Exploratory Data Analysis Project

Brief description of the data set

For this project i'm going to use a Data Set representing Heart Attack classfication .Each case of heart attack has some parameters:

1. age - age in years

2. sex - sex (1 = male; 0 = female)

3. cp - chest pain type (1 = typical angina; 2 = atypical angina; 3 = non-anginal pain; 0 = asymptomatic)

4. trestbps - resting blood pressure (in mm Hg on admission to the hospital)

5. chol - serum cholestoral in mg/dl

6. fbs - fasting blood sugar > 120 mg/dl (1 = true; 0 = false)

7. restecg - resting electrocardiographic results (1 = normal; 2 = having ST-T wave abnormality; 0 = hypertrophy)

8. thalach - maximum heart rate achieved

9. exang - exercise induced angina (1 = yes; 0 = no)

10. oldpeak - ST depression induced by exercise relative to rest

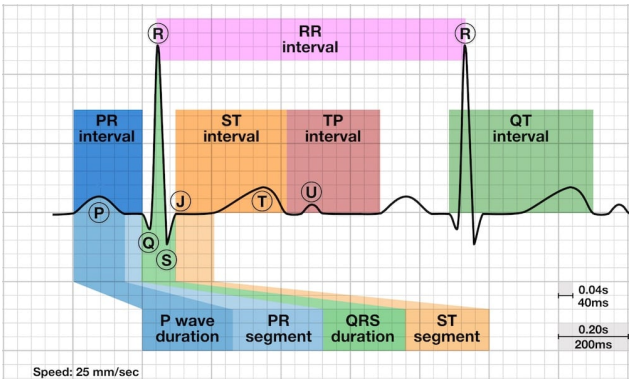
11. slope - the slope of the peak exercise ST segment (2 = upsloping; 1 = flat; 0 = downsloping)

12. ca - number of major vessels (0-3) colored by flourosopy

13. thal - 2 = normal; 1 = fixed defect; 3 = reversable defect

14. num - the predicted attribute - diagnosis of heart disease (angiographic disease status) (Value 0 = < diameter narrowing; Value 1 = > 50% diameter narrowing)

ECG WAVE



Initial plan for data exploration

This analysis is the base step in order to create a model that can predict a heart attack based on some clinical parameters.

1. Data Overview
2. Data Cleaning and Data Engineering Numerical Data
3. Data Cleaning and Data Engineering Categorical Data
4. Hypothesis Testing

Data Overview

To better understand the Data Set we need to know if all the columns are correctly name according to Data Set definition ,visualize all columns data types and the number of rows.

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

```
303
['age', 'sex', 'cp', 'trtbps', 'chol', 'fbs', 'restecg', 'thalachh', 'exng', 'oldpeak', 'slp', 'caa', 'thall',
age           int64
sex           int64
cp            int64
trtbps        int64
chol          int64
fbs           int64
restecg       int64
thalachh      int64
exng          int64
oldpeak     float64
slp           int64
caa           int64
thall         int64
output        int64
dtype: object
```

Categorical Data

Even if all values are numeric ,based on our data knowledge we can identify some data that are Categorical such as RestEcg , Slope,Thall and ChestPain. We need to define some dummy variables for each of these columns.

```
     thall_None  thall_Fixed_Defect  thall_Normal  thall_Reversable
0             0                   1             0                 0
1             0                   0             1                 0
2             0                   0             1                 0
3             0                   0             1                 0
4             0                   0             1                 0
..          ...                 ...           ...               ...
298           0                   0             0                 1
299           0                   0             0                 1
300           0                   0             0                 1
301           0                   0             0                 1
302           0                   0             1                 0

[303 rows x 4 columns]
     slp_Downsloping  slp_Flat  slp_Upsloping
0                  1         0              0
1                  1         0              0
2                  0         0              1
3                  0         0              1
4                  0         0              1
..               ...       ...            ...
298                0         1              0
299                0         1              0
300                0         1              0
301                0         1              0
302                0         1              0

[303 rows x 3 columns]
     cp_asymptomatic  cp_typical_angina  cp_atypical_angina  cp_no_angina
0                  0                  0                   0             1
1                  0                  0                   1             0
2                  0                  1                   0             0
3                  0                  1                   0             0
4                  1                  0                   0             0
..               ...                ...                 ...           ...
298                1                  0                   0             0
299                0                  0                   0             1
300                1                  0                   0             0
301                1                  0                   0             0
302                0                  1                   0             0

[303 rows x 4 columns]
     restecg_hypetrophic  restecg_normal  restecg_abnormal
0                      1               0                 0
1                      0               1                 0
2                      1               0                 0
3                      0               1                 0
4                      0               1                 0
..                   ...             ...               ...
298                    0               1                 0
299                    0               1                 0
300                    0               1                 0
301                    0               1                 0
302                    1               0                 0
```

✓ 0s

At the end of the transformations the dataset has 303 rows and 24 columns

Data Describe

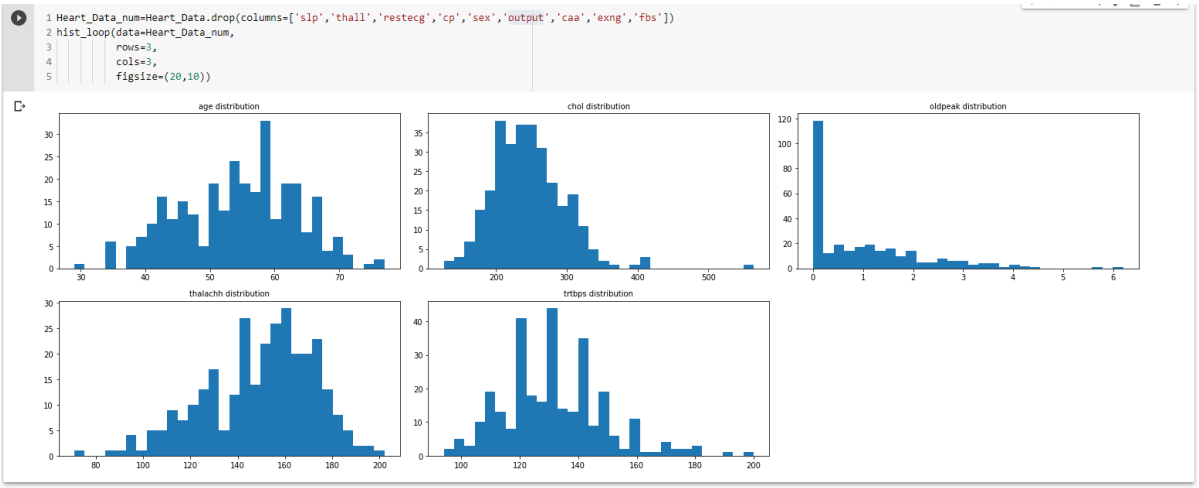
```
#Numerical Data
Heart_Data.describe()
```

	age	sex	trtbps	chol	fbs	thalachh	exng	oldpeak	caa	output	restecg_hypetrophic	restecg_normal	restecg_abnormal	cp_asymptomatic
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	131.623762	246.264026	0.148515	149.646865	0.326733	1.039604	0.729373	0.544554	0.485149	0.501650	0.013201	0.471947
std	9.082101	0.466011	17.538143	51.830751	0.356198	22.905161	0.469794	1.161075	1.022606	0.498835	0.500606	0.500824	0.114325	0.500038
min	29.000000	0.000000	94.000000	126.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	120.000000	211.000000	0.000000	133.500000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	55.000000	1.000000	130.000000	240.000000	0.000000	153.000000	0.000000	0.800000	0.000000	1.000000	0.000000	1.000000	0.000000	0.000000
75%	61.000000	1.000000	140.000000	274.500000	0.000000	166.000000	1.000000	1.600000	1.000000	1.000000	1.000000	1.000000	0.000000	1.000000
max	77.000000	1.000000	200.000000	564.000000	1.000000	202.000000	1.000000	6.200000	4.000000	1.000000	1.000000	1.000000	1.000000	1.000000

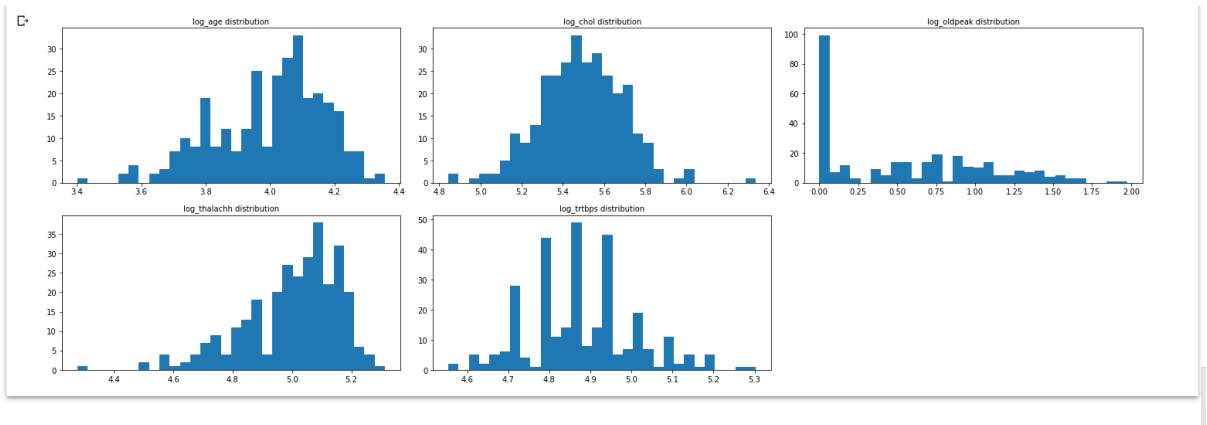
Numerical Data

We need to visualize the numerical columns in order to identify the skewed ones.

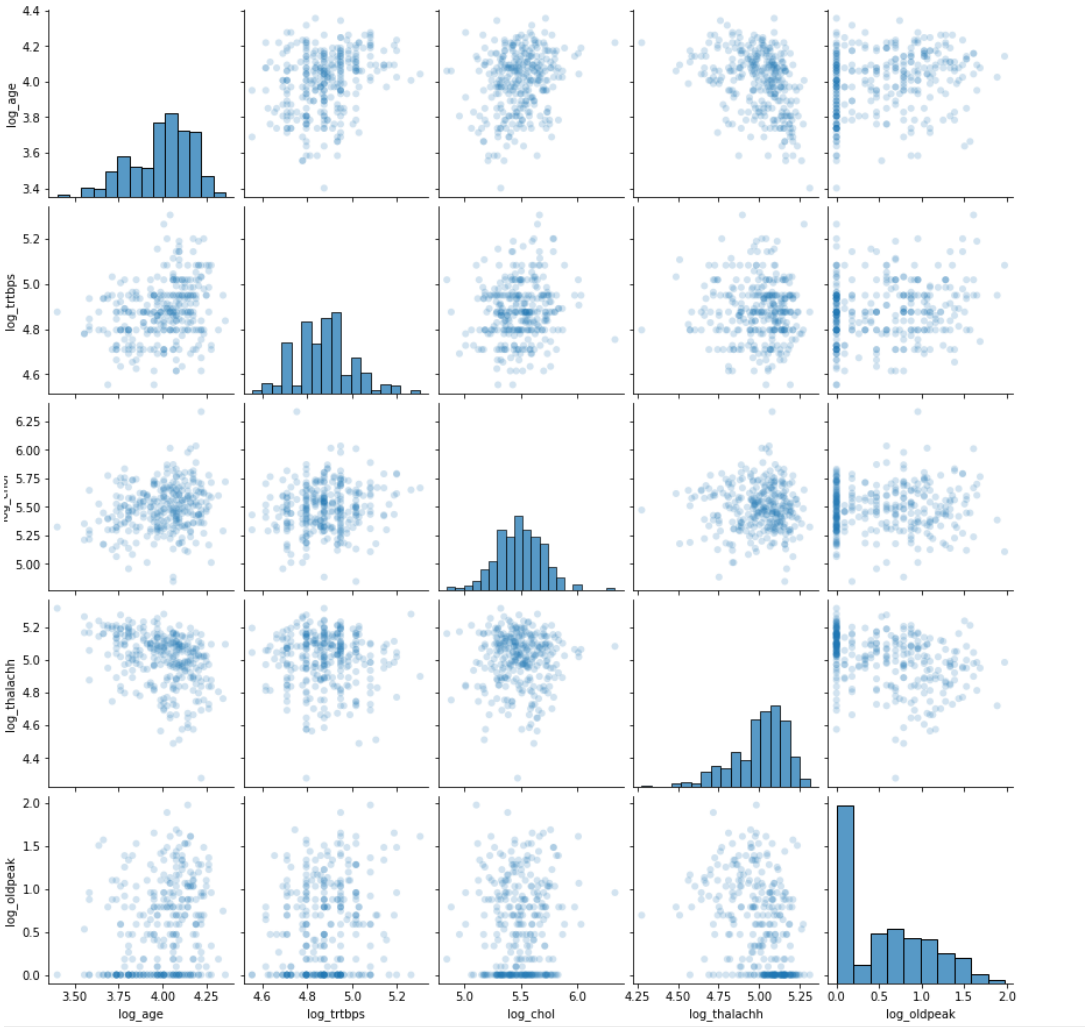
```
Heart_Data_num=Heart_Data.drop(columns=['slp','thall','restecg','cp','sex','output','caa','exng','fbs'])
hist_loop(data=Heart_Data_num,
          rows=3,
          cols=3,
          figsize=(20,10))
```



Chol and oldpeak are slightly skewed , oldpeak is very left skewed.
For the skewed ones we apply the log transformation and check for skewness again.
The



The pair plot doesn’t show any string correlation between all numerical data



Hypothesis Testing

We can test for example that age and cholesterol levels have the same mean using a t-student test.

```
#Hypothesis Testing
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import ttest_ind, t
import math

s1= Heart_Data_Fin['log_age']
s2= Heart_Data_Fin['log_chol']

result = ttest_ind(s1, s2, equal_var=False)
print("t-value" + " "+ str(result.statistic))
print("p-value" + " " + str(result.pvalue))
```

```
t-value -97.56842960009759
p-value 0.0
```

Other test can be done age against resting blood pressure (“trtbps”)

```
s3= Heart_Data_Fin['log_age']
s4= Heart_Data_Fin['log_trtbps']

result2 = ttest_ind(s3, s4, equal_var=False)
print("t-value" + " "+ str(result2.statistic))
print("p-value" + " " + str(result2.pvalue))
```

```
t-value -71.23724124308912
p-value 3.0305521351436204e-283
```

These tests show that in both case we reject the null hypothesis