In [93]:
```
import pandas as pd
import scipy
from scipy import stats
```

In [94]:
```
# Load dataset
ds = pd.read_csv("ant19_binary.csv")
ds.head()
```

Out[94]:

	COMP	LOC	WMC	DIT	NOC	CBO	RFC	LCO
0	org.apache.tools.ant.taskdefs.cvslib.CvsVersio...	317	7	0	0	3	32	
1	org.apache.tools.ant.util.regexp.JakartaRegexp...	96	3	0	0	5	16	
2	org.apache.tools.ant.taskdefs.GUnzip.java	119	4	0	0	5	23	
3	org.apache.tools.ant.taskdefs.condition.Equals...	209	11	1	0	3	18	
4	org.apache.tools.ant.taskdefs.optional.ccm.CCM...	19	1	0	0	1	9	

5 rows × 22 columns

In [95]:
```
# creating a Dataframe object
df = pd.DataFrame(ds)

# Z-Score using scipy
df['LOC'] = stats.zscore(df['LOC'])
df['WMC'] = stats.zscore(df['WMC'])
df['DIT'] = stats.zscore(df['DIT'])
df['NOC'] = stats.zscore(df['NOC'])
df['CBO'] = stats.zscore(df['CBO'])
df['RFC'] = stats.zscore(df['RFC'])
df['LCOM'] = stats.zscore(df['LCOM'])
df['CA'] = stats.zscore(df['CA'])
df['CE'] = stats.zscore(df['CE'])
df['NPM'] = stats.zscore(df['NPM'])
df['LCOM3'] = stats.zscore(df['LCOM3'])
df['DAM'] = stats.zscore(df['DAM'])
df['MOA'] = stats.zscore(df['MOA'])
df['MFA'] = stats.zscore(df['MFA'])
df['CAM'] = stats.zscore(df['CAM'])
df['IC'] = stats.zscore(df['IC'])
df['CBM'] = stats.zscore(df['CBM'])
df['AMC'] = stats.zscore(df['AMC'])
df['CC'] = stats.zscore(df['CC'])
df['MaX_CC'] = stats.zscore(df['MaX_CC'])
#df['Sum_Churn'] = stats.zscore(df['Sum_Churn'])
```

In [96]:
```
#drop LOC outliers
threshold = 3.29
zLOC=df['LOC']
outliers = df[zLOC > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [97]:
```
#drop WMC outliers
threshold = 3.29
zWMC=df['WMC']
outliers = df[zWMC > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [98]:
```
#drop DIT outliers
threshold = 3.29
zDIT=df['DIT']
outliers = df[zDIT > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [99]:
```
#drop NOC outliers
threshold = 3.29
zNOC=df['NOC']
outliers = df[zNOC > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [100]:
```
#drop CBO outliers
threshold = 3.29
zCBO=df['CBO']
outliers = df[zCBO > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [101]:
```
#drop RFC outliers
threshold = 3.29
zRFC=df['RFC']
outliers = df[zRFC > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [102]:
```
#drop LCOM outliers
threshold = 3.29
zLCOM=df['LCOM']
outliers = df[zLCOM > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [103]:
```
#drop CA outliers
threshold = 3.29
zCA=df['CA']
outliers = df[zCA > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [104]:
```
#drop CE outliers
threshold = 3.29
zCE=df['CE']
outliers = df[zCE > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [105]:
```
#drop NPM outliers
threshold = 3.29
zNPM=df['NPM']
outliers = df[zNPM > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [106]:
```
#drop LCOM3 outliers
threshold = 3.29
zLCOM3=df['LCOM3']
outliers = df[zLCOM3 > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [107]:
```
#drop DAM outliers
threshold = 3.29
zDAM=df['DAM']
outliers = df[zDAM > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [108]:
```
#drop MOA outliers
threshold = 3.29
zMOA=df['MOA']
outliers = df[zMOA > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [109]:
```
#drop MFA outliers
threshold = 3.29
zMFA=df['MFA']
outliers = df[zMFA > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [110]:
```
#drop CAM outliers
threshold = 3.29
zCAM=df['CAM']
outliers = df[zCAM > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [111]:
```
#drop IC outliers
threshold = 3.29
zIC=df['IC']
outliers = df[zIC > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [112]:
```
#drop CBM outliers
threshold = 3.29
zCBM=df['CBM']
outliers = df[zCBM > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [113]:
```
#drop AMC outliers
threshold = 3.29
zAMC=df['AMC']
outliers = df[zAMC > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [114]:
```
#drop CC outliers
threshold = 3.29
zCC=df['CC']
outliers = df[zCC > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [115]:
```
#drop MaX_CC outliers
threshold = 3.29
zMaX_CC=df['MaX_CC']
outliers = df[zMaX_CC > threshold]

# Print the outliers
#print(outliers)

# drop rows containing outliers
df = df.drop(outliers.index)
```

In [116]:
```
df.to_excel('Ant19_Standardized_NoOutliers.xlsx', index=False)
```

In [117]:
```
#try to refactor using lool
# replace outliers with median value
#df.loc[z > threshold, 'Height'] = df['Height'].median()
```

In [ ]:
```

```