight, (kg, numeric)	liver_flg	Liver Disease (binary: 0 = no, 1 = yes)
bmi	patient BMI, (numeric)	copd_flg	Chronic obstructive pulmonary disease (binary: 0 = no, 1 = yes)
sapsi_first	first SAPS I score (numeric)	cad_flg	Coronary artery disease (binary: 0 = no, 1 = yes)
sofa_first	first SOFA score (numeric)	stroke_flg	Stroke (binary: 0 = no, 1 = yes)
service_unit	type of service unit (character: FICU, MICU, SICU)	mal_flg	Malignancy (binary: 0 = no, 1 = yes)
mort_day_censored	day post ICU admission of censoring or death (days, numeric)	hgb_first	first Hemoglobin (g/dL, numeric)

Data Dictionary

aline_flg	IAC used (binary, 1 = year, 0 = no)	censor_flg	censored or death (binary: 0 = death, 1 = censored)
service_num	service as a numeric (binary: 0 = MICU or FICU, 1 = SICU)	resp_flg	Respiratory disease (non-COPD) (binary: 0 = no, 1 = yes)
day_icu_intime	day of week of ICU admission (character)	map_1st	Mean arterial pressure (mmHg, numeric)
day_icu_intime_num	day of week of ICU admission (numeric, corresponds with day_icu_intime)	hr_1st	Heart Rate (numeric)
hour_icu_intime	hour of ICU admission (numeric, hour of admission using 24hr clock)	temp_1st	Temperature (F, numeric)
hosp_exp_flg	death in hospital (binary: 1 = yes, 0 = no)	spo2_1st	S_pO_2 (%, numeric)
icu_exp_flg	death in ICU (binary: 1 = yes, 0 = no)	abg_count	arterial blood gas count (number of tests, numeric)
day_28_flg	death within 28 days (binary: 1 = yes, 0 = no)	wbc_first	first White blood cell count (K/uL, numeric)
mort_day_censored	day post ICU admission of censoring or death (days, numeric)	hgb_first	first Hemoglobin (g/dL, numeric)

Data Dictionary

platelet_first	first Platelets (K/u, numericL)
sodium_first	first Sodium (mEq/L, numeric)
potassium_first	first Potassium (mEq/L, numeric)
tco2_first	first Bicarbonate (mEq/L, numeric)
chloride_first	first Chloride (mEq/L, numeric)
bun_first	first Blood urea nitrogen (mg/dL, numeric)
creatinine_first	first Creatinine (mg/dL, numeric)
po2_first	first PaO_2 (mmHg, numeric)
iv_day_1	input fluids by IV on day 1 (mL, numeric)
pco2_first	first PaCO_2 (mmHg, numeric)

Data Glossary

SAPS score: Estimates the probability of mortality for ICU patients on admission.

SOFA score: The Sequential Organ Failure Assessment (SOFA) score is a scoring system that assesses the performance of several organ systems in the body (neurologic, blood, liver, kidney, and blood pressure/hemodynamics).

*REFERENCE:
https://www.mdcalc.com/calc/10403/simplified-acute-physiology-score-saps-3#:~:text=Estimates%20the%20probability%20of%20mortality%20for%20ICU%20patients%20on%20admission.&text=The%20SAPS%203%20Score%20predicts,physiologic%20derangement%20upon%20ICU%20admission.*

Data Overview

A glimpse of the dataset can be achieved through functions such as:

- sample: This will randomly pick some rows to display, or
- Head: Provide first few rows of the dataset, or
- Tail: It will display the last few rows of the dataset.

Functions to view size of the dataset:

- Shape: Tuple of number of rows and columns in the dataset
- len(): to find Length of df, i.e. Rows
- len(df.columns): to find Length of df.columns

Data Overview

To fetch data types of columns in a dataframe:

- df.dtypes: to find the type of data that the dataframe contains

To find null values in a dataframe:

- pd.isnull(): check for null values, and returns the boolean True if Null / NA / NaN
- df.isnull().sum(): returns the sum of null values in each fiel

A glimpse of the dataframe

Data Sample

```
data.head(5)
```

	aline_flg	icu_los_day	hospital_los_day	age	gender_num	weight_first	bmi	sapsi_first	sofa_first	service_unit	...	platelet_first	sodium_firs
0	1	7.63	13	72.36841	1.0	75.0	29.912791	15.0	9.0	SICU	...	354.0	138
1	0	1.14	1	64.92076	0.0	55.0	20.121312	NaN	5.0	MICU	...	NaN	Na
2	0	2.86	5	36.50000	0.0	70.0	27.118272	16.0	5.0	MICU	...	295.0	144
3	1	0.58	3	44.49191	0.0	NaN	NaN	21.0	7.0	SICU	...	262.0	139
4	1	1.75	5	23.74217	1.0	95.2	28.464563	18.0	7.0	SICU	...	22.0	146

5 rows × 46 columns

46 columns:

Each column is a physiological factor for each patient amongst 1776 patients in the dataset.

Size of the Dataset

Data Shape

```
[ ] print("The shape of the original dataset is: ", data.shape)

The shape of the original dataset is:  (1776, 46)
```

Column Names

```
data.columns

Index(['aline_flg', 'icu_los_day', 'hospital_los_day', 'age', 'gender_num',
       'weight_first', 'bmi', 'sapsi_first', 'sofa_first', 'service_unit',
       'service_num', 'day_icu_intime', 'day_icu_intime_num',
       'hour_icu_intime', 'hosp_exp_flg', 'icu_exp_flg', 'day_28_flg',
       'mort_day_censored', 'censor_flg', 'sepsis_flg', 'chf_flg', 'afib_flg',
       'renal_flg', 'liver_flg', 'copd_flg', 'cad_flg', 'stroke_flg',
       'mal_flg', 'resp_flg', 'map_1st', 'hr_1st', 'temp_1st', 'spo2_1st',
       'abg_count', 'wbc_first', 'hgb_first', 'platelet_first', 'sodium_first',
       'potassium_first', 'tco2_first', 'chloride_first', 'bun_first',
       'creatinine_first', 'po2_first', 'pco2_first', 'iv_day_1'],
      dtype='object')
```

1776 rows (each patient)

and

46 columns (physiological and anatomical factors)

Understanding the data types of the columns in the dataframe

- Most of the columns seem to contain either integer or float values.
- **Continuous data values** make it easier for analysis.

```
data.dtypes

aline_flg                 int64
icu_los_day             float64
hospital_los_day          int64
age                     float64
gender_num              float64
weight_first            float64
bmi                     float64
sapsi_first             float64
sofa_first              float64
service_unit             object
service_num               int64
day_icu_intime           object
day_icu_intime_num        int64
hour_icu_intime           int64
hosp_exp_flg              int64
icu_exp_flg               int64
day_28_flg                int64
mort_day_censored       float64
censor_flg                int64
sepsis_flg                int64
chf_flg                   int64
afib_flg                  int64
renal_flg                 int64
liver_flg                 int64
copd_flg                  int64
cad_flg                   int64
stroke_flg                int64
mal_flg                   int64
resp_flg                  int64
map_1st                 float64
hr_1st                    int64
temp_1st                float64
spo2_1st                  int64
abg_count                 int64
wbc_first               float64
hgb_first               float64
platelet_first          float64
sodium_first            float64
potassium_first         float64
tco2_first              float64
chloride_first          float64
bun_first               float64
creatinine_first        float64
po2_first               float64
pco2_first              float64
iv_day_1                float64
```

Statistics - A description of the dataset

Basic Statistics

```
data.describe()
```

	aline_flg	icu_los_day	hospital_los_day	age	gender_num	weight_first	bmi	sapsi_first	sofa_first	service_num	...	platelet_first	so
count	1776.000000	1776.000000	1776.000000	1776.000000	1775.000000	1666.000000	1310.000000	1691.000000	1770.000000	1776.000000	...	1768.000000	
mean	0.554054	3.346498	8.110923	54.379660	0.577465	80.075948	27.827316	14.136606	5.820904	0.552928	...	246.083145	
std	0.497210	3.356261	8.157159	21.062854	0.494102	22.490516	8.210074	4.114302	2.334666	0.497331	...	99.865469	
min	0.000000	0.500000	1.000000	15.180230	0.000000	30.000000	12.784877	3.000000	0.000000	0.000000	...	7.000000	
25%	0.000000	1.370000	3.000000	38.247318	0.000000	65.400000	22.617307	11.000000	4.000000	0.000000	...	182.000000	
50%	1.000000	2.185000	6.000000	53.678585	1.000000	77.000000	26.324846	14.000000	6.000000	1.000000	...	239.000000	
75%	1.000000	4.002500	10.000000	72.762992	1.000000	90.000000	30.796551	17.000000	7.000000	1.000000	...	297.000000	
max	1.000000	28.240000	112.000000	99.110950	1.000000	257.600000	98.797134	32.000000	17.000000	1.000000	...	988.000000	

8 rows × 44 columns

- **Count:** Some values are missing from various columns, as count varies in each columns
- **Mean:** age is 54, SAPS score is 14, SOFA score 5.8
- **Min & max:** reveals the extreme values for various physiological factors, like for platelet 7 (min), 988 (max)
- **The quantiles:** if plotted will boxplot will help us determining outliers for each of the physiological factors.

Check missing values

Check Missing Values

```
data.isna().sum()
```

```
aline_flg              0
icu_los_day            0
hospital_los_day       0
age                    0
gender_num             1
weight_first         110
bmi                  466
sapsi_first           85
sofa_first             6
service_unit           0
service_num            0
day_icu_intime         0
day_icu_intime_num     0
hour_icu_intime        0
hosp_exp_flg           0
icu_exp_flg            0
day_28_flg             0
mort_day_censored      0
censor_flg             0
sepsis_flg             0
chf_flg                0
afib_flg               0
renal_flg              0
liver_flg              0
copd_flg               0
cad_flg                0
stroke_flg             0
mal_flg                0
resp_flg               0
map_1st                0
hr_1st                 0
temp_1st               3
spo2_1st               0
abg_count              0
wbc_first              8
hgb_first              8
platelet_first         8
sodium_first           5
potassium_first        5
tco2_first             5
chloride_first         5
bun_first              5
creatinine_first       6
po2_first            186
pco2_first           186
iv_day_1             143
dtype: int64
```

Factors with missing values:

Weight,
BMI,
SAPS score,
pO2,
pCO2

Complete dataset with all complete values counts for 1690 patients

Remove Missing Values

```
# Finding Index of Missing Values
gender_index = data[data['gender_num'].isna()].index.tolist()
sapsi_index = data[data['sapsi_first'].isna()].index.tolist()

# Accumulate all the indexes in one list
missing_data = [y for x in [sapsi_index, gender_index ] for y in x]

# Remove missing values in each row of every column which will be used for further analysis
new_data = data.drop(labels=missing_data, axis=0)
new_data.shape
```

```
(1690, 46)
```

Complete dataset with all complete values counts for 1690 rows that is 1690 patients.

Strengths & Weaknesses of the Dataset

Strengths:

- Reliable dataset – MIMIC is reputed and open data source for medical data
- Extensive – incorporating many attributes (~46 columns)
- Meaningfulness - Data dictionary is self-explanatory
- Completeness – Less missing values

Weakness of your dataset

- Less instances – 1776 rows depicting 1776 patients
- Validity – source is unknown

HEALTH RELATED QUESTION

The study would help us to have an idea on how various physiological or anatomical factors impact the prevalence of chronic diseases to overall affect the SAPS score of the patient.

Population

Our population is the patients requiring mechanical ventilation who did not require vasopressors or have a diagnosis of sepsis were identified, and the primary outcome was 28-day mortality

Intervention or Exposure Variable

Various chronic Diseases
- a binary variable where 0 is a negative outcome and 1 is a positive.

Comparison

Gender
SAPS score
Age

Outcome Variable

The outcome variable is censored or death which is a binary variable indicative of death when equal to 0 and indicative of censored when equal to 1.

STAKEHOLDERS

- Clinical researchers
- Academic Researchers
  - Students
  - Faculty
- Data Team:
  - Data Analyst
  - Data Extraction Associate
  - Implementation Analyst
  - Software Engineers
- Health Policy workers:
  - Local health officers
  - Epidemiology staff
  - National Health Agencies like CDC, WHO

HEALTHCARE IMPACT (TRIPLE AIM)

Improving the experience of care

Healthcare organizations might consider utilizing a greater portion of the facilities for patients with worsen physiological conditions.

Improving the health of population

Analysis of physiological and anatomical factors leading to chronic diseases can help diminish the chances of deteriorating conditions

Reducing per capita costs of healthcare

As a preventive measure, providing medical attention and care earlier to a vulnerable population will lead to less cost injection in the later deteriorating stages.

OBJECTIVES

- Health Related Question
- Approach (Code Screenshots)
- Solution/Analysis
- Relevance
- Inference

OBJECTIVE

The questions we foresee to answer from this dataset:

1. **Chronic disease prevalence according to gender.**
   - Liver
   - Kidney
   - Heart

2. **Impact of clinical indicators on the occurrence of chronic diseases in patients.**
   a. Stating causation of creatinine levels on renal disease.
   b. Hemoglobin count of patients having Congestive heart failure

3. **Correlation between total number of chronic diseases a person has versus the number of days in hospitalization and also the number of days in ICU.**

4. **How does the number of patients with chronic diseases in each age group affects:**
   - **the SAPS score on ICU admission leading to ICU mortality**
   - **The SOFA score leading to ICU mortality**

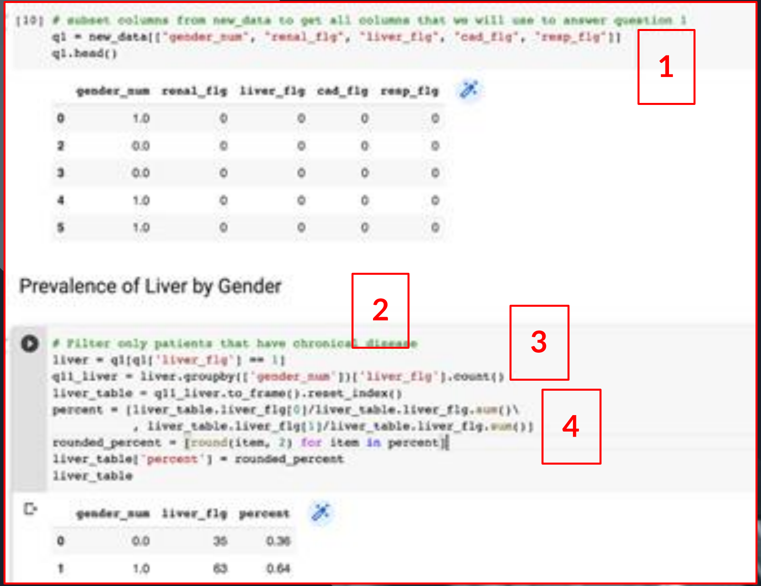
5. **Explore SOFA score by** understanding platelet count effect on SOFA score

Question 1:

Chronic disease prevalence according to gender.

- Liver
- Kidney
- Heart

Approach



```
[10] # subset columns from new_data to get all columns that we will use to answer question 1
     q1 = new_data[["gender_num", "renal_flg", "liver_flg", "cad_flg", "resp_flg"]]
     q1.head()
```

	gender_num	renal_flg	liver_flg	cad_flg	resp_flg
0	1.0	0	0	0	0
2	0.0	0	0	0	0
3	0.0	0	0	0	0
4	1.0	0	0	0	0
5	1.0	0	0	0	0

Prevalence of Liver by Gender

```
# Filter only patients that have chronical disease
liver = q1[q1['liver_flg'] == 1]
q11_liver = liver.groupby(['gender_num'])['liver_flg'].count()
liver_table = q11_liver.to_frame().reset_index()
percent = [liver_table.liver_flg[0]/liver_table.liver_flg.sum()\
           , liver_table.liver_flg[1]/liver_table.liver_flg.sum()]
rounded_percent = [round(item, 2) for item in percent]
liver_table['percent'] = rounded_percent
liver_table
```

	gender_num	liver_flg	percent
0	0.0	35	0.36
1	1.0	63	0.64

- **Step 1**

  Subset the data for required columns (gender and chronic diseases)
- **Step 2**

  Filter the patients with chronic diseases (indicator = 1)
- **Step 3**

  Group by gender in each of the disease filter
- **Step 4**

  Calculate the number in each group to divide by the total and find % prevalence.

Prevalence of Chronic Diseases by Gender

LIVER

```
# Filter only patients that have chronical disease
liver = q1[q1['liver_flg'] == 1]
q11_liver = liver.groupby(['gender_num'])['liver_flg'].count()
liver_table = q11_liver.to_frame().reset_index()
percent = [liver_table.liver_flg[0]/liver_table.liver_flg.sum()\
           , liver_table.liver_flg[1]/liver_table.liver_flg.sum()]
rounded_percent = [round(item, 2) for item in percent]
liver_table['percent'] = rounded_percent
liver_table
```

	gender_num	liver_flg	percent
0	0.0	35	0.36
1	1.0	63	0.64

KIDNEY

```
kidney = q1[q1['renal_flg'] == 1]
q11_renal = kidney.groupby(['gender_num'])['renal_flg'].count()
renal_table = q11_renal.to_frame().reset_index()
percent = [renal_table.renal_flg[0]/renal_table.renal_flg.sum()\
           , renal_table.renal_flg[1]/renal_table.renal_flg.sum()]
rounded_percent = [round(item, 2) for item in percent]
renal_table['percent'] = rounded_percent
renal_table
```

	gender_num	renal_flg	percent
0	0.0	20	0.36
1	1.0	36	0.64

CARDIAC

```
cad = q1[q1['cad_flg'] == 1]
q11_cad = kidney.groupby(['gender_num'])['cad_flg'].count()
cad_table = q11_cad.to_frame().reset_index()
percent = [cad_table.cad_flg[0]/cad_table.cad_flg.sum()\
           , cad_table.cad_flg[1]/cad_table.cad_flg.sum()]
rounded_percent = [round(item, 2) for item in percent]
cad_table['percent'] = rounded_percent
cad_table
```

	gender_num	cad_flg	percent
0	0.0	20	0.36
1	1.0	36	0.64

- **36% female following in the criteria experience the respective chronic condition**
- **Males show a higher rate of prevalence at 64%**

Prevalence of Respiratory Disease by Gender

```
resp= q1[q1['resp_flg'] == 1]
q11_resp = resp.groupby(['gender_num'])['resp_flg'].count()
resp_table = q11_resp.to_frame().reset_index()
percent = [resp_table.resp_flg[0]/resp_table.resp_flg.sum()\
          , resp_table.resp_flg[1]/resp_table.resp_flg.sum()]
rounded_percent = [round(item, 2) for item in percent]
resp_table['percent'] = rounded_percent
resp_table
```

	gender_num	resp_flg	percent
0	0.0	247	0.45
1	1.0	299	0.55

- **45% female following in the criteria experience renal chronic condition**
- **Males show a higher rate of prevalence at 55%**

Prevalence of Patients with **multiple chronic diseases** by gender

```
cd2= q1[(q1['resp_flg'] == 1) & (q1['renal_flg'] == 1)\
        | (q1['renal_flg'] == 1) & (q1['cad_flg'] == 1) | (\
        q1['cad_flg'] == 1) & (q1['liver_flg'] == 1) | (\
        q1['liver_flg'] == 1) & (q1['resp_flg'] == 1\
        ) | (q1['renal_flg'] == 1) & (q1['liver_flg'] == 1\
        ) | (q1['cad_flg'] == 1) & (q1['resp_flg'] == 1)]

cd2.value_counts().to_frame().groupby('gender_num').sum().reset_index()
```

	gender_num	0
0	0.0	42
1	1.0	61

```
cd3 = q1[(q1['renal_flg'] == 1) & (q1['liver_flg'] == 1) & (q1['cad_flg'] == 1\
        ) | (q1['liver_flg'] == 1) & (q1['cad_flg'] == 1) & (q1['resp_flg'] == 1\
        ) | (q1['cad_flg'] == 1) & (q1['resp_flg'] == 1) & (q1['renal_flg'] == 1\
        ) | (q1['renal_flg'] == 1) & (q1['cad_flg'] == 1) & (q1['resp_flg'] == 1\
        ) | (q1['renal_flg'] == 1) & (q1['liver_flg'] == 1) & (q1['resp_flg'] == 1\
        ) ]
cd3.value_counts().to_frame().groupby('gender_num').sum().reset_index()
```

	gender_num	0
0	0.0	1
1	1.0	9

- **Female following in the criteria experience lesser cumulative chronic conditions as compared to male population in the same criteria.**
- **42 females experience 2 chronic diseases comparative to 61 in males, and similarly for three chronic conditions together the females show less prevalence.**

Question 2:

(a) Impact of creatinine levels as a causation for Kidney disease.

(b) Hemoglobin count of patients having congestive heart failure.

Approach

```python
# Finding Index of Missing Values
creatinineIndex = new_data[new_data['creatinine_first'].isna()].index.tolist()

# Remove missing values from columns that will be used for further analysis
creatinine_data = new_data.drop(labels=creatinineIndex, axis=0)
# subset the data
q2 = creatinine_data[["gender_num", "creatinine_first", "renal_flg"]]
q2['early_sign'] = np.where((creatinine_data['creatinine_first'] > 1.2) & (\
                   creatinine_data['gender_num']==0)|(creatinine_data['creatinine_first'] > 1.4) & (\
                   creatinine_data['gender_num']==1), 1, 0)

ct = pd.crosstab(q2.renal_flg,q2.early_sign)
ct
```

1

2

3

early_sign	0	1
renal_flg		
0	1460	173
1	10	46

- **Step 1**
  Data Preparation: Finding missing values and removing them to subset the required columns in a new dataframe.
- **Step 2**
  Apply condition to the new subset dataset.
  Making new column with the condition met.
- **Step 3**
  Performed calculation using crosstab
- **Step 4**
  Create the heatmap (from seaborn library)
  Perform statistical test (scipy.stats)

Question 2 (a):

Impact of creatinine levels as a causation for Kidney disease.

Desired creatinine levels for patients

"Creatinine level of greater than 1.2 for women and greater than 1.4 for men may be an early sign that the kidneys are not working properly. As kidney disease progresses, the level of creatinine in the blood rises."

REFERENCE:

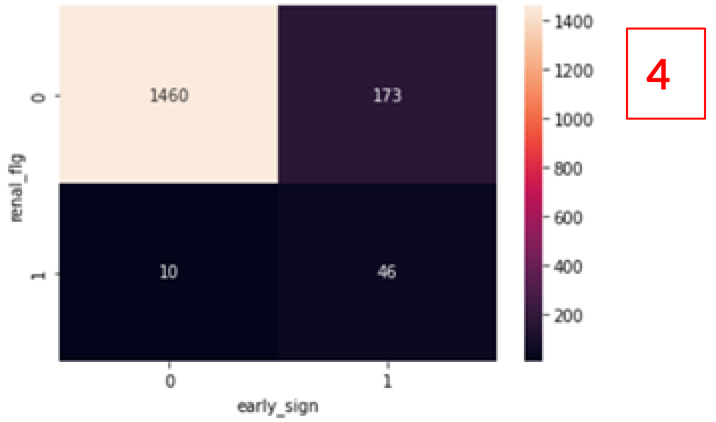
https://www.kidney.org/atoz/content/kidneytests#:~:text=A%20creatinine%20level%20of%20greater,creatinine%20in%20the%20blood%20rises

Statistics & Graphical Representation

Heatmap
Importance of early sign is distinctly visible here.
It also reveals that ONLY early sign could not be causation of presence of renal disease.

```
sns.heatmap(ct,annot=True,fmt='d')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6b42e24150>
```




```
chi2, p, dof, expected = chi2_contingency(ct)
print("chi2 = ", chi2)
print("p-val = ", p)
print("degree of freedom = ",dof)
```

```
chi2 =  239.31175981214867
p-val =  5.5561438972757737e-54
degree of freedom =  1
```

Chi-square Test of Association
As the p-value is greater than the threshold value of 0.05, we can say that creatinine levels are associated with renal disease.

Question 2 (b):

Hemoglobin count of patients having congestive heart failure

Desired hemoglobin levels for patients

"A low hemoglobin count is generally defined as less than 13.2 grams of hemoglobin per deciliter (132 grams per liter) of blood for men and less than 11.6 grams per deciliter (116 grams per liter) for women."

REFERENCE:

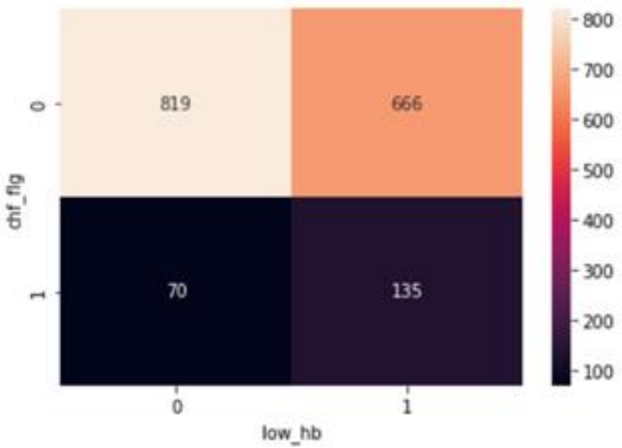
https://www.mayoclinic.org/symptoms/low-hemoglobin/basics/definition/sym-

Statistics & Graphical Representation

Heatmap
Low hemoglobin count **could be** a factor associated with congestive heart failure but there are definitely other causation parameters.

```
sns.heatmap(ctc_hf,annot=True,fmt='d')
```

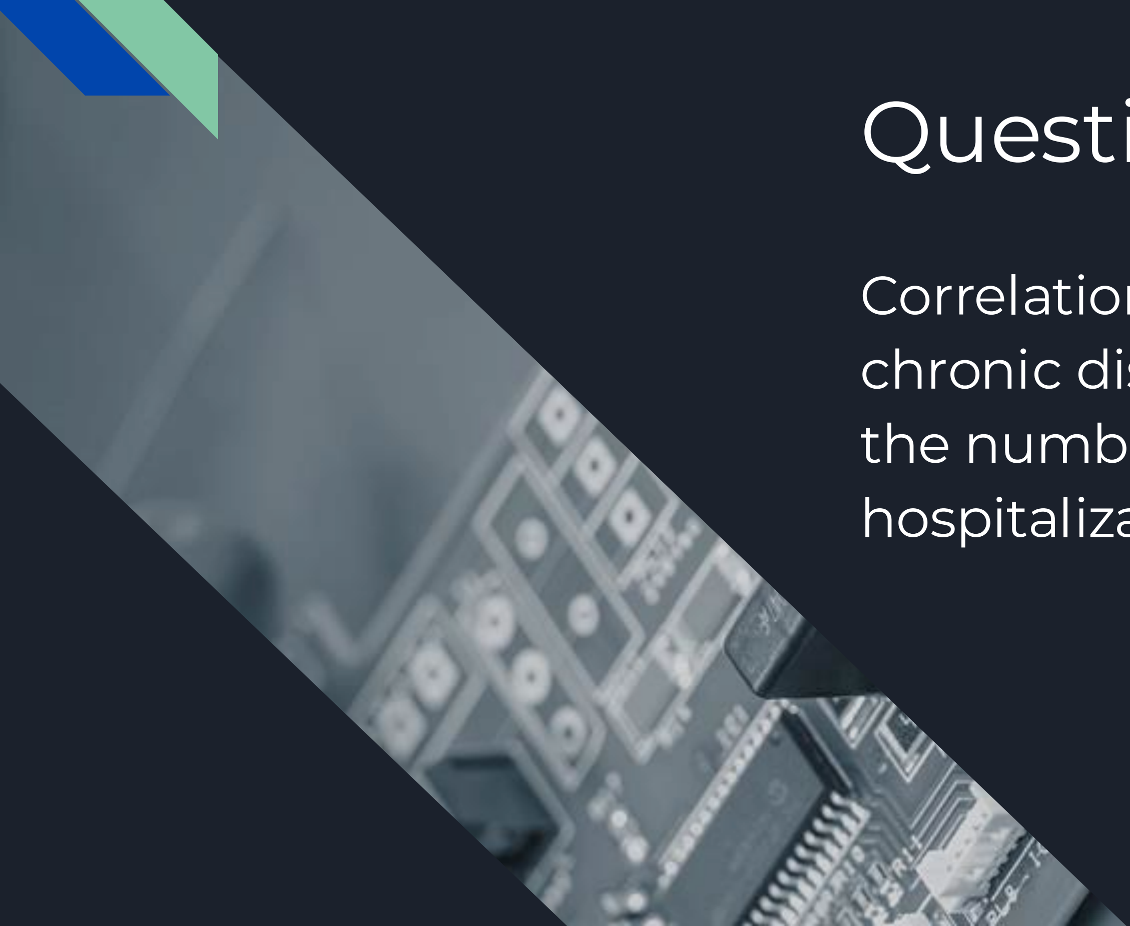
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6b40448fd0>
```




```
chi2, p, dof, expected = chi2_contingency(ctc_hf)
print("chi2 = ", chi2)
print("p-val = ", p)
print("degree of freedom = ",dof)
```

```
chi2 =  31.040651744510726
p-val =  2.5268037522937072e-08
degree of freedom =  1
```

Chi-square Test of Association
As the p-value is greater than the threshold value of 0.05, we can say that haemoglobin counts of patients are associated with the prevalence of congestive heart failure.



Question 3:

Correlation between total number of chronic diseases a person has *versus* the number of days in hospitalization and ICU.

Approach

Correlation between total number of chronic diseases a person has versus the number of days in hospitalization and ICU.

```python
# Subset Data
cd = new_data[['renal_flg', 'liver_flg', 'cad_flg', 'resp_flg', 'sepsis_flg'\
              , 'chf_flg', 'afib_flg', 'copd_flg', 'stroke_flg', 'mal_flg', 'hospital_los_day', 'icu_los_day']]

# Sum up the total of CD for each row
cd['cd_total'] = cd['renal_flg'] + cd['liver_flg'] + cd['cad_flg'] + cd['resp_flg'\
                 ] + cd['sepsis_flg'] + cd['chf_flg'] + cd['afib_flg'] + cd['copd_flg'\
                 ] + cd['stroke_flg'] + cd['mal_flg']

# add colun called status that will be used ofr futher analysis
col         = 'cd_total'
conditions  = [cd[col] == 6, cd[col] == 5, cd[col] == 4, cd[col] == 3, cd[col] == 2, cd[col] == 1]
values      = ['six', 'five', 'four', 'three', 'two', 'one']

cd['status'] = np.select(conditions, values)

cd.sample(10)
```

1

2

	renal_flg	liver_flg	cad_flg	resp_flg	sepsis_flg	chf_flg	afib_flg	copd_flg	stroke_flg	mal_flg	hospital_los_day	icu_los_day	cd_total	status
1159	0	0	0	0	0	0	0	0	0	0	6	3.80	0	0
889	0	0	0	0	0	0	1	0	1	0	11	0.93	2	two
285	0	0	0	0	0	0	0	0	0	0	3	1.31	0	0

- **Step 1**
  Data Preparation: Subset the required data to the new dataset.
- **Step 2**
  Create a separate column with the total number of chronic diseases each patient have.
- **Step 3**
  Create table to view the summary of the 'status' column versus length of stay.
- **Step 4**
  Create the heatmap (from seaborn library)
  Perform statistical test (scipy.stats)

TABULATION

status	renal_flg	liver_flg	cad_flg	resp_flg	sepsis_flg	chf_flg	afib_flg	copd_flg	stroke_flg	mal_flg	hospital_los_day	icu_los_day	cd_total
0	0	0	0	0	0	0	0	0	0	0	4780	1705.09	0
five	7	1	9	15	0	19	16	14	2	12	222	82.61	95
four	15	3	22	34	0	38	30	25	12	13	419	172.11	192
one	3	37	15	236	0	16	20	20	84	93	4776	1988.31	524
six	4	0	4	4	0	3	3	2	1	3	38	9.79	24
three	12	11	30	93	0	69	55	48	25	38	1192	576.76	381
two	15	46	34	164	0	60	67	44	78	90	2674	1291.41	598

The LOS days seem to be greater when less number of chronic diseases.

Rationale: The greater the chronic diseases, the more critical the patient condition might be leading to mortality. And lesser the number of chronic diseases, more medical intervention the patient must be receiving adding up to greater los.

```
hospital_day = smf.ols('hospital_los_day ~ status', cd).fit()
hospital_day.summary()
```

OLS Regression Results

Dep. Variable:	hospital_los_day	R-squared:	0.015
Model:	OLS	Adj. R-squared:	0.012
Method:	Least Squares	F-statistic:	4.342
Date:	Thu, 01 Dec 2022	Prob (F-statistic):	0.000234
Time:	02:50:58	Log-Likelihood:	-5946.0
No. Observations:	1690	AIC:	1.191e+04
Df Residuals:	1683	BIC:	1.194e+04
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	7.1450	0.316	22.597	0.000	6.525	7.765
status[T.five]	4.5392	1.903	2.386	0.017	0.807	8.271
status[T.four]	1.5842	1.222	1.296	0.195	-0.813	3.981
status[T.one]	1.9695	0.477	4.128	0.000	1.034	2.905
status[T.six]	2.3550	4.101	0.574	0.566	-5.689	10.399
status[T.three]	2.2408	0.792	2.831	0.005	0.688	3.793
status[T.two]	1.7982	0.569	3.161	0.002	0.682	2.914

Omnibus:	1540.381	Durbin-Watson:	2.024
Prob(Omnibus):	0.000	Jarque-Bera (JB):	80707.463
Skew:	4.108	Prob(JB):	0.00
Kurtosis:	35.843	Cond. No.	22.1

Null Hypotheses:

The number of chronic disease is associated with the length of stay in Hospital.

Alternate Hypothesis:

The number of chronic disease is associated with the length of stay in Hospital.

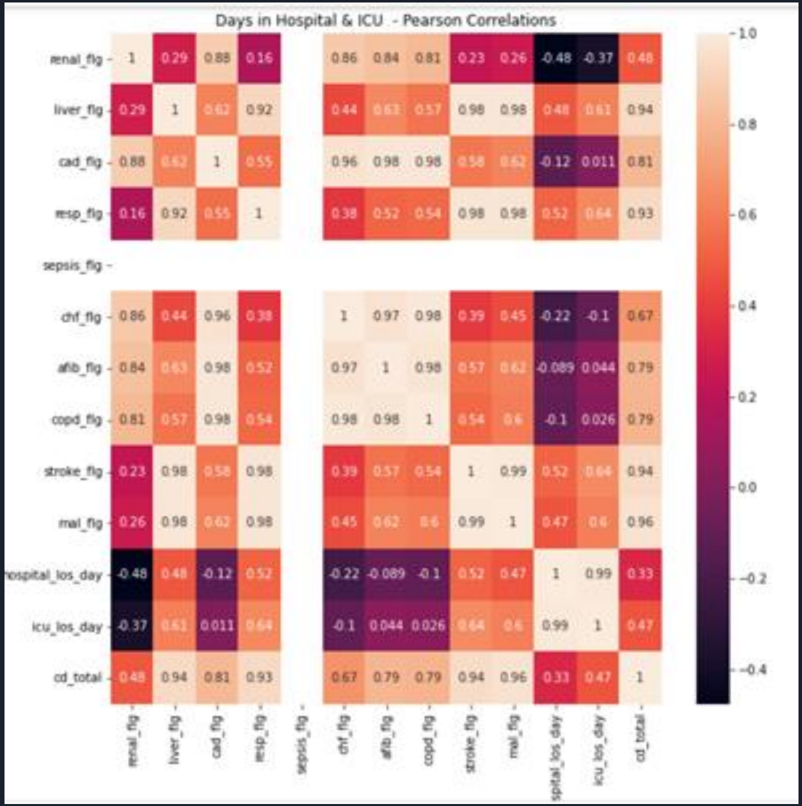
Test results:

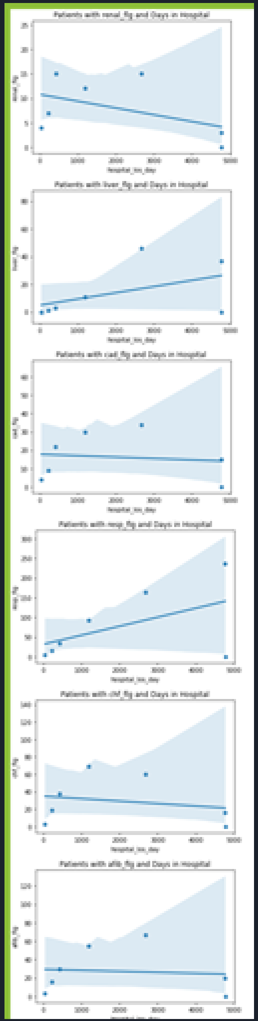
P is greater than threshold, depicting that we are inclined towards accepting the Null Hypotheses.

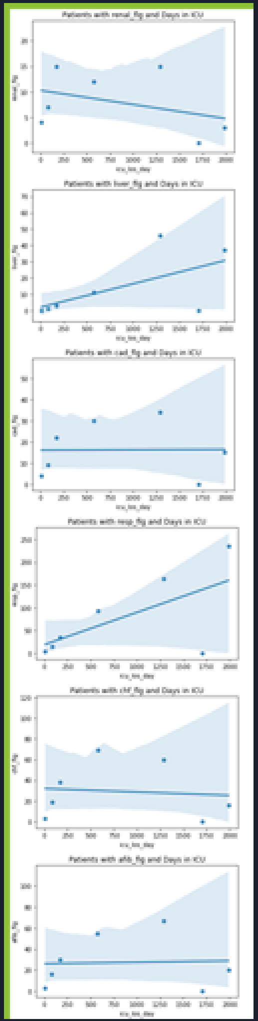
Inference:

The number of chronic disease is associated with the length of stay in Hospital.

VISUAL REPRESENTATION
Each Chronic condition

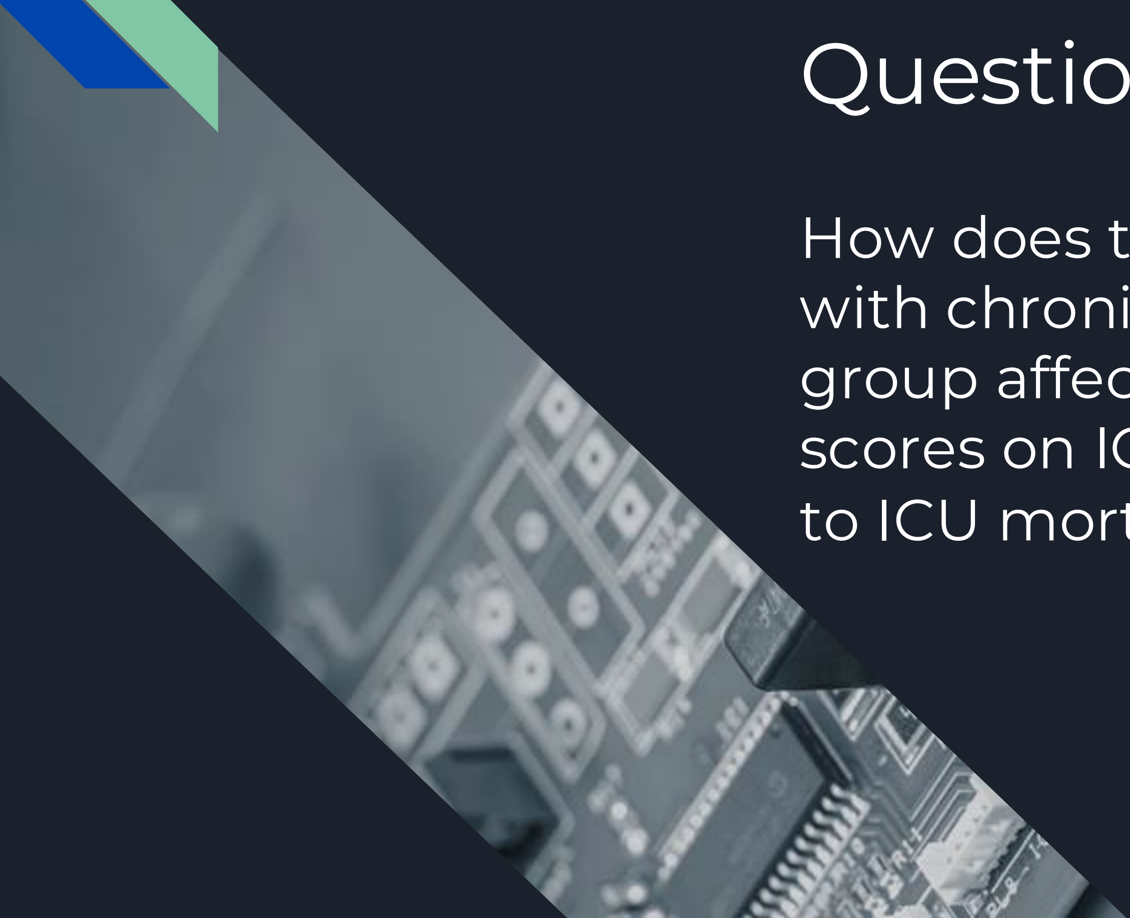






Question 4:

How does the number of patients with chronic diseases in each age group affect the SAPS and SOFA scores on ICU admission leading to ICU mortality.



Approach

SOFA Score

1

```python
q4['age'] = q4['age'].round().astype(int)

q4['age_category'] = pd.cut(x=q4['age'], bins=[0, 39, 59, 69, 74, 79, 100],
                    labels=['<40', '40-59', '60-69', '70-74', '75-79', '>80'])

q4.loc[q4['icu_exp_flg'] == 1, 'icu_exp_flg'] = 'dead'
q4.loc[q4['icu_exp_flg'] == 0, 'icu_exp_flg'] = 'alive'

# casting some columns to calculate sofa score
conditions  = [q4['creatinine_first'] < 1.2, q4['creatinine_first'] < 2, q4['creatinine_first'] < 3.5, q4['creatinine_first'] < 5, q4['creatinine_first'] >= 5]
values      = [0, 1, 2, 3, 4]
q4['renal_factor'] = np.select(conditions, values)

conditions1  = [q4['po2_first'] <= 100, q4['po2_first'] <= 200, q4['po2_first'] <= 300, q4['po2_first'] <= 400, q4['po2_first'] > 400]
values1      = [4, 3, 2, 1, 0]
q4["pafi_factor"] = np.select(conditions1, values1)

conditions2  = [q4['platelet_first'] < 20, q4['platelet_first'] < 50, q4['platelet_first'] < 100, q4['platelet_first'] < 150, q4['platelet_first'] > 150]
values2      = [4, 3, 2, 1, 0]
q4["platelets_factor"] = np.select(conditions2, values2)
```

2

```python
# SOFAScore = PaO2/FIO2Factor + PlateletsFactor + TotalBilirubinFactor + BloodPressure + GlasgowComaScoreFactor + RenalFacto
q4['predicted_sofa'] = q4['renal_factor'] + q4['pafi_factor'] + q4['platelets_factor']
```

3

```python
sofa_table = new_mortality.groupby('age_category')[['sofa_first', 'predicted_sofa']].mean().reset_index()
max = new_mortality['predicted_sofa'].max()
min = new_mortality['predicted_sofa'].min()
range = f'{min}-{max}'
sofa_table['range'] = range
sofa_table
```

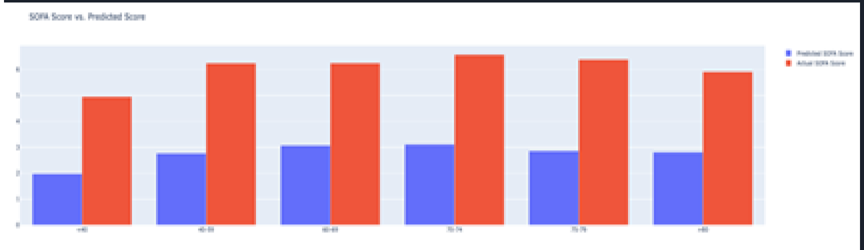
4

```python
import plotly.graph_objects as go
fig = go.Figure(data=[
    go.Bar(name='Predicted SOFA Score', x=sofa_table['age_category'], y=sofa_table['predicted_sofa']),
    go.Bar(name='Actual SOFA Score', x=sofa_table['age_category'], y=sofa_table['sofa_first'])
])
# Change the bar mode
fig.update_layout(barmode='group')
fig.update_layout(title='SOFA Score vs. Predicted Score',
                  yaxis_zeroline=False, xaxis_zeroline=False)
fig.show()
```

- **Step 1**
  Data Preparation: Subset the required data, cast some columns for SOFA predicting score.
- **Step 2**
  Calculate SOFA predicted score.
- **Step 3**
  Create table to view the summary of the predicted score compared to actual score.
- **Step 4**
  Create the barplot (from plotly library)

SOFA Table

	age_category	sofa_first	predicted_sofa	mortality_rate	range
0	<40	4.951754	1.973684	0.014166	0-10
1	40-59	6.243043	2.760668	0.017714	0-10
2	60-69	6.246377	3.067633	0.021297	0-10
3	70-74	6.573643	3.108527	0.024248	0-10
4	75-79	6.383929	2.857143	0.030644	0-10
5	>80	5.909091	2.805785	0.028226	0-10



	mortality_rate	predicted_sofa	sofa_first
age_category			
<40	6.472942	1.946058	4.891892
40-59	9.565094	2.737030	6.189964
60-69	4.439653	3.051163	6.253521
70-74	3.150803	3.091603	6.546154
75-79	3.461376	2.727273	6.308333
>80	6.830629	2.660448	5.805970

Approach

SAPS Score

```python
q4['age'] = q4['age'].round().astype(int)

q4['age_category'] = pd.cut(x=q4['age'], bins=[0, 39, 59, 69, 74, 79, 100],
                            labels=['<40', '40-59', '60-69', '70-74', '75-79', '>80'])

q4.loc[q4['icu_exp_flg'] == 1, 'icu_exp_flg'] = 'dead'
q4.loc[q4['icu_exp_flg'] == 0, 'icu_exp_flg'] = 'alive'
```

1

```python
def ICU_mortality_rate(row):
    #https://www.omnicalculator.com/health/saps-ii
    x = -7.7631 + 0.0737 * row['sapsi_first'] + 0.9971 * np.log(row['sapsi_first'] + 1)
    mortality = math.exp(X)/(1 + math.exp(X))
    return mortality

# add mortality rate calculated with saps score
q4['mortality_rate'] = q4.apply(ICU_mortality_rate, axis=1)
```

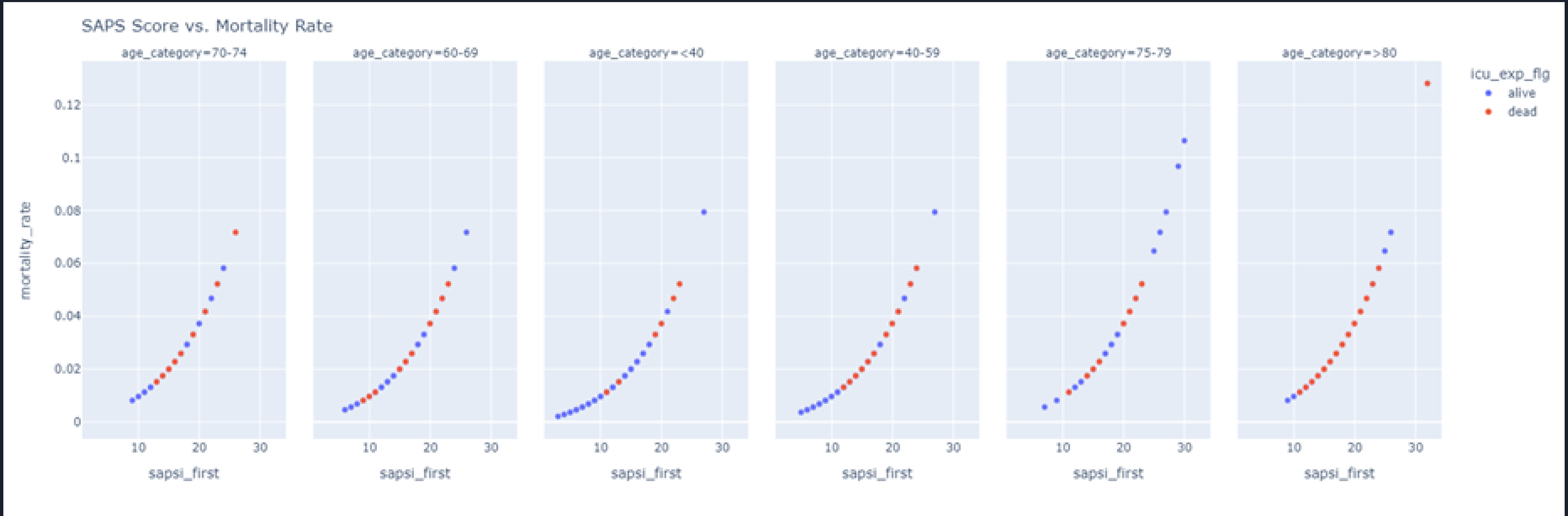
2

```python
fig = px.scatter(q4, x="sapsi_first", y="mortality_rate", color="icu_exp_flg", facet_col="age_category")
fig.update_layout(title='SAPS Score vs. Mortality Rate',
                  yaxis_zeroline=False, xaxis_zeroline=False)
fig.show()
```

3

- **Step 1**
  Data Preparation: Subset the required data.
- **Step 2**
  Made function to create mortality rate using math library.
- **Step 3**
  Create the scatter plot (from plotly library)

For the age category greater than 80 years of age, there are more data points for dead than alive people . And for the dead people, the SAPS score varies roughly linearly as the mortality rate.



SAPS v/s SOFA

age_category	mortality_rate	sofa_first
<40	6.472942	4.891892
40-59	9.565094	6.189964
60-69	4.439653	6.253521
70-74	3.150803	6.546154
75-79	3.461376	6.308333
>80	6.830629	5.805970

age_category	mortality_rate	sapsi_first
<40	6.472942	11.886214
40-59	9.565094	13.359259
60-69	4.439653	14.708134
70-74	3.150803	15.900000
75-79	3.461376	17.610619
>80	6.830629	17.057851

Various studies suggest that SAPS score is more reliable in predicting mortality as compared with SOFA score.

Question 5:

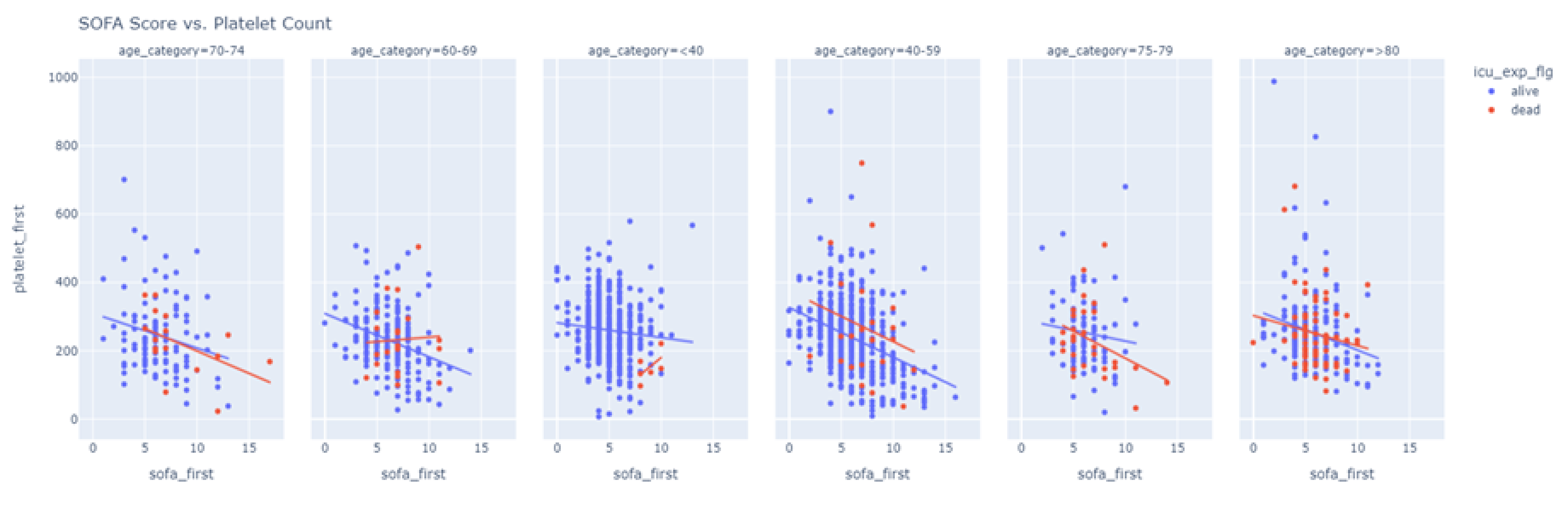
Impact on SOFA score due to platelet count.

APPROACH

```
fig = px.scatter(q4, x="sofa_first", y="platelet_first", color="icu_exp_flg", facet_col="age_category", trendline="ols")
fig.update_layout(title='SOFA Score vs. Platelet Count',
                  yaxis_zeroline=False, xaxis_zeroline=False)
fig.show()
```

Extracting the required data with platelet count and preparing it to facet on the basis of age.

In the age category 60 to 69 years of age, for alive patients, there is negative correlation between SOFA score and platelets count;
for dead patients, there is positive correlation between SOFA score and platelets count.



CONCLUSION

- Overall Inference
- Learnings from the class
- Challenges faced and strategies to overcome

Inference

The mortality and length of stay seems to be impacted by various physiological and anatomical factors on the patients with prevalence of chronic diseases having history of sepsis or requiring mechanical ventilation on first day of ICU admission.

It briefly highlights the overall impact on SAPS and SOFA score under such conditions, where patient is on Indwelling Arterial Catheter when in ICU, and existence of such conditions help in predicting mortality based on the SAPS score or SOFA.

Learnings from Class

- Data Cleaning
- Data Extraction
- Data Manipulation
- Research tactics
- Understanding of Stakeholders
- Project Management strategies and deliverables planning
- Team work
- Health Data Analytics
- Problem-solving capabilities
- Utilize data to answer vital health questions.

Challenges Faced & Strategies to Overcome

Dataset selection

Make sure in advance that the objectives are answerable and of value to the community

Selection of Visualization

Understand the datatype to figure out the best fit visualization, and match it with the ease of inference from that visual.

Understanding new terminologies

Intensive use of web and experiment with MDCalc to dig deeper into new terminologies.

Analysis strategy

Do not hesitate to ask questions from your boss/mentor. Here, Prof. Neha Bhomia, thanks for the guidance.

Motivation

Motivate your team member to keep at it, even if data answers something unexpected.

Thank you!