Question 1:

```
n_at_median, n_at_mode, mode, nzero, nmissing=[],[],[],[],[]
stat=pd.DataFrame()
stat['cardinality']=df.nunique()
stat['mean']=df.mean()
stat['median']=df.median()
stat['stddev']=df.std()
stat['min']=df.min()
stat['max']=df.max()
median=list(stat['median'])
columns=list(df.columns)
j=0
for i in columns:
 m=median[j]
 mo=df[i].mode().iat[0]

 mode.append(mo)
 n_at_median.append((df[i]==m).sum())
 n_at_mode.append((df[i]==mo).sum())
 nzero.append((df[i]==0).sum())
 nmissing.append(df[i].isnull().sum())

 j+=1

stat.insert(3,'n_at_median', n_at_median)
stat.insert(4,'mode', mode)
stat.insert(5,'n_at_mode', n_at_mode)
stat['nzero']=nzero
stat['nmissing']=nmissing
s=['cardinality','mean','median','n_at_median','mode','n_at_mode','stddev'
,'min','max','nzero','nmissing']
stat_tran = stat.T
stat_t=stat_tran
stat_t.insert(0,'stat', s)
stat_t
```

stat_tran

	member	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	bt	target
cardinality	301	41	2	4	49	152	2	3	87	2	40	3	5	4	5	2
mean	48332.657807	54.366337	NaN	0.970199	131.615894	246.59	0.152027	0.529801	149.130597	0.324503	1.043046	1.398671	0.741611	2.313531	NaN	0.544554
median	48340.0	55.0	NaN	1.0	130.0	241.5	0.0	1.0	152.0	0.0	0.8	1.0	0.0	2.0	NaN	1.0
n_at_median	1	8	0	50	36	0	251	152	7	204	13	139	170	166	0	165
mode	46820.0	58	M	0.0	120.0	197.0	0.0	1.0	162.0	0.0	0.0	2.0	0.0	2	O	1
n_at_mode	1	19	206	142	37	6	251	152	9	204	98	141	170	166	115	165
stddev	877.940533	9.082101	NaN	1.032257	17.566716	51.969652	0.359655	0.525849	22.595484	0.468966	1.161452	0.616872	1.026753	0.612277	NaN	0.498835
min	46820.0	29.0	NaN	0.0	94.0	126.0	0.0	0.0	71.0	0.0	0.0	0.0	0.0	0.0	NaN	0.0
max	49840.0	77.0	NaN	3.0	200.0	564.0	1.0	2.0	202.0	1.0	6.2	2.0	4.0	3.0	NaN	1.0
nzero	0	0	0	142	0	0	251	146	0	204	98	21	170	2	0	138
nmissing	2	0	1	1	1	3	7	1	35	1	1	2	5	0	5	0

Question 2:

```
stat_t.to_excel('/content/statistics.xlsx', index = False)
```

The result for question 2 is in the spreadsheet named statistics_question2.xlsx

Question3:

a.The type of the feature (ID, target, or feature – and what type of feature – continuous, binary, interval, categorical, etc.)

Member- Continuous
'Age'- Numeric
'Sex'- categorical
'Cp'- categorical/numeric
'Trestbps'- numeric
'Chol'- numeric
'Fbs'- Binary
'restecg'-Binary
'Thalach'- Numeric
'Exang'-Binary
'Oldpeak'- Numeric
'Slope'- Categorical/numeric
'Ca'- Categorical/numeric
'Thal'- Numeric
'bt'-Categorical
'Target'- Binary

b. How many values are missing and how many are invalid

```
for i in columns:
 print(i,'-',df[i].isnull().sum(),' missing')
```

member - 2 missing
age - 0 missing
sex - 1 missing
cp - 1 missing
trestbps - 1 missing
chol - 3 missing
fbs - 7 missing
restecg - 1 missing
thalach - 35 missing
exang - 1 missing
oldpeak - 1 missing
slope - 2 missing
ca - 5 missing
thal - 0 missing
bt - 5 missing
target - 0 missing

c. What should be done about the missing values – BE SPECIFIC

member - Member is most likely to be member ID and is continuous data , so we can generate a new ID that is not in the column.

age - No missing data

sex - we have only one missing data, so we can add the frequent sex

cp - Only one missing data and is categorical data hence, we can choose the value for cp from the row with much resemblance.

trestbps - we have one variable missing and is a numeric data, hence we could select a mean value from rows that is male, cp=1 and target is one

chol - We have 3 variables missing and is also a numeric data, we can perform the same calculation as did for trestbps. Get the mean value from the rows that resembles the category

fbs - We have 7 missing data, and is a binary data, and imputing the column would be hard. But I will follow approach as I did in cp, choose the value for cp from the row with much resemblance.

restecg - We have 1 missing value, it's an binary data as well and we will follow the steps as in fbs.

thalach - We have 35 missing values, we can

exang - 1 missing and it's a categorical variable, so we can impute it with value of the row with most resemblance

oldpeak - 1 missing and it's numeric. I will impute it with taking mean of the class and sex

slope - 2 missing and is a categorical variable, so we can impute it with value of the row with most resemblance

ca - 5 missing and is a categorical variable, so we can impute it with value of the row with most resemblance

thal - No missing data

bt - 5 missing and is a categorical variable, so we can impute it with value of the row with most resemblance

target - No missing data

d. Whether the feature contains a significant number of outliers

```
# outliers
for c in col:
 q1=df[c].quantile(0.25)
 q3=df[c].quantile(0.75)
 iqr=q3-q1
 l_lim=q1-(1.5*iqr)
 u_lim=q3+(1.5*iqr)
 mem=list(df[c])
 o_l=[i for i in mem if i<l_lim]
 u_l=[i for i in mem if i>u_lim]
 print("outliers in ",c," : ",len(o_l)+len(u_l))
```

Results:

```
outliers in  member  :  0
outliers in  age  :  0
outliers in  cp  :  0
outliers in  trestbps  :  9
outliers in  chol  :  5
outliers in  fbs  :  45
outliers in  restecg  :  0
outliers in  thalach  :  1
outliers in  exang  :  0
outliers in  oldpeak  :  5
outliers in  slope  :  0
outliers in  ca  :  25
outliers in  thal  :  2
outliers in  target  :  0
```

e. Whether the feature should be ignored in modeling (by removing the column).

Members- No it is an important feature
Age- It has no outlier, so no
Cp- No it will not be removed
Trestbps- no it will not be removed
Chol- no it will not be removed
Fbs- yes it may be removed. It has 7 missing data and as well as 45 outliers so it's better to remove it
Restecg - no it will not be removed
Thalach-it will be removed (it's relationship with member is strong, but yet it as lots of missing value, with further analysis it can be determined whether it should be removed or not)
Exang - No it will not be removed
Oldpeak- it will not be removed
Slope- it will not be removed
Ca- ca has 25 outliers and 5 missing data, which is 10% of the dataset. As of now it will not be removed. Will be determined after training the model
Thal- no it will not be removed

Question 4:

```
cov=pd.DataFrame.cov(df)
cor=df.corr()
col='member', 'age', 'cp', 'trestbps', 'chol', 'fbs', 'restecg',
'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'
```

```
cov.insert(0,'column', col)
cor.insert(0,'column', col)
cov.to_excel('/content/cov.xlsx', index = False)
cor.to_excel('/content/cor.xlsx', index = False)
```

Correlation matrix

cov

```
cor
```

	column	member	age	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
member	member	1.000000	0.185808	-0.408005	0.110312	0.018508	-0.017950	-0.019009	-0.409539	0.358983	0.300974	-0.278008	0.374141	0.256390	-0.863288
age	age	0.185808	1.000000	-0.065252	0.279508	0.217346	0.117935	-0.114066	-0.362776	0.106317	0.212657	-0.170891	0.275932	0.068001	-0.225439
cp	cp	-0.408005	-0.065252	1.000000	0.045633	-0.083626	0.092840	0.044350	0.288622	-0.410762	-0.154484	0.121477	-0.181187	-0.158809	0.437885
trestbps	trestbps	0.110312	0.279508	0.045633	1.000000	0.121418	0.176945	-0.113304	-0.037984	0.071432	0.193315	-0.121440	0.100169	0.062470	-0.145548
chol	chol	0.018508	0.217346	-0.083626	0.121418	1.000000	0.019904	-0.155900	0.013848	0.071040	0.058719	-0.000195	0.069296	0.093505	-0.079997
fbs	fbs	-0.017950	0.117935	0.092840	0.176945	0.019904	1.000000	-0.083272	-0.015373	0.024151	0.005434	-0.060239	0.139262	-0.037614	-0.019244
restecg	restecg	-0.019009	-0.114066	0.044350	-0.113304	-0.155900	-0.083272	1.000000	0.021833	-0.070922	-0.059621	0.092035	-0.070504	-0.008272	0.134078
thalach	thalach	-0.409539	-0.362776	0.288622	-0.037984	0.013848	-0.015373	0.021833	1.000000	-0.360526	-0.343250	0.385144	-0.168381	-0.082123	0.413508
exang	exang	0.358983	0.106317	-0.410762	0.071432	0.071040	0.024151	-0.070922	-0.360526	1.000000	0.279031	-0.252018	0.117729	0.202530	-0.433899
oldpeak	oldpeak	0.300974	0.212657	-0.154484	0.193315	0.058719	0.005434	-0.059621	-0.343250	0.279031	1.000000	-0.579058	0.231374	0.209090	-0.435385
slope	slope	-0.278008	-0.170891	0.121477	-0.121440	-0.000195	-0.060239	0.092035	0.385144	-0.252018	-0.579058	1.000000	-0.089046	-0.104535	0.346633
ca	ca	0.374141	0.275932	-0.181187	0.100169	0.069296	0.139262	-0.070504	-0.168381	0.117729	0.231374	-0.089046	1.000000	0.143738	-0.385114
thal	thal	0.256390	0.068001	-0.158809	0.062470	0.093505	-0.037614	-0.008272	-0.082123	0.202530	0.209090	-0.104535	0.143738	1.000000	-0.344029
target	target	-0.863288	-0.225439	0.437885	-0.145548	-0.079997	-0.019244	0.134078	0.413508	-0.433899	-0.435385	0.346633	-0.385114	-0.344029	1.000000

Covariance

cor

```
cov
```

column	member	age	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	targe
member	770779.579181	1485.511960	-368.049052	1703.156299	849.032043	-5.554433	-8.759532	-8177.454310	147.931327	308.318584	-150.857669	332.204650	138.158361	-378.27740
age	1485.511960	82.484558	-0.611362	44.662472	102.963211	0.384963	-0.545181	-73.330510	0.451123	2.244928	-0.958793	2.565035	0.378139	-1.02134
cp	-368.049052	-0.611362	1.065554	0.829635	-4.488575	0.034763	0.024086	6.564360	-0.198062	-0.185462	0.077503	-0.191942	-0.100328	0.22554
trestbps	1703.156299	44.662472	0.829635	308.589514	110.953649	1.129367	-1.048328	-15.404621	0.589535	3.956998	-1.317874	1.820320	0.672724	-1.27577
chol	849.032043	102.963211	-4.488575	110.953649	2700.844716	0.370144	-4.264641	16.464951	1.733934	3.545596	-0.006282	3.706157	2.959699	-2.07551
fbs	-5.554433	0.384963	0.034763	1.129367	0.370144	0.129352	-0.015796	-0.126614	0.004093	0.002277	-0.013338	0.051147	-0.008280	-0.00345
restecg	-8.759532	-0.545181	0.024086	-1.048328	-4.264641	-0.015796	0.276518	0.261412	-0.017464	-0.036440	0.029866	-0.038163	-0.002662	0.03515
thalach	-8177.454310	-73.330510	6.564360	-15.404621	16.464951	-0.126614	0.261412	510.555914	-3.798130	-8.988380	5.357866	-3.984863	-1.151168	4.66739
exang	147.931327	0.451123	-0.198062	0.589535	1.733934	0.004093	-0.017464	-3.798130	0.219929	0.150930	-0.072910	0.056830	0.058129	-0.10147
oldpeak	308.318584	2.244928	-0.185462	3.956998	3.545596	0.002277	-0.036440	-8.988380	0.150930	1.348971	-0.415897	0.275694	0.148872	-0.25216
slope	-150.857669	-0.958793	0.077503	-1.317874	-0.006282	-0.013338	0.029866	5.357866	-0.072910	-0.415897	0.380532	-0.055749	-0.039579	0.10672
ca	332.204650	2.565035	-0.191942	1.820320	3.706157	0.051147	-0.038163	-3.984863	0.056830	0.275694	-0.055749	1.054222	0.090254	-0.19750
thal	138.158361	0.378139	-0.100328	0.672724	2.959699	-0.008280	-0.002662	-1.151168	0.058129	0.148872	-0.039579	0.090254	0.374883	-0.10507
target	-378.277409	-1.021343	0.225540	-1.275770	-2.075518	-0.003459	0.035159	4.667393	-0.101472	-0.252168	0.106722	-0.197501	-0.105075	0.24883

Question 5:

Three most strongly correlated predictors:

1. Member: -0.863288260173753
2. Cp : 0.4378852211182
3. Oldpeak: -0.435385064430489

Three highly correlated features

1. Slope- Oldpeak: 0.579058203414419
2. Cp - Exang: -0.41076227621935
3. Member- Thalach: -0.409539374909683