breast cancer

March 19, 2023

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

[2]:
```
df=pd.read_csv('cancer.csv')
```

Attribute Information:

1) ID number
2) Diagnosis (M = malignant, B = benign) 3-32)

Ten real-valued features are computed for each cell nucleus:

a) radius (mean of distances from center to points on the perimeter)
b) texture (standard deviation of gray-scale values)
c) perimeter
d) area
e) smoothness (local variation in radius lengths)
f) compactness (perimeter^2 / area - 1.0)
g) concavity (severity of concave portions of the contour)
h) concave points (number of concave portions of the contour)
i) symmetry
j) fractal dimension ("coastline approximation" - 1)

[3]:
```
df.head()
```

[3]:

	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]
```

[4]:
```
df.columns
```

[4]:
```
Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       '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')
```

[5]:
```
df.drop(columns='Unnamed: 32', inplace=True, axis=1)
```

[6]:
```
df.head()
```

[6]:
```
         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

   …  radius_worst  texture_worst  perimeter_worst  area_worst  \
0  …         25.38          17.33           184.60      2019.0
1  …         24.99          23.41           158.80      1956.0
2  …         23.57          25.53           152.50      1709.0
3  …         14.91          26.50            98.87       567.7
4  …         22.54          16.67           152.20      1575.0

   smoothness_worst  compactness_worst  concavity_worst  concave points_worst  \
0            0.1622             0.6656           0.7119                0.2654
1            0.1238             0.1866           0.2416                0.1860
2            0.1444             0.4245           0.4504                0.2430
3            0.2098             0.8663           0.6869                0.2575
4            0.1374             0.2050           0.4000                0.1625

   symmetry_worst  fractal_dimension_worst
0          0.4601                  0.11890
1          0.2750                  0.08902
2          0.3613                  0.08758
3          0.6638                  0.17300
4          0.2364                  0.07678

[5 rows x 32 columns]
```

[7]:
```
df.shape
```

[7]:
```
(569, 32)
```

[8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 32 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
dtypes: float64(30), int64(1), object(1)
memory usage: 142.4+ KB
```

[9]:
```
df.drop(columns= 'id', inplace=True, axis=1)
```

[10]:
```
df.head()
```

[10]:

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	\
0	M	17.99	10.38	122.80	1001.0	
1	M	20.57	17.77	132.90	1326.0	
2	M	19.69	21.25	130.00	1203.0	
3	M	11.42	20.38	77.58	386.1	
4	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

   symmetry_mean  …  radius_worst  texture_worst  perimeter_worst  \
0         0.2419  …         25.38          17.33           184.60
1         0.1812  …         24.99          23.41           158.80
2         0.2069  …         23.57          25.53           152.50
3         0.2597  …         14.91          26.50            98.87
4         0.1809  …         22.54          16.67           152.20

   area_worst  smoothness_worst  compactness_worst  concavity_worst  \
0      2019.0            0.1622             0.6656           0.7119
1      1956.0            0.1238             0.1866           0.2416
2      1709.0            0.1444             0.4245           0.4504
3       567.7            0.2098             0.8663           0.6869
4      1575.0            0.1374             0.2050           0.4000

   concave points_worst  symmetry_worst  fractal_dimension_worst
0                0.2654          0.4601                  0.11890
1                0.1860          0.2750                  0.08902
2                0.2430          0.3613                  0.08758
3                0.2575          0.6638                  0.17300
4                0.1625          0.2364                  0.07678

[5 rows x 31 columns]
```

[11]:
```
df.isnull().sum()
```

[11]:
```
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
dtype: int64
```

[12]:
```
df.describe()
```

[12]:
```
       radius_mean  texture_mean  perimeter_mean    area_mean  \
count   569.000000    569.000000     569.000000   569.000000
mean     14.127292     19.289649      91.969033   654.889104
std       3.524049      4.301036      24.298981   351.914129
min       6.981000      9.710000      43.790000   143.500000
25%      11.700000     16.170000      75.170000   420.300000
50%      13.370000     18.840000      86.240000   551.100000
75%      15.780000     21.800000     104.100000   782.700000
max      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  fractal_dimension_mean  …  radius_worst  \
count     569.000000              569.000000  …    569.000000
mean        0.181162                0.062798  …     16.269190
std         0.027414                0.007060  …      4.833242
min         0.106000                0.049960  …      7.930000
25%         0.161900                0.057700  …     13.010000
50%         0.179200                0.061540  …     14.970000
75%         0.195700                0.066120  …     18.790000
max         0.304000                0.097440  …     36.040000

       texture_worst  perimeter_worst   area_worst  smoothness_worst  \
count     569.000000       569.000000   569.000000        569.000000
```

```
mean       25.677223       107.261213   880.583128          0.132369
std         6.146258        33.602542   569.356993          0.022832
min        12.020000        50.410000   185.200000          0.071170
25%        21.080000        84.110000   515.300000          0.116600
50%        25.410000        97.660000   686.500000          0.131300
75%        29.720000       125.400000  1084.000000          0.146000
max        49.540000       251.200000  4254.000000          0.222600

       compactness_worst  concavity_worst  concave points_worst  \
count         569.000000       569.000000            569.000000
mean            0.254265         0.272188              0.114606
std             0.157336         0.208624              0.065732
min             0.027290         0.000000              0.000000
25%             0.147200         0.114500              0.064930
50%             0.211900         0.226700              0.099930
75%             0.339100         0.382900              0.161400
max             1.058000         1.252000              0.291000

       symmetry_worst  fractal_dimension_worst
count      569.000000               569.000000
mean         0.290076                 0.083946
std          0.061867                 0.018061
min          0.156500                 0.055040
25%          0.250400                 0.071460
50%          0.282200                 0.080040
75%          0.317900                 0.092080
max          0.663800                 0.207500

[8 rows x 30 columns]
```

as we see median < mean so it is positive skewed data

[13]:
```
df['diagnosis'].value_counts()
```

[13]:
```
B    357
M    212
Name: diagnosis, dtype: int64
```

[14]:
```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
labels= le.fit_transform(df['diagnosis'])
df['Target']= labels
df.drop(columns='diagnosis', inplace= True, axis=0)
```

[15]:
```
df.head()
```

[15]:
```
   radius_mean  texture_mean  perimeter_mean  area_mean  smoothness_mean  \
0        17.99         10.38          122.80     1001.0          0.11840
```

```
1        20.57         17.77           132.90     1326.0          0.08474
2        19.69         21.25           130.00     1203.0          0.10960
3        11.42         20.38            77.58      386.1          0.14250
4        20.29         14.34           135.10     1297.0          0.10030

   compactness_mean  concavity_mean  concave points_mean  symmetry_mean  \
0           0.27760          0.3001              0.14710         0.2419
1           0.07864          0.0869              0.07017         0.1812
2           0.15990          0.1974              0.12790         0.2069
3           0.28390          0.2414              0.10520         0.2597
4           0.13280          0.1980              0.10430         0.1809

   fractal_dimension_mean  …  texture_worst  perimeter_worst  area_worst  \
0                 0.07871  …          17.33           184.60      2019.0
1                 0.05667  …          23.41           158.80      1956.0
2                 0.05999  …          25.53           152.50      1709.0
3                 0.09744  …          26.50            98.87       567.7
4                 0.05883  …          16.67           152.20      1575.0

   smoothness_worst  compactness_worst  concavity_worst  concave points_worst  \
0            0.1622             0.6656           0.7119                0.2654
1            0.1238             0.1866           0.2416                0.1860
2            0.1444             0.4245           0.4504                0.2430
3            0.2098             0.8663           0.6869                0.2575
4            0.1374             0.2050           0.4000                0.1625

   symmetry_worst  fractal_dimension_worst  Target
0          0.4601                  0.11890       1
1          0.2750                  0.08902       1
2          0.3613                  0.08758       1
3          0.6638                  0.17300       1
4          0.2364                  0.07678       1

[5 rows x 31 columns]
```

[16]:
```
df['Target'].value_counts()
```

[16]:
```
0    357
1    212
Name: Target, dtype: int64
```

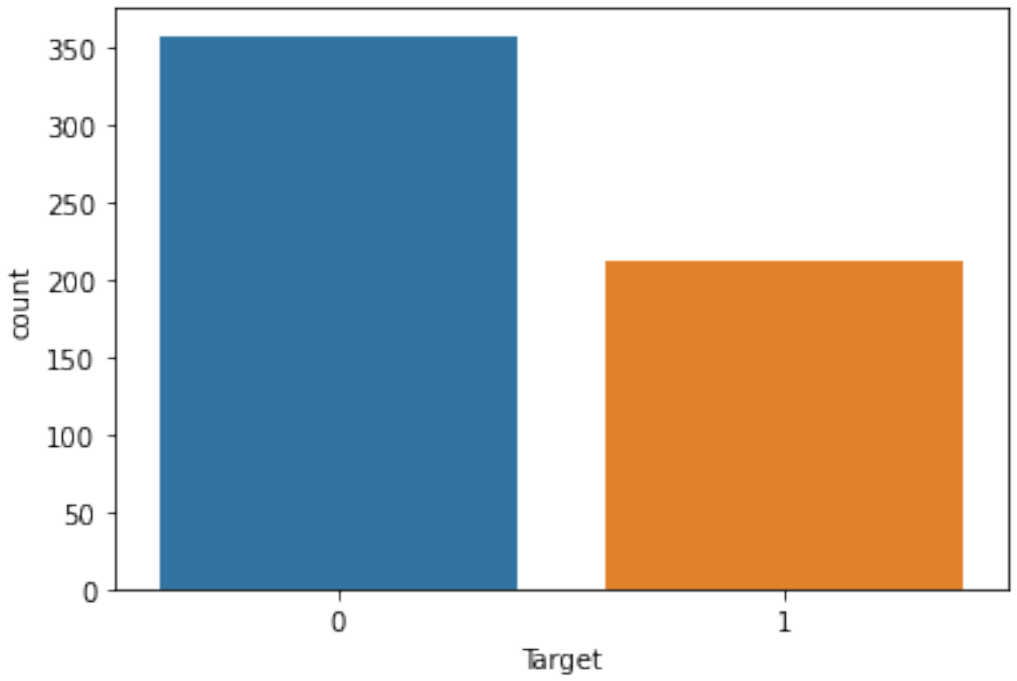
[17]:
```
df.groupby('Target').mean()
```

[17]:
```
        radius_mean  texture_mean  perimeter_mean   area_mean  \
Target
0         12.146524     17.914762       78.075406  462.790196
1         17.462830     21.604906      115.365377  978.376415
```

```
        smoothness_mean  compactness_mean  concavity_mean  \
Target
0              0.092478          0.080085        0.046058
1              0.102898          0.145188        0.160775

        concave points_mean  symmetry_mean  fractal_dimension_mean  …  \
Target                                                               …
0                  0.025717       0.174186                0.062867  …
1                  0.087990       0.192909                0.062680  …

        radius_worst  texture_worst  perimeter_worst   area_worst  \
Target
0          13.379801      23.515070        87.005938   558.899440
1          21.134811      29.318208       141.370330  1422.286321

        smoothness_worst  compactness_worst  concavity_worst  \
Target
0               0.124959           0.182673         0.166238
1               0.144845           0.374824         0.450606

        concave points_worst  symmetry_worst  fractal_dimension_worst
Target
0                   0.074444        0.270246                 0.079442
1                   0.182237        0.323468                 0.091530

[2 rows x 30 columns]
```

[18]:
```
sns.countplot(x='Target', data=df)
```

[18]:
```
<AxesSubplot:xlabel='Target', ylabel='count'>
```

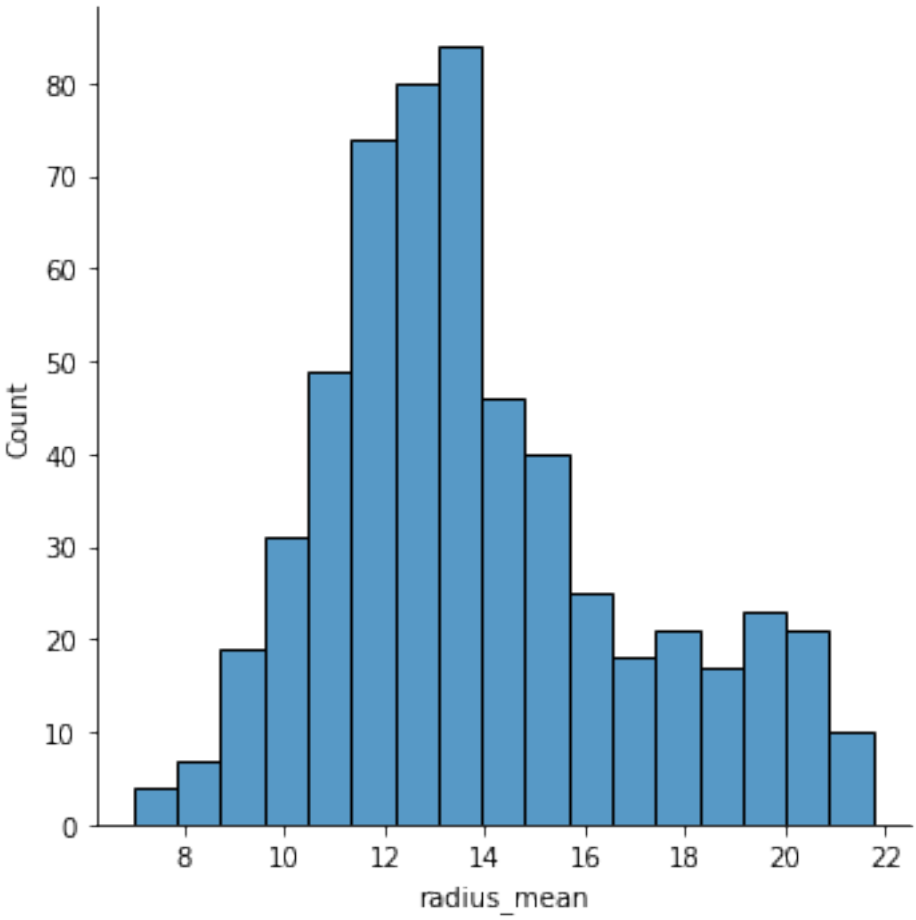


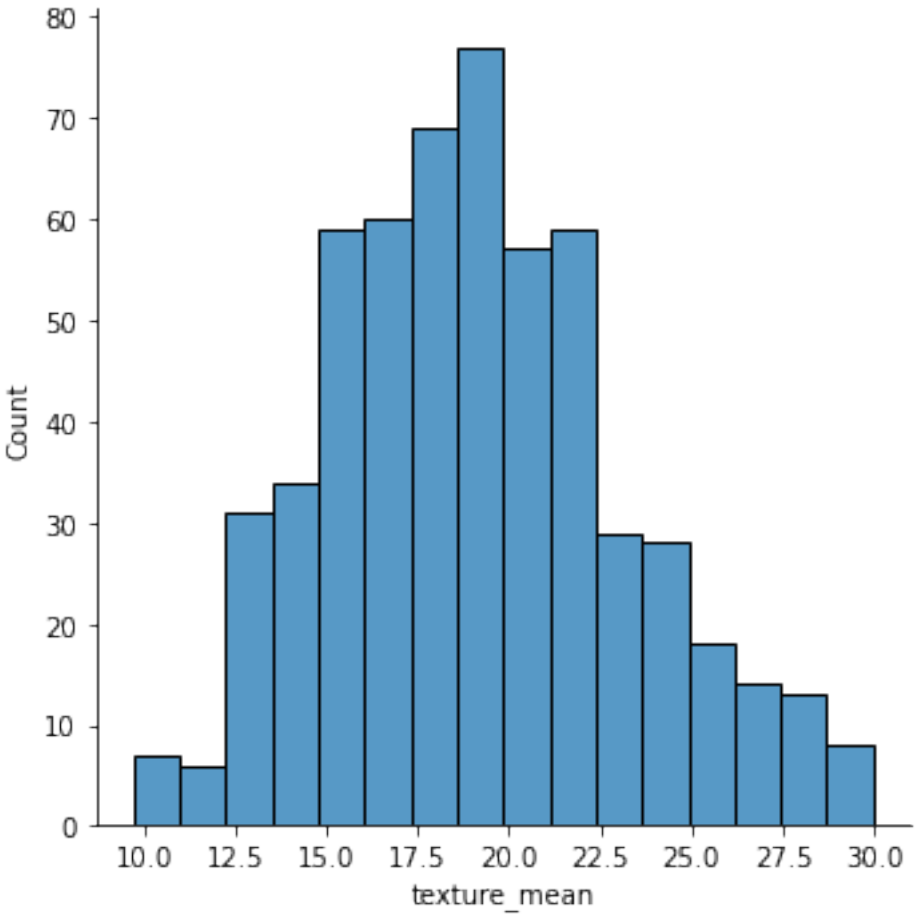
[19]:
```
for column in df :
    print(column)
```

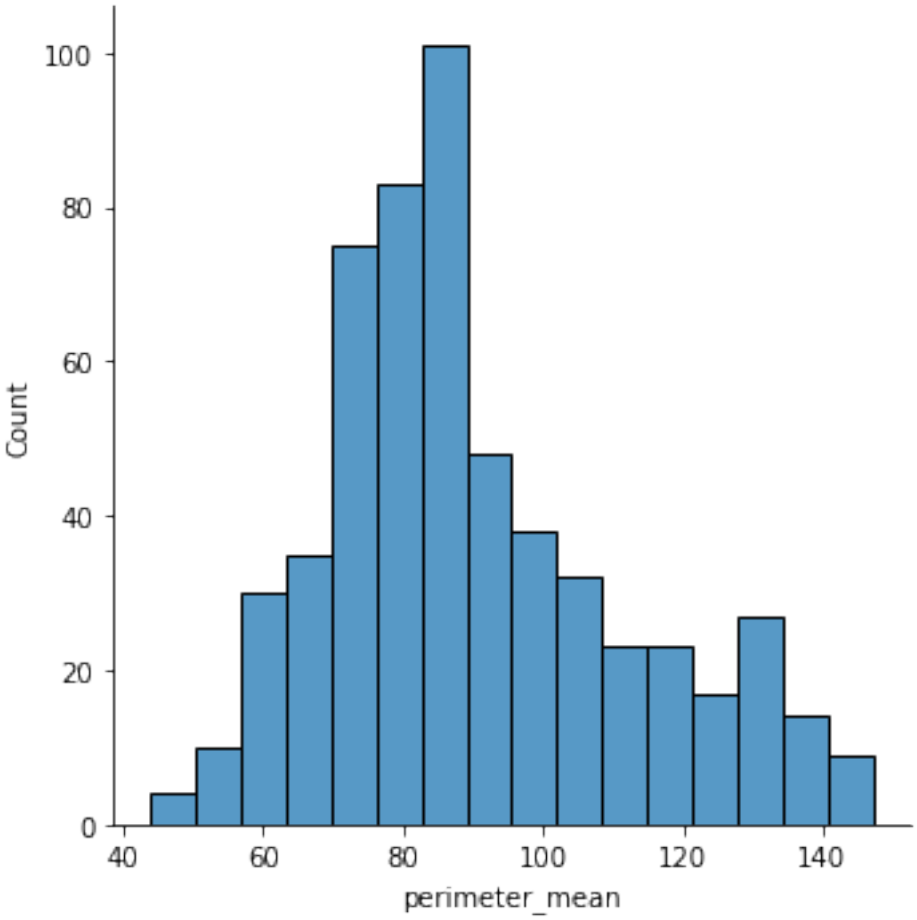
```
radius_mean
texture_mean
perimeter_mean
area_mean
smoothness_mean
compactness_mean
concavity_mean
concave points_mean
symmetry_mean
fractal_dimension_mean
radius_se
texture_se
perimeter_se
area_se
smoothness_se
compactness_se
concavity_se
concave points_se
symmetry_se
fractal_dimension_se
radius_worst
texture_worst
```

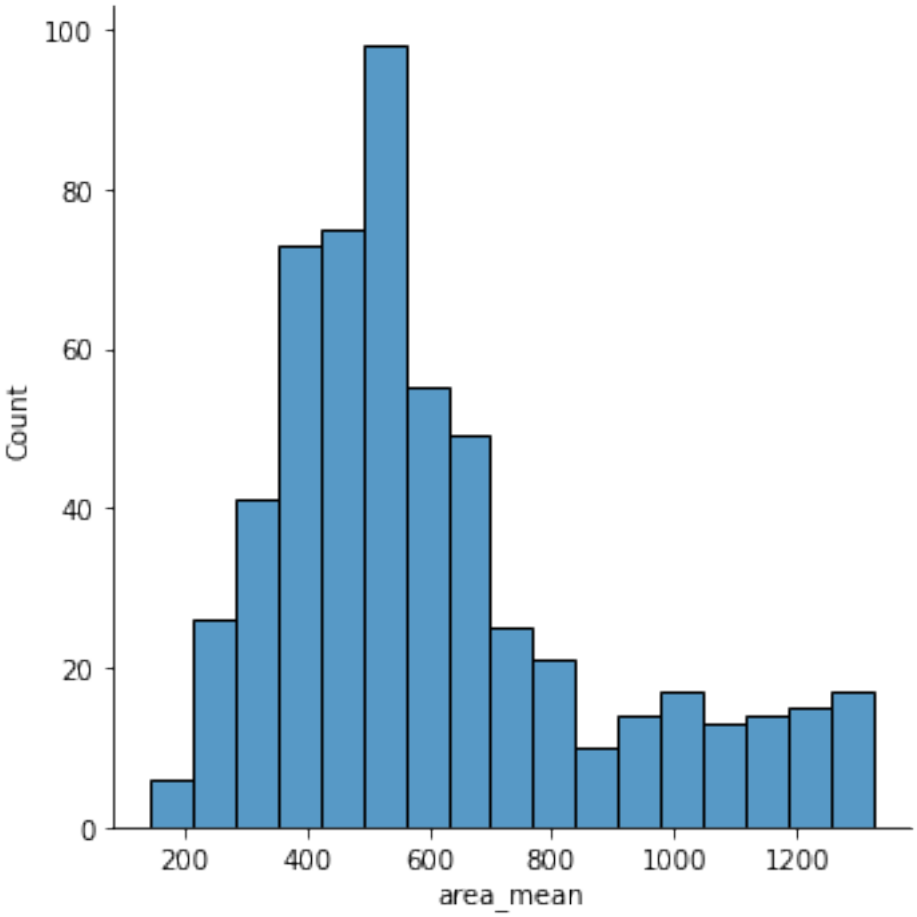
```
perimeter_worst
area_worst
smoothness_worst
compactness_worst
concavity_worst
concave points_worst
symmetry_worst
fractal_dimension_worst
Target
```

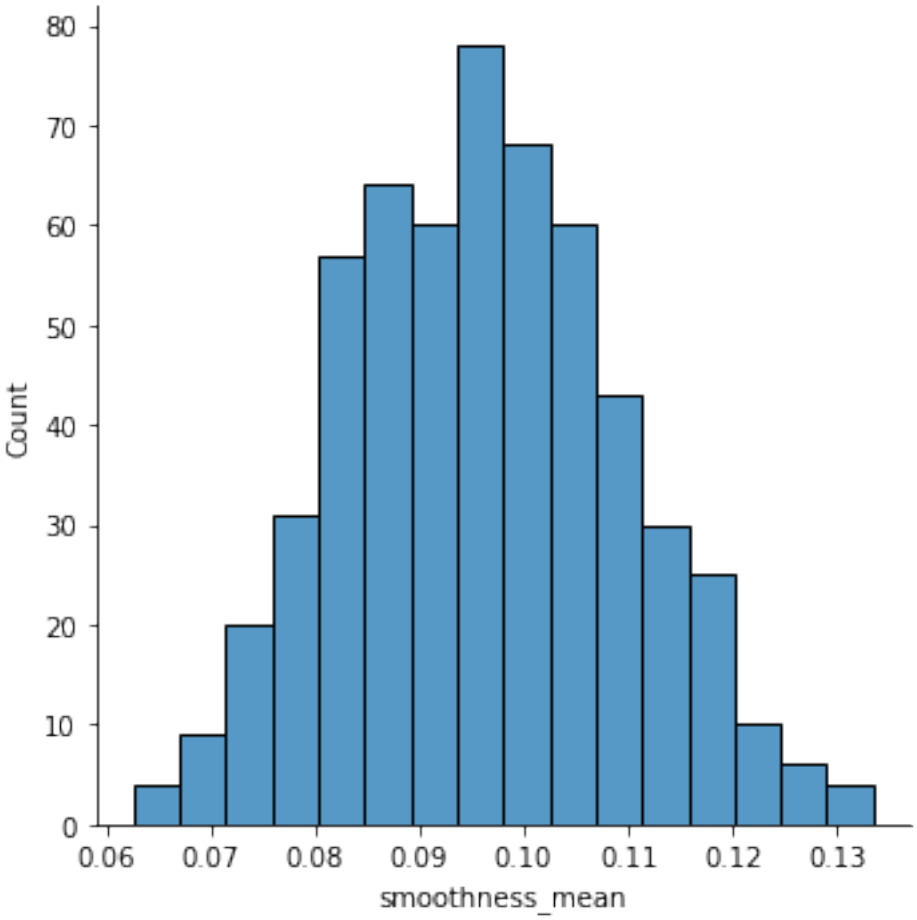
[44]:
```
for column in df:
    sns.displot(x=column, data=df)
```

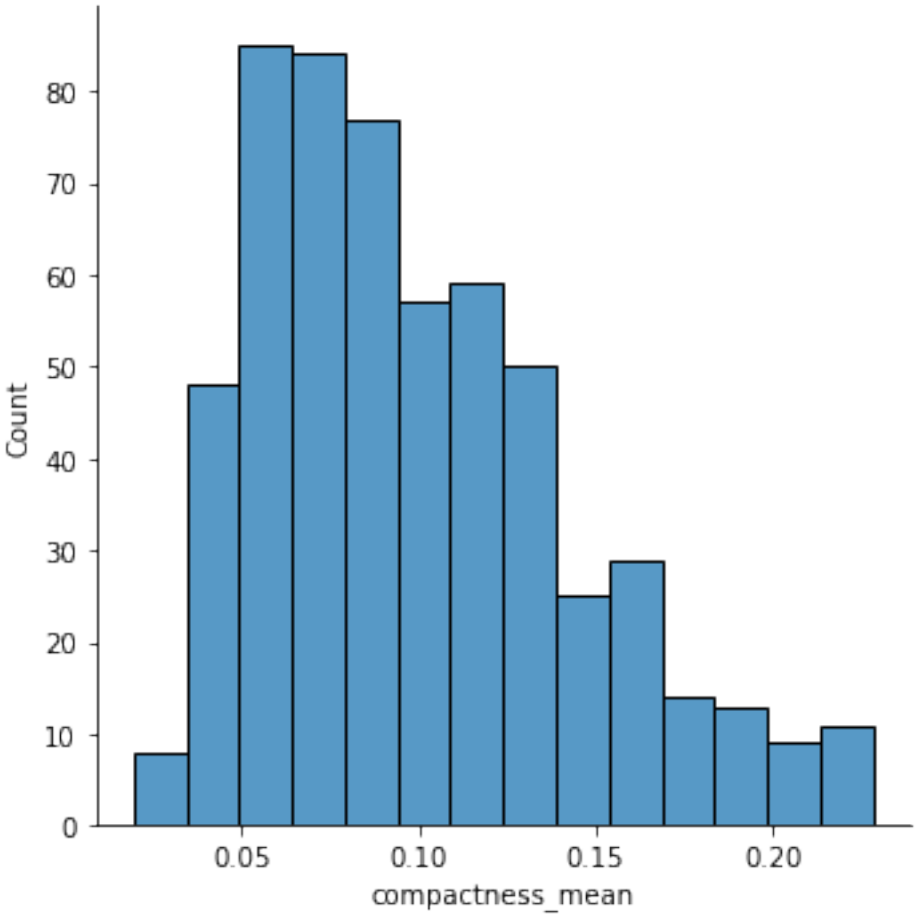


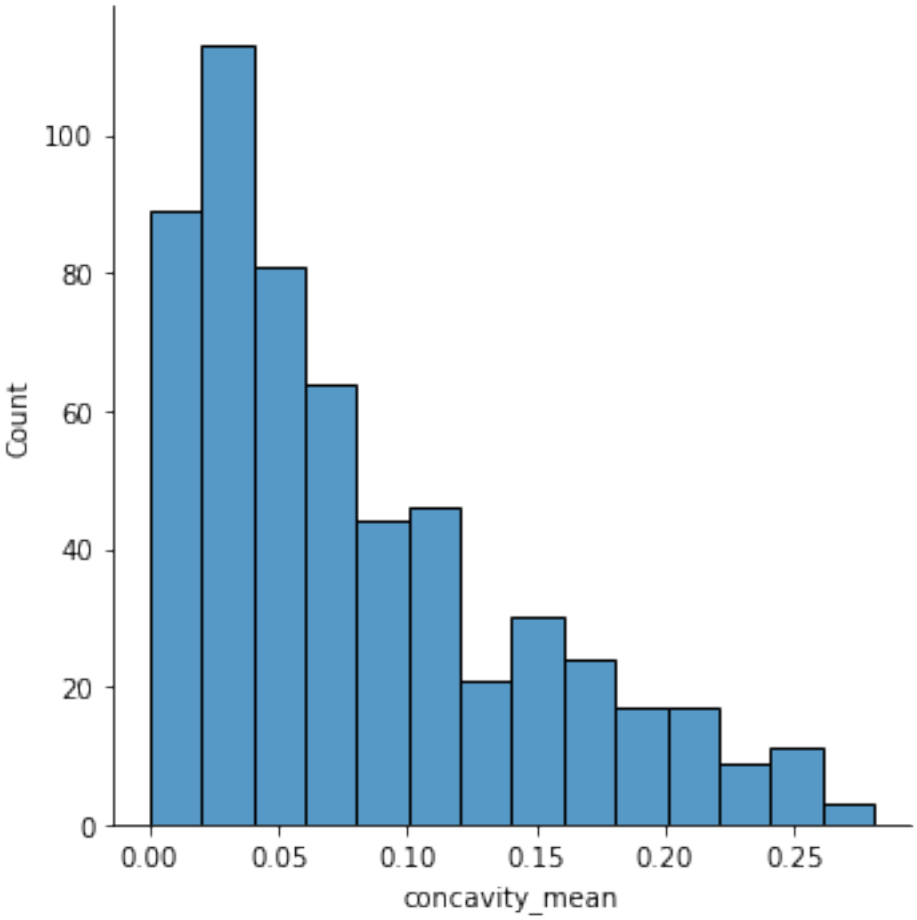

80
70
60
50
40
30
20
10
0
Count
10.0
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0
texture_mean

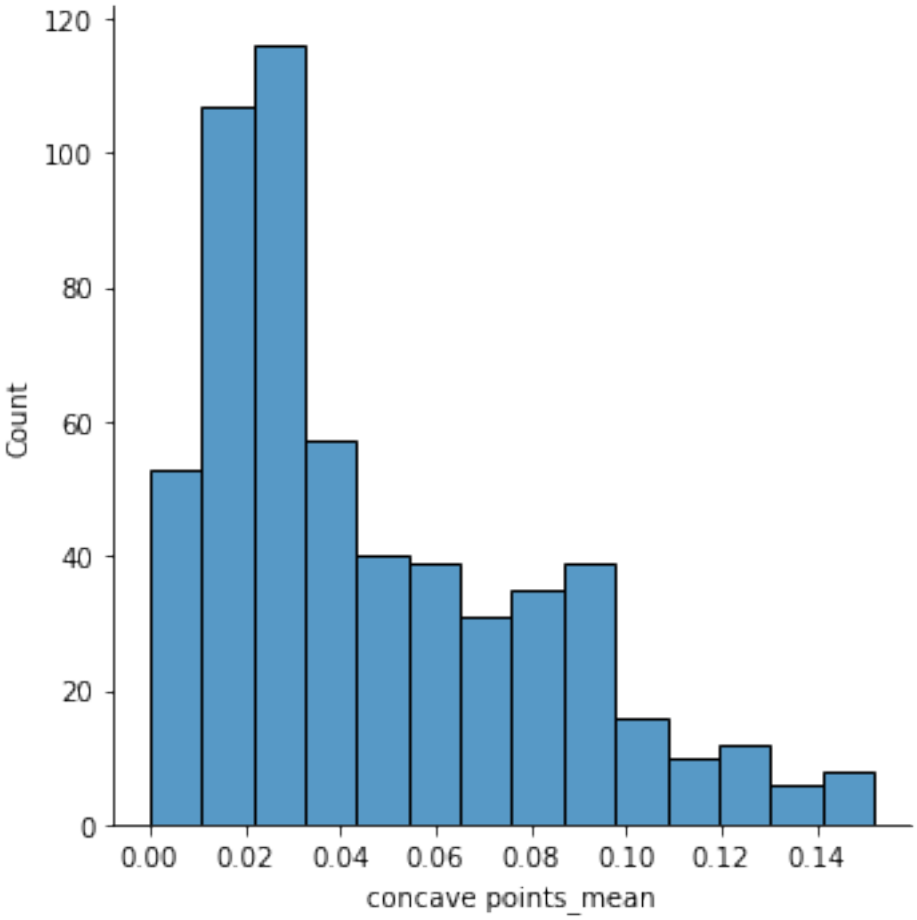

Count
100
80
60
40
20
0
40
60
80
100
120
140
perimeter_mean

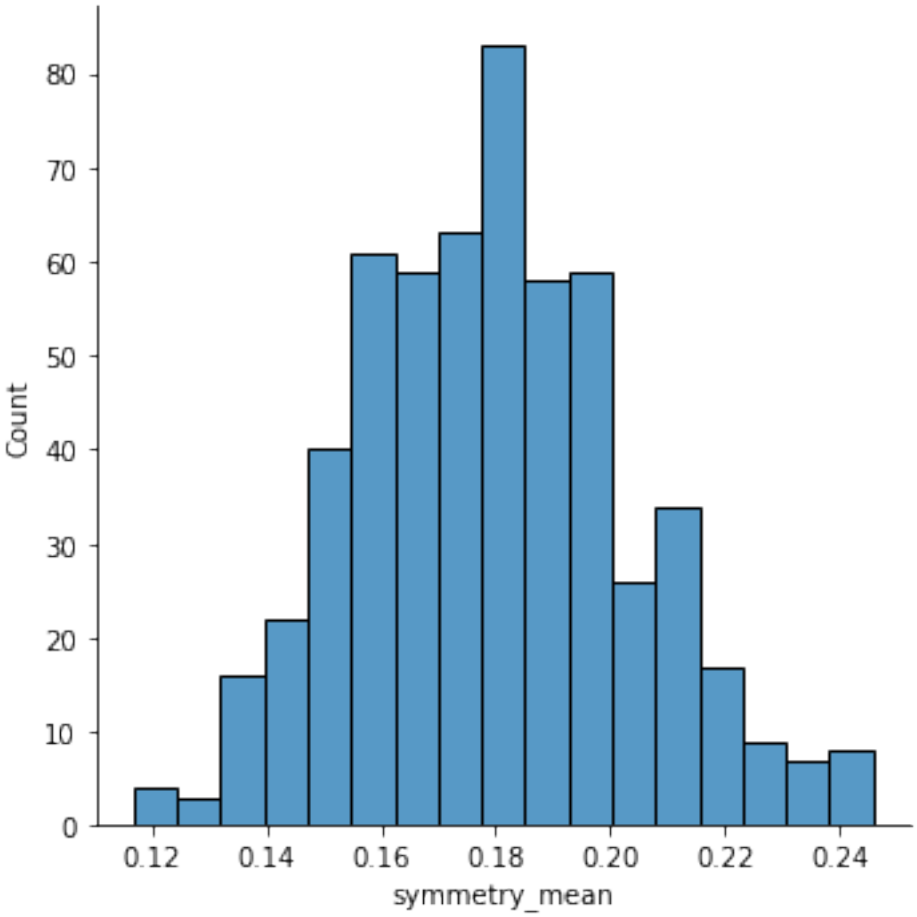

100
80
60
40
20
0
Count
200
400
600
800
1000
1200
area_mean

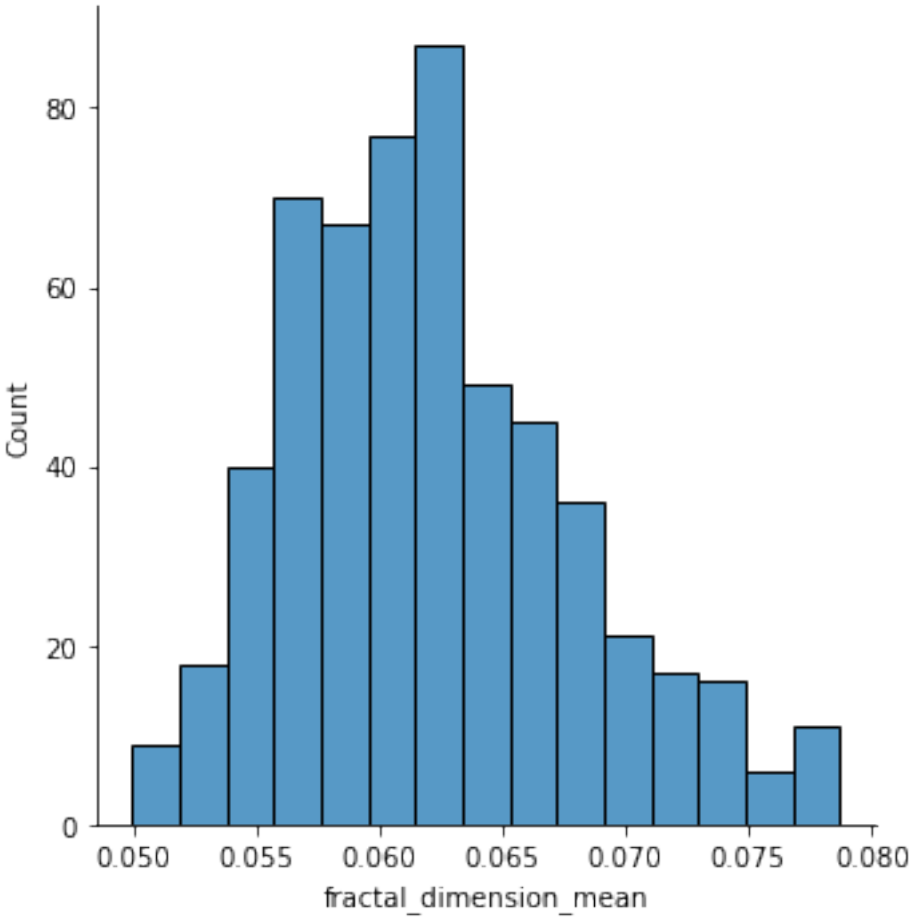

80
70
60
50
40
30
20
10
0
Count
0.06
0.07
0.08
0.09
0.10
0.11
0.12
0.13
smoothness_mean

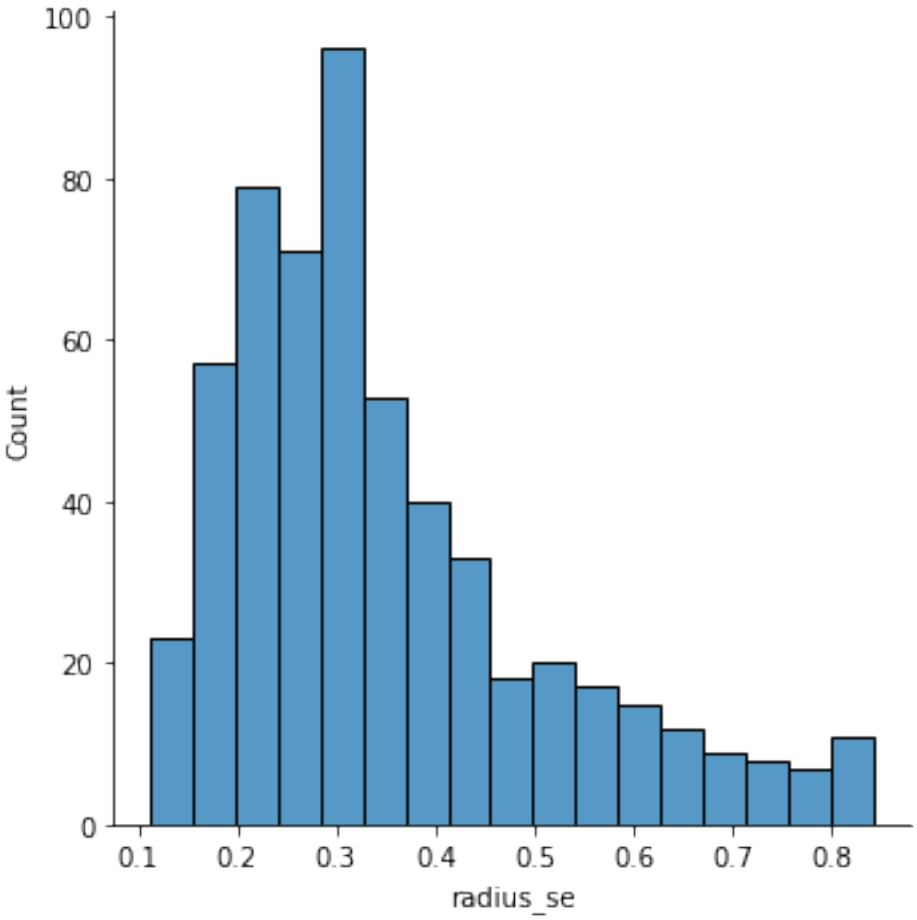

Count
80
70
60
50
40
30
20
10
0
0.05
0.10
0.15
0.20
compactness_mean

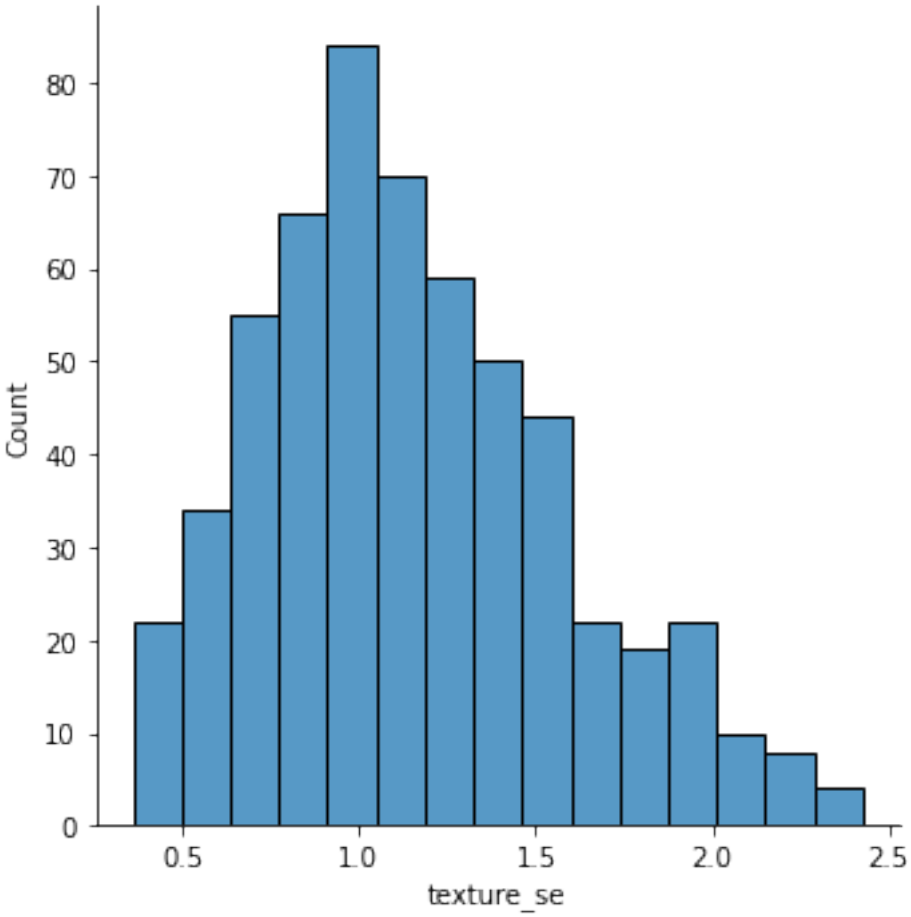

Count
100
80
60
40
20
0
0.00
0.05
0.10
0.15
0.20
0.25
concavity_mean

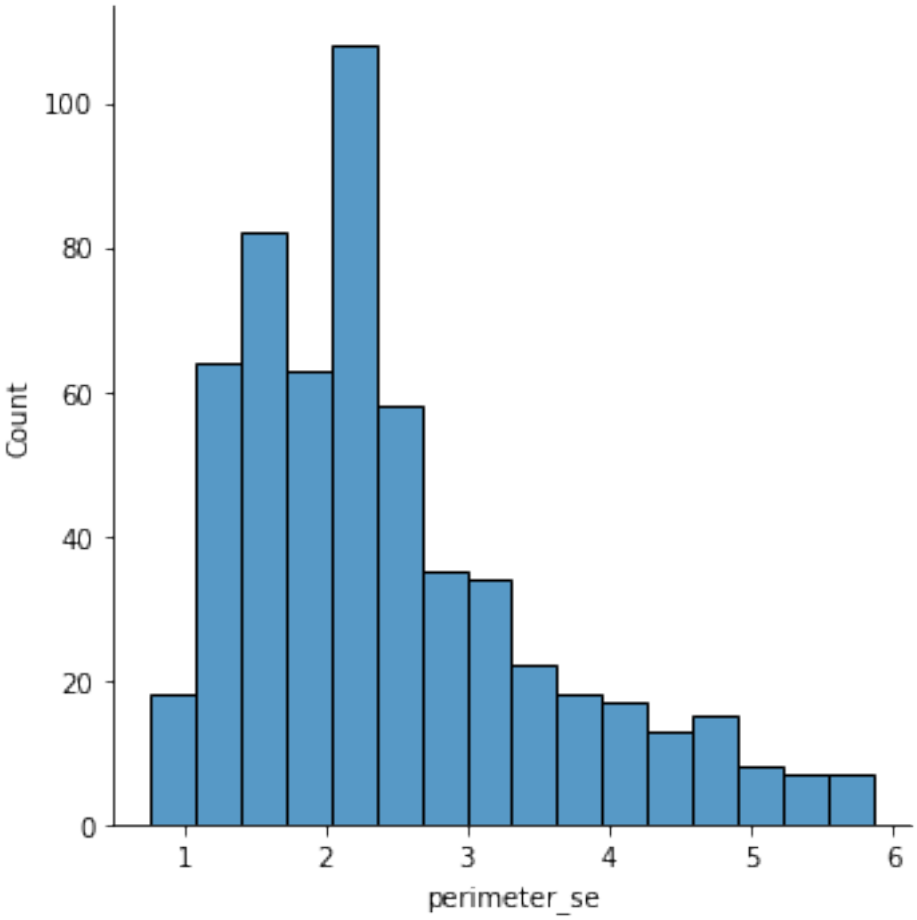

120
100
80
60
40
20
0
Count
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
concave points_mean

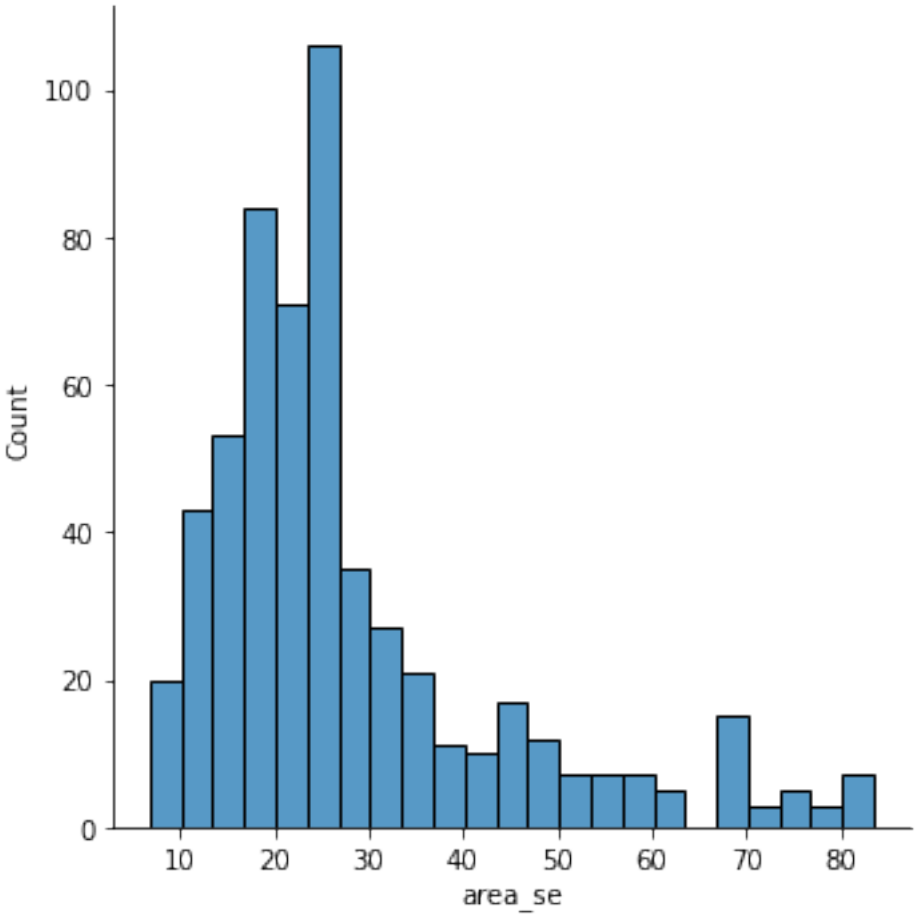

Count
0
10
20
30
40
50
60
70
80
0.12
0.14
0.16
0.18
0.20
0.22
0.24
symmetry_mean

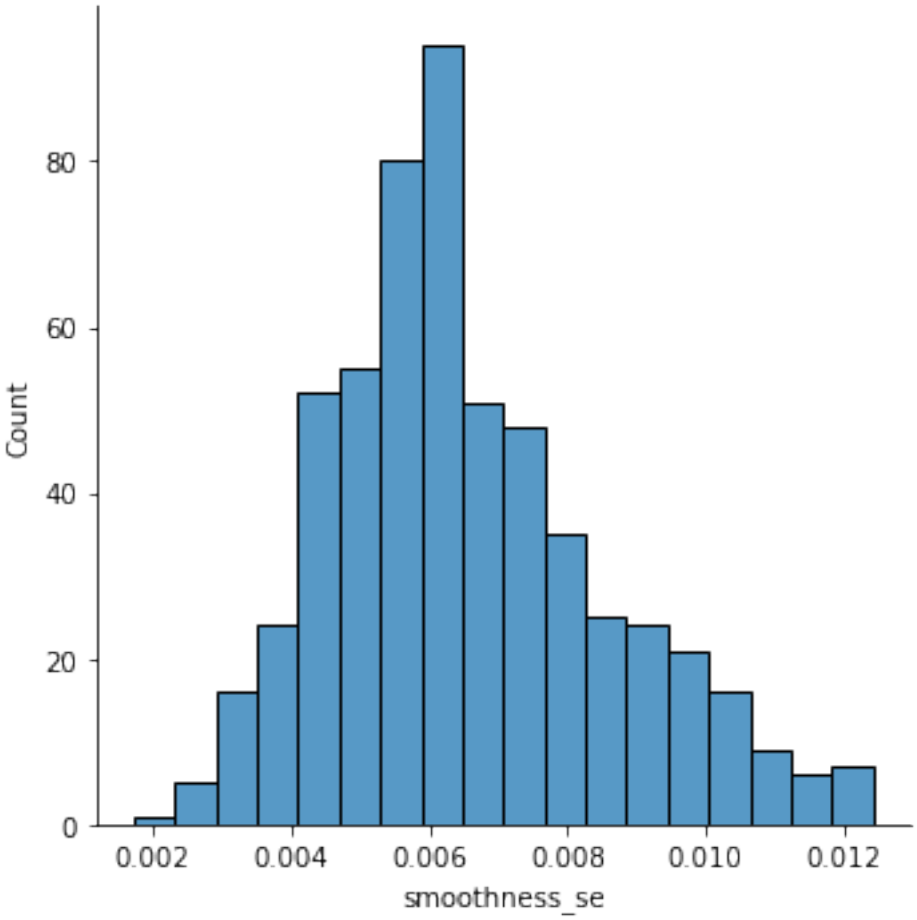

Count
80
60
40
20
0
0.050
0.055
0.060
0.065
0.070
0.075
0.080
fractal_dimension_mean

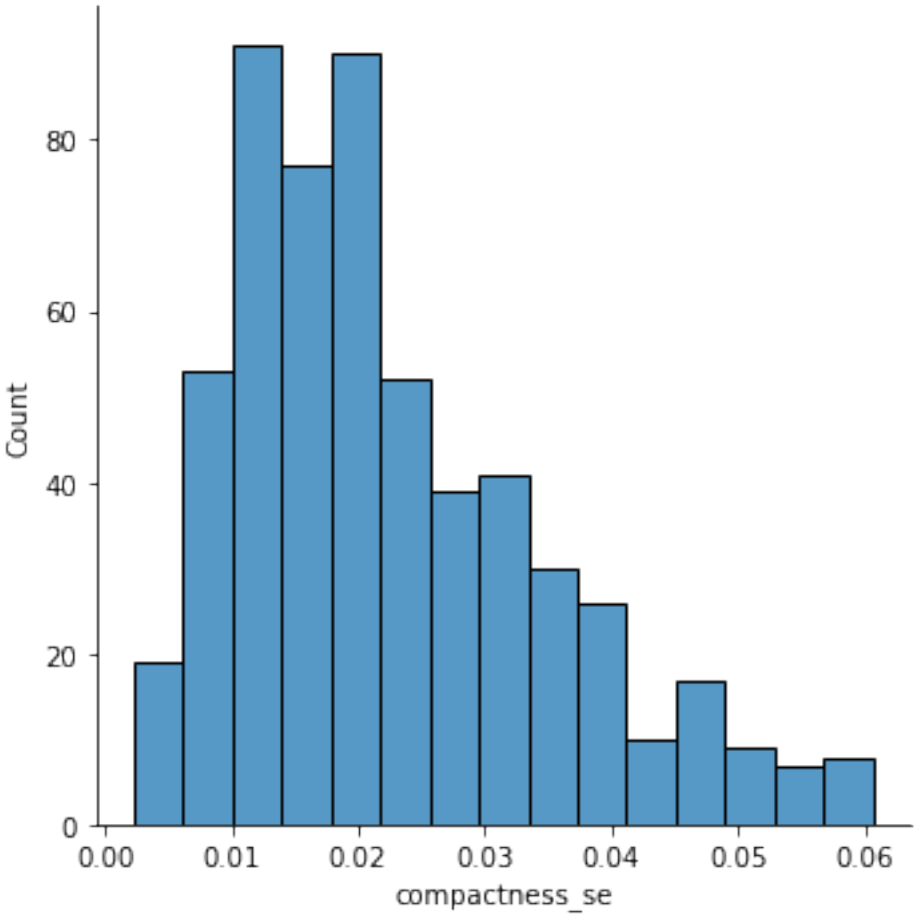

100
80
60
40
20
0
Count
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
radius_se

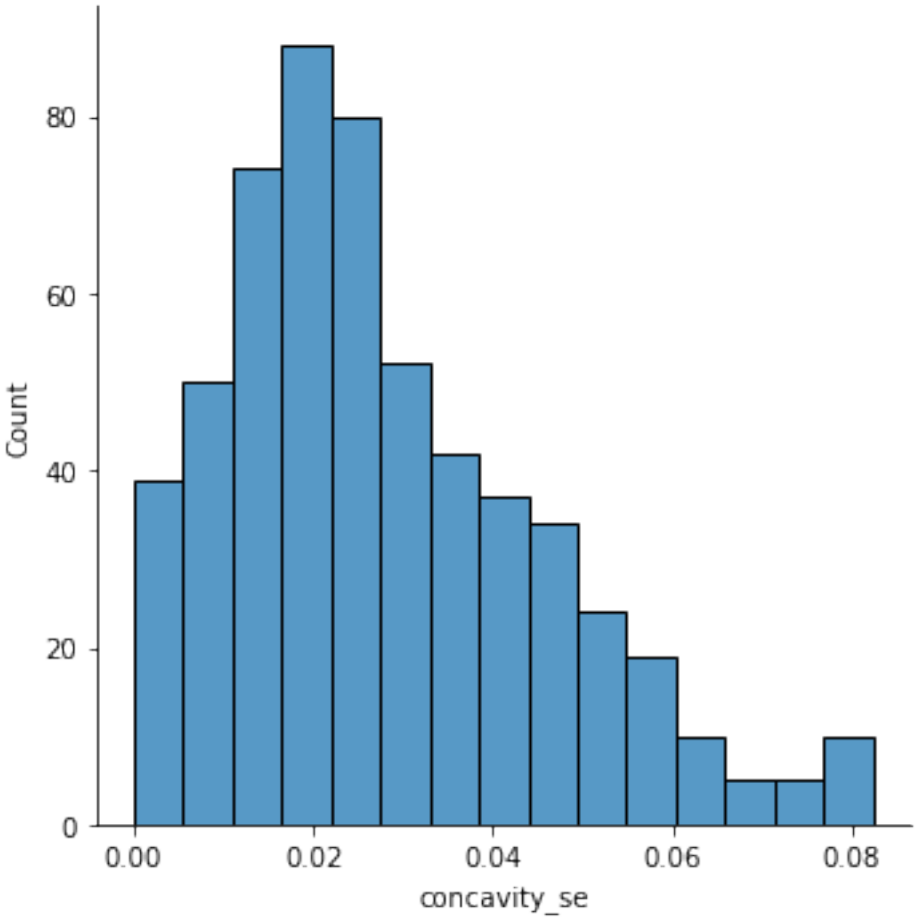

Count
80
70
60
50
40
30
20
10
0
0.5
1.0
1.5
2.0
2.5
texture_se

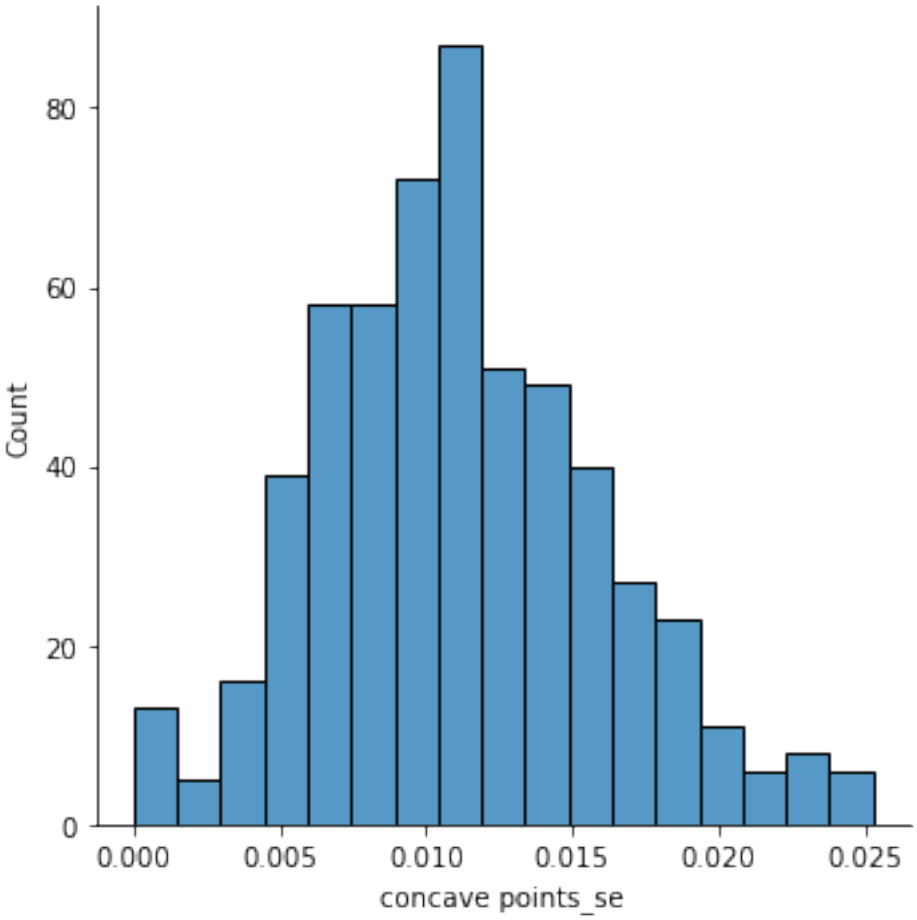

Count
100
80
60
40
20
0
1
2
3
4
5
6
perimeter_se

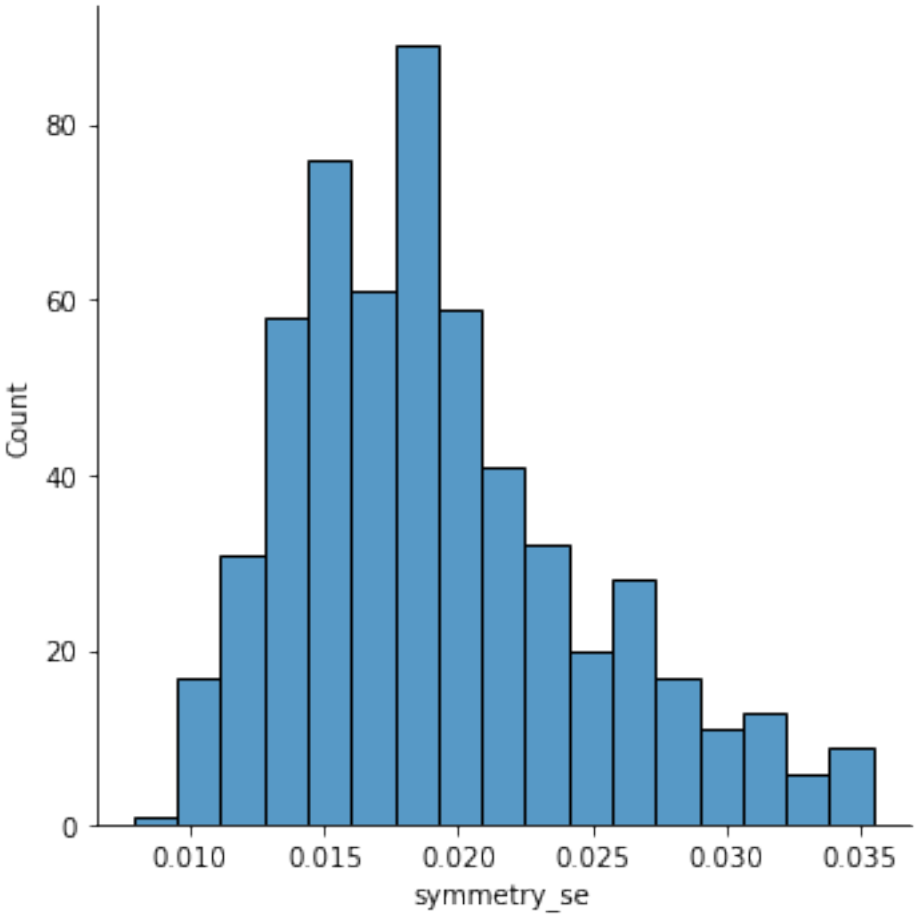

100
80
60
40
20
0
Count
10
20
30
40
50
60
70
80
area_se

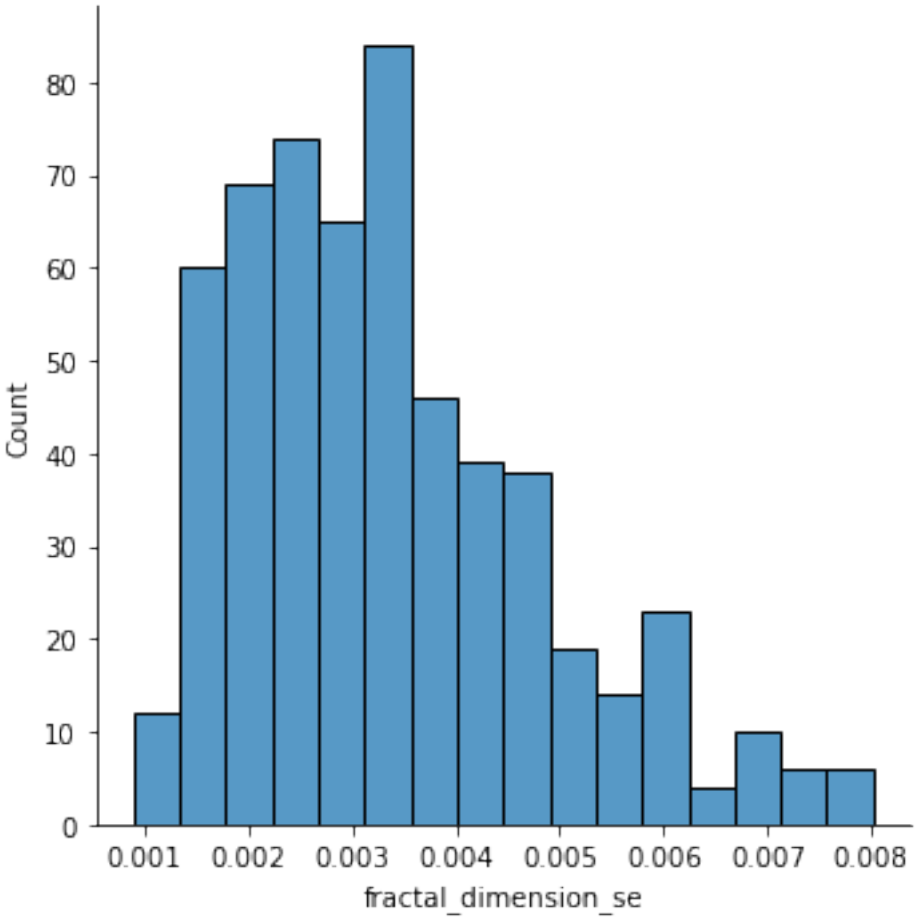

Count
80
60
40
20
0
0.002
0.004
0.006
0.008
0.010
0.012
smoothness_se

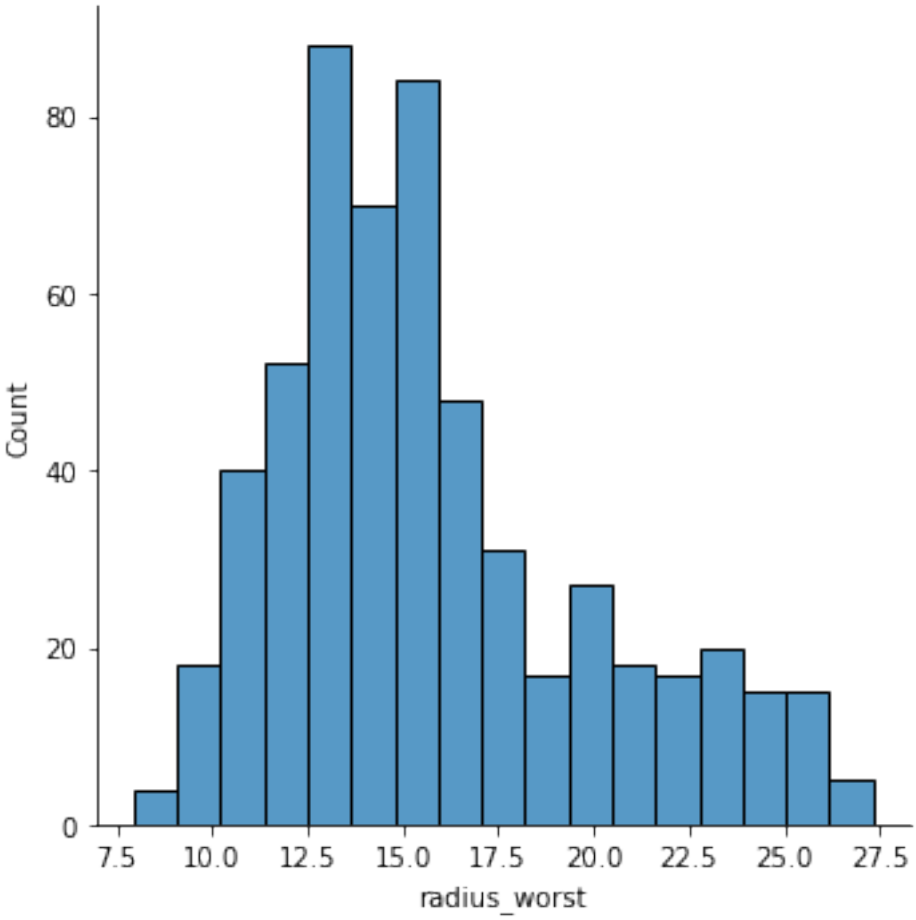

Count
80
60
40
20
0
0.00
0.01
0.02
0.03
0.04
0.05
0.06
compactness_se

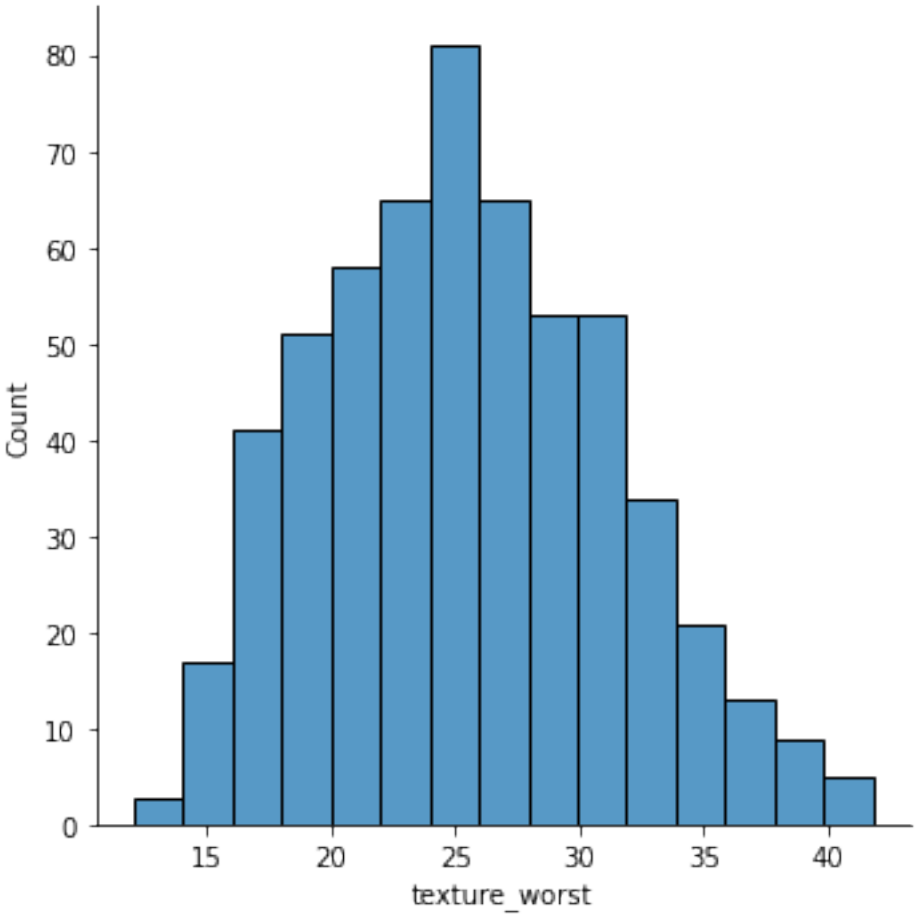

Count
80
60
40
20
0
0.00
0.02
0.04
0.06
0.08
concavity_se

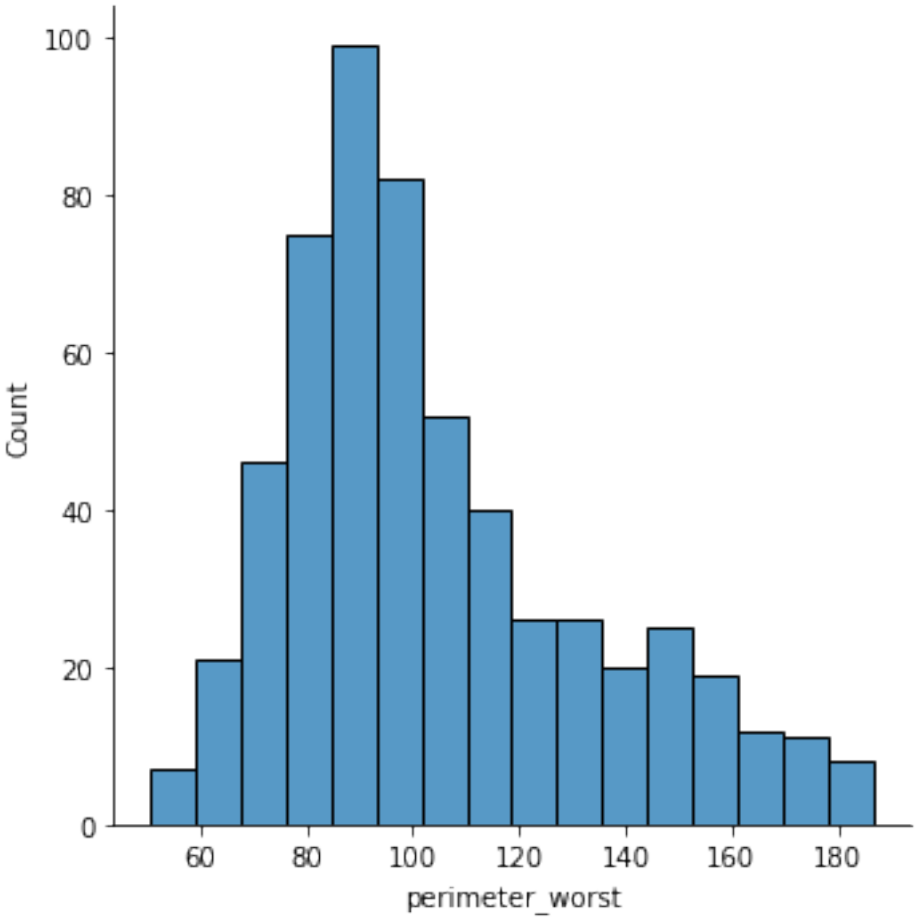

Count
80
60
40
20
0
0.000
0.005
0.010
0.015
0.020
0.025
concave points_se

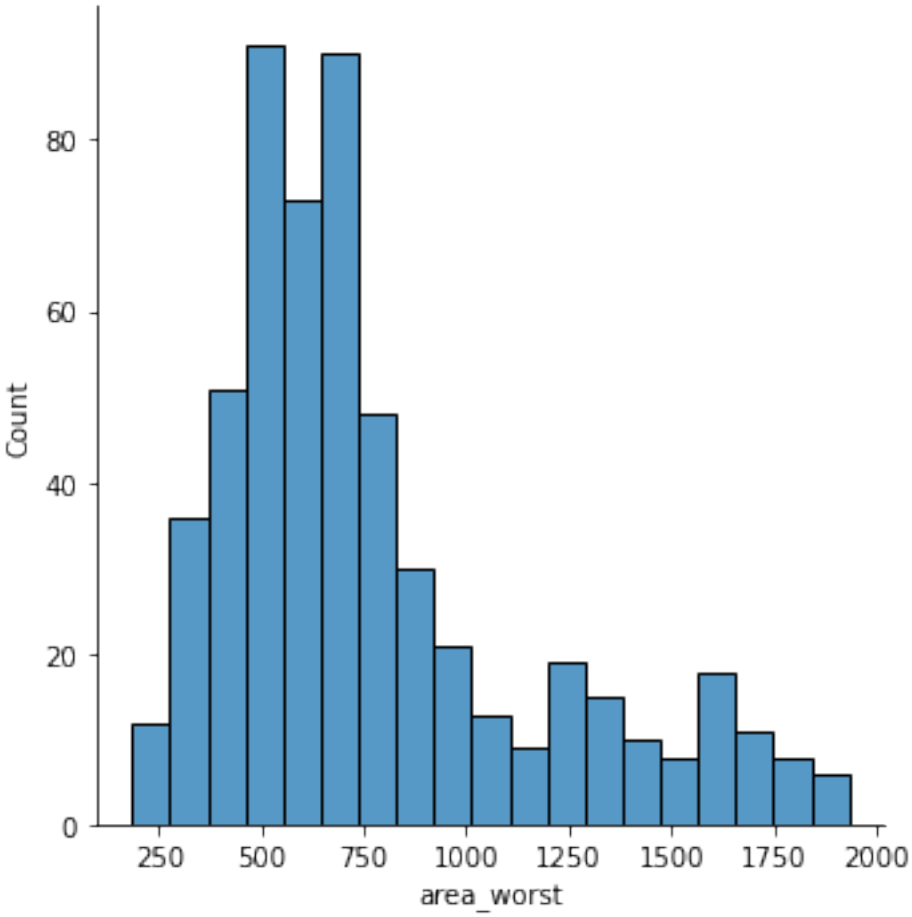

Count
80
60
40
20
0
0.010
0.015
0.020
0.025
0.030
0.035
symmetry_se

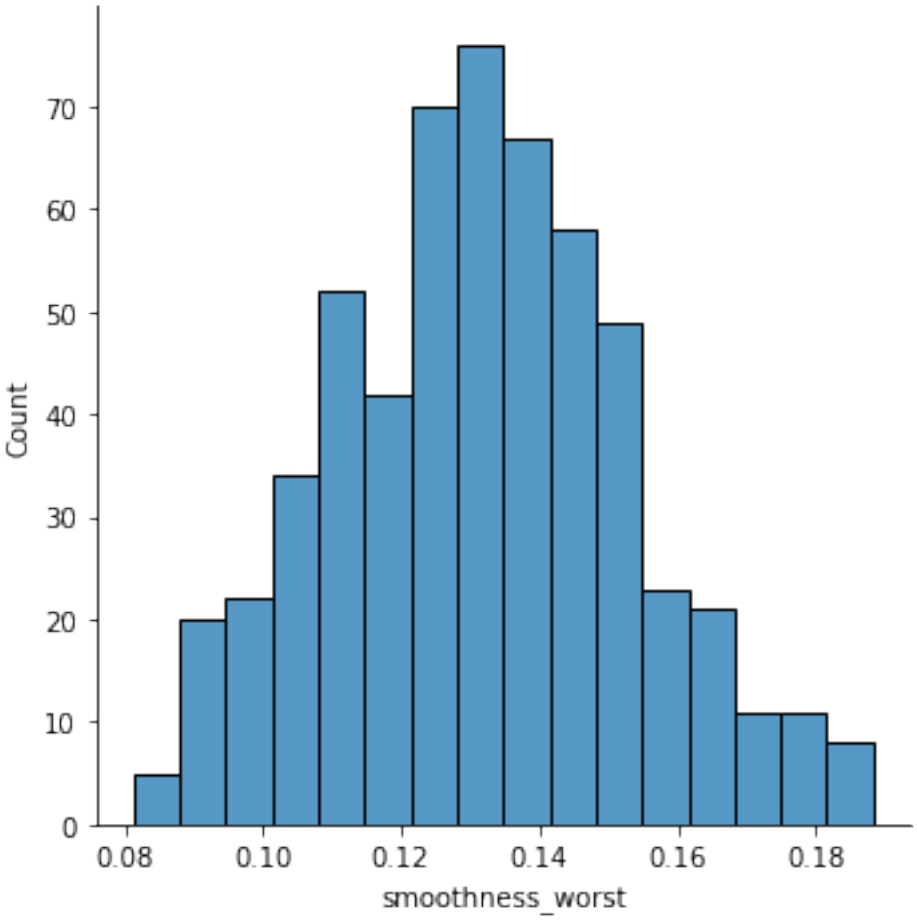

Count
80
70
60
50
40
30
20
10
0
0.001
0.002
0.003
0.004
0.005
0.006
0.007
0.008
fractal_dimension_se

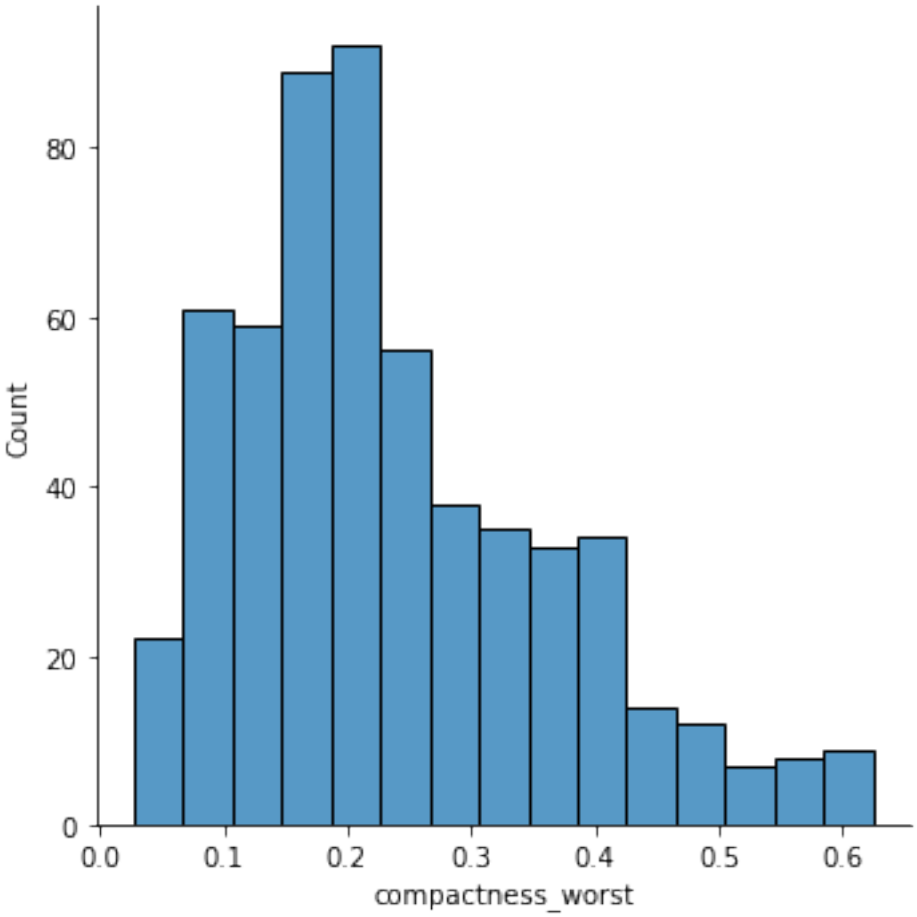

Count
0
20
40
60
80
7.5
10.0
12.5
15.0
17.5
20.0
22.5
25.0
27.5
radius_worst

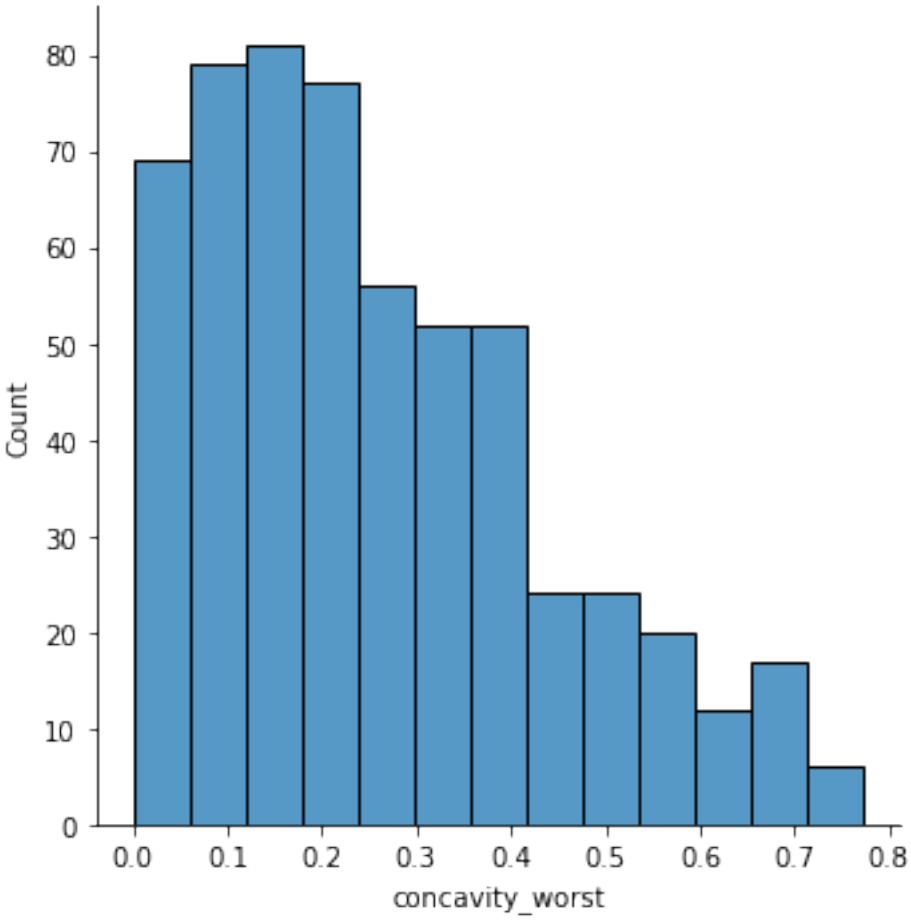

Count
texture_worst
0
10
20
30
40
50
60
70
80
15
20
25
30
35
40

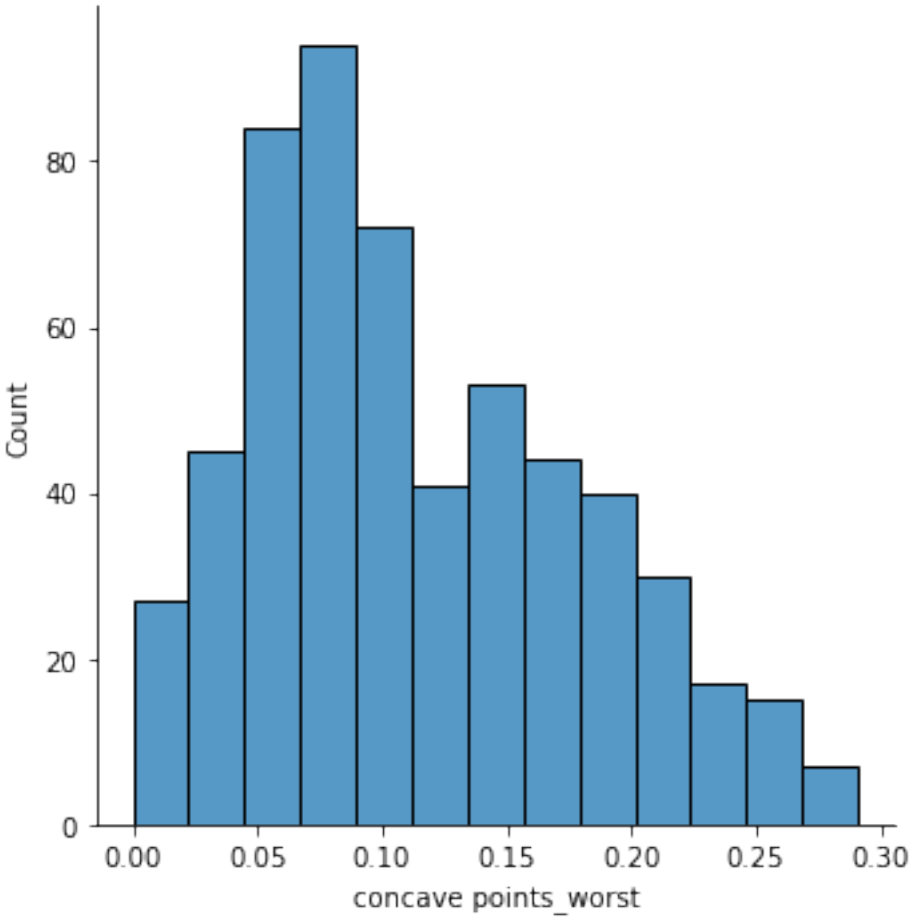

100
80
60
40
20
0
Count
60
80
100
120
140
160
180
perimeter_worst

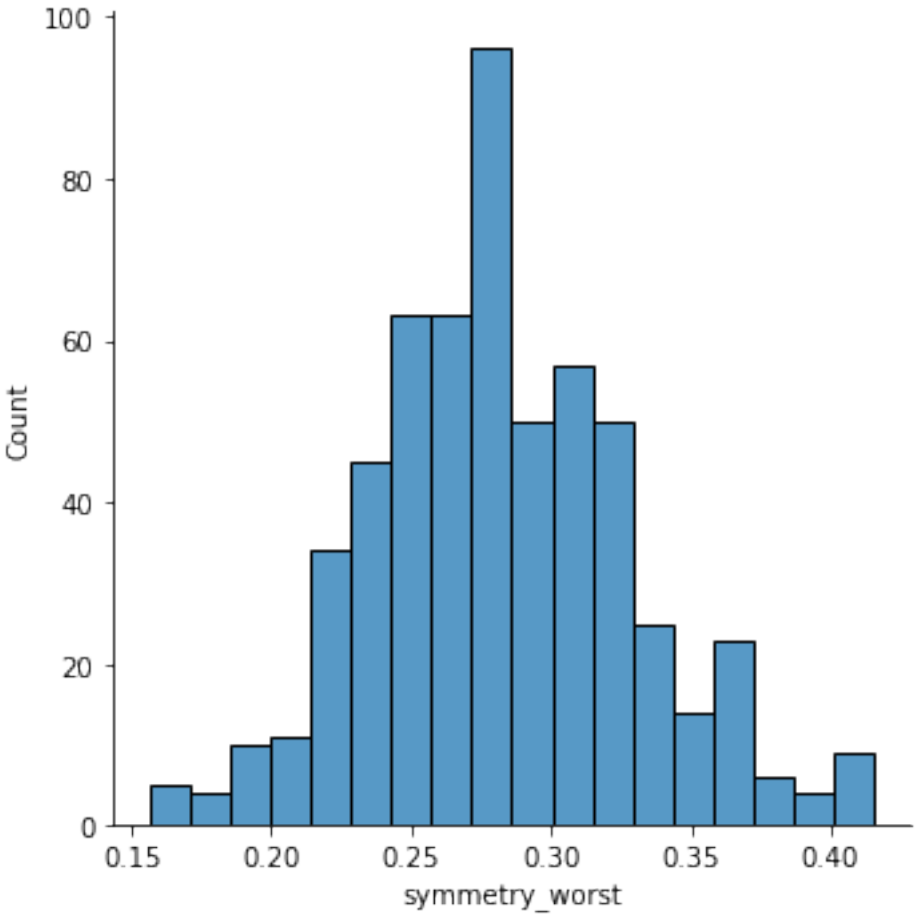

Count
0
20
40
60
80
250
500
750
1000
1250
1500
1750
2000
area_worst

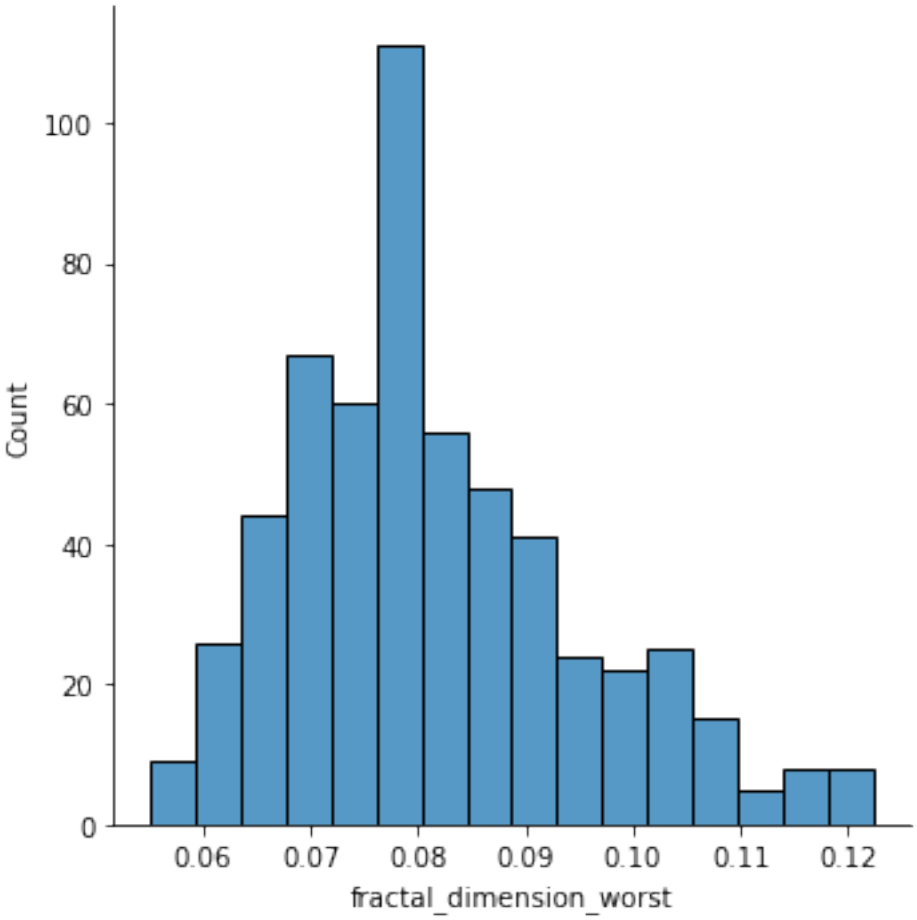

Count
70
60
50
40
30
20
10
0
0.08
0.10
0.12
0.14
0.16
0.18
smoothness_worst

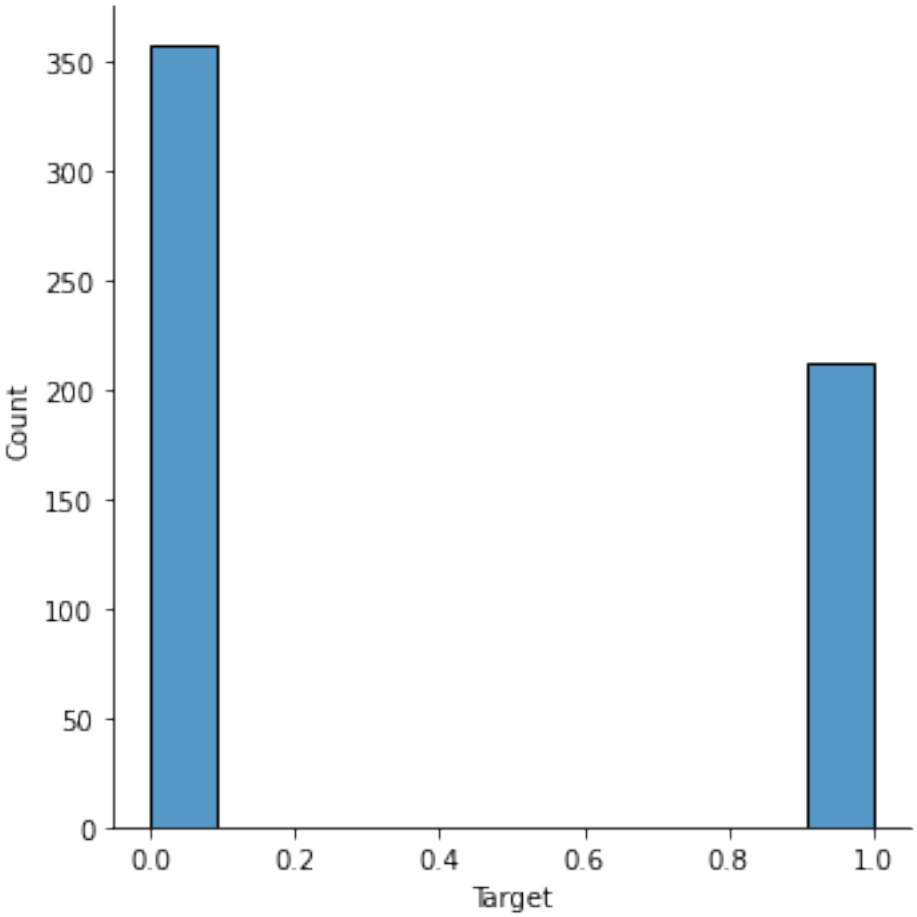

Count
80
60
40
20
0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
compactness_worst


Count
80
70
60
50
40
30
20
10
0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
concavity_worst


Count
0
20
40
60
80
0.00
0.05
0.10
0.15
0.20
0.25
0.30
concave points_worst


100
80
60
40
20
0
Count
0.15
0.20
0.25
0.30
0.35
0.40
symmetry_worst


Count
100
80
60
40
20
0
0.06
0.07
0.08
0.09
0.10
0.11
0.12
fractal_dimension_worst



As we see data is right sckewed , most of the value are left side

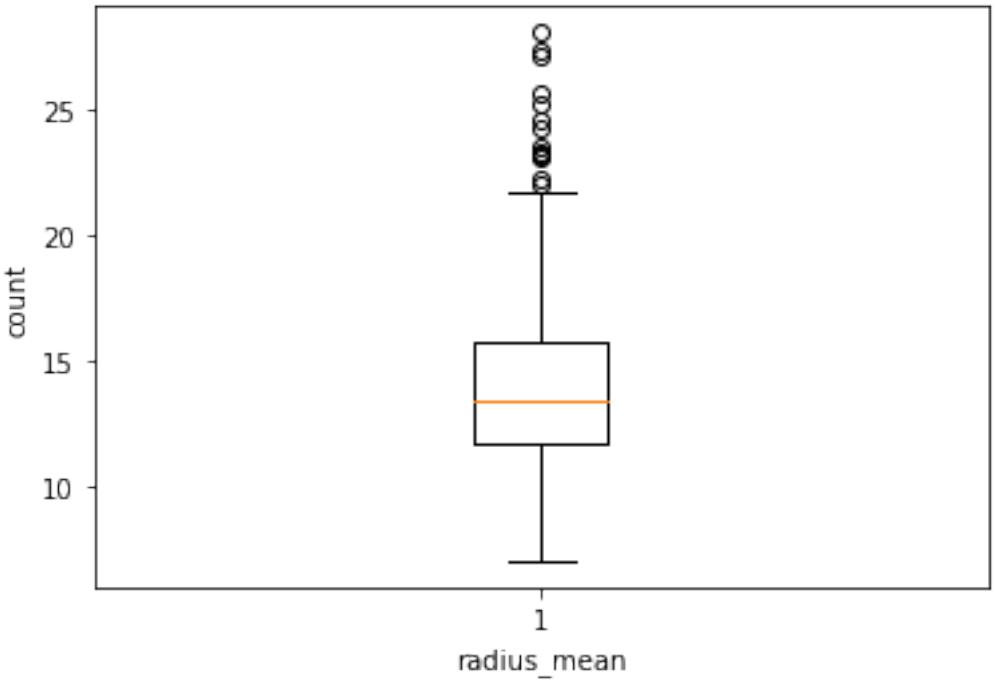
[21]:
```
df.skew()
```

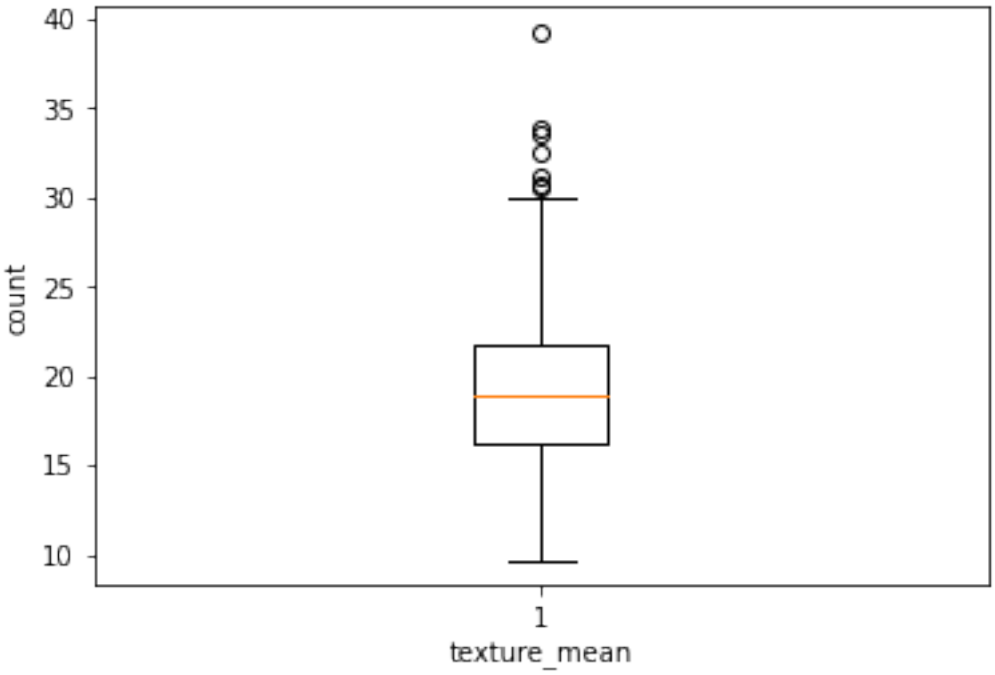
[21]:
```
radius_mean               0.942380
texture_mean              0.650450
perimeter_mean            0.990650
area_mean                 1.645732
smoothness_mean           0.456324
compactness_mean          1.190123
concavity_mean            1.401180
concave points_mean       1.171180
symmetry_mean             0.725609
fractal_dimension_mean    1.304489
radius_se                 3.088612
texture_se                1.646444
perimeter_se              3.443615
area_se                   5.447186
```

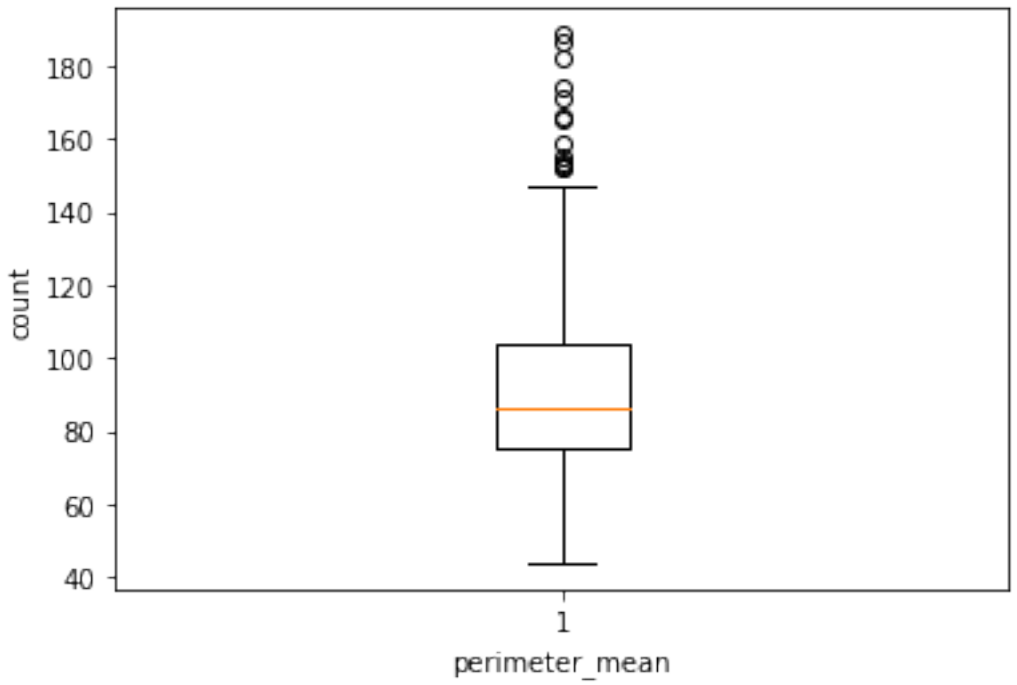
```
smoothness_se              2.314450
compactness_se             1.902221
concavity_se               5.110463
concave points_se          1.444678
symmetry_se                2.195133
fractal_dimension_se       3.923969
radius_worst               1.103115
texture_worst              0.498321
perimeter_worst            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
Target                     0.528461
dtype: float64
```

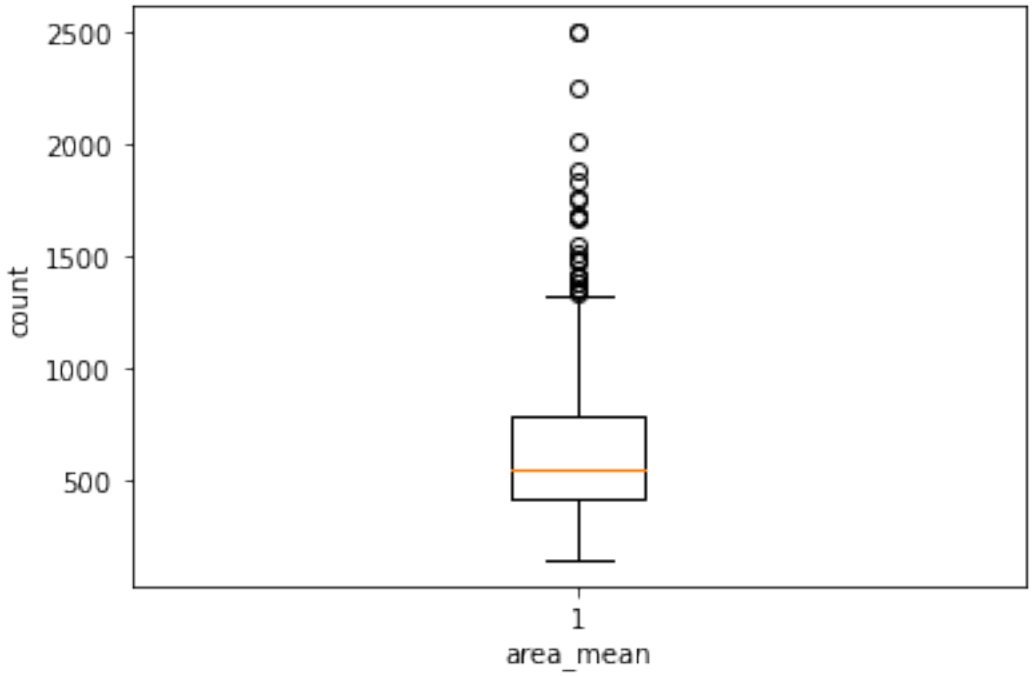
If the skewness is between -0.5 and 0.5, the data is approximately symmetric. If the skewness is between -1 and -0.5 or between 0.5 and 1, the data is moderately skewed. If the skewness is less than -1 or greater than 1, the data is highly skewed.

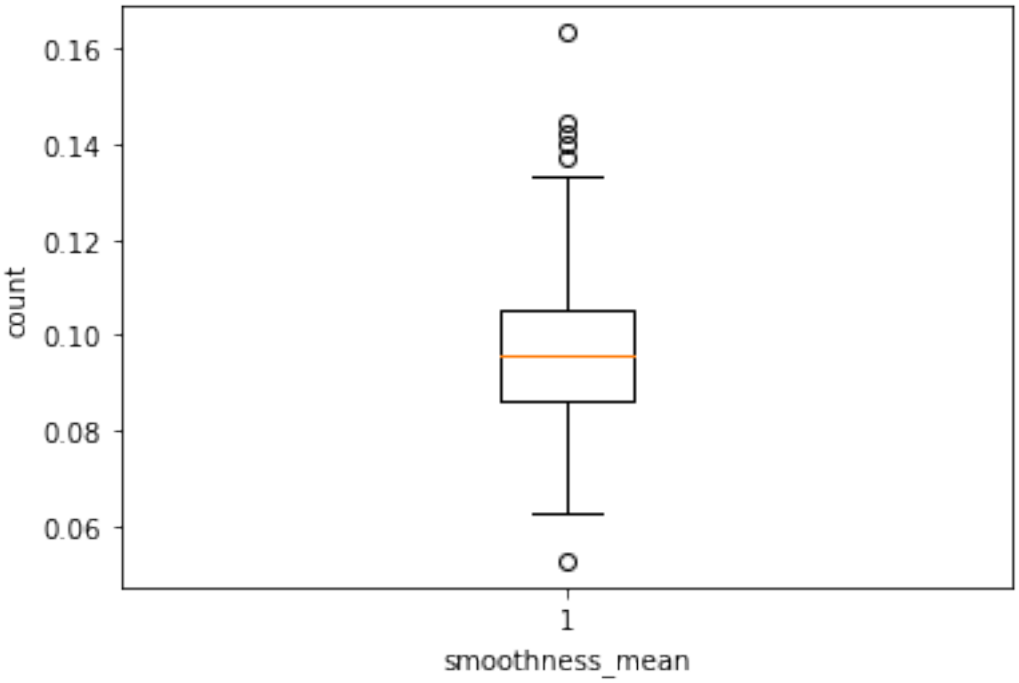
[22]:
```
for col_name in df.columns:
    if(df[col_name].dtypes=='int64' or df[col_name].dtypes=='float64'):
        plt.boxplot(df[col_name])
        plt.xlabel(col_name)
        plt.ylabel('count')
        plt.show()
```

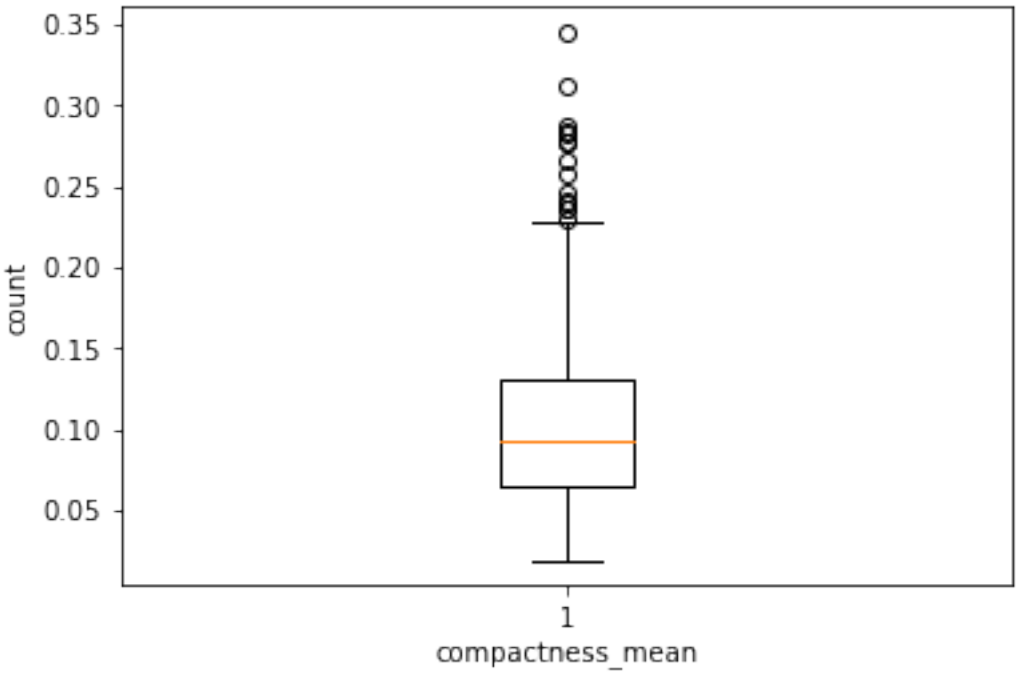

25
20
15
10
count
1
radius_mean

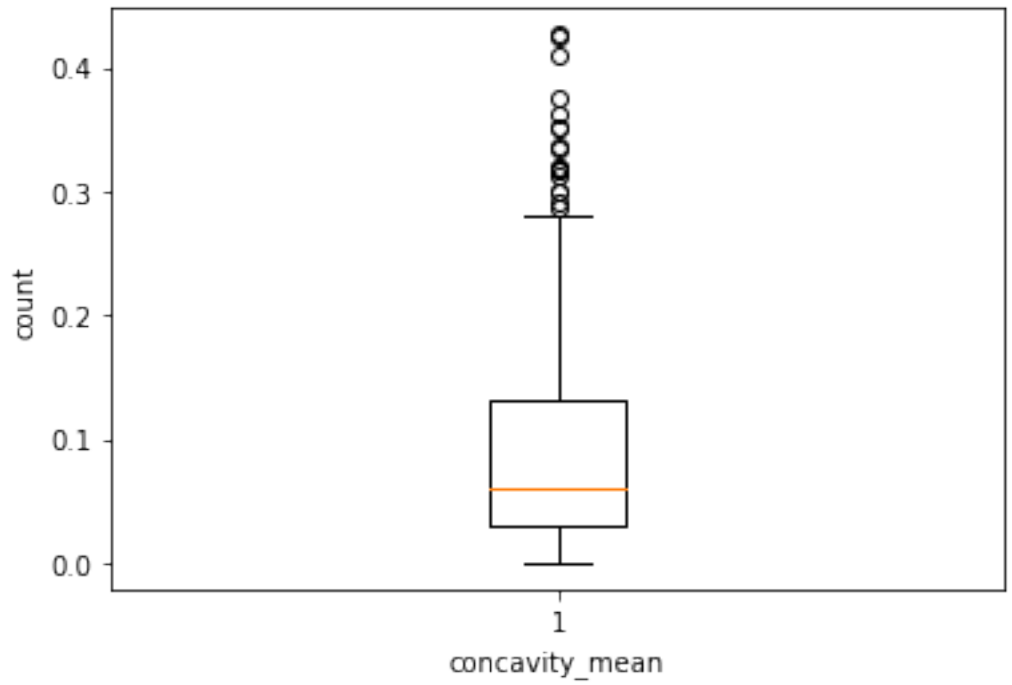

40
35
30
25
20
15
10
count
1
texture_mean

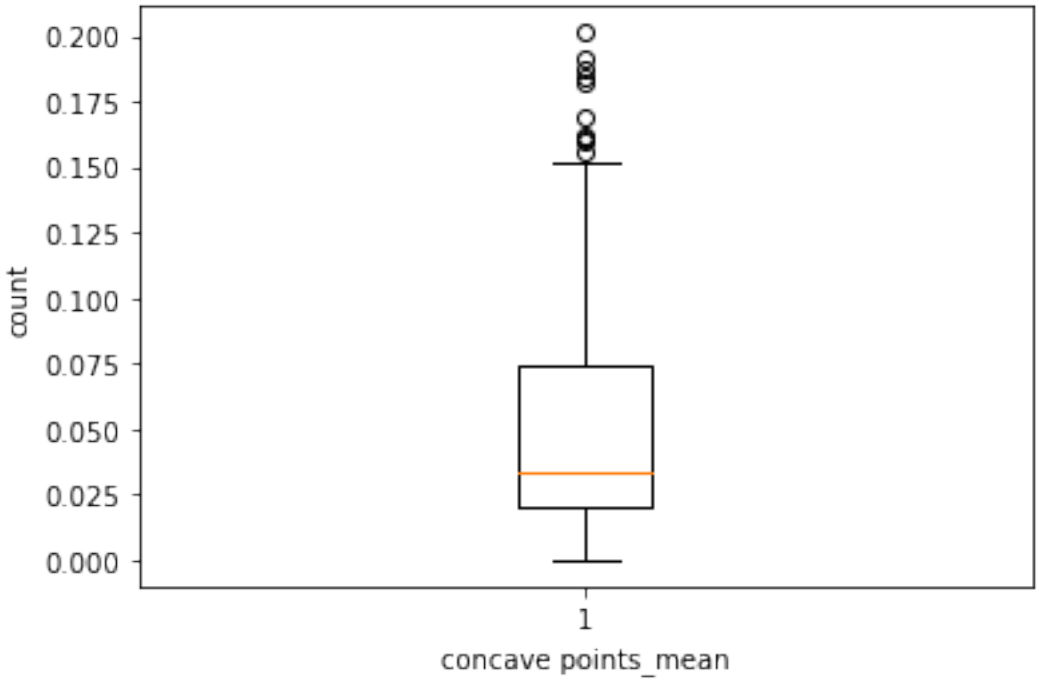

180
160
140
120
100
80
60
40
count
1
perimeter_mean

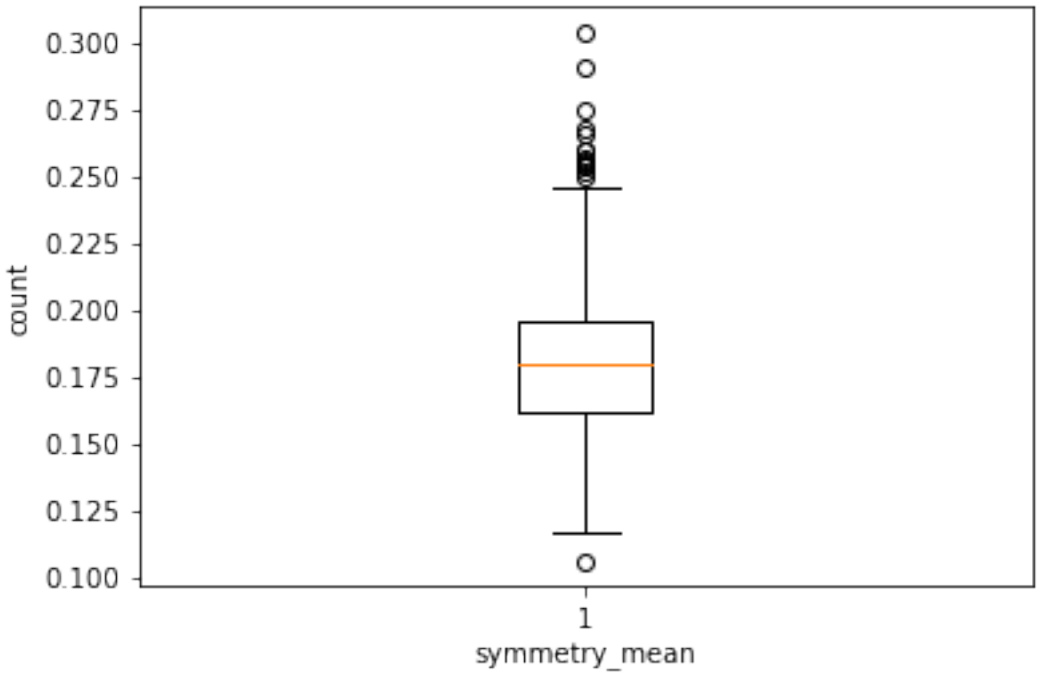

2500
2000
1500
1000
500
count
1
area_mean

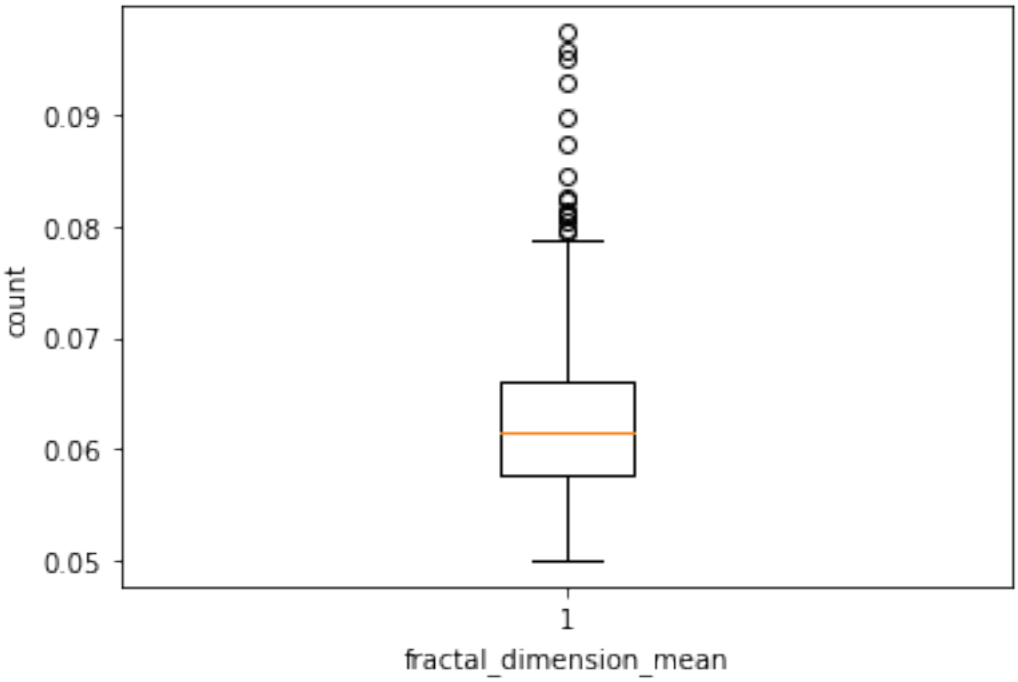

0.16
0.14
0.12
0.10
0.08
0.06
count
1
smoothness_mean

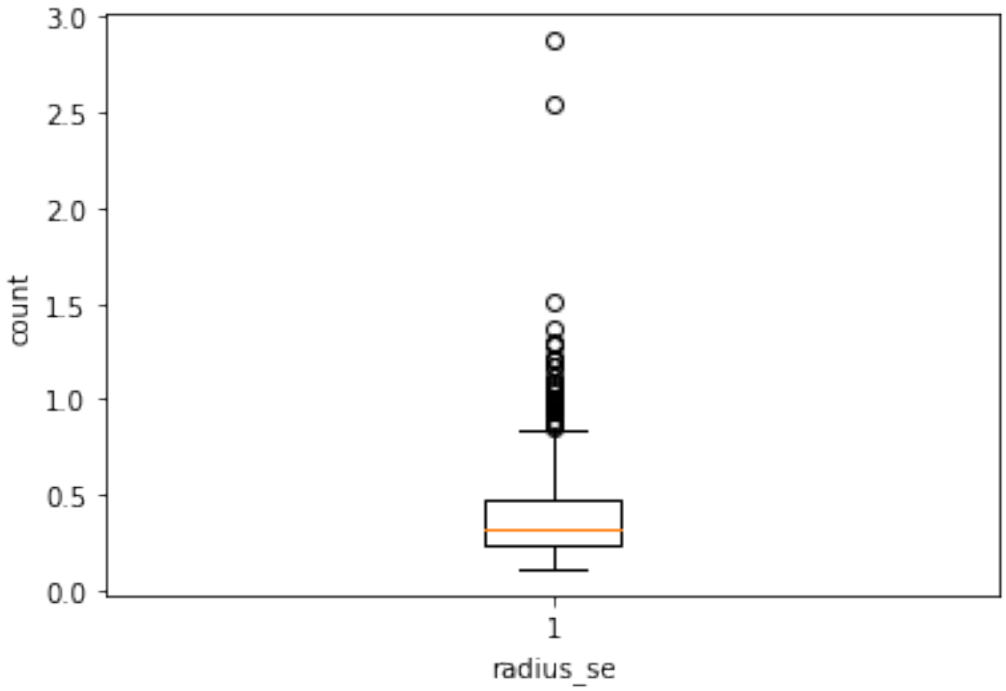

0.35
0.30
0.25
0.20
0.15
0.10
0.05
count
1
compactness_mean

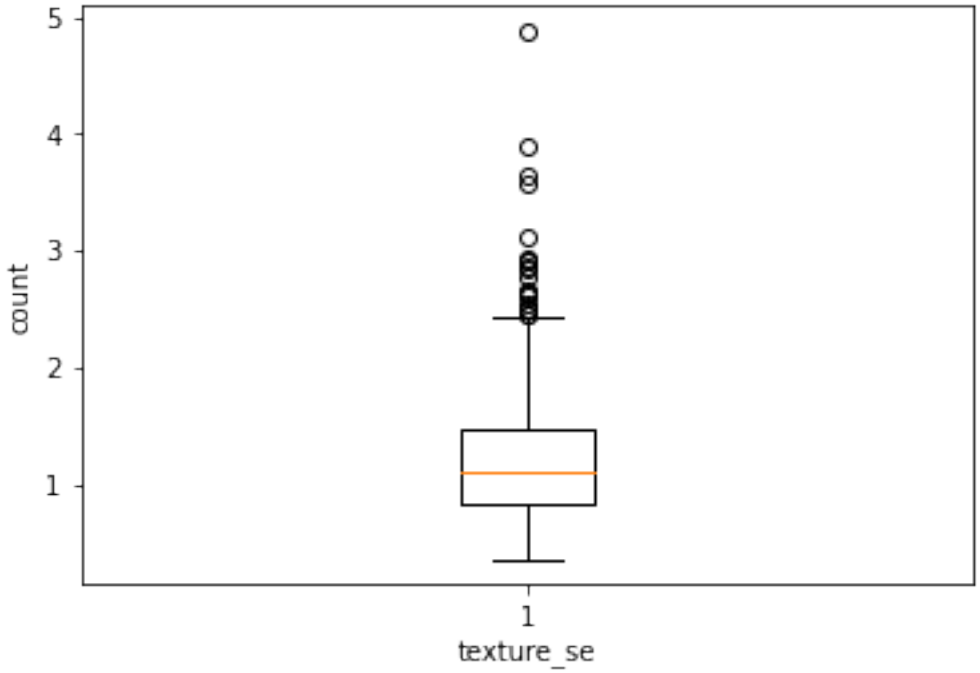

0.4
0.3
0.2
0.1
0.0
count
1
concavity_mean

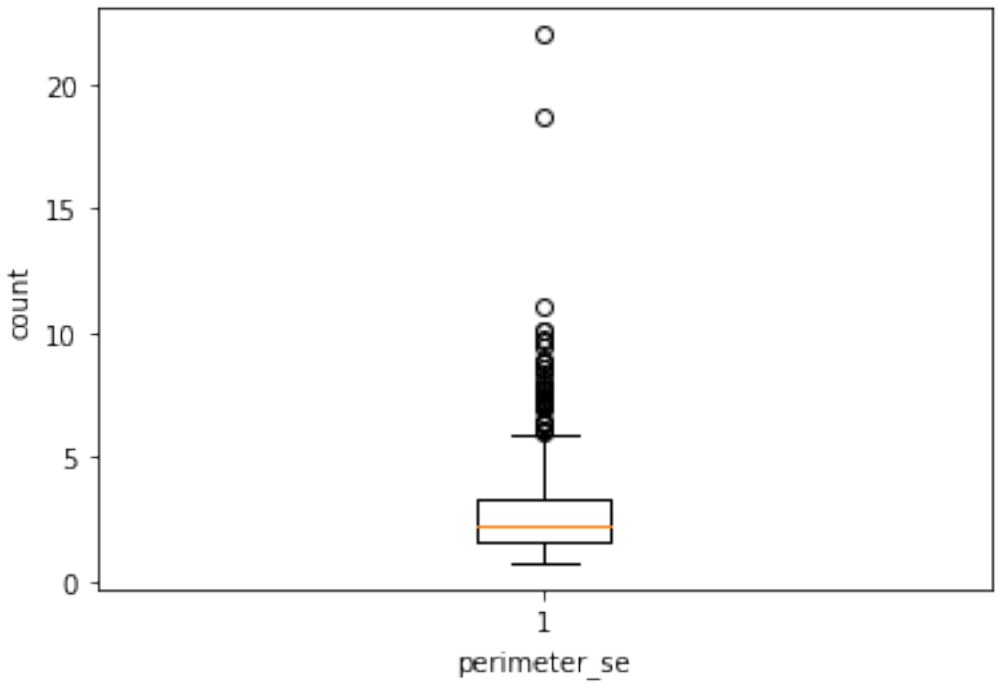

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
count
1
concave points_mean

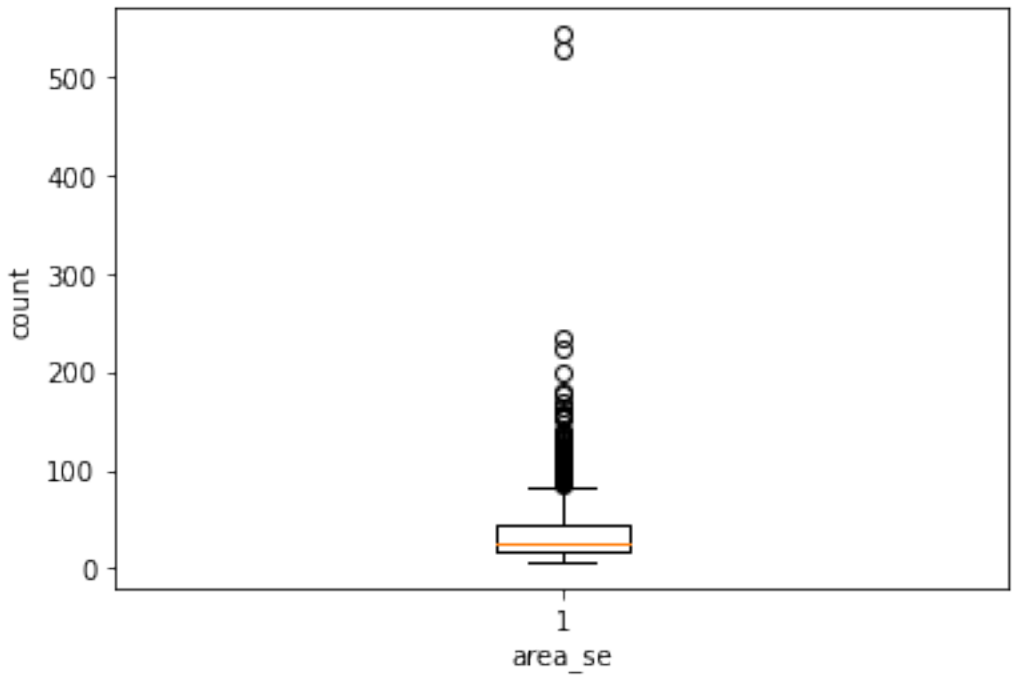

0.300
0.275
0.250
0.225
0.200
0.175
0.150
0.125
0.100
count
1
symmetry_mean

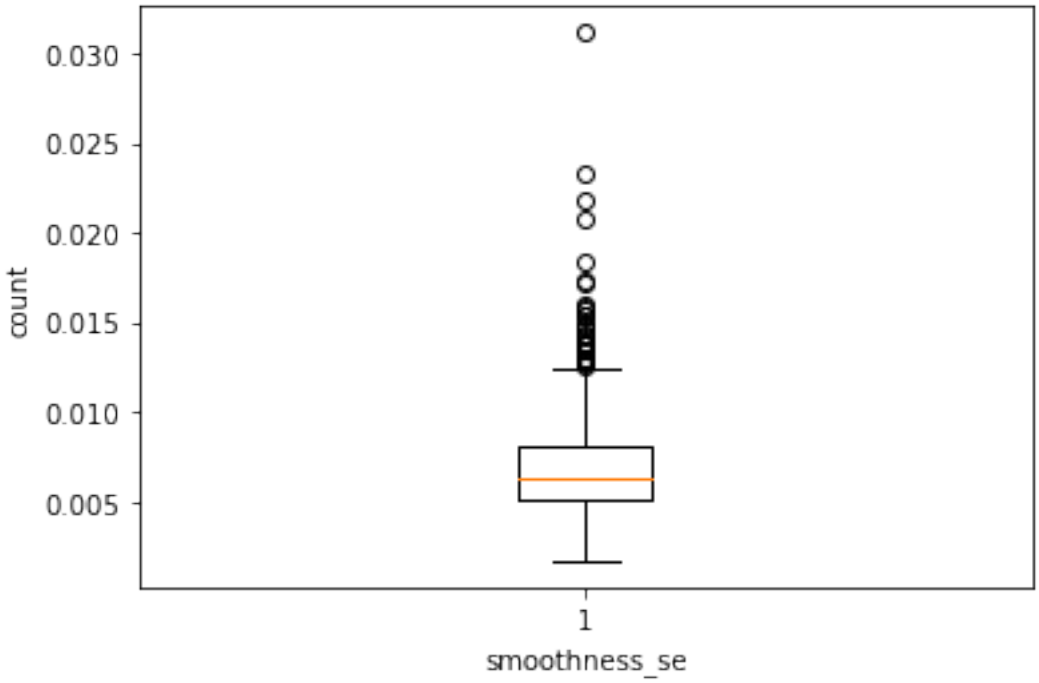

0.09
0.08
0.07
0.06
0.05
count
1
fractal_dimension_mean

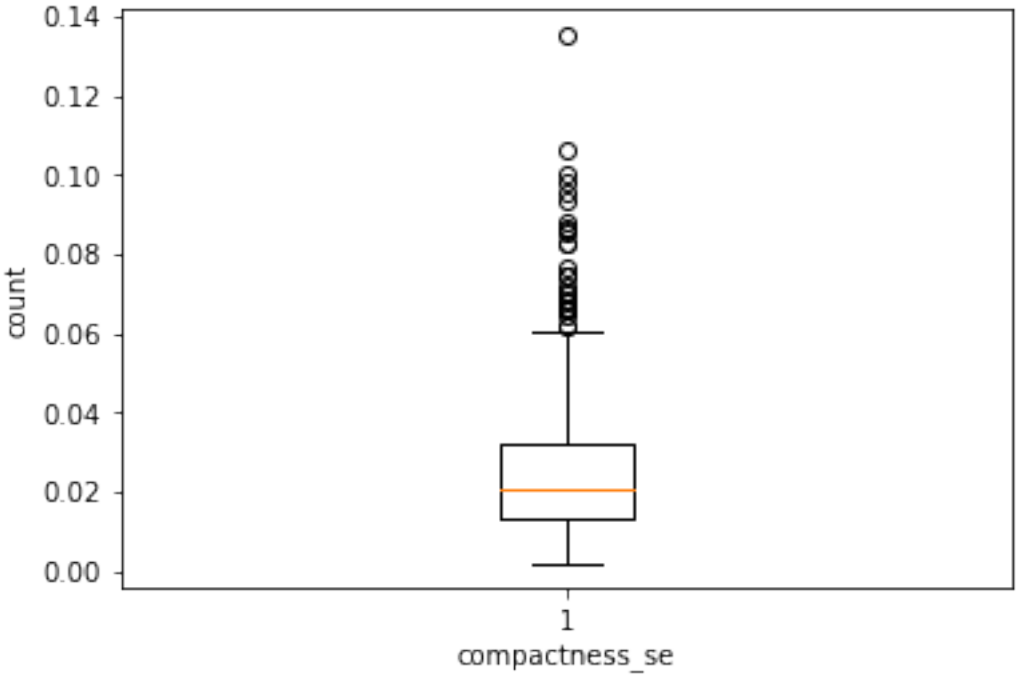

3.0
2.5
2.0
1.5
1.0
0.5
0.0
count
1
radius_se

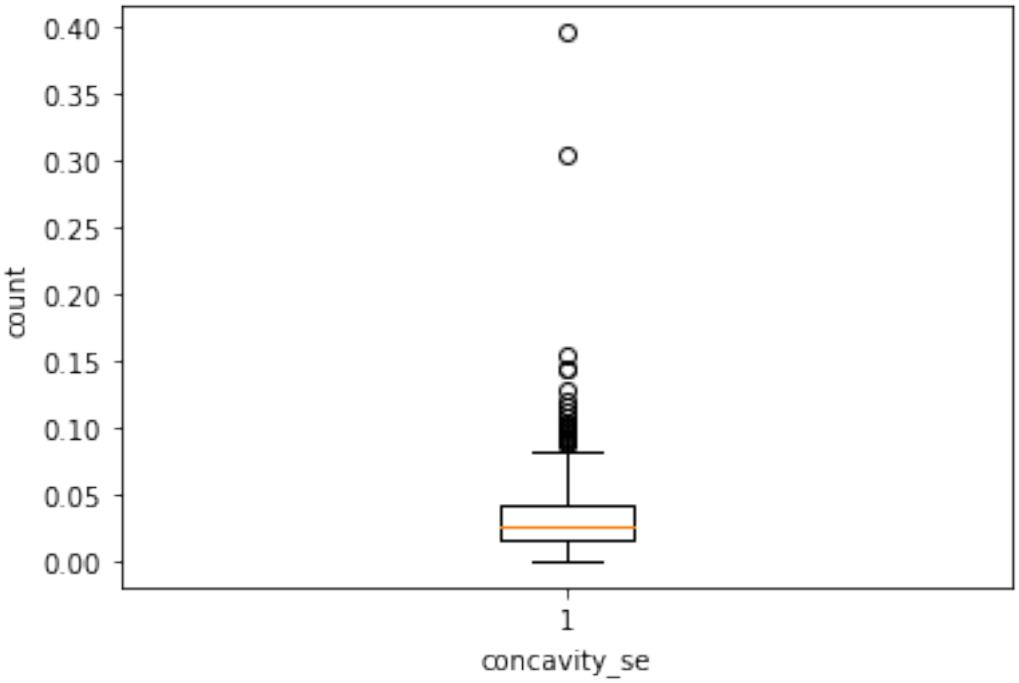

5
4
3
2
1
count
1
texture_se

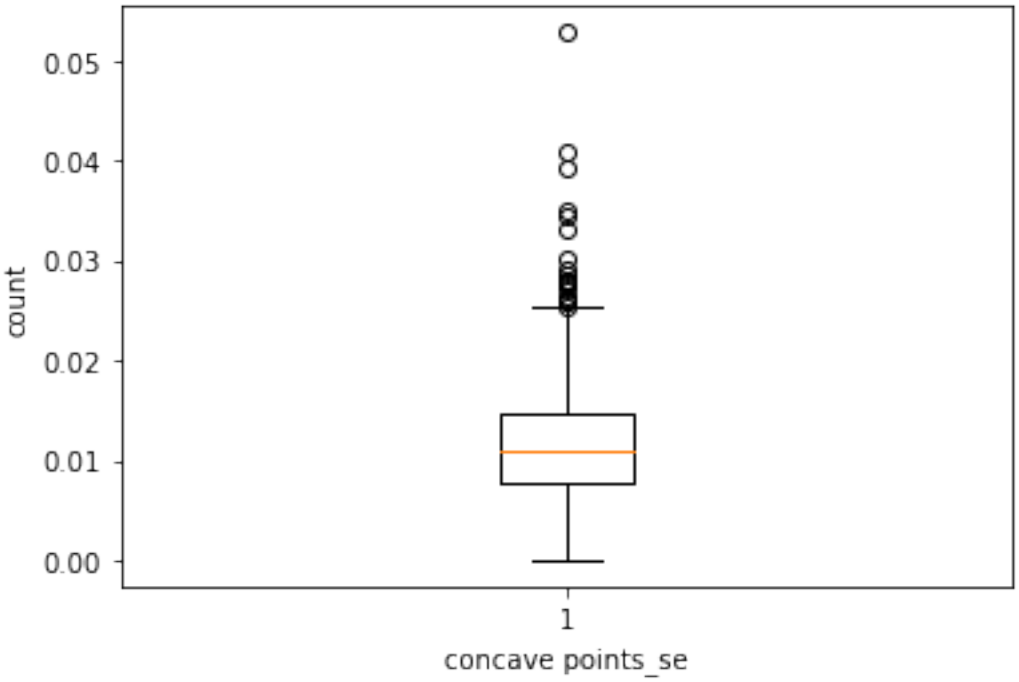

20
15
10
5
0
count
1
perimeter_se

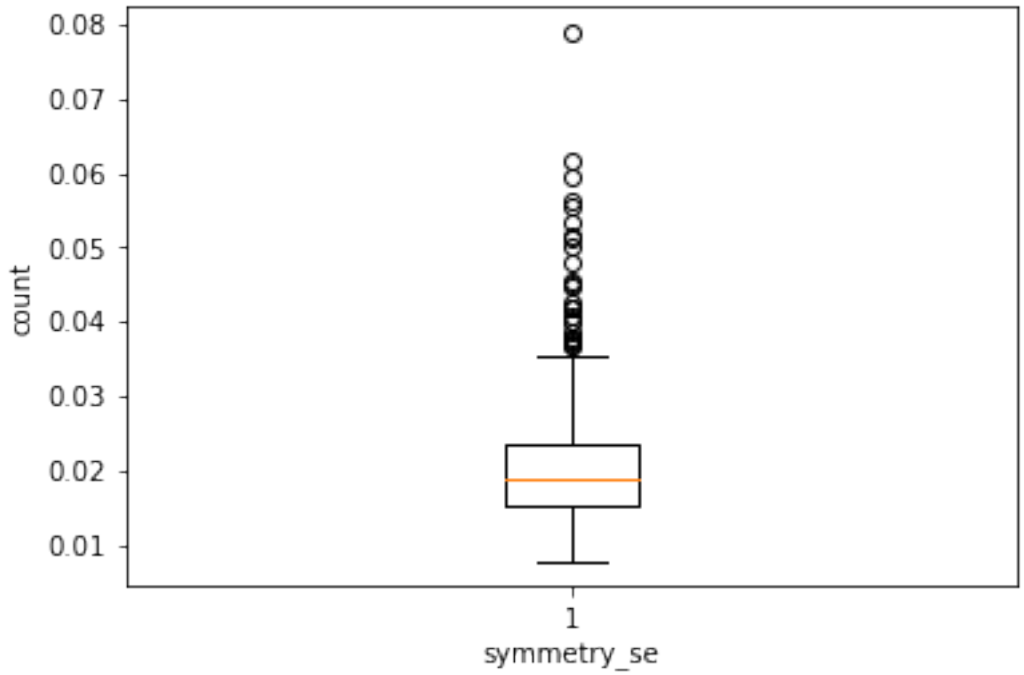

500
400
300
200
100
0
count
1
area_se

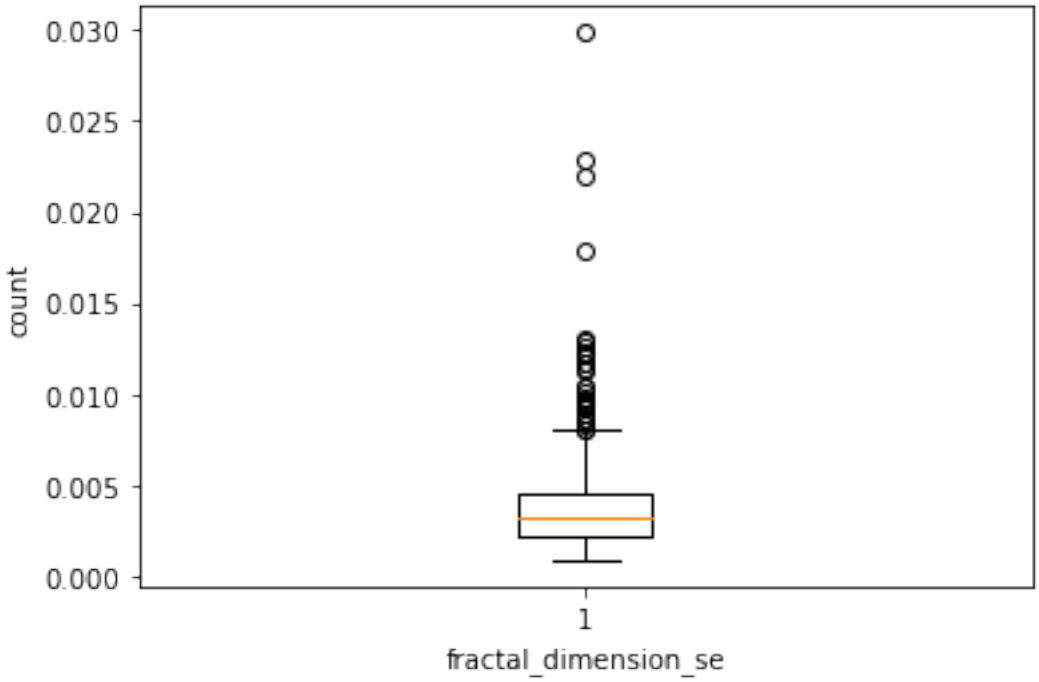

0.030
0.025
0.020
0.015
0.010
0.005
count
1
smoothness_se

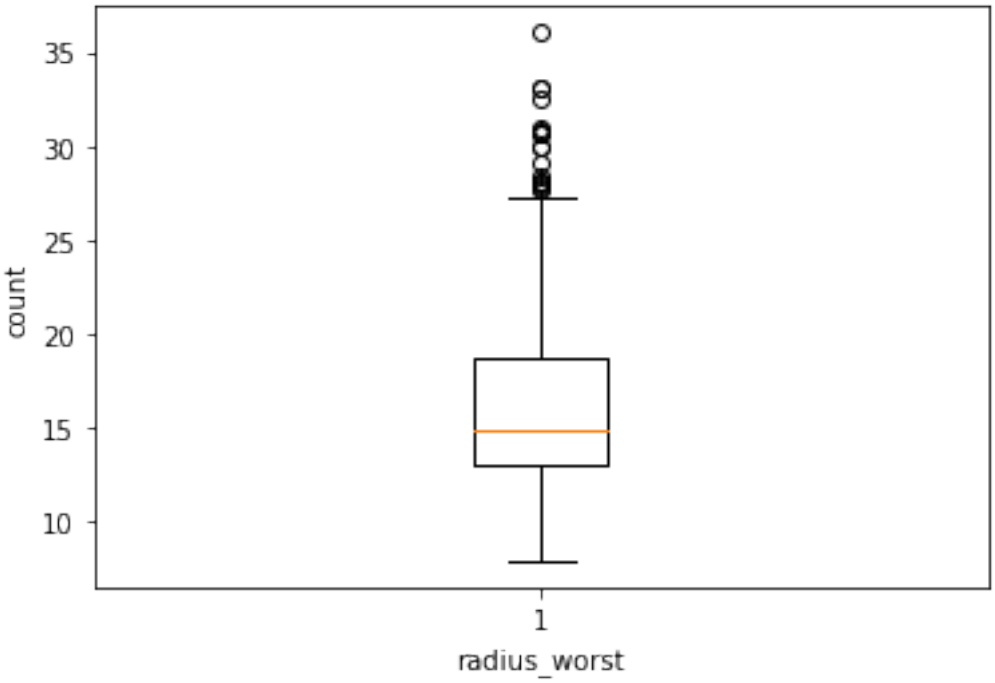

0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
count
1
compactness_se

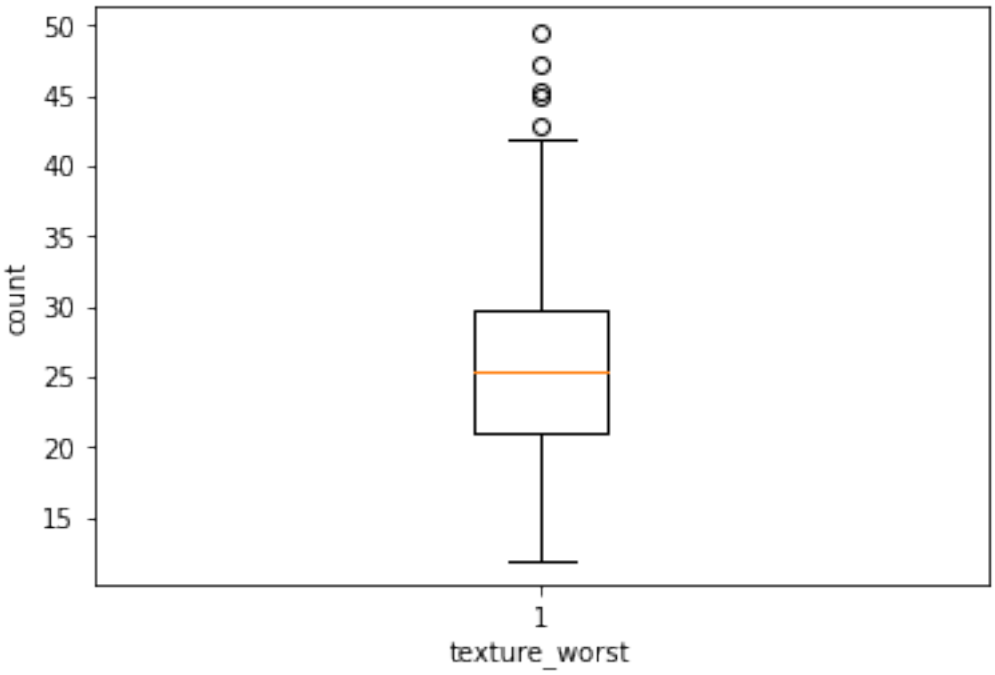

0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
count
1
concavity_se

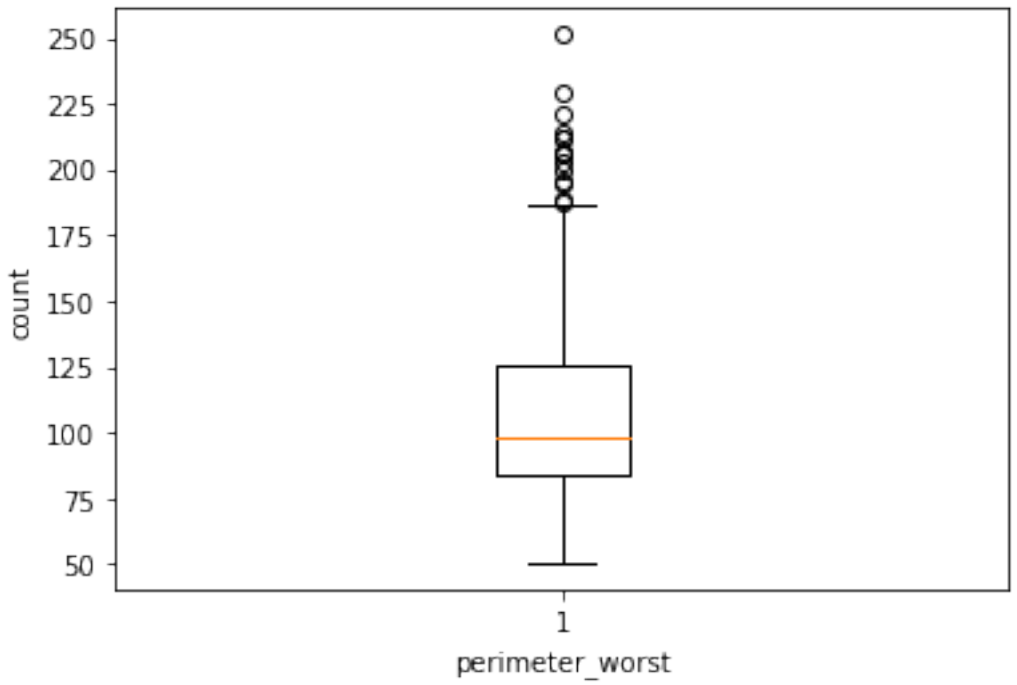

0.05
0.04
0.03
0.02
0.01
0.00
count
1
concave points_se

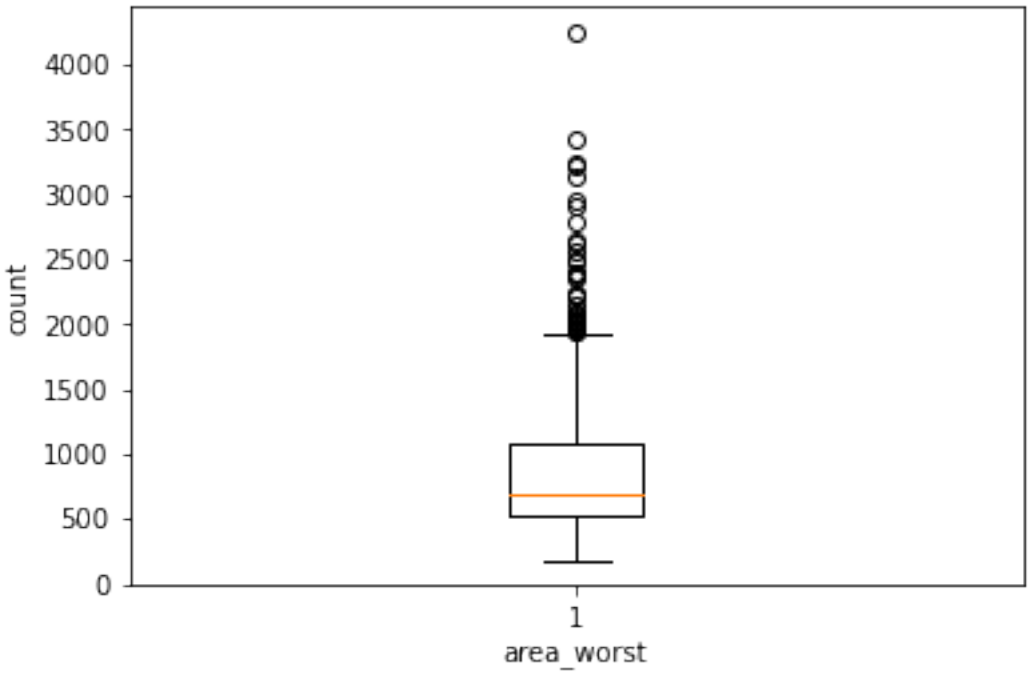

0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
count
1
symmetry_se

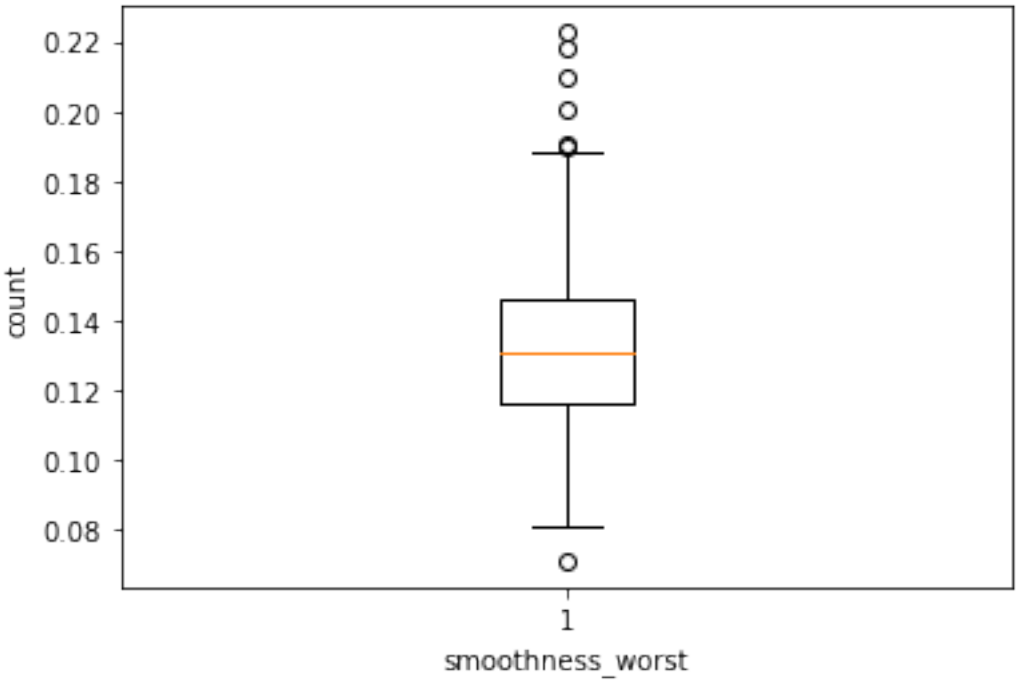

0.030
0.025
0.020
0.015
0.010
0.005
0.000
count
1
fractal_dimension_se

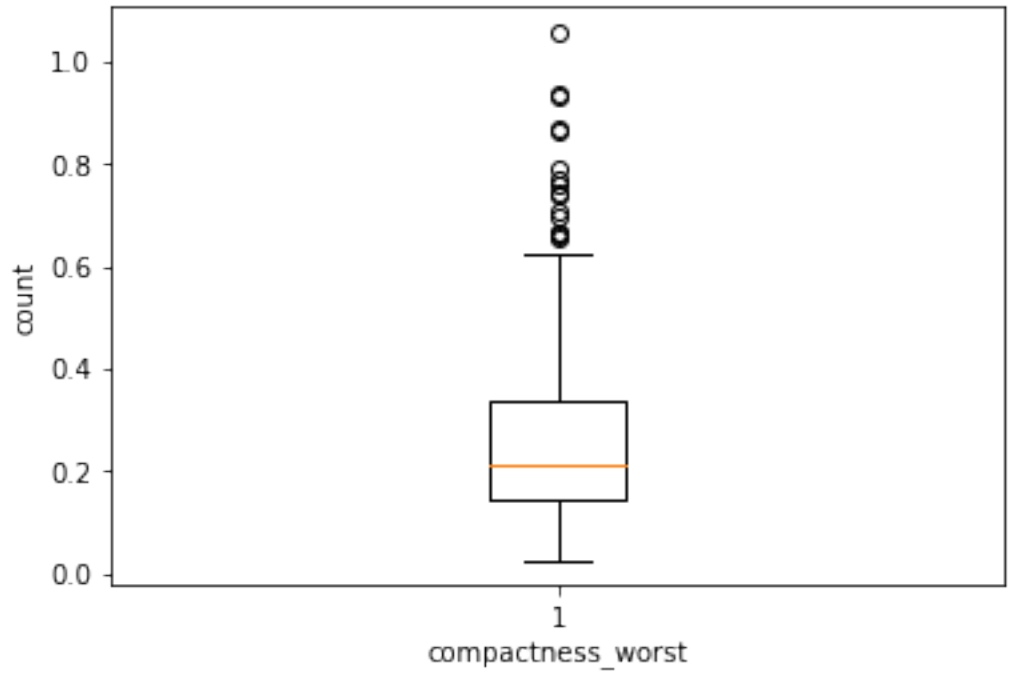

35
30
25
20
15
10
count
1
radius_worst

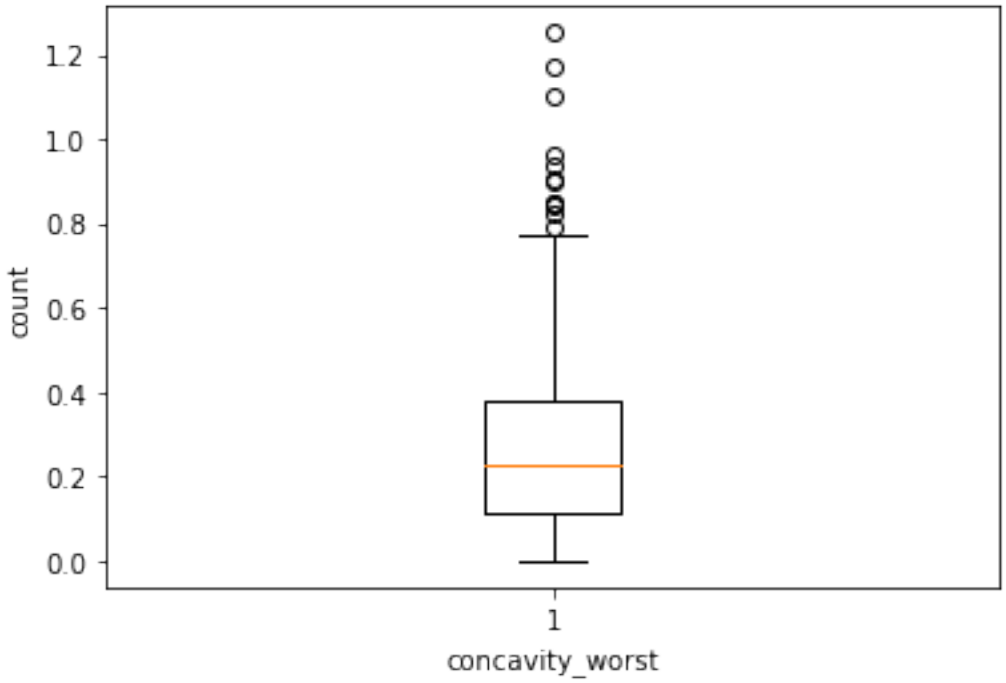

50
45
40
35
30
25
20
15
count
1
texture_worst

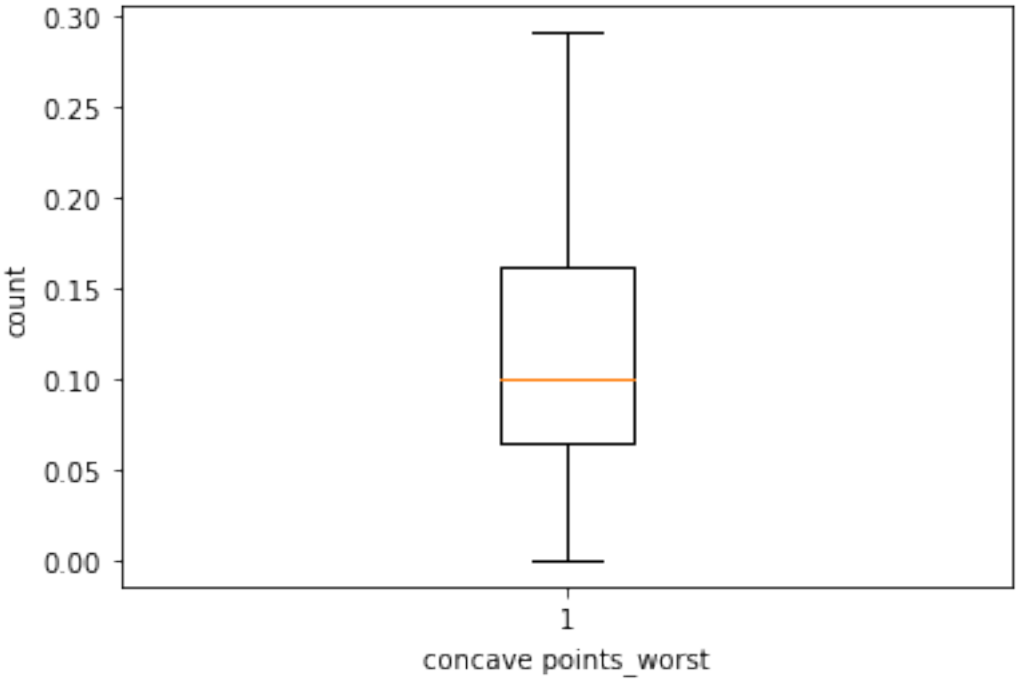

250
225
200
175
150
125
100
75
50
count
1
perimeter_worst

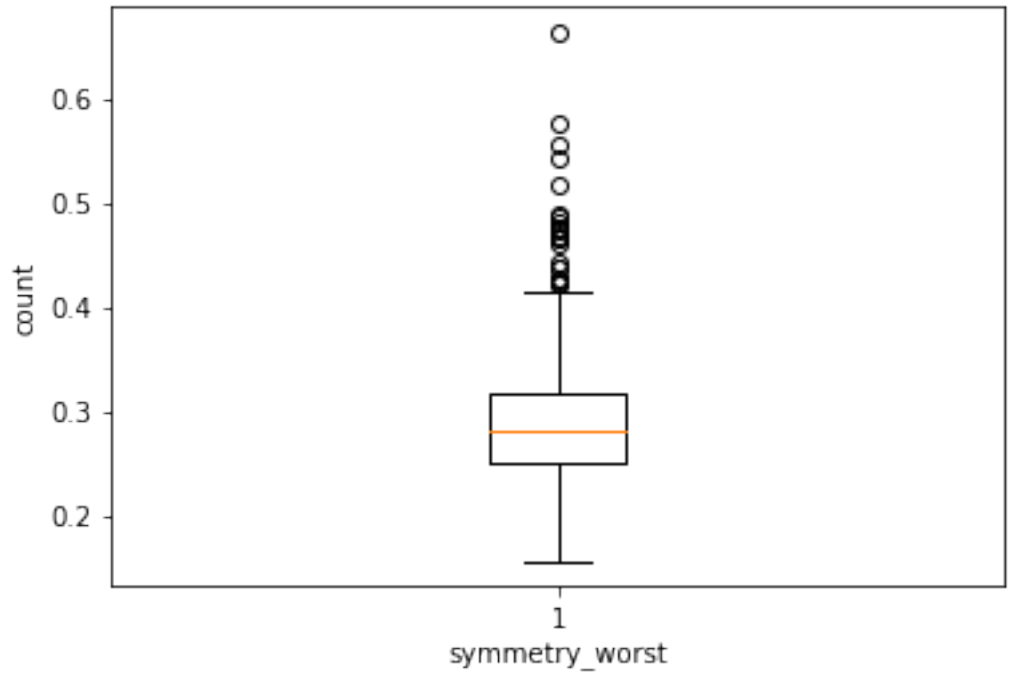

4000
3500
3000
2500
2000
1500
1000
500
0
count
1
area_worst

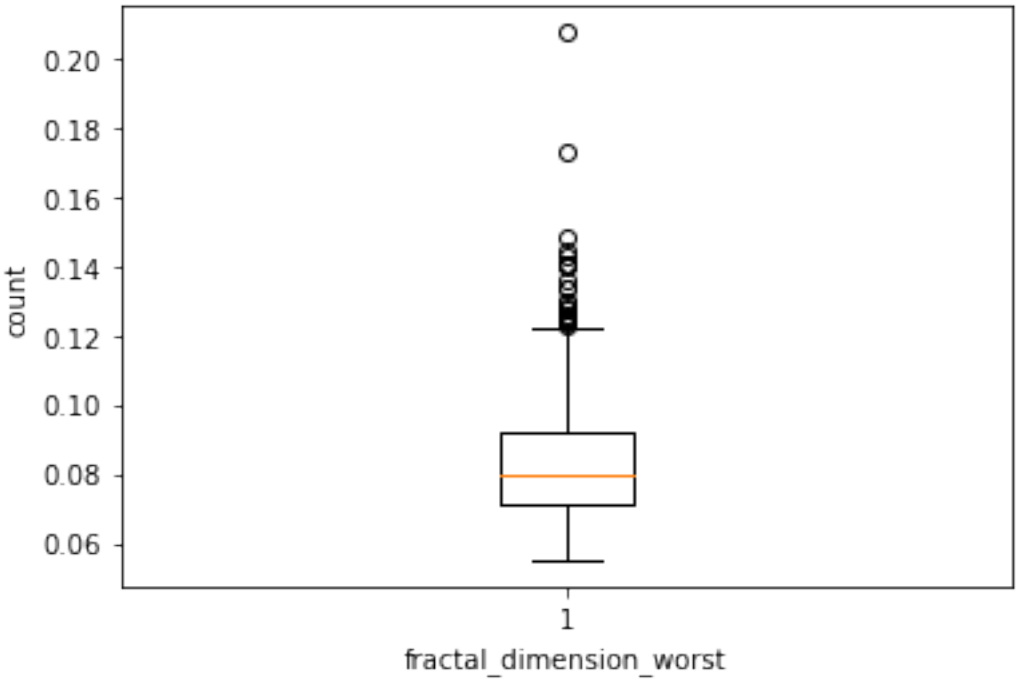

0.22
0.20
0.18
0.16
0.14
0.12
0.10
0.08
count
1
smoothness_worst


1.0
0.8
0.6
0.4
0.2
0.0
count
1
compactness_worst


1.2
1.0
0.8
0.6
0.4
0.2
0.0
count
1
concavity_worst


0.30
0.25
0.20
0.15
0.10
0.05
0.00
count
1
concave points_worst


0.6
0.5
0.4
0.3
0.2
count
1
symmetry_worst

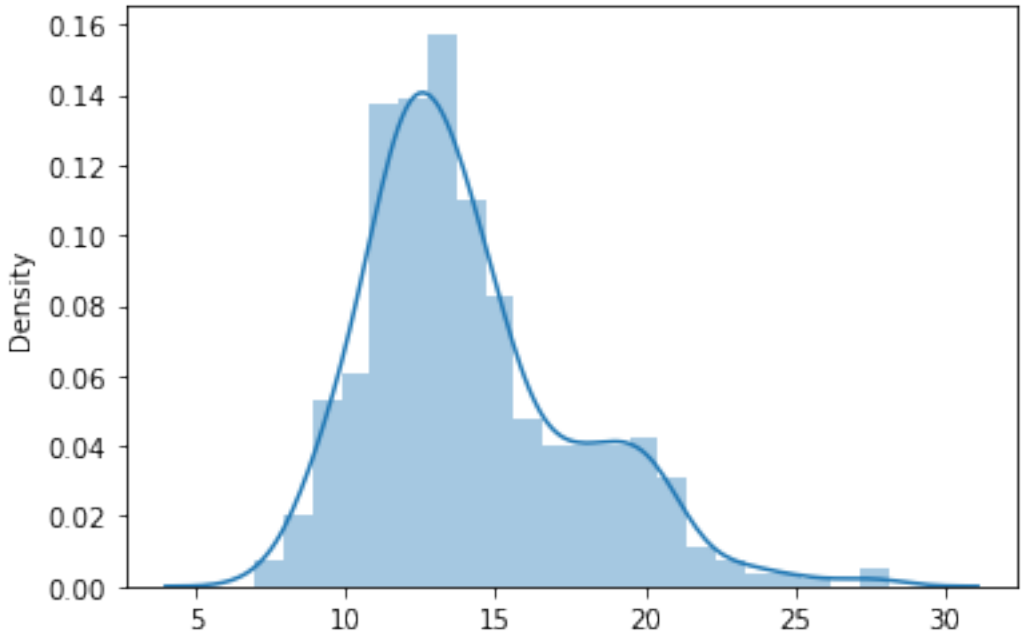

0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
count
1
fractal_dimension_worst

[23]:
```
df.shape
```

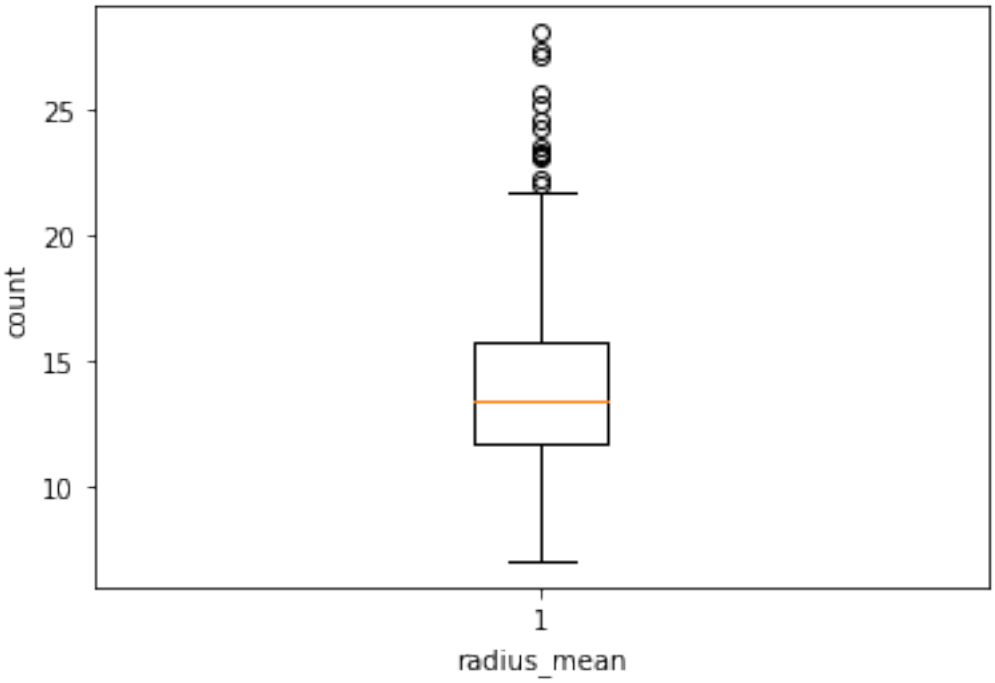
[23]: (569, 31)

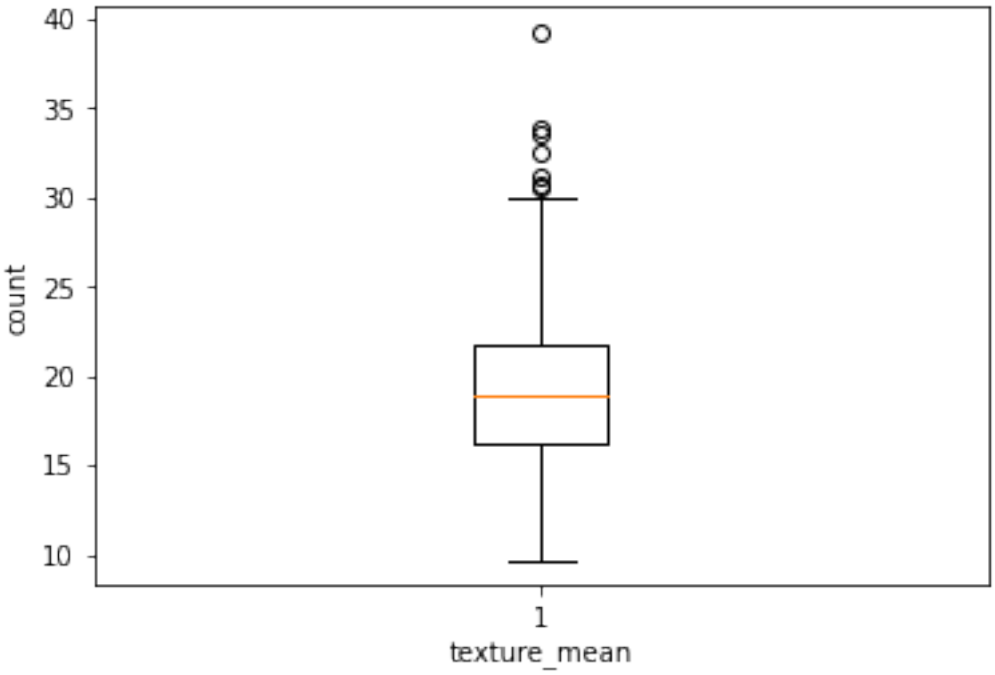
[24]:
```
sns.distplot(x=df.radius_mean)
```

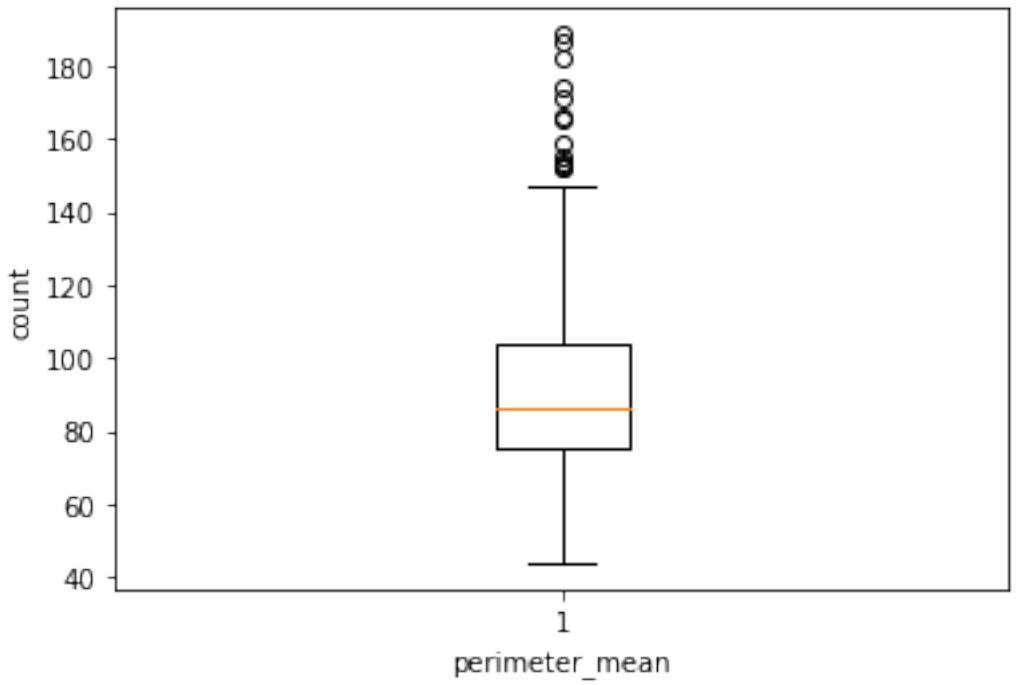
[24]: <AxesSubplot:ylabel='Density'>

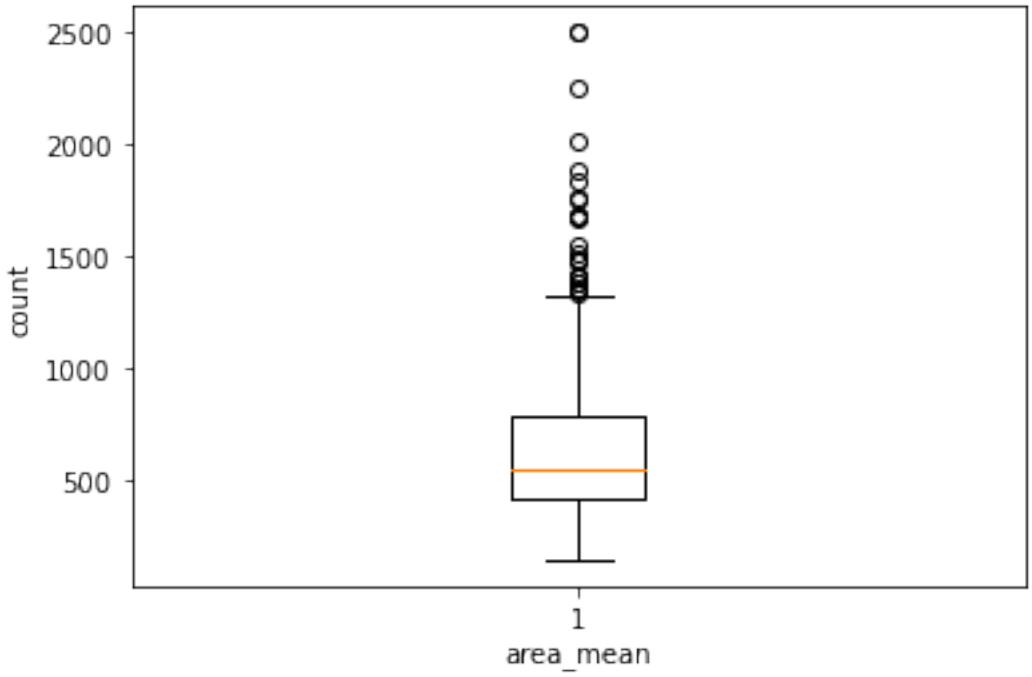


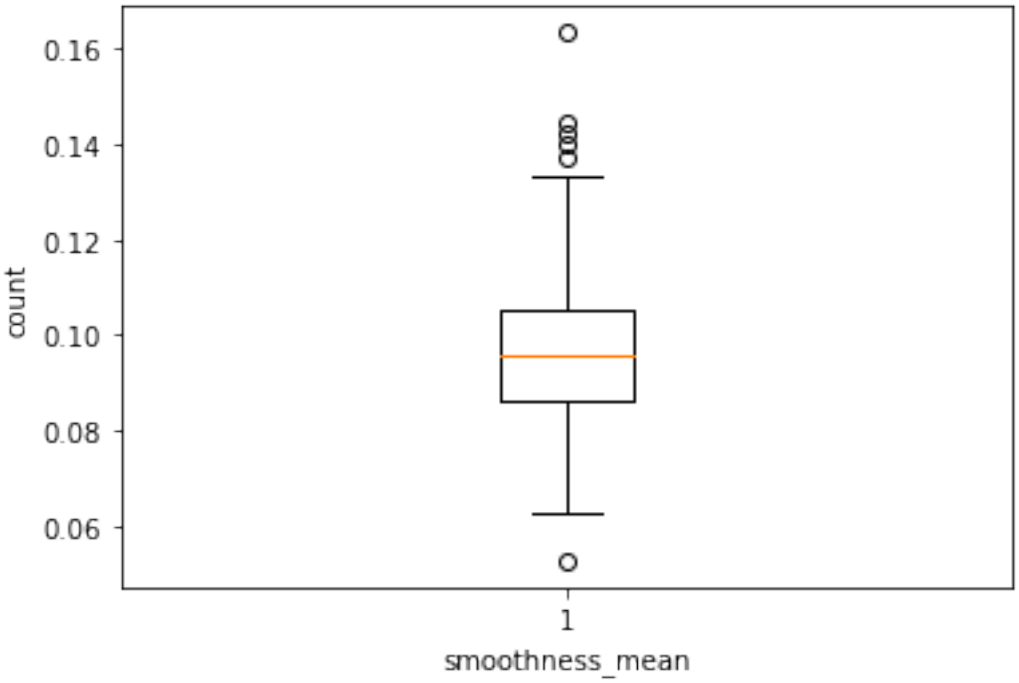
[25]:
```
for col_name in df.columns:
    if(df[col_name].dtypes=='int64' or df[col_name].dtypes=='float64'):
        plt.boxplot(df[col_name] )
        plt.xlabel(col_name)
        plt.ylabel('count')
        plt.show()
```

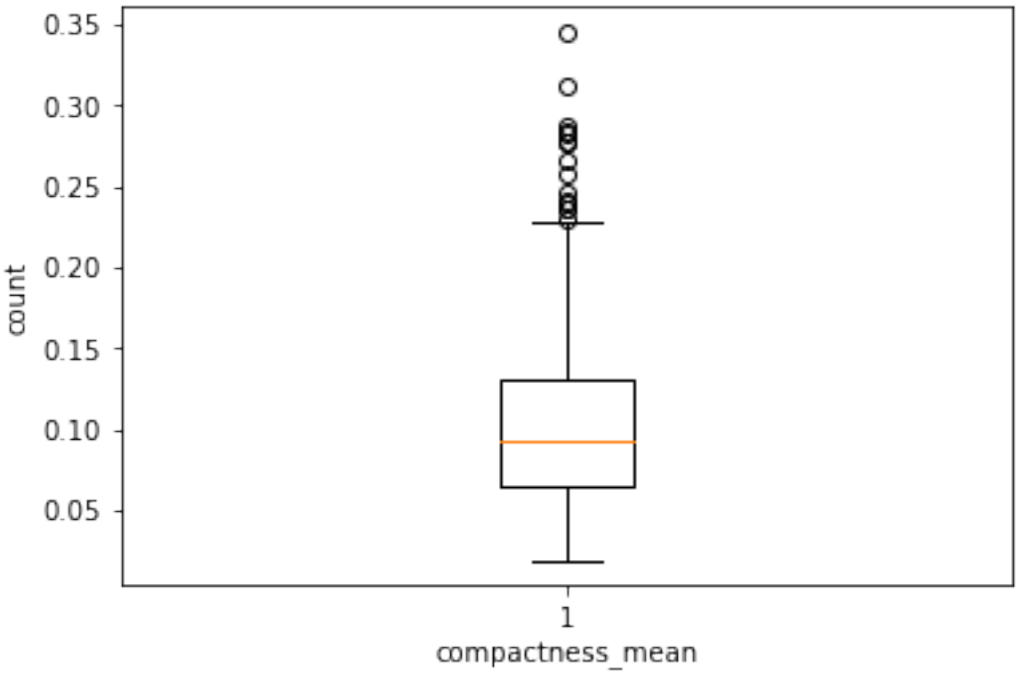

25
20
15
10
count
1
radius_mean

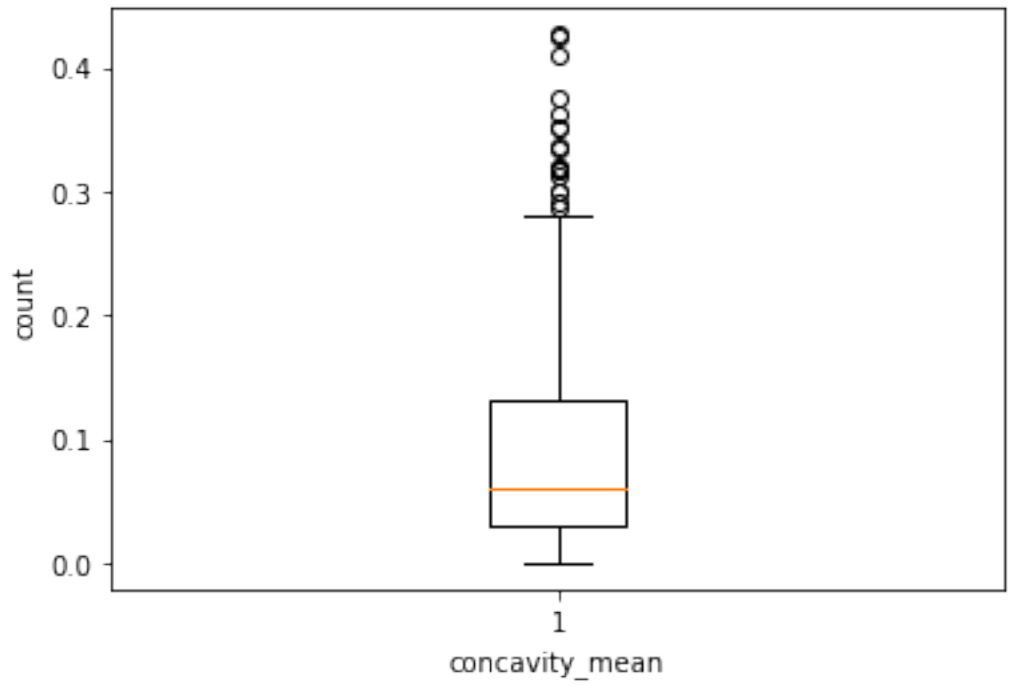

40
35
30
25
20
15
10
count
1
texture_mean

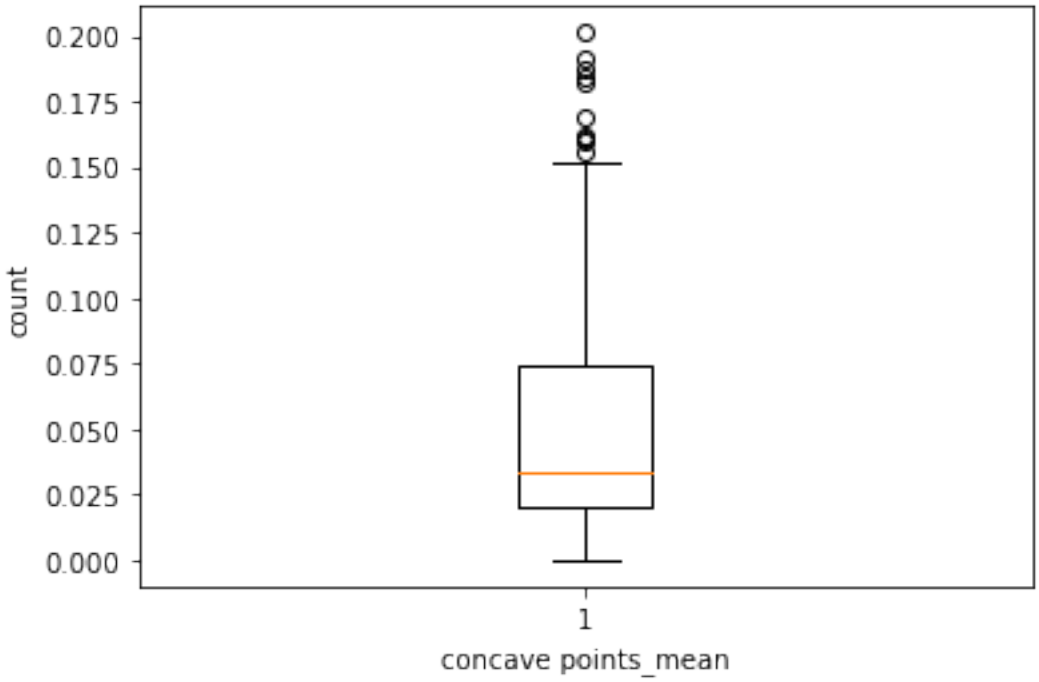

180
160
140
120
100
80
60
40
count
1
perimeter_mean

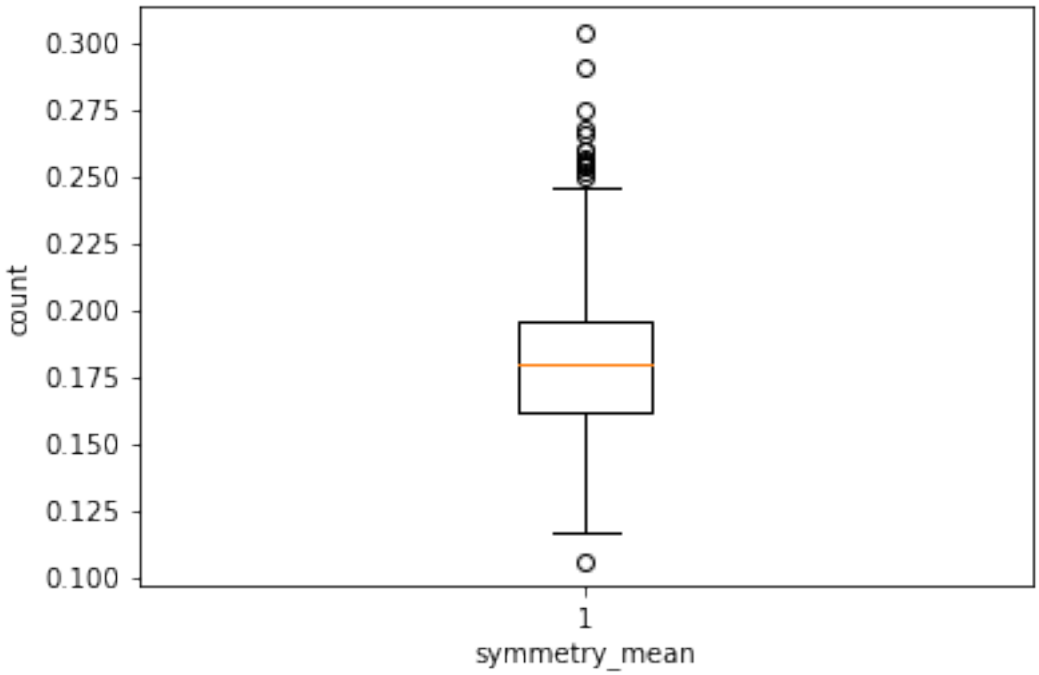

2500
2000
1500
1000
500
count
1
area_mean

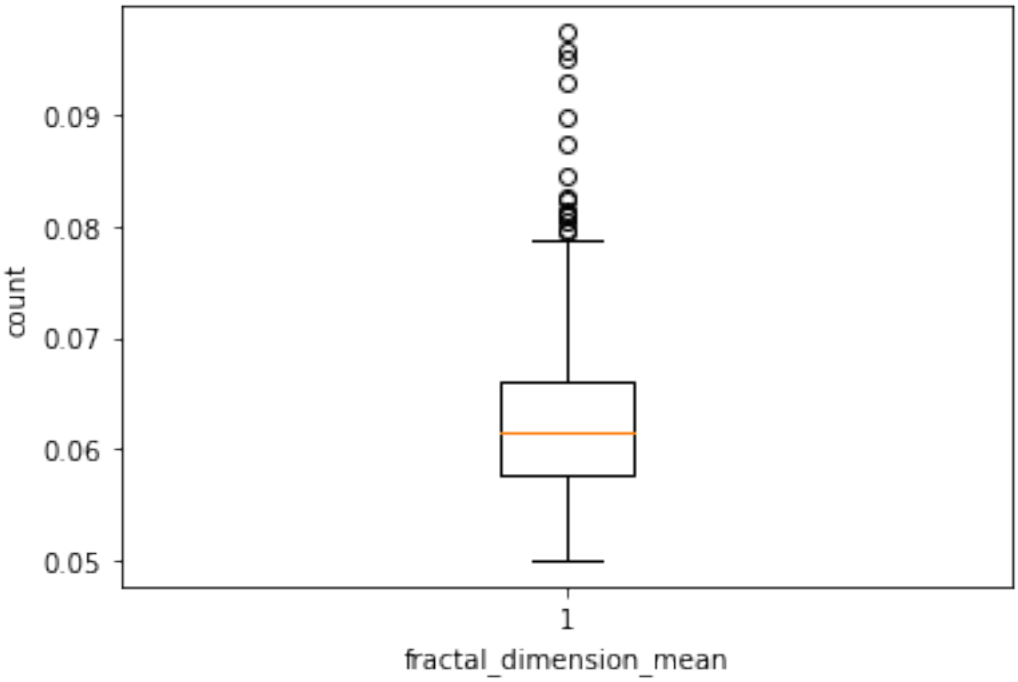

0.16
0.14
0.12
0.10
0.08
0.06
count
1
smoothness_mean

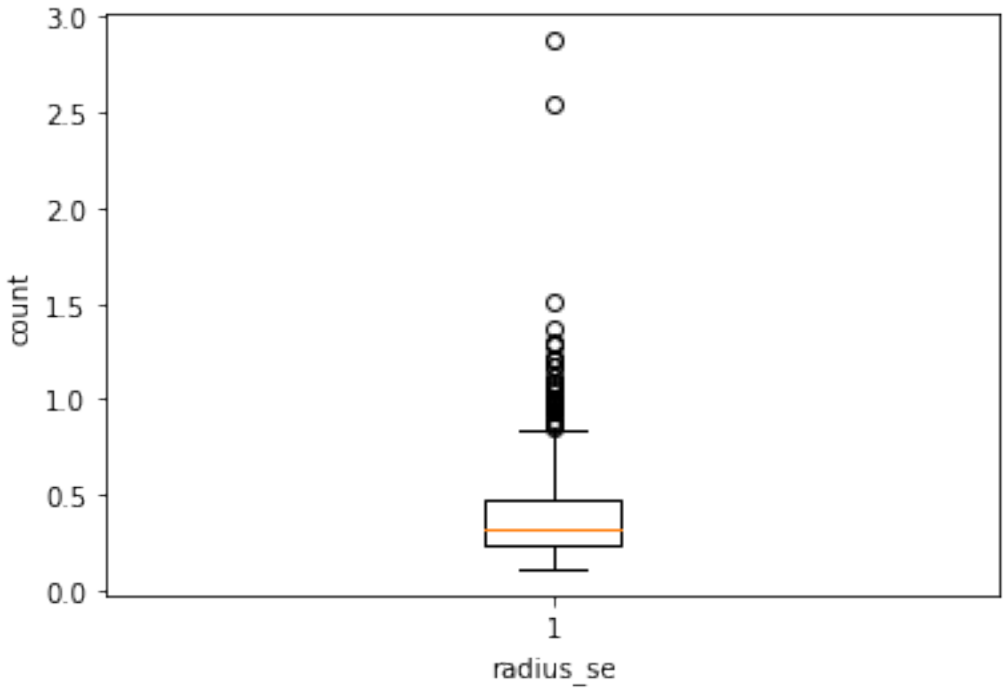

0.35
0.30
0.25
0.20
0.15
0.10
0.05
count
1
compactness_mean

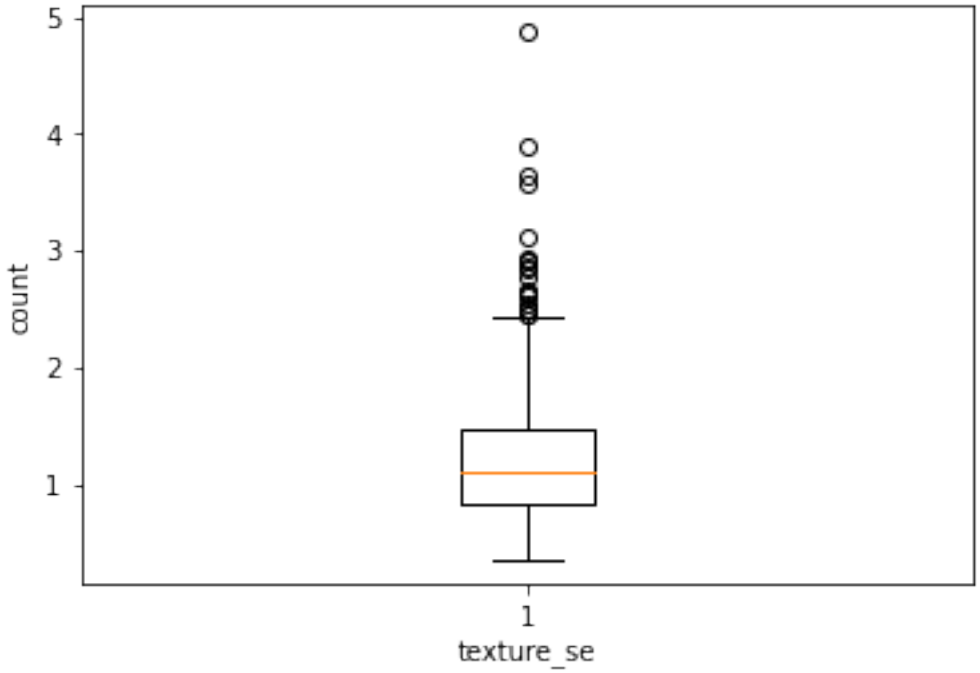

0.4
0.3
0.2
0.1
0.0
count
1
concavity_mean

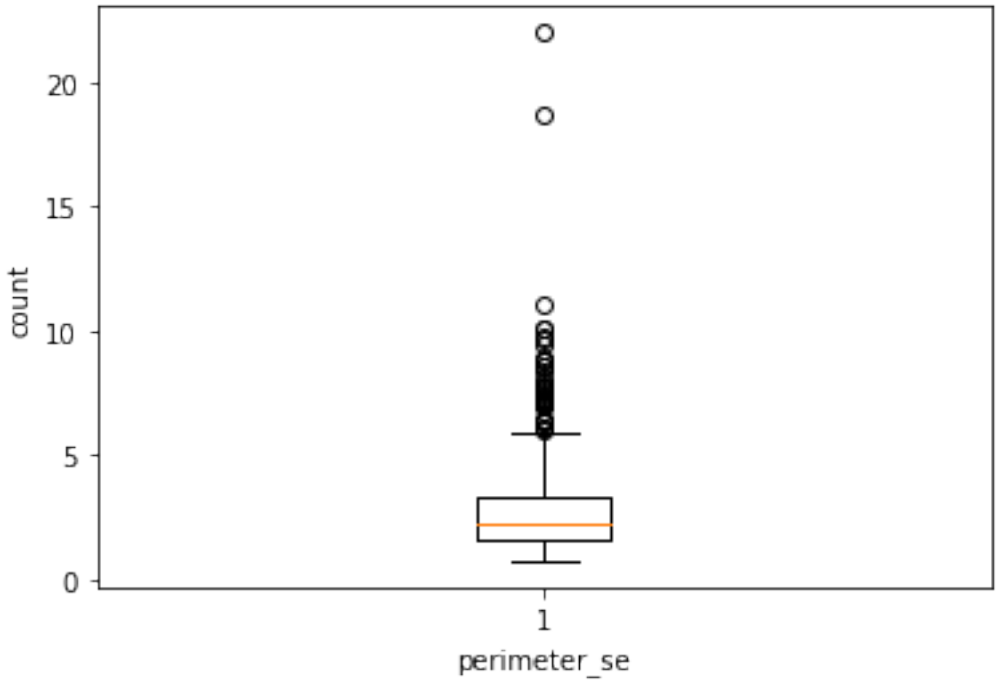

0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
count
1
concave points_mean

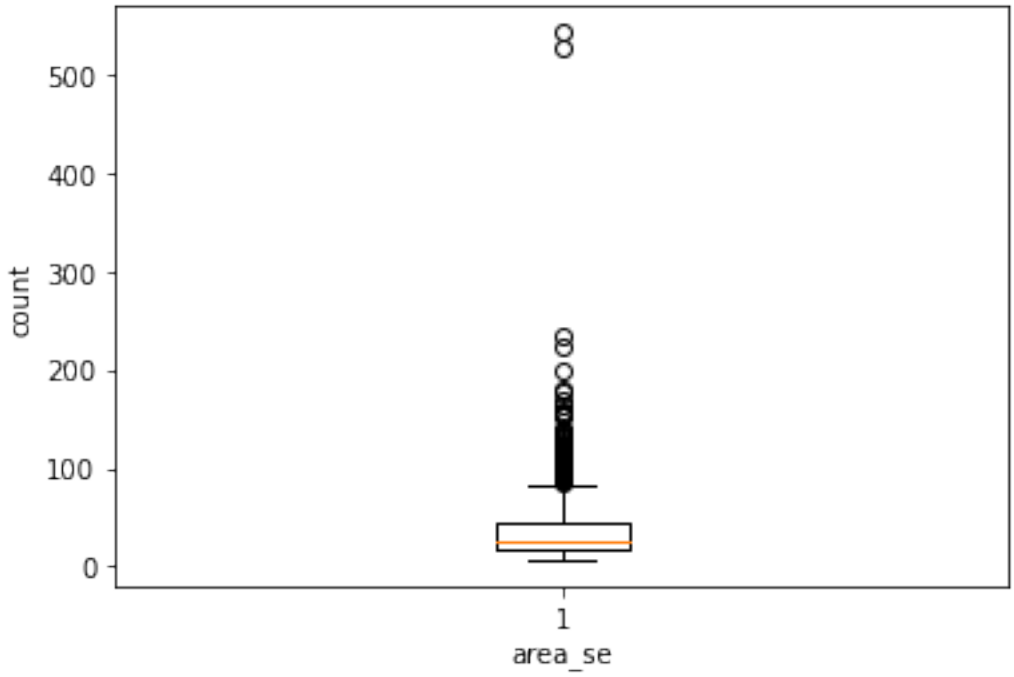

0.300
0.275
0.250
0.225
0.200
0.175
0.150
0.125
0.100
count
1
symmetry_mean

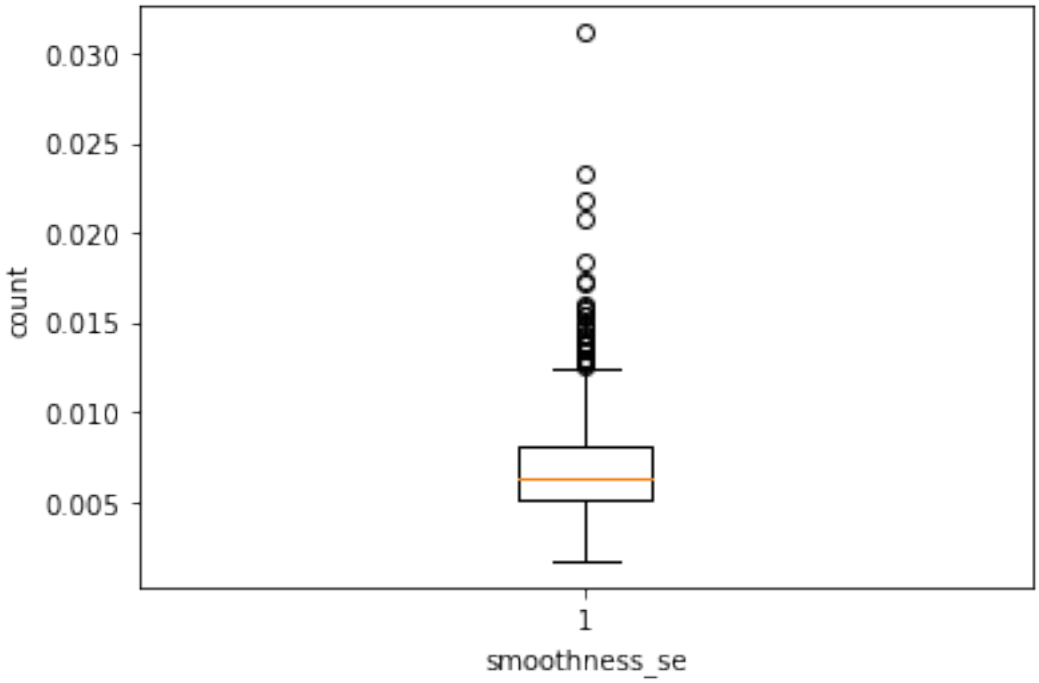

0.09
0.08
0.07
0.06
0.05
count
1
fractal_dimension_mean

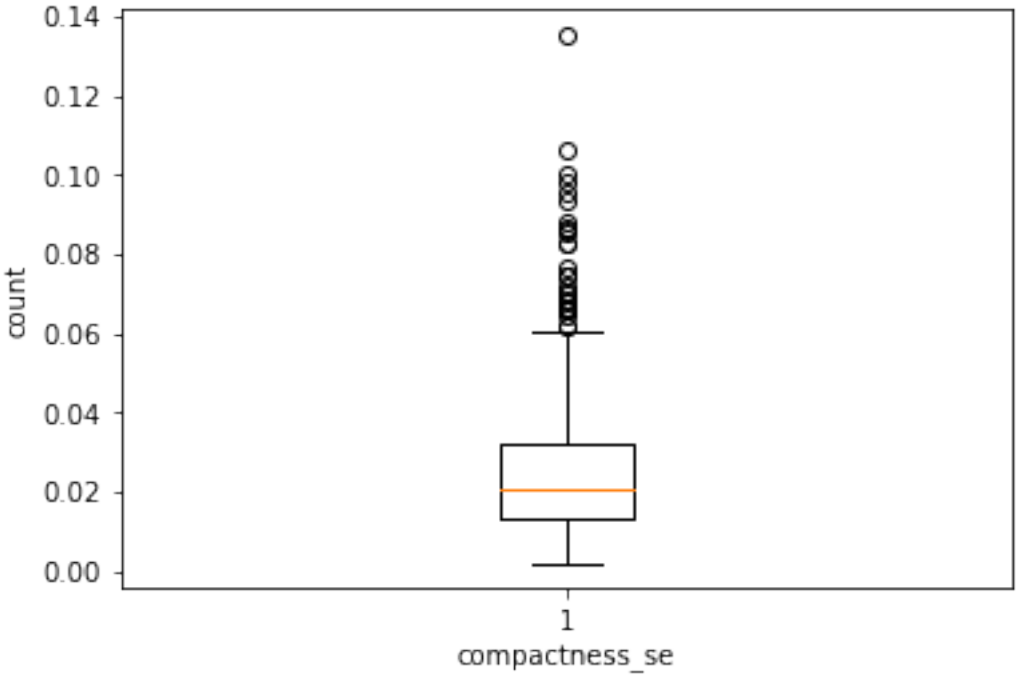

3.0
2.5
2.0
1.5
1.0
0.5
0.0
count
1
radius_se

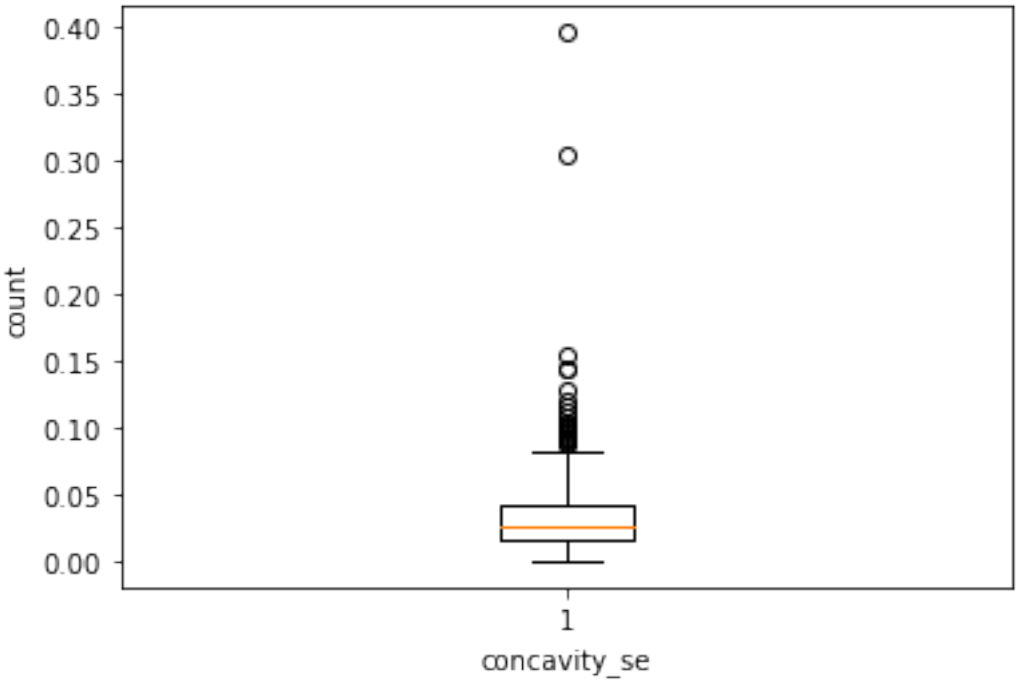

5
4
3
2
1
count
1
texture_se

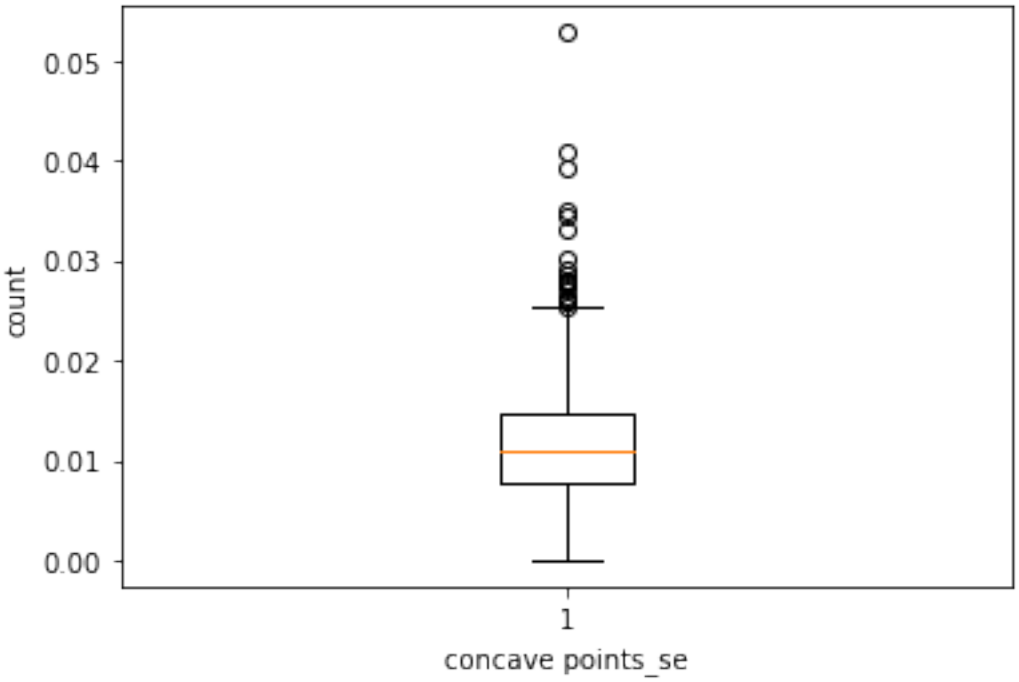

20
15
10
5
0
count
1
perimeter_se

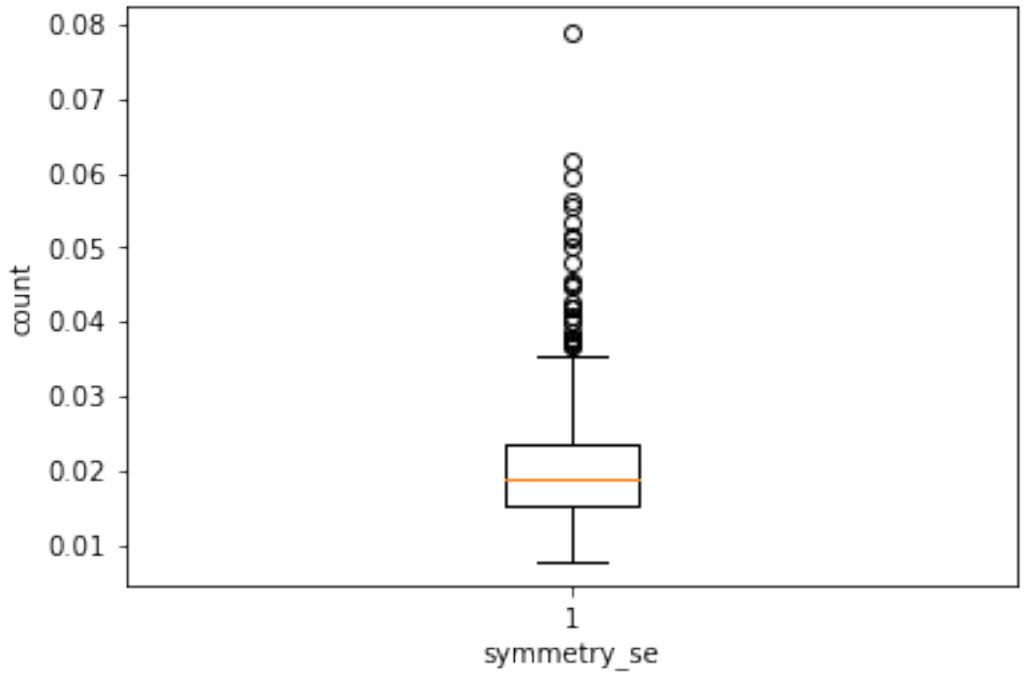

500
400
300
200
100
0
count
1
area_se

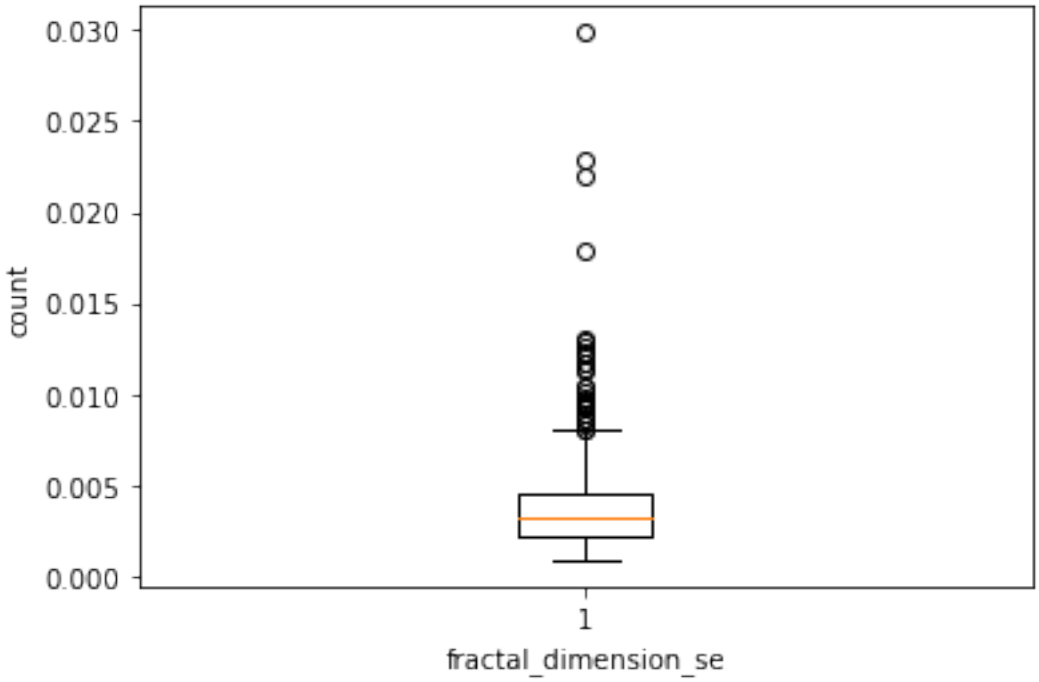

0.030
0.025
0.020
0.015
0.010
0.005
count
1
smoothness_se

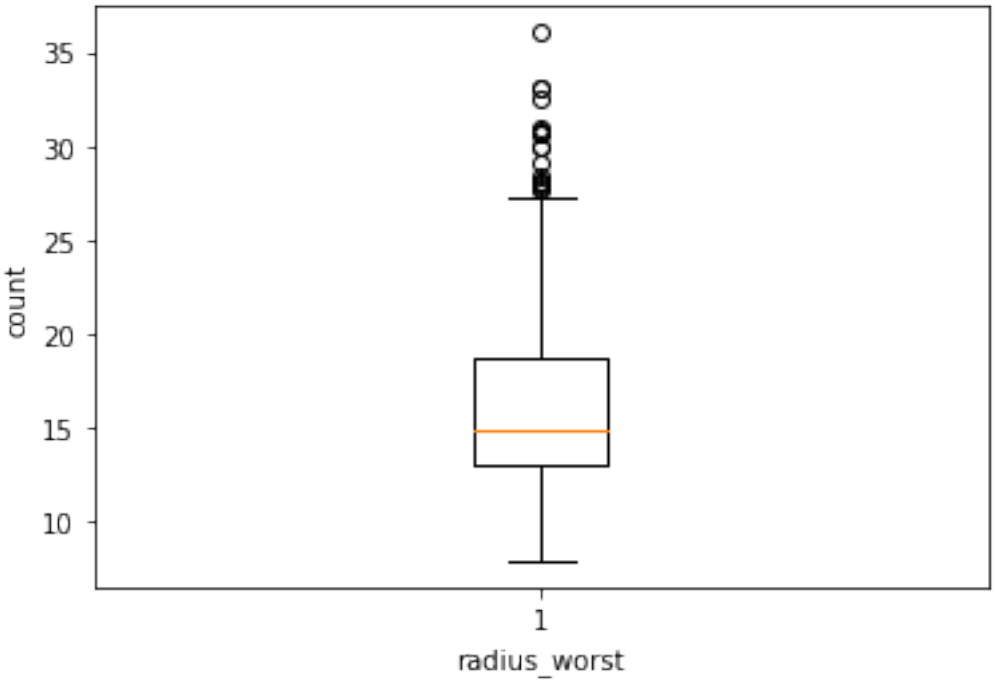

0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
count
1
compactness_se

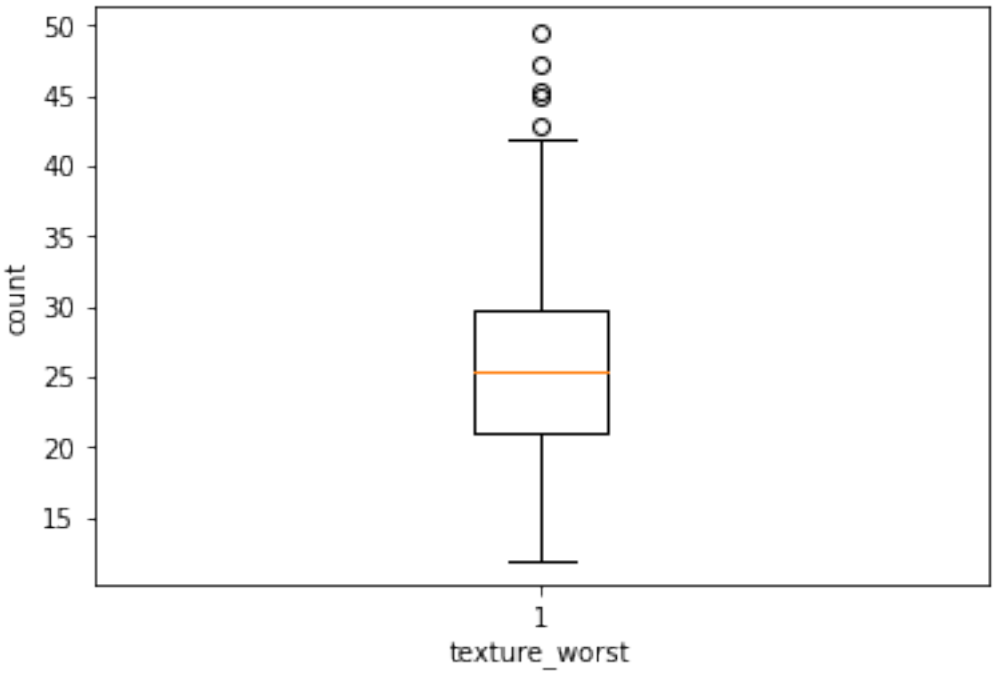

0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
count
1
concavity_se

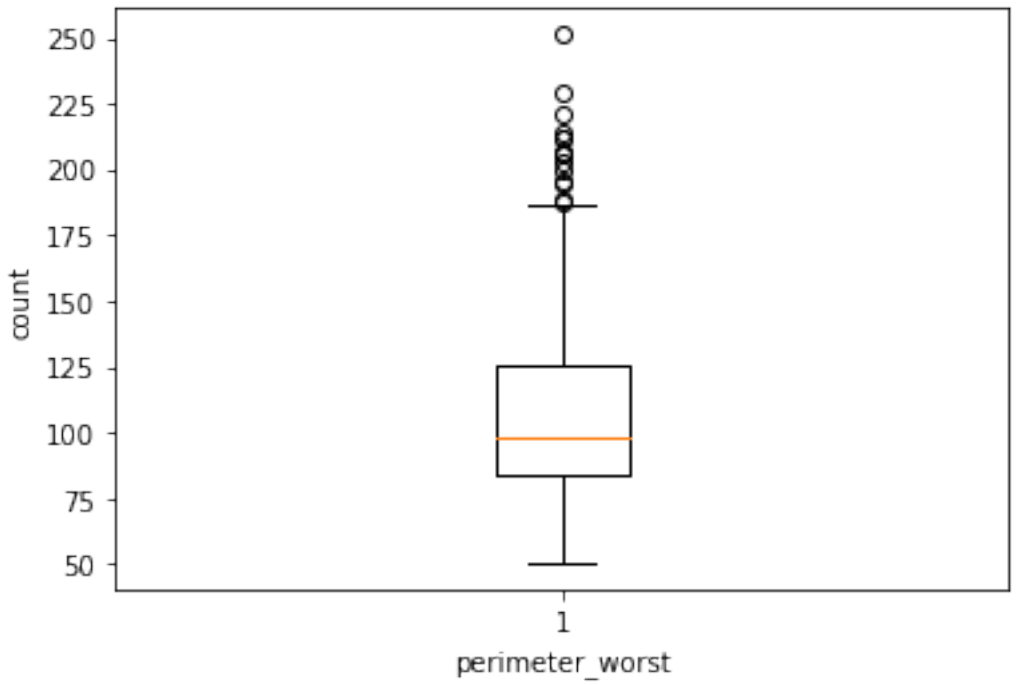

0.05
0.04
0.03
0.02
0.01
0.00
count
1
concave points_se

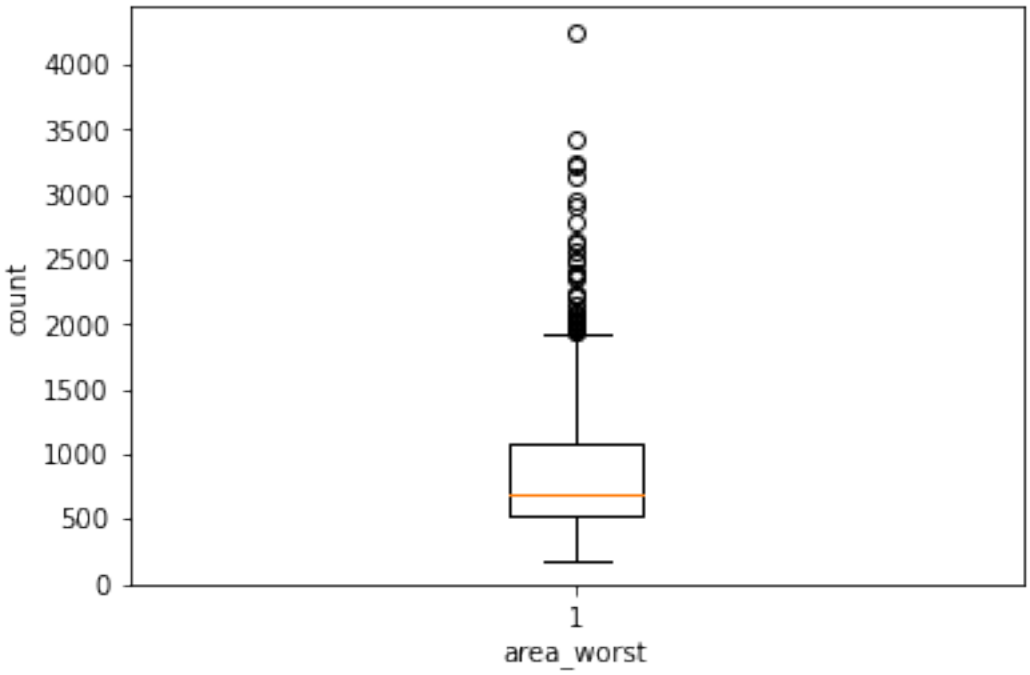

0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
count
1
symmetry_se

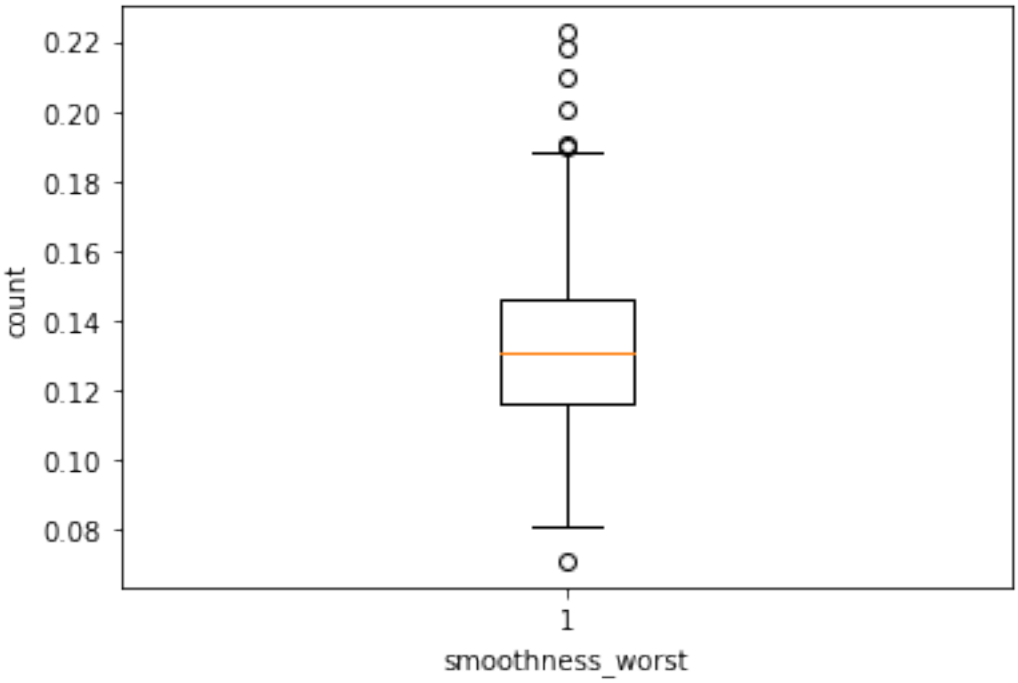

0.030
0.025
0.020
0.015
0.010
0.005
0.000
count
1
fractal_dimension_se

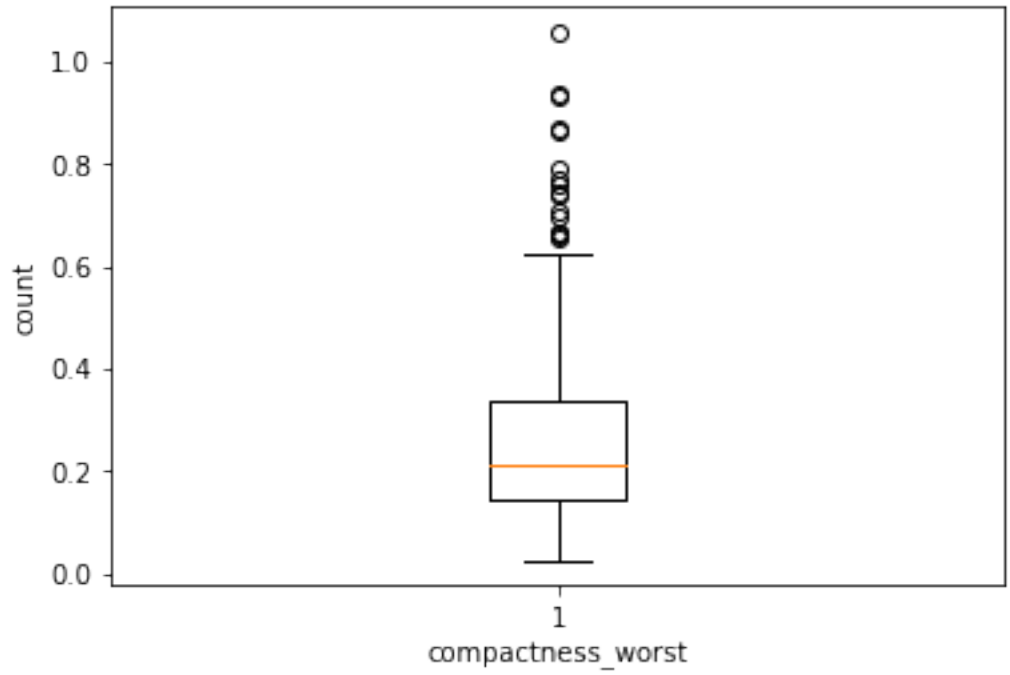

35
30
25
20
15
10
count
1
radius_worst

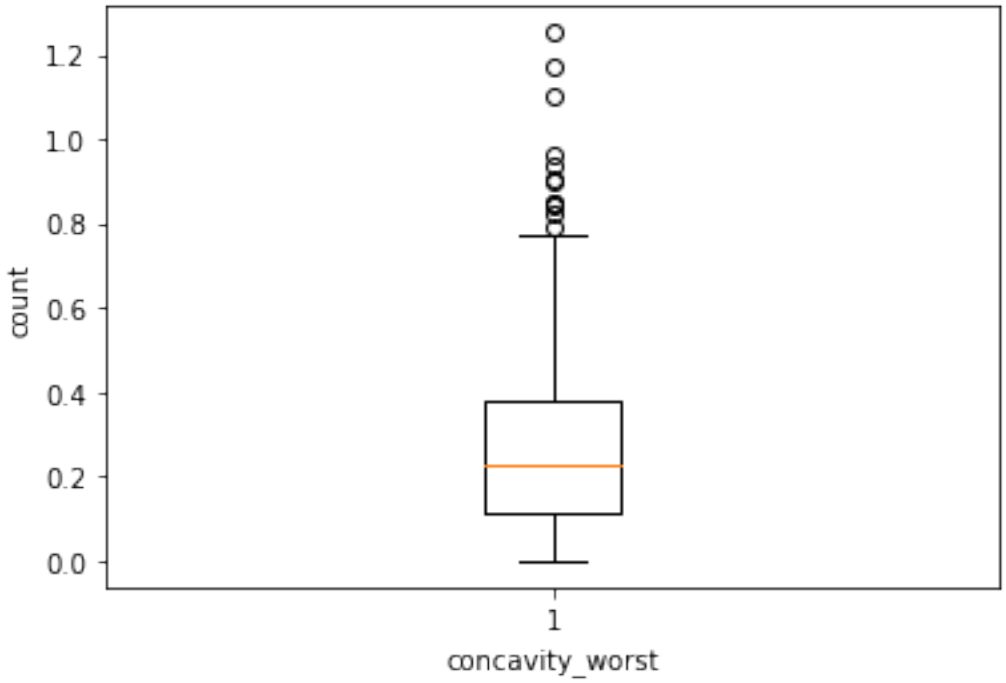

50
45
40
35
30
25
20
15
count
1
texture_worst

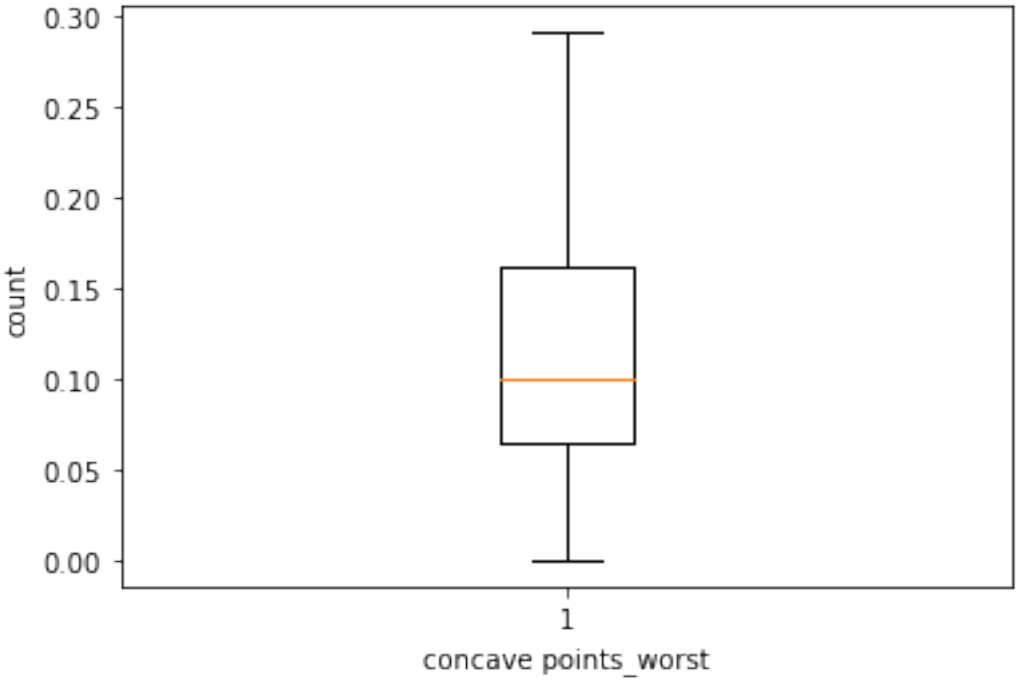

250
225
200
175
150
125
100
75
50
count
1
perimeter_worst

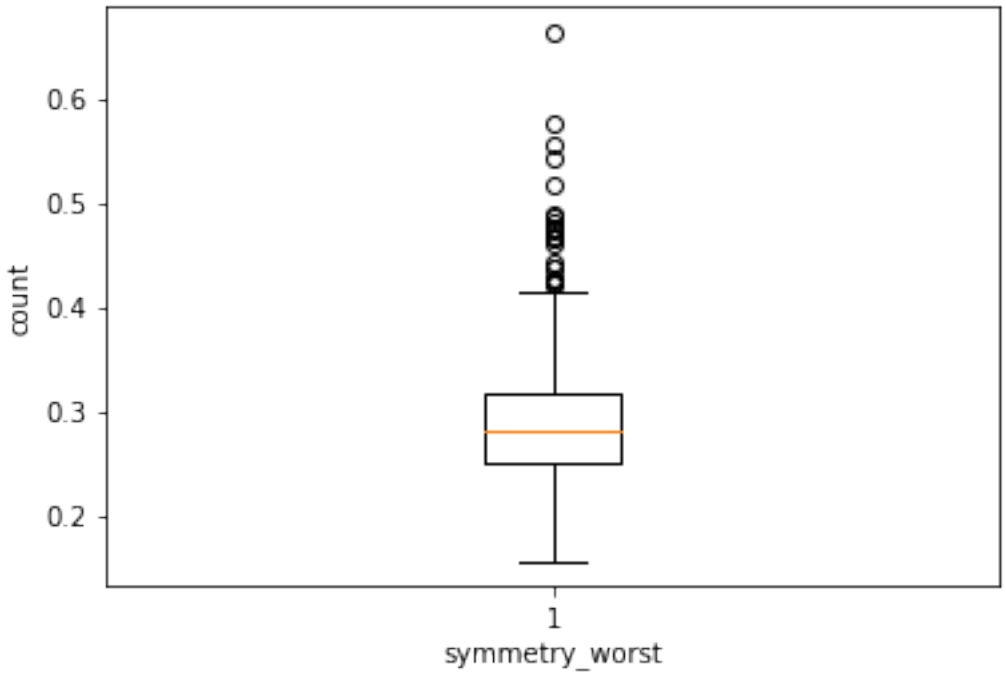

4000
3500
3000
2500
2000
1500
1000
500
0
count
1
area_worst

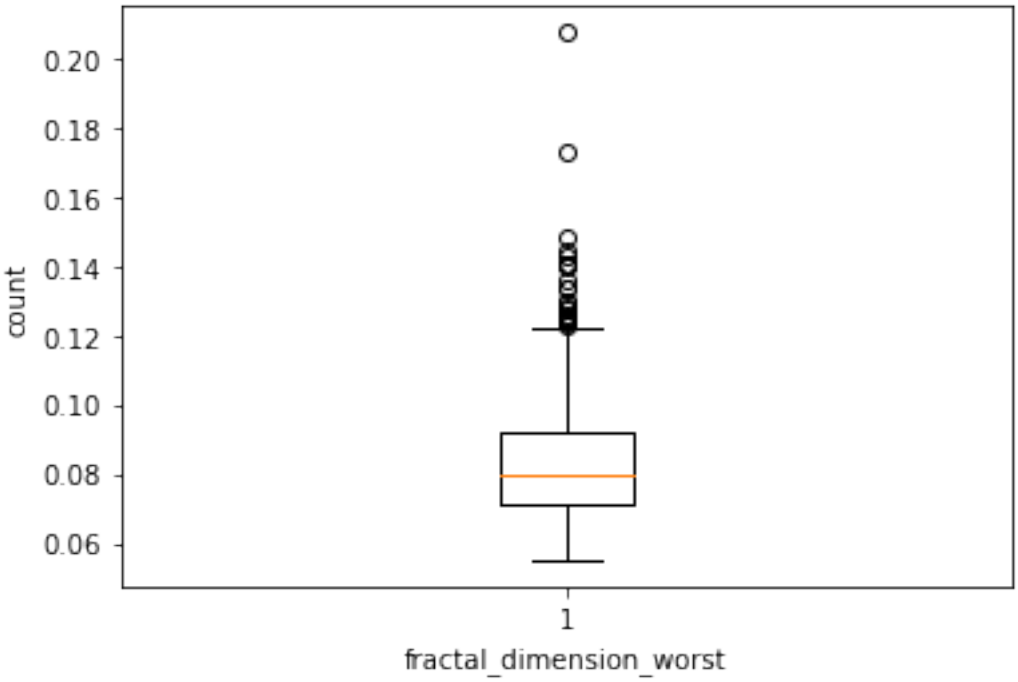

0.22
0.20
0.18
0.16
0.14
0.12
0.10
0.08
count
1
smoothness_worst


1.0
0.8
0.6
0.4
0.2
0.0
count
1
compactness_worst


1.2
1.0
0.8
0.6
0.4
0.2
0.0
count
1
concavity_worst


0.30
0.25
0.20
0.15
0.10
0.05
0.00
count
1
concave points_worst


0.6
0.5
0.4
0.3
0.2
count
1
symmetry_worst


0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
count
1
fractal_dimension_worst

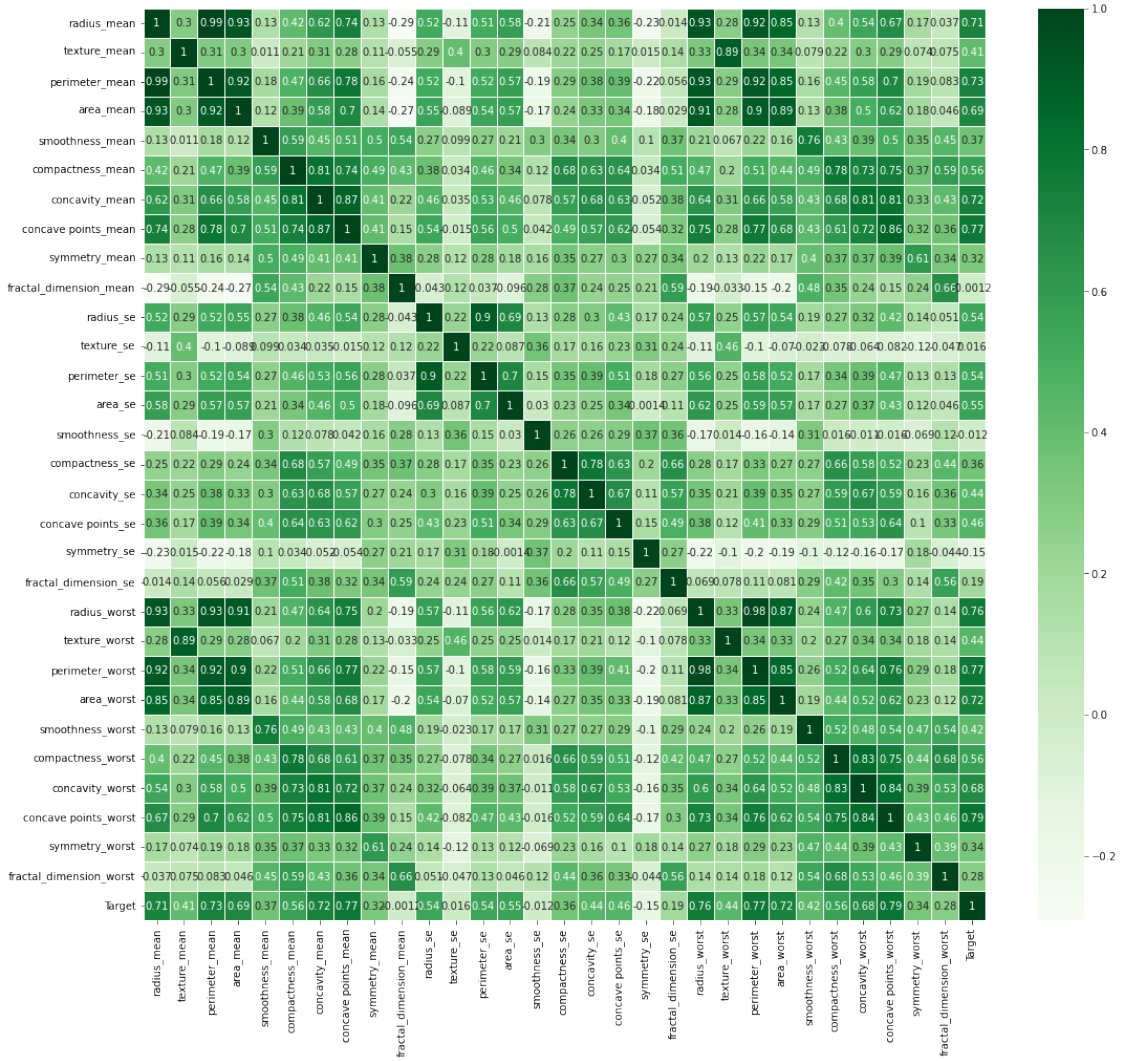
1 Get the values of Outliers

[26]:
```
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).sum()
```

[26]:
```
radius_mean               14
texture_mean               7
perimeter_mean            13
area_mean                 25
smoothness_mean            6
compactness_mean          16
concavity_mean            18
concave points_mean       10
symmetry_mean             15
fractal_dimension_mean    15
radius_se                 38
texture_se                20
perimeter_se              38
area_se                   65
smoothness_se             30
compactness_se            28
concavity_se              22
concave points_se         19
symmetry_se               27
fractal_dimension_se      28
radius_worst              17
texture_worst              5
perimeter_worst           15
area_worst                35
smoothness_worst           7
compactness_worst         16
concavity_worst           12
concave points_worst       0
symmetry_worst            23
fractal_dimension_worst   24
Target                     0
dtype: int64
```

[45]:
```
plt.figure(figsize=(18,16))
sns.heatmap(df.corr(),annot=True,linewidth=.5,cmap='Greens')
```

[45]:
```
<AxesSubplot:>
```



[43]:
```
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).sum()
```

[43]:
```
radius_mean                6
texture_mean               2
perimeter_mean             4
area_mean                 40
smoothness_mean            1
compactness_mean           6
concavity_mean             8
concave points_mean        9
symmetry_mean              1
```

```
fractal_dimension_mean      7
radius_se                  28
texture_se                  5
perimeter_se               21
area_se                    52
smoothness_se               9
compactness_se              9
concavity_se               13
concave points_se           6
symmetry_se                14
fractal_dimension_se       13
radius_worst               21
texture_worst               0
perimeter_worst            13
area_worst                 48
smoothness_worst            1
compactness_worst          12
concavity_worst             2
concave points_worst        0
symmetry_worst             11
fractal_dimension_worst    11
Target                      0
dtype: int64
```

2 Outlier Treatment

[41]:
```
def outlier_detect(df):
    for i in df.describe().columns:
        Q1=df.describe().at['25%',i]
        Q3=df.describe().at['75%',i]
        IQR=Q3 - Q1
        LTV=Q1 - 1.5 * IQR
        UTV=Q3 + 1.5 * IQR
        x=np.array(df[i])
        p=[]
        for j in x:
            if j < LTV or j>UTV:
                p.append(df[i].median())
            else:
                p.append(j)
        df[i]=p
    return df
```

[30]:
```
outlier_detect(df)
```

[30]:

	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	\
0	17.99	10.38	122.80	1001.0	0.11840	
1	20.57	17.77	132.90	1326.0	0.08474	
2	19.69	21.25	130.00	1203.0	0.10960	
3	11.42	20.38	77.58	386.1	0.09587	
4	20.29	14.34	135.10	1297.0	0.10030	
..	…	…	…	…	…	
564	21.56	22.39	142.00	551.1	0.11100	
565	20.13	28.25	131.20	1261.0	0.09780	
566	16.60	28.08	108.30	858.1	0.08455	
567	20.60	29.33	140.10	1265.0	0.11780	
568	7.76	24.54	47.92	181.0	0.09587	

	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	\
0	0.09263	0.06154	0.14710	0.2419	
1	0.07864	0.08690	0.07017	0.1812	
2	0.15990	0.19740	0.12790	0.2069	
3	0.09263	0.24140	0.10520	0.1792	
4	0.13280	0.19800	0.10430	0.1809	
..	…	…	…	…	
564	0.11590	0.24390	0.13890	0.1726	
565	0.10340	0.14400	0.09791	0.1752	
566	0.10230	0.09251	0.05302	0.1590	
567	0.09263	0.06154	0.15200	0.2397	
568	0.04362	0.00000	0.00000	0.1587	

	fractal_dimension_mean	…	texture_worst	perimeter_worst	area_worst	\
0	0.07871	…	17.33	184.60	686.5	
1	0.05667	…	23.41	158.80	686.5	
2	0.05999	…	25.53	152.50	1709.0	
3	0.06154	…	26.50	98.87	567.7	
4	0.05883	…	16.67	152.20	1575.0	
..	…	…	…	…	…	
564	0.05623	…	26.40	166.10	686.5	
565	0.05533	…	38.25	155.00	1731.0	
566	0.05648	…	34.12	126.70	1124.0	
567	0.07016	…	39.42	184.60	1821.0	
568	0.05884	…	30.37	59.16	268.6	

	smoothness_worst	compactness_worst	concavity_worst	\
0	0.16220	0.21190	0.7119	
1	0.12380	0.18660	0.2416	
2	0.14440	0.42450	0.4504	
3	0.13130	0.21190	0.6869	
4	0.13740	0.20500	0.4000	
..	…	…	…	
564	0.14100	0.21130	0.4107	

```
565           0.11660           0.19220           0.3215
566           0.11390           0.30940           0.3403
567           0.16500           0.21190           0.2267
568           0.08996           0.06444           0.0000

     concave points_worst  symmetry_worst  fractal_dimension_worst  Target
0                  0.2654          0.2822                  0.11890       1
1                  0.1860          0.2750                  0.08902       1
2                  0.2430          0.3613                  0.08758       1
3                  0.2575          0.2822                  0.08004       1
4                  0.1625          0.2364                  0.07678       1
..                    …               …                        …      …
564                0.2216          0.2060                  0.07115       1
565                0.1628          0.2572                  0.06637       1
566                0.1418          0.2218                  0.07820       1
567                0.2650          0.4087                  0.08004       1
568                0.0000          0.2871                  0.07039       0

[569 rows x 31 columns]
```

3 Spliting the data

[32]:
```
X= df.drop(columns='Target', axis=1)
Y=df['Target']
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test=train_test_split(X,Y, test_size=20,␣
 ↪random_state=2)
```

4 Model Training

[33]:
```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
```

[34]:
```
model.fit(X_train,Y_train)
```

[34]:
```
LogisticRegression()
```

5 Model Evaluation

[35]:
```
#accuracy of training data
from sklearn.metrics import accuracy_score
x_train_prediction= model.predict(X_train)
train_accuracy=accuracy_score(Y_train,x_train_prediction )
```

[36]:
```
print('training accuracy =',train_accuracy*100 )
```

```
training accuracy = 91.07468123861567
```

[37]:
```
#accuracy of test data
from sklearn.metrics import accuracy_score
x_test_prediction= model.predict(X_test)
test_accuracy=accuracy_score(Y_test,x_test_prediction )
```

[38]:
```
print('testing accuracy =',test_accuracy*100 )
```

```
testing accuracy = 85.0
```

6 building prection system

[40]:
```
input_value= (17.99,10.38,122.8,1001,0.1184,0.2776,0.3001,0.1471,0.2419,0.
↪07871,1.095,0.9053,8.589,153.4,0.006399,0.04904,0.05373,0.01587,0.03003,0.
↪006193,25.38,17.33,184.6,2019,0.1622,0.6656,0.7119,0.2654,0.4601,0.1189)

new_input=np.asarray(input_value)
#reshaping
reshape_input= new_input.reshape(1,-1)
prediction=model.predict(reshape_input)
print(prediction)
if (prediction[0]==1):
    print("the brist cancer is malignant")
else:
    print("the brist cancer is benign")
```

```
[1]
the brist cancer is malignant
```

[ ]:
```

```