EX-06-Feature-Transformation

AIM

To Perform the various feature transformation techniques on a dataset and save the data to a file.

Explanation

Feature Transformation is a mathematical transformation in which we apply a mathematical formula to a particular column(feature) and transform the values which are useful for our further analysis.

ALGORITHM

STEP 1

Read the given Data

STEP 2

Clean the Data Set using Data Cleaning Process

STEP 3

Apply Feature Transformation techniques to all the feature of the data set

STEP 4

Save the data to the file

CODE

Developed by:SAI SONICA CH
Reg no:212219040130

Data_to_Transform.csv :

```
# importing packages
import numpy as np
```

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
import scipy.stats as stats
from sklearn.preprocessing import QuantileTransformer
qt=QuantileTransformer(output_distribution='normal')

df=pd.read_csv("Data_to_Transform.csv")
df

#checking and analysing the data
df.isnull().sum()

#checking for skewness of data
df.skew()

#applying data transformations
dfmp=pd.DataFrame()
#for Moderate Positive Skew
#function transformation
dfmp["Moderate Positive Skew"]=df["Moderate Positive Skew"]
dfmp["MPS_log"]=np.log(df["Moderate Positive Skew"])
dfmp["MPS_rp"]=np.reciprocal(df["Moderate Positive Skew"])
dfmp["MPS_sqr"]=np.sqrt(df["Moderate Positive Skew"])
#power transformation
dfmp["MPS_yj"], parameters=stats.yeojohnson(df["Moderate Positive Skew"])
dfmp["MPS_bc"], parameters=stats.boxcox(df["Highly Positive Skew"])
#quantile transformation
dfmp["MPS_qt"]=qt.fit_transform(df[["Moderate Positive Skew"]])
dfmp.skew()

dfmp.drop('MPS_rp',axis=1,inplace=True)
dfmp.skew()

dfmp

#for Highly Positive Skew
#function transformation
dfhp=pd.DataFrame()
dfhp["Highly Positive Skew"]=df["Highly Positive Skew"]
dfhp["HPS_log"]=np.log(df["Highly Positive Skew"])
dfhp["HPS_rp"]=np.reciprocal(df["Highly Positive Skew"])
dfhp["HPS_sqr"]=np.sqrt(df["Highly Positive Skew"])
#power transformation
dfhp["HPS_yj"], parameters=stats.yeojohnson(df["Highly Positive Skew"])
dfhp["HPS_bc"], parameters=stats.boxcox(df["Highly Positive Skew"])
#quantile transformation
dfhp["HPS_qt"]=qt.fit_transform(df[["Highly Positive Skew"]])
dfhp.skew()

dfhp.drop('HPS_sqr',axis=1,inplace=True)
dfhp.skew()
```

```
dfhp

#for Moderate Negative Skew
dfmn=pd.DataFrame()
#function transformation
dfmn["Moderate Negative Skew"]=df["Moderate Negative Skew"]
dfmn["MNS_rp"]=np.reciprocal(df["Moderate Negative Skew"])
dfmn["MNS_sq"]=np.square(df["Moderate Negative Skew"])
#power transformation
dfmn["MNS_yj"], parameters=stats.yeojohnson(df["Moderate Negative Skew"])
#quantile transformation
dfmn["MNS_qt"]=qt.fit_transform(df[["Moderate Negative Skew"]])
dfmn.skew()

dfmn.drop('MNS_rp',axis=1,inplace=True)
dfmn.skew()

dfmn

#for Highly Negative Skew
dfhn=pd.DataFrame()
#function transformation
dfhn["Highly Negative Skew"]=df["Highly Negative Skew"]
dfhn["HNS_rp"]=np.reciprocal(df["Highly Negative Skew"])
dfhn["HNS_sq"]=np.square(df["Highly Negative Skew"])
#phwer transformation
dfhn["HNS_yj"], parameters=stats.yeojohnson(df["Highly Negative Skew"])
#quantile transformation
dfhn["HNS_qt"]=qt.fit_transform(df[["Highly Negative Skew"]])
dfhn.skew()

dfhn.drop('HNS_rp',axis=1,inplace=True)
dfhn.skew()

dfhn

#graphical representation
#for Moderate Positive Skew
df["Moderate Positive Skew"].hist()
dfmp["MPS_log"].hist()
dfmp["MPS_sqr"].hist()
dfmp["MPS_bc"].hist()
dfmp["MPS_yj"].hist()
sm.qqplot(df['Moderate Positive Skew'],line='45')
plt.show()
sm.qqplot(dfmp['MPS_qt'],line='45')
plt.show()

#for Highly Positive Skew
df["Highly Positive Skew"].hist()
dfhp["HPS_log"].hist()
dfhp["HPS_rp"].hist()
```

```
dfhp["HPS_bc"].hist()
dfhp["HPS_yj"].hist()
sm.qqplot(df['Highly Positive Skew'],line='45')
plt.show()
sm.qqplot(dfhp['HPS_qt'],line='45')
plt.show()

#for Moderate Negative Skew
df["Moderate Negative Skew"].hist()
dfmn["MNS_sq"].hist()
dfmn["MNS_yj"].hist()
sm.qqplot(df['Moderate Negative Skew'],line='45')
plt.show()
sm.qqplot(dfmn['MNS_qt'],line='45')
plt.show()

#for Highly Negative Skew
df["Highly Negative Skew"].hist()
dfhn["HNS_sq"].hist()
dfhn["HNS_yj"].hist()
sm.qqplot(df['Highly Negative Skew'],line='45')
plt.show()
sm.qqplot(dfhn['HNS_qt'],line='45')
plt.show()
```

OUPUT

	Moderate Positive Skew	Highly Positive Skew	Moderate Negative Skew	Highly Negative Skew
0	0.899990	2.895074	11.180748	9.027485
1	1.113554	2.962385	10.842938	9.009762
2	1.156830	2.966378	10.817934	9.006134
3	1.264131	3.000324	10.764570	9.000125
4	1.323914	3.012109	10.753117	8.981296
...	...	...	...	...
9995	14.749050	16.289513	-2.980821	-3.254882
9996	14.854474	16.396252	-3.147526	-3.772332
9997	15.262103	17.102991	-3.517256	-4.717950
9998	15.269983	17.628467	-4.689833	-5.670496
9999	16.204517	18.052331	-6.335679	-7.036091

10000 rows × 4 columns

```
#checking and analysing the data
df.isnull().sum()
```

```
Moderate Positive Skew    0
Highly Positive Skew      0
Moderate Negative Skew    0
Highly Negative Skew      0
dtype: int64
```

```
#checking for skewness of data
df.skew()
```

```
Moderate Positive Skew    0.656308
Highly Positive Skew      1.271249
Moderate Negative Skew   -0.690244
Highly Negative Skew     -1.201891
dtype: float64
```

```
Moderate Positive Skew    0.656308
MPS_log                  -0.392229
MPS_rp                    2.029045
MPS_sqr                   0.152479
MPS_yj                   -0.001168
MPS_bc                    0.023089
MPS_qt                    0.000895
dtype: float64
```

```
dfmp.drop('MPS_rp',axis=1,inplace=True)
dfmp.skew()
```

```
Moderate Positive Skew    0.656308
MPS_log                  -0.392229
MPS_sqr                   0.152479
MPS_yj                   -0.001168
MPS_bc                    0.023089
MPS_qt                    0.000895
dtype: float64
```

	Moderate Positive Skew	MPS_log	MPS_sqr	MPS_yj	MPS_bc	MPS_qt
0	0.899990	-0.105371	0.948678	0.690865	0.812909	-5.199338
1	1.113554	0.107557	1.055251	0.815560	0.825921	-3.392734
2	1.156830	0.145684	1.075560	0.839629	0.826679	-3.341853
3	1.264131	0.234385	1.124336	0.897735	0.833058	-3.243698
4	1.323914	0.280593	1.150615	0.929191	0.835247	-3.200142
...	...	...	...	...	...	...
9995	14.749050	2.691179	3.840449	3.828849	1.457701	3.203464
9996	14.854474	2.698301	3.854150	3.841318	1.459189	3.225052
9997	15.262103	2.725373	3.906674	3.888934	1.468681	3.326574
9998	15.269983	2.725889	3.907683	3.889845	1.475357	3.328914
9999	16.204517	2.785290	4.025483	3.995584	1.480525	5.199338

10000 rows × 6 columns

```
Highly Positive Skew    1.271249
HPS_log                 0.398780
HPS_rp                  0.283884
HPS_sqr                 0.800174
HPS_yj                  0.031338
HPS_bc                  0.023089
HPS_qt                 -0.000408
dtype: float64
```

```
dfhp.drop('HPS_sqr',axis=1,inplace=True)
dfhp.skew()
```

```
Highly Positive Skew    1.271249
HPS_log                 0.398780
HPS_rp                  0.283884
HPS_yj                  0.031338
HPS_bc                  0.023089
HPS_qt                 -0.000408
dtype: float64
```

```
dfhp
```

	Highly Positive Skew	HPS_log	HPS_rp	HPS_yj	HPS_bc	HPS_qt
0	2.895074	1.063011	0.345414	0.839656	0.812909	-5.199338
1	2.962385	1.085995	0.337566	0.845576	0.825921	-3.342974
2	2.966378	1.087342	0.337112	0.845921	0.826679	-3.326950
3	3.000324	1.098720	0.333297	0.848834	0.833058	-3.216858
4	3.012109	1.102640	0.331993	0.849835	0.835247	-3.186281
...	...	...	...	...	...	...
9995	16.289513	2.790522	0.061389	1.146703	1.457701	3.198360
9996	16.396252	2.797053	0.060990	1.147374	1.459189	3.216327
9997	17.102991	2.839253	0.058469	1.151638	1.468681	3.372712
9998	17.628467	2.869515	0.056726	1.154622	1.475357	3.588705
9999	18.052331	2.893275	0.055395	1.156922	1.480525	5.199338

10000 rows × 6 columns

```
Moderate Negative Skew    -0.690244
MNS_rp                   -16.162250
MNS_sq                     0.347926
MNS_yj                    -0.119651
MNS_qt                    -0.001436
dtype: float64
```

```
dfmn.drop('MNS_rp',axis=1,inplace=True)
dfmn.skew()
```

```
Moderate Negative Skew   -0.690244
MNS_sq                    0.347926
MNS_yj                   -0.119651
MNS_qt                   -0.001436
dtype: float64
```

```
dfmn
```

	Moderate Negative Skew	MNS_sq	MNS_yj	MNS_qt
0	11.180748	125.009116	29.137805	5.199338
1	10.842938	117.569305	27.885272	3.227288
2	10.817934	117.027691	27.793301	3.206801
3	10.764570	115.875961	27.597360	3.167111
4	10.753117	115.629522	27.555368	3.159208
...	...	...	...	...
9995	-2.980821	8.885297	-1.949345	-3.147619
9996	-3.147526	9.906918	-2.028952	-3.162489
9997	-3.517256	12.371089	-2.199693	-3.198205
9998	-4.689833	21.994536	-2.697151	-3.350199
9999	-6.335679	40.140828	-3.311402	-5.199338

10000 rows × 4 columns

```
Highly Negative Skew   -1.201891
HNS_rp                 -2.136295
HNS_sq                 -0.156984
HNS_yj                 -0.274676
HNS_qt                  0.003126
dtype: float64
```

```
dfhn.drop('HNS_rp',axis=1,inplace=True)
dfhn.skew()
```

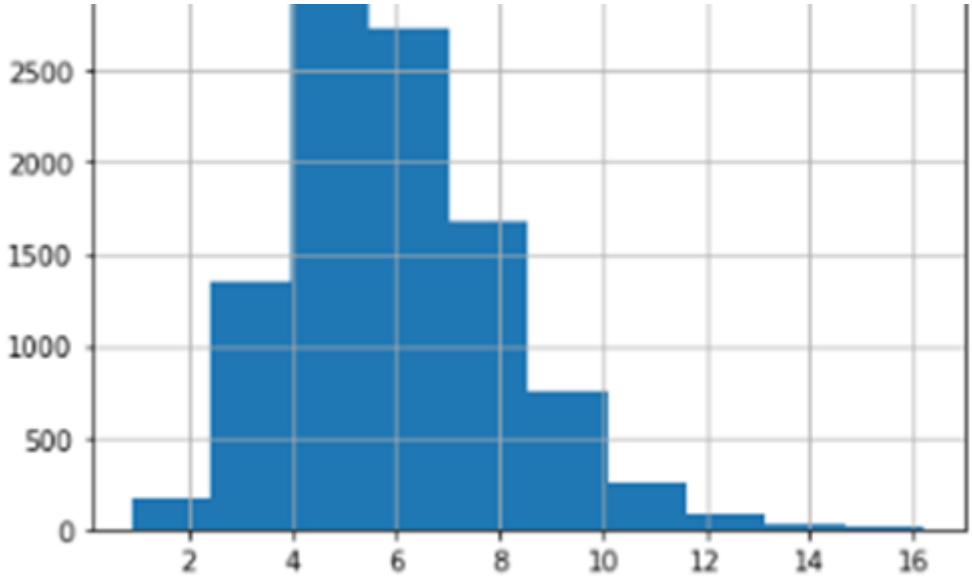
```
Highly Negative Skew   -1.201891
HNS_sq                 -0.156984
HNS_yj                 -0.274676
HNS_qt                  0.003126
dtype: float64
```

```
dfhn
```

	Highly Negative Skew	HNS_sq	HNS_yj	HNS_qt
0	9.027485	81.495480	51.081487	5.199338
1	9.009762	81.175811	50.898041	3.503580
2	9.006134	81.110452	50.860530	3.453669
3	9.000125	81.002257	50.798432	3.386210
4	8.981296	80.663680	50.604084	3.239746
...	...	...	...	...
9995	-3.254882	10.594259	-1.433326	-3.131880
9996	-3.772332	14.230487	-1.545673	-3.174835
9997	-4.717950	22.259048	-1.722267	-3.272809
9998	-5.670496	32.154520	-1.872430	-3.419532
9999	-7.036091	49.506580	-2.053503	-5.199338

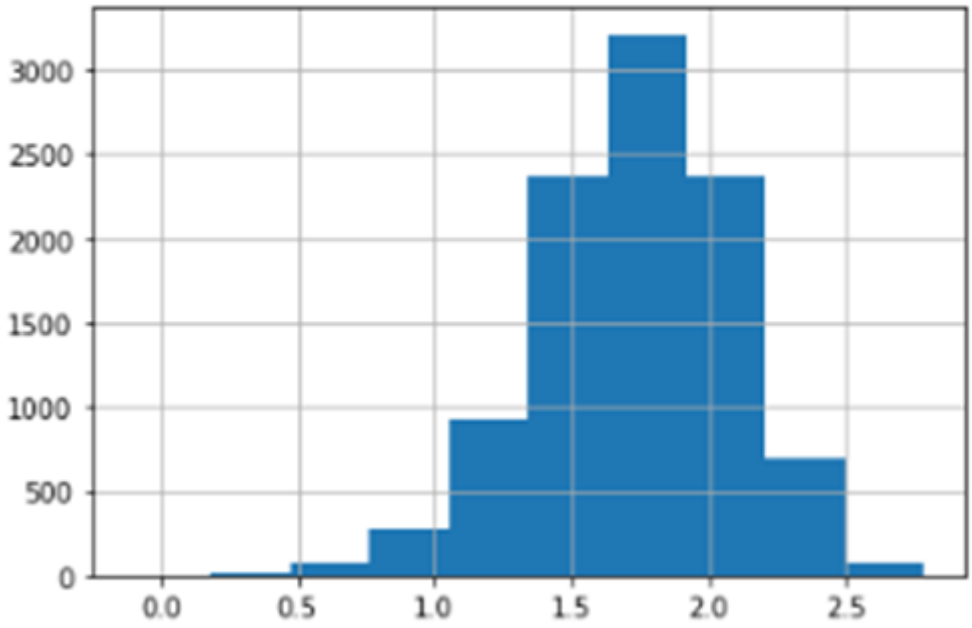
10000 rows × 4 columns

```
<AxesSubplot:>
```



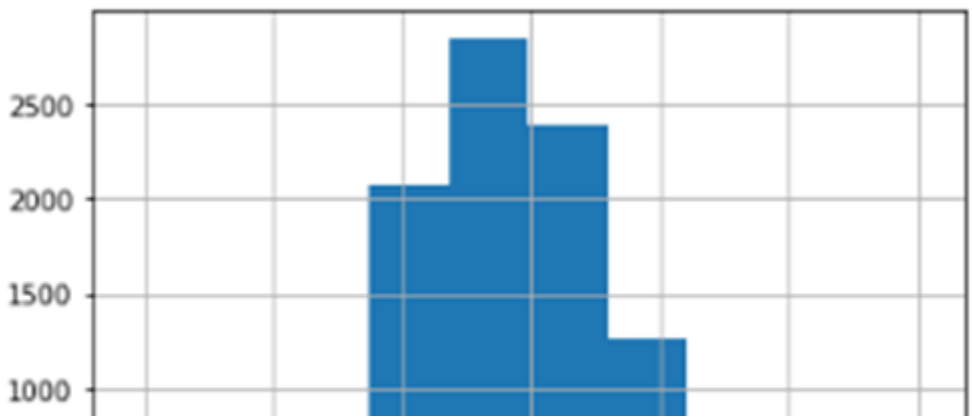

```
dfmp["MPS_log"].hist()
```

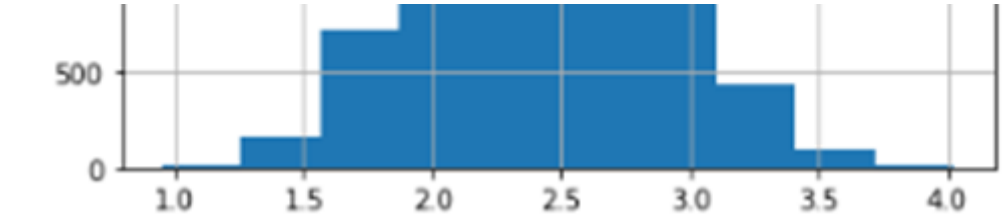
```
<AxesSubplot:>
```



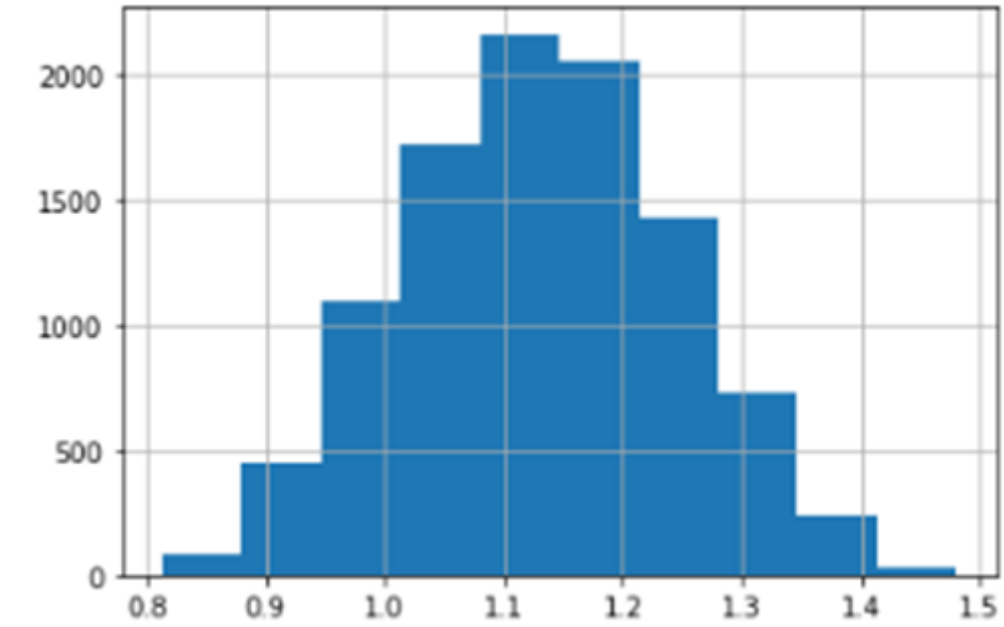

```
dfmp["MPS_sqr"].hist()
```

```
<AxesSubplot:>
```



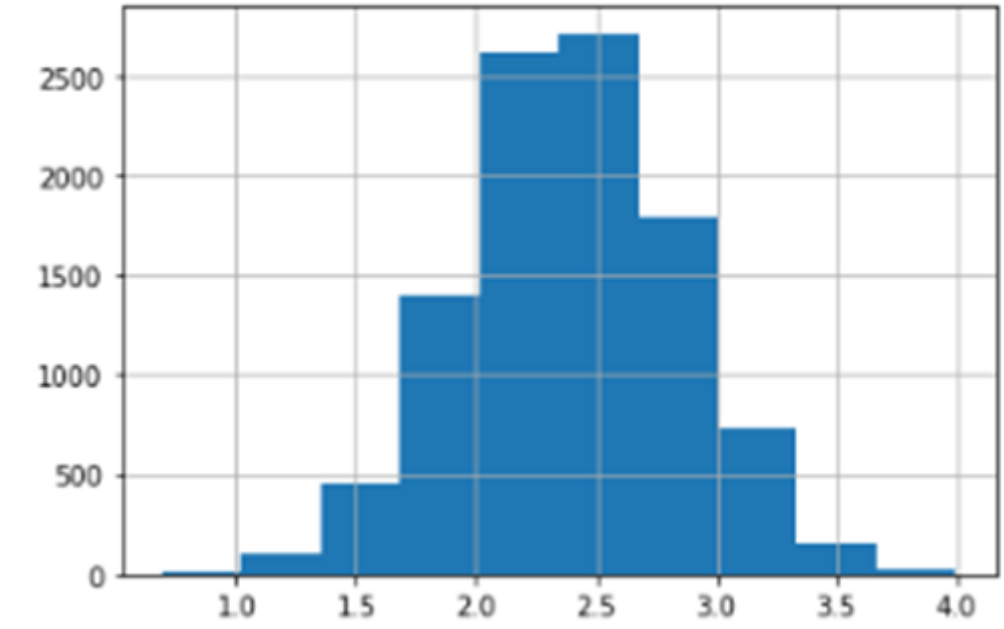



```
<AxesSubplot:>
```

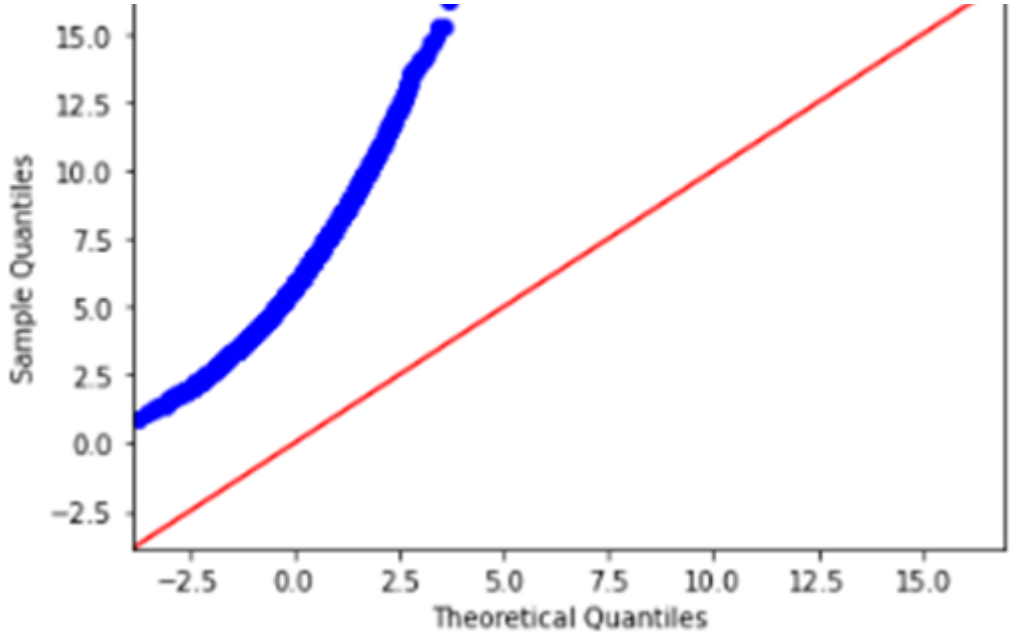


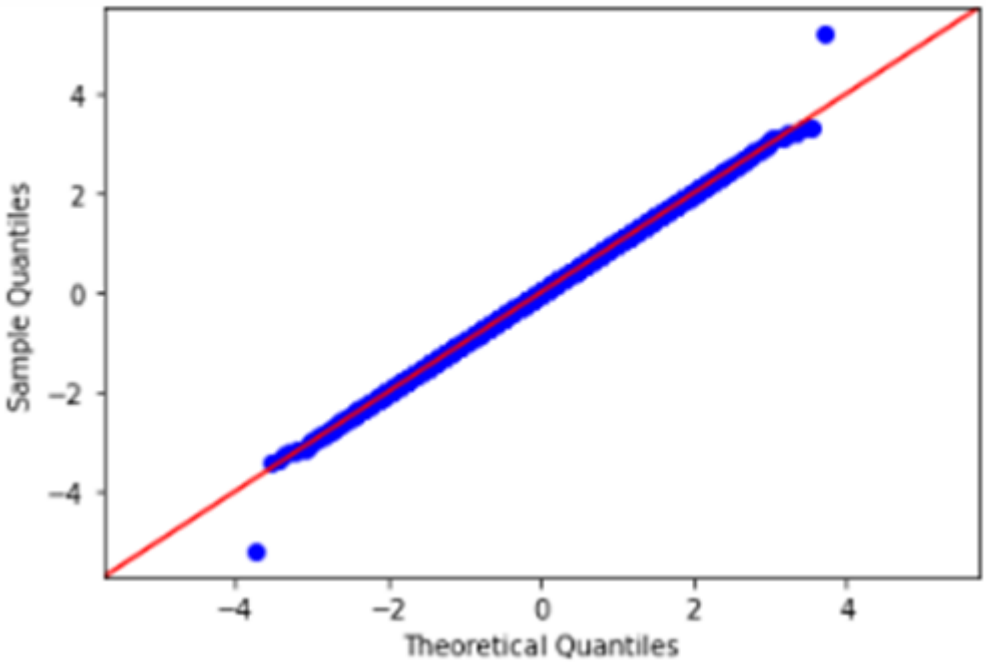

```
dfmp["MPS_yj"].hist()
```

```
<AxesSubplot:>
```



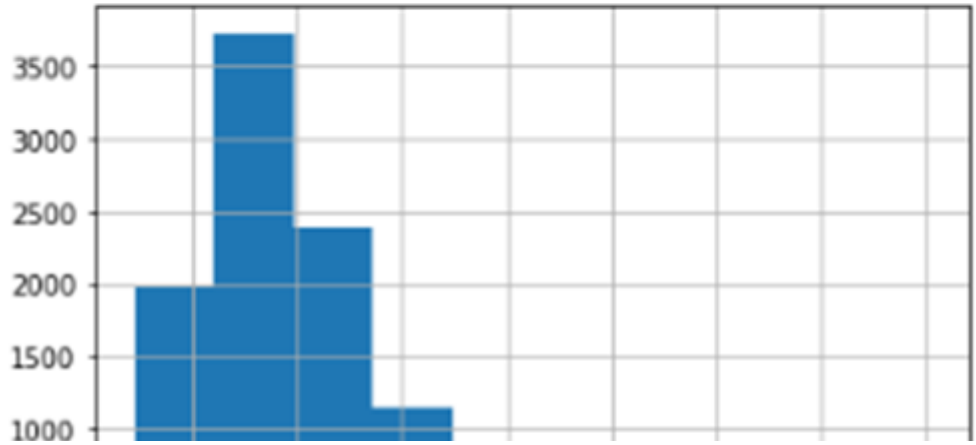

```
sm.qqplot(df['Moderate Positive Skew'],line='45')
plt.show()
```

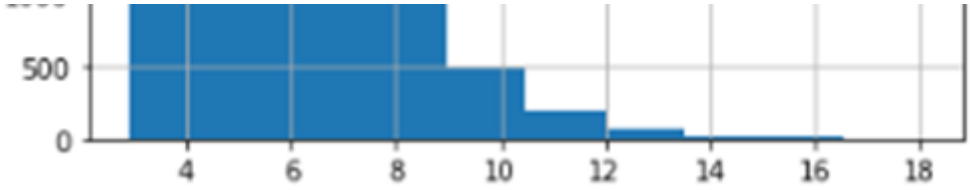





```
#for Highly Positive Skew
df["Highly Positive Skew"].hist()
```

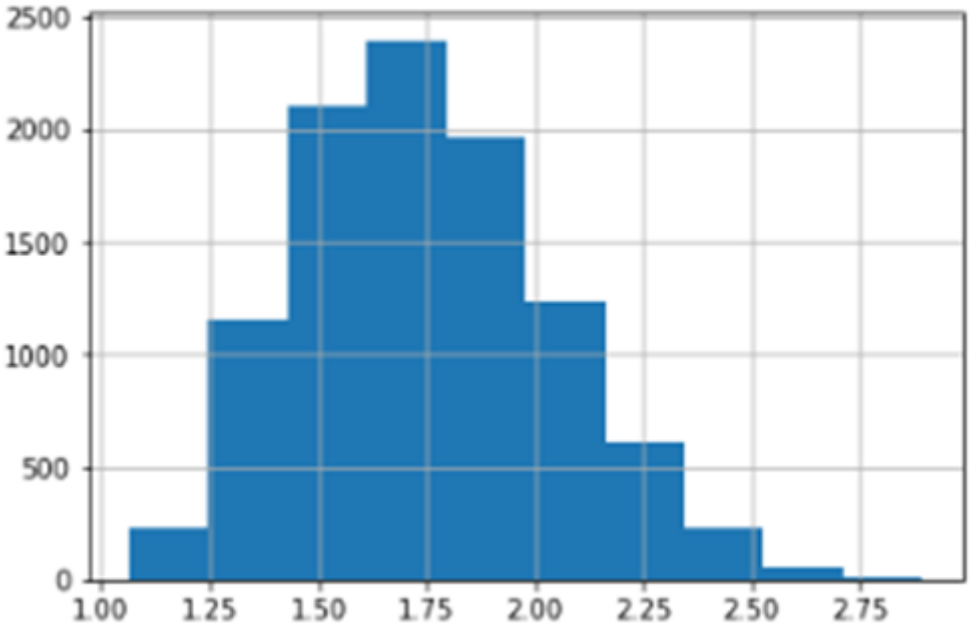
```
<AxesSubplot:>
```



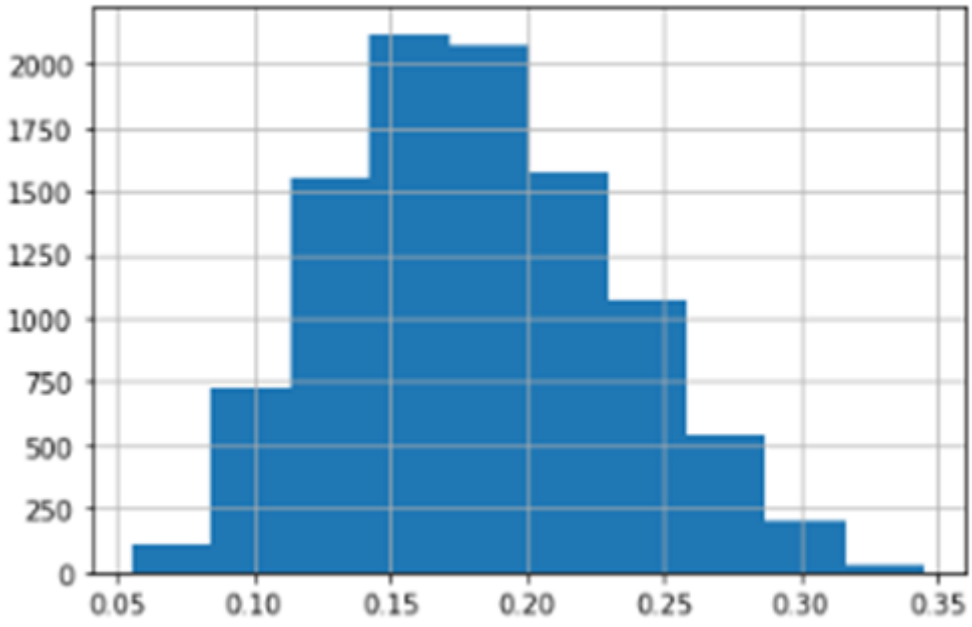



```
dfhp["HPS_log"].hist()
```

```
<AxesSubplot:>
```

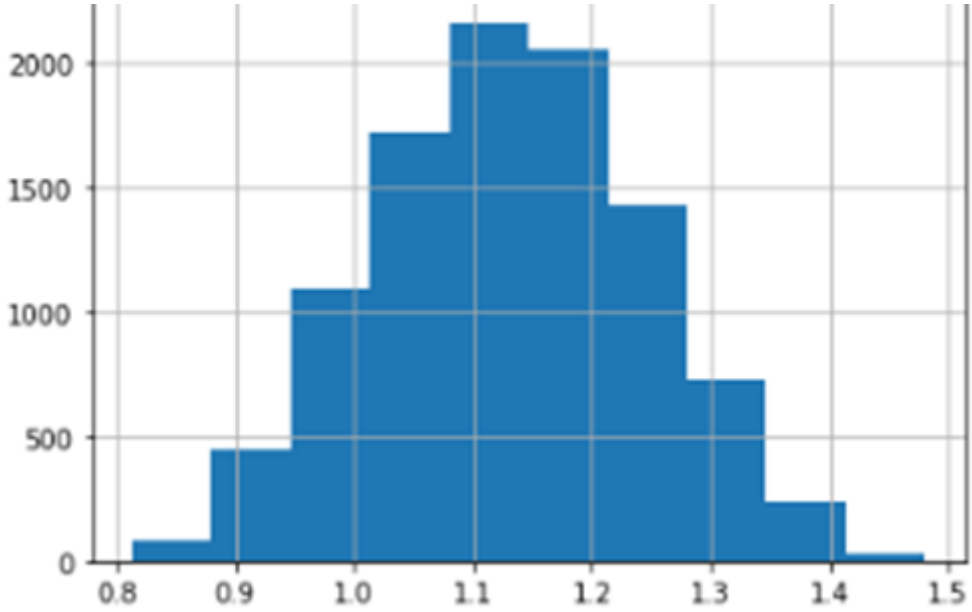



```
<AxesSubplot:>
```



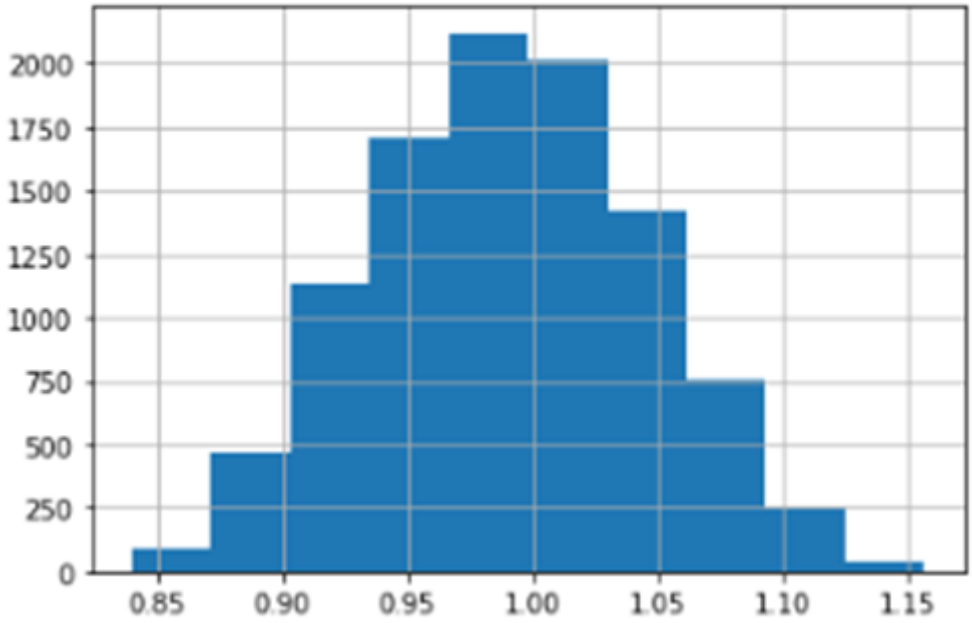

```
dfhp["HPS_bc"].hist()
```

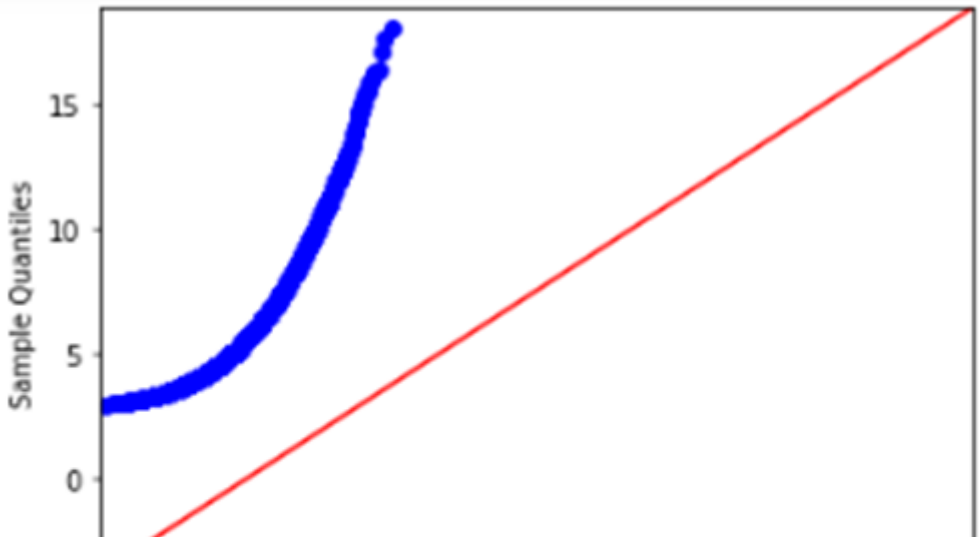
```
<AxesSubplot:>
```

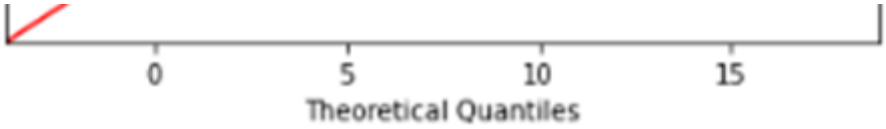



```
dfhp["HPS_yj"].hist()
```

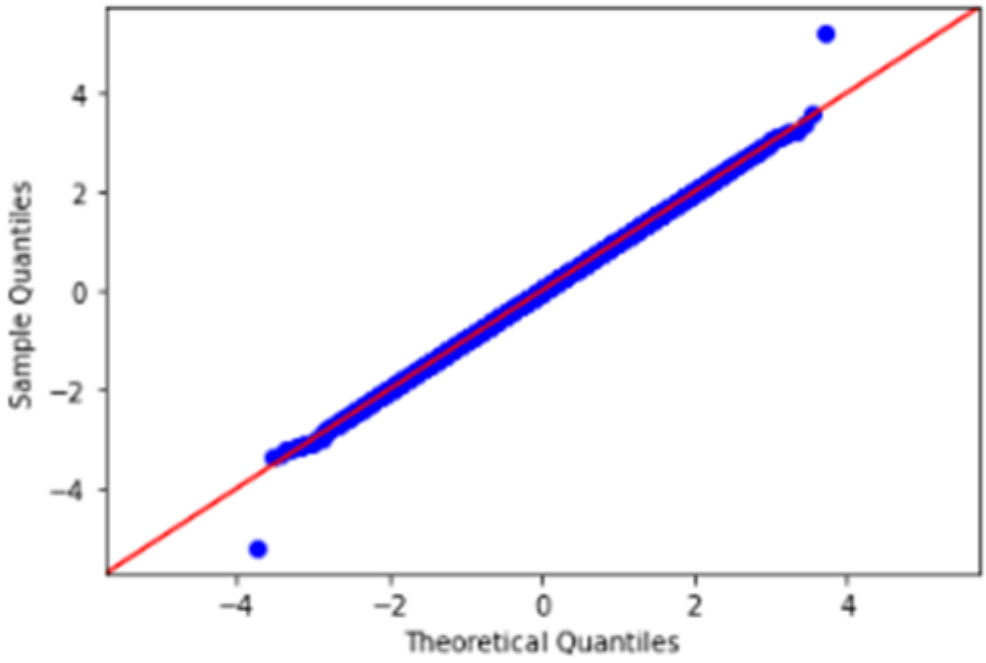
```
<AxesSubplot:>
```





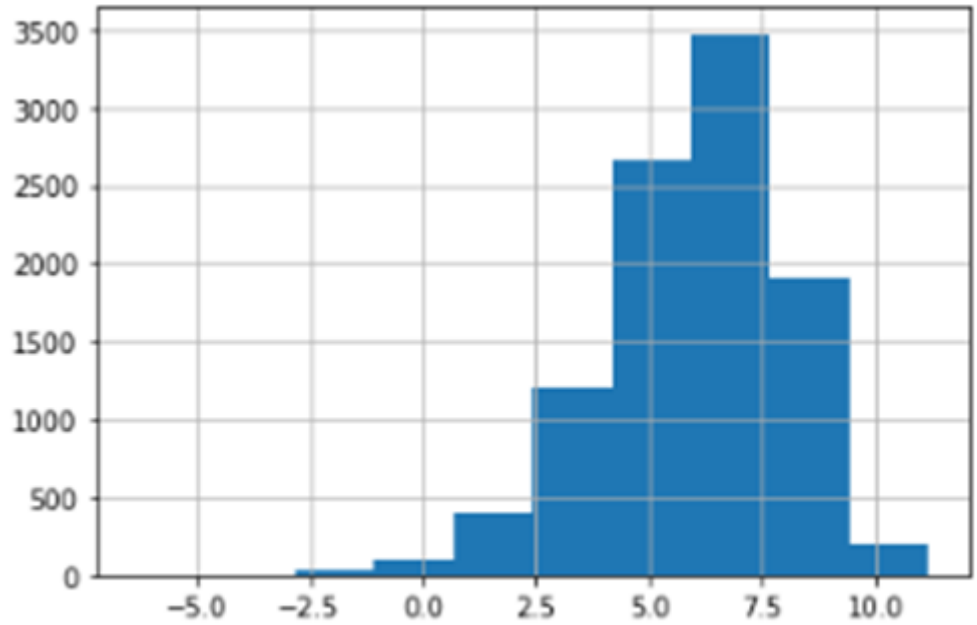



```
sm.qqplot(dfhp['HPS_qt'],line='45')
plt.show()
```

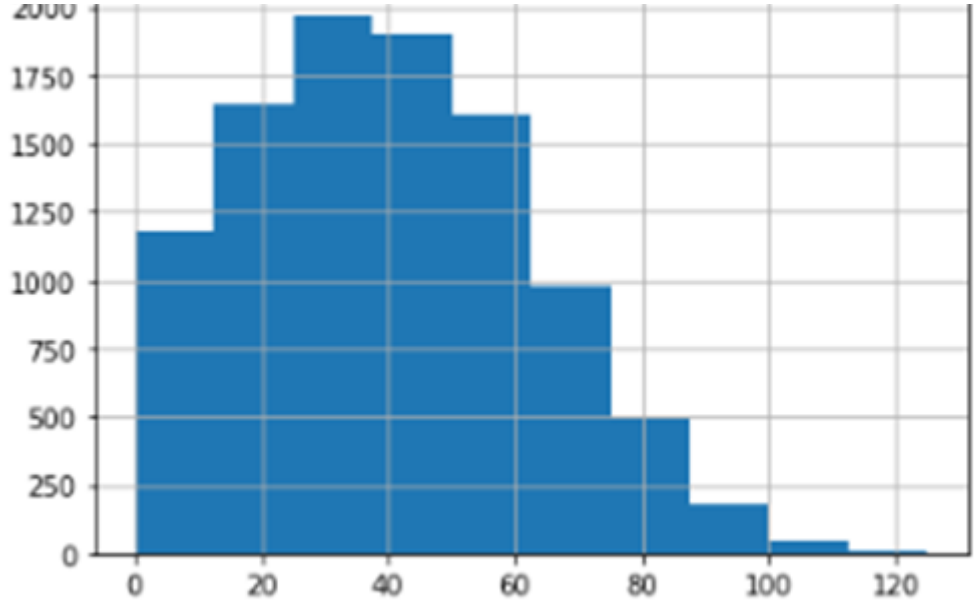



```
#for Moderate Negative Skew
df["Moderate Negative Skew"].hist()
```

```
<AxesSubplot:>
```

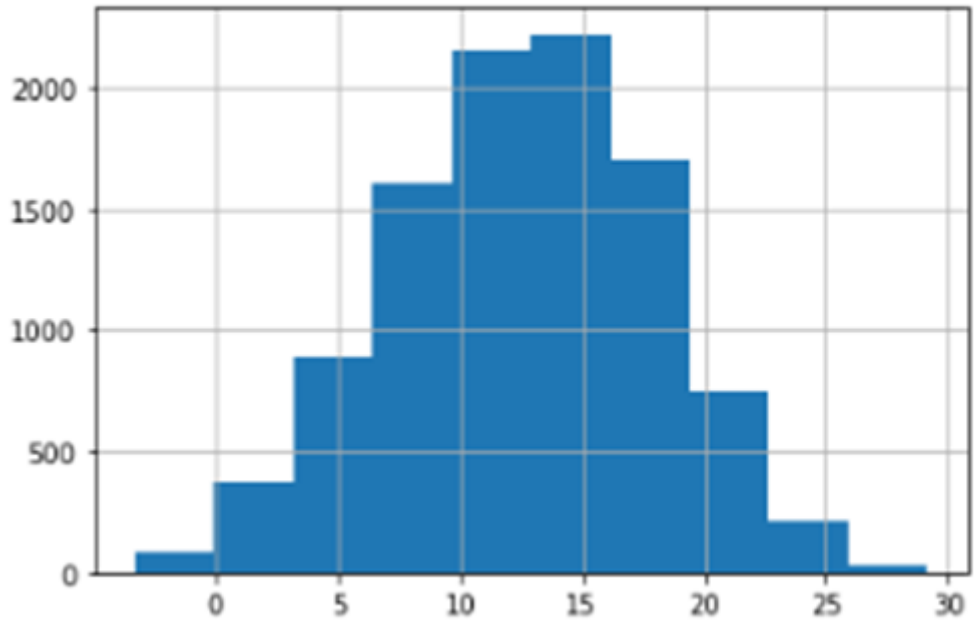



```
<AxesSubplot:>
```

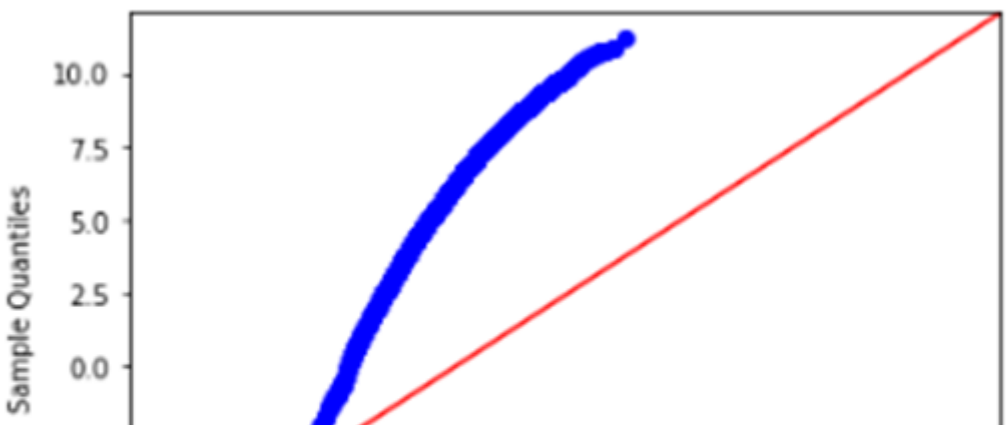


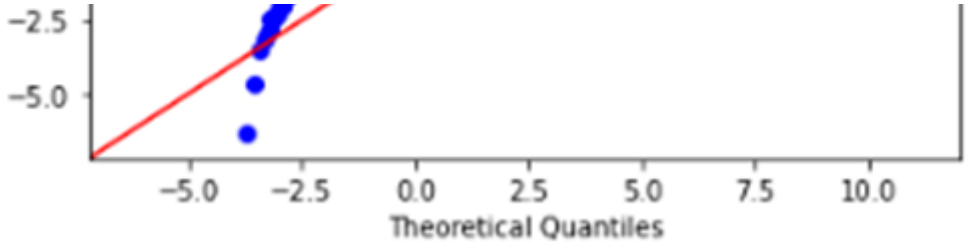

```
dfmn["MNS_yj"].hist()
```

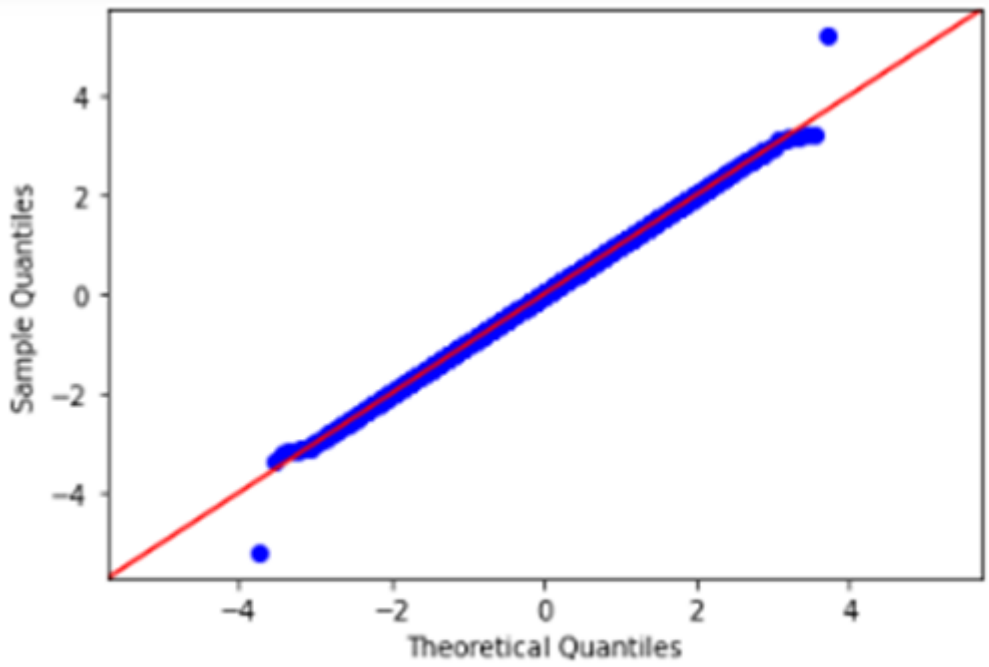
```
<AxesSubplot:>
```




```
sm.qqplot(df['Moderate Negative Skew'],line='45')
plt.show()
```

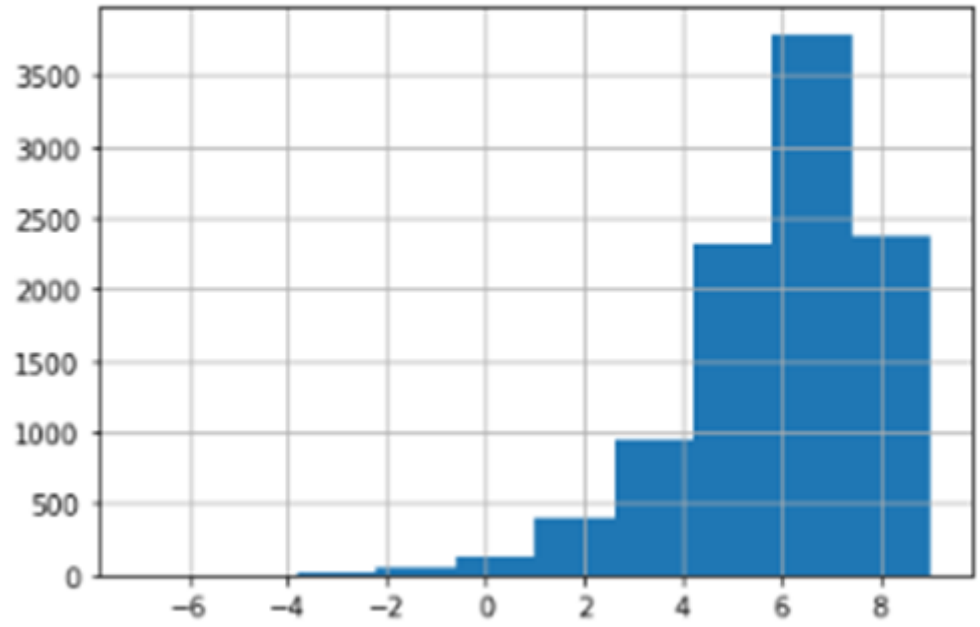






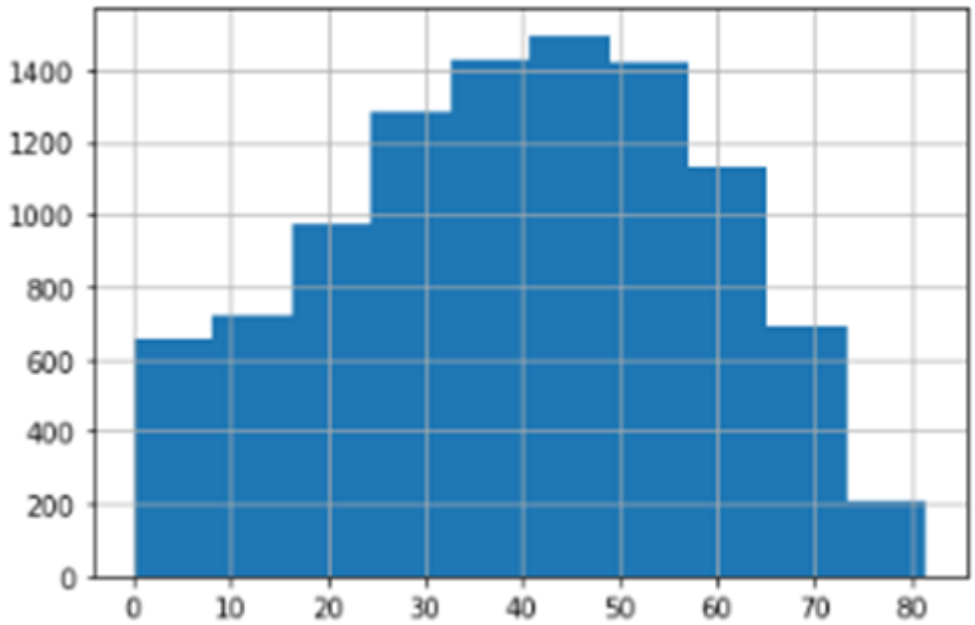

```
#for Highly Negative Skew
df["Highly Negative Skew"].hist()
```

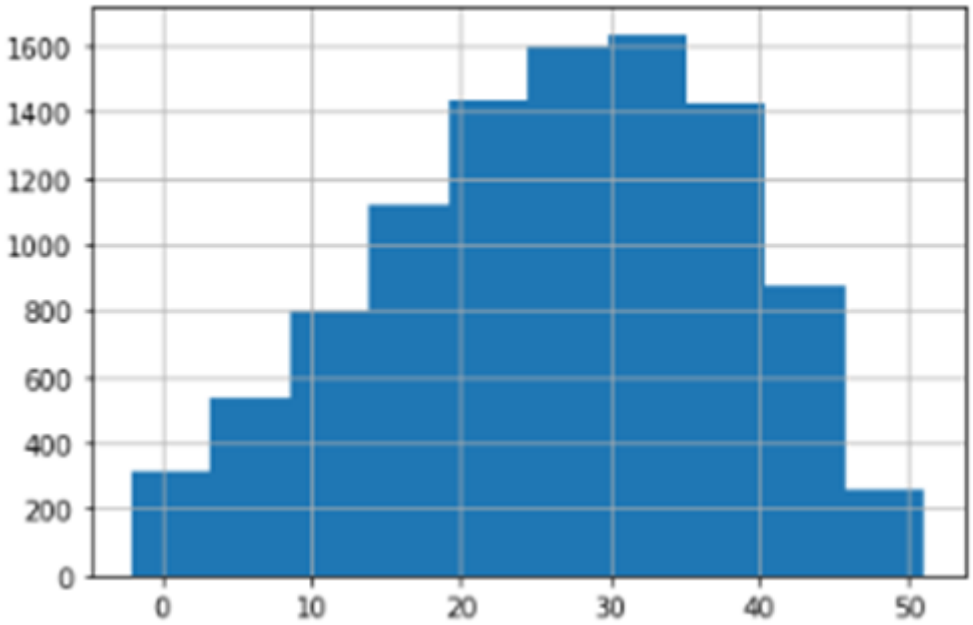
```
<AxesSubplot:>
```



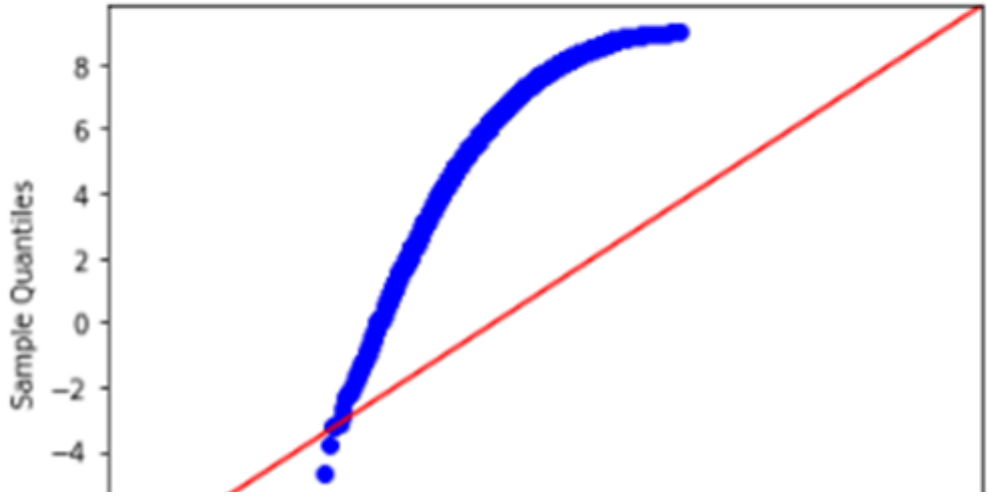

```
dfhn["HNS_sq"].hist()
```

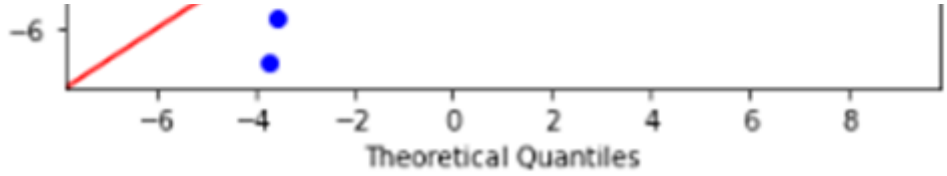
```
<AxesSubplot:>
```



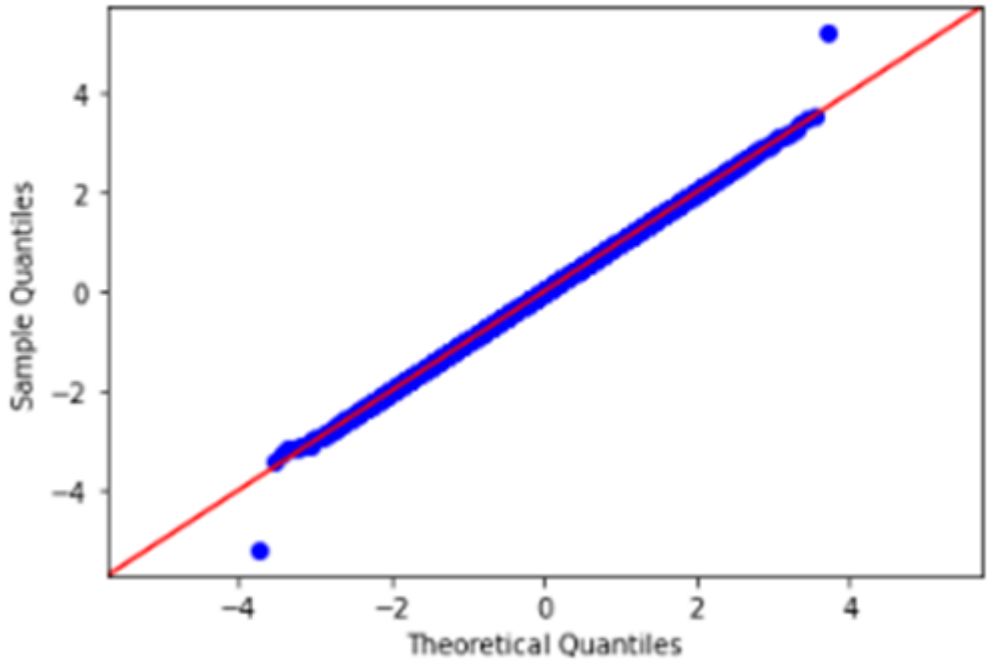



```
sm.qqplot(df['Highly Negative Skew'],line='45')
plt.show()
```






```
sm.qqplot(dfhn['HNS_qt'],line='45')
plt.show()
```



For Titanic_dataset.csv:

```
# importing packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
import scipy.stats as stats
from sklearn.preprocessing import QuantileTransformer
qt=QuantileTransformer(output_distribution='normal',n_quantiles=891)

df=pd.read_csv("titanic_dataset.csv")
df

#checking and analysing the data
df.isnull().sum()
#cleaning data
df.drop('Cabin',axis=1,inplace=True)
df.drop('Name',axis=1,inplace=True)
df.drop('Ticket',axis=1,inplace=True)
df.drop('PassengerId',axis=1,inplace=True)
```

```
df['Age']=df['Age'].fillna(df['Age'].median())
df['Embarked']=df['Embarked'].fillna(df['Embarked'].mode()[0])
df.isnull().sum()
# encoding categorical data
from sklearn.preprocessing import OrdinalEncoder
embark=["C","S","Q"]
emb=OrdinalEncoder(categories=[embark])
df["Embarked"]=emb.fit_transform(df[["Embarked"]])

from category_encoders import BinaryEncoder
be=BinaryEncoder()
df["Sex"]=be.fit_transform(df[["Sex"]])
df.skew()

#feature in 0.5 to -0.5 range: Embarked Survived Age (A,S pos)(E neg)
#features that are in skew
#neg:Pclass
#pos:Sex SibSp Parch Fare
df["Age_1"]=qt.fit_transform(df[["Age"]])
df["Survived_1"]=qt.fit_transform(df[["Survived"]])
df["Embarked_1"]=qt.fit_transform(df[["Embarked"]])
df["Pclass_sq"]=np.square(df["Pclass"])
df["Pclass_qt"]=qt.fit_transform(df[["Pclass"]])
df["SibSp_yj"], parameters=stats.yeojohnson(df["SibSp"])
df["SibSp_qt"]=qt.fit_transform(df[["SibSp"]])

df["Parch_yj"], parameters=stats.yeojohnson(df["Parch"])
df["Parch_qt"]=qt.fit_transform(df[["Parch"]])

df["Fare_yj"], parameters=stats.yeojohnson(df["Fare"])
df["Fare_qt"]=qt.fit_transform(df[["Fare"]])

df["Sex_yj"], parameters=stats.yeojohnson(df["Sex"])
df["Sex_qt"]=qt.fit_transform(df[["Sex"]])
df.skew()

#taking closer to range skew values
df.drop('Sex_yj',axis=1,inplace=True)
df.drop('Pclass_qt',axis=1,inplace=True)
df.drop('SibSp_qt',axis=1,inplace=True)
df.drop('Parch_qt',axis=1,inplace=True)
df.drop('Fare_qt',axis=1,inplace=True)
df.skew()

#graph representation
df["Sex"].hist()
df["Sex_qt"].hist()
df["SibSp"].hist()
df["SibSp_yj"].hist()
df["Parch"].hist()
df["Parch_yj"].hist()
df["Fare"].hist()
df["Fare_yj"].hist()
```

```
df["Pclass"].hist()
df["Pclass_sq"].hist()
```

OUTPUT:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

```
Survived    0
Pclass      0
Sex         0
Age         0
SibSp       0
Parch       0
Fare        0
Embarked    0
dtype: int64
```

```
Survived    0.478523
Pclass     -0.630548
Sex         0.618921
Age         0.510245
SibSp       3.695352
Parch       2.749117
Fare        4.787317
Embarked   -0.147331
dtype: float64
```

```
Survived      0.478523
Pclass       -0.630548
Sex           0.618921
Age           0.510245
SibSp         3.695352
Parch         2.749117
Fare          4.787317
Embarked     -0.147331
Age_1        -0.006827
Survived_1    0.478523
Embarked_1   -0.243123
Pclass_sq    -0.444064
Pclass_qt    -0.569343
SibSp_yj      0.808608
SibSp_qt      0.869466
Parch_yj      1.228795
Parch_qt      1.257239
Fare_yj      -0.040329
Fare_qt      -0.928213
Sex_yj        0.618921
```

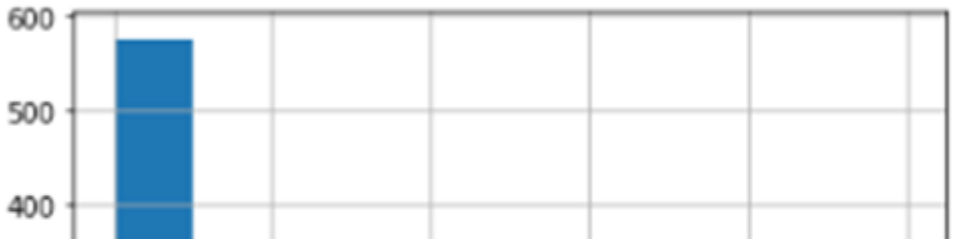
```
Sex_qt         0.618921
dtype: float64
```

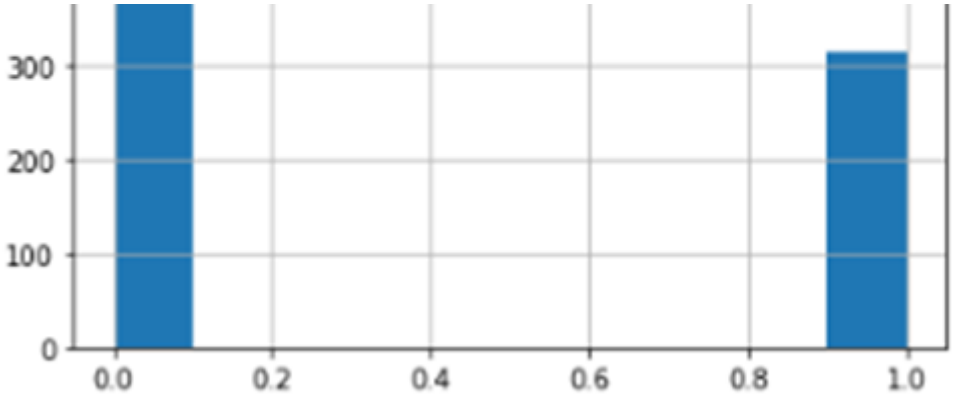
```
#taking closer to range skew values
df.drop('Sex_yj',axis=1,inplace=True)
df.drop('Pclass_qt',axis=1,inplace=True)
df.drop('SibSp_qt',axis=1,inplace=True)
df.drop('Parch_qt',axis=1,inplace=True)
df.drop('Fare_qt',axis=1,inplace=True)
df.skew()
```

```
Survived       0.478523
Pclass        -0.630548
Sex            0.618921
Age            0.510245
SibSp          3.695352
Parch          2.749117
Fare           4.787317
Embarked      -0.147331
Age_1         -0.006827
Survived_1     0.478523
Embarked_1    -0.243123
Pclass_sq     -0.444064
SibSp_yj       0.808608
Parch_yj       1.228795
Fare_yj       -0.040329
Sex_qt         0.618921
dtype: float64
```

Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked	Age_1	Survived_1	Embarked_1	Pclass_sq	SibSp_yj	Parch_yj	Fare_yj	Sex_qt
0	3	0	22.0	1	0	7.2500	1.0	-0.695859	-5.199338	0.128501	9	0.323389	-0.000000	1.906724	-5.199338
1	1	1	38.0	1	0	71.2833	0.0	0.823696	5.199338	-5.199338	1	0.323389	-0.000000	3.497640	5.199338
1	3	1	26.0	0	0	7.9250	1.0	-0.391395	5.199338	0.128501	9	-0.000000	-0.000000	1.970459	5.199338
1	1	1	35.0	1	0	53.1000	1.0	0.662165	5.199338	0.128501	1	0.323389	-0.000000	3.304258	5.199338
0	3	0	35.0	0	0	8.0500	1.0	0.662165	-5.199338	0.128501	9	-0.000000	-0.000000	1.981680	-5.199338
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
0	2	0	27.0	0	0	13.0000	1.0	-0.337215	-5.199338	0.128501	4	-0.000000	-0.000000	2.326029	-5.199338
1	1	1	19.0	0	0	30.0000	1.0	-0.957723	5.199338	0.128501	1	-0.000000	-0.000000	2.916885	5.199338
0	3	1	28.0	1	2	23.4500	1.0	-0.021125	-5.199338	0.128501	9	0.323389	0.243296	2.745246	5.199338
1	1	0	26.0	0	0	30.0000	0.0	-0.391395	5.199338	-5.199338	1	-0.000000	-0.000000	2.916885	-5.199338
0	3	0	32.0	0	0	7.7500	2.0	0.493940	-5.199338	5.199338	9	-0.000000	-0.000000	1.954457	-5.199338

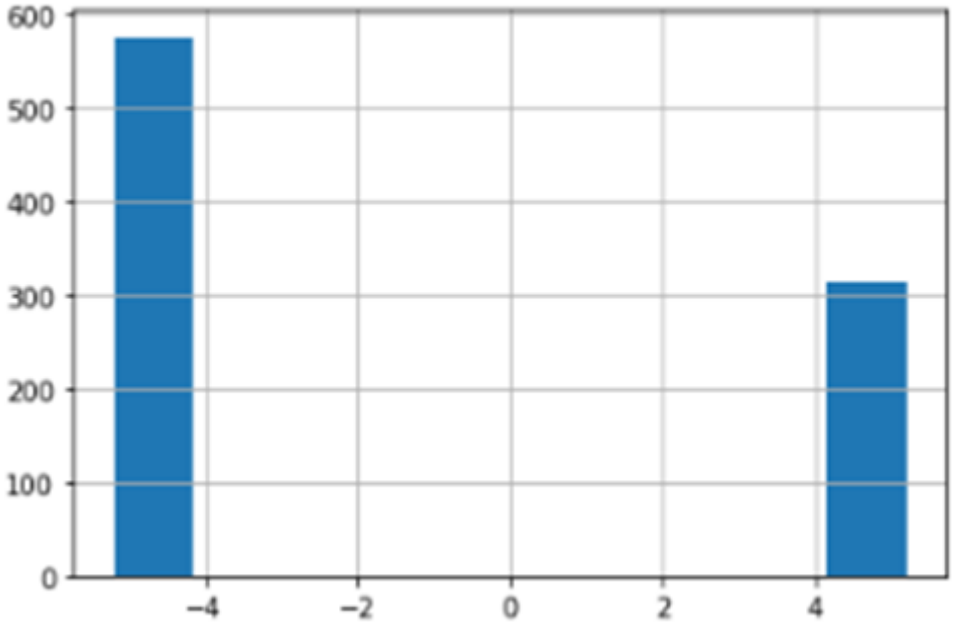
```
<AxesSubplot:>
```





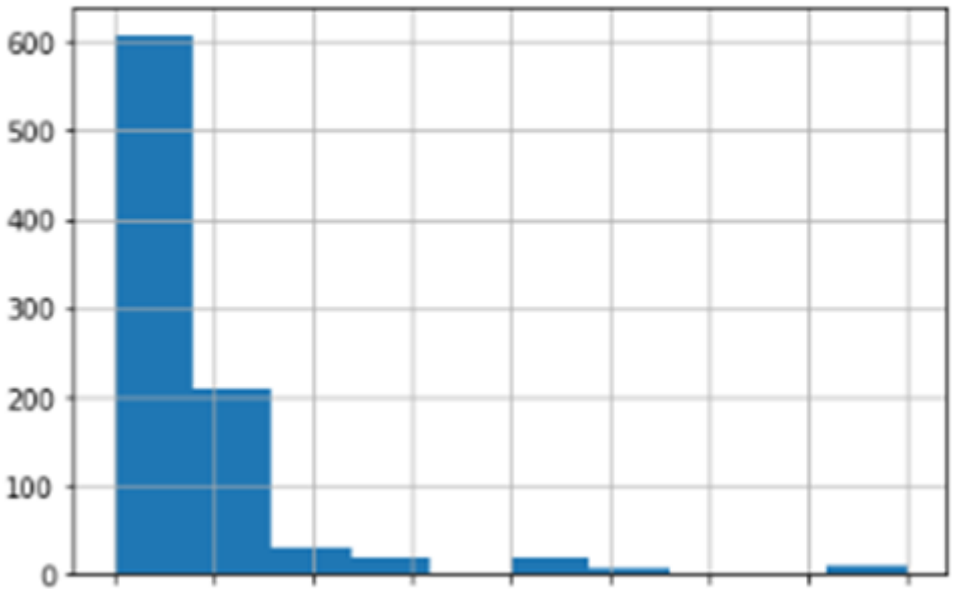

```
df["Sex_qt"].hist()
```

```
<AxesSubplot:>
```

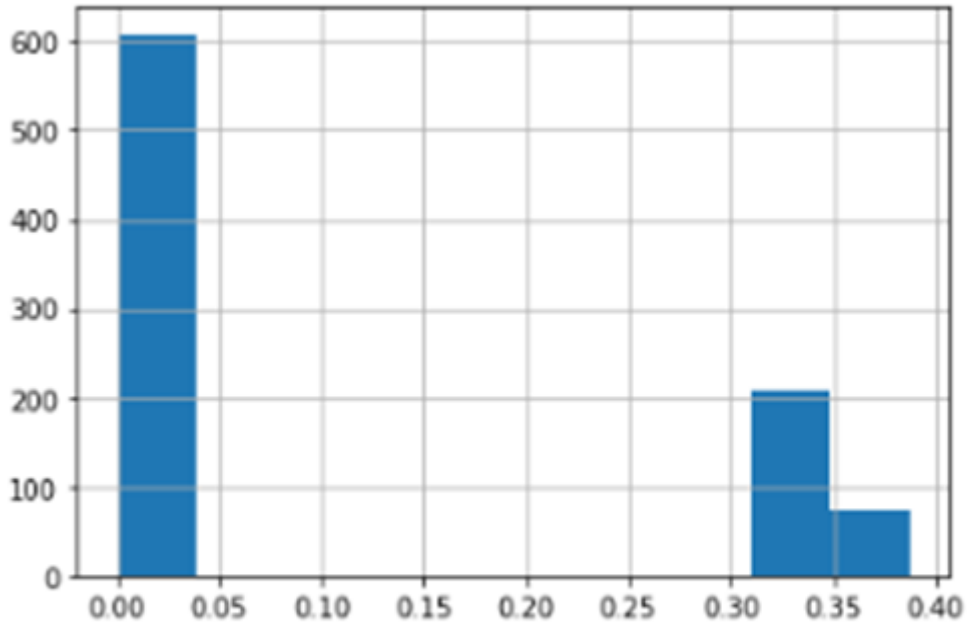



```
df["SibSp"].hist()
```

```
<AxesSubplot:>
```

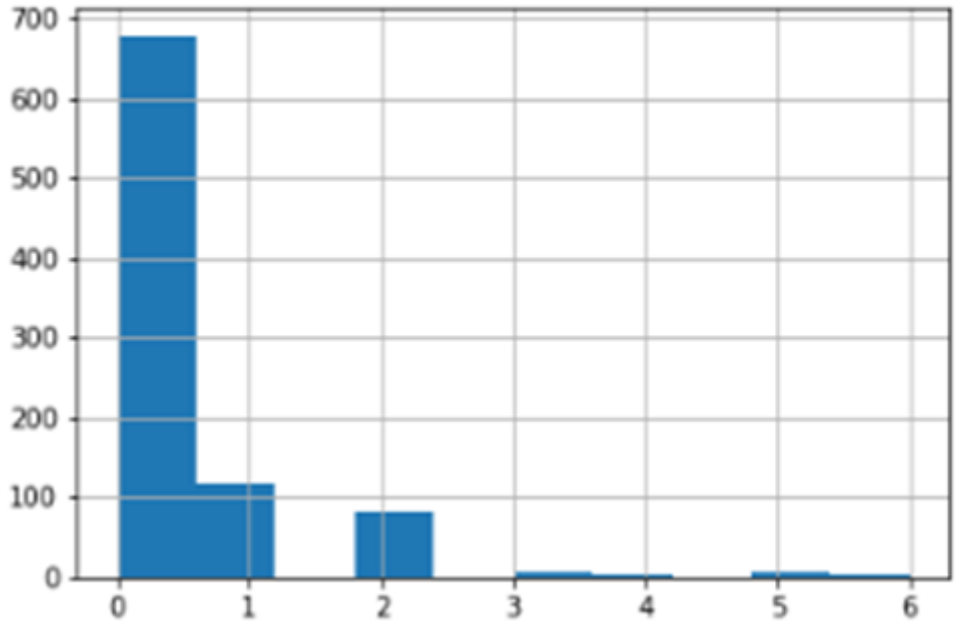


0 1 2 3 4 5 6 7 8



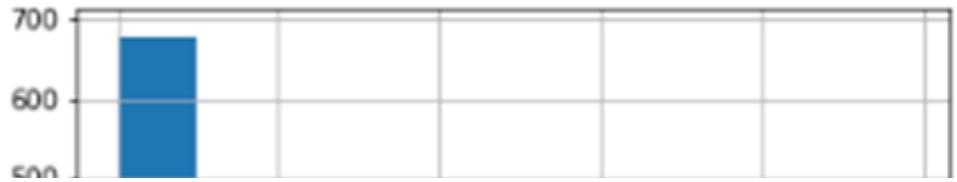

```
df["Parch"].hist()
```

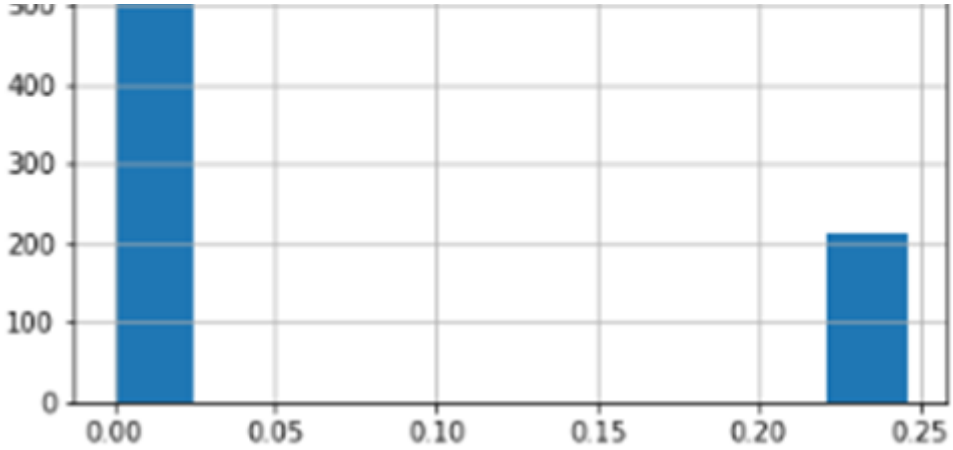
```
<AxesSubplot:>
```

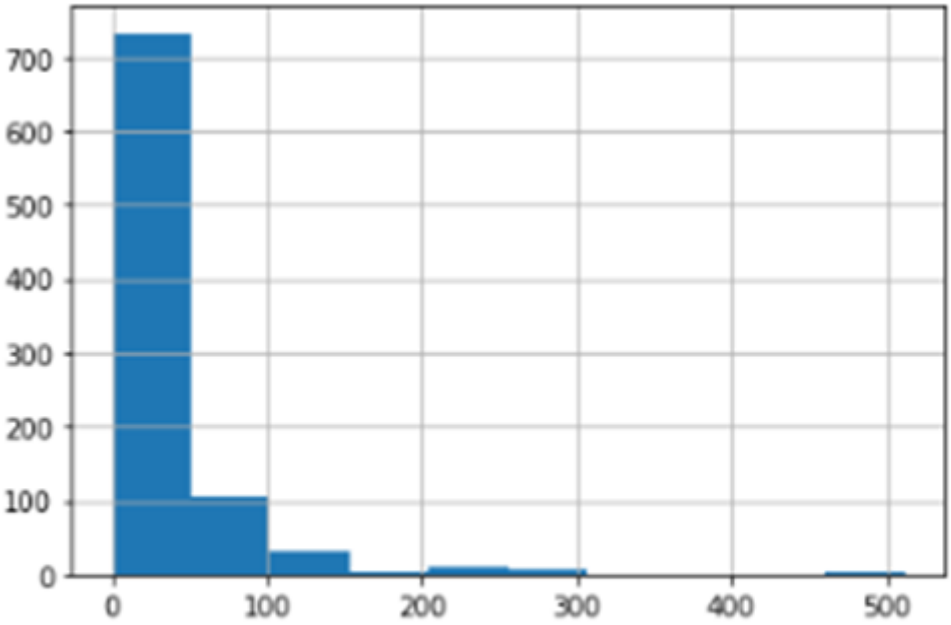



```
df["Parch_yj"].hist()
```

```
<AxesSubplot:>
```

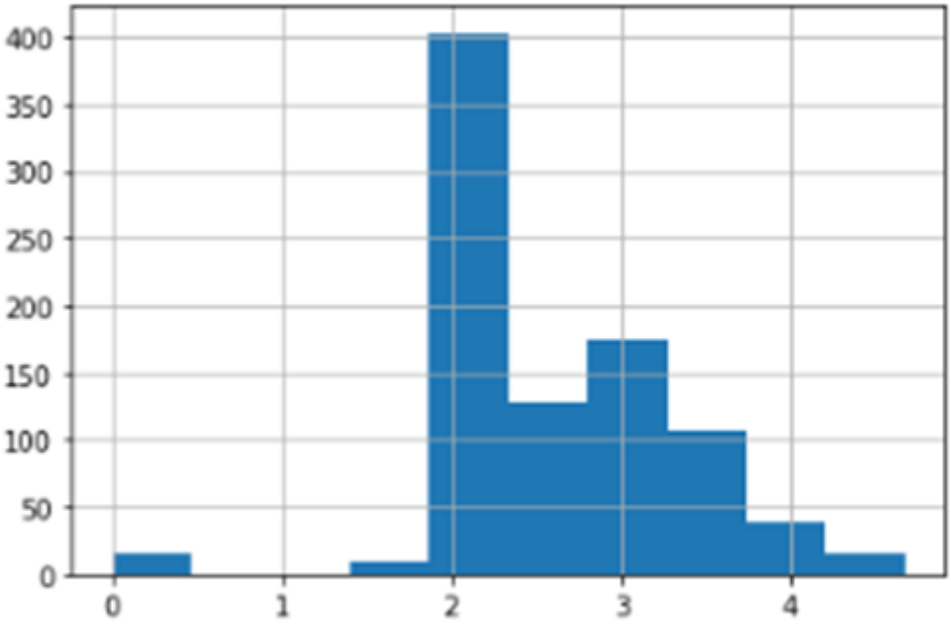






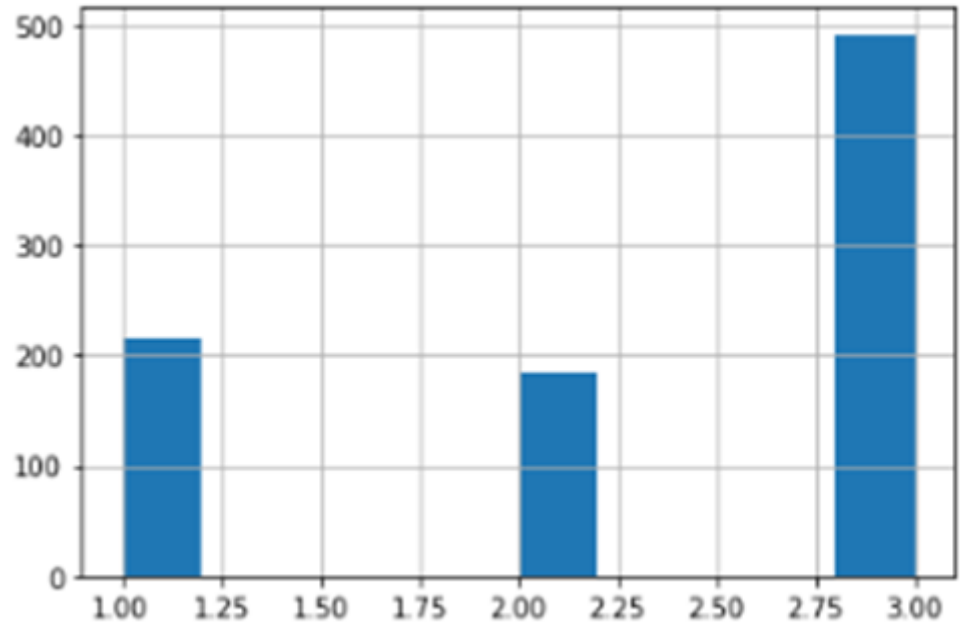

```
df["Fare_yj"].hist()
```

```
<AxesSubplot:>
```

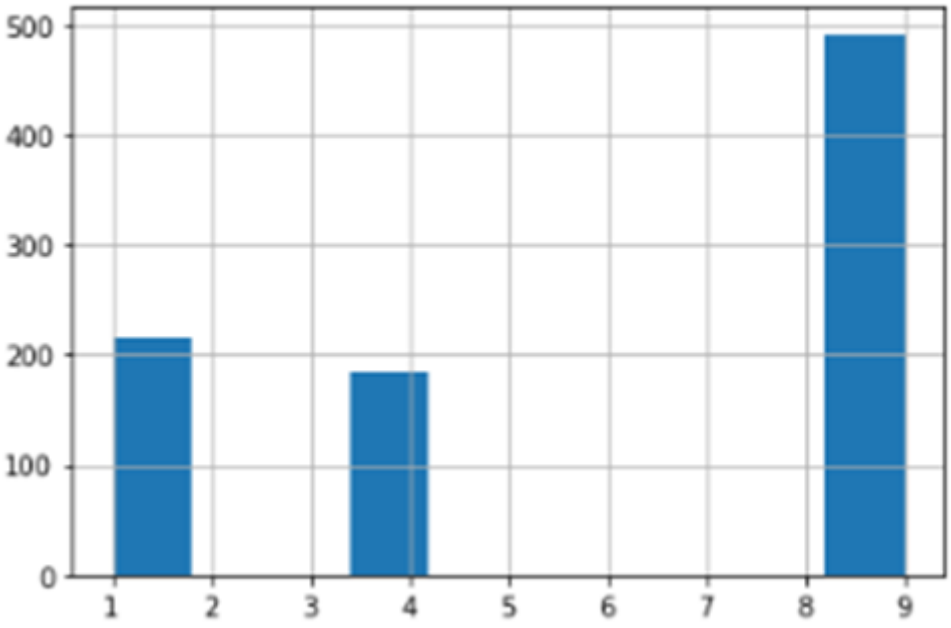


```
df["Pclass"].hist()
```

```
<AxesSubplot:>
```




```
<AxesSubplot:>
```



RESULT:

The various feature transformation techniques has been performed on the given datasets and the data are saved to a file.