In [ ]:

```
#import all packages and liabraries
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import pandas as pd
import seaborn as sns

#To ignore the warnings
import warnings
warnings.filterwarnings('ignore')

#To display all columns and row
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

In [1]:

```
import pandas as pd

# Read the CSV file into a DataFrame
df = pd.read_csv("data.csv")

# Display the first few rows of the DataFrame
print(df.head())
```

```
         id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean
\
0    842302         M        17.99         10.38          122.80     1001.0
1    842517         M        20.57         17.77          132.90     1326.0
2  84300903         M        19.69         21.25          130.00     1203.0
3  84348301         M        11.42         20.38           77.58      386.1
4  84358402         M        20.29         14.34          135.10     1297.0

   smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0          0.11840           0.27760          0.3001              0.14710
1          0.08474           0.07864          0.0869              0.07017
2          0.10960           0.15990          0.1974              0.12790
3          0.14250           0.28390          0.2414              0.10520
4          0.10030           0.13280          0.1980              0.10430

   ...  texture_worst  perimeter_worst  area_worst  smoothness_worst  \
0  ...          17.33           184.60      2019.0            0.1622
1  ...          23.41           158.80      1956.0            0.1238
2  ...          25.53           152.50      1709.0            0.1444
3  ...          26.50            98.87       567.7            0.2098
4  ...          16.67           152.20      1575.0            0.1374

   compactness_worst  concavity_worst  concave points_worst  symmetry_worst
\
0             0.6656           0.7119                0.2654          0.4601
1             0.1866           0.2416                0.1860          0.2750
2             0.4245           0.4504                0.2430          0.3613
3             0.8663           0.6869                0.2575          0.6638
4             0.2050           0.4000                0.1625          0.2364

   fractal_dimension_worst  Unnamed: 32
0                  0.11890          NaN
1                  0.08902          NaN
2                  0.08758          NaN
3                  0.17300          NaN
4                  0.07678          NaN

[5 rows x 33 columns]
```

```
In [2]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   id                       569 non-null    int64  
 1   diagnosis                569 non-null    object 
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
 32  Unnamed: 32              0 non-null      float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
```

In [3]:

```
df.columns
```

Out[3]:

```
Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mea
n',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_s
e',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_s
e',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst', 'Unnamed: 32'],
      dtype='object')
```

```
In [4]: df.shape
```

```
Out[4]: (569, 33)
```

```
In [6]: df.describe().T
```

Out[6]:

| | count | mean | std | min | 25% |
|---|---|---|---|---|---|
| **id** | 569.0 | 3.037183e+07 | 1.250206e+08 | 8670.000000 | 869218.000000 |
| **radius_mean** | 569.0 | 1.412729e+01 | 3.524049e+00 | 6.981000 | 11.700000 |
| **texture_mean** | 569.0 | 1.928965e+01 | 4.301036e+00 | 9.710000 | 16.170000 |
| **perimeter_mean** | 569.0 | 9.196903e+01 | 2.429898e+01 | 43.790000 | 75.170000 |
| **area_mean** | 569.0 | 6.548891e+02 | 3.519141e+02 | 143.500000 | 420.300000 |
| **smoothness_mean** | 569.0 | 9.636028e-02 | 1.406413e-02 | 0.052630 | 0.086370 |
| **compactness_mean** | 569.0 | 1.043410e-01 | 5.281276e-02 | 0.019380 | 0.064920 |
| **concavity_mean** | 569.0 | 8.879932e-02 | 7.971981e-02 | 0.000000 | 0.029560 |
| **concave points_mean** | 569.0 | 4.891915e-02 | 3.880284e-02 | 0.000000 | 0.020310 |
| **symmetry_mean** | 569.0 | 1.811619e-01 | 2.741428e-02 | 0.106000 | 0.161900 |
| **fractal_dimension_mean** | 569.0 | 6.279761e-02 | 7.060363e-03 | 0.049960 | 0.057700 |
| **radius_se** | 569.0 | 4.051721e-01 | 2.773127e-01 | 0.111500 | 0.232400 |
| **texture_se** | 569.0 | 1.216853e+00 | 5.516484e-01 | 0.360200 | 0.833900 |
| **perimeter_se** | 569.0 | 2.866059e+00 | 2.021855e+00 | 0.757000 | 1.606000 |
| **area_se** | 569.0 | 4.033708e+01 | 4.549101e+01 | 6.802000 | 17.850000 |
| **smoothness_se** | 569.0 | 7.040979e-03 | 3.002518e-03 | 0.001713 | 0.005169 |
| **compactness_se** | 569.0 | 2.547814e-02 | 1.790818e-02 | 0.002252 | 0.013080 |
| **concavity_se** | 569.0 | 3.189372e-02 | 3.018606e-02 | 0.000000 | 0.015090 |
| **concave points_se** | 569.0 | 1.179614e-02 | 6.170285e-03 | 0.000000 | 0.007638 |
| **symmetry_se** | 569.0 | 2.054230e-02 | 8.266372e-03 | 0.007882 | 0.015160 |
| **fractal_dimension_se** | 569.0 | 3.794904e-03 | 2.646071e-03 | 0.000895 | 0.002248 |
| **radius_worst** | 569.0 | 1.626919e+01 | 4.833242e+00 | 7.930000 | 13.010000 |
| **texture_worst** | 569.0 | 2.567722e+01 | 6.146258e+00 | 12.020000 | 21.080000 |
| **perimeter_worst** | 569.0 | 1.072612e+02 | 3.360254e+01 | 50.410000 | 84.110000 |
| **area_worst** | 569.0 | 8.805831e+02 | 5.693570e+02 | 185.200000 | 515.300000 |
| **smoothness_worst** | 569.0 | 1.323686e-01 | 2.283243e-02 | 0.071170 | 0.116600 |
| **compactness_worst** | 569.0 | 2.542650e-01 | 1.573365e-01 | 0.027290 | 0.147200 |
| **concavity_worst** | 569.0 | 2.721885e-01 | 2.086243e-01 | 0.000000 | 0.114500 |
| **concave points_worst** | 569.0 | 1.146062e-01 | 6.573234e-02 | 0.000000 | 0.064930 |
| **symmetry_worst** | 569.0 | 2.900756e-01 | 6.186747e-02 | 0.156500 | 0.250400 |
| **fractal_dimension_worst** | 569.0 | 8.394582e-02 | 1.806127e-02 | 0.055040 | 0.071460 |
| **Unnamed: 32** | 0.0 | NaN | NaN | NaN | NaN |

Loading [MathJax]/extensions/Safe.js

In [7]:

```
df.dtypes
```

Out[7]:

```
id                           int64
diagnosis                   object
radius_mean                float64
texture_mean               float64
perimeter_mean             float64
area_mean                  float64
smoothness_mean            float64
compactness_mean           float64
concavity_mean             float64
concave points_mean        float64
symmetry_mean              float64
fractal_dimension_mean     float64
radius_se                  float64
texture_se                 float64
perimeter_se               float64
area_se                    float64
smoothness_se              float64
compactness_se             float64
concavity_se               float64
concave points_se          float64
symmetry_se                float64
fractal_dimension_se       float64
radius_worst               float64
texture_worst              float64
perimeter_worst            float64
area_worst                 float64
smoothness_worst           float64
compactness_worst          float64
concavity_worst            float64
concave points_worst       float64
symmetry_worst             float64
fractal_dimension_worst    float64
Unnamed: 32                float64
dtype: object
```

In [8]:

```
df.isnull().sum()
```

```
Out[8]: id                           0
        diagnosis                    0
        radius_mean                  0
        texture_mean                 0
        perimeter_mean               0
        area_mean                    0
        smoothness_mean              0
        compactness_mean             0
        concavity_mean               0
        concave points_mean          0
        symmetry_mean                0
        fractal_dimension_mean       0
        radius_se                    0
        texture_se                   0
        perimeter_se                 0
        area_se                      0
        smoothness_se                0
        compactness_se               0
        concavity_se                 0
        concave points_se            0
        symmetry_se                  0
        fractal_dimension_se         0
        radius_worst                 0
        texture_worst                0
        perimeter_worst              0
        area_worst                   0
        smoothness_worst             0
        compactness_worst            0
        concavity_worst              0
        concave points_worst         0
        symmetry_worst               0
        fractal_dimension_worst      0
        Unnamed: 32                569
        dtype: int64
```

```
In [9]: df.skew().sort_values(ascending=False)
```

```
C:\Users\intel\AppData\Local\Temp\ipykernel_2424\4024944668.py:1: FutureWarni
ng: The default value of numeric_only in DataFrame.skew is deprecated. In a f
uture version, it will default to False. In addition, specifying 'numeric_onl
y=None' is deprecated. Select only valid columns or specify the value of nume
ric_only to silence this warning.
  df.skew().sort_values(ascending=False)
```

Loading [MathJax]/extensions/Safe.js

Out[9]:

```
id                         6.473752
area_se                    5.447186
concavity_se               5.110463
fractal_dimension_se       3.923969
perimeter_se               3.443615
radius_se                  3.088612
smoothness_se              2.314450
symmetry_se                2.195133
compactness_se             1.902221
area_worst                 1.859373
fractal_dimension_worst    1.662579
texture_se                 1.646444
area_mean                  1.645732
compactness_worst          1.473555
concave points_se          1.444678
symmetry_worst             1.433928
concavity_mean             1.401180
fractal_dimension_mean     1.304489
compactness_mean           1.190123
concave points_mean        1.171180
concavity_worst            1.150237
perimeter_worst            1.128164
radius_worst               1.103115
perimeter_mean             0.990650
radius_mean                0.942380
symmetry_mean              0.725609
texture_mean               0.650450
texture_worst              0.498321
concave points_worst       0.492616
smoothness_mean            0.456324
smoothness_worst           0.415426
Unnamed: 32                     NaN
dtype: float64
```

In [ ]:

```
Exploratory Data Analysis (EDA)
1.Examine the data distribution
2.Handling missing values of the dataset(a most common issue with every data
3.Removing the insignificant columns
4.Removing duplicate data
5.Encoding the categorical variables
6.Skewness checking and removing the skewness from the data
```

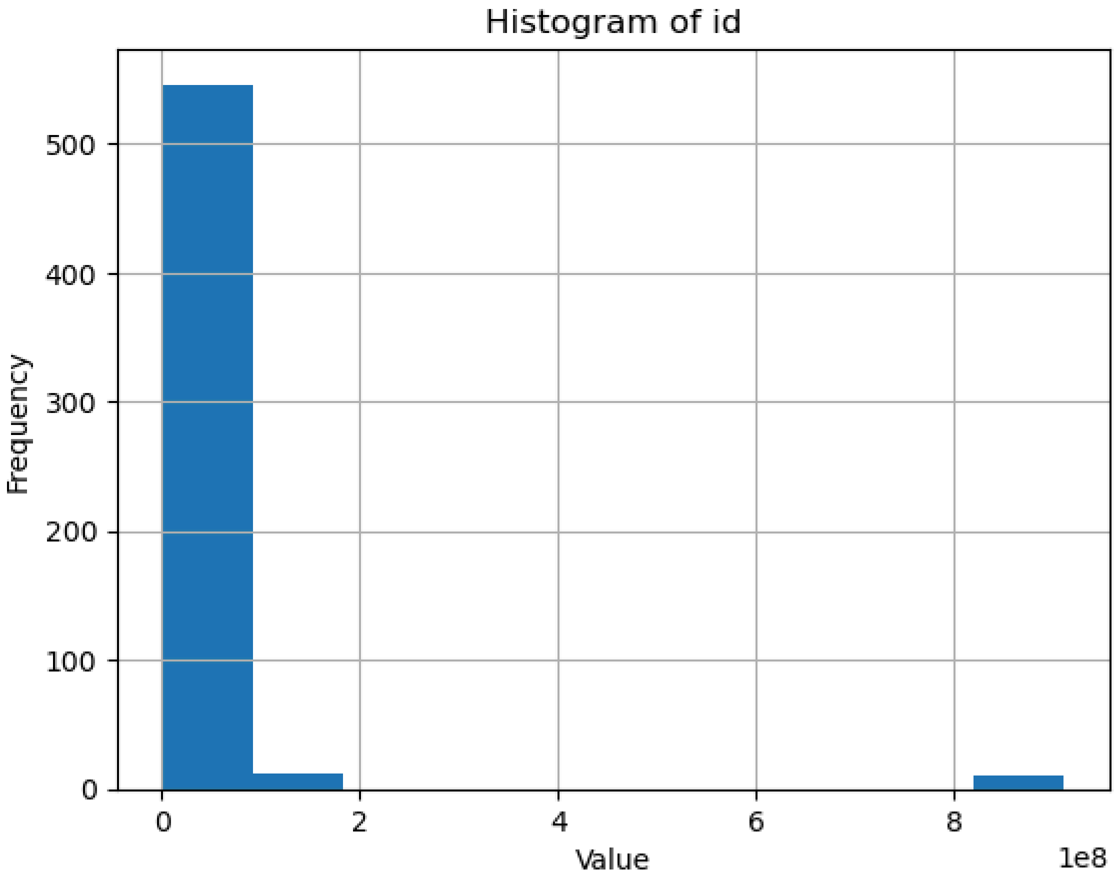
In [ ]:

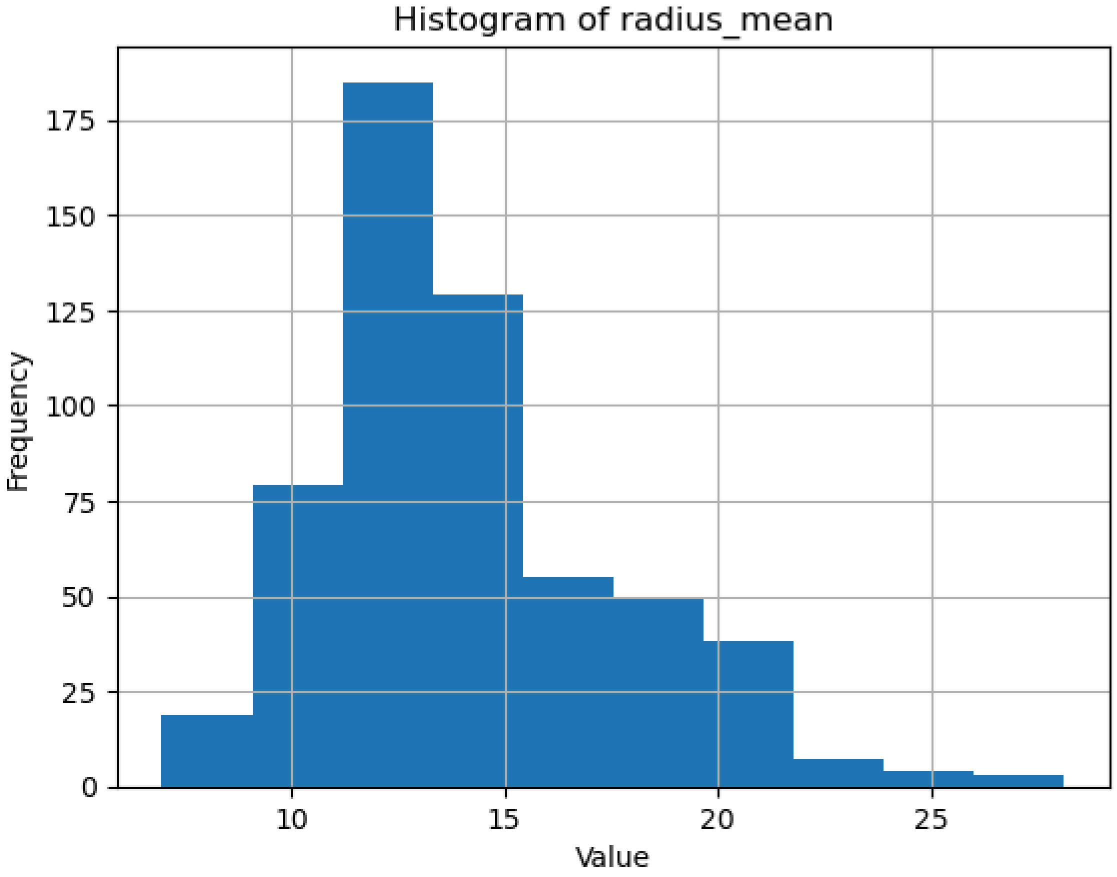
```
1.Examine the data distribution
```

In [10]:

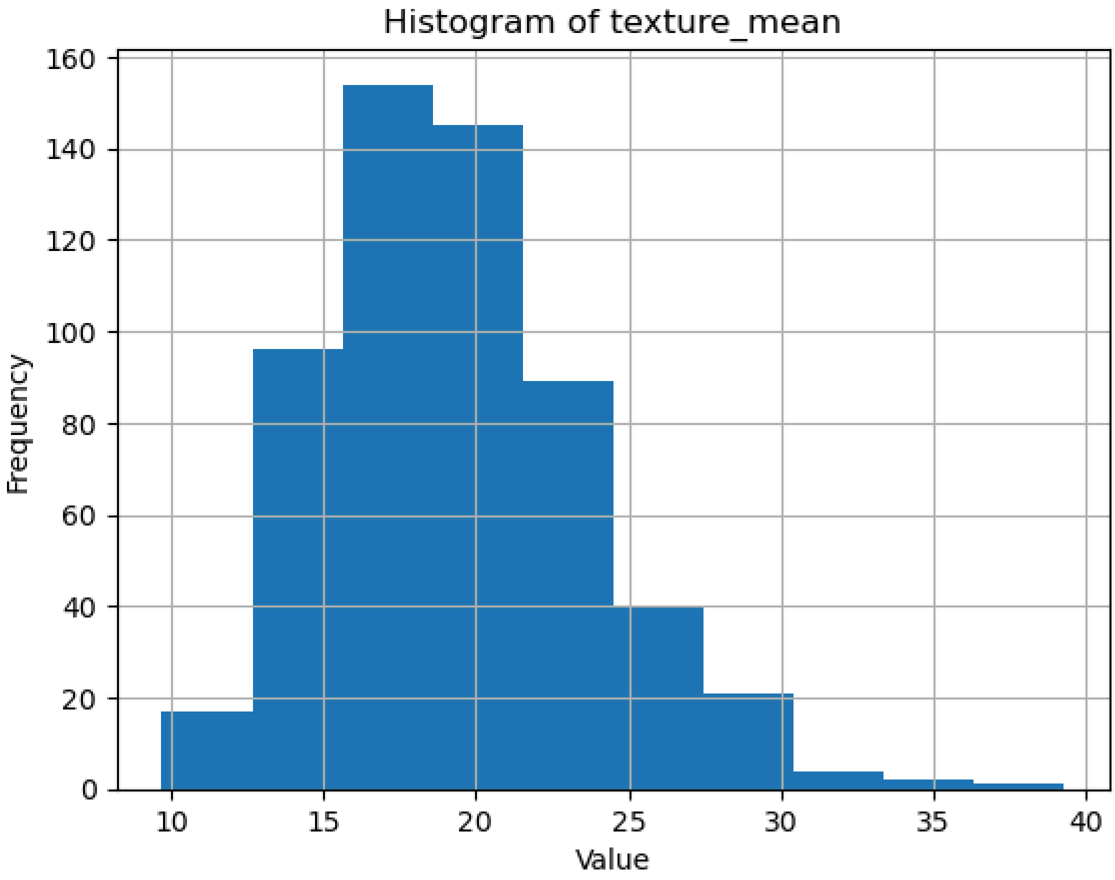
```python
import matplotlib.pyplot as plt

numerical_columns = df.select_dtypes(include=['int64', 'float64']).columns

for col in numerical_columns:
    df[col].hist()
    plt.title(f'Histogram of {col}')
    plt.xlabel('Value')
    plt.ylabel('Frequency')
    plt.show()
```

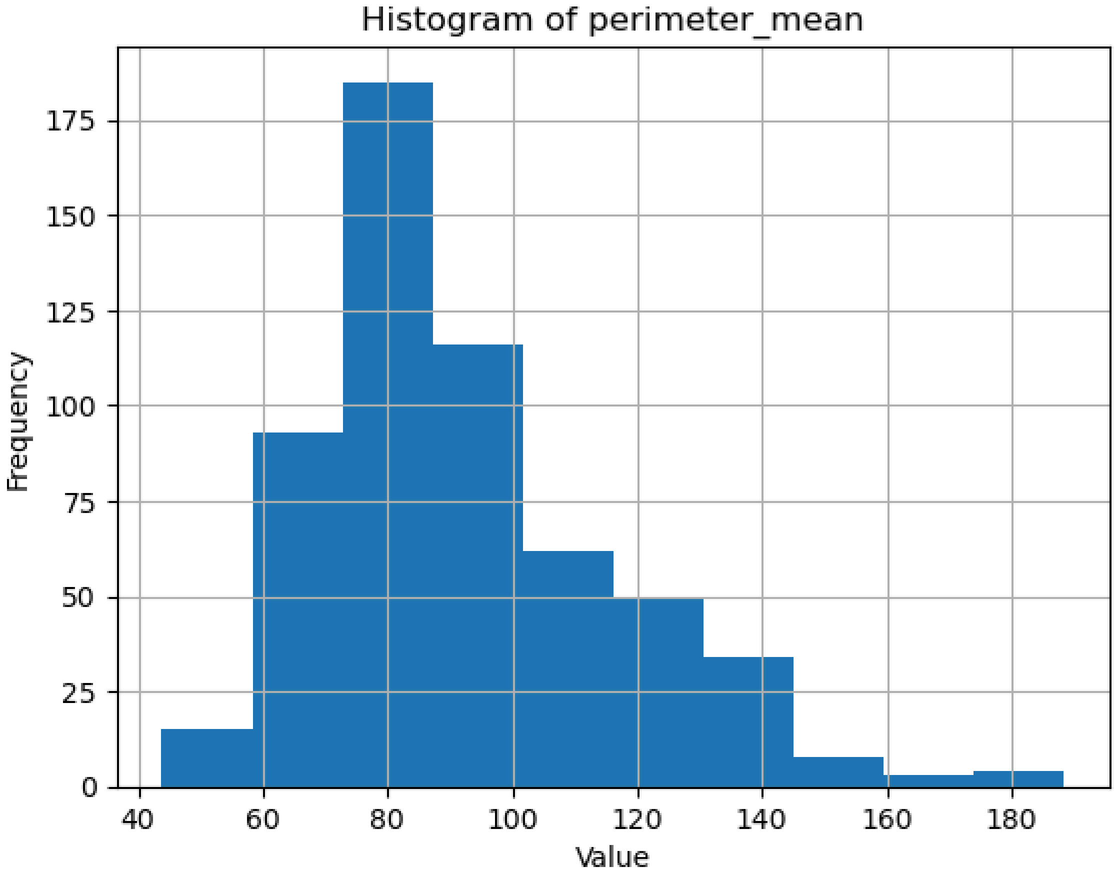
Loading [MathJax]/extensions/Safe.js



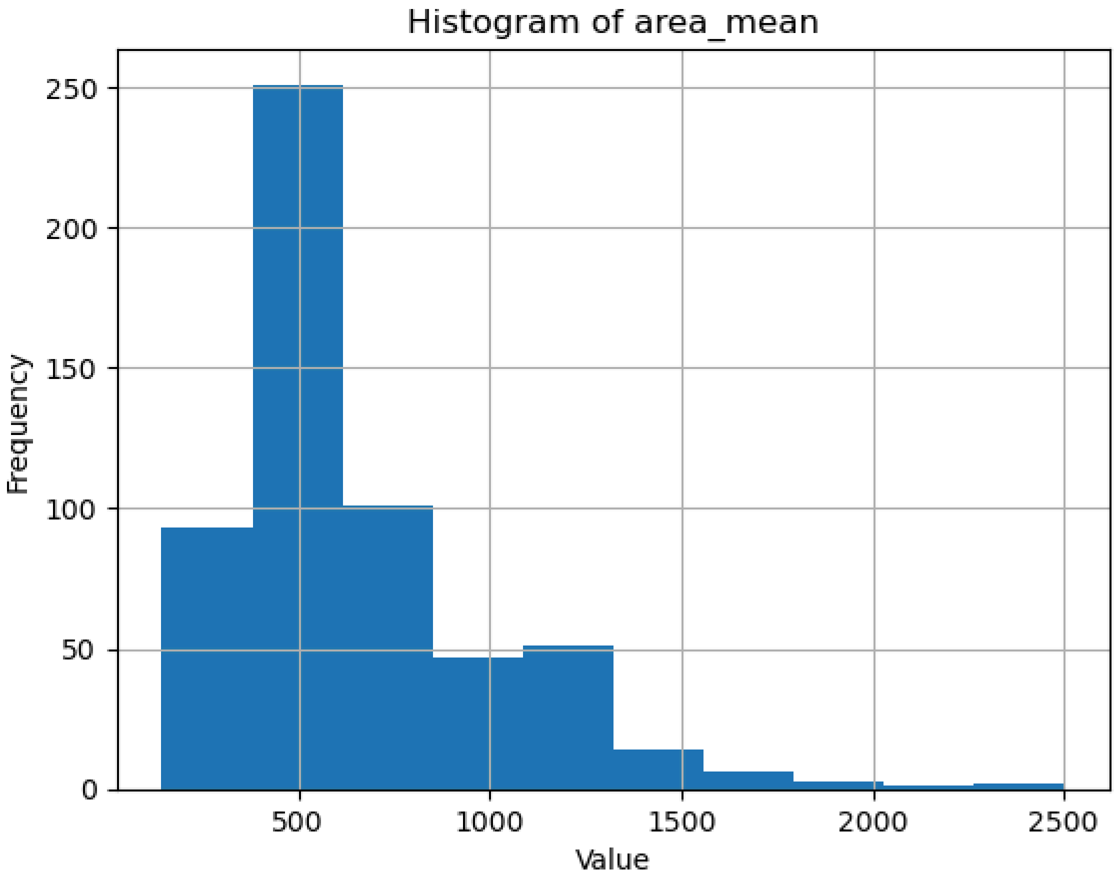


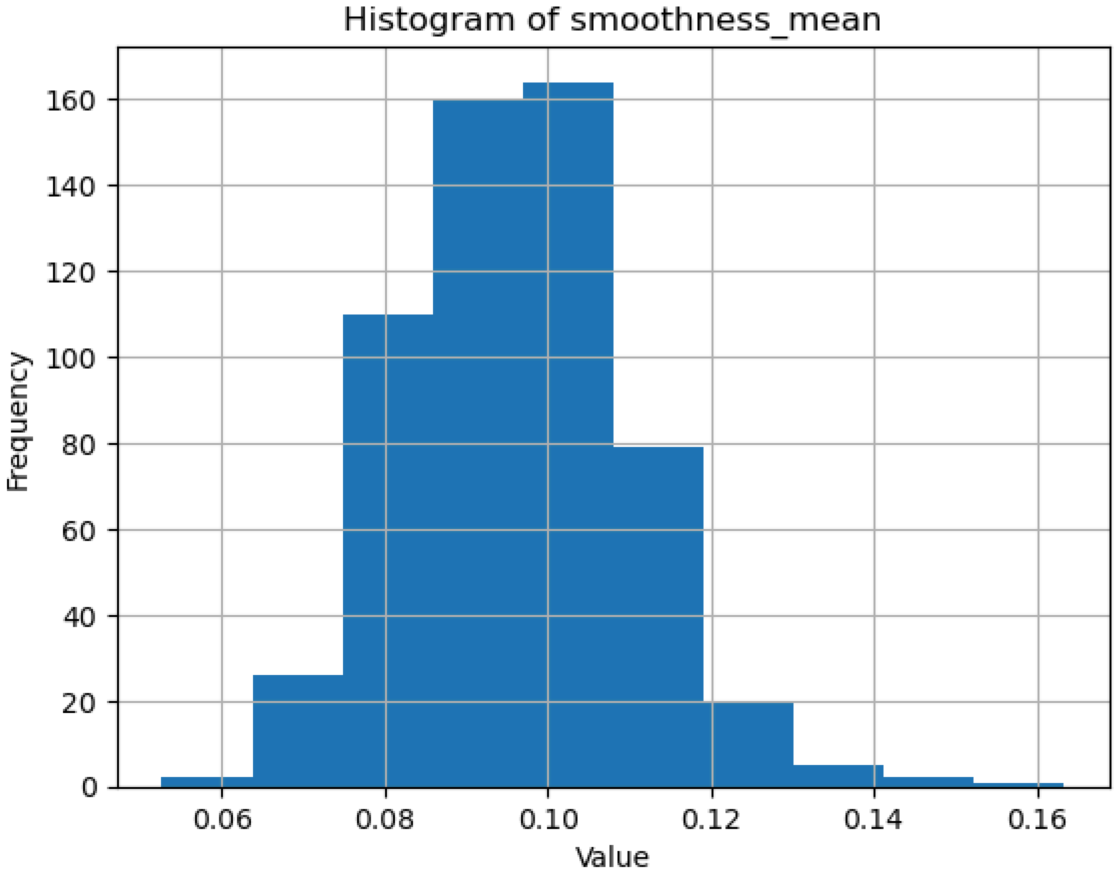
Loading [MathJax]/extensions/Safe.js

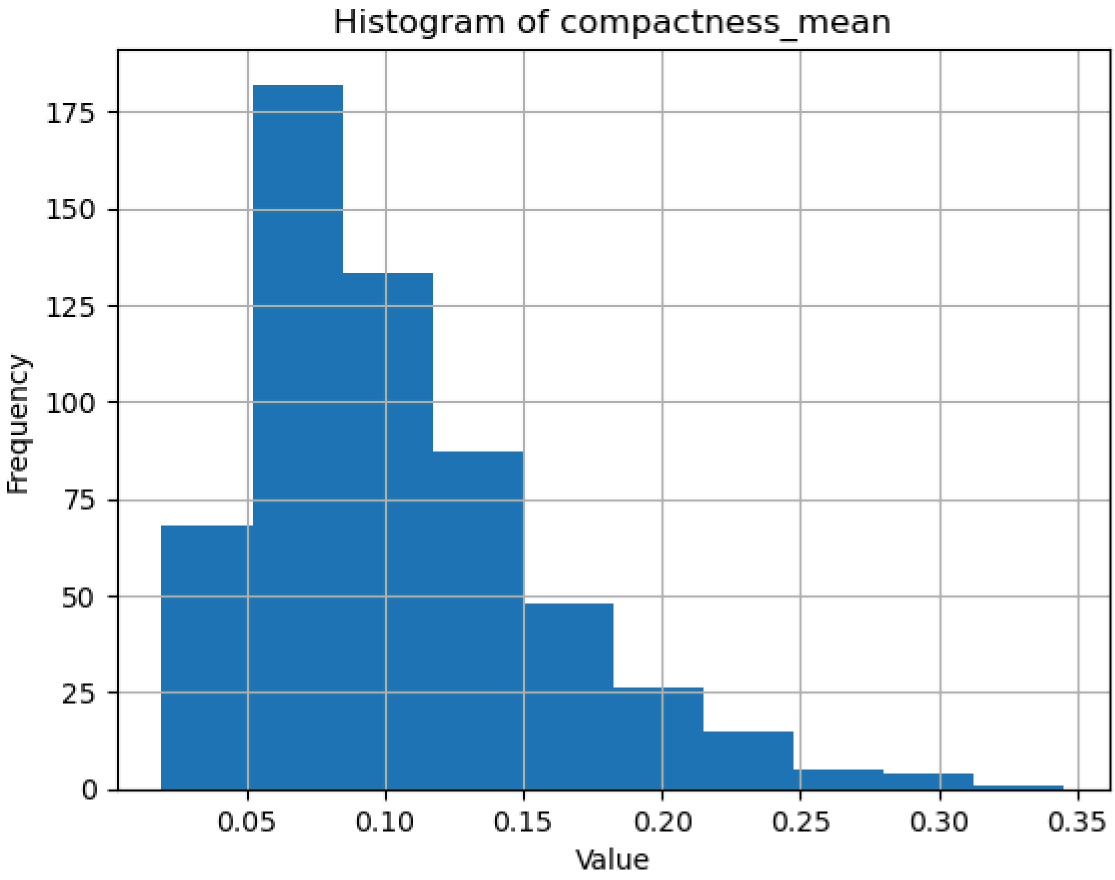


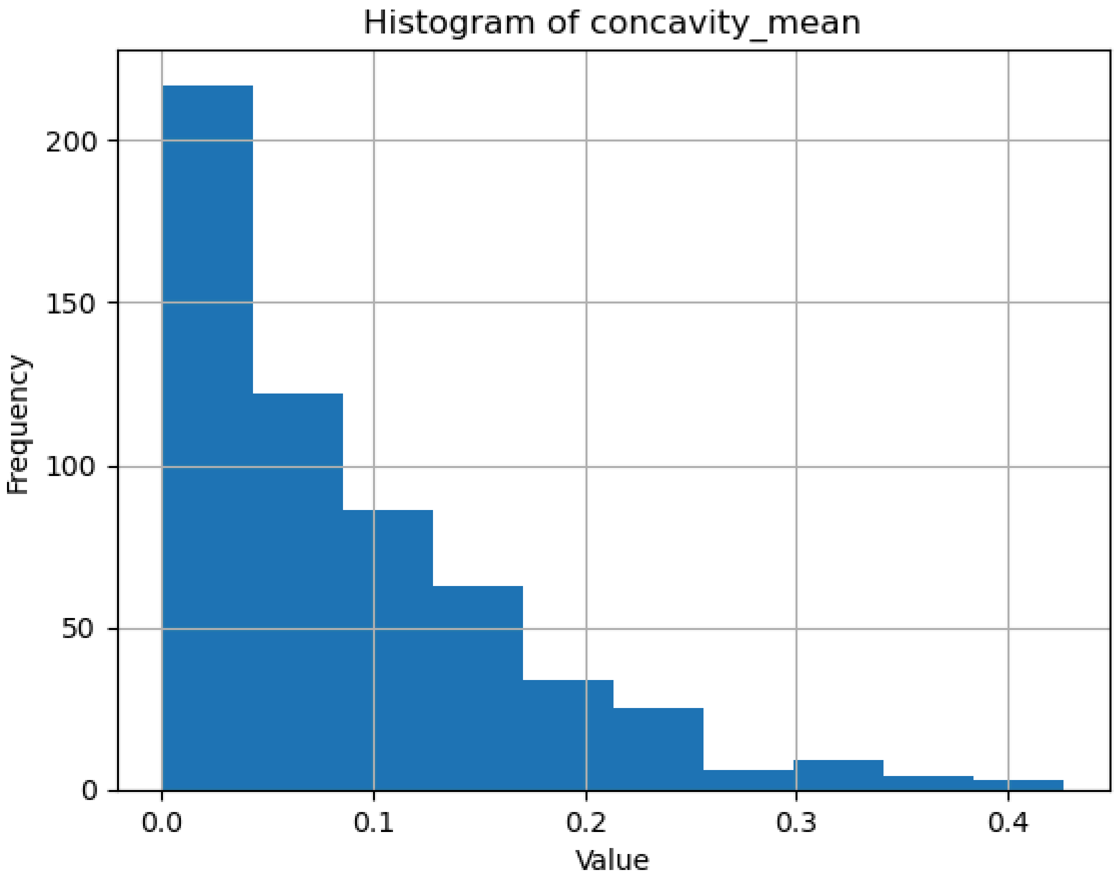


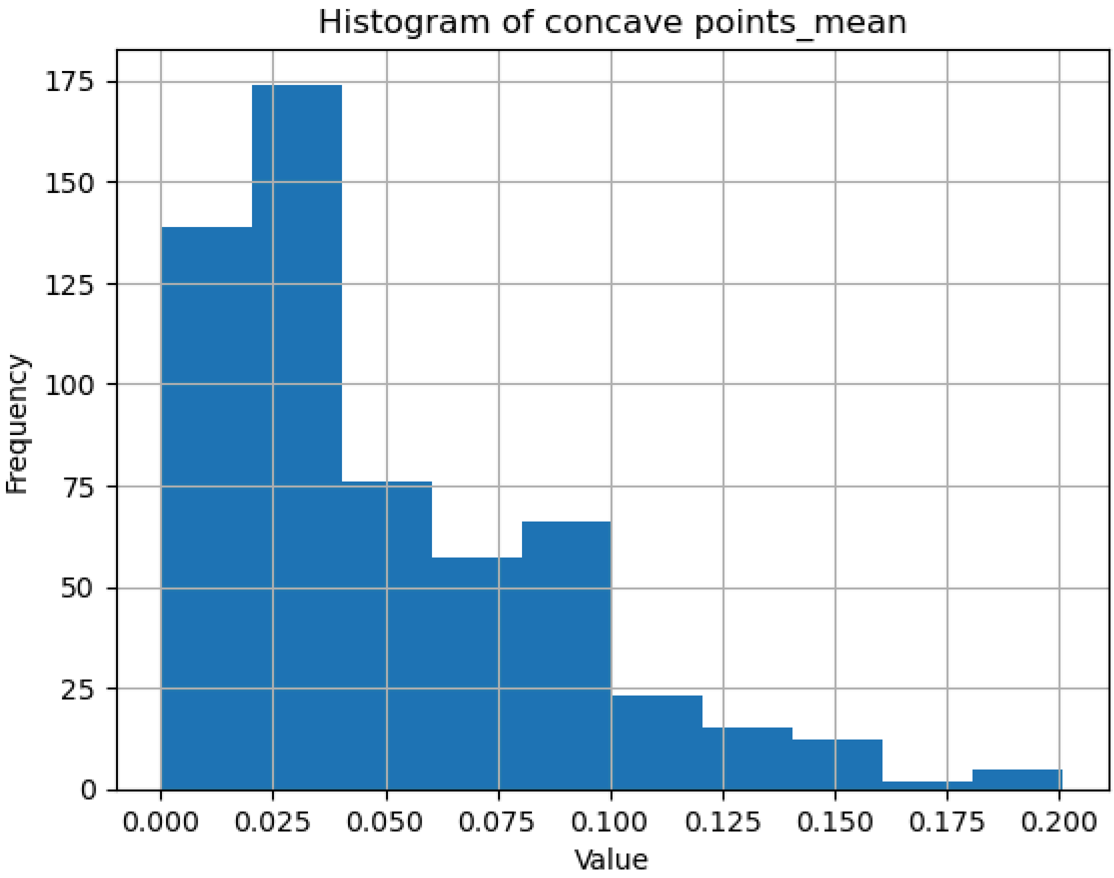
Loading [MathJax]/extensions/Safe.js

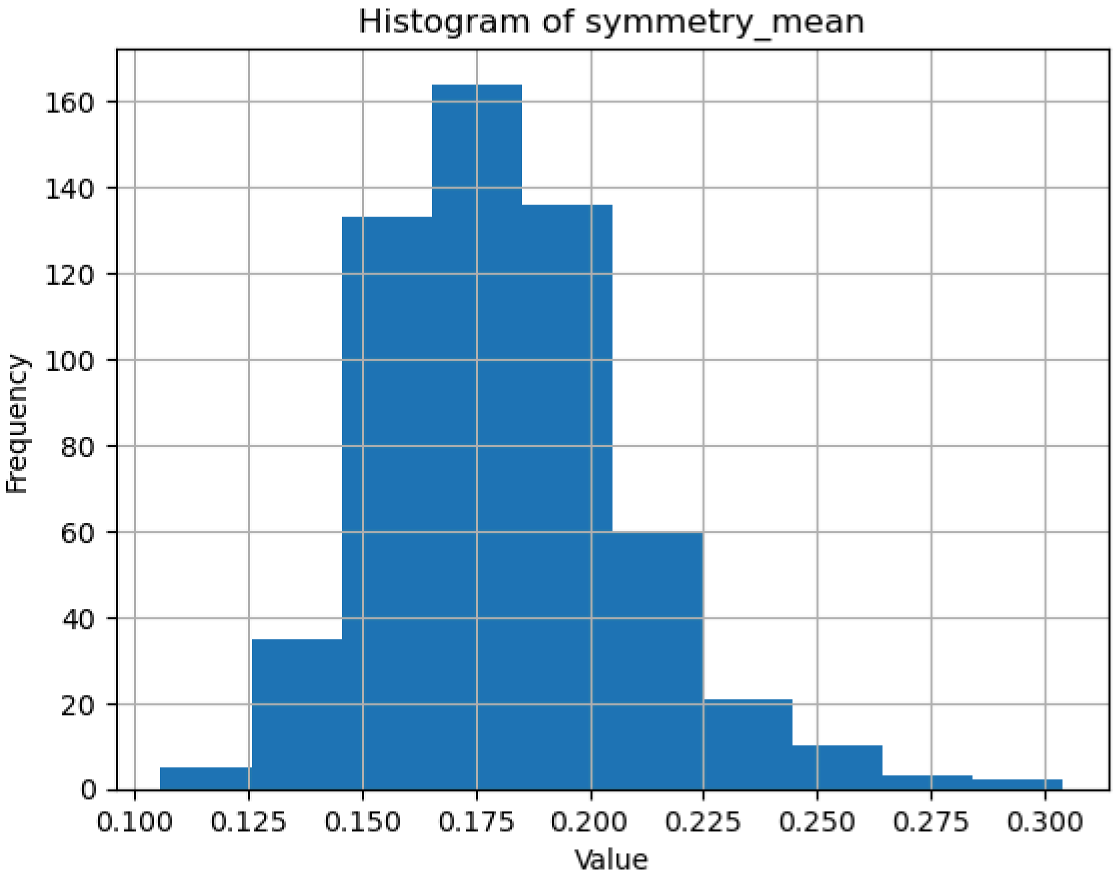

Histogram of area_mean

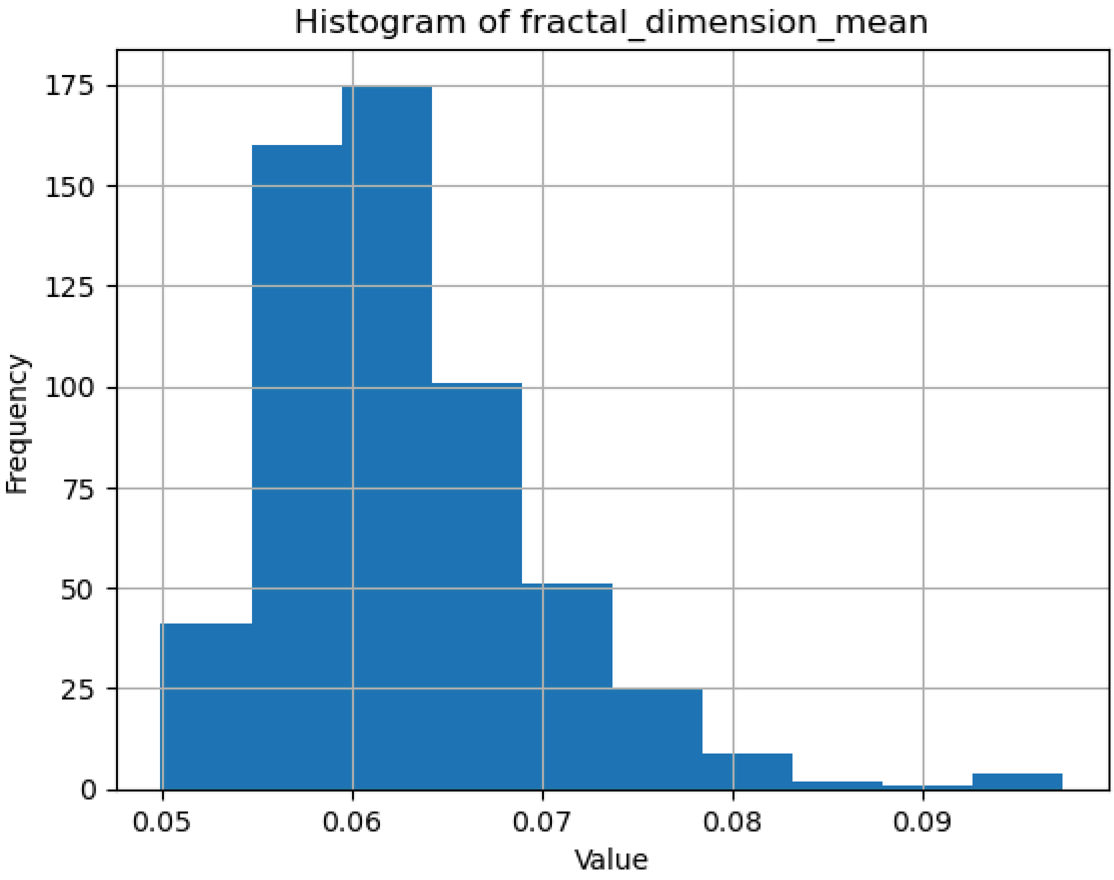

Histogram of smoothness_mean

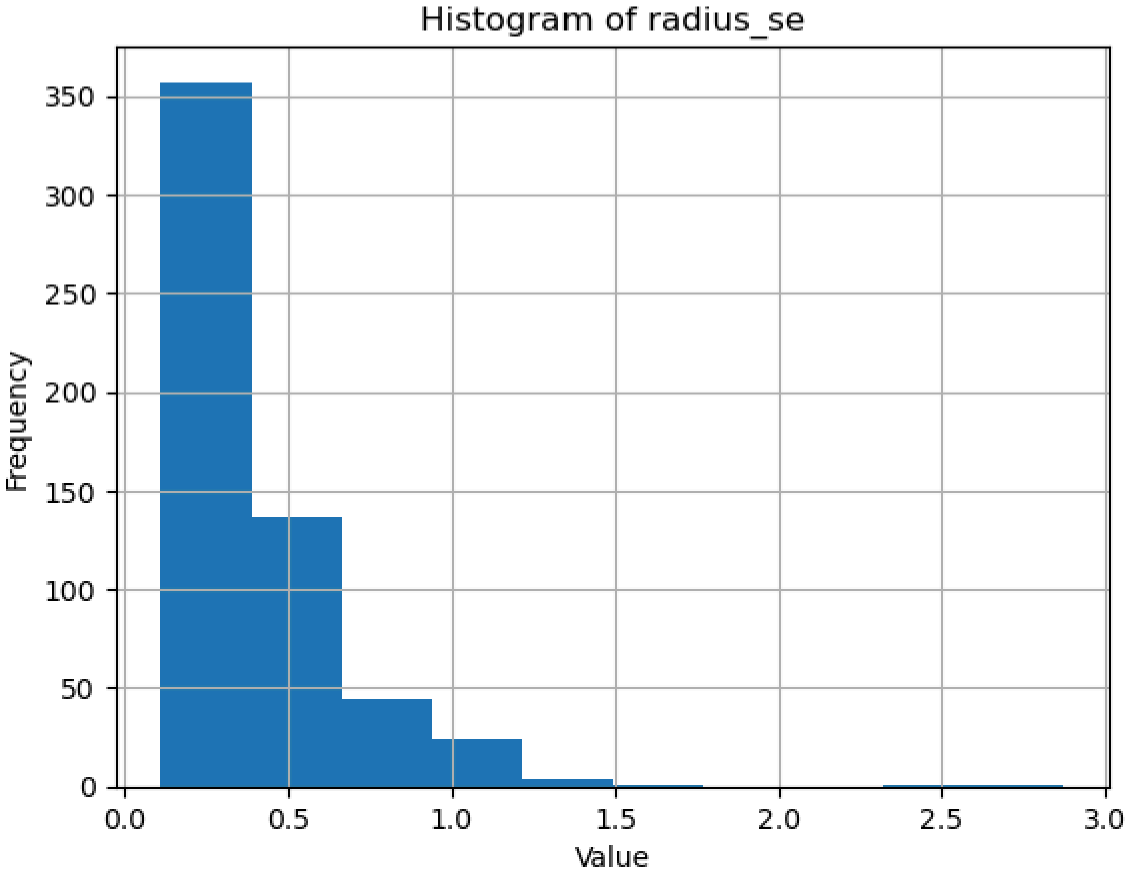

Histogram of compactness_mean
Frequency
175
150
125
100
75
50
25
0
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Value

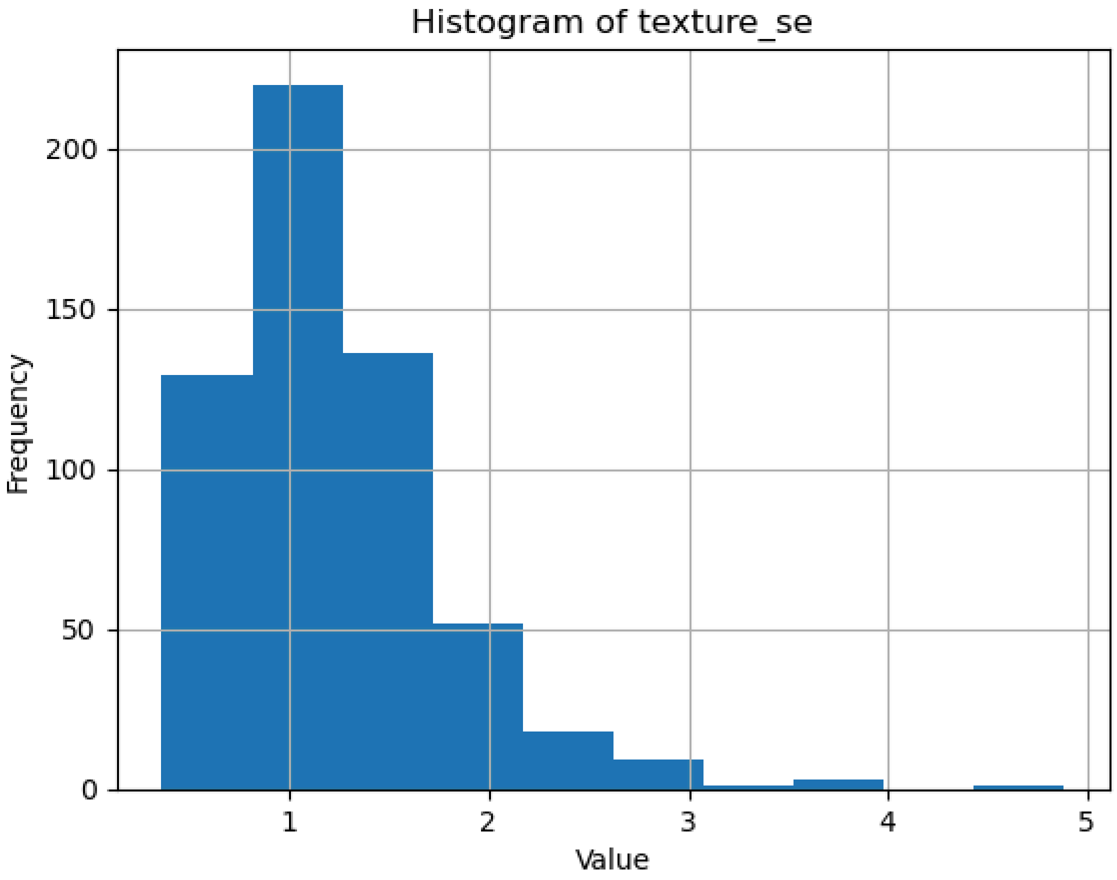

Histogram of concavity_mean
Frequency
200
150
100
50
0
0.0
0.1
0.2
0.3
0.4
Value

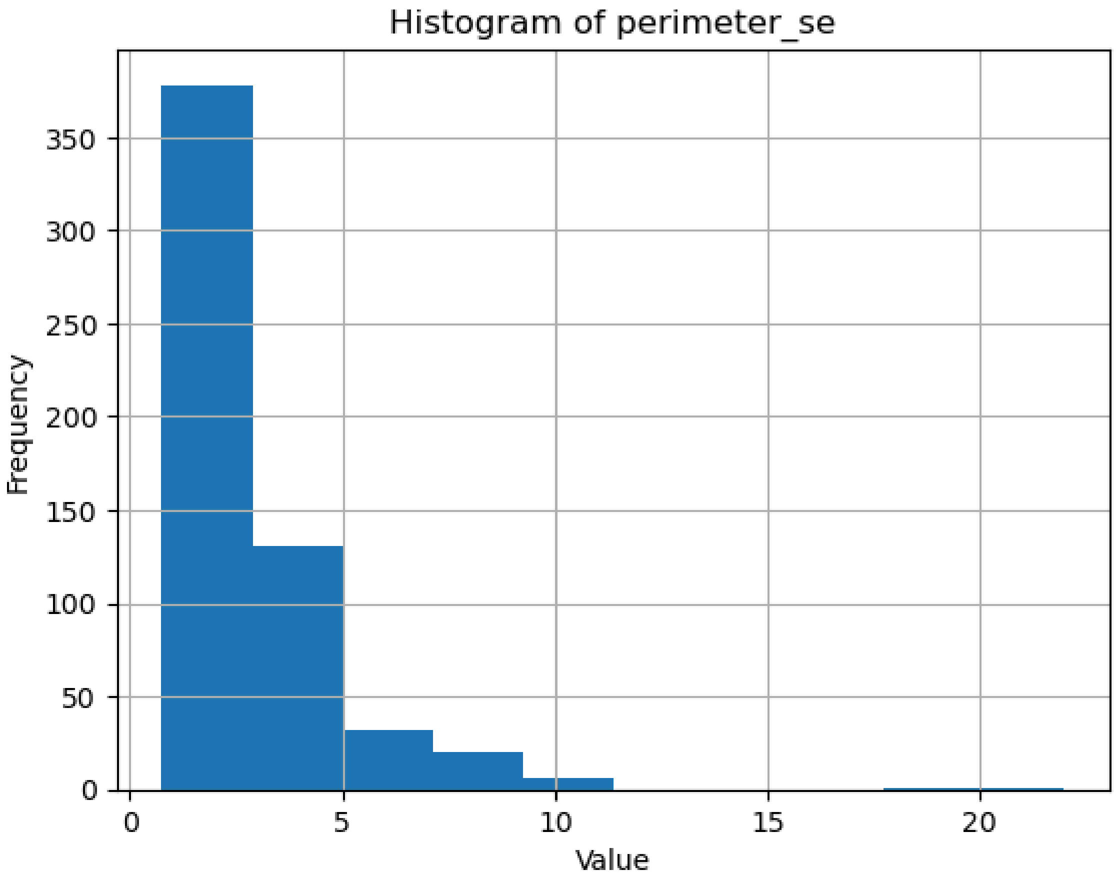

Histogram of concave points_mean
Frequency
175
150
125
100
75
50
25
0
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
Value

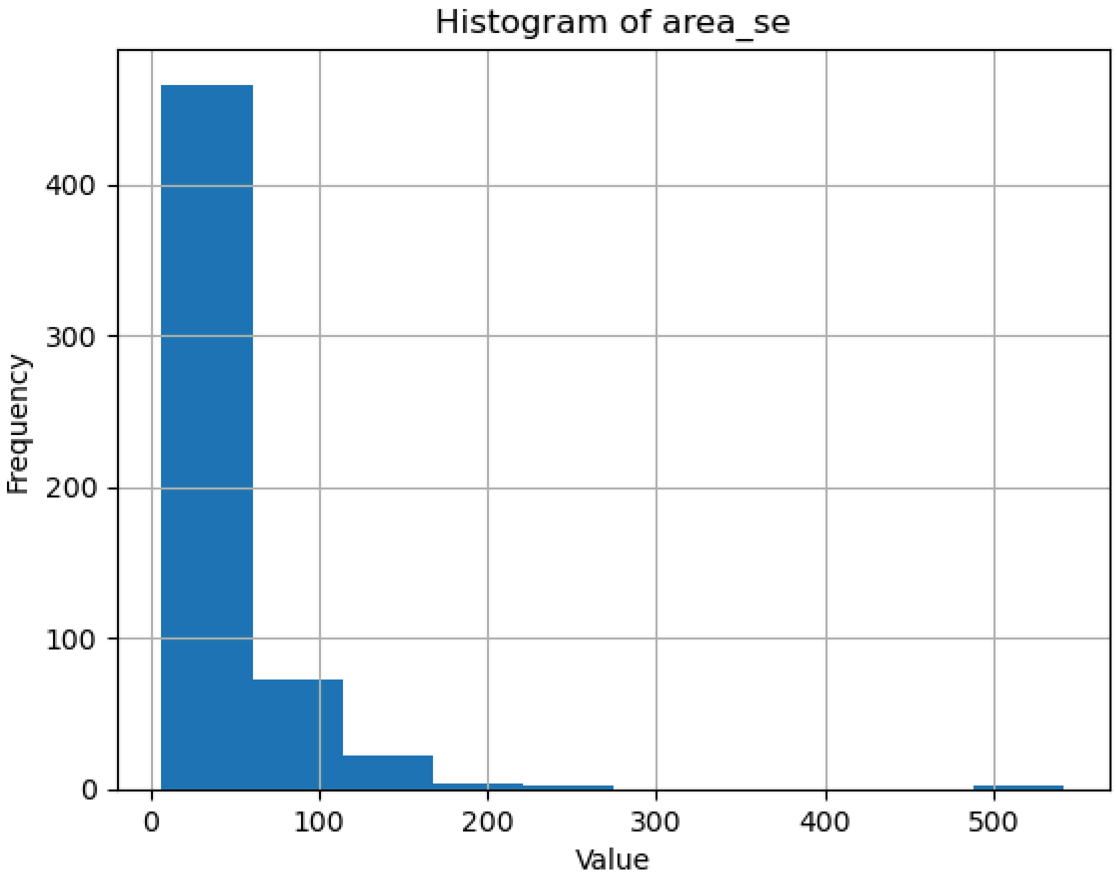

Histogram of symmetry_mean
Frequency
160
140
120
100
80
60
40
20
0
0.100
0.125
0.150
0.175
0.200
0.225
0.250
0.275
0.300
Value

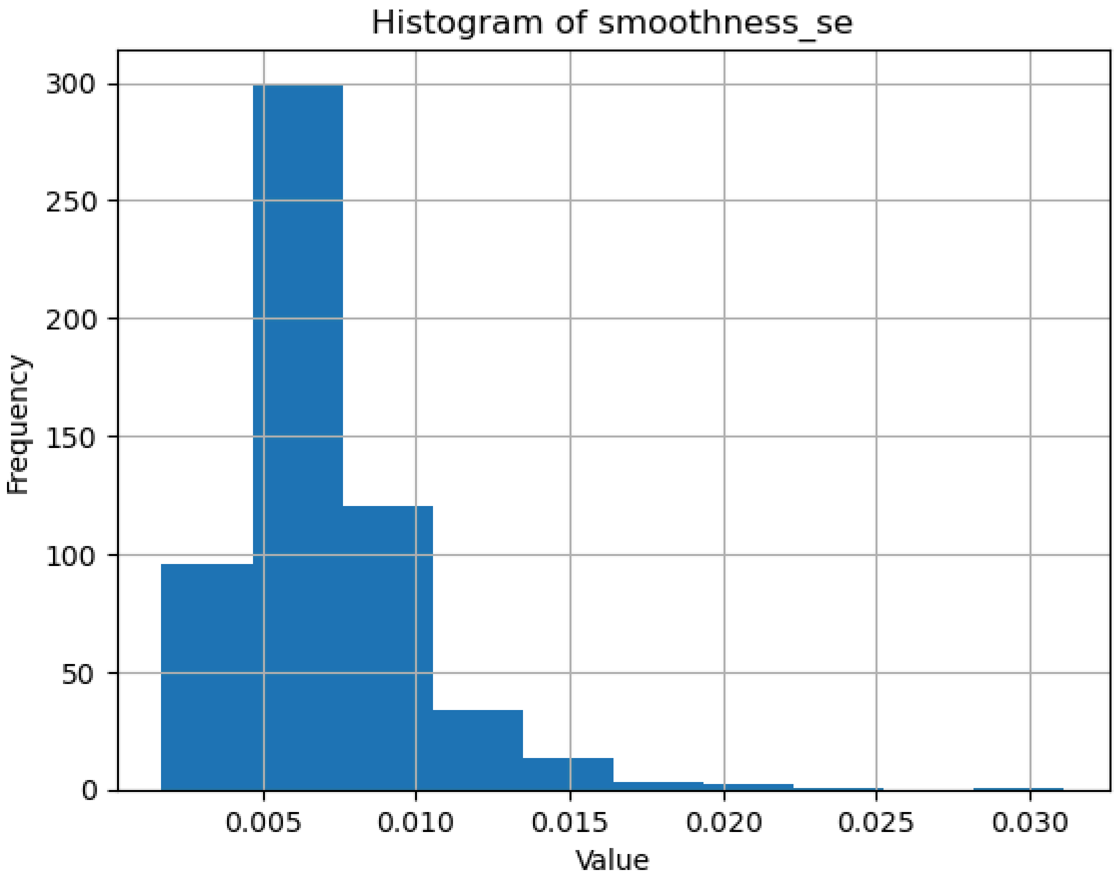

Histogram of fractal_dimension_mean
Frequency
175
150
125
100
75
50
25
0
0.05
0.06
0.07
0.08
0.09
Value


Histogram of radius_se
Frequency
350
300
250
200
150
100
50
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Value

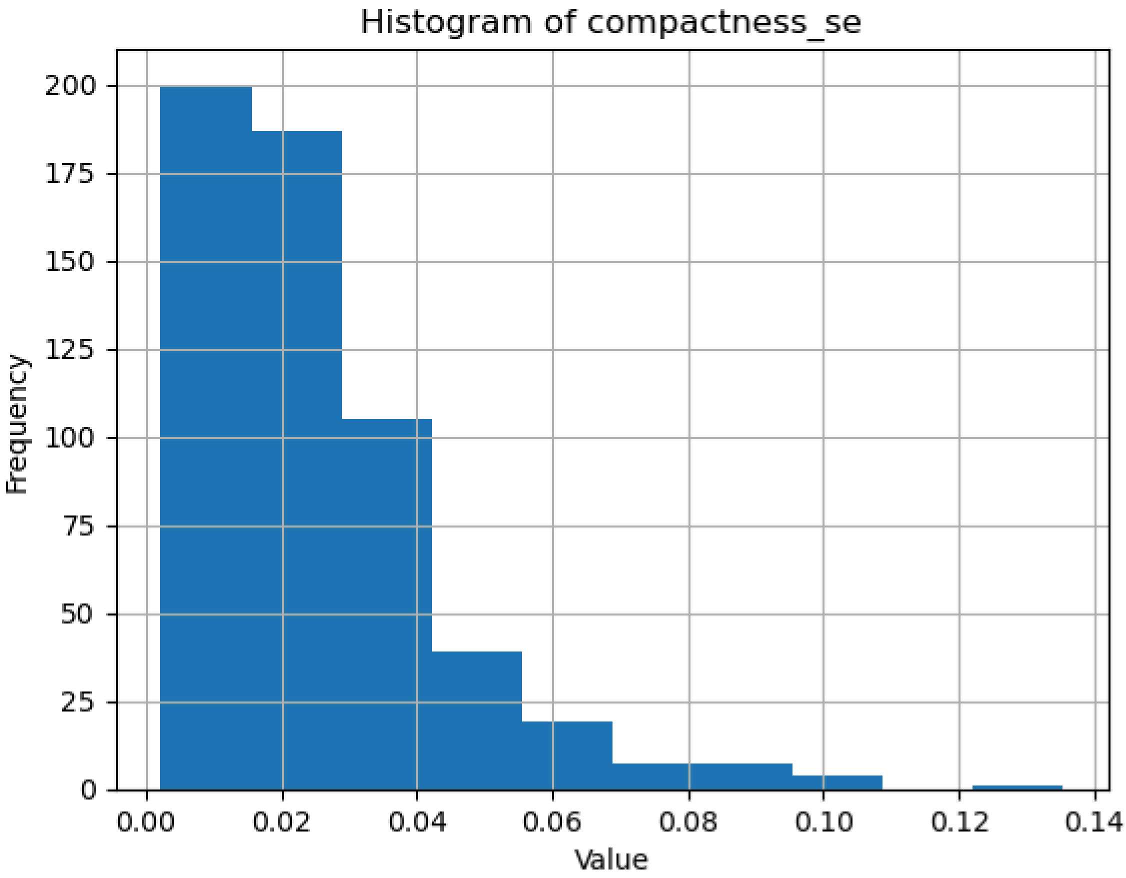

Histogram of texture_se

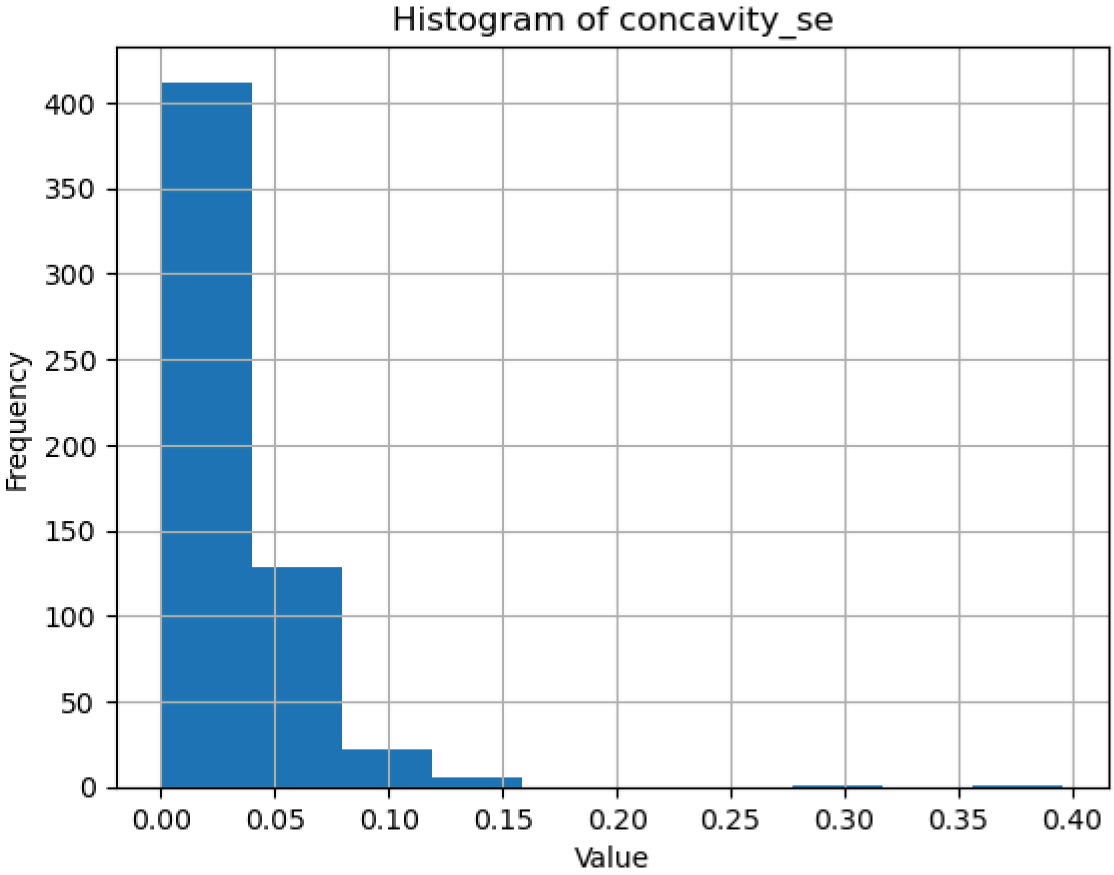

Histogram of perimeter_se

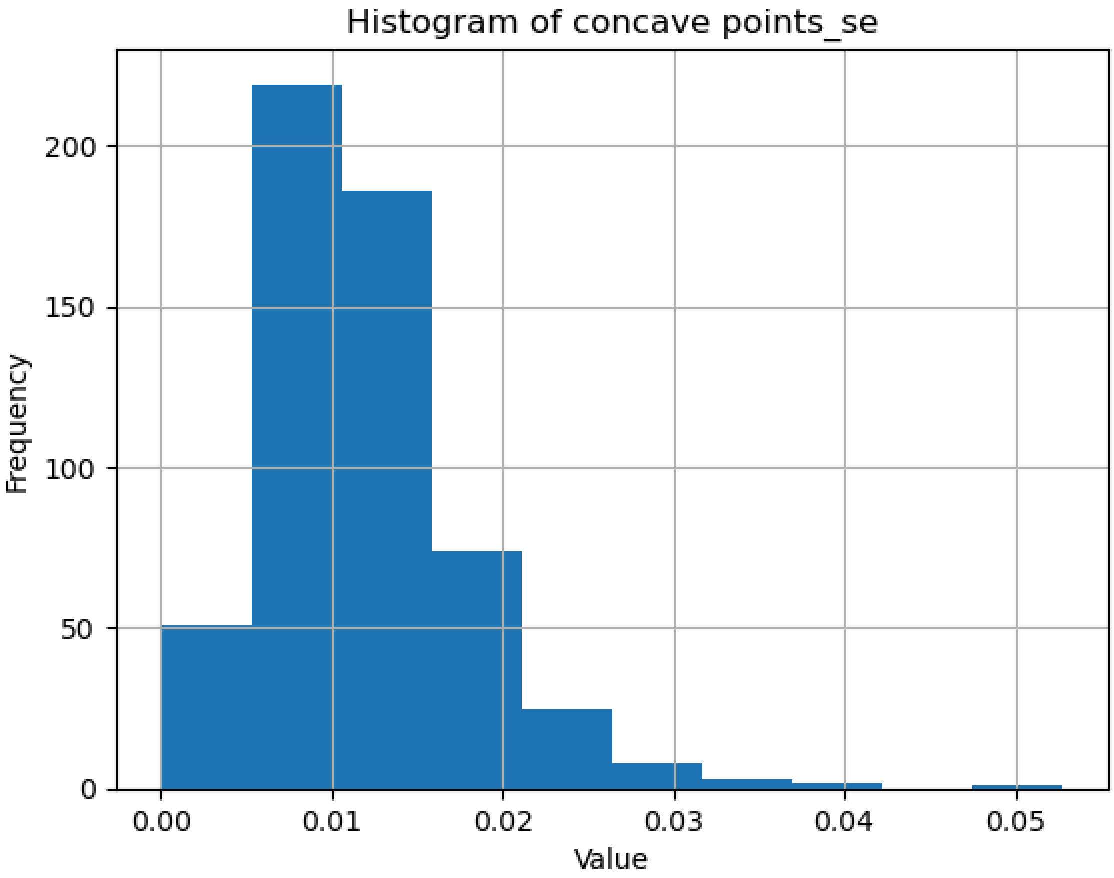


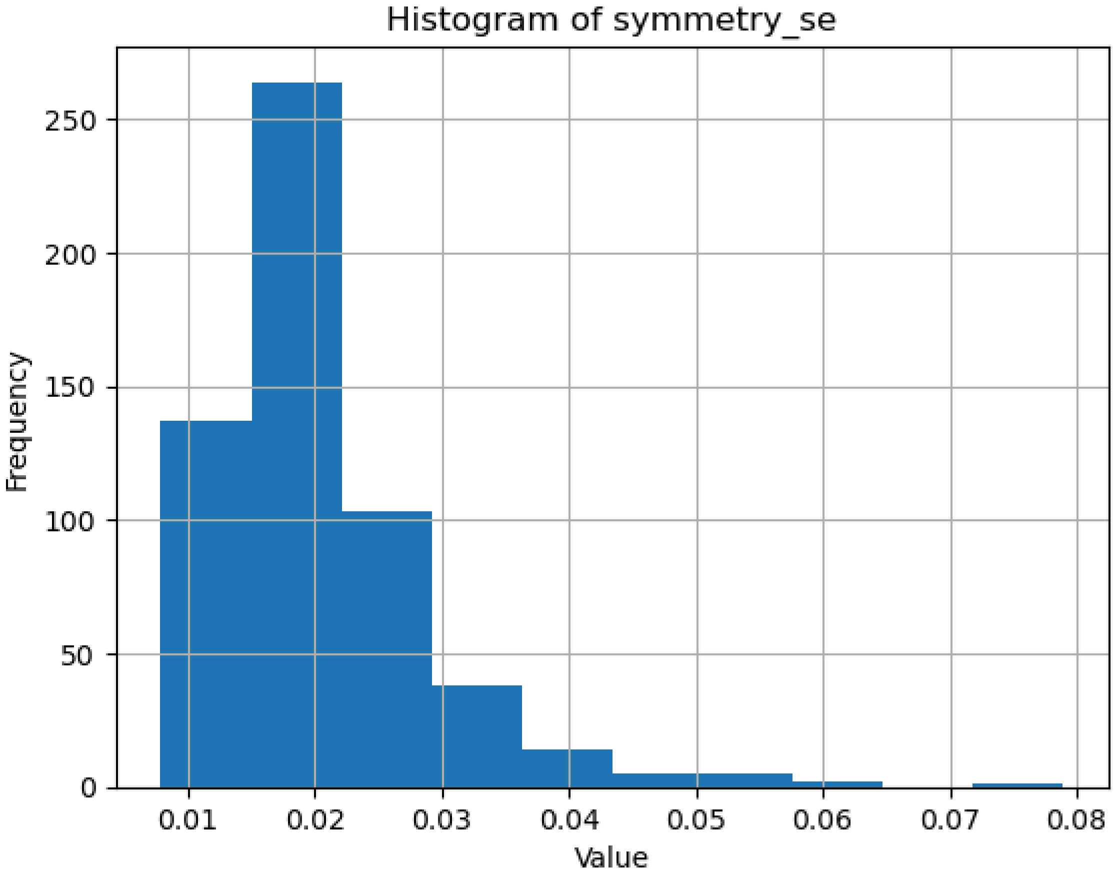


Loading [MathJax]/extensions/Safe.js

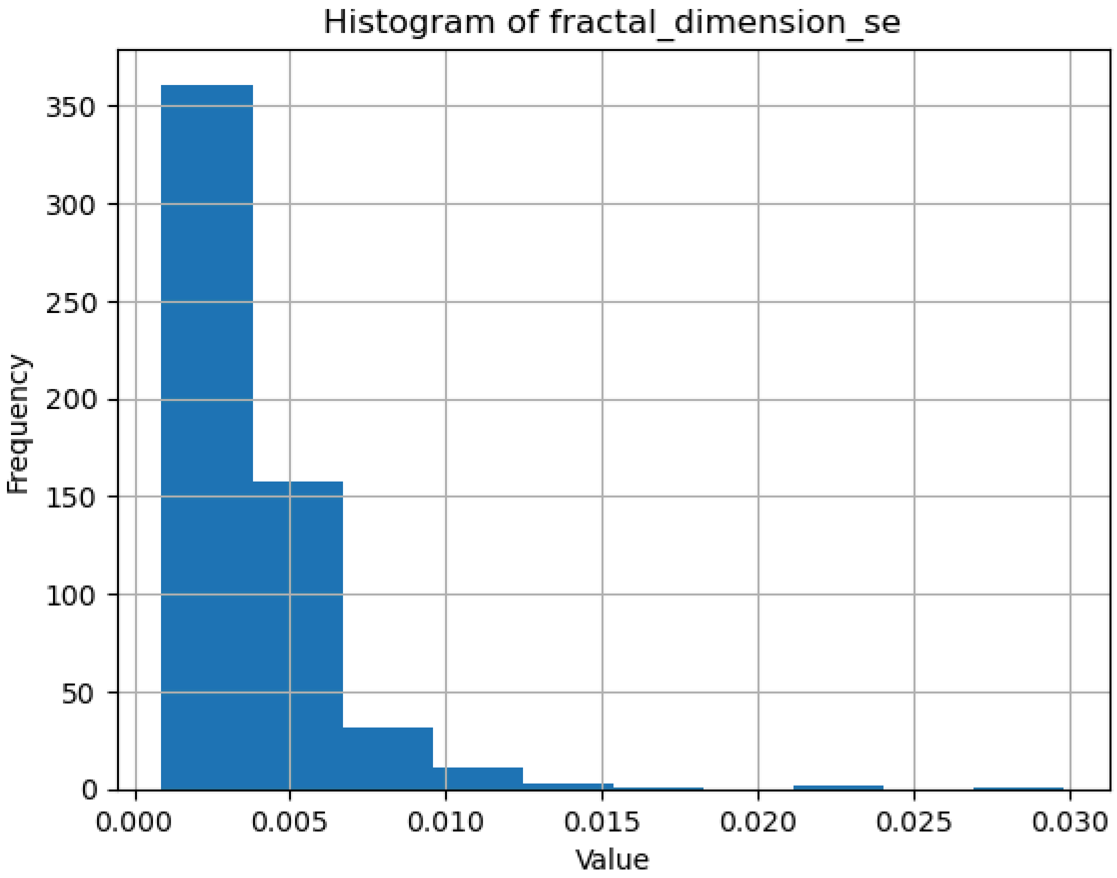

Histogram of compactness_se
Frequency
Value
0
25
50
75
100
125
150
175
200
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

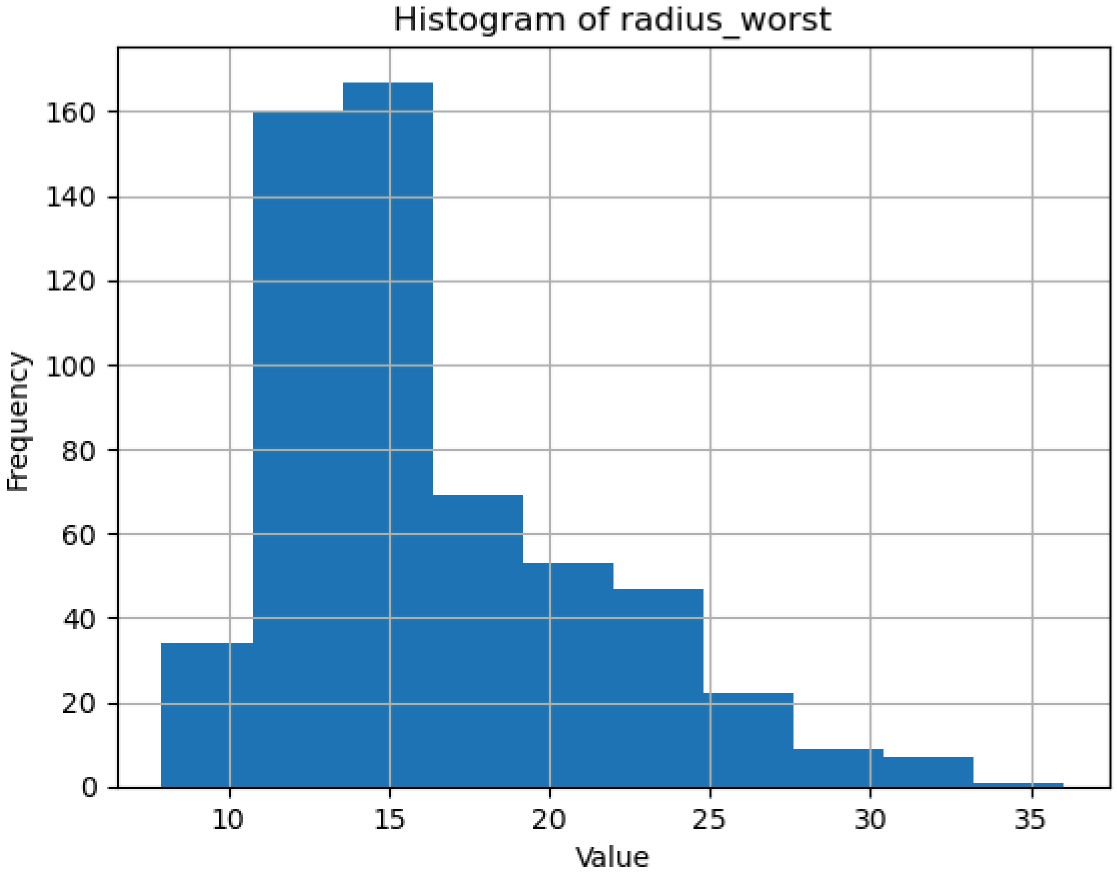

Histogram of concavity_se
Frequency
Value
0
50
100
150
200
250
300
350
400
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40



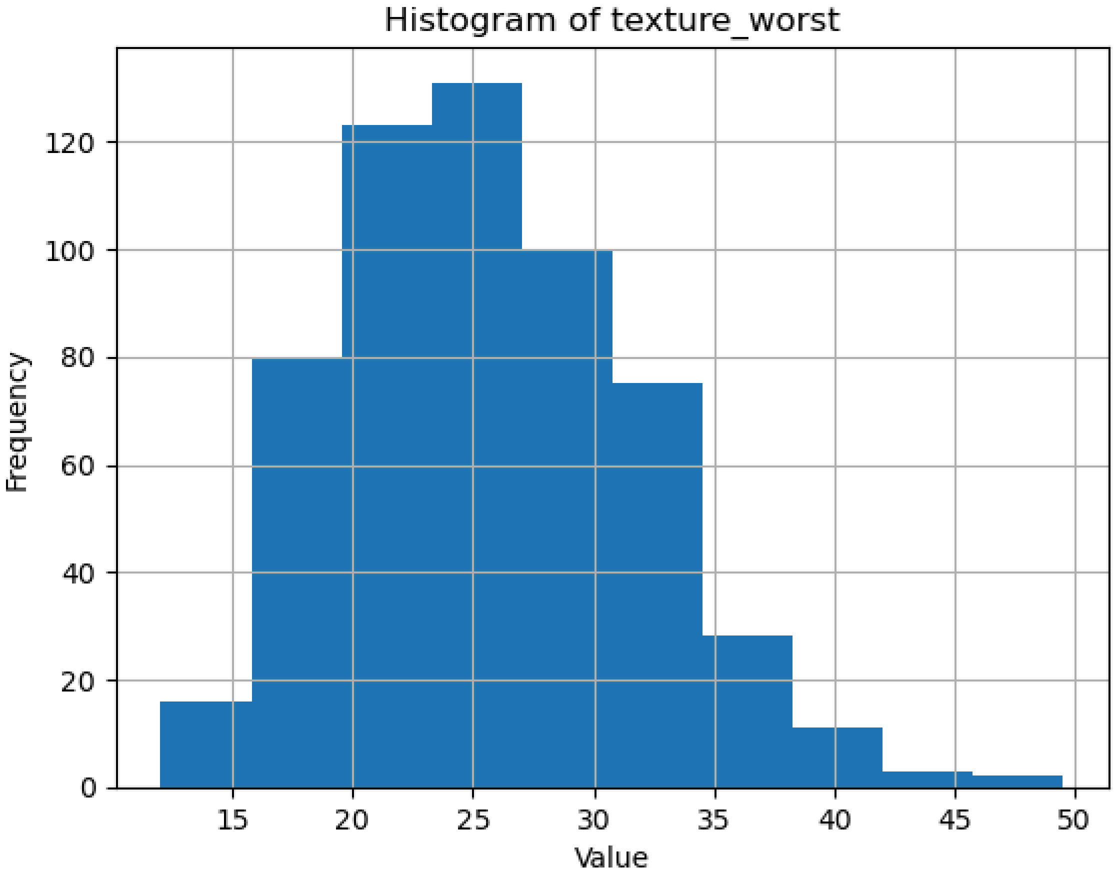


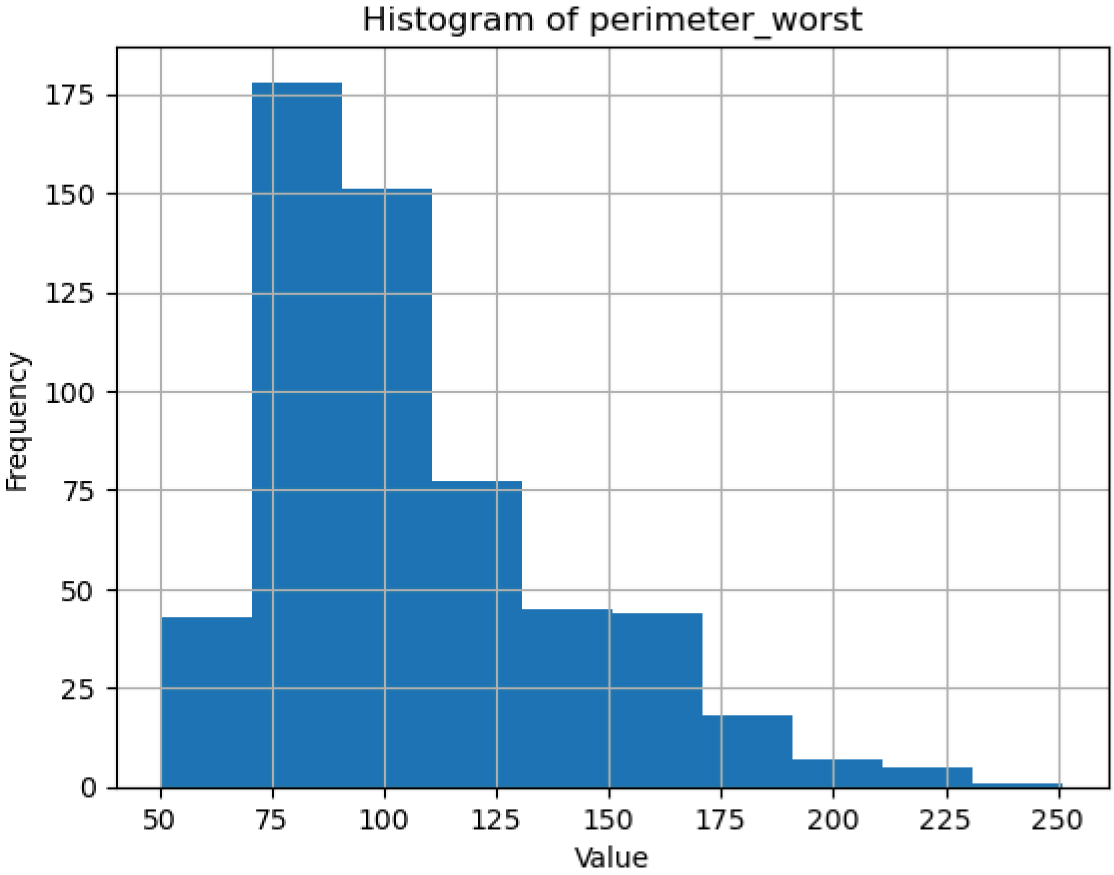
Loading [MathJax]/extensions/Safe.js

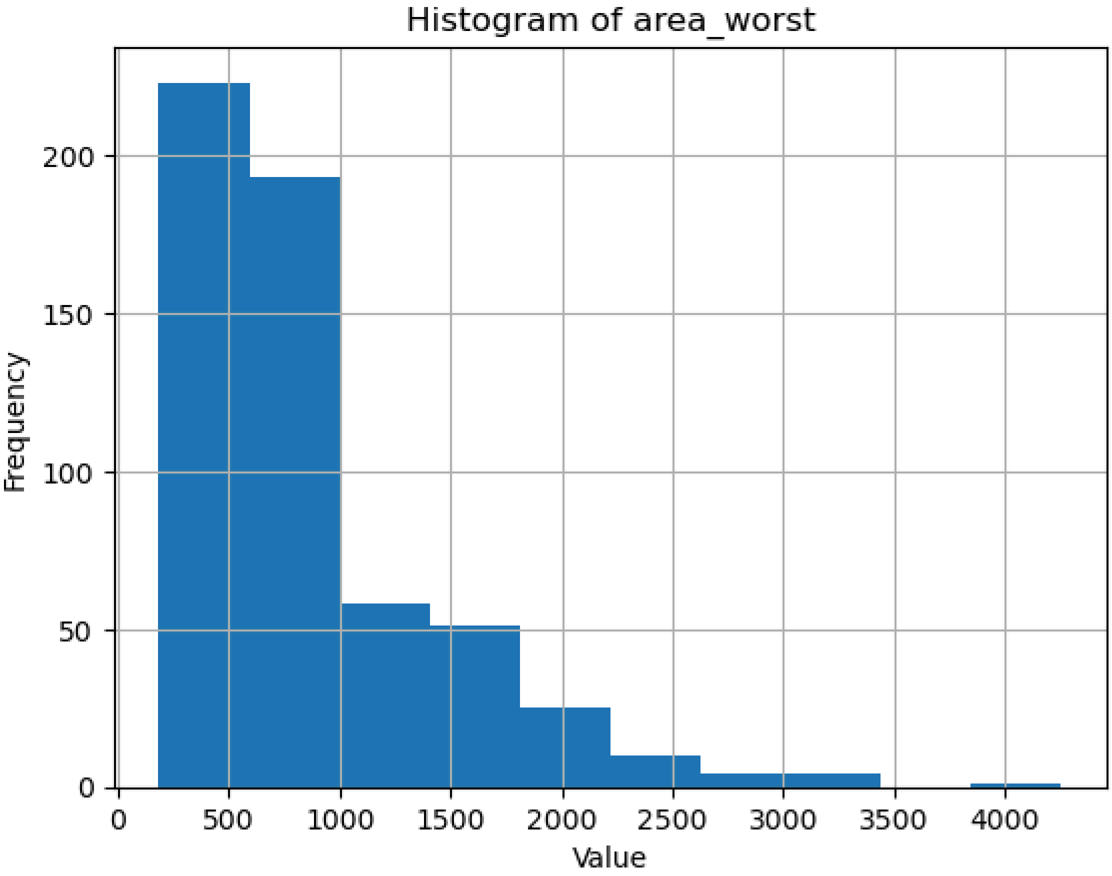


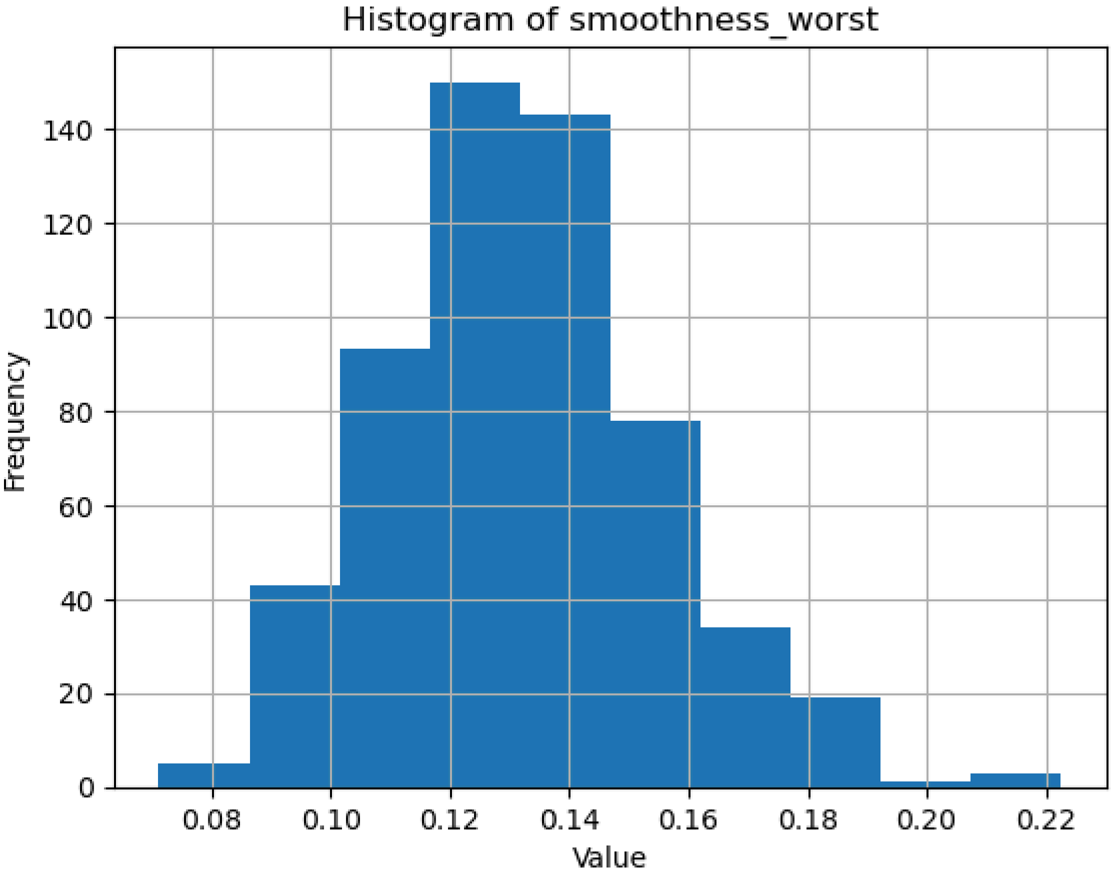


Loading [MathJax]/extensions/Safe.js

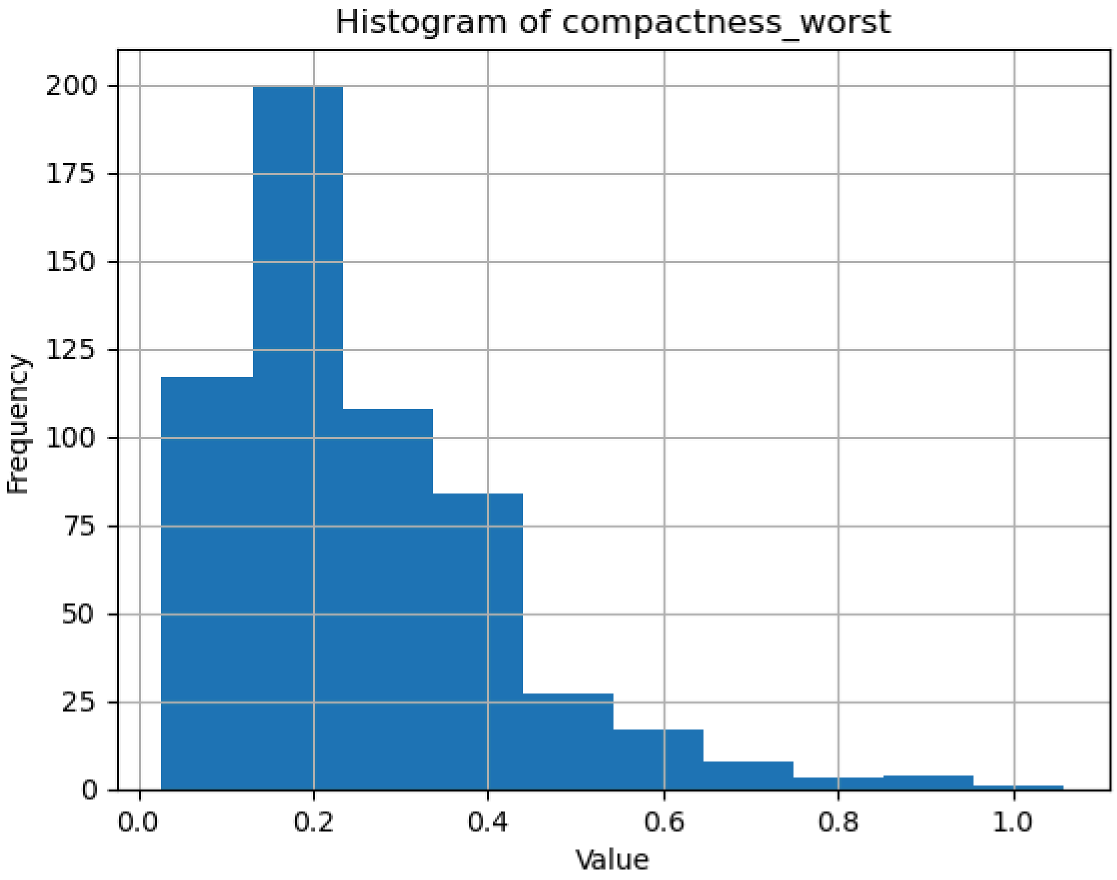

Histogram of texture_worst
Frequency
Value
0
20
40
60
80
100
120
15
20
25
30
35
40
45
50

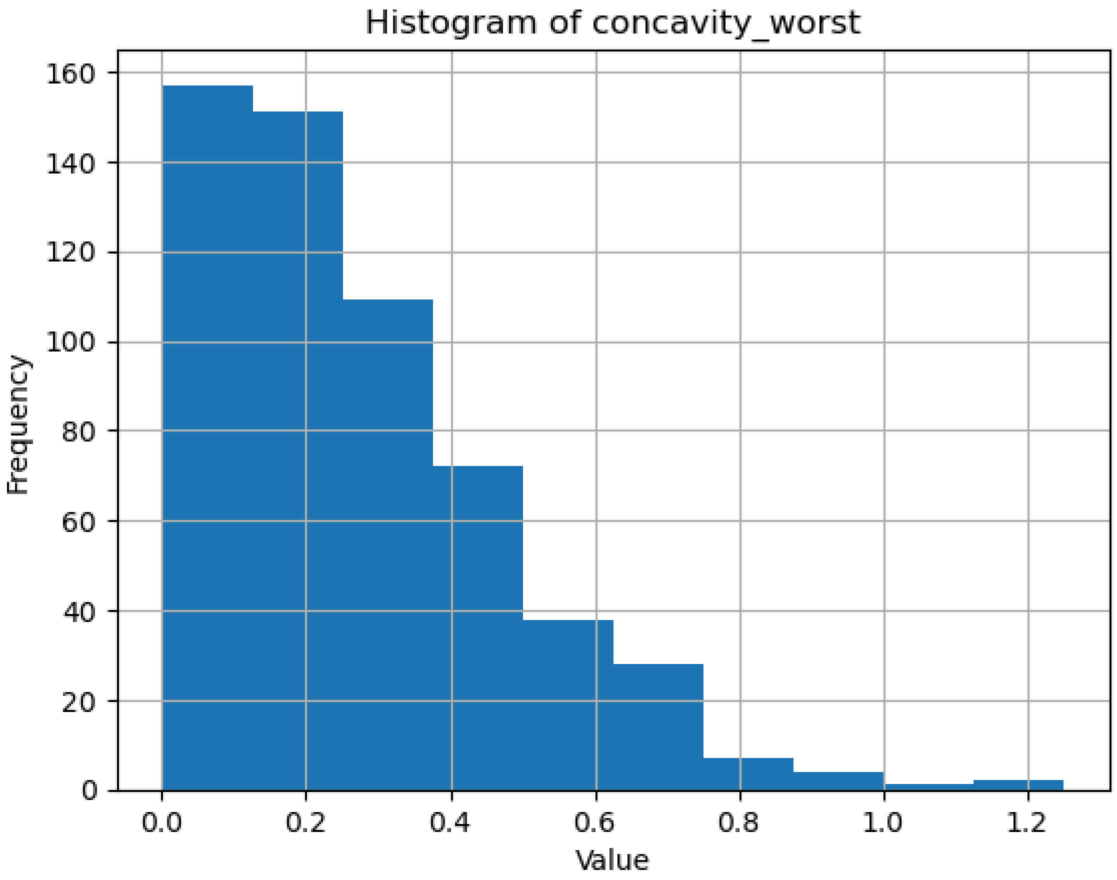

Histogram of perimeter_worst
Frequency
Value
0
25
50
75
100
125
150
175
50
75
100
125
150
175
200
225
250

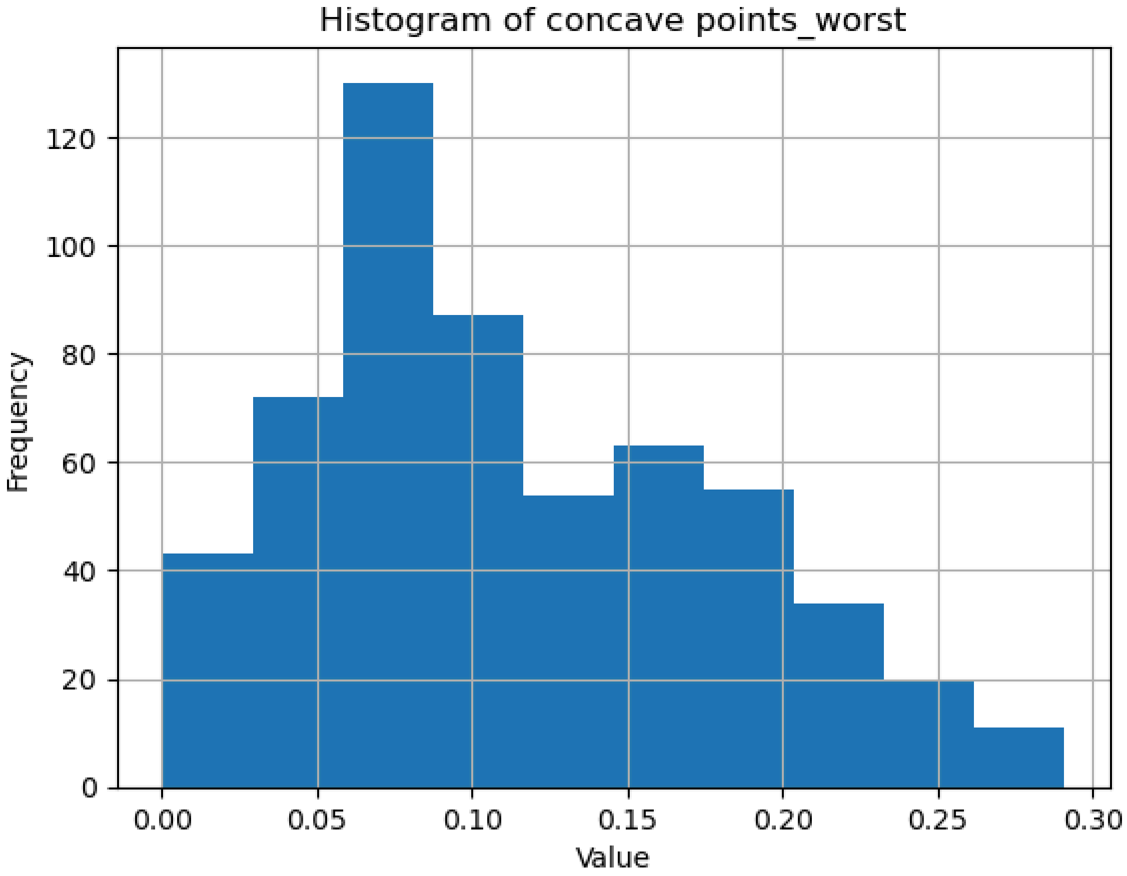


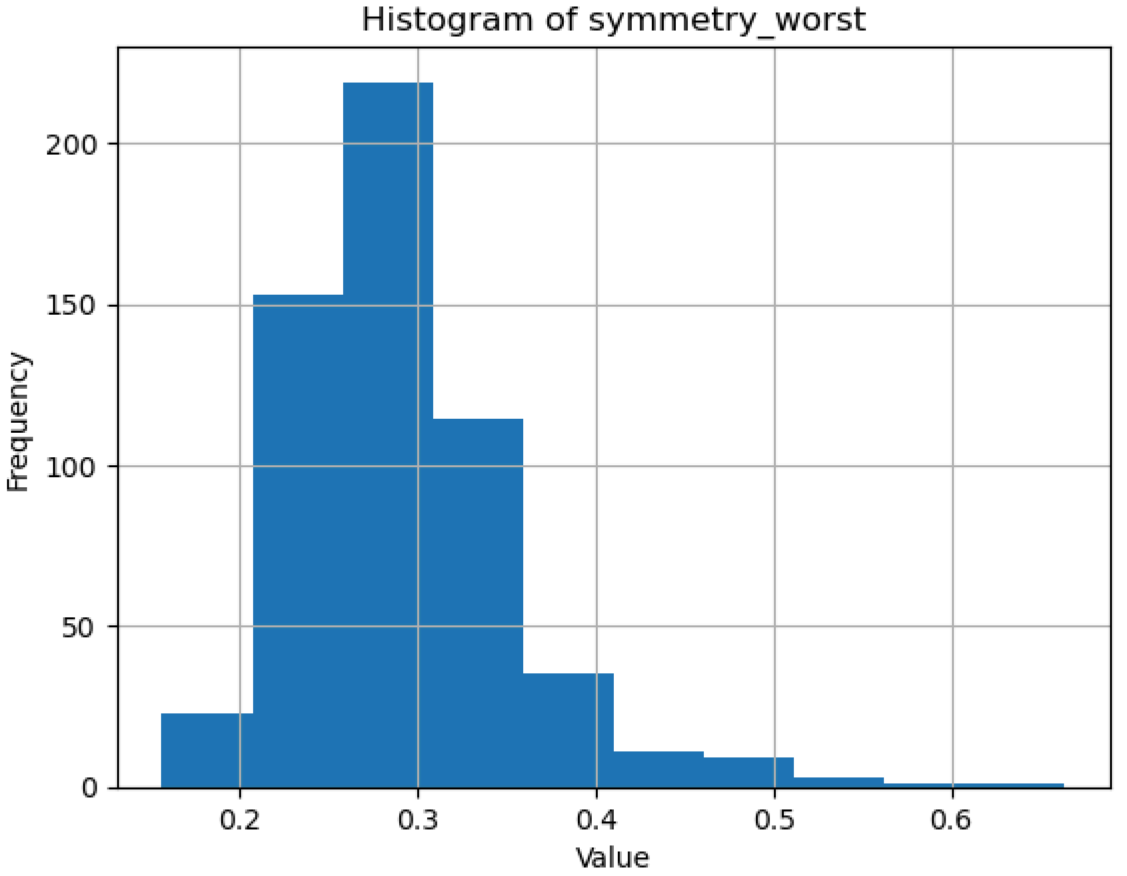


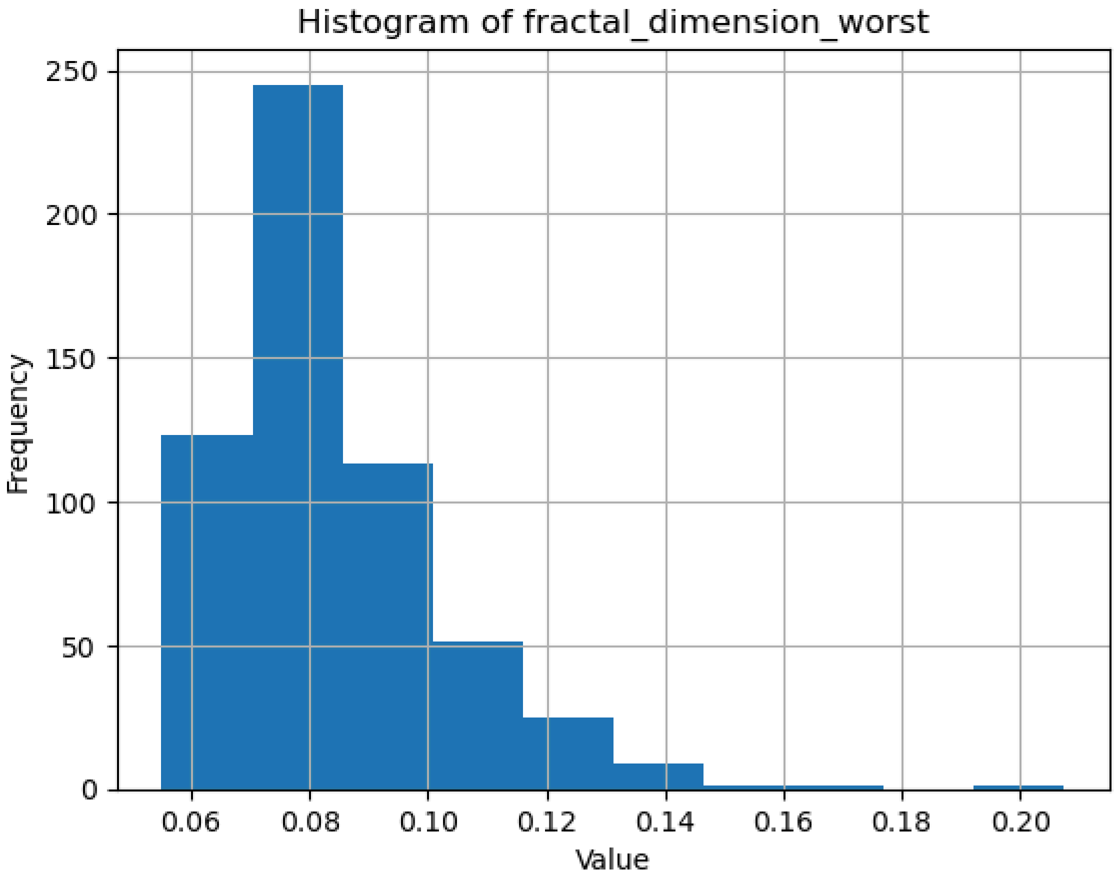
Loading [MathJax]/extensions/Safe.js

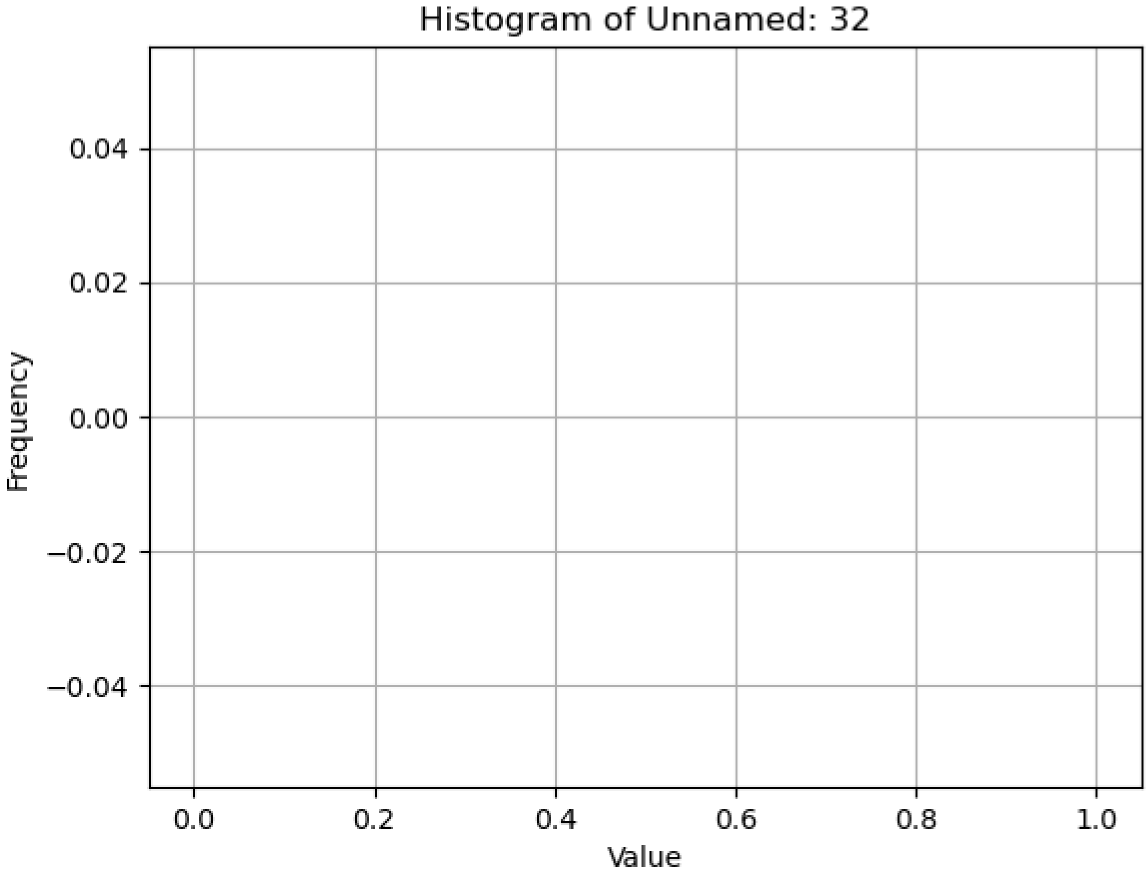

Histogram of compactness_worst
Frequency
Value
0
25
50
75
100
125
150
175
200
0.0
0.2
0.4
0.6
0.8
1.0


Histogram of concavity_worst
Frequency
Value
0
20
40
60
80
100
120
140
160
0.0
0.2
0.4
0.6
0.8
1.0
1.2


Histogram of concave points_worst
Frequency
0
20
40
60
80
100
120
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Value


Histogram of symmetry_worst
Frequency
0
50
100
150
200
0.2
0.3
0.4
0.5
0.6
Value





Loading [MathJax]/extensions/Safe.js

In [11]:

```
summary_stats = df.describe()
print(summary_stats)
```

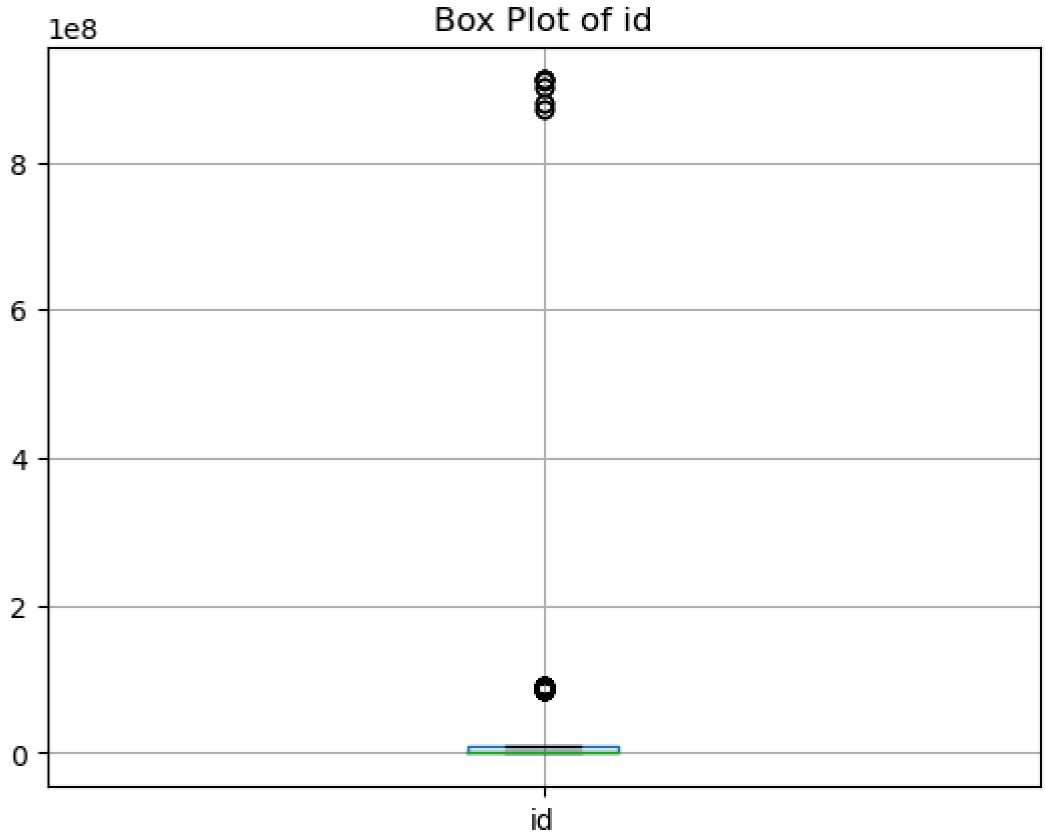
Loading [MathJax]/extensions/Safe.js

```
                 id  radius_mean  texture_mean  perimeter_mean    area_mean  
\
count  5.690000e+02   569.000000    569.000000      569.000000   569.000000   
mean   3.037183e+07    14.127292     19.289649       91.969033   654.889104   
std    1.250206e+08     3.524049      4.301036       24.298981   351.914129   
min    8.670000e+03     6.981000      9.710000       43.790000   143.500000   
25%    8.692180e+05    11.700000     16.170000       75.170000   420.300000   
50%    9.060240e+05    13.370000     18.840000       86.240000   551.100000   
75%    8.813129e+06    15.780000     21.800000      104.100000   782.700000   
max    9.113205e+08    28.110000     39.280000      188.500000  2501.000000   

       smoothness_mean  compactness_mean  concavity_mean  concave points_mean  
\
count       569.000000        569.000000      569.000000           569.000000   
mean          0.096360          0.104341        0.088799             0.048919   
std           0.014064          0.052813        0.079720             0.038803   
min           0.052630          0.019380        0.000000             0.000000   
25%           0.086370          0.064920        0.029560             0.020310   
50%           0.095870          0.092630        0.061540             0.033500   
75%           0.105300          0.130400        0.130700             0.074000   
max           0.163400          0.345400        0.426800             0.201200   

       symmetry_mean  ...  texture_worst  perimeter_worst   area_worst  \
count     569.000000  ...     569.000000       569.000000   569.000000   
mean        0.181162  ...      25.677223       107.261213   880.583128   
std         0.027414  ...       6.146258        33.602542   569.356993   
min         0.106000  ...      12.020000        50.410000   185.200000   
25%         0.161900  ...      21.080000        84.110000   515.300000   
50%         0.179200  ...      25.410000        97.660000   686.500000   
75%         0.195700  ...      29.720000       125.400000  1084.000000   
max         0.304000  ...      49.540000       251.200000  4254.000000   

       smoothness_worst  compactness_worst  concavity_worst  \
count        569.000000         569.000000       569.000000   
mean           0.132369           0.254265         0.272188   
std            0.022832           0.157336         0.208624   
min            0.071170           0.027290         0.000000   
25%            0.116600           0.147200         0.114500   
50%            0.131300           0.211900         0.226700   
75%            0.146000           0.339100         0.382900   
max            0.222600           1.058000         1.252000   

       concave points_worst  symmetry_worst  fractal_dimension_worst  \
count            569.000000      569.000000               569.000000   
mean               0.114606        0.290076                 0.083946   
std                0.065732        0.061867                 0.018061   
min                0.000000        0.156500                 0.055040   
25%                0.064930        0.250400                 0.071460   
50%                0.099930        0.282200                 0.080040   
75%                0.161400        0.317900                 0.092080   
max                0.291000        0.663800                 0.207500   

       Unnamed: 32  
count          0.0  
mean           NaN  
std            NaN  
```

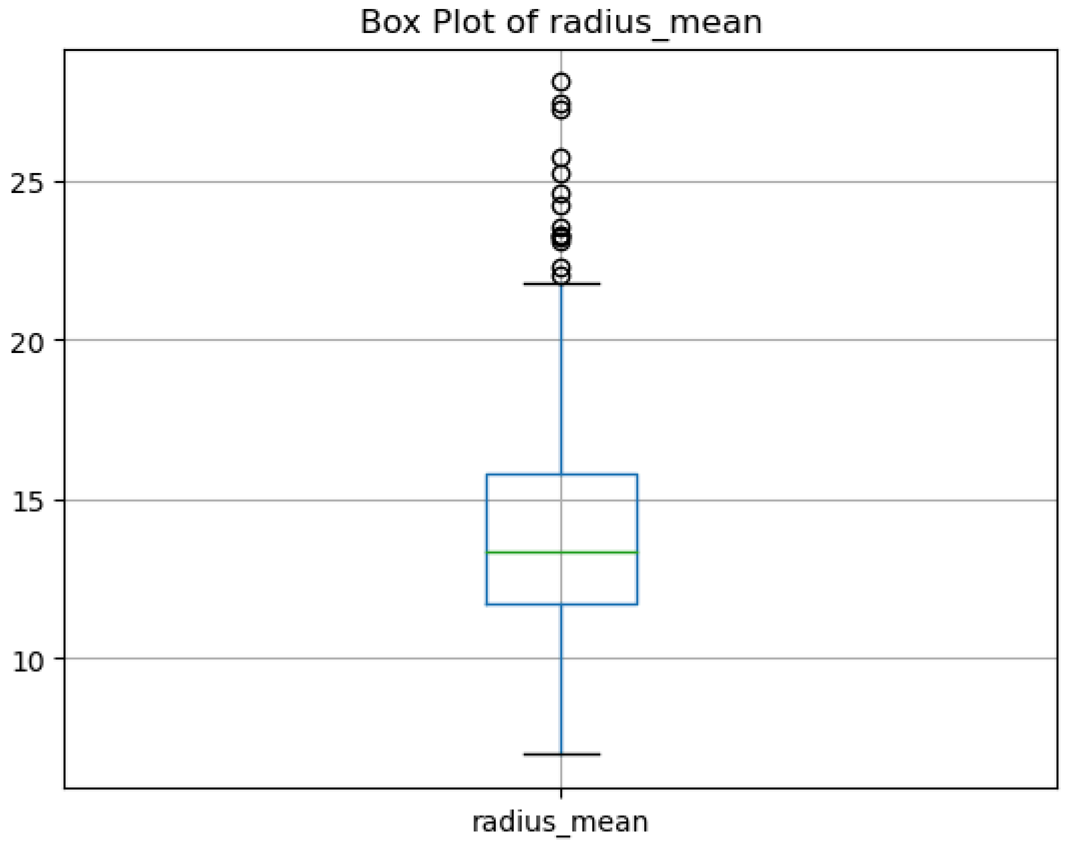
Loading [MathJax]/extensions/Safe.js

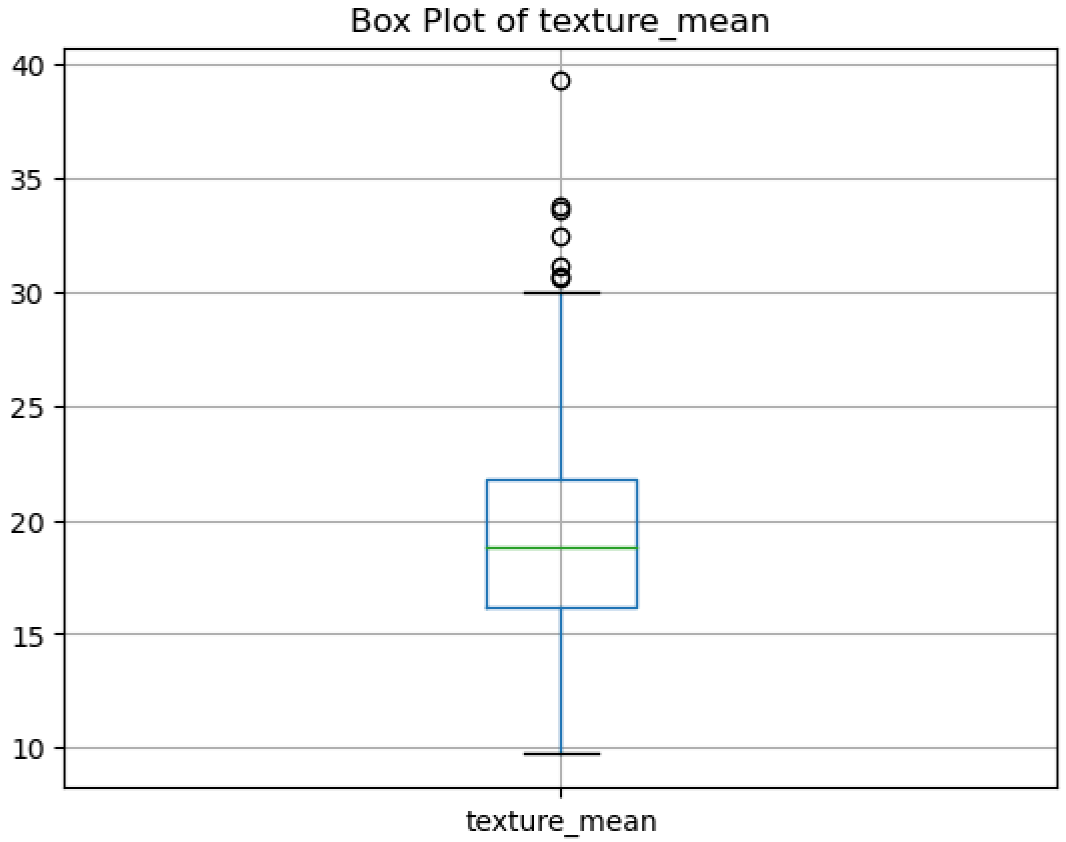
```
min               NaN
25%               NaN
50%               NaN
75%               NaN
max               NaN

[8 rows x 32 columns]
```

In [12]:
```
for col in numerical_columns:
    df.boxplot(column=col)
    plt.title(f'Box Plot of {col}')
    plt.show()
```

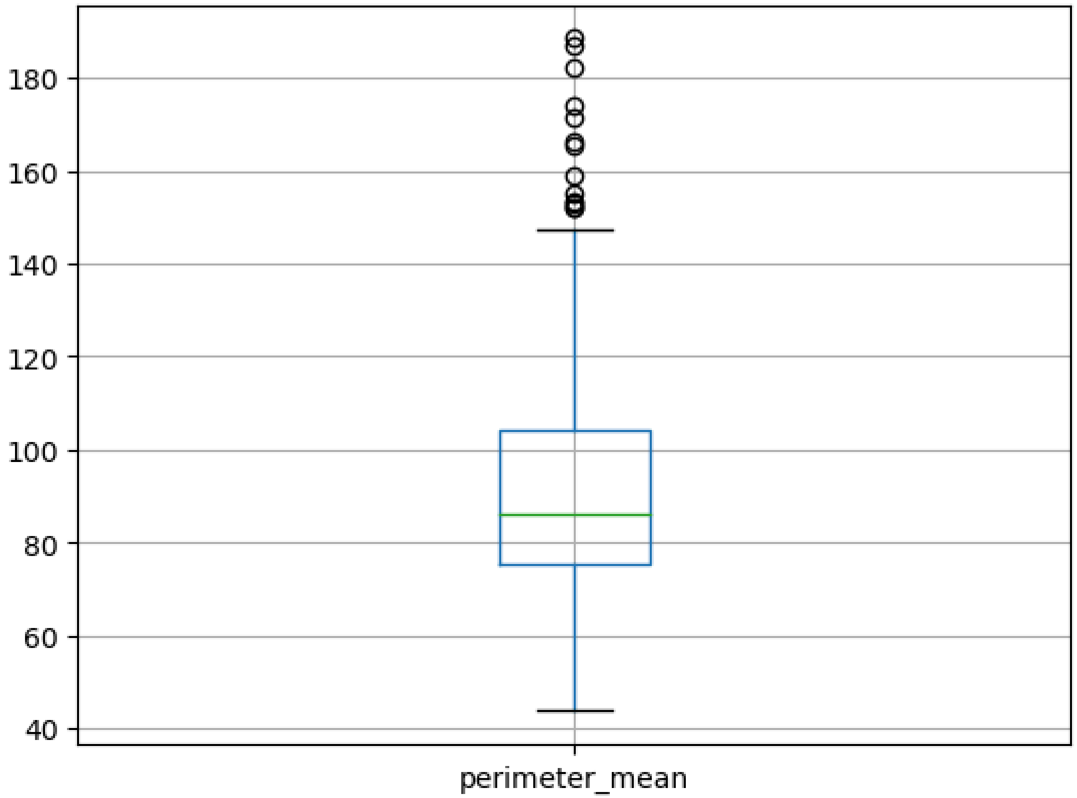


Loading [MathJax]/extensions/Safe.js

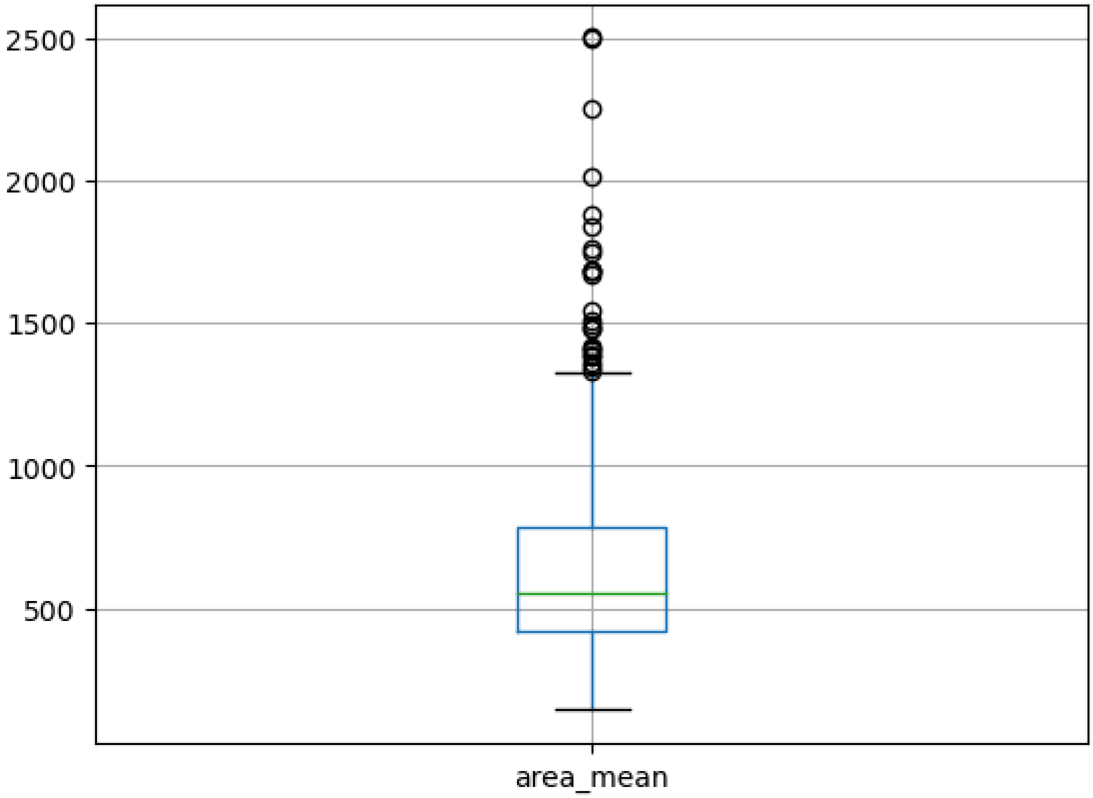

Box Plot of radius_mean


Box Plot of texture_mean

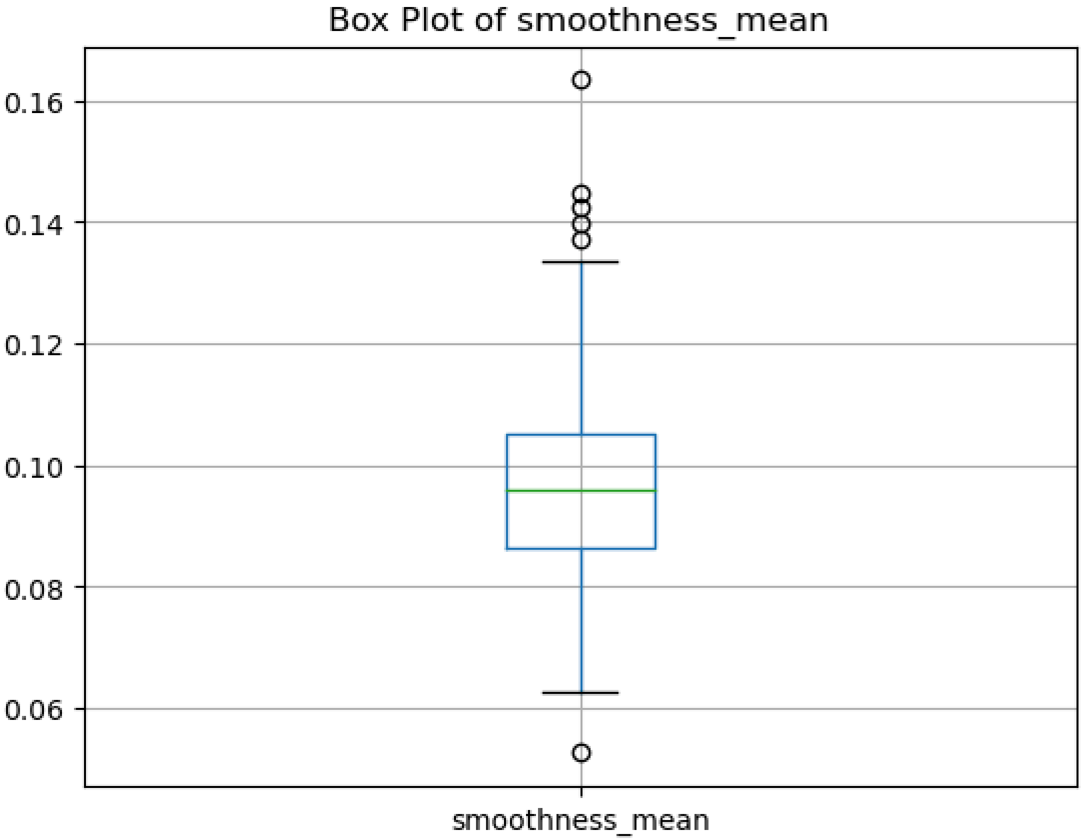
Box Plot of perimeter_mean

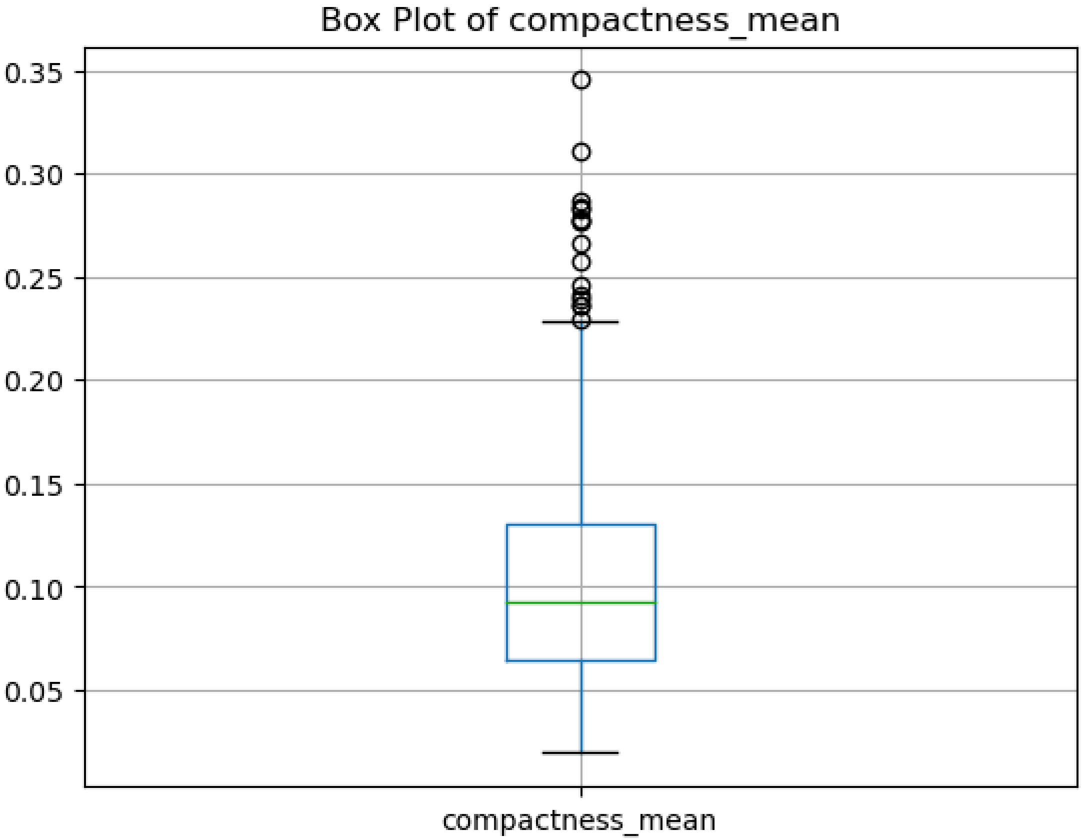


Box Plot of area_mean

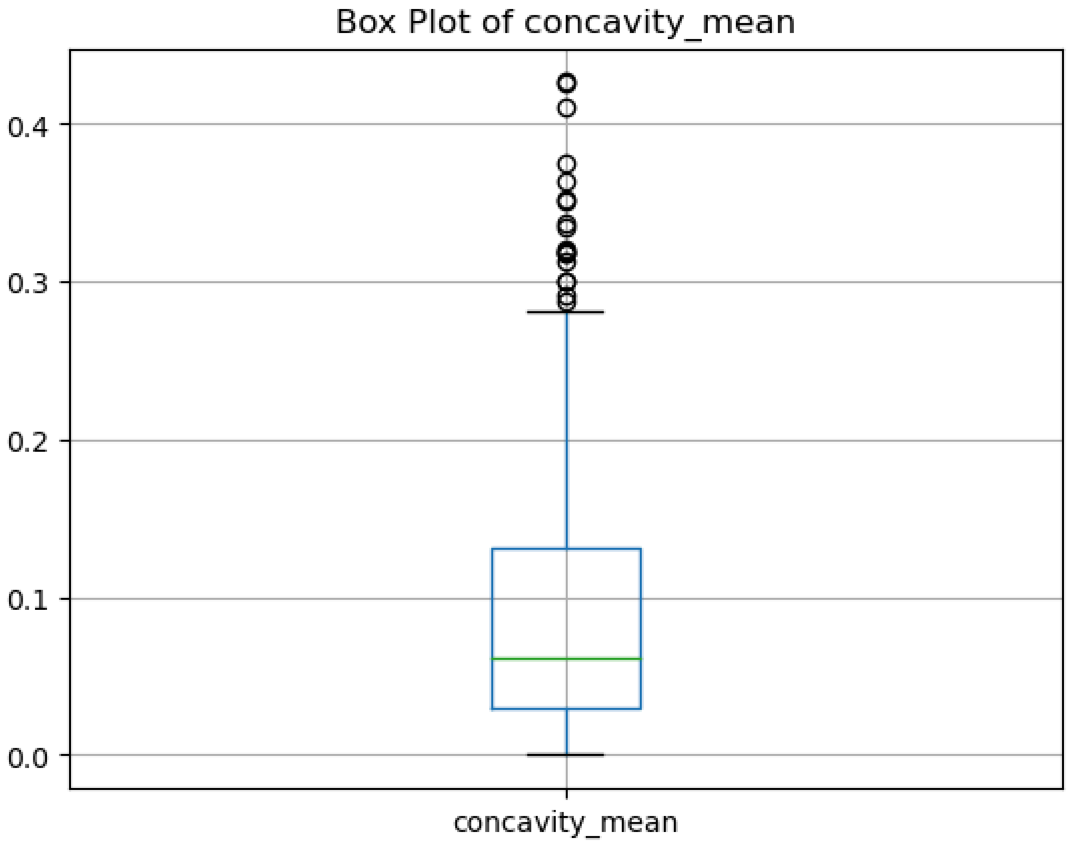


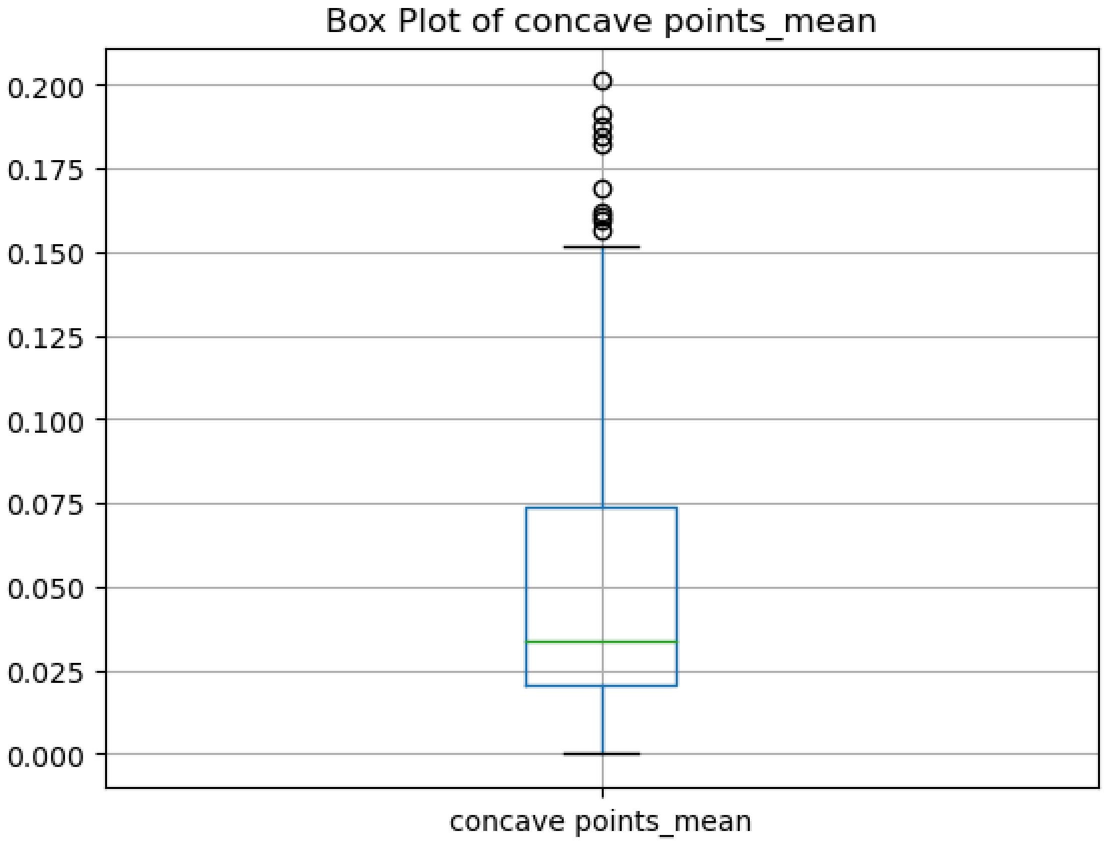
Loading [MathJax]/extensions/Safe.js

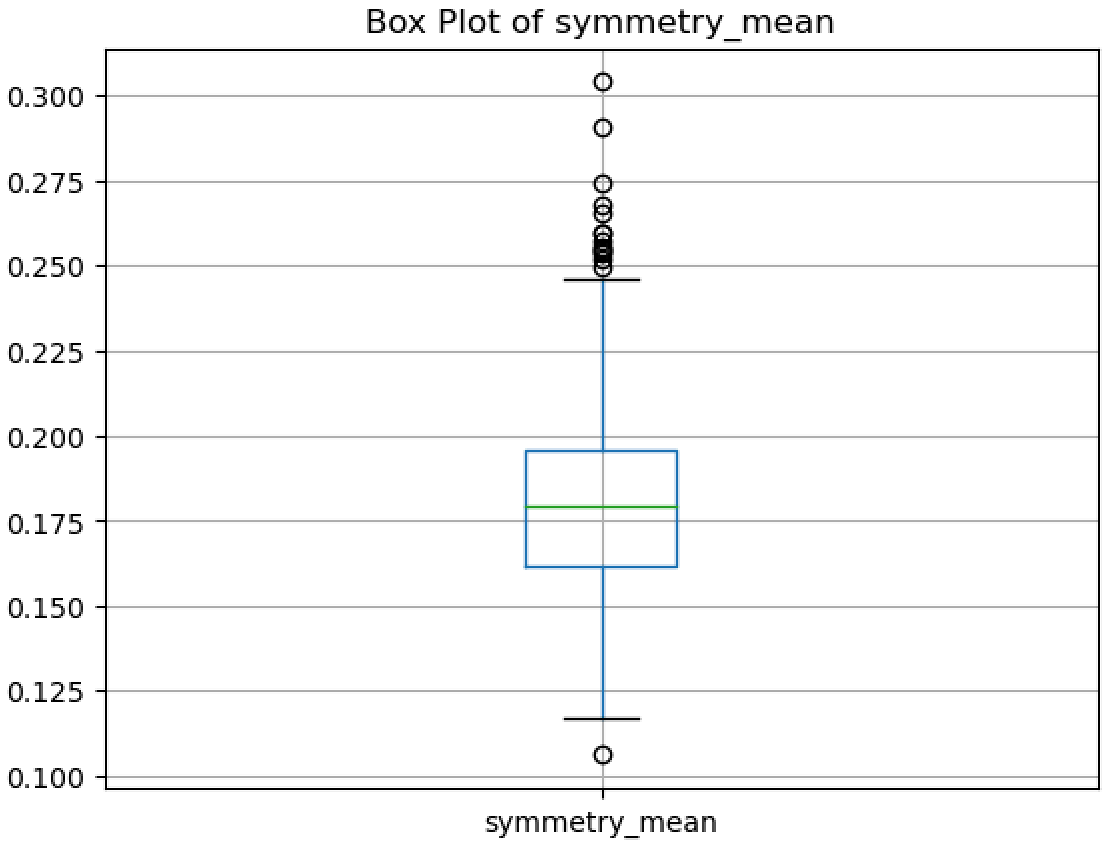


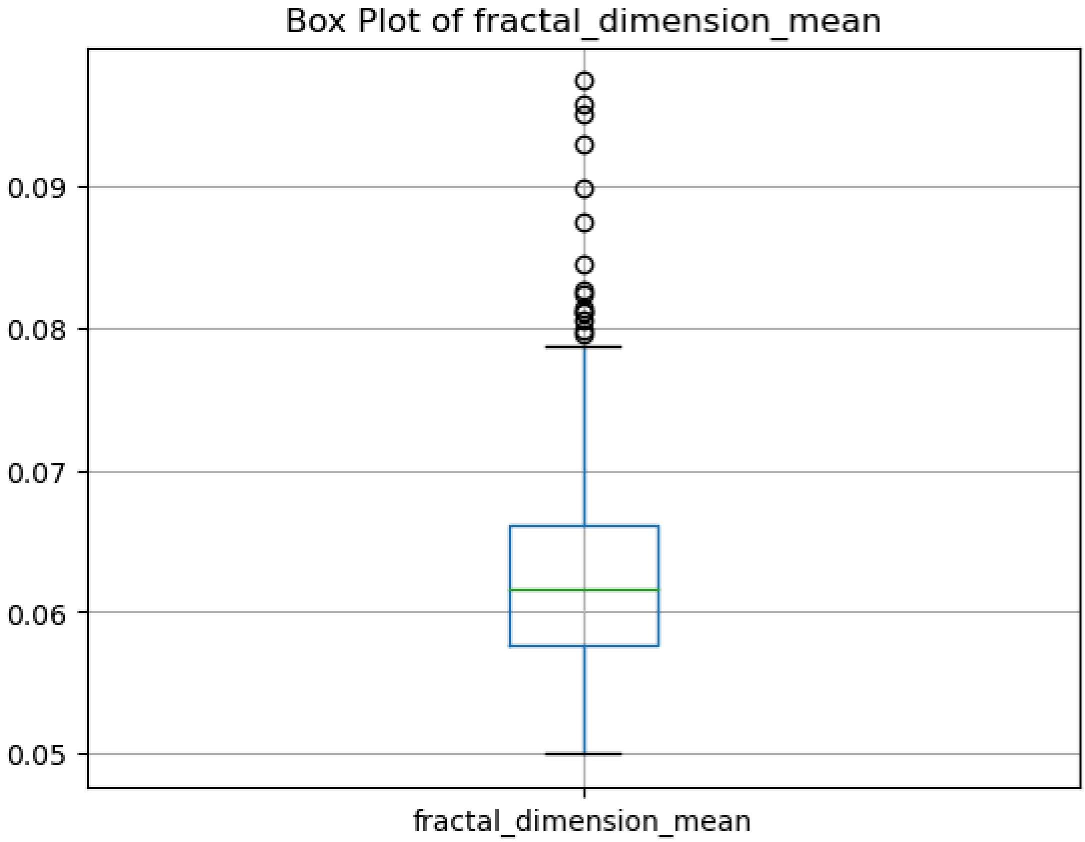


Loading [MathJax]/extensions/Safe.js

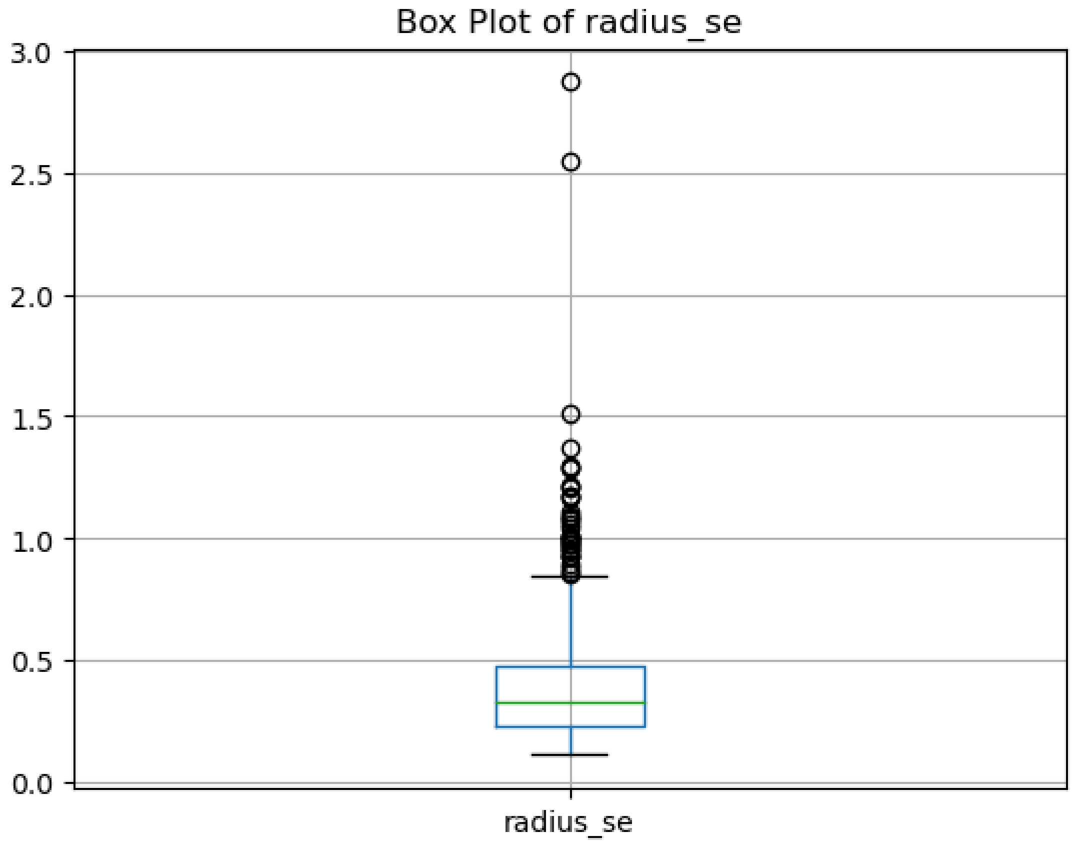

Box Plot of concavity_mean
0.4
0.3
0.2
0.1
0.0
concavity_mean

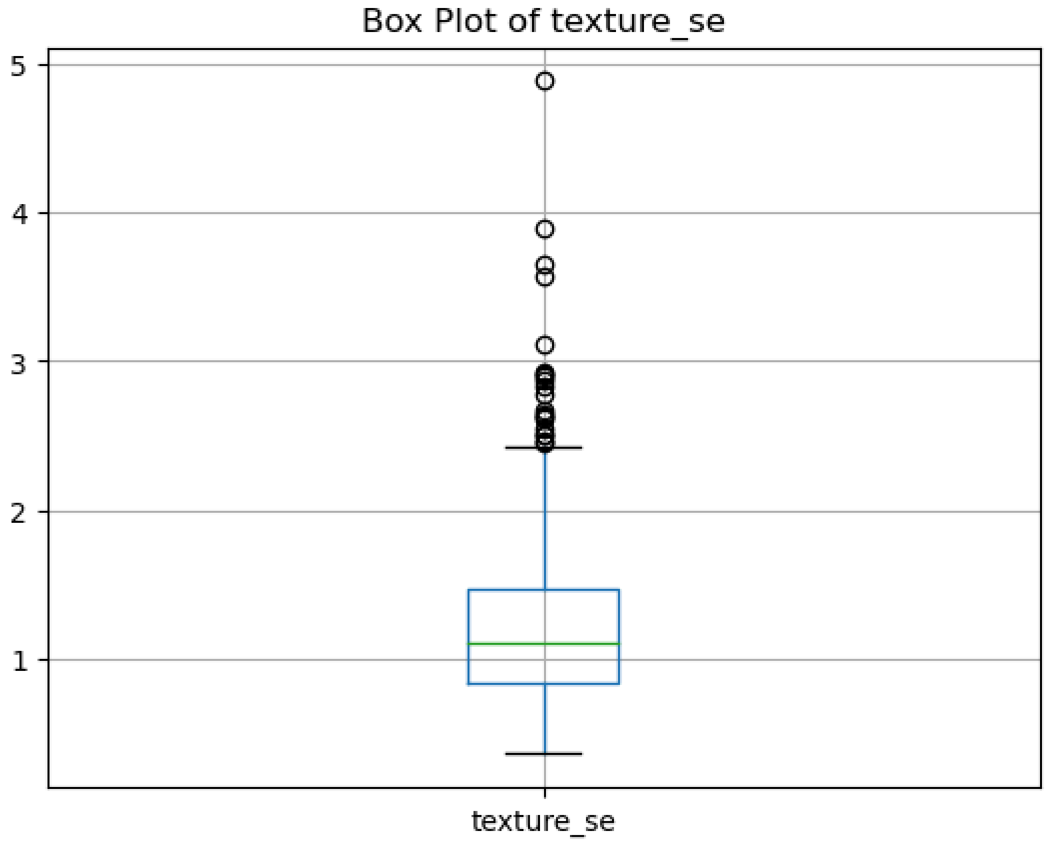

Box Plot of concave points_mean
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
concave points_mean



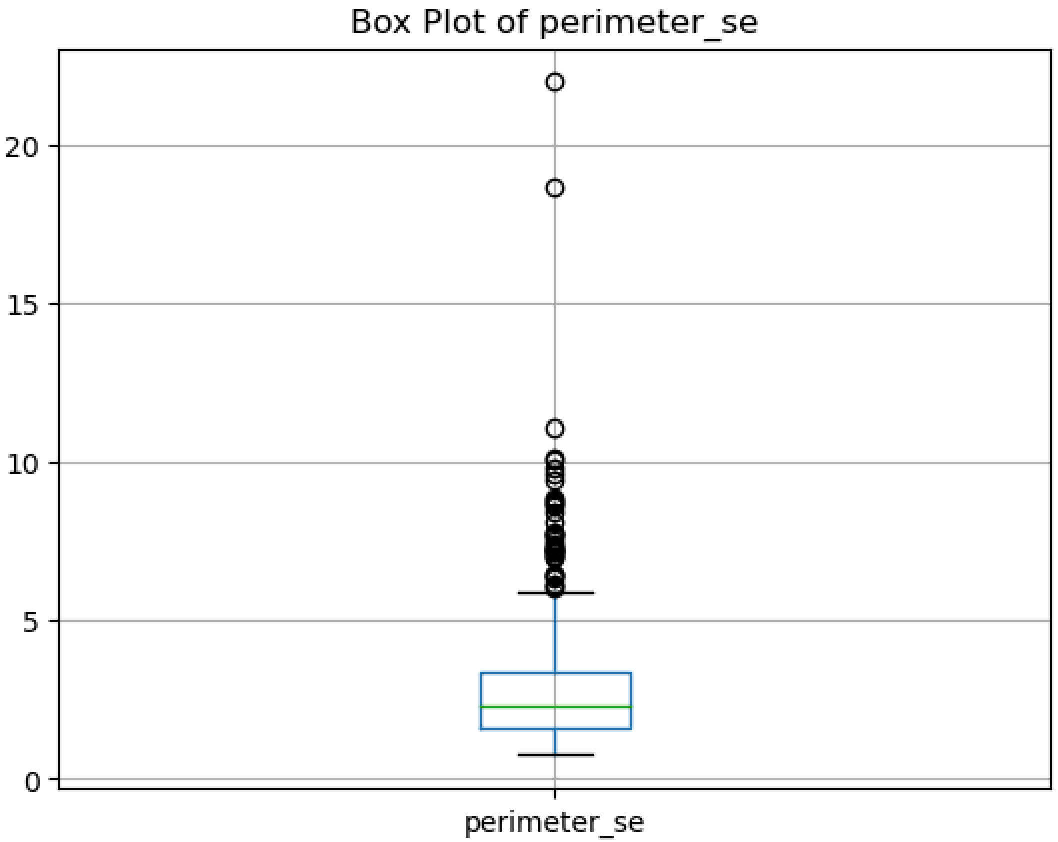


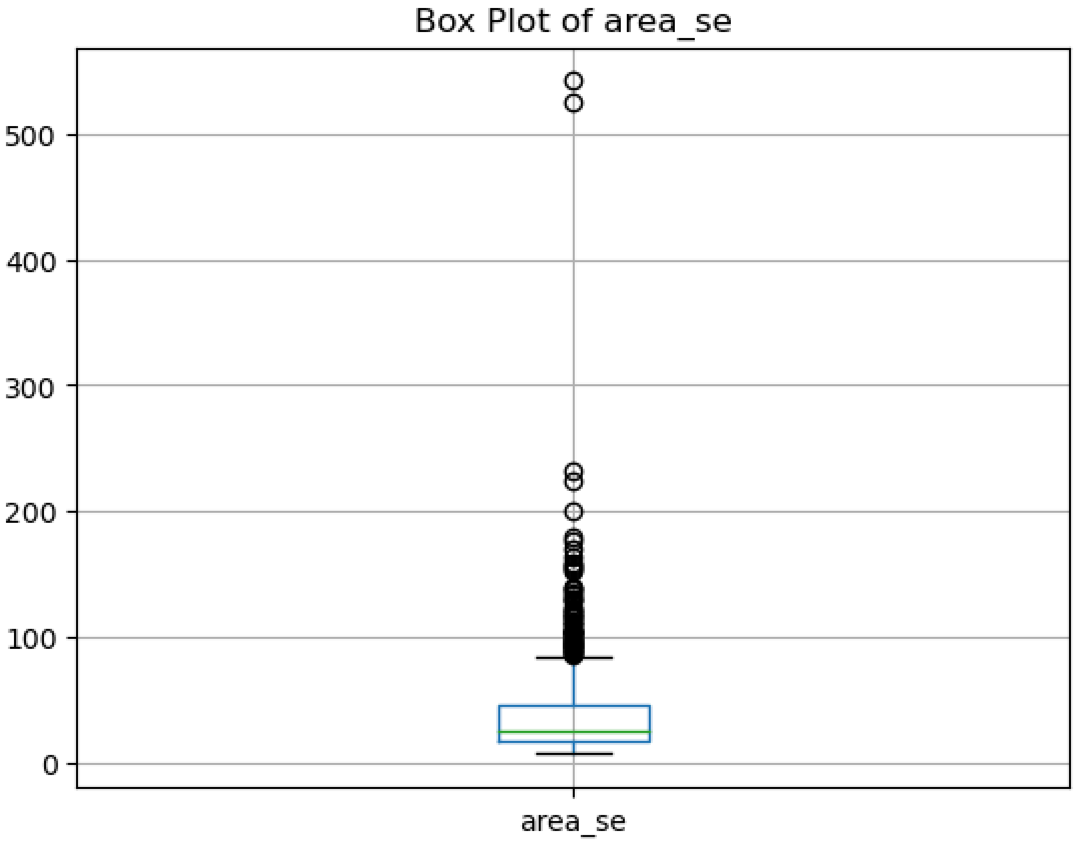
Loading [MathJax]/extensions/Safe.js



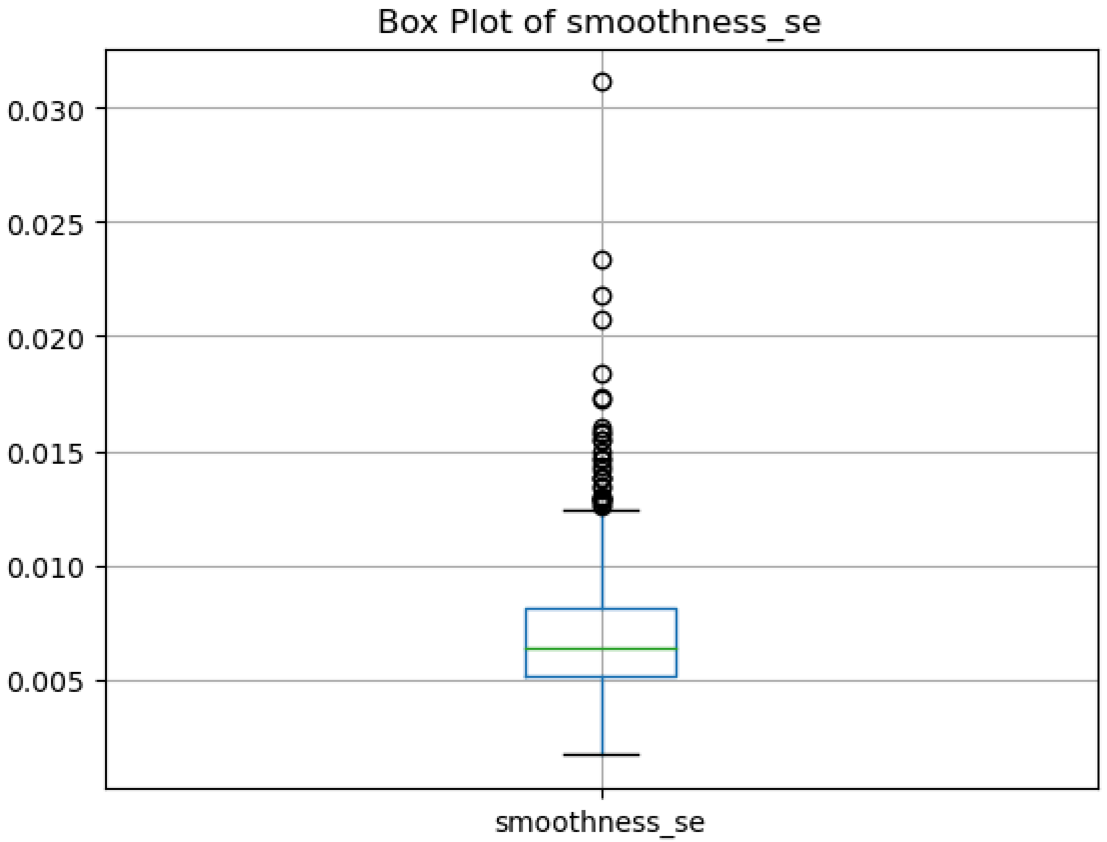


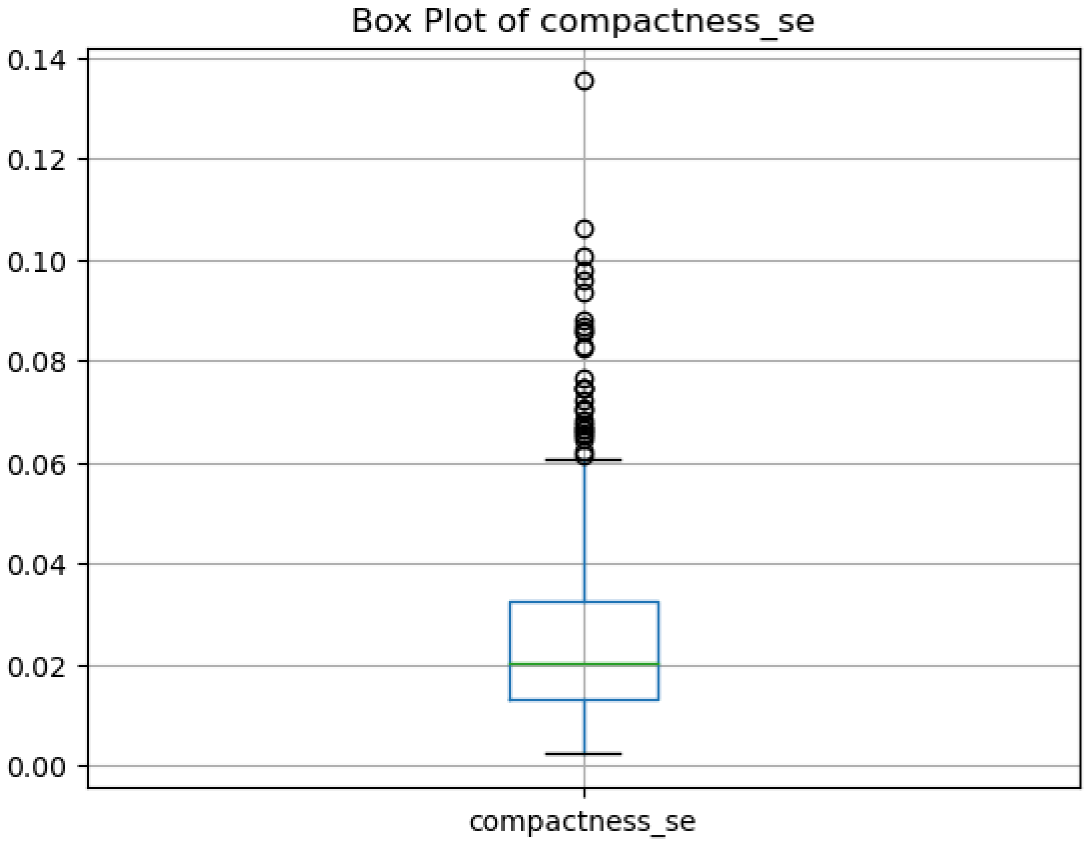
Loading [MathJax]/extensions/Safe.js



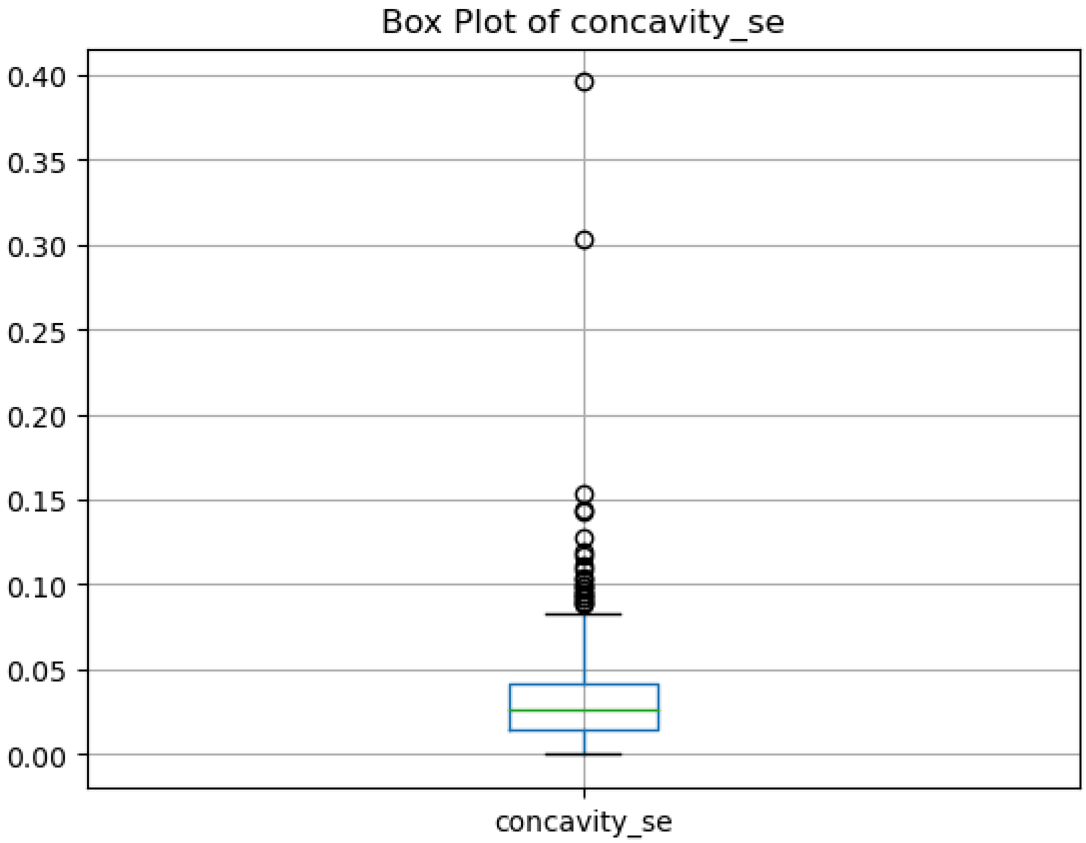


Loading [MathJax]/extensions/Safe.js

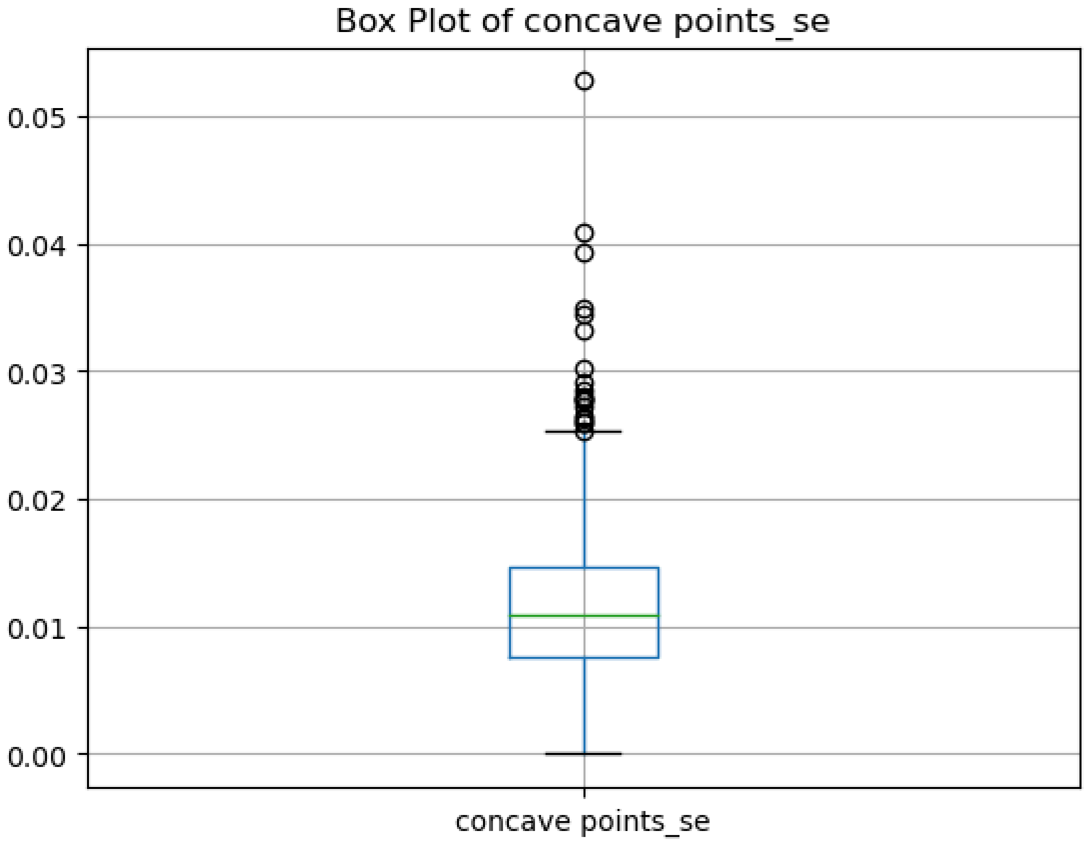




Loading [MathJax]/extensions/Safe.js

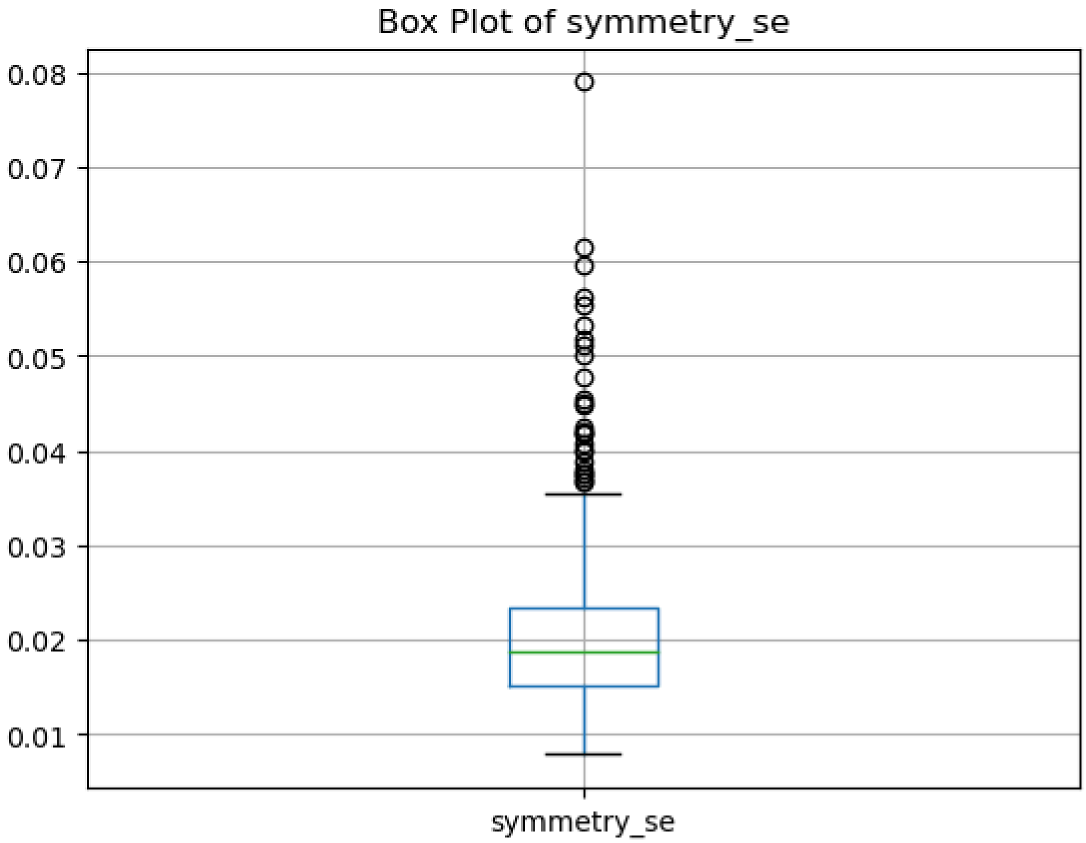


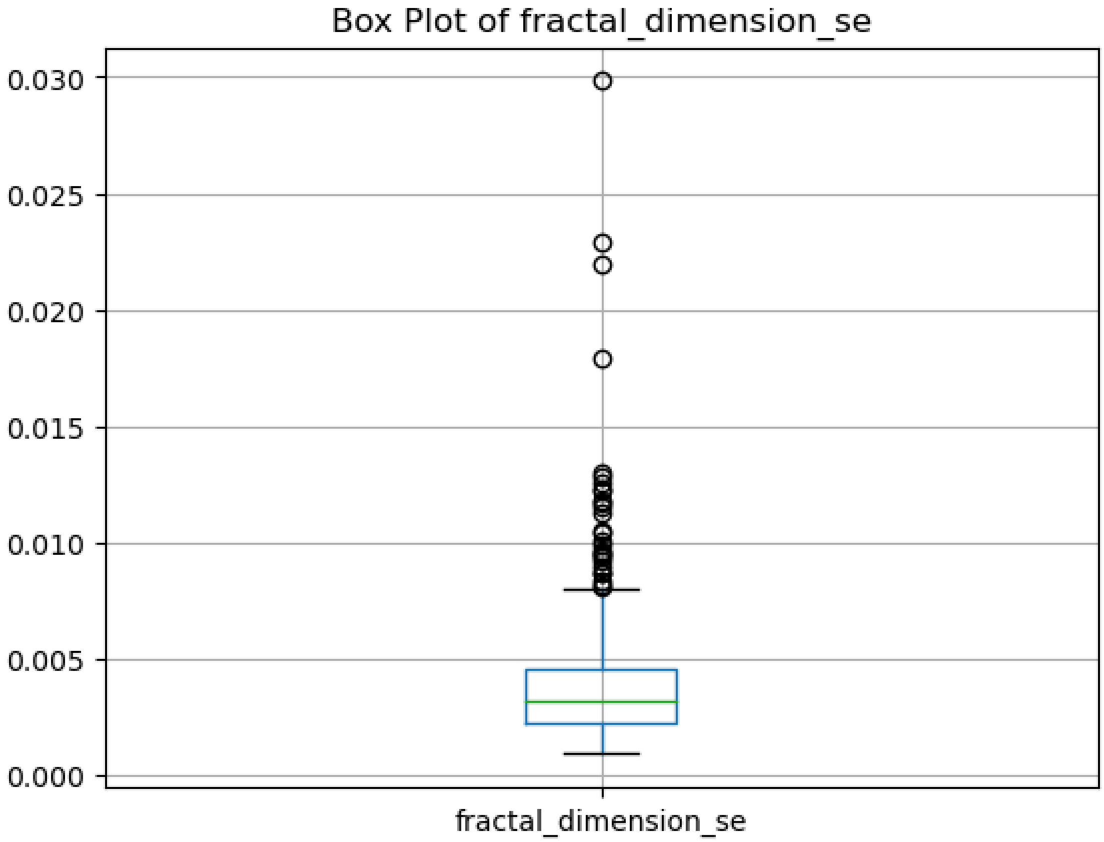
Box Plot of concavity_se

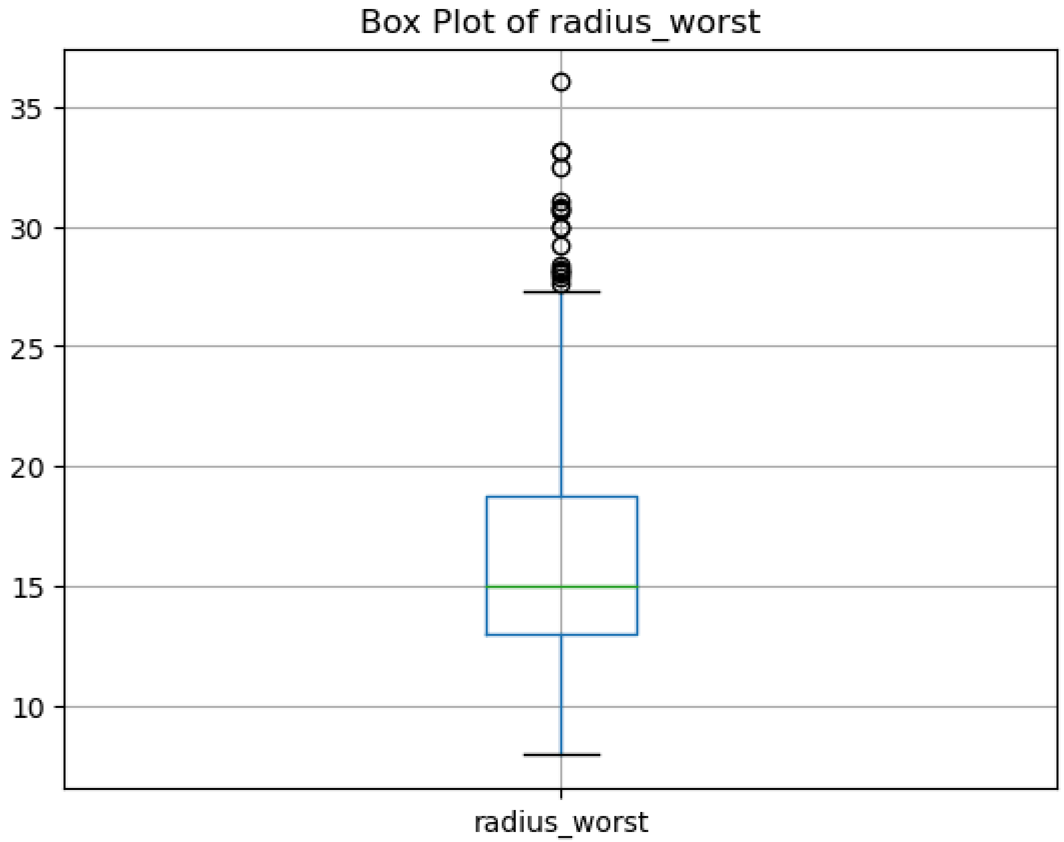


Box Plot of concave points_se

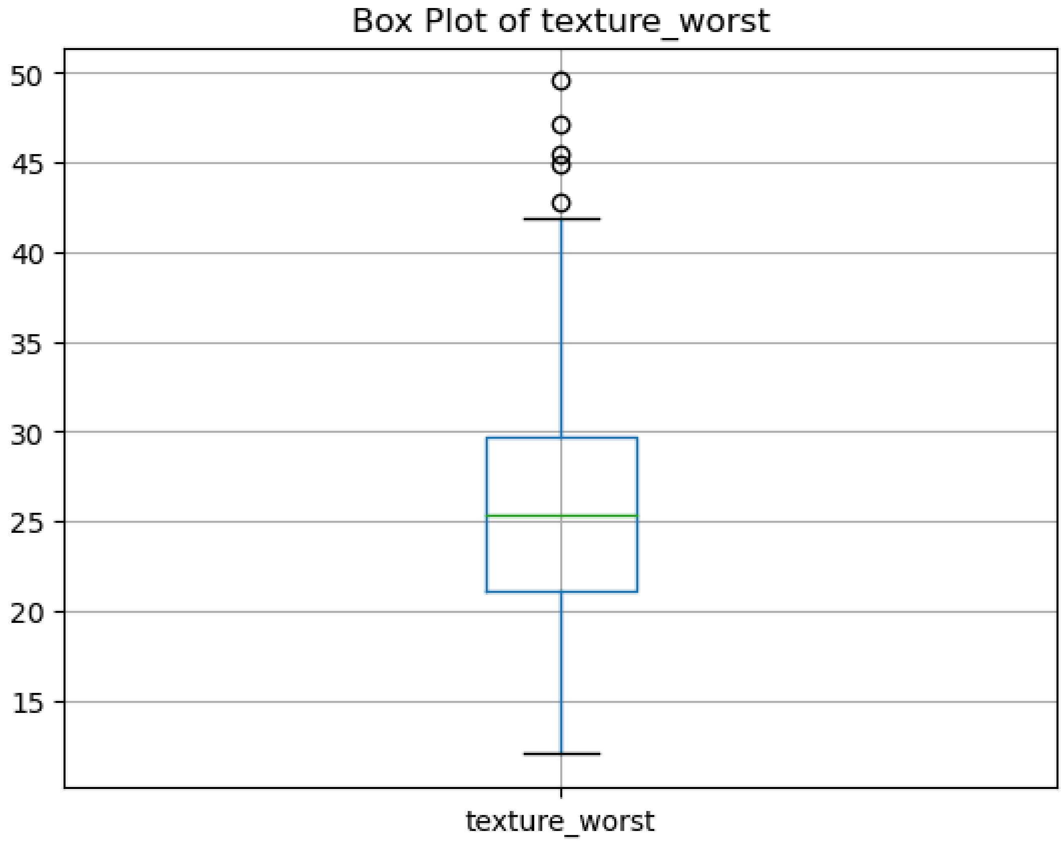
Loading [MathJax]/extensions/Safe.js


Box Plot of symmetry_se
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
symmetry_se

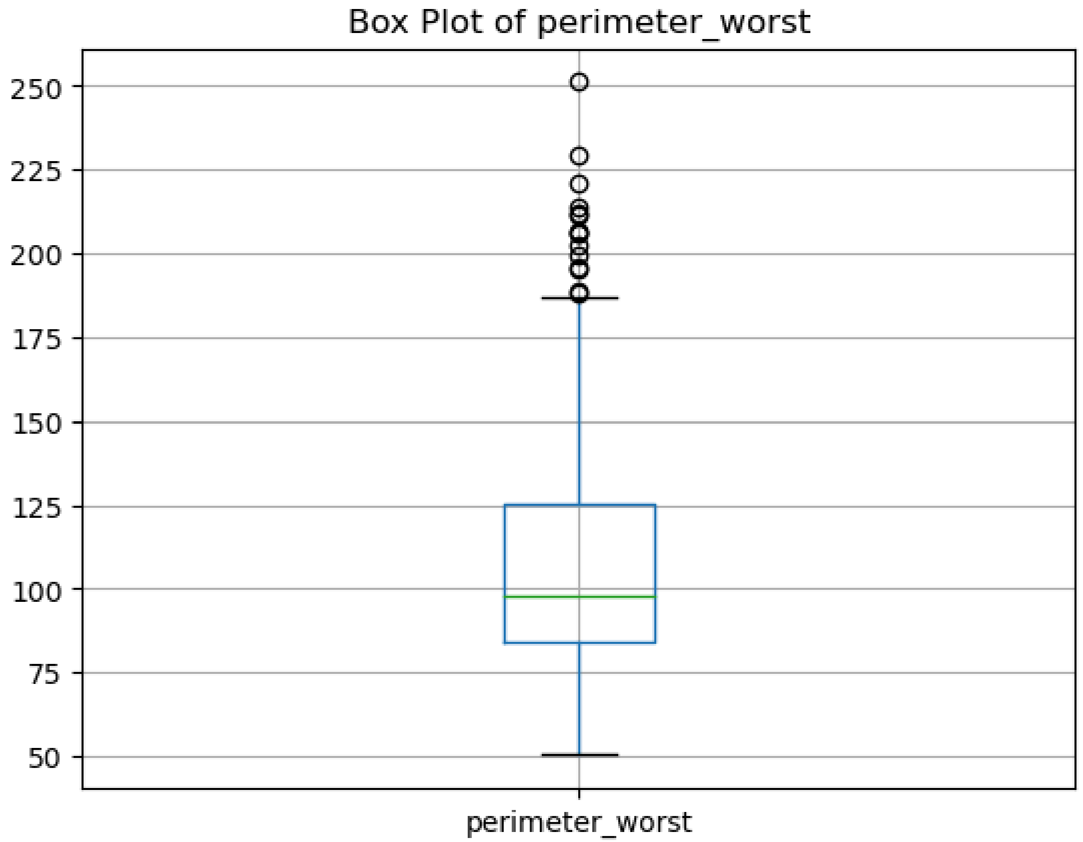

Box Plot of fractal_dimension_se
0.030
0.025
0.020
0.015
0.010
0.005
0.000
fractal_dimension_se

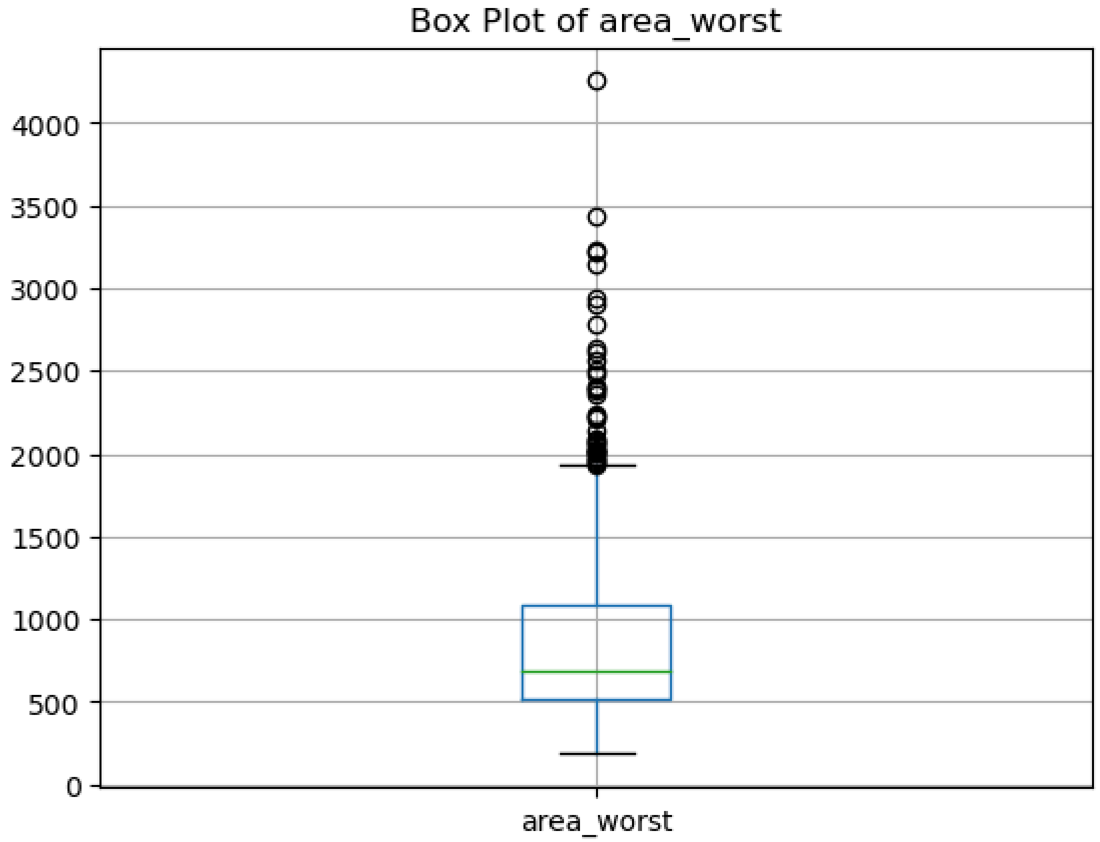


Box Plot of radius_worst

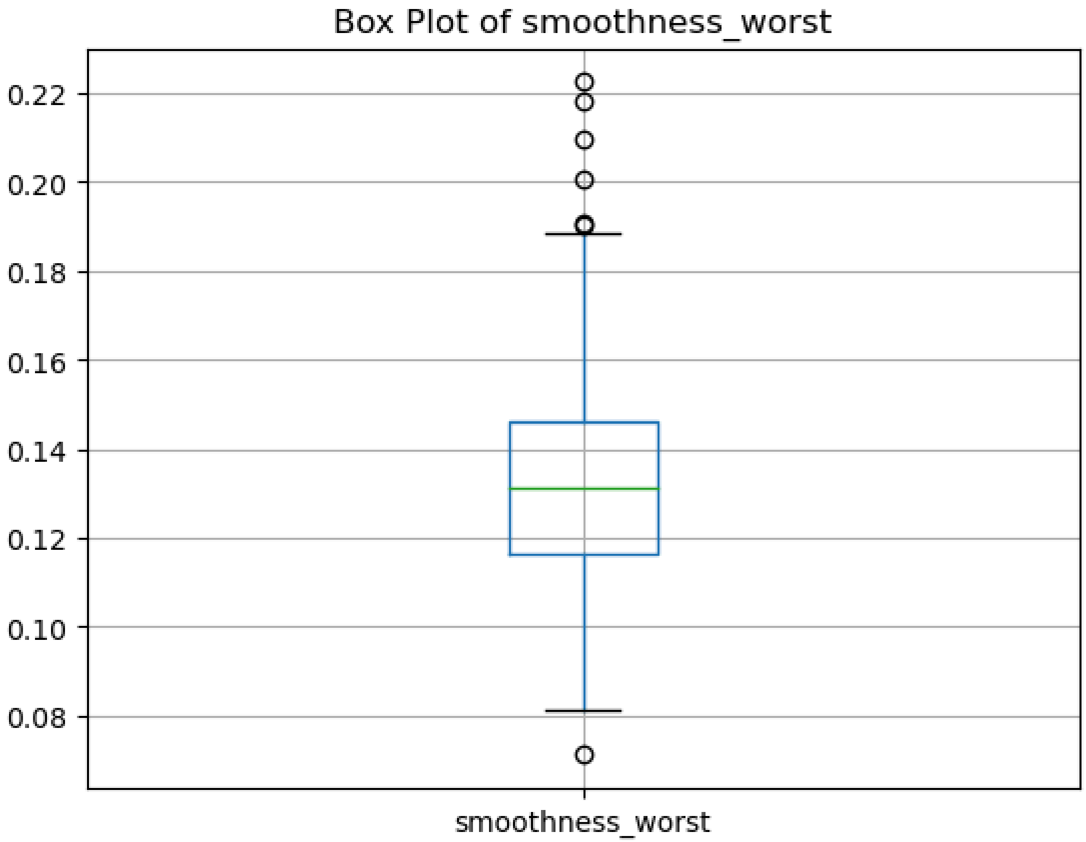


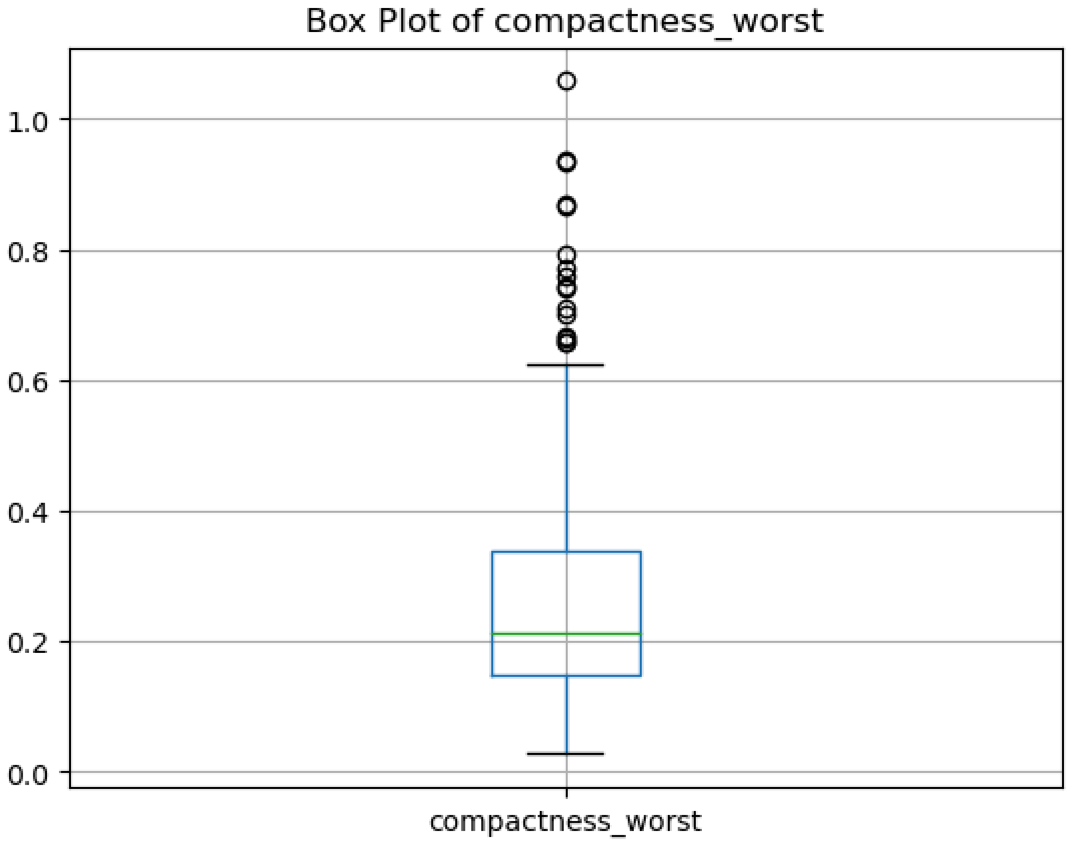
Box Plot of texture_worst



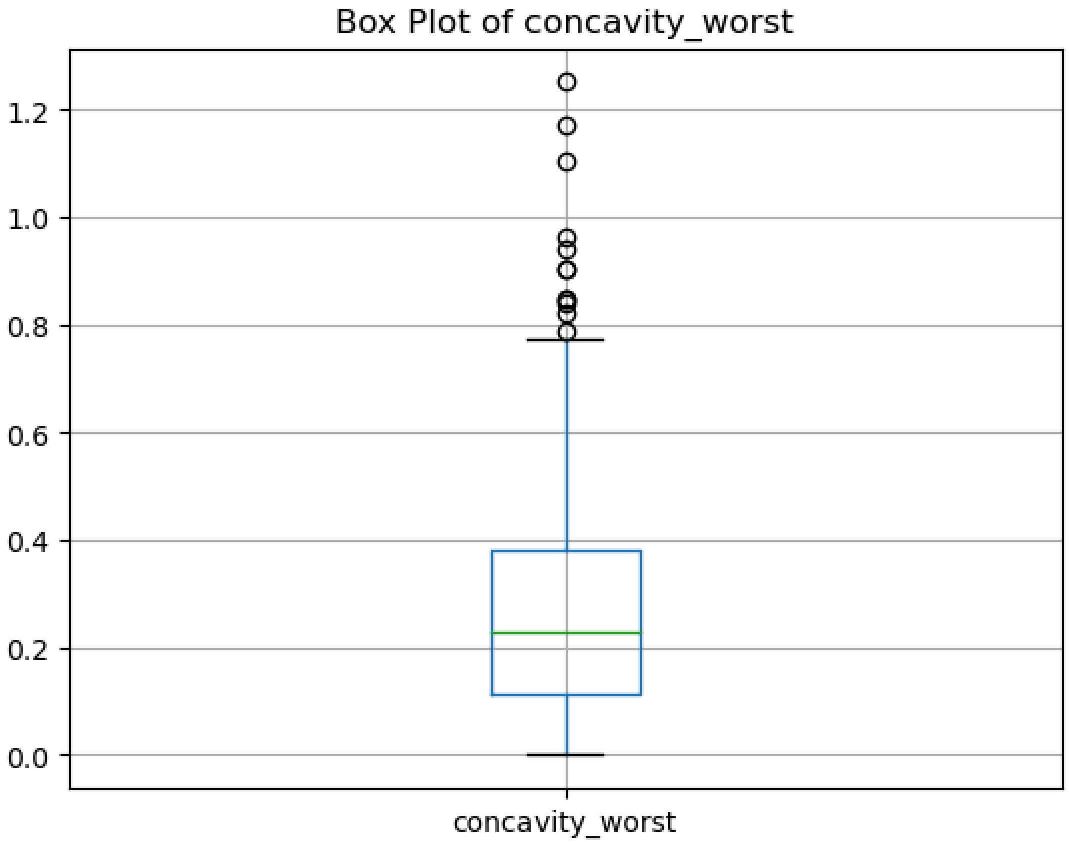


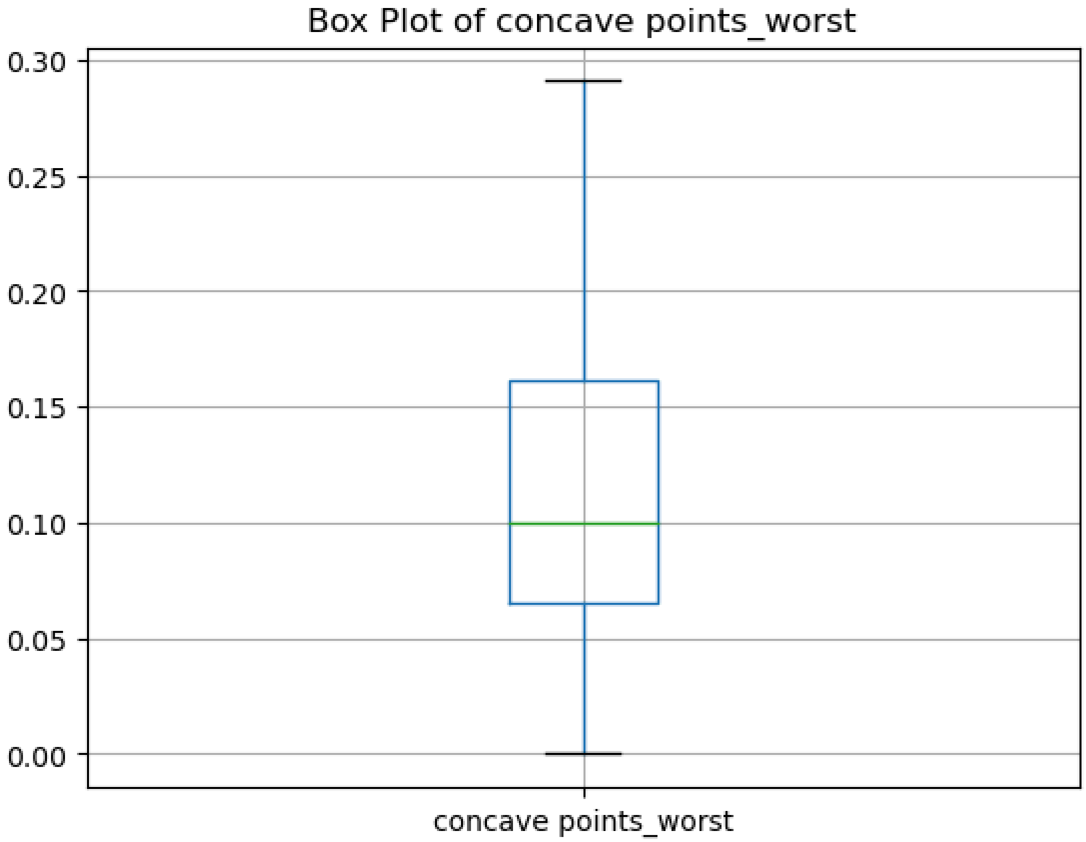
Loading [MathJax]/extensions/Safe.js



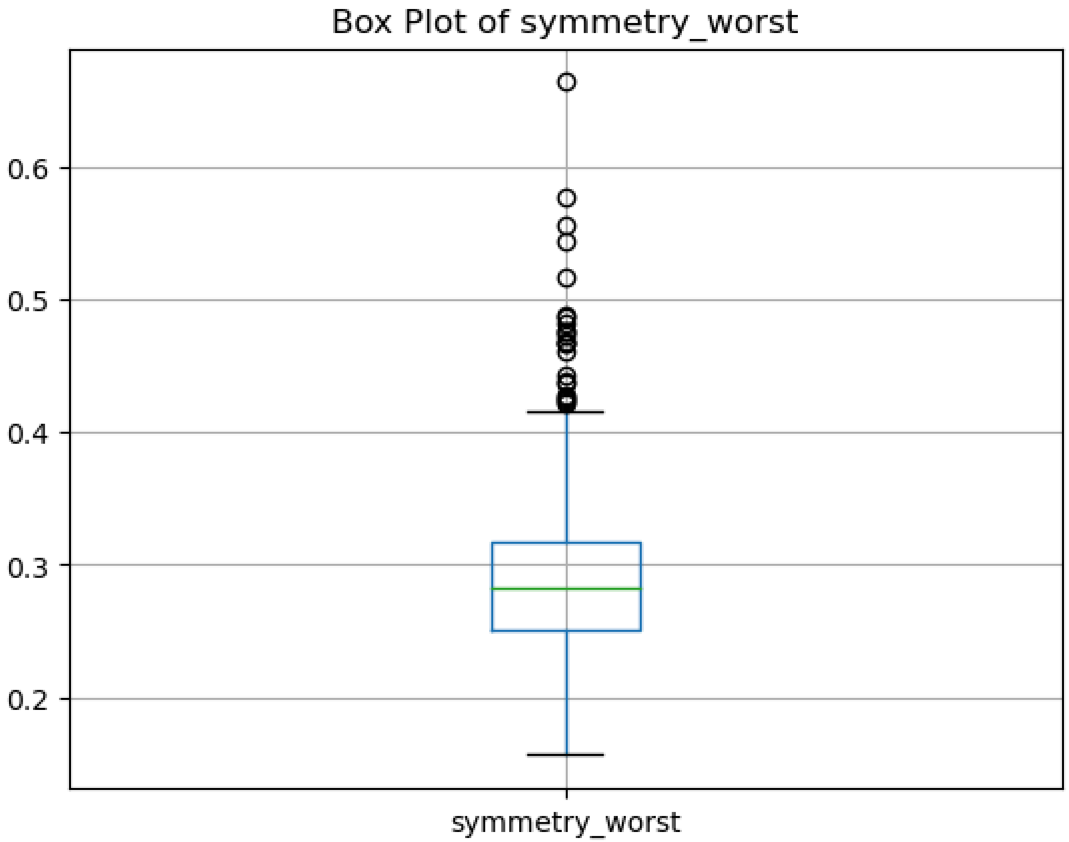


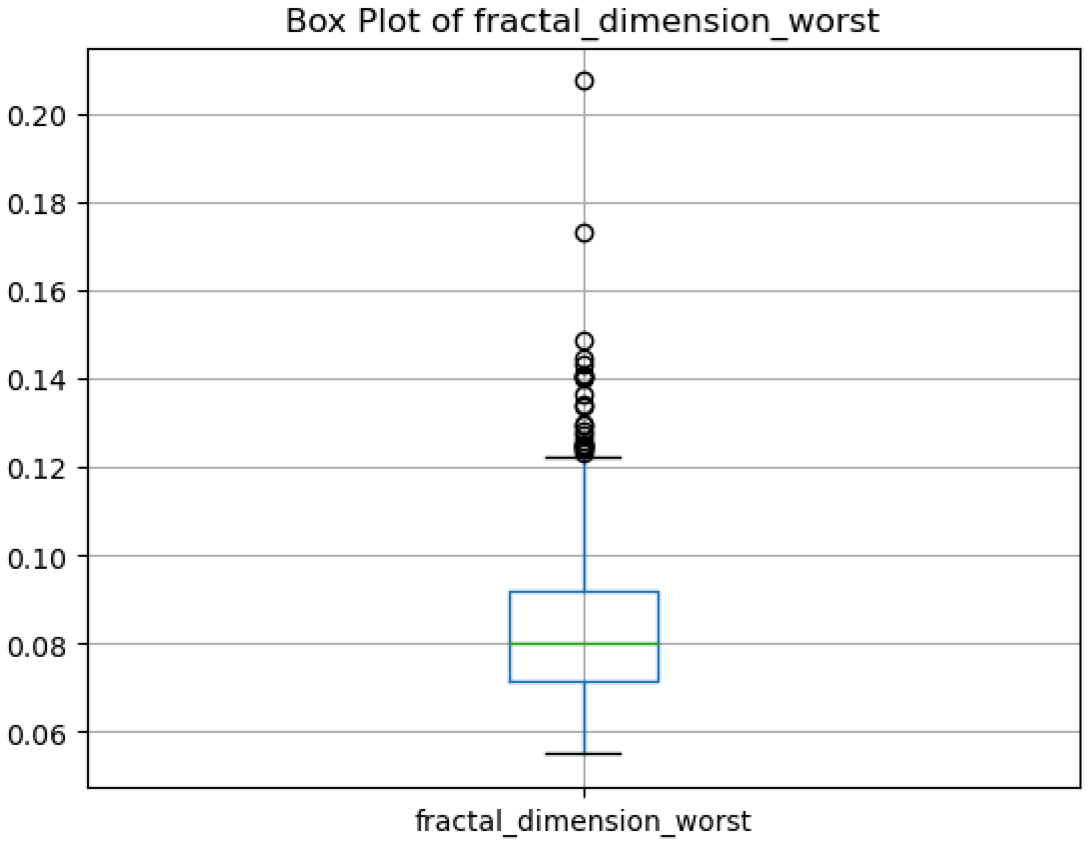
Loading [MathJax]/extensions/Safe.js



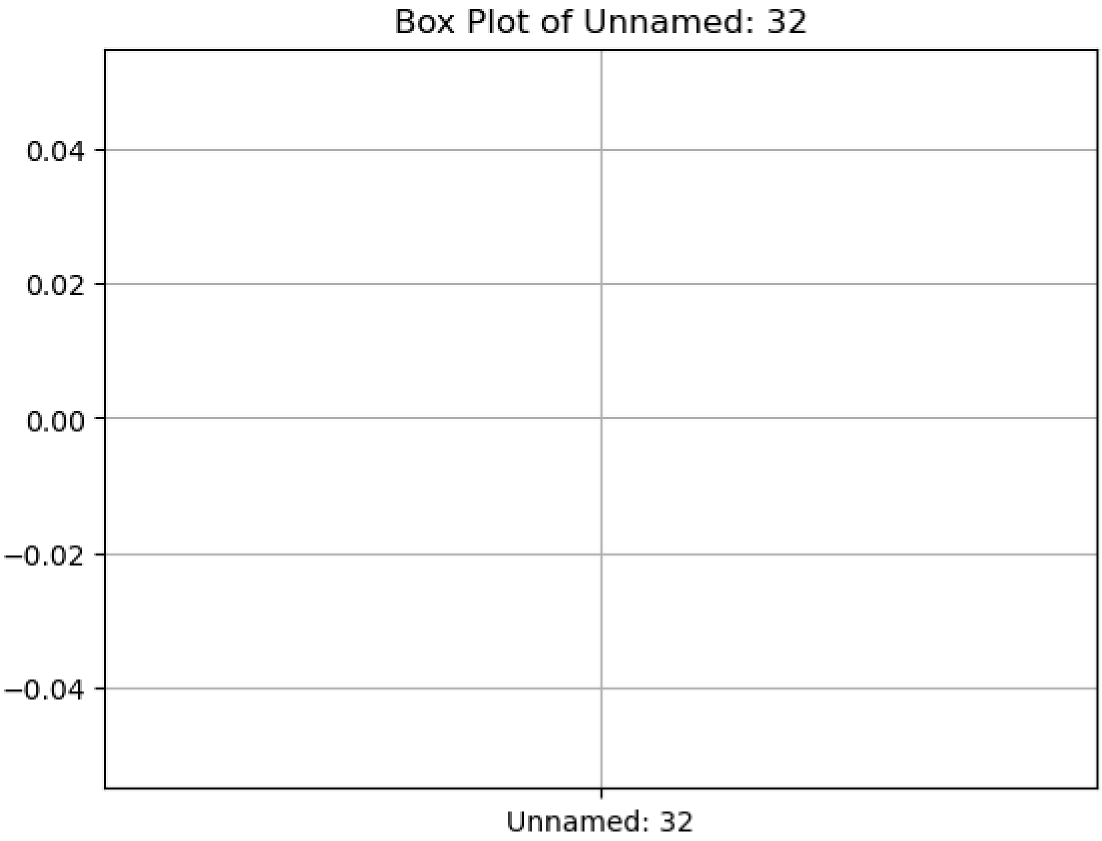


Loading [MathJax]/extensions/Safe.js





Loading [MathJax]/extensions/Safe.js



In [ ]:

```
2.Handling missing values of the dataset(a most common issue with every data
```

In [13]:

```
missing_values = df.isnull().sum()
print(missing_values)
```

```
id                          0
diagnosis                   0
radius_mean                 0
texture_mean                0
perimeter_mean              0
area_mean                   0
smoothness_mean             0
compactness_mean            0
concavity_mean              0
concave points_mean         0
symmetry_mean               0
fractal_dimension_mean      0
radius_se                   0
texture_se                  0
perimeter_se                0
area_se                     0
smoothness_se               0
compactness_se              0
concavity_se                0
concave points_se           0
symmetry_se                 0
fractal_dimension_se        0
radius_worst                0
texture_worst               0
perimeter_worst             0
area_worst                  0
smoothness_worst            0
compactness_worst           0
concavity_worst             0
concave points_worst        0
symmetry_worst              0
fractal_dimension_worst     0
Unnamed: 32               569
dtype: int64
```

```
In [14]: df_cleaned = df.dropna()
```

```
In [ ]: 3.Removing the insignificant columns
```

```
In [20]: df.drop(['id','Unnamed: 32'],axis=1,inplace=True)
```

```
In [21]: df.sample(10)
```

Out[21]:

| | diagnosis | radius_mean | texture_mean | perimeter_mean | area_mean | smoothness_me |
|---|---|---|---|---|---|---|
| **42** | M | 19.07 | 24.81 | 128.30 | 1104.0 | 0.090 |
| **354** | B | 11.14 | 14.07 | 71.24 | 384.6 | 0.072 |
| **488** | B | 11.68 | 16.17 | 75.49 | 420.5 | 0.112 |
| **348** | B | 11.47 | 16.03 | 73.02 | 402.7 | 0.090 |
| **86** | M | 14.48 | 21.46 | 94.25 | 648.2 | 0.094 |
| **104** | B | 10.49 | 19.29 | 67.41 | 336.1 | 0.099 |
| **568** | B | 7.76 | 24.54 | 47.92 | 181.0 | 0.052 |
| **180** | M | 27.22 | 21.87 | 182.10 | 2250.0 | 0.109 |
| **547** | B | 10.26 | 16.58 | 65.85 | 320.8 | 0.088 |
| **370** | M | 16.35 | 23.29 | 109.00 | 840.4 | 0.097 |

10 rows × 31 columns

In [ ]:
```
4.Removing duplicate data
```

In [22]:
```
duplicated_rows = df.duplicated()
```

In [23]:
```
duplicates = df[duplicated_rows]
```

In [24]:
```
cleaned_df = df.drop_duplicates()
```

In [25]:
```
df.drop_duplicates(inplace=True)
```

In [26]:
```
print("Original DataFrame size:", df.shape)
print("Cleaned DataFrame size:", cleaned_df.shape)
```

```
Original DataFrame size: (569, 31)
Cleaned DataFrame size: (569, 31)
```

In [27]:
```
df.duplicated().sum()
```

Out[27]:
```
0
```

In [ ]:
```
5. Encoding the categorical variables
```

In [30]:
```
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()
df['diagnosis']= label_encoder.fit_transform(df['diagnosis'])

df['diagnosis'].unique()
```

Out[30]:
```
array([1, 0])
```

Loading [MathJax]/extensions/Safe.js

In [32]:
```
df['diagnosis'].value_counts()

df.head()
```

Out[32]:

| | diagnosis | radius_mean | texture_mean | perimeter_mean | area_mean | smoothness_mea |
|---|---|---|---|---|---|---|
| **0** | 1 | 17.99 | 10.38 | 122.80 | 1001.0 | 0.1184 |
| **1** | 1 | 20.57 | 17.77 | 132.90 | 1326.0 | 0.0847 |
| **2** | 1 | 19.69 | 21.25 | 130.00 | 1203.0 | 0.1096 |
| **3** | 1 | 11.42 | 20.38 | 77.58 | 386.1 | 0.1425 |
| **4** | 1 | 20.29 | 14.34 | 135.10 | 1297.0 | 0.1003 |

5 rows × 31 columns

In [33]:
```
df.dtypes
```

Out[33]:
```
diagnosis                    int32
radius_mean                float64
texture_mean               float64
perimeter_mean             float64
area_mean                  float64
smoothness_mean            float64
compactness_mean           float64
concavity_mean             float64
concave points_mean        float64
symmetry_mean              float64
fractal_dimension_mean     float64
radius_se                  float64
texture_se                 float64
perimeter_se               float64
area_se                    float64
smoothness_se              float64
compactness_se             float64
concavity_se               float64
concave points_se          float64
symmetry_se                float64
fractal_dimension_se       float64
radius_worst               float64
texture_worst              float64
perimeter_worst            float64
area_worst                 float64
smoothness_worst           float64
compactness_worst          float64
concavity_worst            float64
concave points_worst       float64
symmetry_worst             float64
fractal_dimension_worst    float64
dtype: object
```

In [ ]:
```
6. Skewness checking and removing the skewness from the data
```

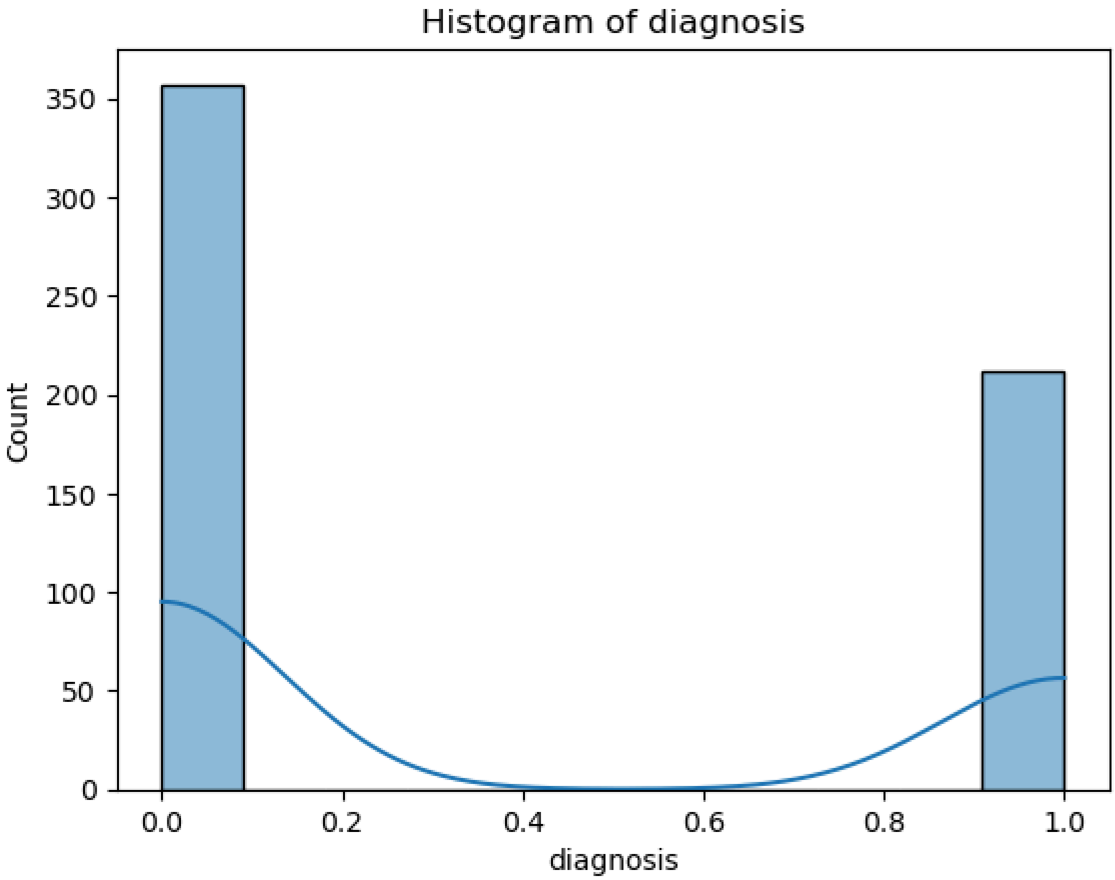
In [41]:

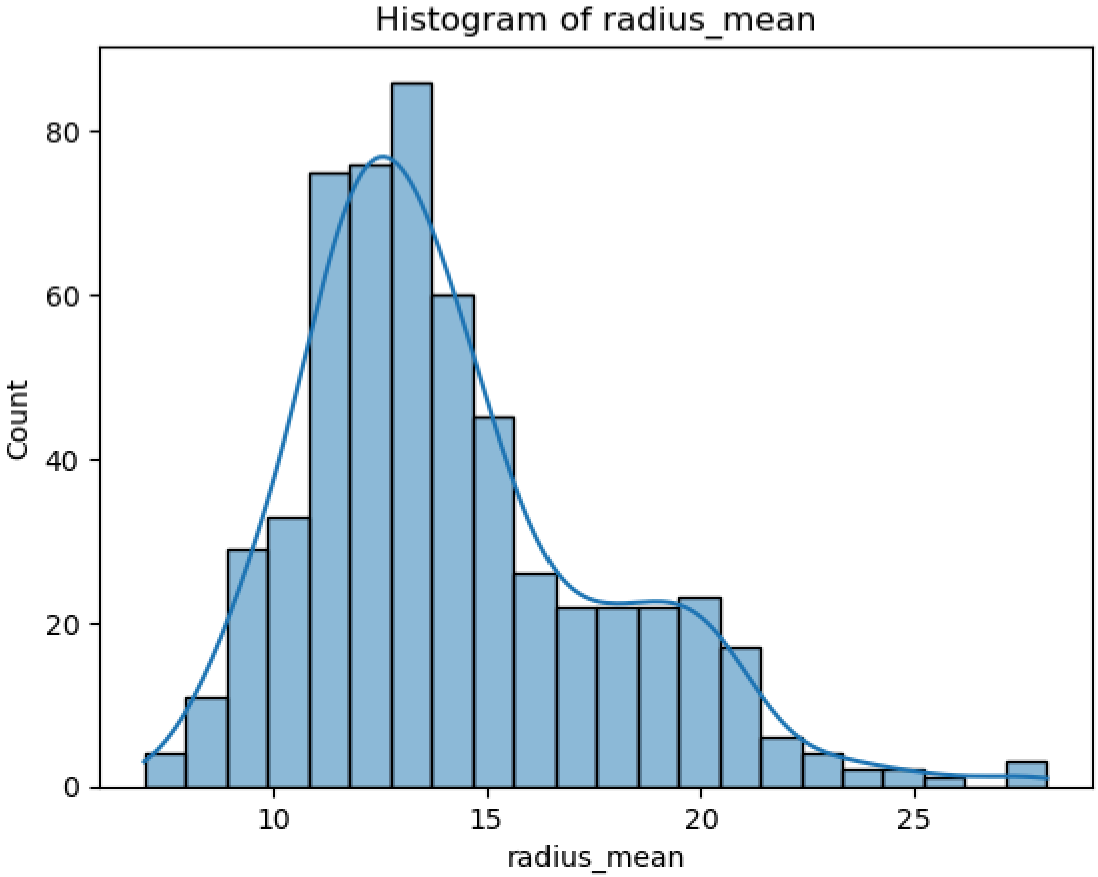
```
skewness = df.skew()
print(skewness)
```

```
diagnosis                  0.528461
radius_mean                0.942380
texture_mean               0.650450
perimeter_mean             0.990650
area_mean                  1.645732
smoothness_mean            0.456324
compactness_mean           1.190123
concavity_mean             1.401180
concave points_mean        1.171180
symmetry_mean              0.725609
fractal_dimension_mean     1.304489
radius_se                  3.088612
texture_se                 1.646444
perimeter_se               3.443615
area_se                    5.447186
smoothness_se              2.314450
compactness_se             1.902221
concavity_se               5.110463
concave points_se          1.444678
symmetry_se                2.195133
fractal_dimension_se       3.923969
radius_worst               1.103115
texture_worst              0.498321
perimeter_worst            1.128164
area_worst                 1.859373
smoothness_worst           0.415426
compactness_worst          1.473555
concavity_worst            1.150237
concave points_worst       0.492616
symmetry_worst             1.433928
fractal_dimension_worst    1.662579
dtype: float64
```

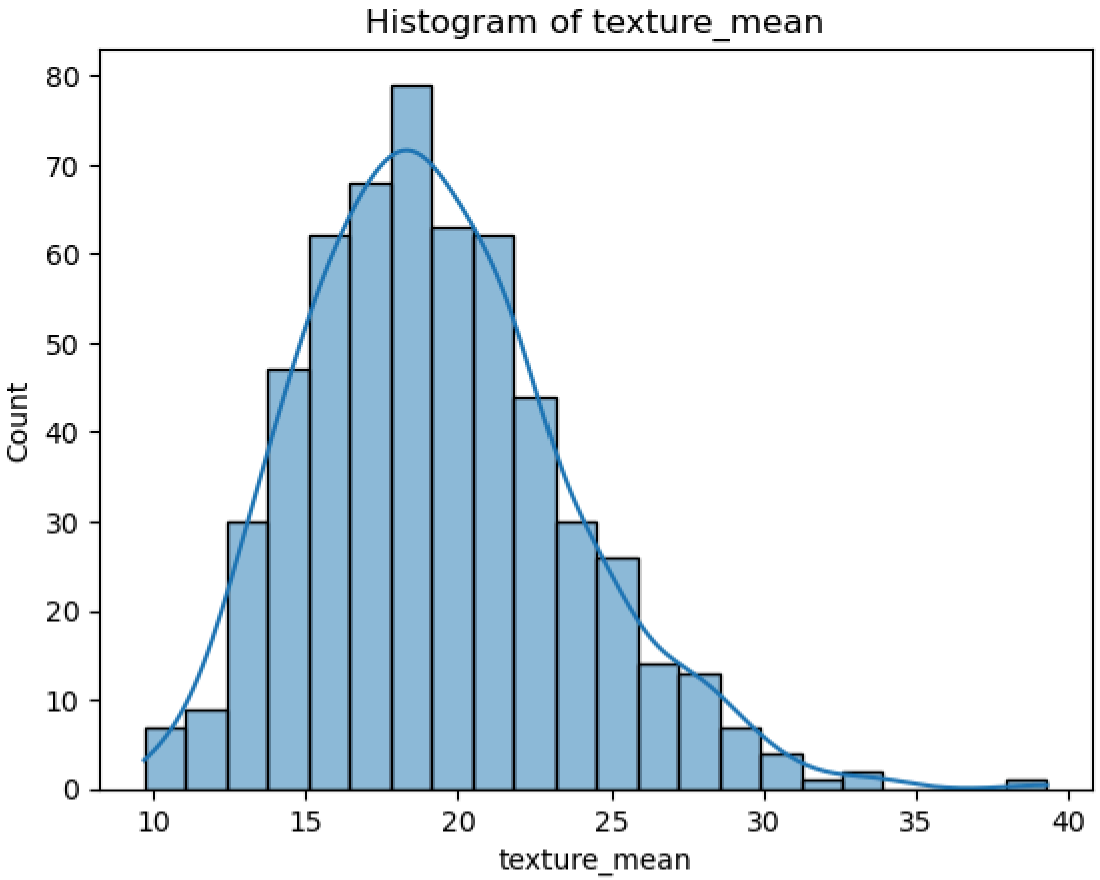
In [42]:

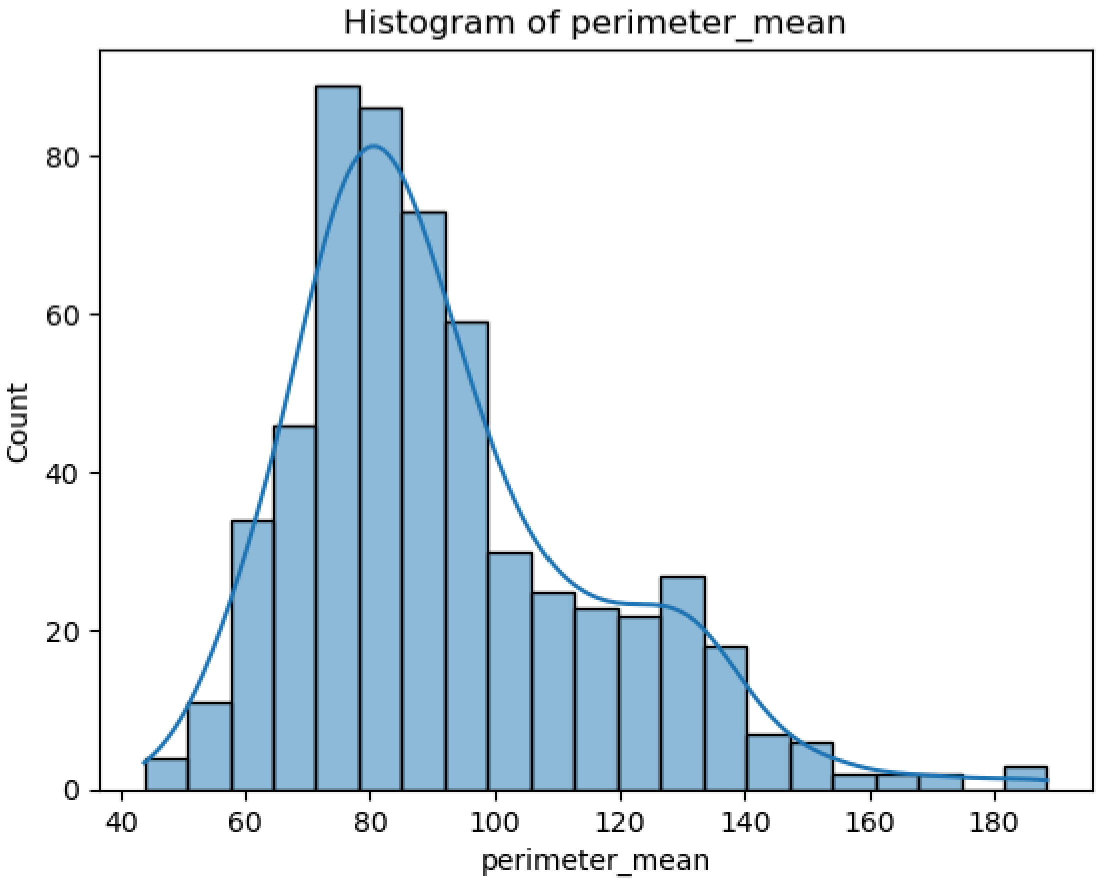
```
import seaborn as sns
import matplotlib.pyplot as plt

for feature in df.columns:
    sns.histplot(df[feature], kde=True)
    plt.title(f'Histogram of {feature}')
    plt.show()
```



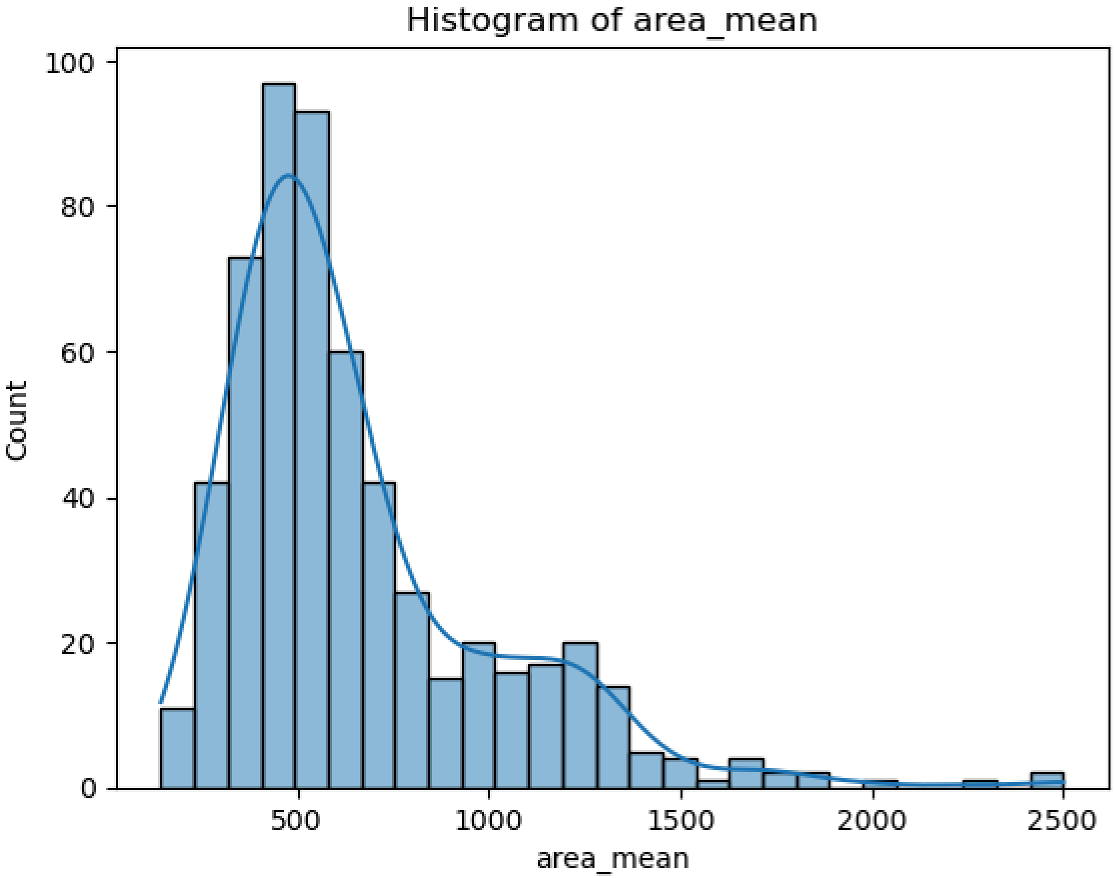


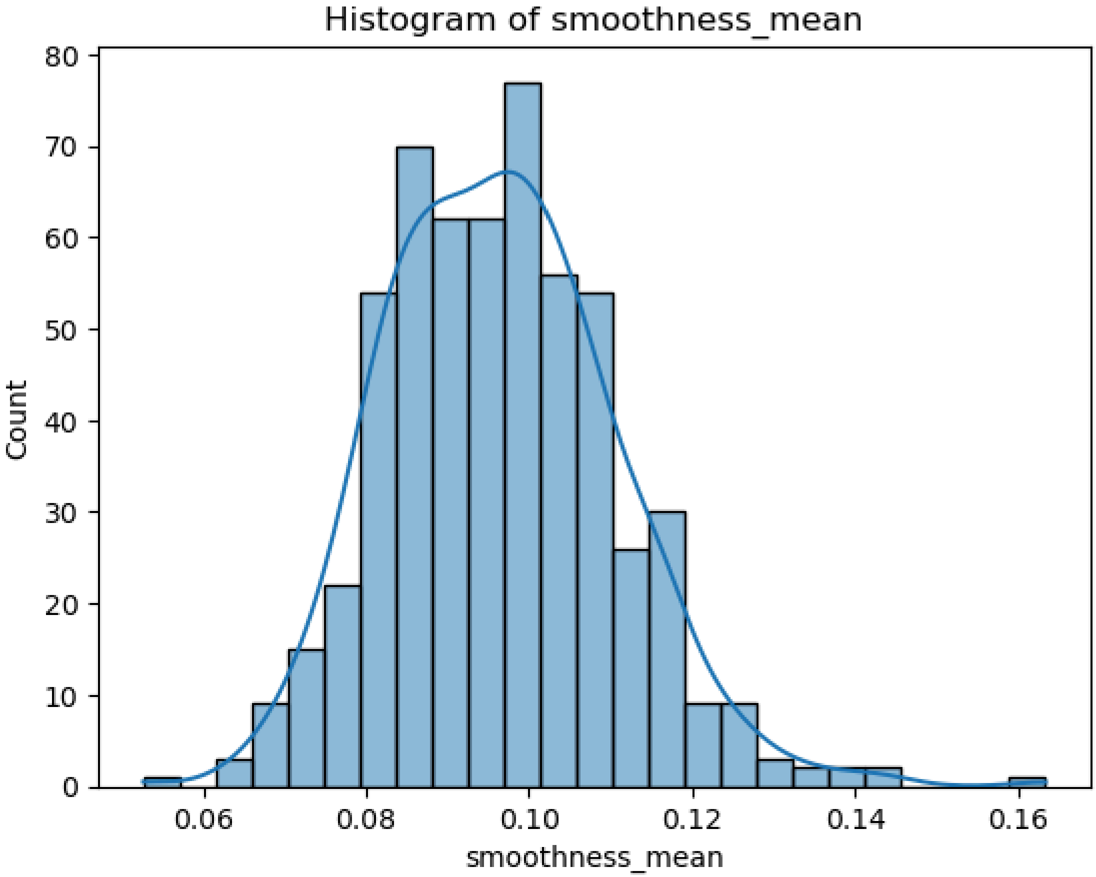
Loading [MathJax]/extensions/Safe.js



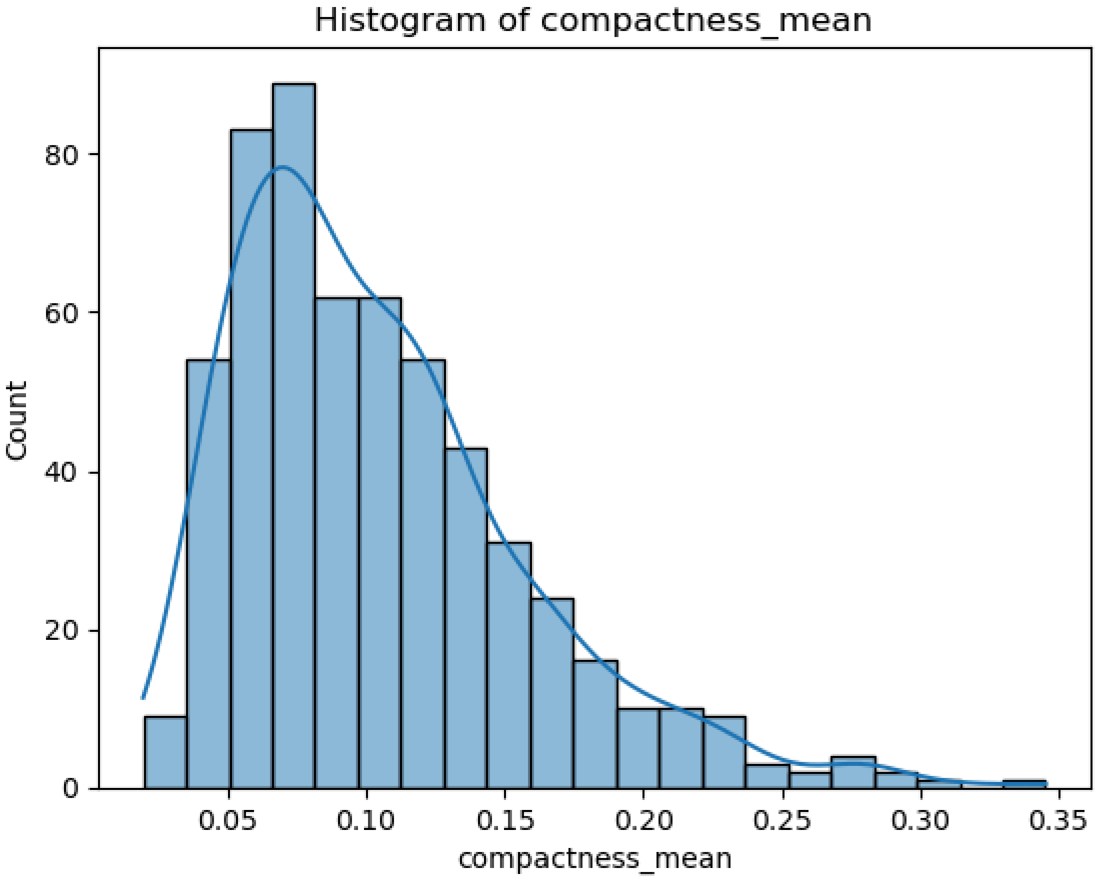


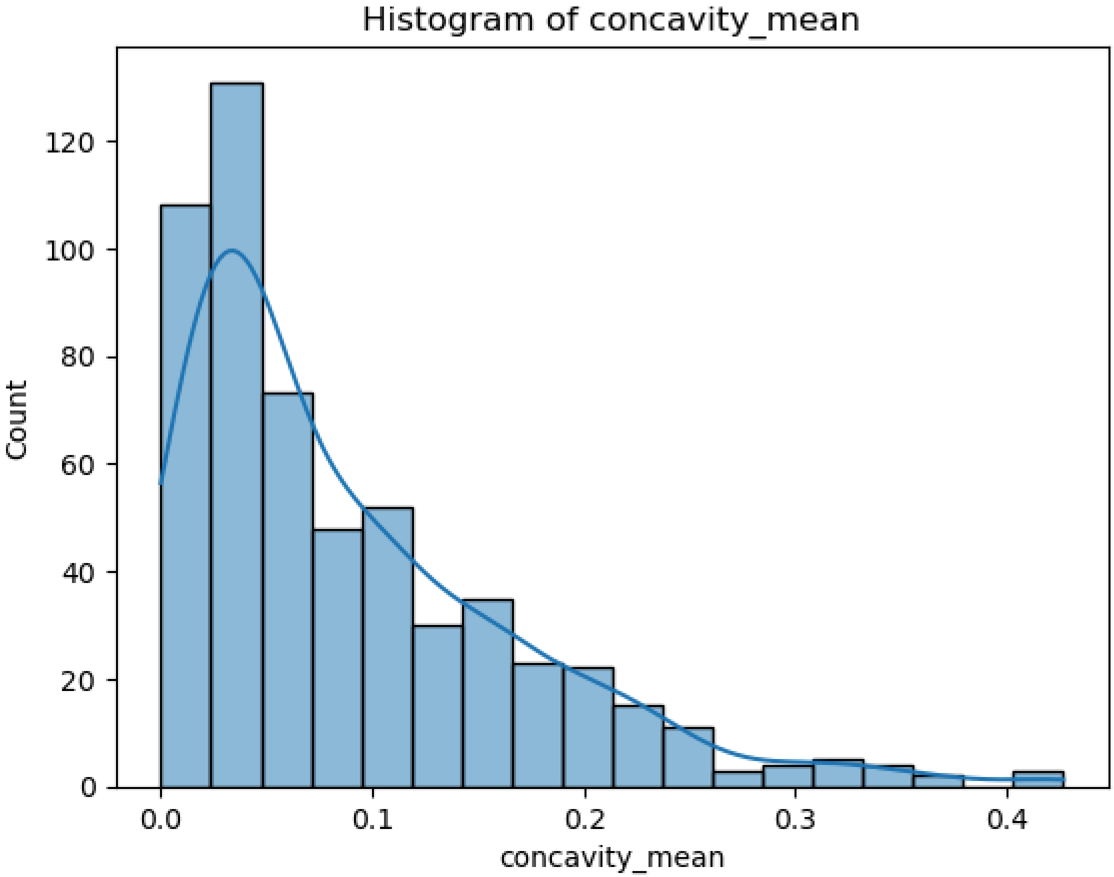
Loading [MathJax]/extensions/Safe.js

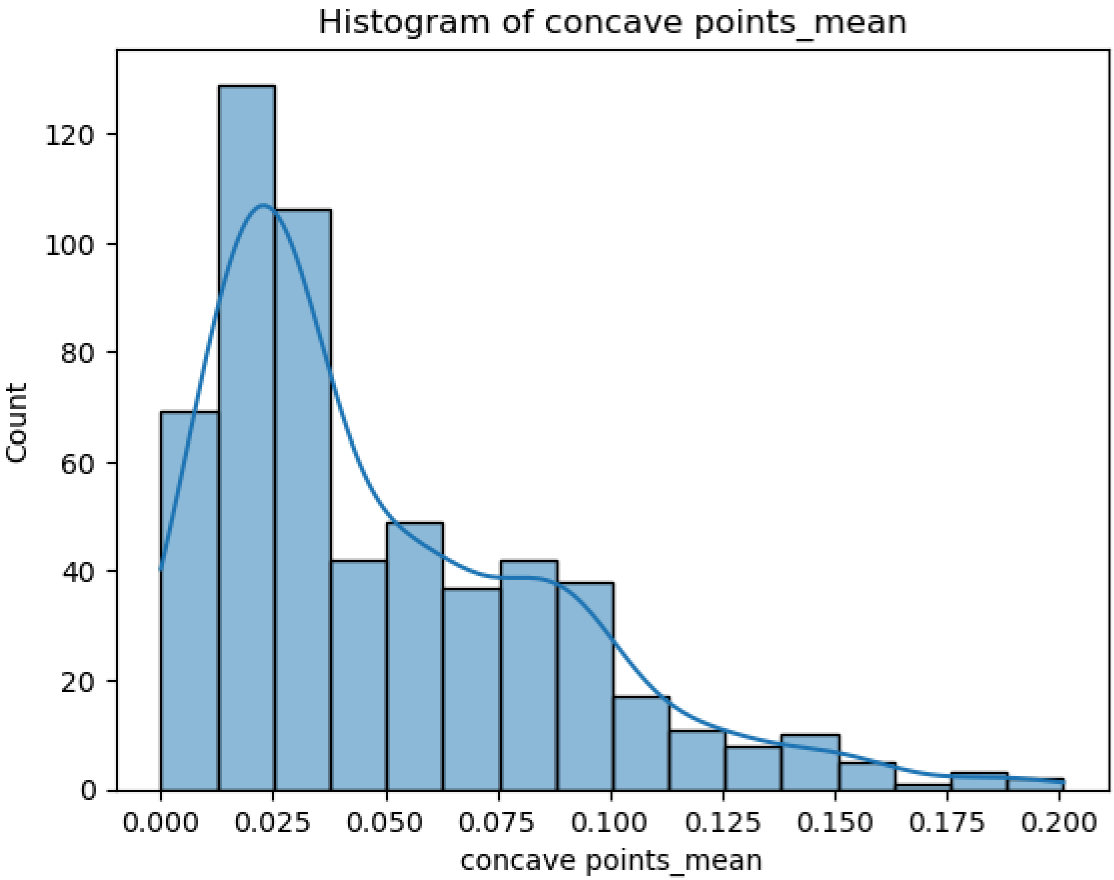


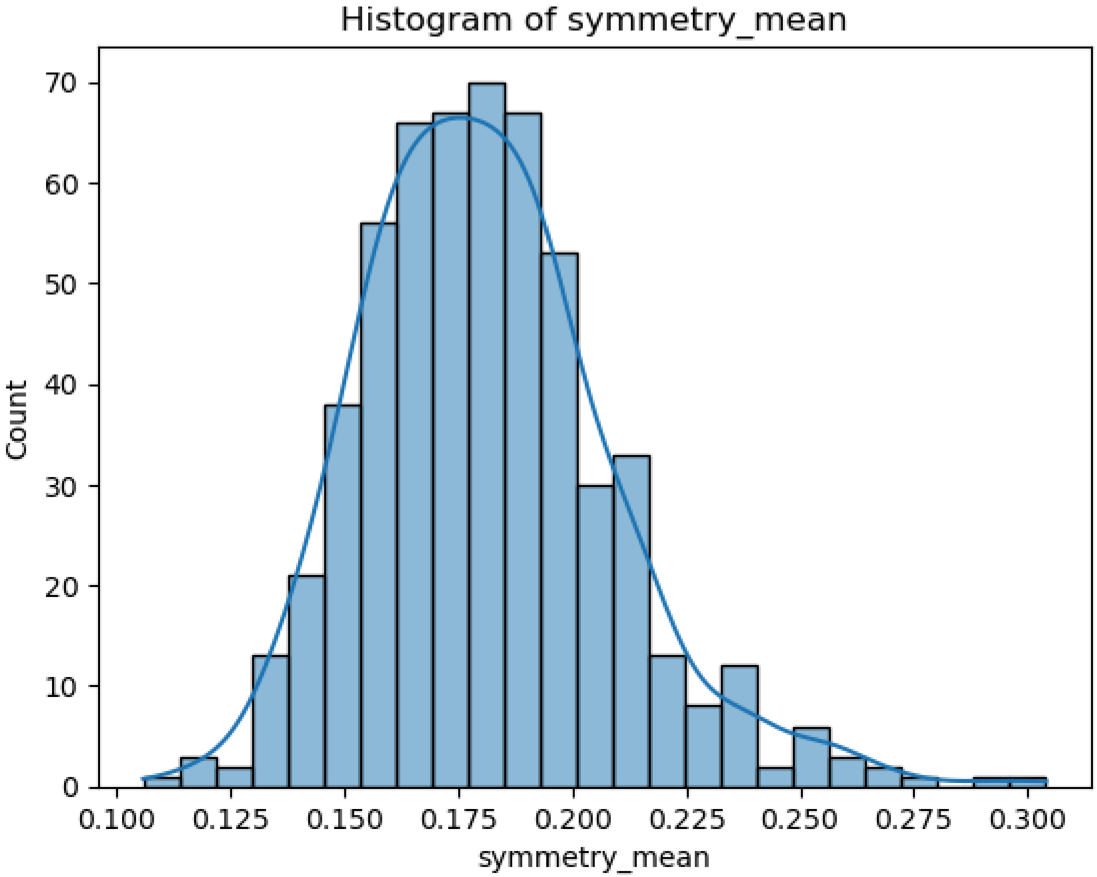


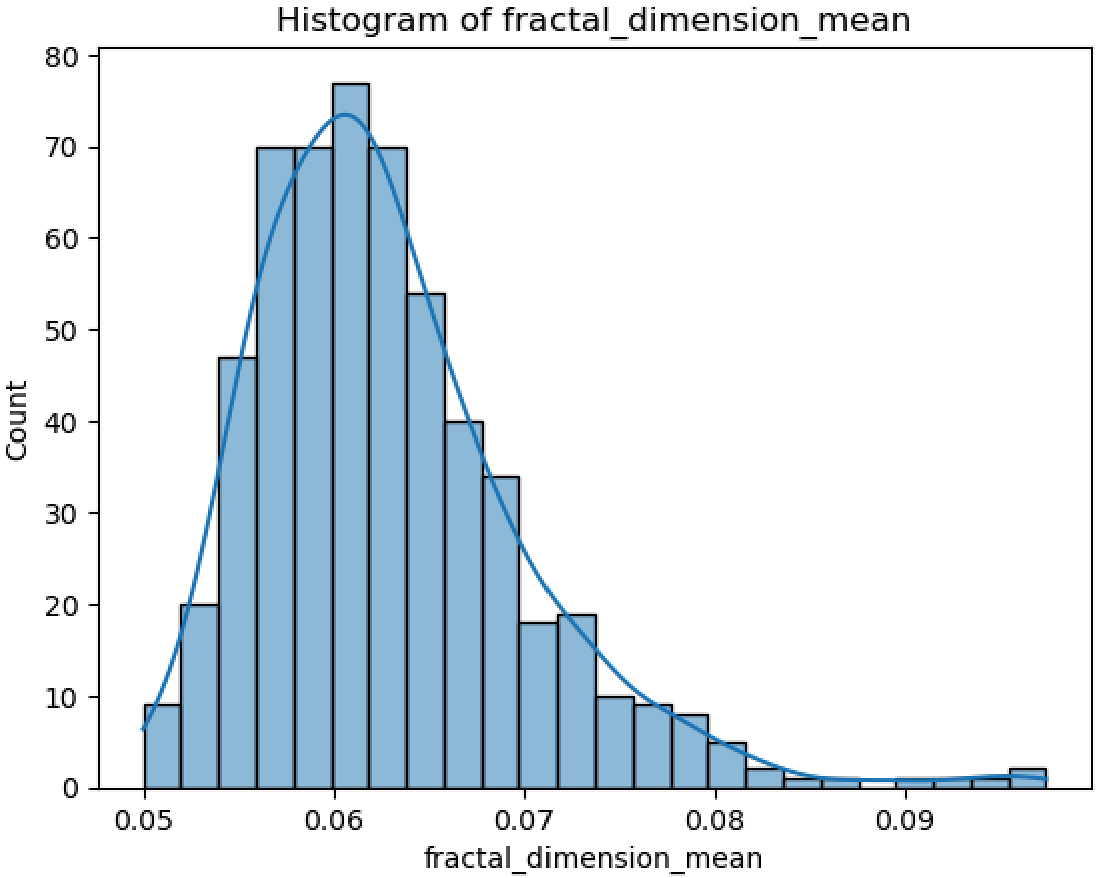
Loading [MathJax]/extensions/Safe.js

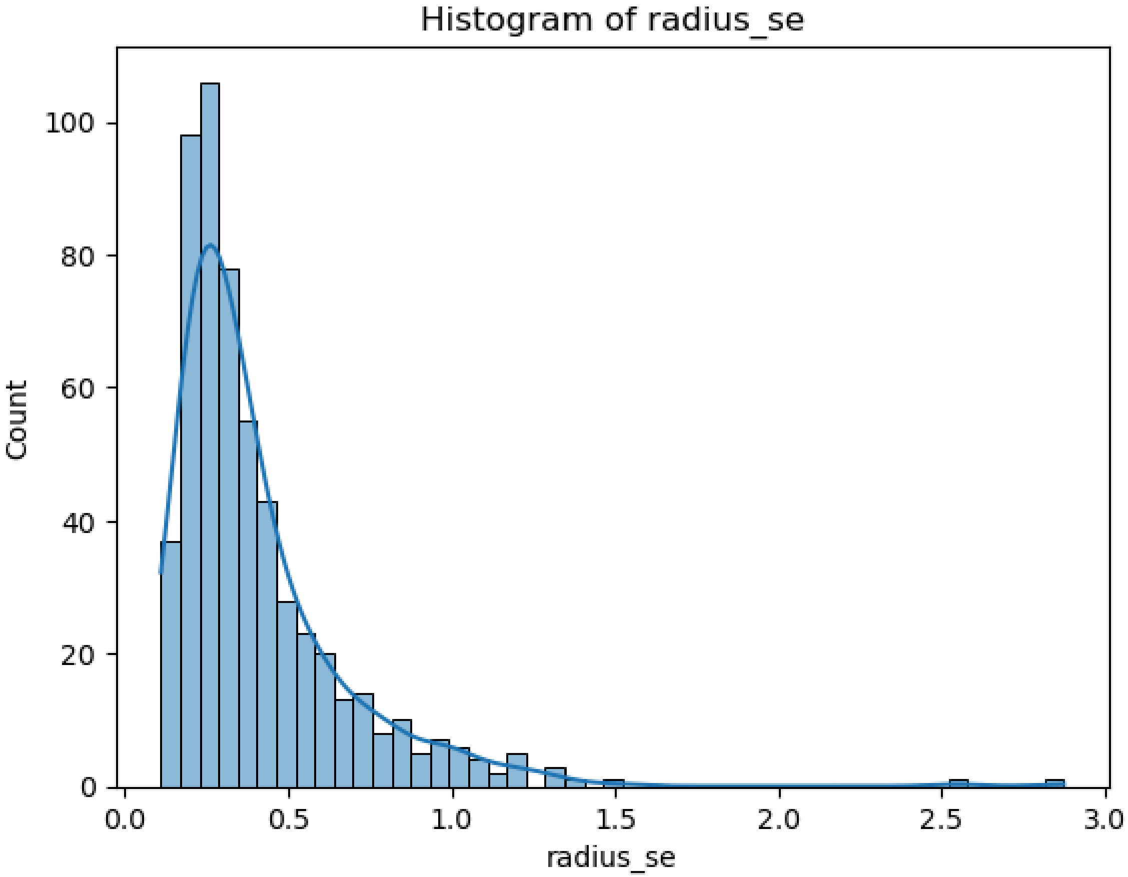

Histogram of compactness_mean
Count
0
20
40
60
80
0.05
0.10
0.15
0.20
0.25
0.30
0.35
compactness_mean

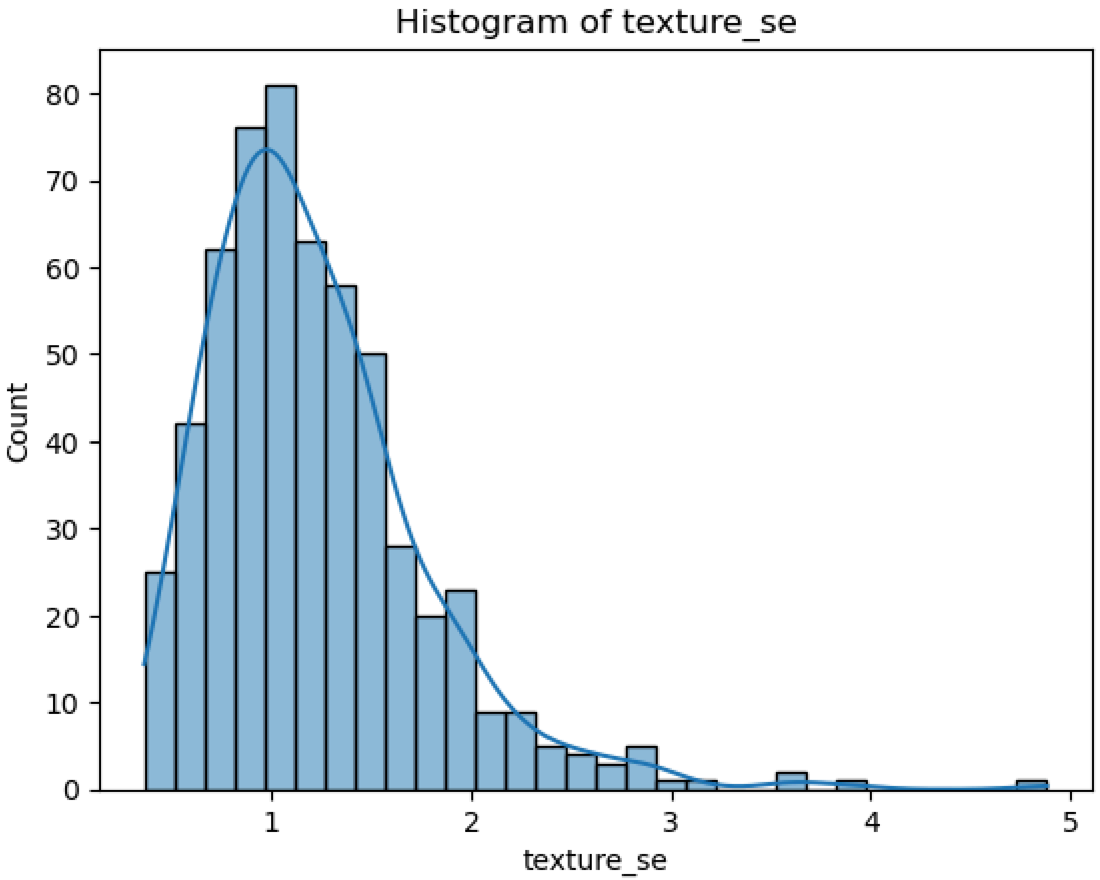

Histogram of concavity_mean
Count
0
20
40
60
80
100
120
0.0
0.1
0.2
0.3
0.4
concavity_mean


Histogram of concave points_mean
Count
0
20
40
60
80
100
120
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
concave points_mean

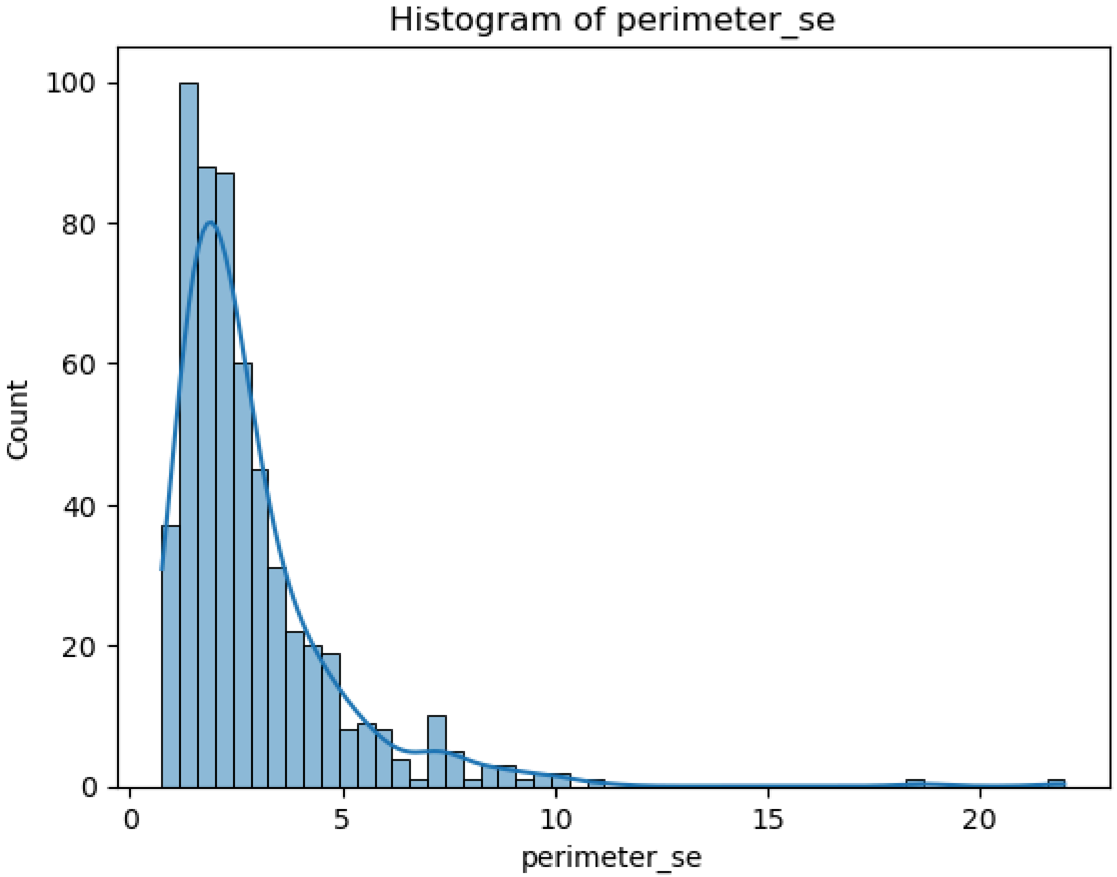

Histogram of symmetry_mean
Count
0
10
20
30
40
50
60
70
0.100
0.125
0.150
0.175
0.200
0.225
0.250
0.275
0.300
symmetry_mean


Histogram of fractal_dimension_mean
Count
0
10
20
30
40
50
60
70
80
0.05
0.06
0.07
0.08
0.09
fractal_dimension_mean

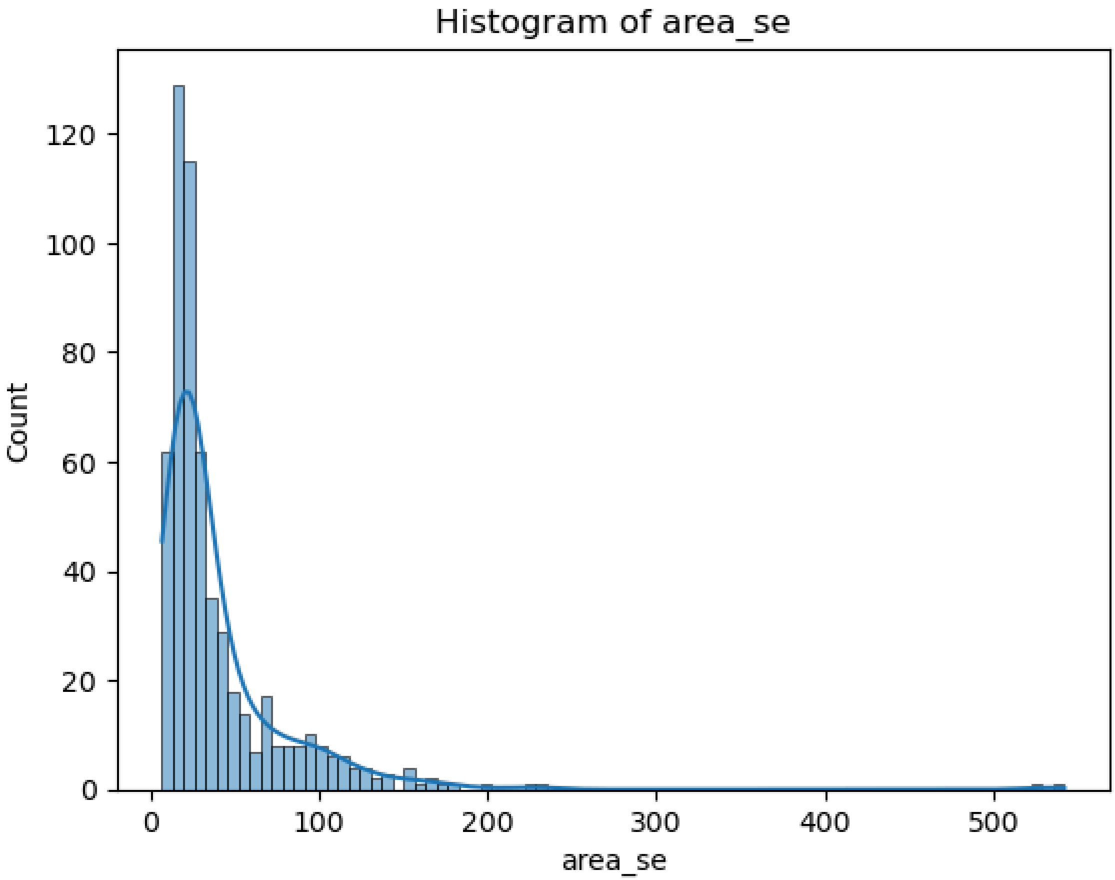

Histogram of radius_se
Count
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
3.0
radius_se

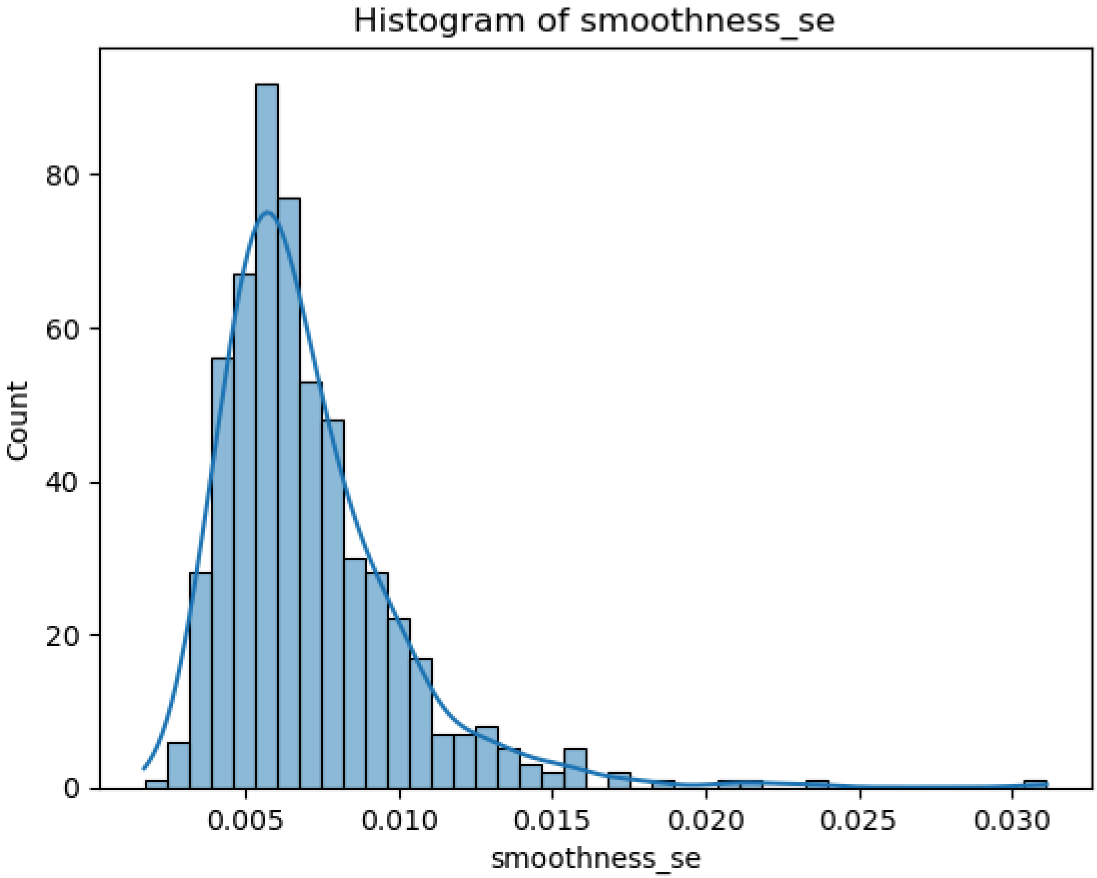


Histogram of texture_se

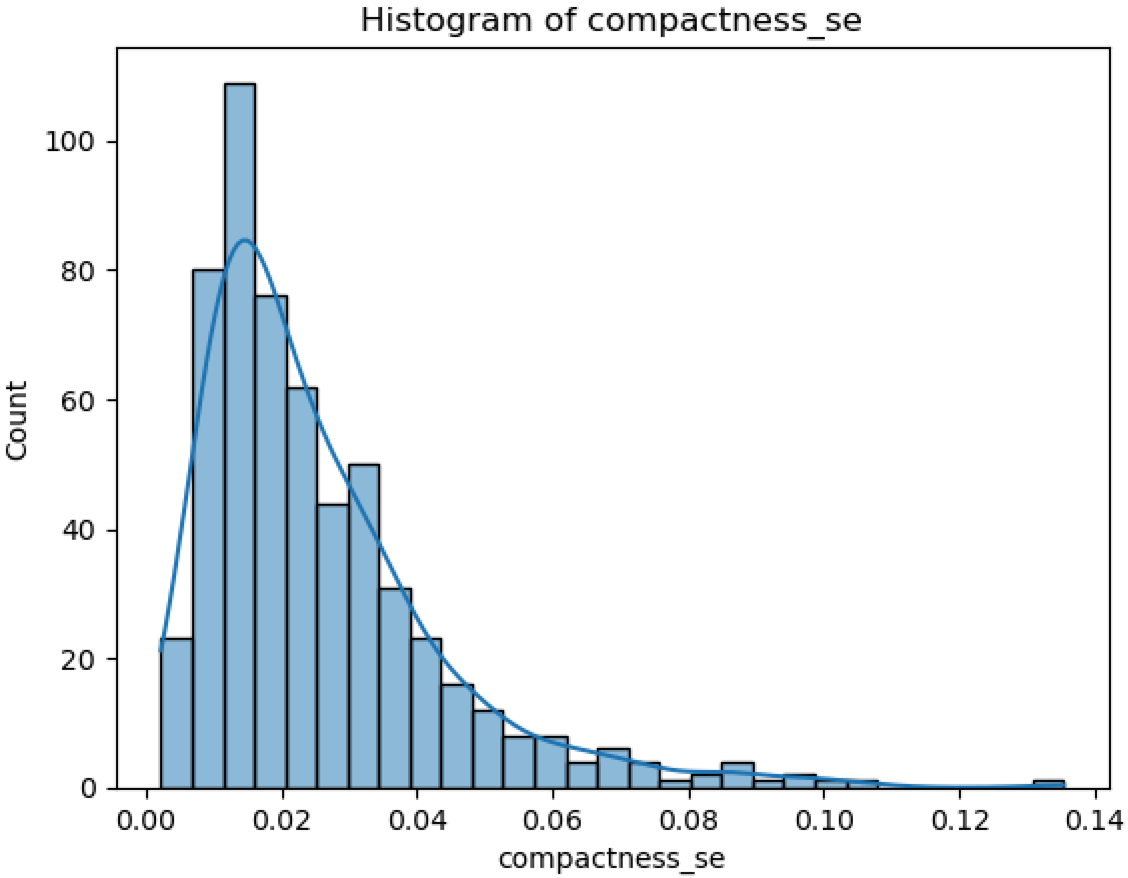


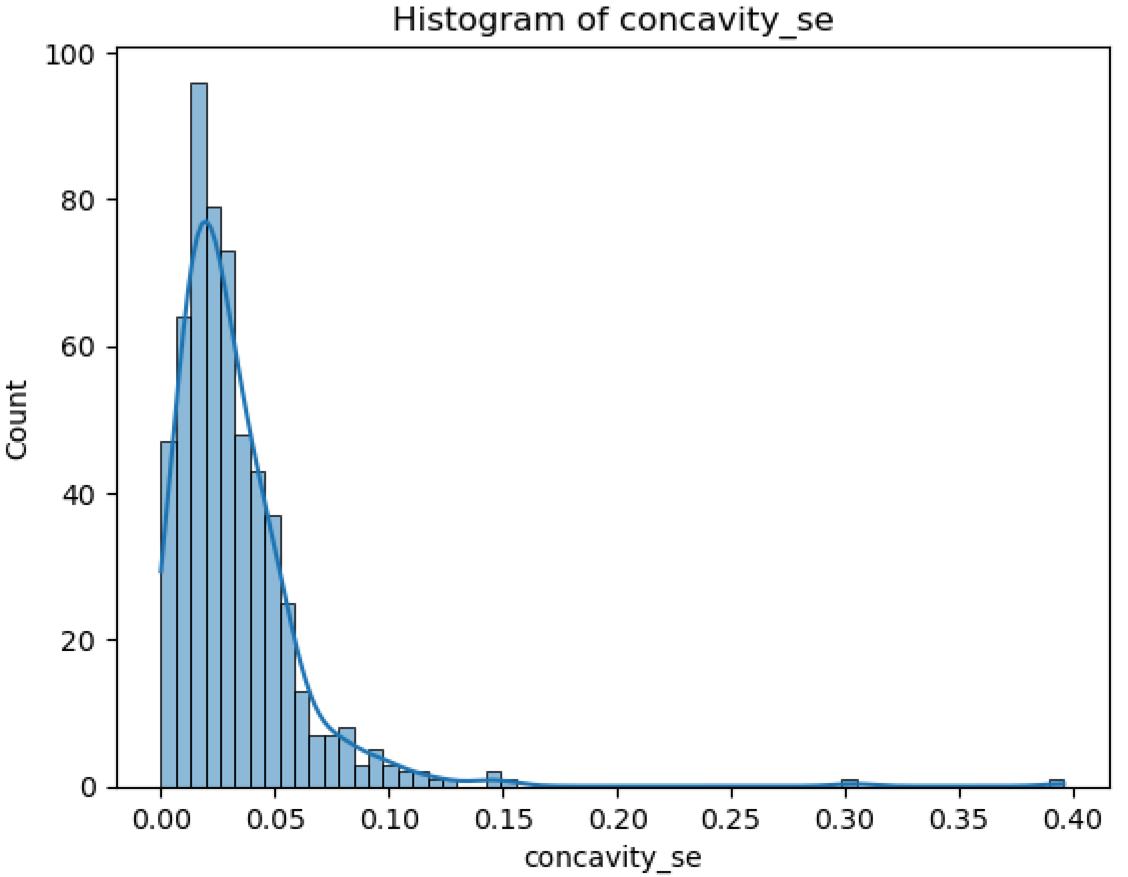
Histogram of perimeter_se

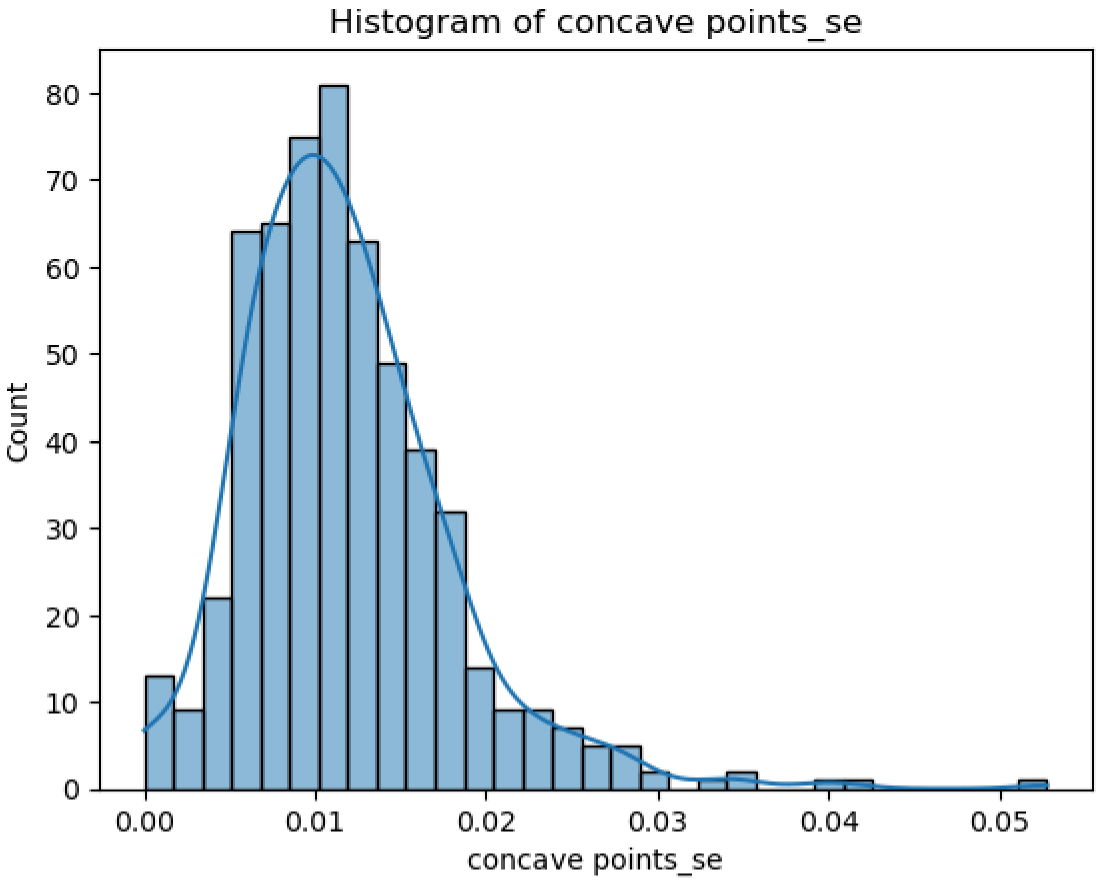


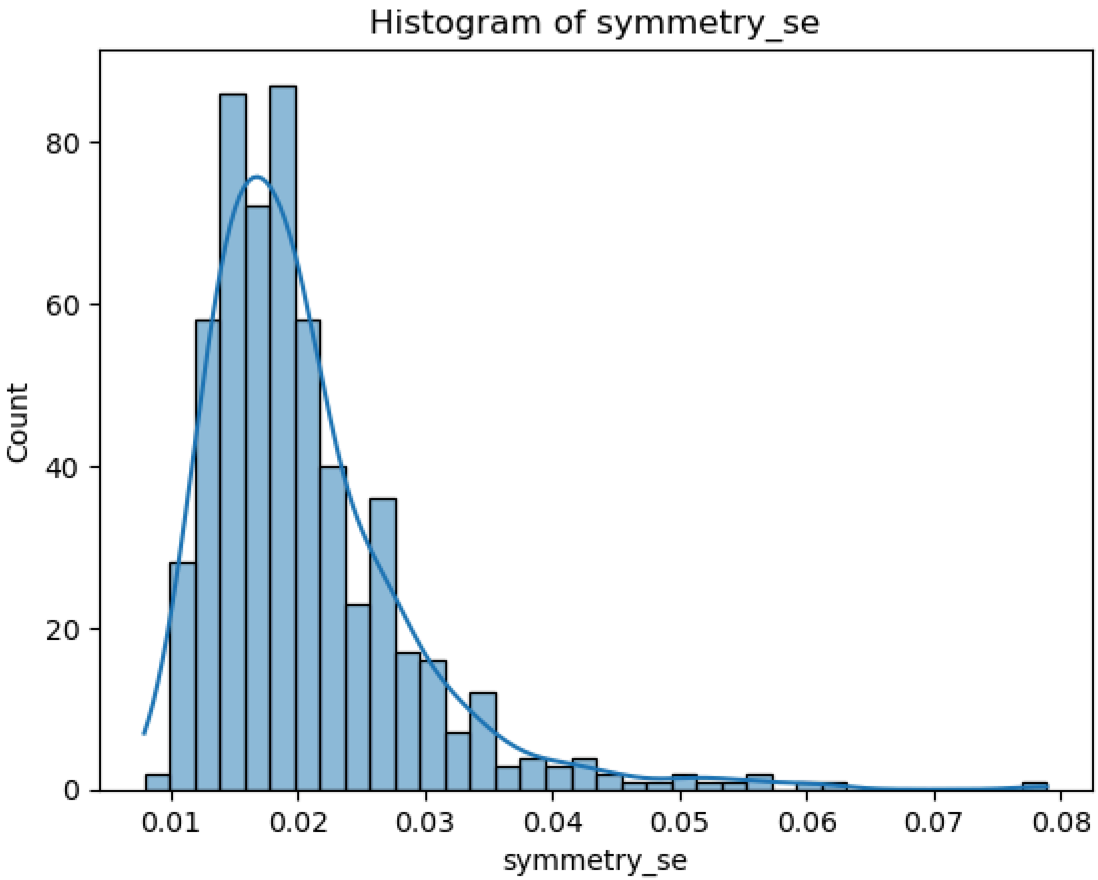


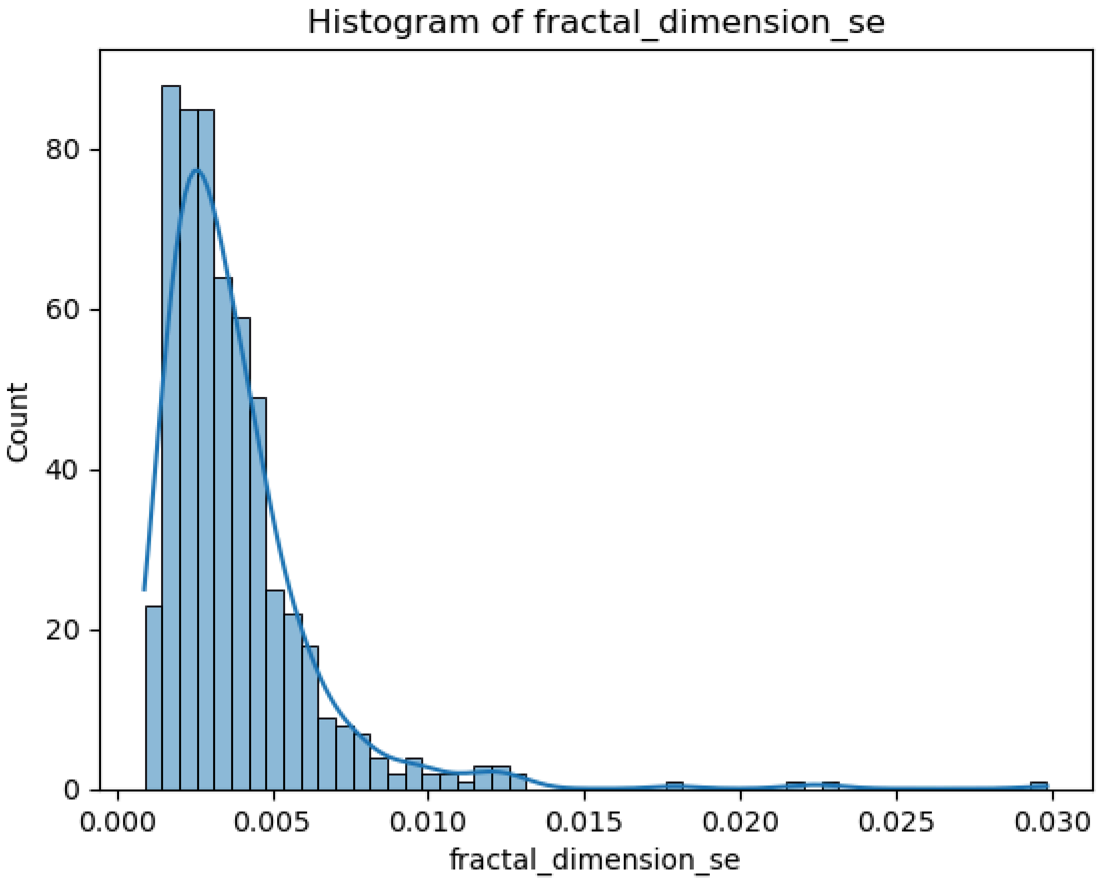
Loading [MathJax]/extensions/Safe.js

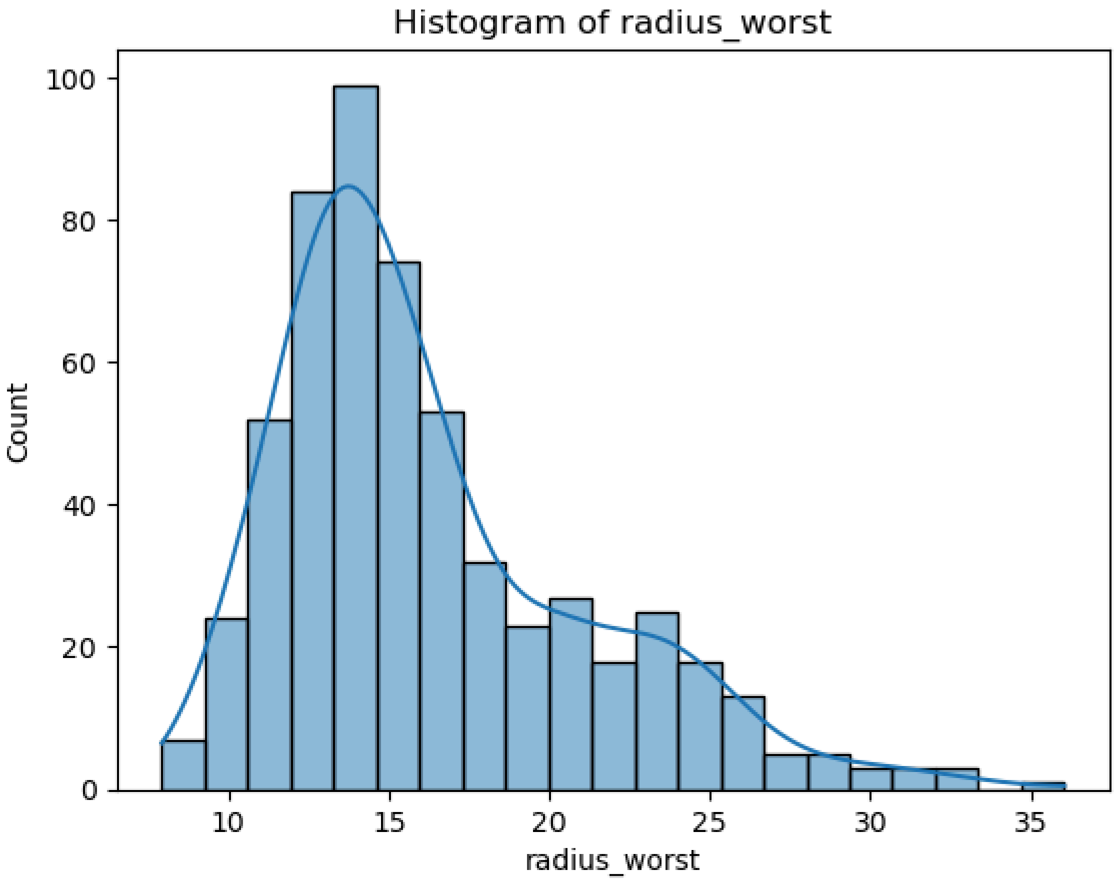

Histogram of compactness_se
Count
0
20
40
60
80
100
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
compactness_se

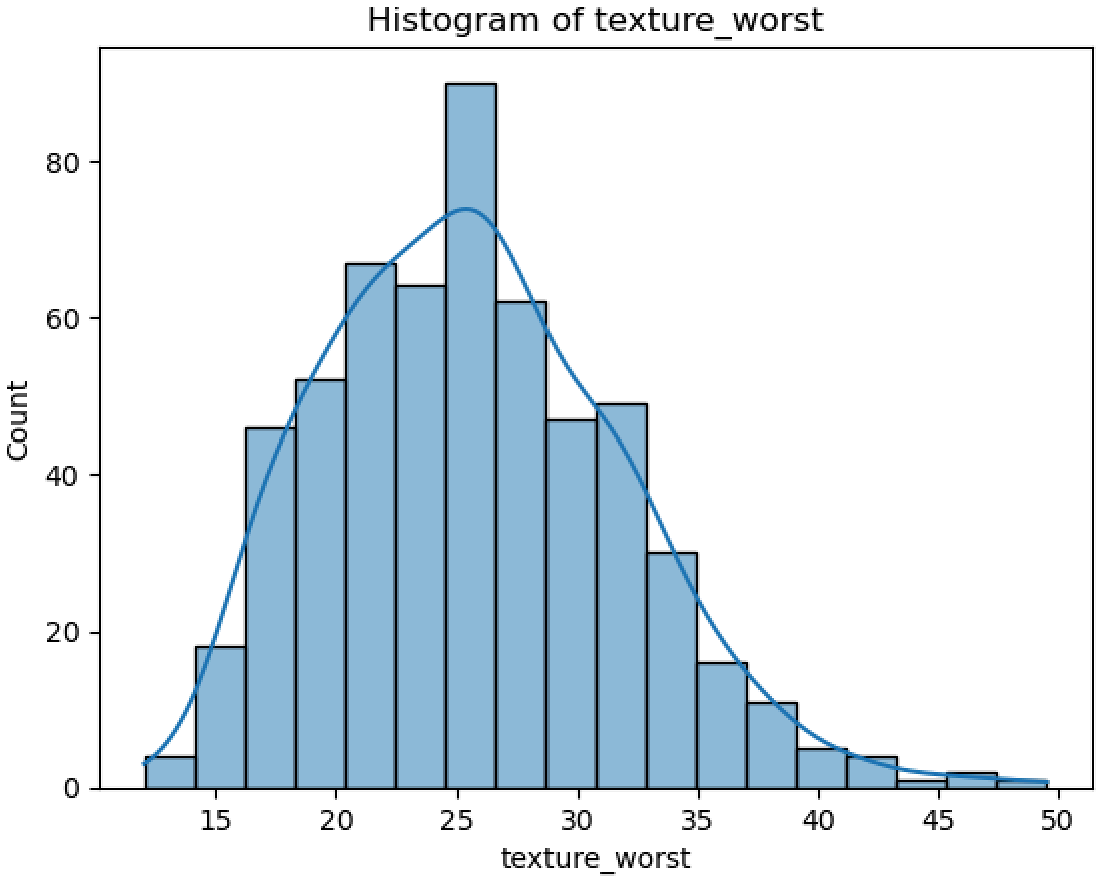

Histogram of concavity_se
Count
0
20
40
60
80
100
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
concavity_se

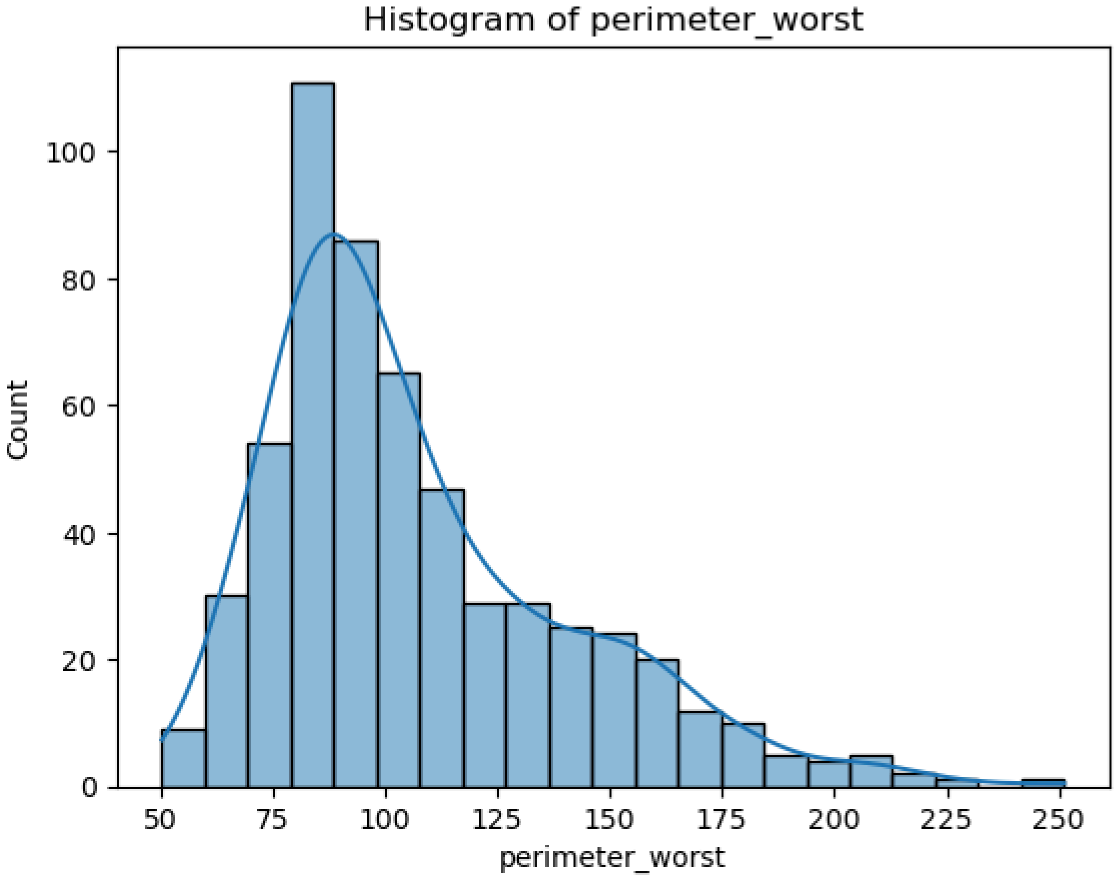

Histogram of concave points_se
Count
0
10
20
30
40
50
60
70
80
0.00
0.01
0.02
0.03
0.04
0.05
concave points_se


Histogram of symmetry_se
Count
0
20
40
60
80
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
symmetry_se

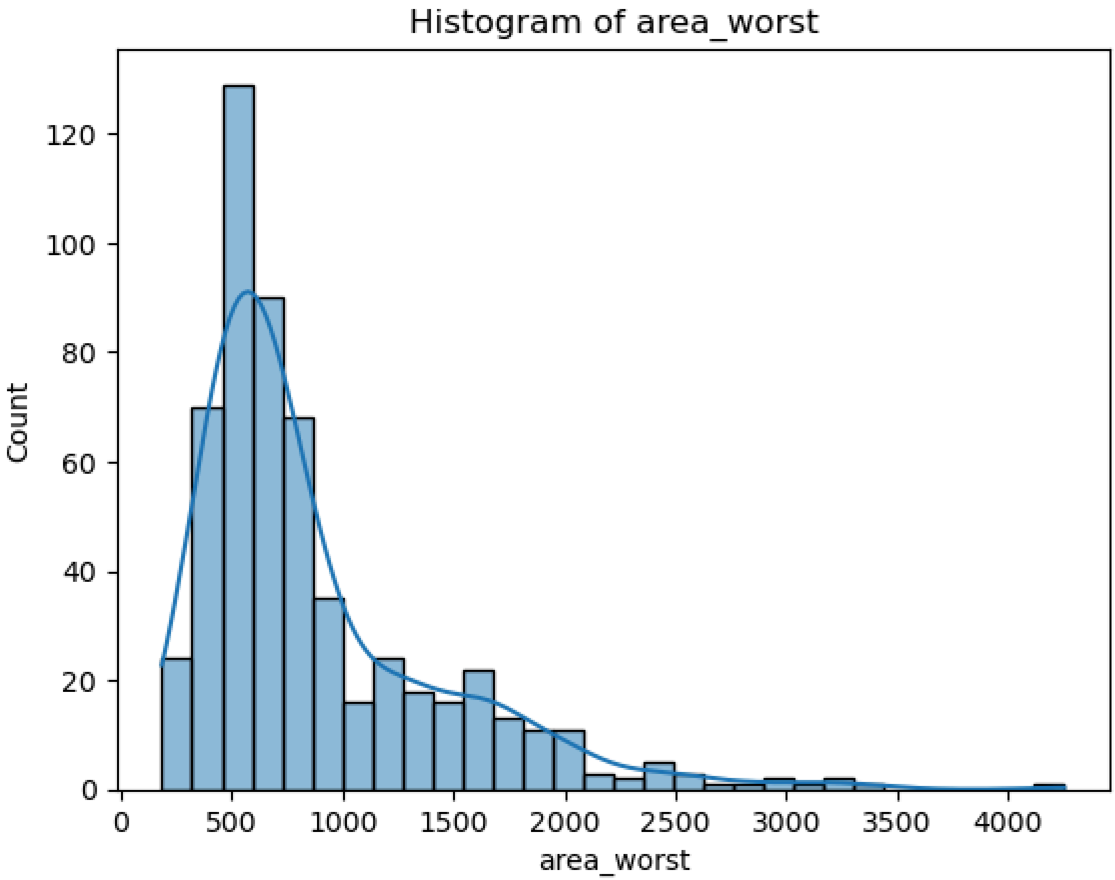

Histogram of fractal_dimension_se
Count
0
20
40
60
80
0.000
0.005
0.010
0.015
0.020
0.025
0.030
fractal_dimension_se

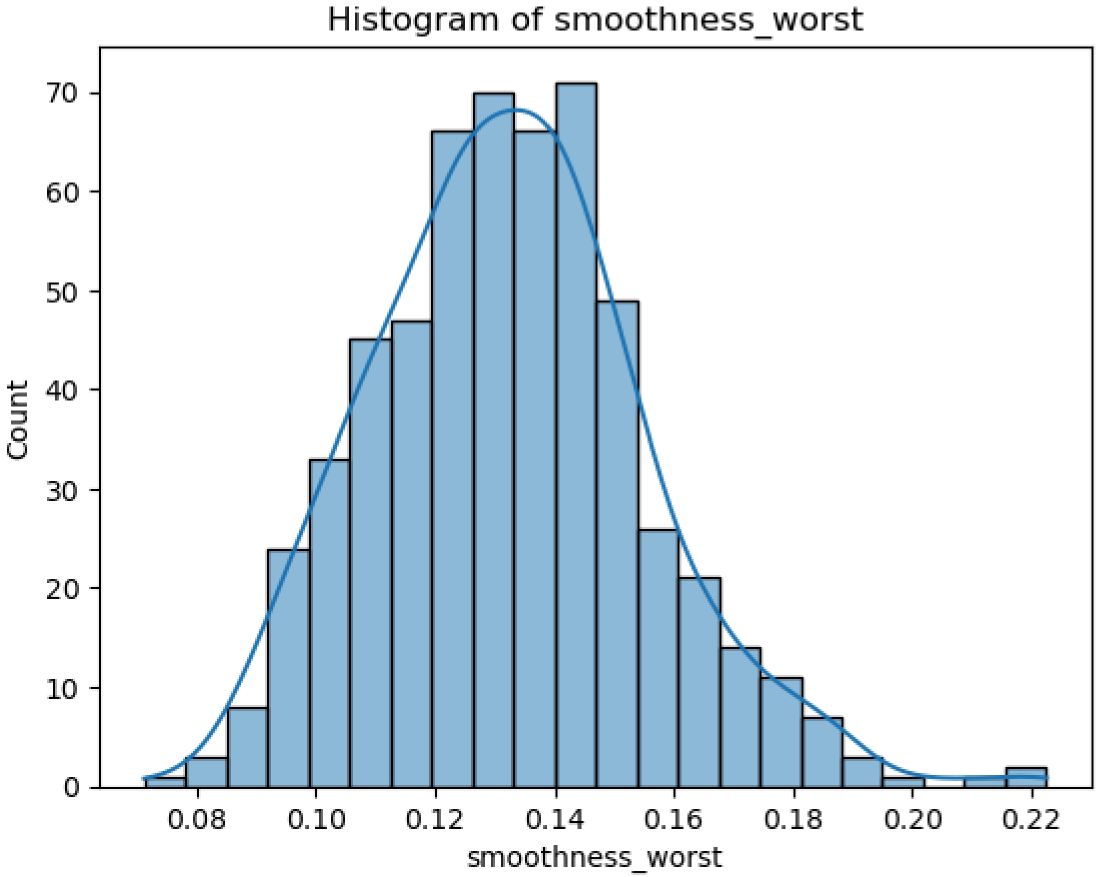

Histogram of radius_worst
Count
0
20
40
60
80
100
10
15
20
25
30
35
radius_worst



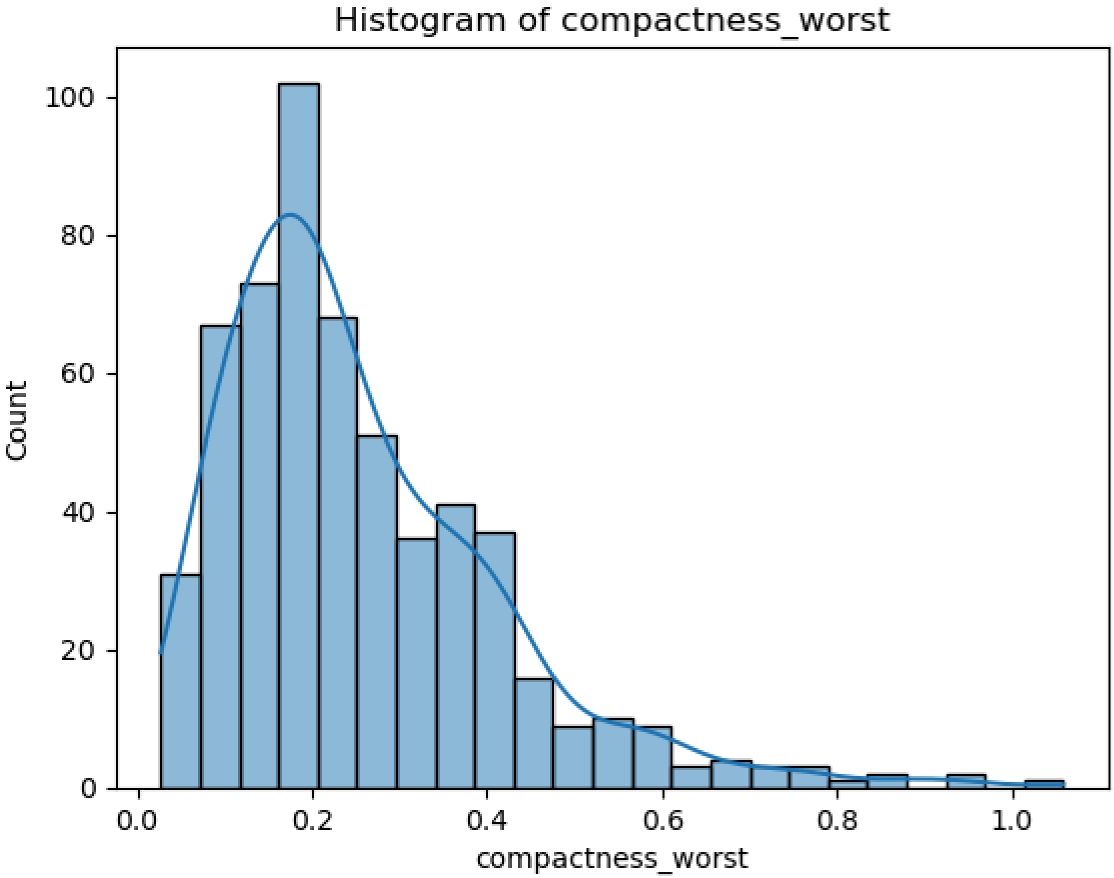


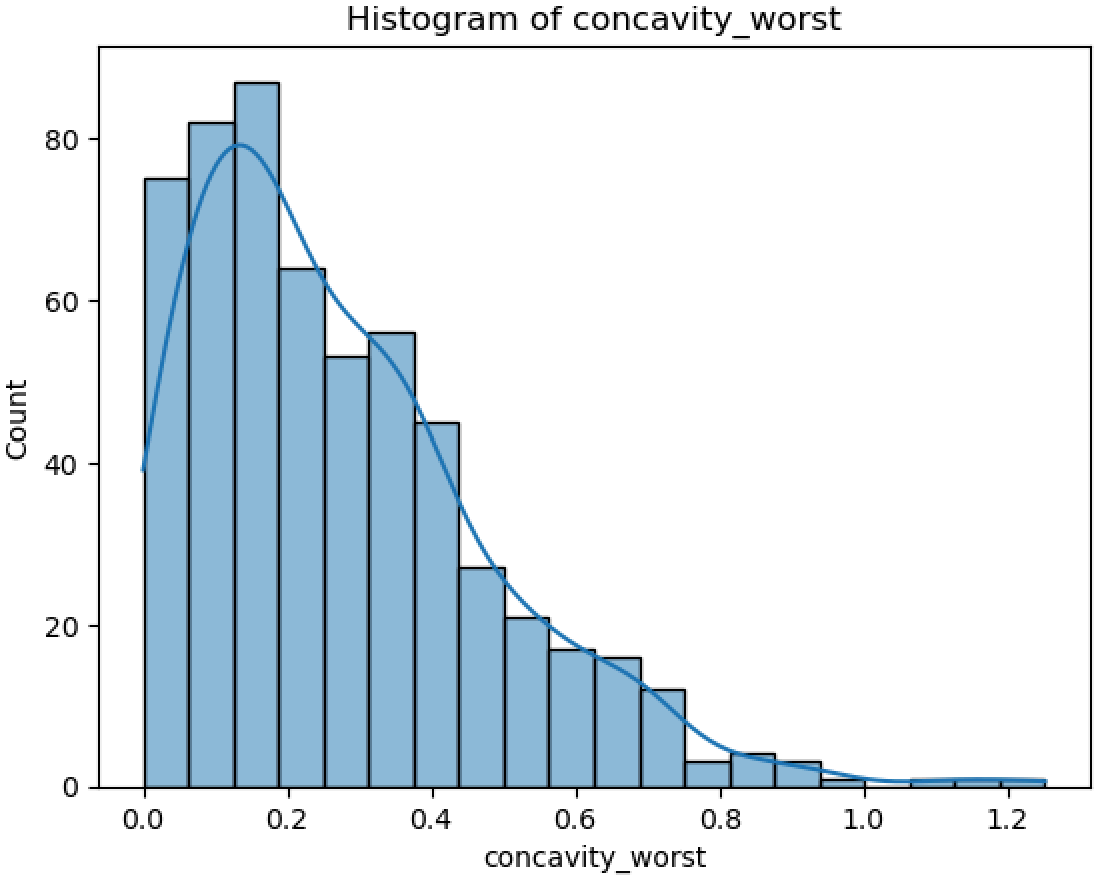
Loading [MathJax]/extensions/Safe.js

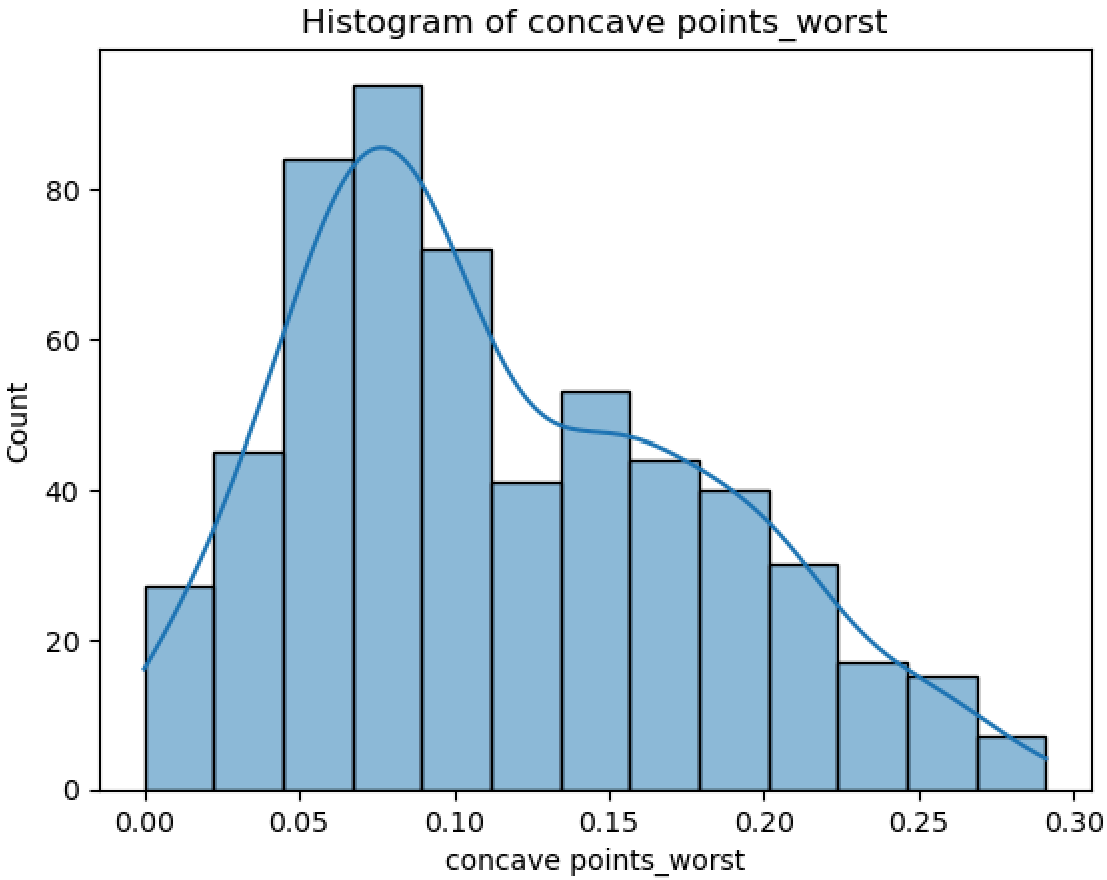




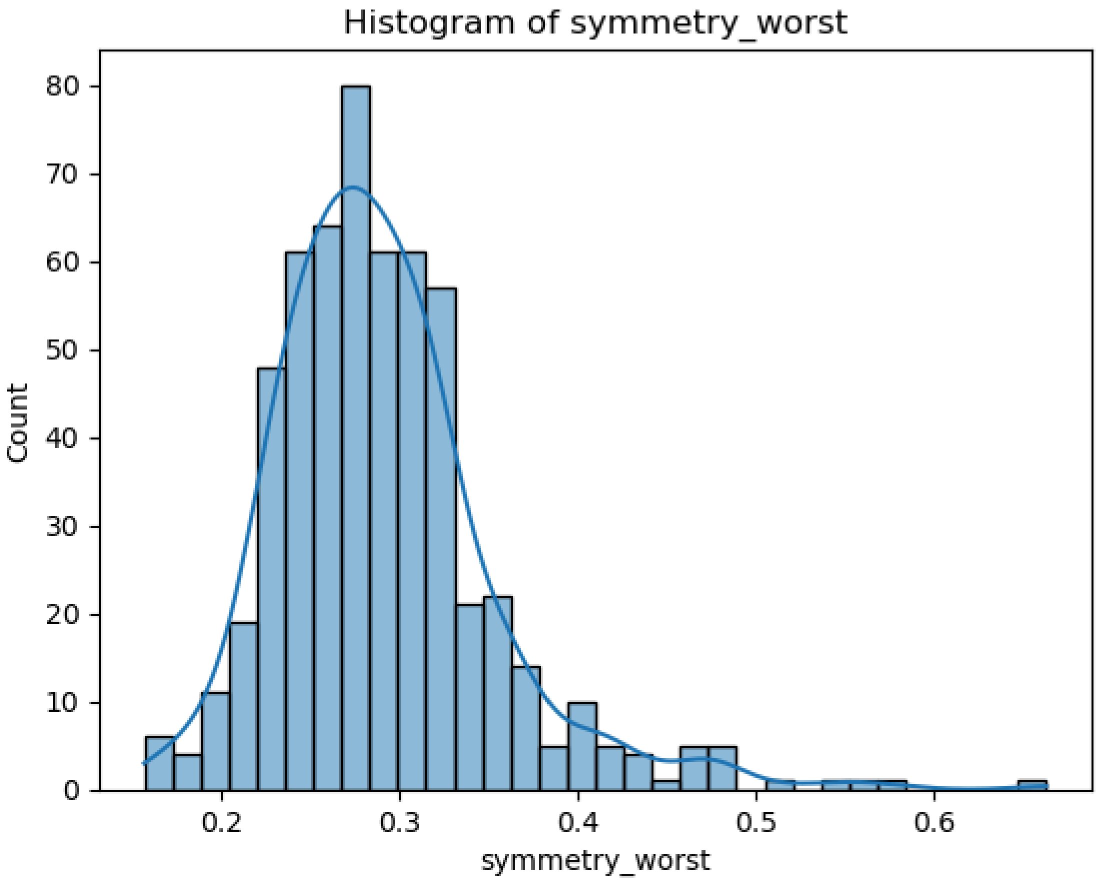
Loading [MathJax]/extensions/Safe.js


Histogram of compactness_worst
Count
0
20
40
60
80
100
0.0
0.2
0.4
0.6
0.8
1.0
compactness_worst


Histogram of concavity_worst
Count
0
20
40
60
80
0.0
0.2
0.4
0.6
0.8
1.0
1.2
concavity_worst

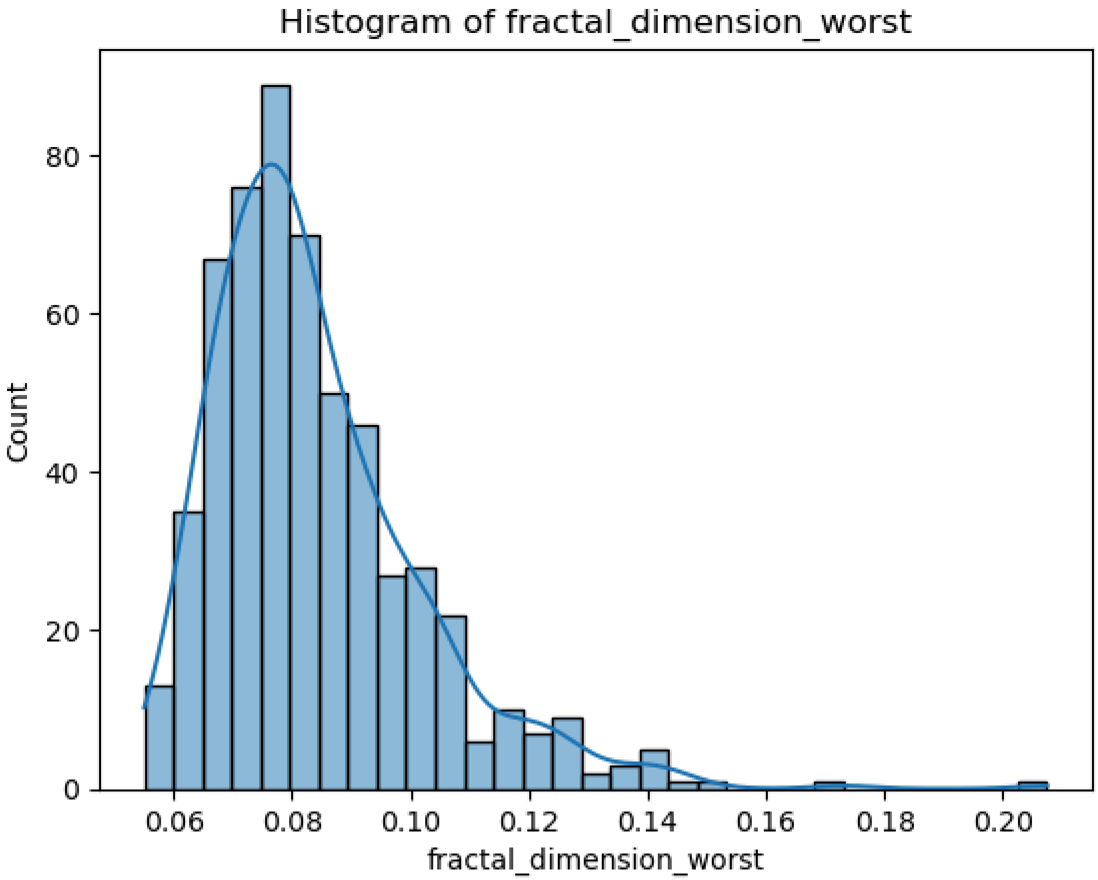


Histogram of concave points_worst



Histogram of symmetry_worst

Loading [MathJax]/extensions/Safe.js



```
In [43]: df.count()
```

Out[43]:

```
diagnosis                  569
radius_mean                569
texture_mean               569
perimeter_mean             569
area_mean                  569
smoothness_mean            569
compactness_mean           569
concavity_mean             569
concave points_mean        569
symmetry_mean              569
fractal_dimension_mean     569
radius_se                  569
texture_se                 569
perimeter_se               569
area_se                    569
smoothness_se              569
compactness_se             569
concavity_se               569
concave points_se          569
symmetry_se                569
fractal_dimension_se       569
radius_worst               569
texture_worst              569
perimeter_worst            569
area_worst                 569
smoothness_worst           569
compactness_worst          569
concavity_worst            569
concave points_worst       569
symmetry_worst             569
fractal_dimension_worst    569
dtype: int64
```

In [ ]:

Loading [MathJax]/extensions/Safe.js