

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix, classification_report, roc_auc_score, roc_c
```

```
df=pd.read_csv("/content/data.csv")
```

```
df
```



```
df.tail()
```



	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothn
564	926424	M	21.56	22.39	142.00	1479.0	
565	926682	M	20.13	28.25	131.20	1261.0	
566	926954	M	16.60	28.08	108.30	858.1	
567	927241	M	20.60	29.33	140.10	1265.0	
568	92751	B	7.76	24.54	47.92	181.0	

5 rows × 33 columns

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
```

```
df.describe()
```



	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0
max	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	0

8 rows × 32 columns

```
df.shape
```



```
(569, 33)
```

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



```
np.int64(569)
```

```
for col in df.select_dtypes(include=['number']).columns:
    df[col].fillna(df[col].mean(), inplace=True)
```



```
/tmp/ipython-input-2823199234.py:2: FutureWarning: A value is trying to be set on a Categorical column which is currently all NaN. The behavior will change in pandas 3.0. This inplace method will never work because t
```

```
For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({
```

```
df[col].fillna(df[col].mean(), inplace=True)
```

```
for col in df.select_dtypes(include=['object']).columns:
    df[col].fillna(df[col].mode()[0], inplace=True)
```



```
/tmp/ipython-input-4072275310.py:2: FutureWarning: A value is trying to be set on a Categorical column which is currently all NaN. The behavior will change in pandas 3.0. This inplace method will never work because t
```

```
For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({
```

```
df[col].fillna(df[col].mode()[0], inplace=True)
```

```
df = df.drop(columns=['Unnamed: 32'])
```

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

```
⇒ np.int64(0)
```

```
df.duplicated().sum()
```

```
⇒ np.int64(0)
```

```
num_cols = df.select_dtypes(include=['number']).columns.tolist()
```

```
print(df[num_cols].describe())
```

```
for col in num_cols:
```

```
    plt.figure(figsize=(6, 4))
```

```
    sns.histplot(df[col], kde=True, bins=30)
```

```
    plt.title(f"Distribution of {col}", fontsize=14)
```

```
    plt.xlabel(col)
```

```
    plt.ylabel("Count")
```

```
    plt.tight_layout()
```

```
    plt.show()
```



	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	...	radius_worst	texture_worst	perimeter_worst	\
count	569.000000	...	569.000000	569.000000	569.000000	
mean	0.181162	...	16.269190	25.677223	107.261213	
std	0.027414	...	4.833242	6.146258	33.602542	
min	0.106000	...	7.930000	12.020000	50.410000	
25%	0.161900	...	13.010000	21.080000	84.110000	
50%	0.179200	...	14.970000	25.410000	97.660000	
75%	0.195700	...	18.790000	29.720000	125.400000	
max	0.304000	...	36.040000	49.540000	251.200000	

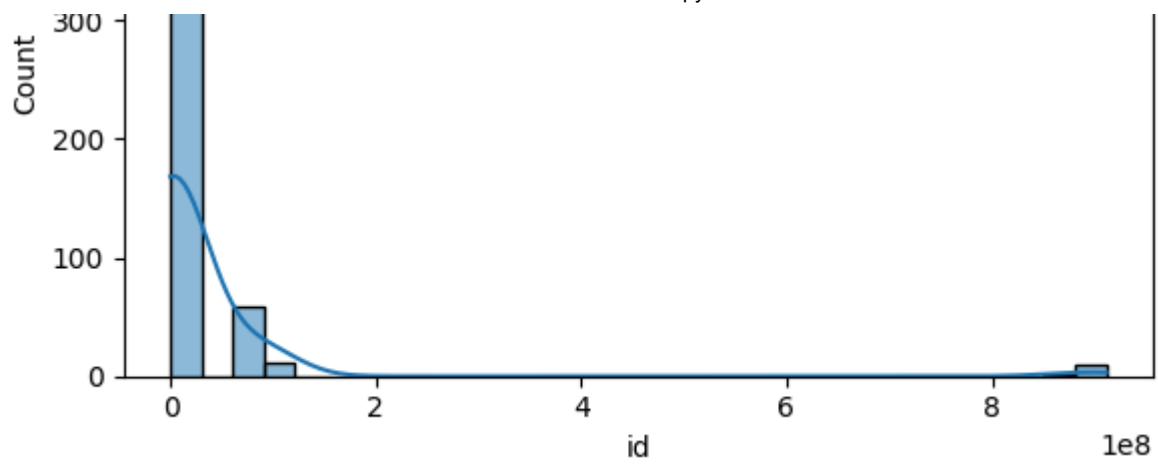
	area_worst	smoothness_worst	compactness_worst	concavity_worst	\
count	569.000000	569.000000	569.000000	569.000000	
mean	880.583128	0.132369	0.254265	0.272188	
std	569.356993	0.022832	0.157336	0.208624	
min	185.200000	0.071170	0.027290	0.000000	
25%	515.300000	0.116600	0.147200	0.114500	
50%	686.500000	0.131300	0.211900	0.226700	
75%	1084.000000	0.146000	0.339100	0.382900	
max	4254.000000	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

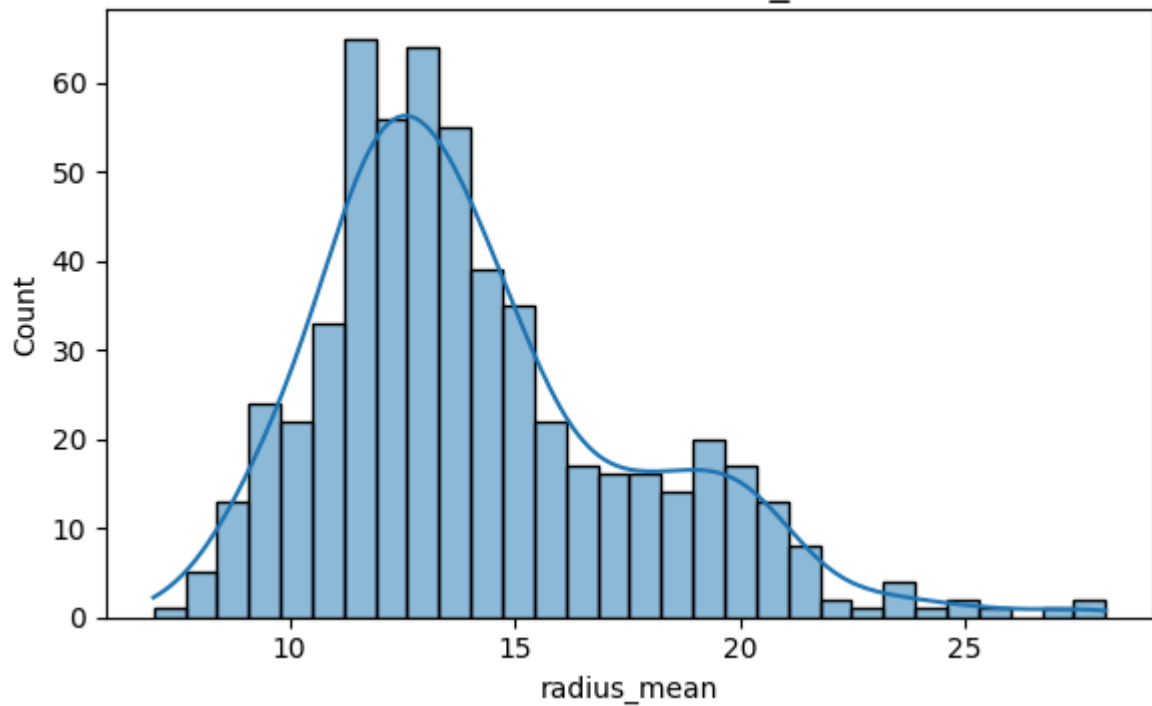
[8 rows x 31 columns]

Distribution of id

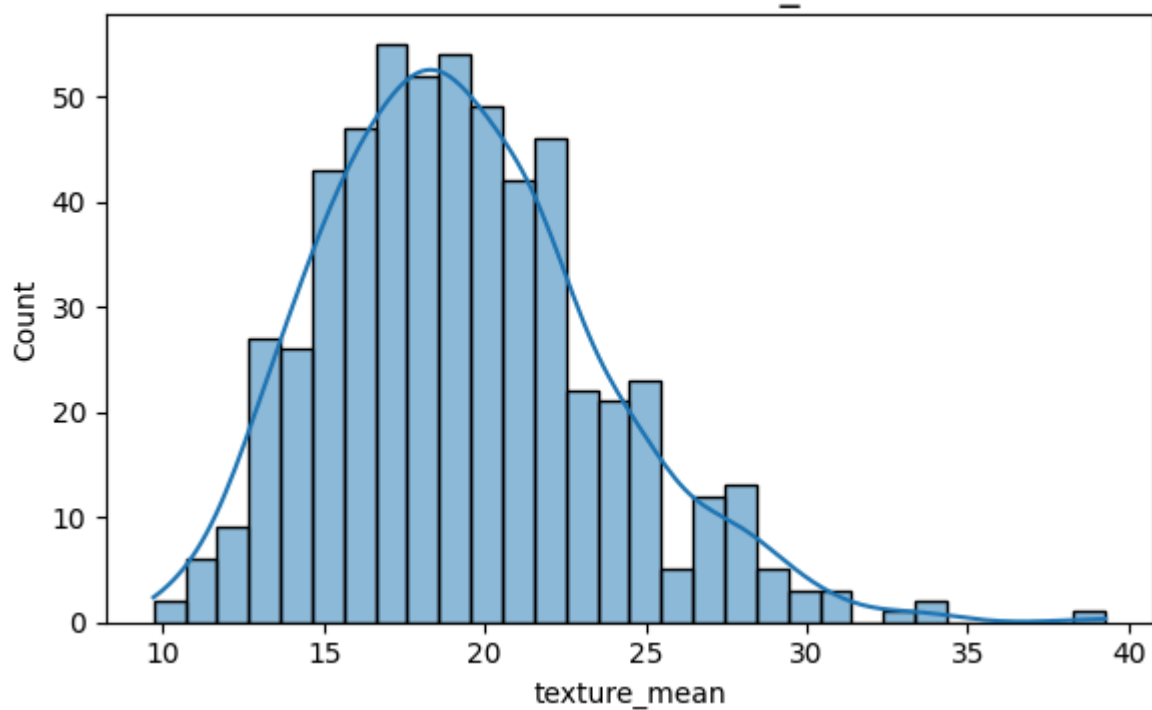




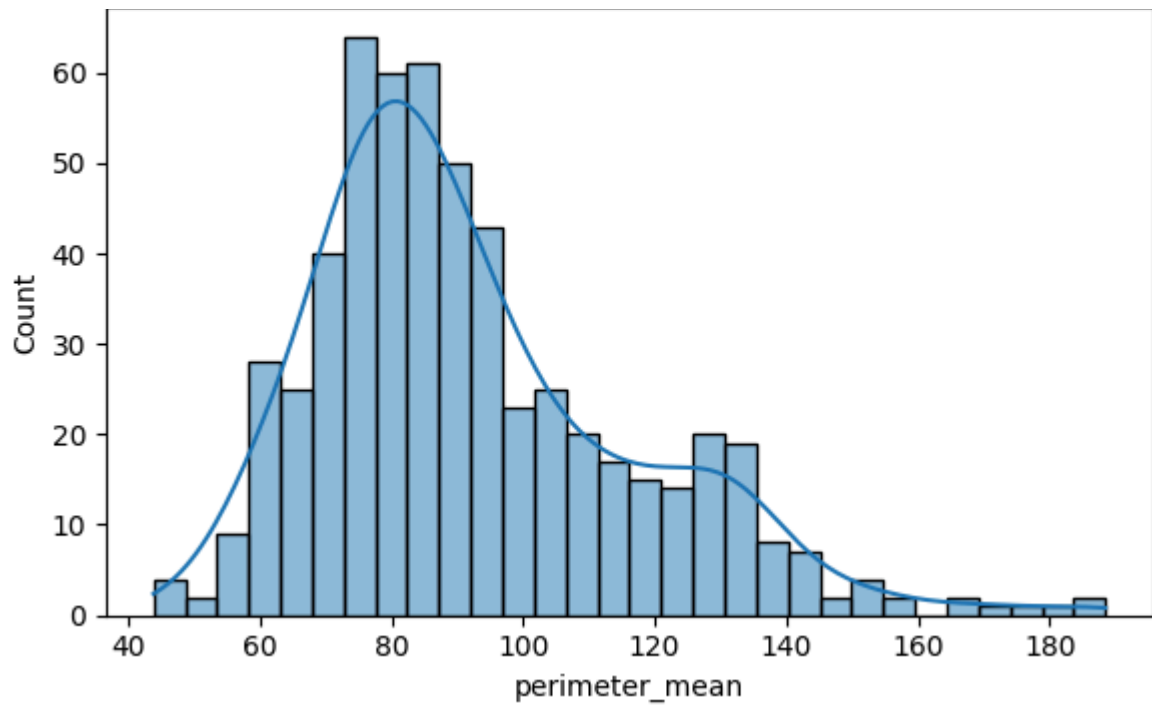
Distribution of radius_mean



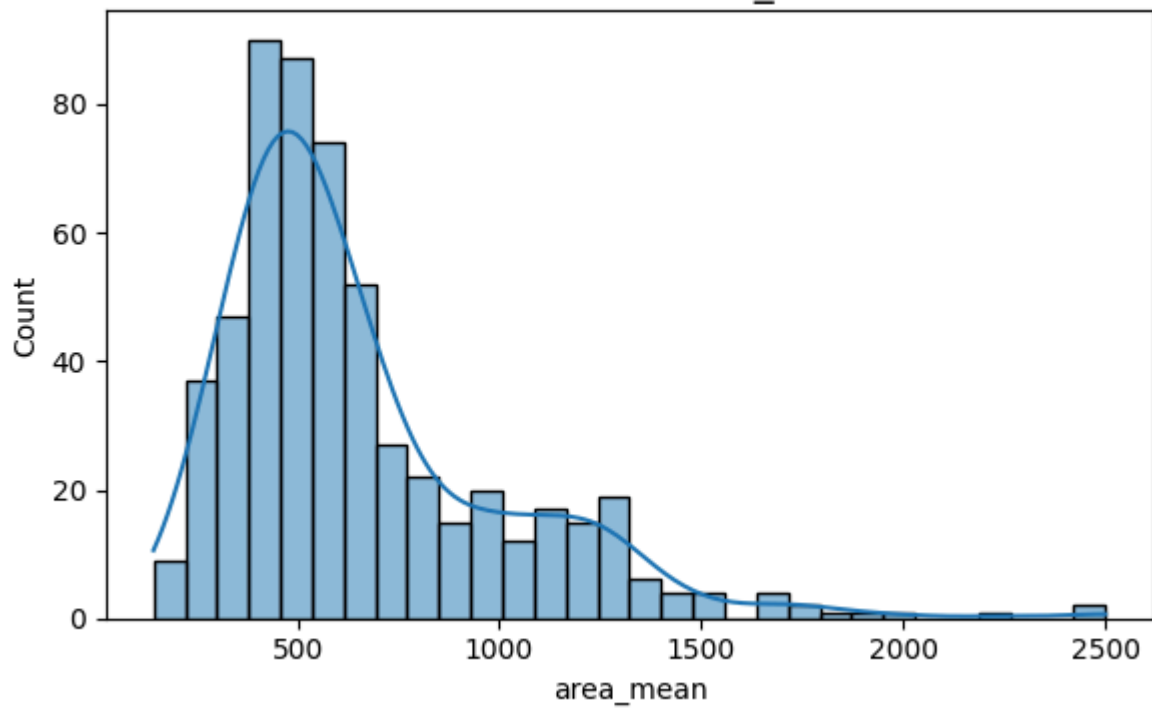
Distribution of texture_mean



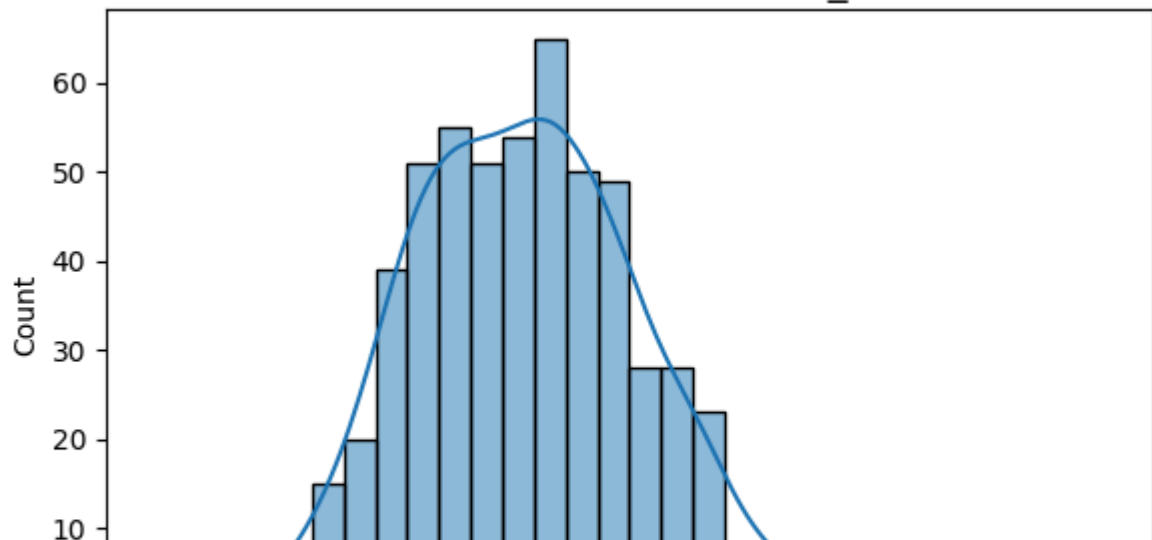
Distribution of perimeter_mean

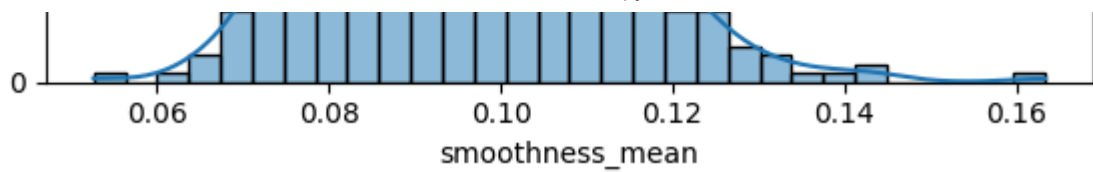


Distribution of area_mean

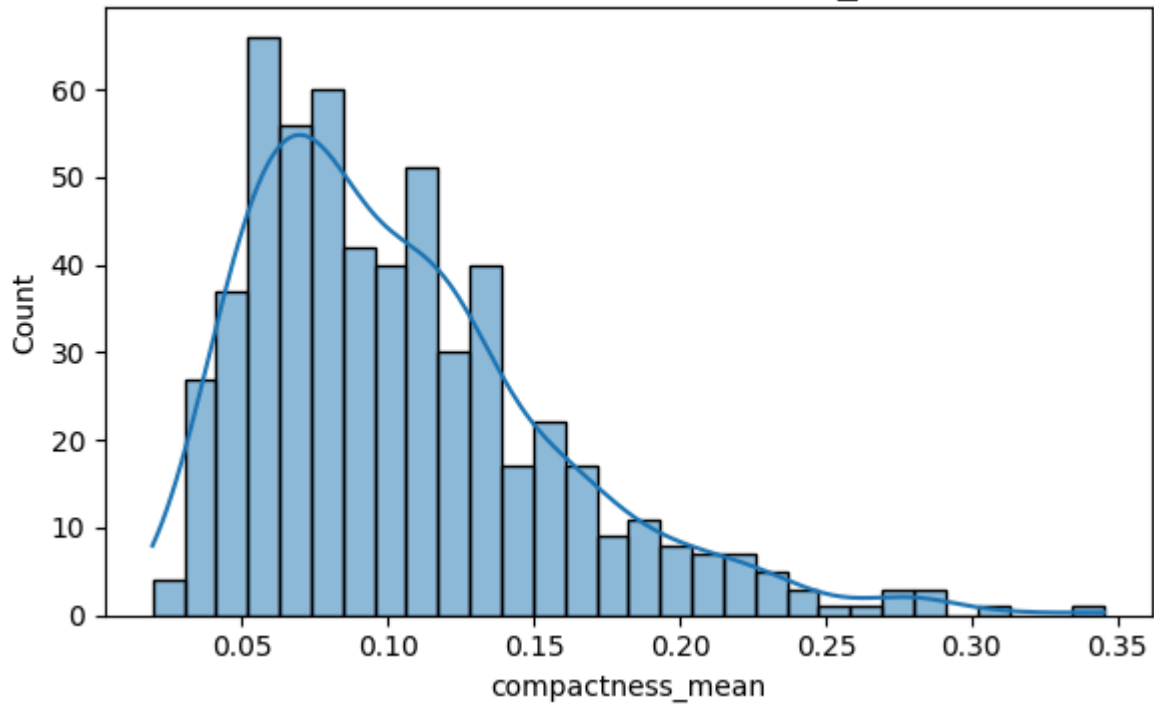


Distribution of smoothness_mean

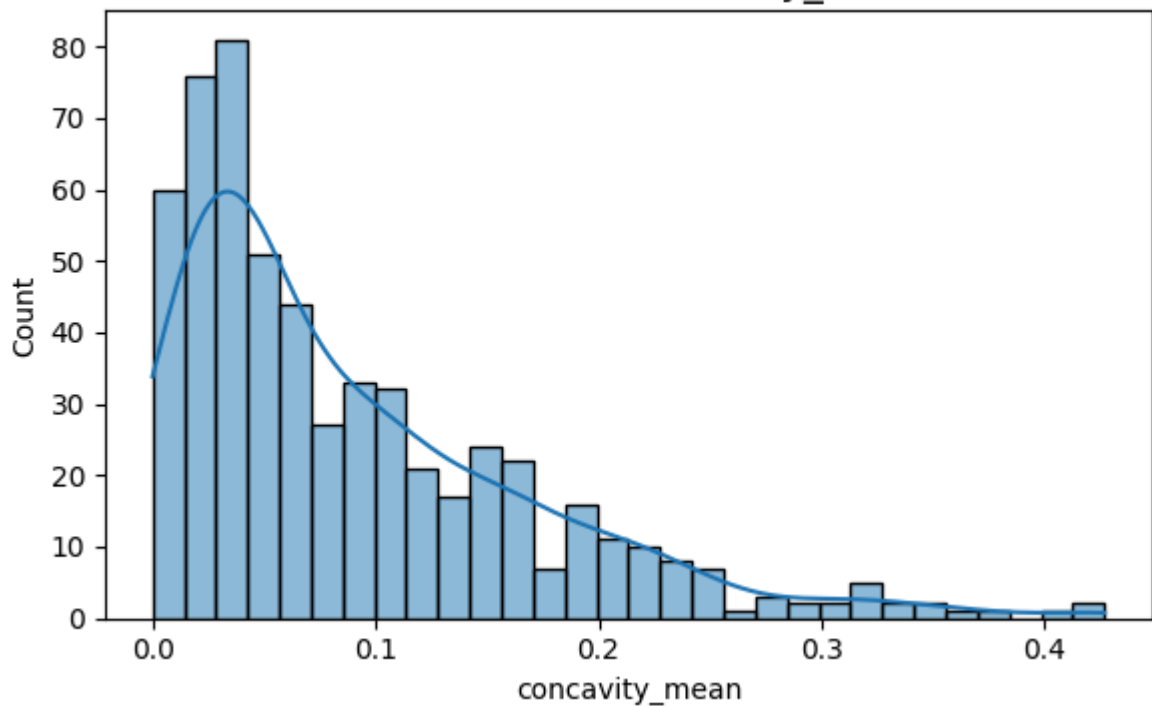




Distribution of compactness_mean

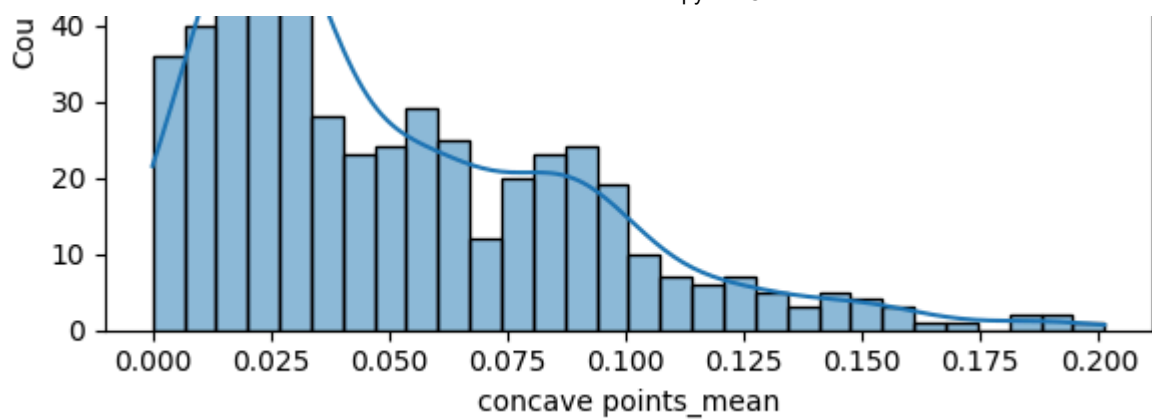


Distribution of concavity_mean

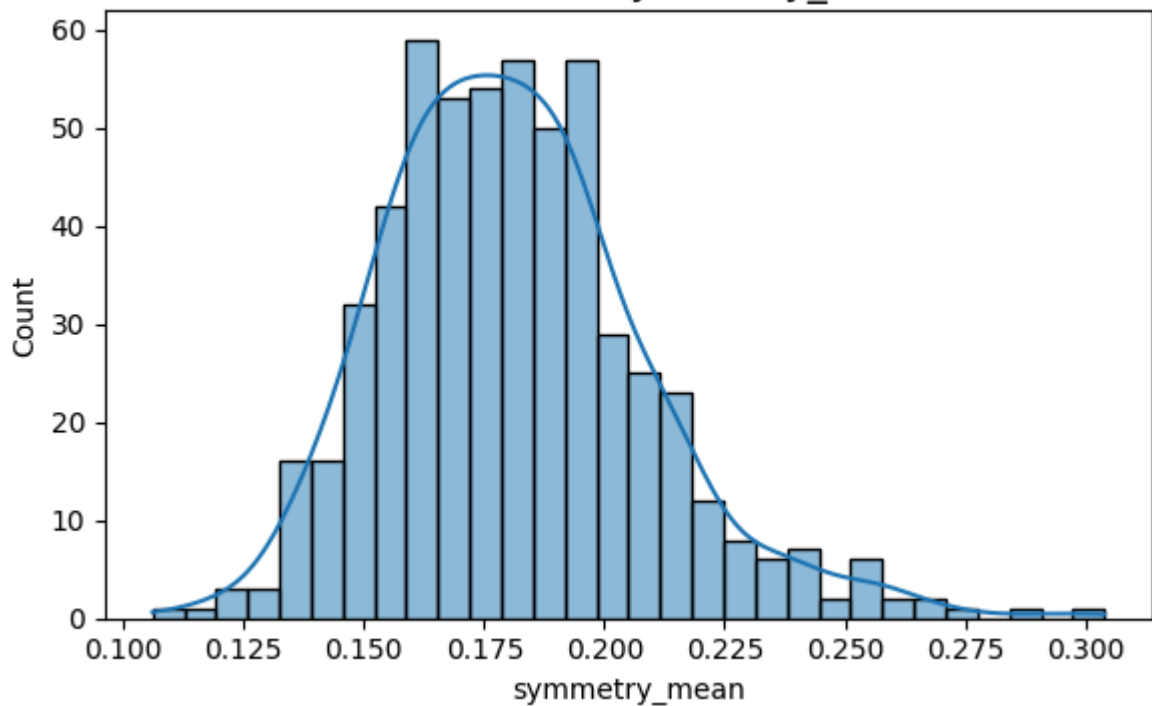


Distribution of concave points_mean

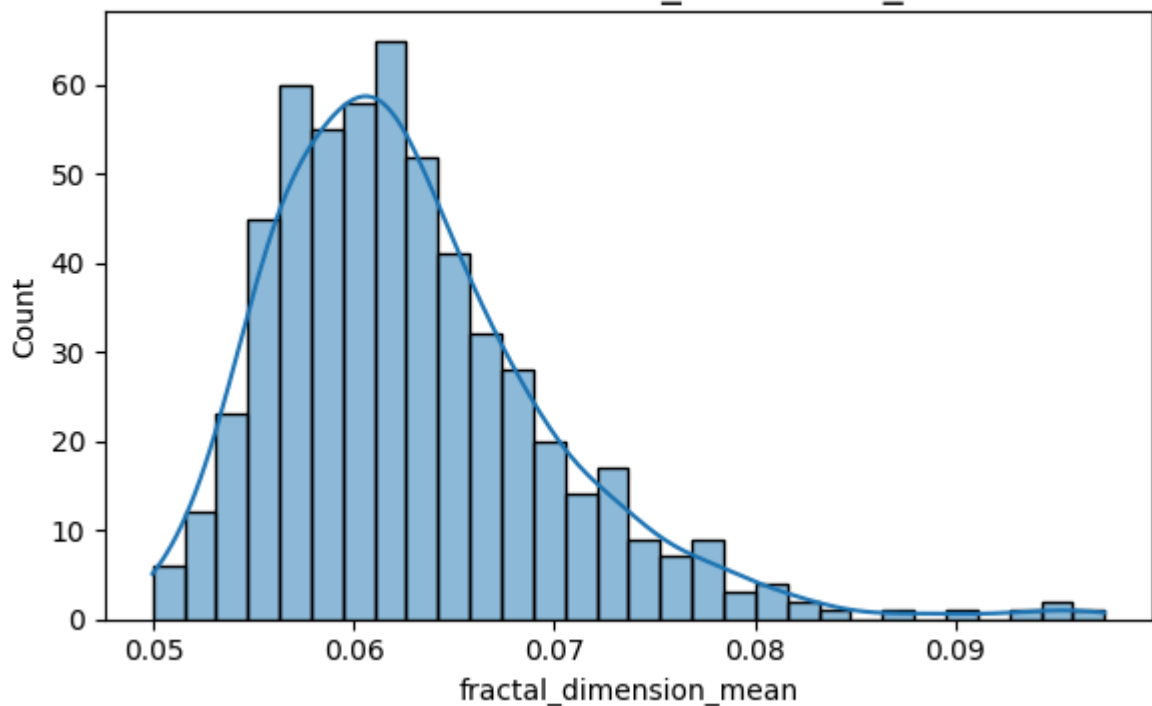




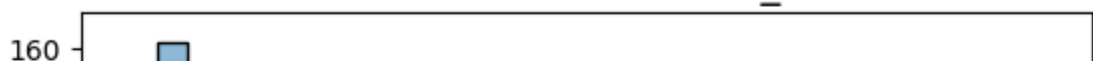
Distribution of symmetry_mean

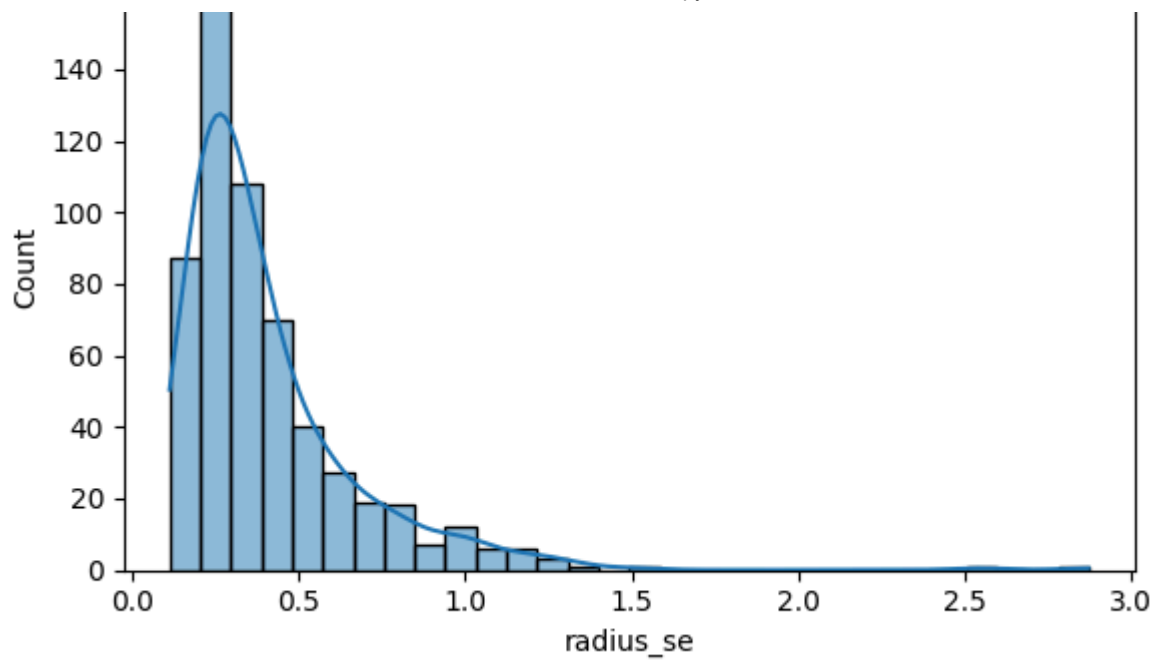


Distribution of fractal_dimension_mean

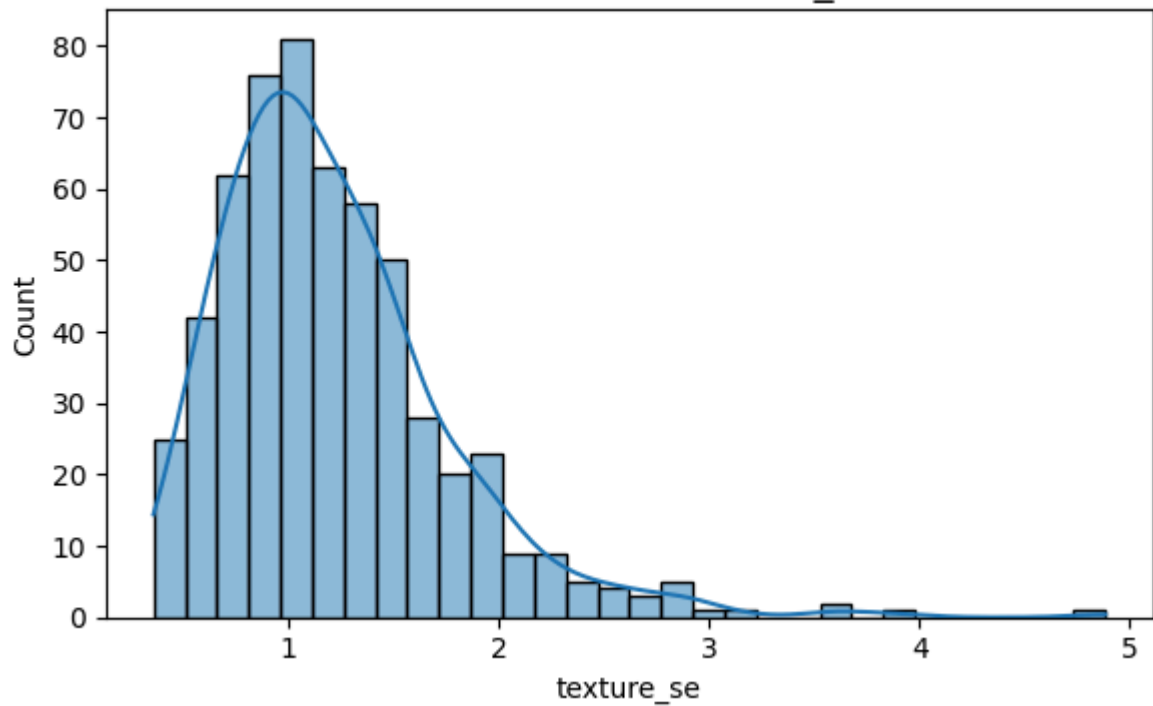


Distribution of radius_se

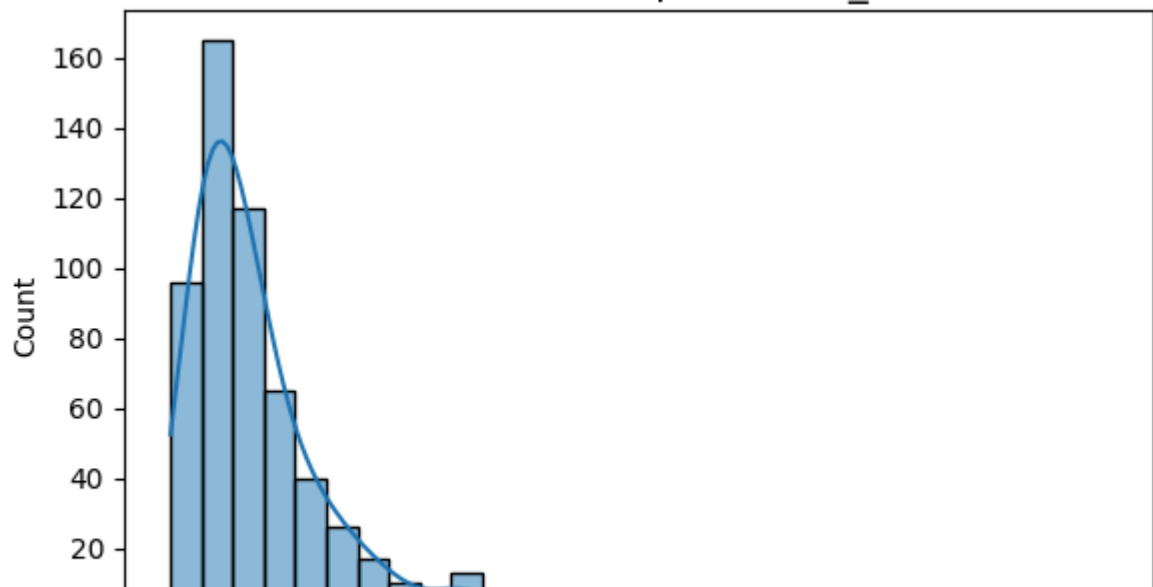


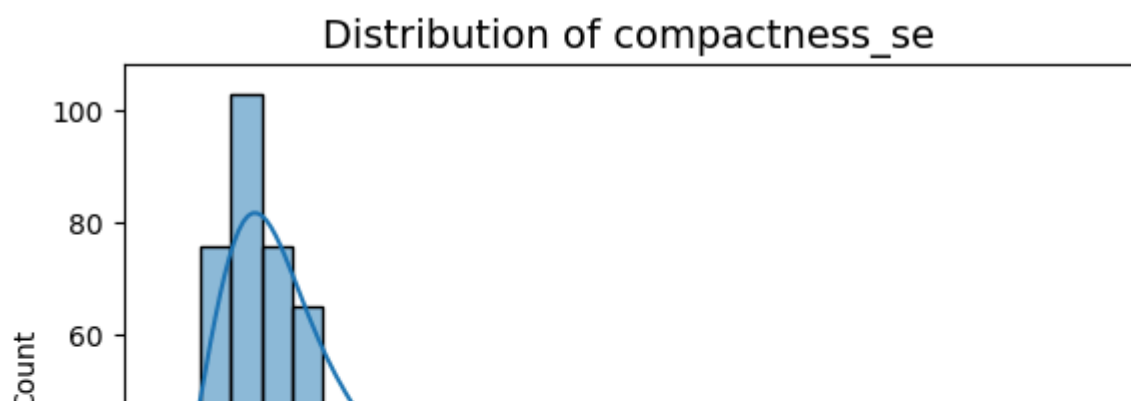
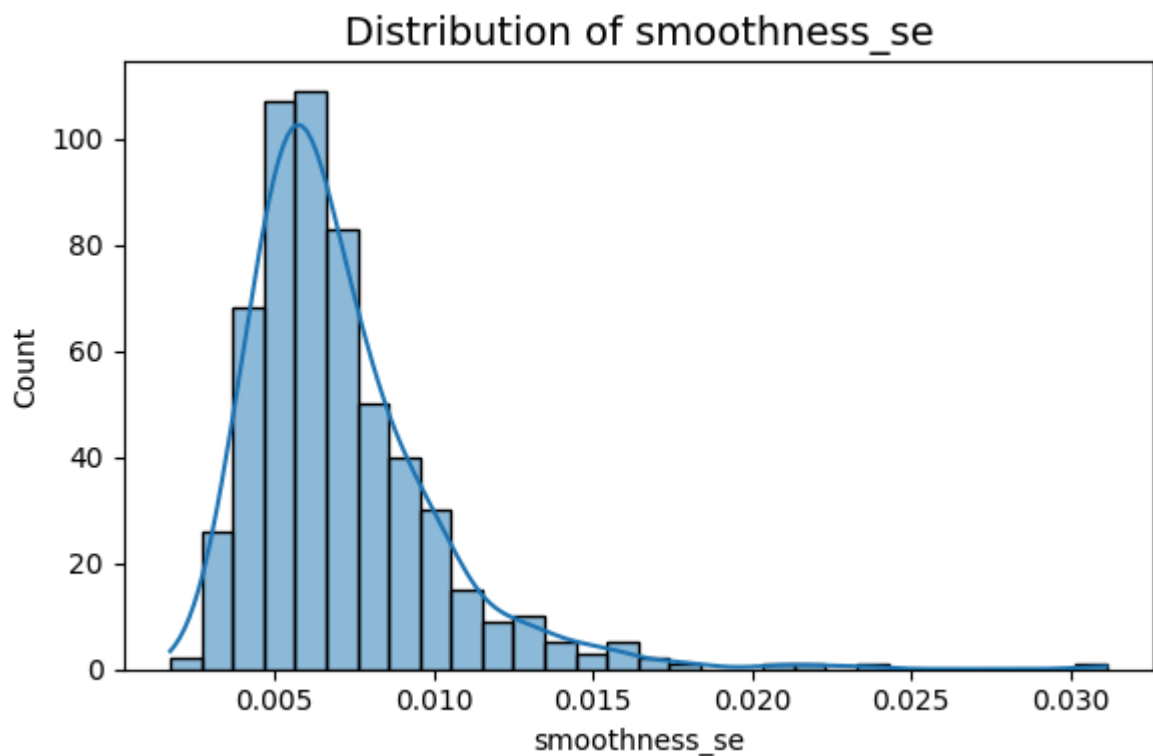
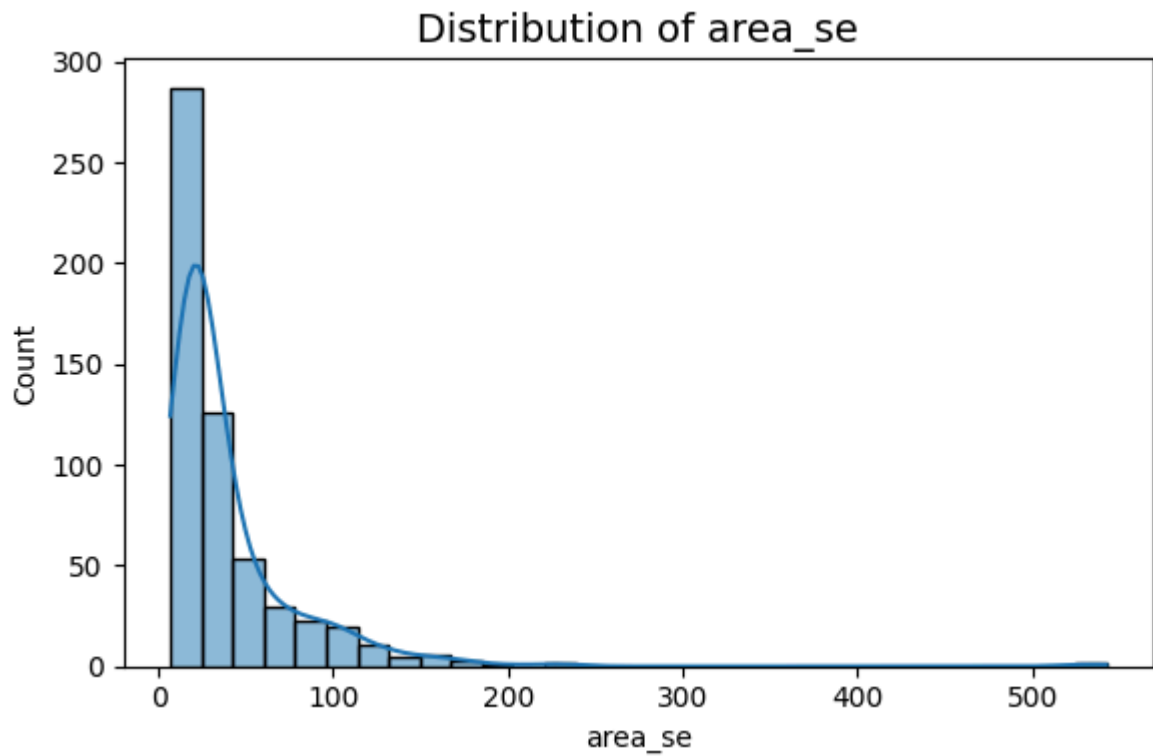
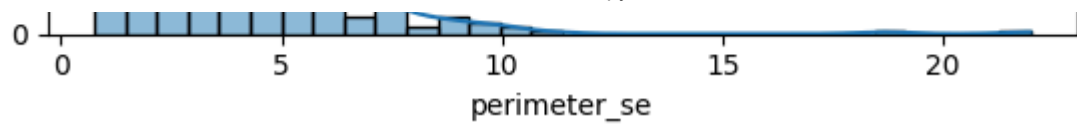


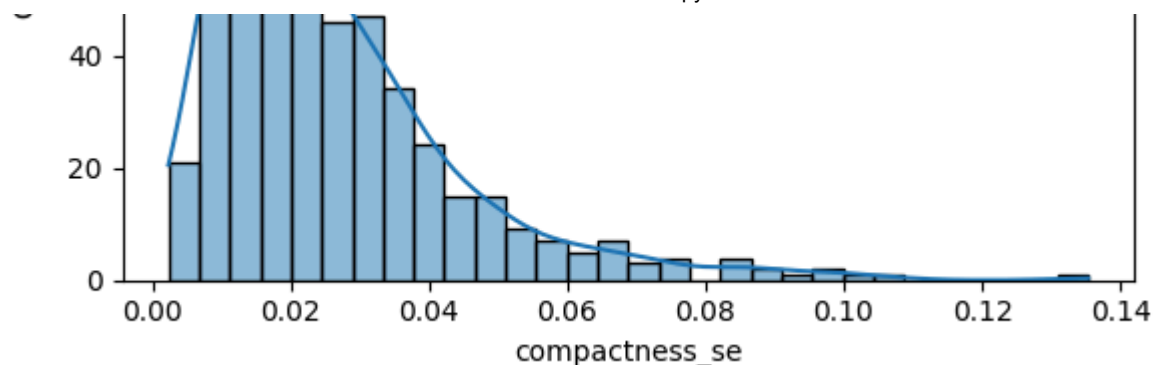
Distribution of texture_se



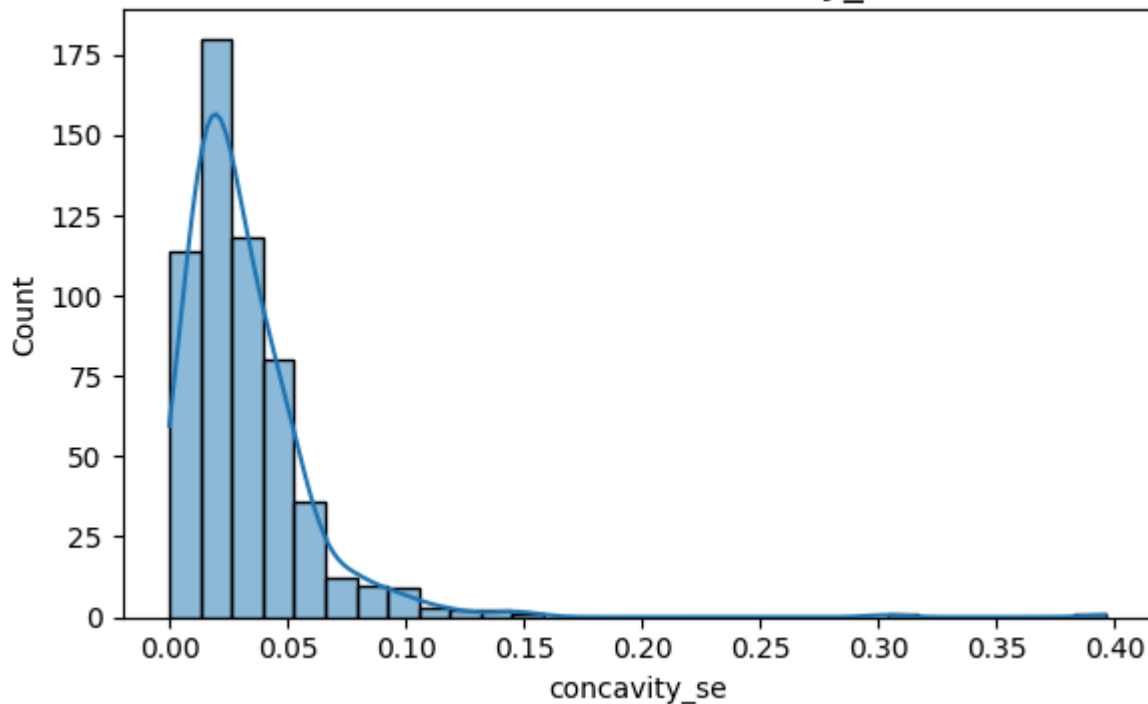
Distribution of perimeter_se



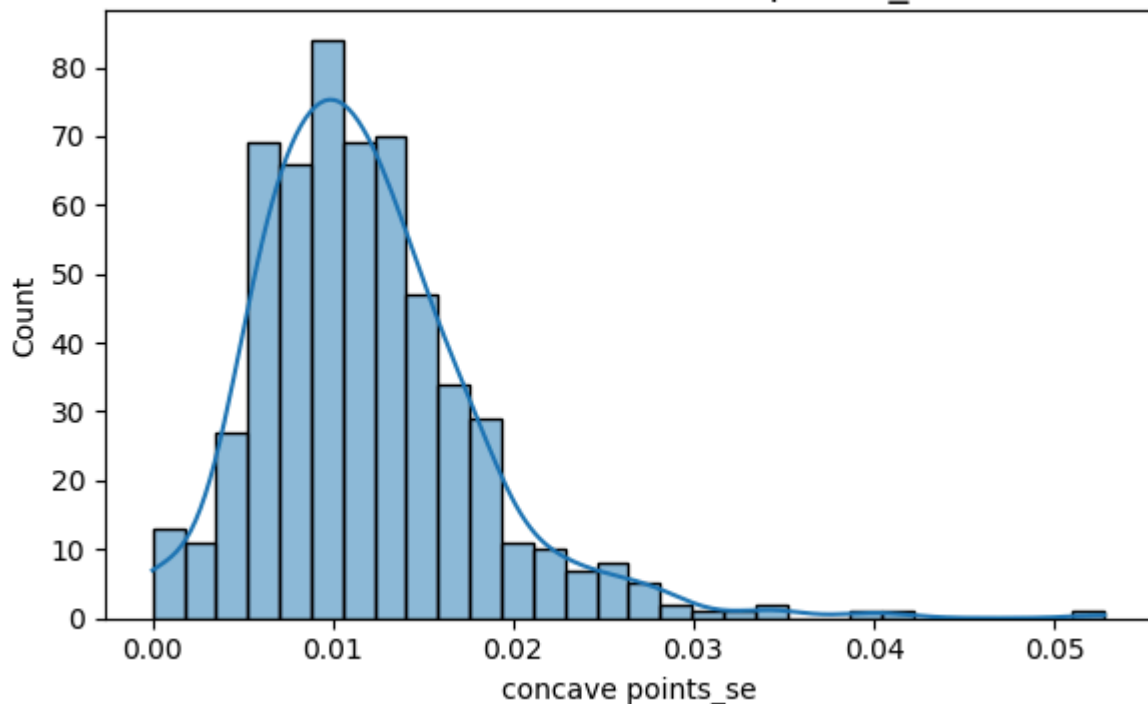




Distribution of concavity_se



Distribution of concave points_se



Distribution of symmetry_se

