Business Intelligence (CIS 5270)

Project-2: Python Project

Topic: Unraveling Insights into Heart Failure: A Business Intelligence Approach

Team members: Nagender Chauhan & Lekha Ajit Kumar

INTRODUCTION

In an era where data is abundant, businesses and industries are constantly seeking innovative ways to leverage this valuable resource to gain insights and make informed decisions. In the field of healthcare, where the stakes are high and lives are at risk, business intelligence plays a vital role in unraveling complex patterns and extracting meaningful information. This project delves into the realm of business intelligence to analyze a comprehensive dataset focused on heart failure, aiming to uncover valuable insights and contribute to the advancements in cardiovascular care.

Heart failure is a pressing global health issue, affecting millions of people worldwide. Understanding the factors and characteristics that contribute to the development and progression of heart failure is crucial for effective prevention, early detection, and management strategies. This project centers around a rich dataset encompassing various attributes, ranging from demographic information to clinical measurements, designed to shed light on the multifaceted nature of heart failure.

The dataset includes essential fields such as Age, Height, Sex, Smoke, ChestPainType, RestingBP, ATA, Cholesterol, FastingBS, RestingECG, MaxHR, ExerciseAngina, Oldpeak, ST_Slope, and HeartDisease. By harnessing the power of business intelligence, we aim to extract meaningful insights from this data, unraveling hidden correlations, and identifying key predictors and risk factors associated with heart failure.

Through our meticulous analysis, we seek to answer critical questions such as:

What is the relationship between age, sex, and heart failure? Are there specific age groups or gender disparities that contribute to a higher risk of heart failure?

How do lifestyle factors like smoking and exercise-induced angina influence the development of heart failure? Are there discernible patterns or associations?

Can clinical measurements such as resting blood pressure, cholesterol levels, and maximum heart rate provide valuable predictive indicators for heart failure?

What role do variables like chest pain type, resting electrocardiogram (ECG), and ST slope play in understanding the pathophysiology of heart failure?

By applying sophisticated business intelligence techniques, statistical analysis, and machine learning algorithms, we aim to uncover significant correlations, detect patterns, and build predictive models to enhance our understanding of heart failure. The insights gained from this project have the potential to drive evidence-based decision-making, inform public health policies, and guide personalized patient care strategies.

We recognize the urgency of addressing heart failure as a critical public health issue, and this project endeavors to contribute to the ongoing efforts in reducing the burden of this condition. We strive to provide actionable insights that can support healthcare professionals, policymakers, and stakeholders in their mission to prevent, detect, and manage heart failure effectively.

Keywords: heart failure, business intelligence, data analysis, insights, predictors, risk factors, public health, personalized care.

References:

1. Title: Evaluating risk prediction models for adults with heart failure: A systematic literature review.

   Authors: Di Tanna GL, Wirtz H, Burrows KL, Globe G

   Year: 2020 Jan 15

   URL:https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6961879/

2. Title: Predicting Heart Failure With Preserved and Reduced Ejection Fraction

   Authors:Jennifer E. Ho, Danielle Enserro, Frank P. Brouwers

   Year: 2016 June

   URL:https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6961879/

DATA DESCRIPTION

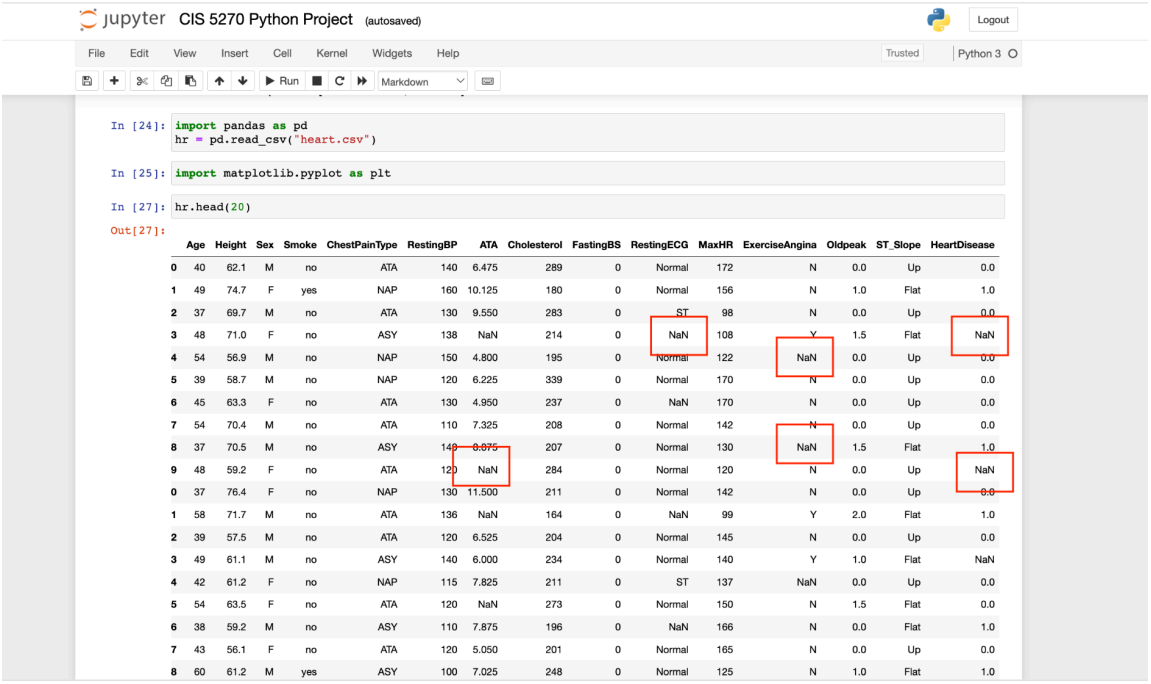
Data set link: https://www.kaggle.com/datasets/rahulsharma51/heart-failure-lagacy-dataset

Column	**Description**	**Example Value**
Age	age of the patient [years]	Ex: 36
sex	sex of the patient	Ex: [M: Male, F: Female]
Height	height of the patient in inches	Ex: 62.1
Smoke	does the person smoke or not	Ex: [Y: Yes, N: No]
ChestPainType	chest pain type [TA: Typical Angina, ATA: Atypical Angina,	Ex: TA

	NAP: Non-Anginal Pain, ASY: Asymptomatic]	
RestingBP	resting blood pressure [mm Hg]	Ex: 130
Cholesterol	serum cholesterol [mm/dl]	Ex: 180
FastingBS	fasting blood sugar	Ex: 0 or 1
ATA	maximum heart ATA	Ex: 4.8
RestingECG	resting electrocardiogram results [Normal: Normal, ST:abnormality]	Ex: Normal
MaxHR	maximum heart rate	Ex: 170
ExerciseAngina	exercise-induced angina [Y: Yes, N: No]	Ex: No
Oldpeak	ST [Numeric value measured in depression]	Ex: 1.5
ST_Slope	the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]	Ex: flat
HeartDisease	output class [1: heart disease, 0: Normal]	Ex: 0 or 1

DATA CLEANING

1. Finding the total number of null values in the data set



jupyter CIS 5270 Python Project (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Run Markdown

```
In [24]: import pandas as pd
         hr = pd.read_csv("heart.csv")

In [25]: import matplotlib.pyplot as plt

In [27]: hr.head(20)
```

Out[27]:

	Age	Height	Sex	Smoke	ChestPainType	RestingBP	ATA	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	62.1	M	no	ATA	140	6.475	289	0	Normal	172	N	0.0	Up	0.0
1	49	74.7	F	yes	NAP	160	10.125	180	0	Normal	156	N	1.0	Flat	1.0
2	37	69.7	M	no	ATA	130	9.550	283	0	ST	98	N	0.0	Up	0.0
3	48	71.0	F	no	ASY	138	NaN	214	0	NaN	108	Y	1.5	Flat	NaN
4	54	56.9	M	no	NAP	150	4.800	195	0	Normal	122	NaN	0.0	Up	0.0
5	39	58.7	M	no	NAP	120	6.225	339	0	Normal	170	N	0.0	Up	0.0
6	45	63.3	F	no	ATA	130	4.950	237	0	NaN	170	N	0.0	Up	0.0
7	54	70.4	M	no	ATA	110	7.325	208	0	Normal	142	N	0.0	Up	0.0
8	37	70.5	M	no	ASY	140	8.875	207	0	Normal	130	NaN	1.5	Flat	1.0
9	48	59.2	F	no	ATA	120	NaN	284	0	Normal	120	N	0.0	Up	NaN
0	37	76.4	F	no	NAP	130	11.500	211	0	Normal	142	N	0.0	Up	0.0
1	58	71.7	M	no	ATA	136	NaN	164	0	NaN	99	Y	2.0	Flat	1.0
2	39	57.5	M	no	ATA	120	6.525	204	0	Normal	145	N	0.0	Up	0.0
3	49	61.1	M	no	ASY	140	6.000	234	0	Normal	140	Y	1.0	Flat	NaN
4	42	61.2	F	no	NAP	115	7.825	211	0	ST	137	NaN	0.0	Up	0.0
5	54	63.5	F	no	ATA	120	NaN	273	0	Normal	150	N	1.5	Flat	0.0
6	38	59.2	M	no	ASY	110	7.875	196	0	NaN	166	N	0.0	Flat	1.0
7	43	56.1	F	no	ATA	120	5.050	201	0	Normal	165	N	0.0	Up	0.0
8	60	61.2	M	yes	ASY	100	7.025	248	0	Normal	125	N	1.0	Flat	1.0

Running this line of code to find the following results

```
hr.isnull().sum()
```

```
In [4]: #data cleaning 1

        hr.isnull().sum()

Out[4]: Age              0
        Height           0
        Sex              0
        Smoke            0
        ChestPainType    0
        RestingBP        0
        ATA              4
        Cholesterol      0
        FastingBS        0
        RestingECG       4
        MaxHR            0
        ExerciseAngina   3
        Oldpeak          0
        ST_Slope         0
        HeartDisease     3
        dtype: int64
```

The total number of null values in each row is being totalled and the sum of total values is displayed in a tabulated manner.

2. Changing the values in the column sex. Changing M to Male and F to Female for better readability and understanding of data.

Height	Sex	Smoke	ChestPainType	F
62.1	M	no	ATA	
74.7	F	yes	NAP	
69.7	M	no	ATA	
71.0	F	no	ASY	
56.9	M	no	NAP	
58.7	M	no	NAP	
63.3	F	no	ATA	
70.4	M	no	ATA	
70.5	M	no	ASY	
59.2	F	no	ATA	
76.4	F	no	NAP	
71.7	M	no	ATA	
57.5	M	no	ATA	
61.1	M	no	ASY	
61.2	F	no	NAP	
63.5	F	no	ATA	
59.2	M	no	ASY	

Using the following code we replaced the values

```
hr.Sex.replace( { "M" : "Male", "F" :"Female"} , inplace = True)
```

```
#datacleaning 3
hr['ATA'].fillna(hr['ATA'].mea

hr.head(5)
```

	Age	Height	Sex	Smoke	ChestP
0	40	62.1	Male	no	
1	49	74.7	Female	yes	
2	37	69.7	Male	no	
3	48	71.0	Female	no	
4	54	56.9	Male	no	

3. The ATA column has null values. Here we tried to fill to take the mean of the entire ATA column and then fill all the null values with the the mean of ATA.

ChestPainType	RestingBP	ATA	Cholesterol	FastingBS	RestingECG
ATA	140	6.475	289	0	Normal
NAP	160	10.125	180	0	Normal
ATA	130	9.550	283	0	ST
ASY	138	NaN	214	0	NaN
NAP	150	4.800	195	0	Normal
NAP	120	6.225	339	0	Normal
ATA	130	4.950	237	0	NaN
ATA	110	7.325	208	0	Normal
ASY	140	8.875	207	0	Normal
ATA	120	NaN	284	0	Normal
NAP	130	11.500	211	0	Normal
ATA	136	NaN	164	0	NaN
ATA	120	6.525	204	0	Normal
ASY	140	6.000	234	0	Normal

Using the following code we were able to achieve these results.

```
hr['ATA'].fillna(hr['ATA'].mean(),inplace = True )
```

ChestPainType	RestingBP	ATA	Cholesterol	FastingBS	RestingECG	MaxHR	Ex
ATA	140	6	289	0	Normal	172	
NAP	160	10	180	0	Normal	156	
ATA	130	9	283	0	ST	98	
ASY	138	7	214	0	NaN	108	
NAP	150	4	195	0	Normal	122	

4. After filling the ATA columns with the mean value. The numbers datatype in ATA column became double. We wanted to convert these double-precision numbers into simple integer numbers for more convenient readability.

Sex	Smoke	ChestPainType	RestingBP	ATA	Cholesterol	FastingBS	RestingECG	M
Male	no	ATA	140	6.475000	289	0	Normal	
Female	yes	NAP	160	10.125000	180	0	Normal	
Male	no	ATA	130	9.550000	283	0	ST	
Female	no	ASY	138	7.827551	214	0	NaN	
Male	no	NAP	150	4.800000	195	0	Normal	
Male	no	NAP	120	6.225000	339	0	Normal	
Female	no	ATA	130	4.950000	237	0	NaN	
Male	no	ATA	110	7.325000	208	0	Normal	
Male	no	ASY	140	8.875000	207	0	Normal	
Female	no	ATA	120	7.827551	284	0	Normal	

Using the following code we were able to achieve this result.

```
hr["ATA"]=hr["ATA"].astype('int64')
```

ChestPainType	RestingBP	ATA	Cholesterol	FastingBS	RestingECG	M
ATA	140	6	289	0	Normal	
NAP	160	10	180	0	Normal	
ATA	130	9	283	0	ST	
NAP	120	6	339	0	Normal	
ATA	110	7	208	0	Normal	
NAP	130	11	211	0	Normal	
ATA	120	6	204	0	Normal	
ATA	120	7	273	0	Normal	
ATA	120	5	201	0	Normal	
ASY	100	7	248	0	Normal	

You can also the datatype of the ATA column has been changed. It was previously double and now it has changed to a 64-bit integer.

```
print(hr.dtypes)
```

```
Age                int64
Height           float64
Sex               object
Smoke             object
ChestPainType     object
RestingBP          int64
ATA                int64
Cholesterol        int64
FastingBS          int64
RestingECG        object
MaxHR              int64
ExerciseAngina    object
Oldpeak          float64
ST_Slope          object
HeartDisease     float64
dtype: object
```

5. After cleaning all the these columns, we still had a few columns with null values as seen in the picture bellow. These columns had a string value inside them. Hence we could not fill it with mean. So we decide to drop all the remaining rows which had null vales.

;	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
)	Normal	172	N	0.0	Up	0.0
)	Normal	156	N	1.0	Flat	1.0
)	ST	98	N	0.0	Up	0.0
)	NaN	108	Y	1.5	Flat	NaN
)	Normal	122	NaN	0.0	Up	0.0
)	Normal	170	N	0.0	Up	0.0
)	NaN	170	N	0.0	Up	0.0
)	Normal	142	N	0.0	Up	0.0
)	Normal	130	NaN	1.5	Flat	1.0
)	Normal	120	N	0.0	Up	NaN

We were able to achieve this objective with the following code .

```
hr.dropna(inplace=True)
```

After dropping the null values this what the output looked like. You can see that there are no null values in the entire dataset.

	Age	Height	Sex	Smoke	ChestPainType	RestingBP	ATA	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisea
0	40	62.1	Male	no	ATA	140	6	289	0	Normal	172	N	0.0	Up	
1	49	74.7	Female	yes	NAP	160	10	180	0	Normal	156	N	1.0	Flat	
2	37	69.7	Male	no	ATA	130	9	283	0	ST	98	N	0.0	Up	
5	39	58.7	Male	no	NAP	120	6	339	0	Normal	170	N	0.0	Up	
7	54	70.4	Male	no	ATA	110	7	208	0	Normal	142	N	0.0	Up	
10	37	76.4	Female	no	NAP	130	11	211	0	Normal	142	N	0.0	Up	
12	39	57.5	Male	no	ATA	120	6	204	0	Normal	145	N	0.0	Up	
15	54	63.5	Female	no	ATA	120	7	273	0	Normal	150	N	1.5	Flat	
17	43	56.1	Female	no	ATA	120	5	201	0	Normal	165	N	0.0	Up	
18	60	61.2	Male	yes	ASY	100	7	248	0	Normal	125	N	1.0	Flat	
19	36	70.6	Male	no	ATA	120	9	267	0	Normal	160	N	3.0	Flat	
20	43	57.3	Female	no	TA	100	3	223	0	Normal	142	N	0.0	Up	
21	44	59.7	Male	no	ATA	120	5	184	0	Normal	142	N	1.0	Flat	
22	49	72.4	Female	no	ATA	124	10	201	0	Normal	164	N	0.0	Up	
23	44	68.0	Male	no	ATA	150	8	288	0	Normal	150	Y	3.0	Flat	
24	40	65.7	Male	no	NAP	130	9	215	0	Normal	138	N	0.0	Up	
25	36	61.3	Male	no	NAP	130	8	209	0	Normal	178	N	0.0	Up	
26	53	60.7	Male	no	ASY	124	6	260	0	ST	112	Y	3.0	Flat	
27	52	65.6	Male	no	ATA	120	9	284	0	Normal	118	N	0.0	Up	
28	53	48.7	Female	no	ATA	113	1	468	0	Normal	127	N	0.0	Up	

We also wanted to know how many rows were reduced before and after the dropna method. We used the following code to achieve this objective.

```
#data cleaning 5
print("before cleaning rows count:",len(hr))
hr.dropna(inplace=True)
print("after cleaning rows count:",len(hr))
```

```
before cleaning rows count: 918
after cleaning rows count: 909
```

We can see that 9 rows were removed when we used the dropna method. Now we wanted to double-check and see if we have anymore null values so we ran this following code again.

```
: hr.isnull().sum()
```

```
: Age               0
  Height            0
  Sex               0
  Smoke             0
  ChestPainType     0
  RestingBP         0
  ATA               0
  Cholesterol       0
  FastingBS         0
  RestingECG        0
  MaxHR             0
  ExerciseAngina    0
  Oldpeak           0
  ST_Slope          0
  HeartDisease      0
  dtype: int64
```

Now we can see that are no null values in our dataset and it has been cleaned completely and is ready to use for the purpose of analysis and visualizations.

SUMMARY STATISTICS

Statistical data can provide valuable insights into the likelihood of heart diseases and help predict their occurrence in individuals. There are several reasons why statistical data is essential for predicting heart diseases.

Identify risk factors: Statistical data can help identify risk factors for heart diseases, such as high blood pressure, cholesterol, smoking, obesity, and diabetes. By analyzing the data, we can determine the probability of a person developing heart disease based on their risk factors.

By analyzing statistical data, we can identify trends and patterns that can help early analysis of heart diseases. For example, data may reveal that certain populations are at a higher risk of heart disease, which can help target interventions and resources for those groups.

Overall, statistical data plays a crucial role in predicting heart diseases by providing insights into risk factors.

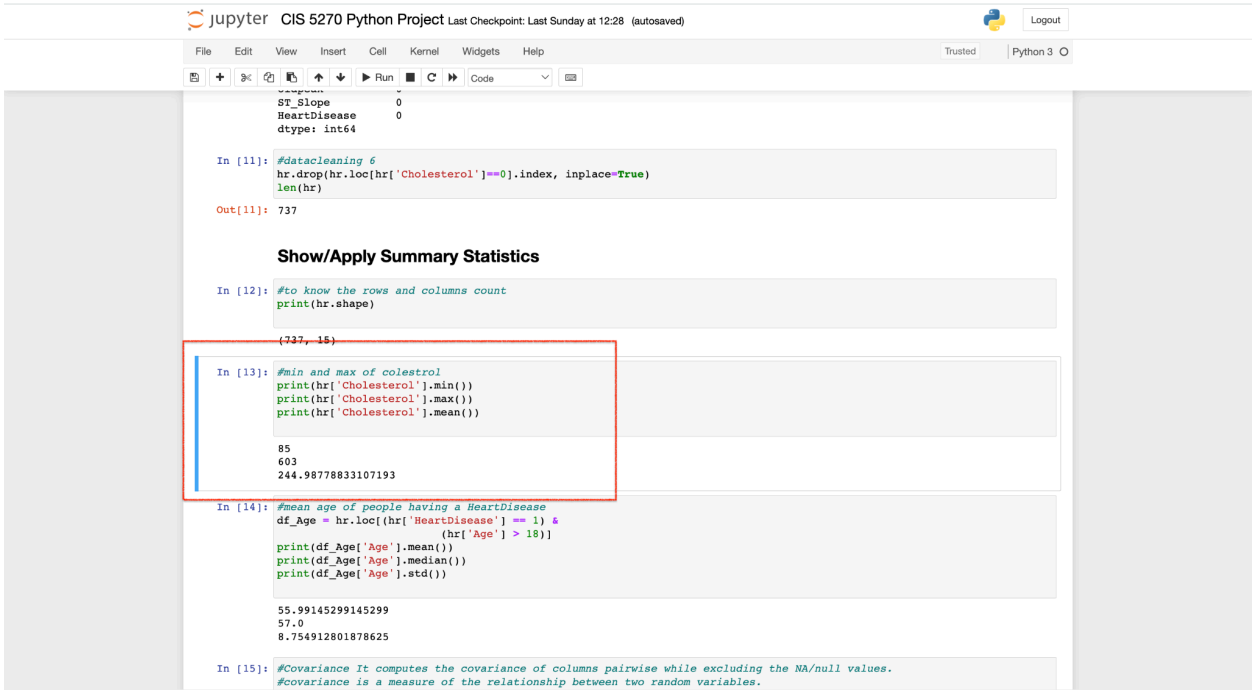
We have used 5 Statistical analysis methods to get meaningful insights.

1. Minimum, Maximum and Mean.

We have found the minimum, maximum and mean of the cholesterol levels in the dataset.

Cholesterol is a type of fat that is necessary for the proper functioning of our body, but too much of it can increase the risk of heart disease.

We found the following results using this code.

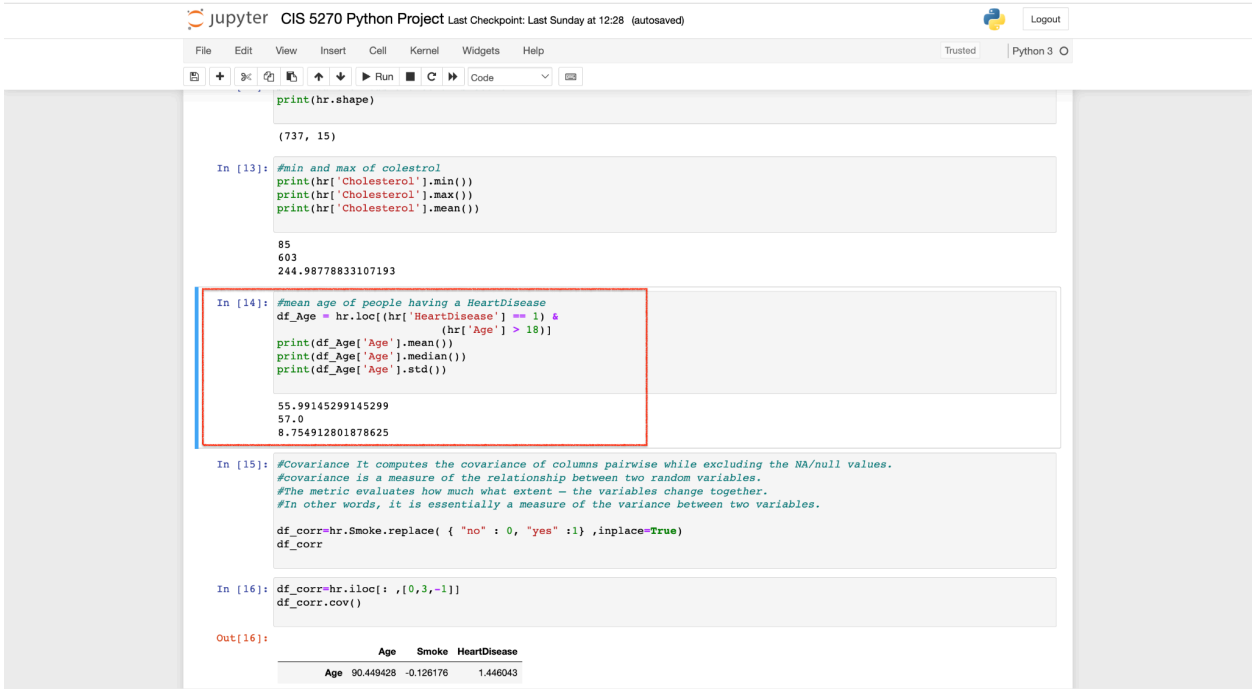



```
print(hr['Cholesterol'].min())
print(hr['Cholesterol'].max())
print(hr['Cholesterol'].mean())
```

```
85
603
244.98778833107193
```

2. Mean, Median and Standard deviation.

We wanted to know what is the age at which people experienced heart disease. So we created a data frame where we filtered the results to when the person had an active history of heart disease and the person should be above age 18. By this we know the age group of the people who had a heart disease.




```
#mean age of people having a HeartDisease
df_Age = hr.loc[(hr['HeartDisease'] == 1) &
                                (hr['Age'] > 18)]
print(df_Age['Age'].mean())
print(df_Age['Age'].median())
print(df_Age['Age'].std())
```

```
55.99145299145299
57.0
8.754912801878625
```

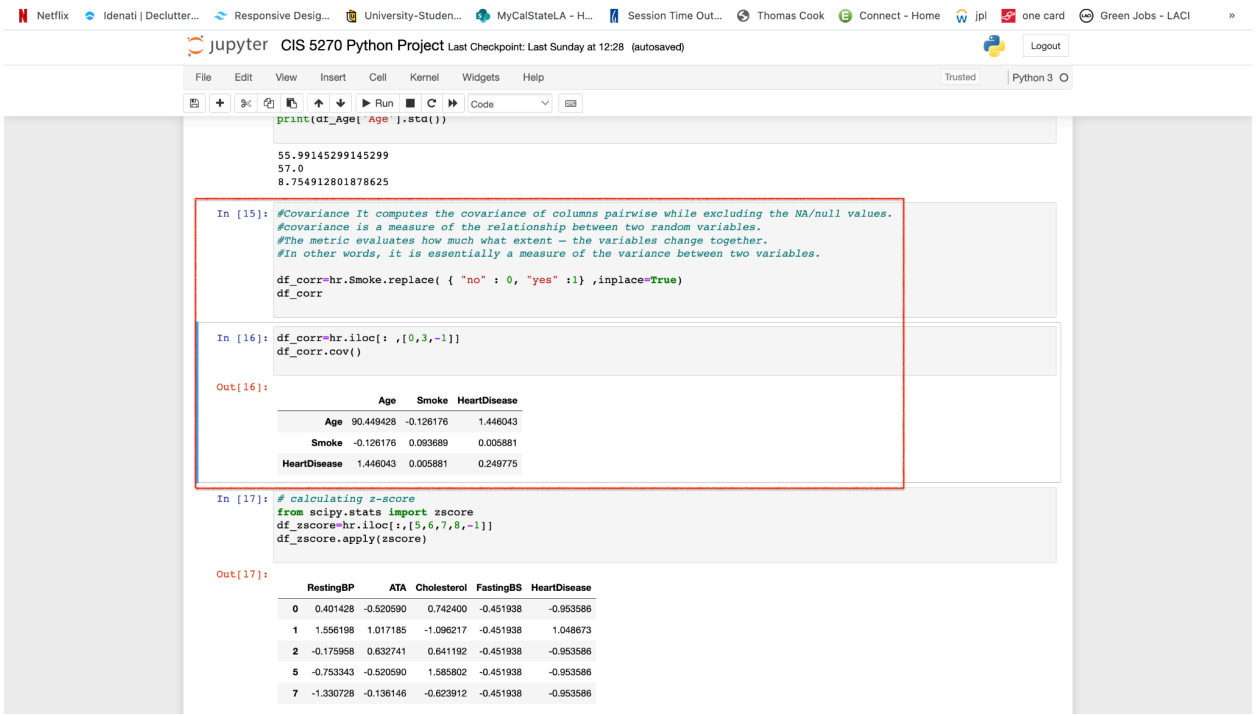
3. Covariance

Covariance computes the covariance of columns pairwise while excluding the NA/null values. Covariance is a measure of the relationship between two random variables. The metric evaluates how much and to what extent – the variables change together. In other words, it is essentially a measure of the variance between two variables.

We found the Covariance between three variables namely Age, Smoke and heart disease. By this, we know what is the relation or how much these three factors are interdependent on each other.

We first created a data frame where we only chose the three columns namely Age, Smoke and Herat diseases. Smoke column was a string variable and hence we could not apply this function to string variables so we changed it into a numeric value. After converting smoke values to integer values we put the three variables into a dataframe. We used the loc and iloc method to put it into a dataframe and then we called the cov()method.

This is the following results.



```
df_corr=hr.Smoke.replace( { "no" : 0, "yes" :1} ,inplace=True)
df_corr
```

In [16]:
```
df_corr=hr.iloc[: ,[0,3,-1]]
df_corr.cov()
```

Out[16]:

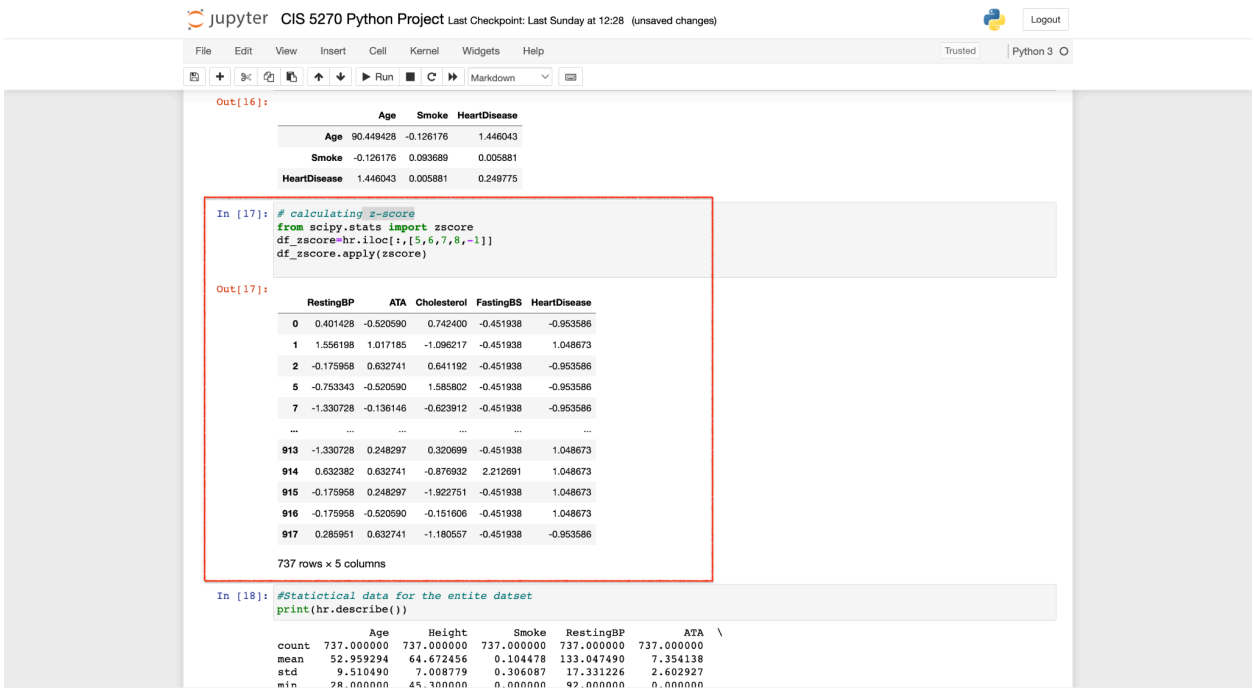
	Age	Smoke	HeartDisease
Age	90.449428	-0.126176	1.446043
Smoke	-0.126176	0.093689	0.005881
HeartDisease	1.446043	0.005881	0.249775

4. Z-score.

What is Z-score ? A Z-score, also known as a standard score, is a statistical measure that indicates how many standard deviations an observation or data point is from the mean of a population. It is a way of standardizing data to allow for meaningful comparisons between different sets of data.

A positive Z-score indicates that the value is above the mean of the population, while a negative Z-score indicates that the value is below the mean. A Z-score of zero means that the value is equal to the mean.

For example, if the mean cholesterol level of a population is 244, and the standard deviation is 100, a person has a cholesterol level of 444 would have a Z-score of 2, indicating that they are two standard deviations above the mean.




```
# calculating z-score
from scipy.stats import zscore
df_zscore=hr.iloc[:,[5,6,7,8,-1]]
df_zscore.apply(zscore)
```

	RestingBP	ATA	Cholesterol	FastingBS	HeartDisease
0	0.401428	-0.520590	0.742400	-0.451938	-0.953586
1	1.556198	1.017185	-1.096217	-0.451938	1.048673
2	-0.175958	0.632741	0.641192	-0.451938	-0.953586
5	-0.753343	-0.520590	1.585802	-0.451938	-0.953586
7	-1.330728	-0.136146	-0.623912	-0.451938	-0.953586
...	...	...	...	...	...
913	-1.330728	0.248297	0.320699	-0.451938	1.048673
914	0.632382	0.632741	-0.876932	2.212691	1.048673
915	-0.175958	0.248297	-1.922751	-0.451938	1.048673
916	-0.175958	-0.520590	-0.151606	-0.451938	1.048673
917	0.285951	0.632741	-1.180557	-0.451938	-0.953586

737 rows × 5 columns

5. Describe.

The describe() method in Pandas provides a summary of descriptive statistics for a data frame or series. This includes the count, mean, standard deviation, minimum, maximum, and quartile values for the data. It only calculates these statistics on the numeric datatype in the dataframe.

```
: #Statictical data for the entite datset
  print(hr.describe())
```

```
                Age      Height       Smoke   RestingBP         ATA  \
count   737.000000  737.000000  737.000000  737.000000  737.000000
mean     52.959294   64.672456    0.104478  133.047490    7.354138
std       9.510490    7.008779    0.306087   17.331226    2.602927
min      28.000000   45.300000    0.000000   92.000000    0.000000
25%      46.000000   59.900000    0.000000  120.000000    6.000000
50%      54.000000   65.300000    0.000000  130.000000    7.000000
75%      59.000000   69.400000    0.000000  140.000000    9.000000
max      77.000000   81.800000    1.000000  200.000000   14.000000

        Cholesterol   FastingBS       MaxHR     Oldpeak  HeartDisease
count    737.000000  737.000000  737.000000  737.000000    737.000000
mean     244.987788    0.169607  140.321574    0.904478      0.476255
std       59.323964    0.375542   24.531335    1.075616      0.499775
min       85.000000    0.000000   69.000000   -0.100000      0.000000
25%      208.000000    0.000000  122.000000    0.000000      0.000000
50%      238.000000    0.000000  140.000000    0.500000      0.000000
75%      275.000000    0.000000  160.000000    1.500000      1.000000
max      603.000000    1.000000  202.000000    6.200000      1.000000
```

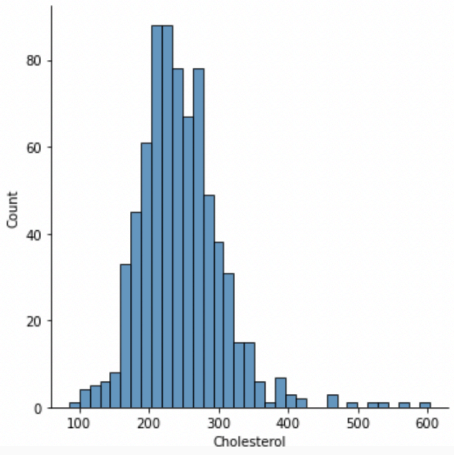
Chart 1

What is the rage of colestrol level of all the records in the data set?

```
In [19]: #Seaborn is a Python data visualization library based on matplotlib.
         #It provides a high-level interface for drawing attractive and informative statistical graphics
         import seaborn as sns
         plt.figure(figsize = (30, 30))
         sns.displot(hr['Cholesterol'])

         plt.show()
```

```
<Figure size 2160x2160 with 0 Axes>
```



Upon analyzing the graph, it becomes apparent that the distribution of cholesterol levels among individuals exhibits a bell-shaped or Gaussian-like pattern. The majority of individuals fall within the middle range of cholesterol levels, as evidenced by the tall bar in the center of the graph. This suggests a relatively balanced distribution, where the most common cholesterol levels are clustered around the average. Understanding this distribution allows healthcare professionals and policymakers to better allocate resources and design interventions aimed at promoting cardiovascular health in the population.

In conclusion, the bar graph depicting cholesterol levels versus count provides valuable insights into the distribution and characteristics of cholesterol profiles within the dataset. These insights

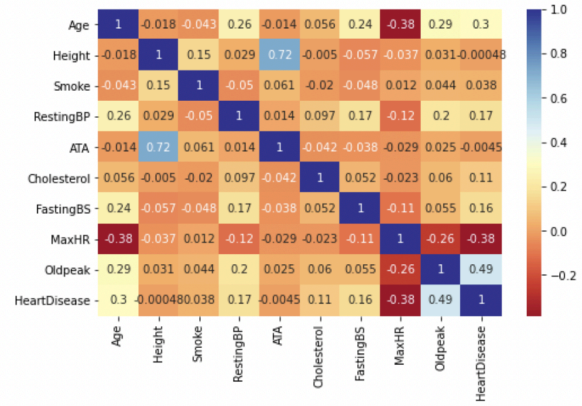
enable healthcare professionals, policymakers, and stakeholders to identify the prevailing cholesterol trends, pinpoint potential outliers, and design targeted interventions to improve cardiovascular health outcomes. By leveraging these insights, informed decisions can be made to promote public health initiatives, enhance preventive strategies, and ultimately contribute to a healthier population.

Chart 2

What factors influence the most and their featur importance in accordance to heart failure?

```
In [20]: #A heatmap is a graphical representation of data that uses a system of color-coding to represent different values
         plt.figure(figsize=(8,5))
         sns.heatmap(hr.corr(), annot=True, cmap='RdYlBu')
```

```
Out[20]: <AxesSubplot:>
```



The purpose of this map is to show the range of values in the data set and give a colour feel to it for better understanding.From the heat map we can see the darker colurs(dark orange and red) shows the high values and the lighter colours(dark blue and light blue show low values of the data set.

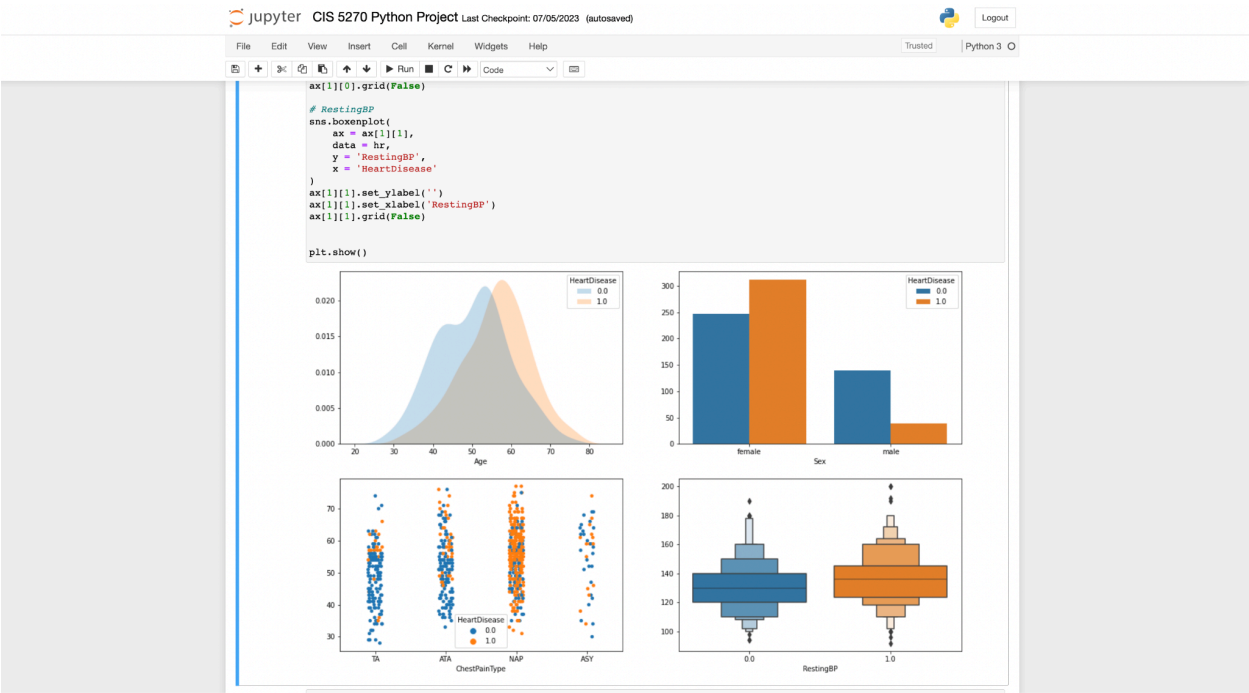
From a careful examination of the heat map, we observe that darker colors, such as dark orange and red, are indicative of high values within the dataset. These darker hues signify strong positive correlations between specific fields, suggesting a significant association between them. On the other hand, lighter colors, including dark blue and light blue, represent low values. These lighter shades denote weaker correlations or even negative correlations between certain fields, implying a weaker relationship or potential inverse association.

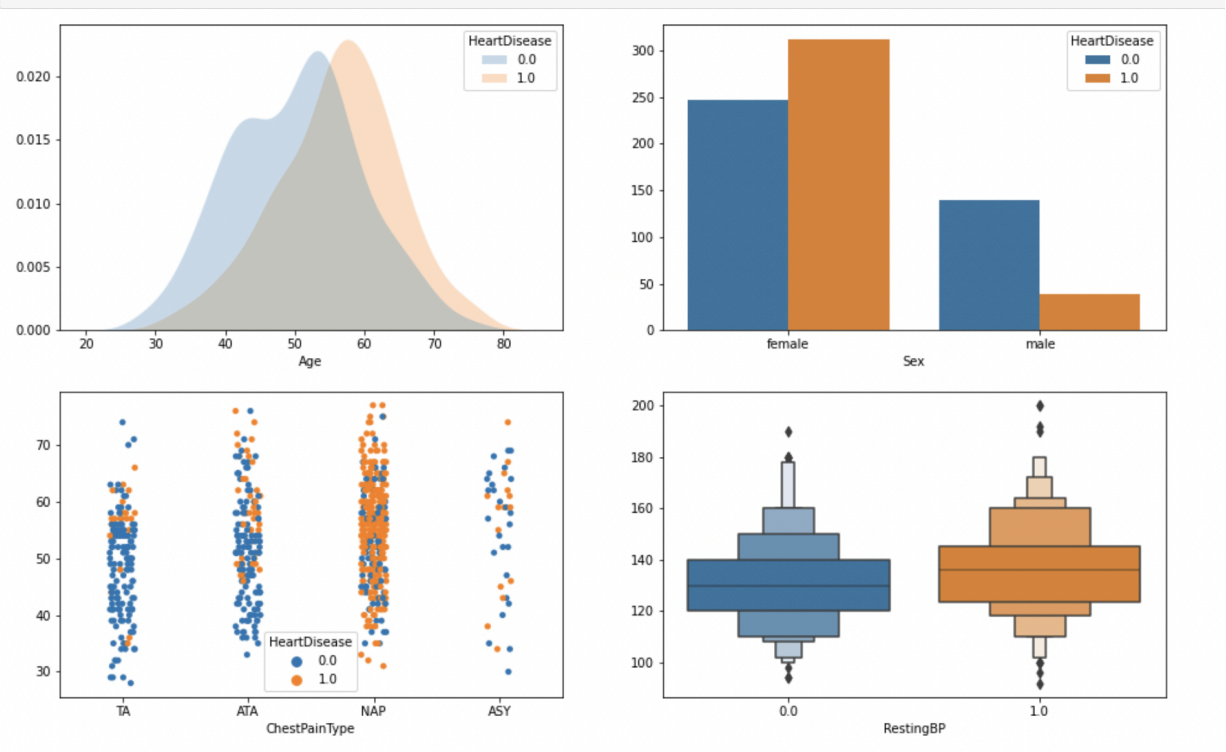
The heat map serves as a powerful tool to visualize the complex web of correlations within the dataset. It enables us to identify clusters of variables that exhibit strong positive associations, thus revealing potential patterns or dependencies between them. Conversely, the map also highlights areas where variables demonstrate weak or negative correlations, indicating the absence of substantial relationships. By deciphering these correlations, business analysts and decision-makers can gain valuable insights into the interconnectedness of the dataset's fields and leverage this knowledge to make informed decisions, devise targeted strategies, and unlock hidden opportunities for improvement or optimization.

In conclusion, the heat map provides a comprehensive and visually engaging representation of the correlations between all the fields in the dataset. By leveraging colors to depict the magnitude of values, the heat map offers an intuitive understanding of the interrelationships. This invaluable tool equips business analysts with the ability to identify strong positive associations and weak or negative correlations, empowering them to derive meaningful insights, drive data-informed decisions, and explore avenues for enhancement or optimization within the realm of the dataset.

Chart 3

What role do variables like chest pain type, resting BP (ECG), Age and Sex slope play in understanding the pathophysiology of heart failure?





1. Heart Diseases vs. Age:

   The chart depicting heart diseases vs. age provides intriguing insights into the relationship between age and the occurrence of heart diseases. Upon analyzing the chart, we observe a gradual increase in the prevalence of heart diseases as age progresses. This finding aligns with existing medical knowledge, as cardiovascular health tends to decline with advancing age. The chart suggests that individuals in older age groups may have a higher susceptibility to heart diseases, emphasizing the importance of proactive measures such as regular check-ups, lifestyle modifications, and early detection strategies for early intervention and prevention among the elderly population.

2. Heart Diseases vs. Sex:

   The chart illustrating heart diseases vs. sex sheds light on the gender disparities concerning heart diseases. By examining the chart, we can discern any variations in the prevalence of heart diseases between males and females. It is possible to observe whether one gender is more susceptible to heart diseases than the other. The insights derived from this chart can guide targeted interventions and preventive measures tailored to specific gender groups. For instance, if the chart reveals a higher incidence of heart diseases among males, it may necessitate the implementation of focused awareness campaigns and health initiatives specifically targeting men to improve cardiovascular health outcomes.

3. Heart Diseases vs. Chest Pain Type:

   The chart showcasing heart diseases vs. chest pain type provides valuable insights into the relationship between chest pain symptoms and the occurrence of heart diseases. By examining the chart, we can identify any patterns or trends regarding the chest pain types associated with heart diseases. This information can guide healthcare professionals in diagnosing and triaging patients presenting with chest pain, as certain chest pain types may indicate a higher likelihood of underlying heart diseases. The insights derived from this chart can aid in optimizing medical resources, refining diagnostic protocols, and improving patient care pathways for efficient management of individuals at risk.

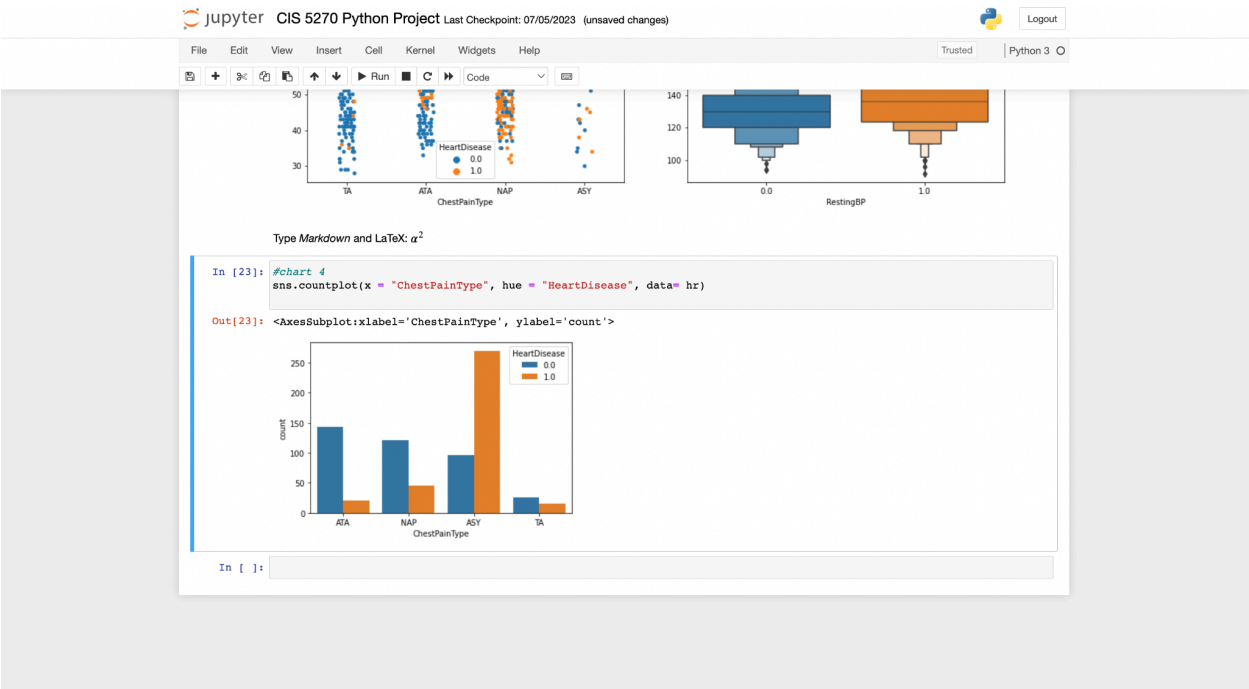
4. Heart Diseases vs. RestingBP:

   The chart illustrating heart diseases vs. restingBP offers valuable insights into the association between resting blood pressure levels and the prevalence of heart diseases. Analyzing the chart allows us to discern any patterns or trends regarding the relationship between resting blood pressure and the likelihood of heart diseases. Higher resting blood pressure levels might indicate an increased risk of heart diseases. This insight can inform healthcare professionals in monitoring and managing blood pressure levels to mitigate the risk of developing heart diseases. Moreover, it emphasizes the significance of promoting healthy blood pressure levels through lifestyle modifications and appropriate medical interventions.

In conclusion, the insights derived from these four charts provide a comprehensive understanding of the relationship between heart diseases and various factors such as age, sex, chest pain type, and resting blood pressure. These insights enable healthcare professionals, policymakers, and stakeholders to identify risk factors, design targeted interventions, and enhance prevention and management strategies to reduce the burden of heart diseases and improve cardiovascular health outcomes.

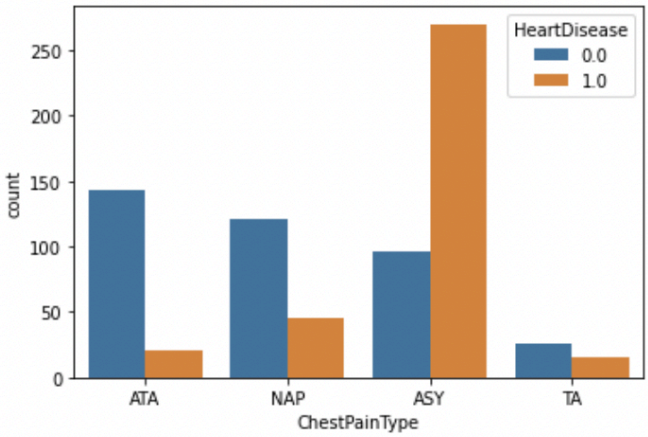
Chart 4

Different type of chest pains and their correlation with heart disease ?




```
In [23]: #chart 4
sns.countplot(x = "ChestPainType", hue = "HeartDisease", data= hr)
```

```
Out[23]: <AxesSubplot:xlabel='ChestPainType', ylabel='count'>
```



Upon analyzing the graph depicting heart diseases against chest pain types (TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic), a notable insight emerges. The bar representing the relationship between heart diseases and the chest pain type ASY (Asymptomatic) stands out as the highest among the different chest pain types. This finding suggests a strong association between individuals with asymptomatic chest pain and the presence of heart diseases.

The prominence of the ASY bar indicates that a significant number of individuals in the dataset who exhibited no apparent chest pain experienced heart diseases. This observation is particularly intriguing because asymptomatic individuals may not exhibit the classic symptoms typically associated with heart diseases, making early detection and diagnosis challenging. It underscores the importance of comprehensive cardiovascular assessments, including diagnostic tests, in identifying heart diseases even in the absence of noticeable chest pain.

These insights prompt further investigation into the characteristics, risk factors, and potential underlying mechanisms associated with asymptomatic individuals and their increased likelihood of heart diseases. It also emphasizes the significance of proactive healthcare measures, such as regular screenings, risk factor assessments, and comprehensive medical evaluations, to identify and manage heart diseases effectively, especially in individuals who may not present with the traditional chest pain symptoms. By focusing on early detection and appropriate interventions for asymptomatic individuals, healthcare professionals can potentially reduce the incidence and severity of heart diseases, improving overall cardiovascular health outcomes.