

```
# رابط تحميل kaggle dataset
# https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data
```

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
```

```
df = pd.read_csv('data.csv')
```

```
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]

df.dtypes

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
```

```
df.columns
```

```
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')
```

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

```
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

```

```
pip install scikit-learn
```

```

Requirement already satisfied: scikit-learn in c:\users\abdulalah\
anaconda3\envs\coding\lib\site-packages (1.7.2)
Requirement already satisfied: numpy>=1.22.0 in c:\users\abdulalah\
anaconda3\envs\coding\lib\site-packages (from scikit-learn) (2.3.4)
Requirement already satisfied: scipy>=1.8.0 in c:\users\abdulalah\
anaconda3\envs\coding\lib\site-packages (from scikit-learn) (1.16.3)
Requirement already satisfied: joblib>=1.2.0 in c:\users\abdulalah\
anaconda3\envs\coding\lib\site-packages (from scikit-learn) (1.5.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\
abdulalah\anaconda3\envs\coding\lib\site-packages (from scikit-learn)
(3.6.0)

```

Note: you may need to restart the kernel to use updated packages.

```
import pandas as pd
```

```
df = pd.read_csv('data.csv')
```

```

# عرض أول 5 صفوف للتأكد من التحميل
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]

حذف العمود الفارغ

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

```
from sklearn.preprocessing import LabelEncoder
```

```
label_encoder = LabelEncoder()
df['diagnosis'] = label_encoder.fit_transform(df['diagnosis'])
df.head()
```

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

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

```
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
dtype: int64
```

```
pip install seaborn
```

Collecting seaborn

Downloading seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\

```
abdulelah\anaconda3\envs\coding\lib\site-packages (from seaborn)
(2.3.4)
Requirement already satisfied: pandas>=1.2 in c:\users\abdulelah\
anaconda3\envs\coding\lib\site-packages (from seaborn) (2.3.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\users\
abdulelah\anaconda3\envs\coding\lib\site-packages (from seaborn)
(3.10.7)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\abdulelah\
anaconda3\envs\coding\lib\site-packages (from matplotlib!=3.6.1,>=3.4-
>seaborn) (1.3.3)
Requirement already satisfied: cycler>=0.10 in c:\users\abdulelah\
anaconda3\envs\coding\lib\site-packages (from matplotlib!=3.6.1,>=3.4-
>seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\
abdulelah\anaconda3\envs\coding\lib\site-packages (from matplotlib!
=3.6.1,>=3.4->seaborn) (4.60.1)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\
abdulelah\anaconda3\envs\coding\lib\site-packages (from matplotlib!
=3.6.1,>=3.4->seaborn) (1.4.9)
Requirement already satisfied: packaging>=20.0 in c:\users\abdulelah\
anaconda3\envs\coding\lib\site-packages (from matplotlib!=3.6.1,>=3.4-
>seaborn) (25.0)
Requirement already satisfied: pillow>=8 in c:\users\abdulelah\
anaconda3\envs\coding\lib\site-packages (from matplotlib!=3.6.1,>=3.4-
>seaborn) (12.0.0)
Requirement already satisfied: pyparsing>=3 in c:\users\abdulelah\
anaconda3\envs\coding\lib\site-packages (from matplotlib!=3.6.1,>=3.4-
>seaborn) (3.2.5)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\
abdulelah\anaconda3\envs\coding\lib\site-packages (from matplotlib!
=3.6.1,>=3.4->seaborn) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\abdulelah\
anaconda3\envs\coding\lib\site-packages (from pandas>=1.2->seaborn)
(2025.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\abdulelah\
anaconda3\envs\coding\lib\site-packages (from pandas>=1.2->seaborn)
(2025.2)
Requirement already satisfied: six>=1.5 in c:\users\abdulelah\
anaconda3\envs\coding\lib\site-packages (from python-dateutil>=2.7-
>matplotlib!=3.6.1,>=3.4->seaborn) (1.17.0)
Downloading seaborn-0.13.2-py3-none-any.whl (294 kB)
Installing collected packages: seaborn
Successfully installed seaborn-0.13.2
Note: you may need to restart the kernel to use updated packages.
```

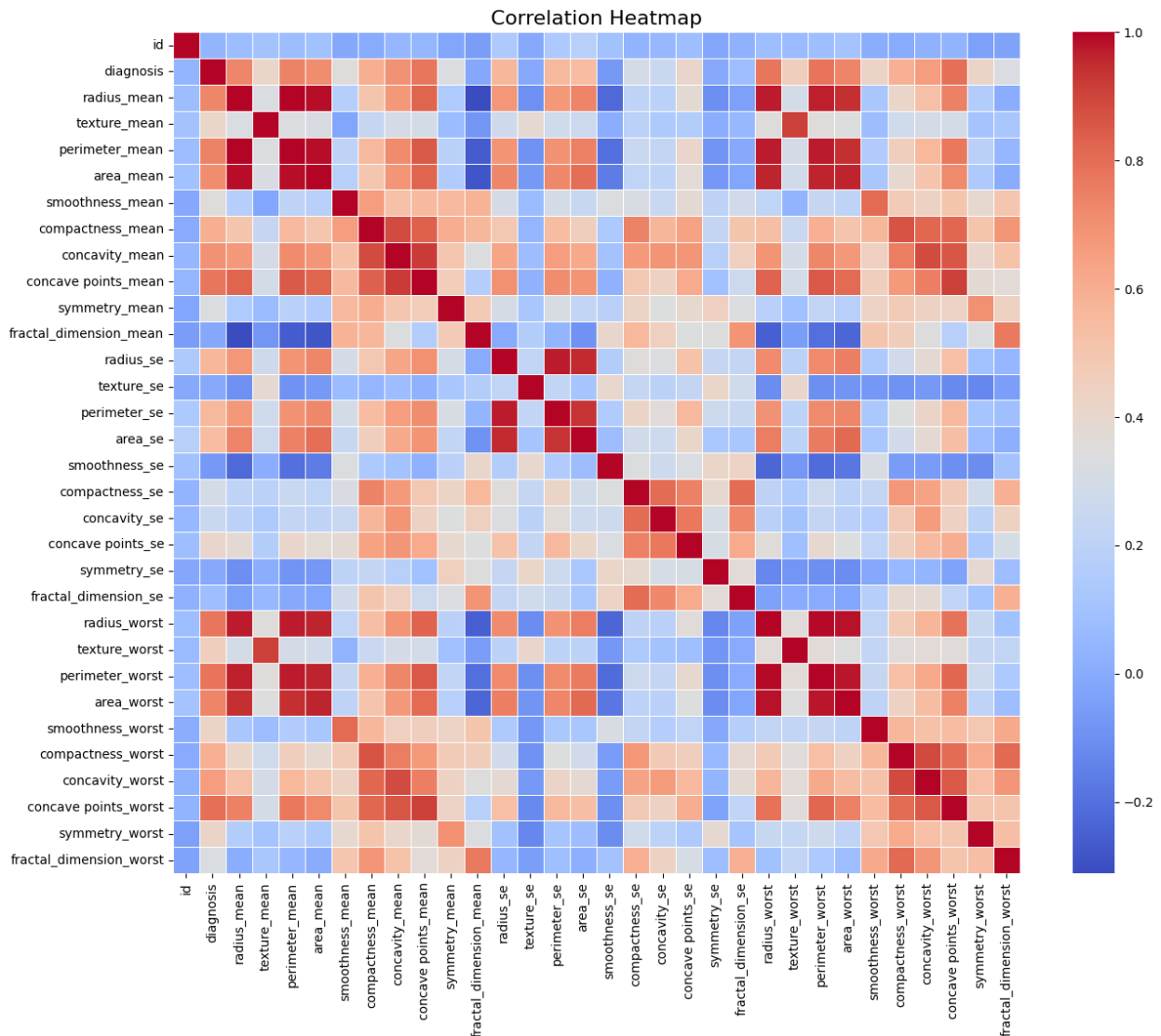
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
```

```
# تحميل البيانات
df = pd.read_csv('data.csv')

# حذف العمود الفارغ
if 'Unnamed: 32' in df.columns:
    df = df.drop(columns=['Unnamed: 32'])

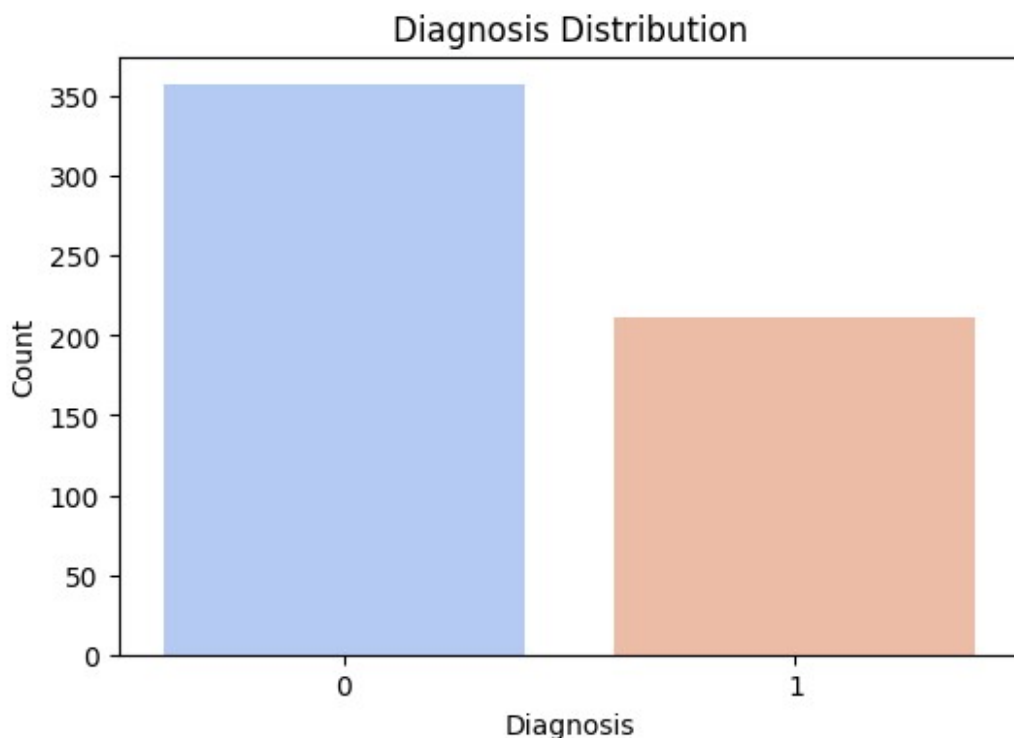
# ترميز العمود diagnosis
label_encoder = LabelEncoder()
df['diagnosis'] = label_encoder.fit_transform(df['diagnosis'])

# رسم خريطة الارتباط
plt.figure(figsize=(15, 12))
sns.heatmap(df.corr(numeric_only=True), cmap='coolwarm',
linewidths=0.5)
plt.title('Correlation Heatmap', fontsize=16)
plt.show()
```



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

plt.figure(figsize=(6, 4))
sns.countplot(x='diagnosis', hue='diagnosis', data=df,
palette='coolwarm', legend=False)
plt.title('Diagnosis Distribution')
plt.xlabel('Diagnosis')
plt.ylabel('Count')
plt.show()
```



تحديد العمود الذي نريد فحصه

```
col = 'radius_mean'
```

والمدى بينهما Q1 و Q3 حساب الربيعين

```
Q1 = df[col].quantile(0.25)
```

```
Q3 = df[col].quantile(0.75)
```

```
IQR = Q3 - Q1
```

تحديد الحدود الدنيا والعليا

```
lower_bound = Q1 - 1.5 * IQR
```

```
upper_bound = Q3 + 1.5 * IQR
```

تحديد القيم المتطرفة

```
outliers = df[(df[col] < lower_bound) | (df[col] > upper_bound)]
```

```
print(f'عدد القيم المتطرفة في العمود {col}: {len(outliers)}')
```

```
outliers.head()
```

radius_mean: 14 عدد القيم المتطرفة في العمود

	id	diagnosis	radius_mean	texture_mean	perimeter_mean
area_mean \					
82	8611555	1	25.22	24.91	171.5
1878.0					
108	86355	1	22.27	19.67	152.8
1509.0					
122	865423	1	24.25	20.20	166.2

1761.0					
164	8712289	1	23.27	22.04	152.1
1686.0					
180	873592	1	27.22	21.87	182.1
2250.0					

	smoothness_mean	compactness_mean	concavity_mean	concave
points_mean \				
82	0.10630	0.2665	0.3339	
0.18450				
108	0.13260	0.2768	0.4264	
0.18230				
122	0.14470	0.2867	0.4268	
0.20120				
164	0.08439	0.1145	0.1324	
0.09702				
180	0.10940	0.1914	0.2871	
0.18780				

	radius_worst	texture_worst	perimeter_worst	area_worst	\
82	30.00	33.62	211.7	2562.0	
108	28.40	28.01	206.8	2360.0	
122	26.02	23.99	180.9	2073.0	
164	28.01	28.22	184.2	2403.0	
180	33.12	32.85	220.8	3216.0	

	smoothness_worst	compactness_worst	concavity_worst	\
82	0.1573	0.6076	0.6476	
108	0.1701	0.6997	0.9608	
122	0.1696	0.4244	0.5803	
164	0.1228	0.3583	0.3948	
180	0.1472	0.4034	0.5340	

	concave	points_worst	symmetry_worst	fractal_dimension_worst
82	0.2867	0.2355		0.10510
108	0.2910	0.4055		0.09789
122	0.2248	0.3222		0.08009
164	0.2346	0.3589		0.09187
180	0.2688	0.2856		0.08082

[5 rows x 32 columns]

```

from scipy import stats
import numpy as np

# اختيار الأعمدة الرقمية فقط
numeric_cols = df.select_dtypes(include=['float64', 'int64']).columns

# حساب z-scores
z_scores = np.abs(stats.zscore(df[numeric_cols], nan_policy='omit'))

```

```
# تحديد أي صف يحتوي على outlier
outliers = (z_scores > 3)

# عدد القيم المتطرفة لكل عمود
outlier_count = np.sum(outliers, axis=0)

outlier_summary = pd.DataFrame({
    'Feature': numeric_cols,
    'Outlier_Count': outlier_count
}).sort_values(by='Outlier_Count', ascending=False)
```

outlier_summary

	Feature	Outlier_Count
17	compactness_se	12
0	id	11
20	symmetry_se	11
21	fractal_dimension_se	10
25	area_worst	10
27	compactness_worst	10
13	texture_se	9
7	compactness_mean	9
30	symmetry_worst	9
31	fractal_dimension_worst	9
8	concavity_mean	9
5	area_mean	8
14	perimeter_se	8
12	radius_se	7
11	fractal_dimension_mean	7
4	perimeter_mean	7
28	concavity_worst	7
16	smoothness_se	7
18	concavity_se	6
22	radius_worst	6
24	perimeter_worst	6
15	area_se	6
9	concave points_mean	6
19	concave points_se	6
2	radius_mean	5
6	smoothness_mean	5
10	symmetry_mean	5
3	texture_mean	4
23	texture_worst	4
26	smoothness_worst	3
1	diagnosis	0
29	concave points_worst	0

```
# إزالة أي صف يحتوي على أي قيمة متطرفة
df_clean = df[(z_scores < 3).all(axis=1)]
```

```
print(f'عدد الصفوف بعد إزالة القيم المتطرفة: {len(df_clean)}')
```

عدد الصفوف بعد إزالة القيم المتطرفة: 487

```
outlier_dict = {} # تخزين عدد القيم المتطرفة لكل عمود
```

```
for col in numeric_cols:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # استخراج القيم المتطرفة
    outliers = df[(df[col] < lower_bound) | (df[col] > upper_bound)]

    outlier_dict[col] = len(outliers)
```

```
# عرض عدد القيم المتطرفة لكل عمود
```

```
outlier_dict
```

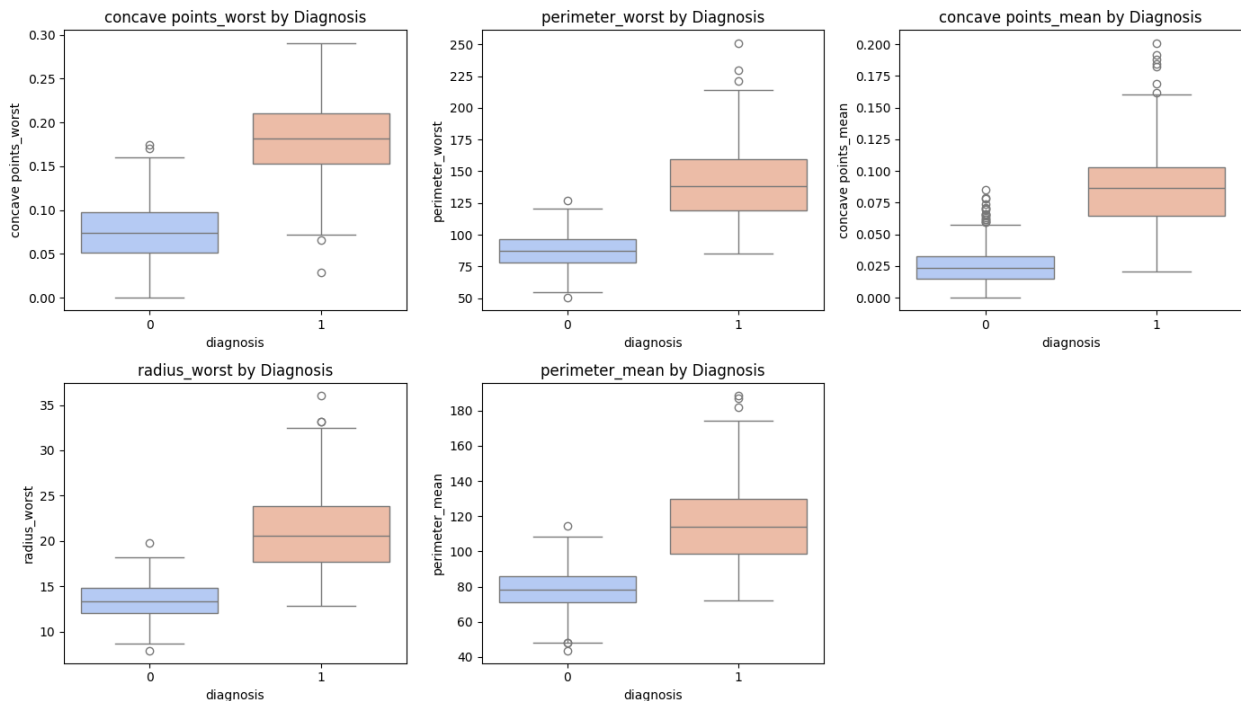
```
{'id': 81,
 'diagnosis': 0,
 'radius_mean': 14,
 'texture_mean': 7,
 'perimeter_mean': 13,
 'area_mean': 25,
 'smoothness_mean': 6,
 'compactness_mean': 16,
 'concavity_mean': 18,
 'concave points_mean': 10,
 'symmetry_mean': 15,
 'fractal_dimension_mean': 15,
 'radius_se': 38,
 'texture_se': 20,
 '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}
```

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

```
# إيجاد أهم 5 خصائص مرتبطة بالتشخيص
corr = df.corr(numeric_only=True)
['diagnosis'].sort_values(ascending=False)
top5 = corr.index[1:6]
```

```
# لكل خاصية boxplot رسم
plt.figure(figsize=(14, 8))
for i, feature in enumerate(top5, 1):
    plt.subplot(2, 3, i)
    sns.boxplot(x='diagnosis', y=feature, hue='diagnosis', data=df,
palette='coolwarm', legend=False)
    plt.title(f'{feature} by Diagnosis')
plt.tight_layout()
plt.show()
```



```
from scipy import stats
import numpy as np
```

```
# نسخة من البيانات
```

```

df_clean_z = df.copy()

# الأعمدة الرقمية فقط
numeric_cols = df_clean_z.select_dtypes(include=['float64',
'int64']).columns

# حساب Z-scores
z_scores = np.abs(stats.zscore(df_clean_z[numeric_cols],
nan_policy='omit'))

# Z-score < 3 الاحتفاظ بالصفوف التي جميع قيمها
df_clean_z = df_clean_z[(z_scores < 3).all(axis=1)]

print(f'Z-score عدد الصفوف بعد إزالة القيم المتطرفة باستخدام:
{len(df_clean_z)}')

Z-score عدد الصفوف بعد إزالة القيم المتطرفة باستخدام: 487

import matplotlib.pyplot as plt
import seaborn as sns

# عمل نسخة من البيانات الأصلية
df_clean = df.copy()

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

# لكل عمود IQR إزالة القيم المتطرفة باستخدام
for col in numeric_cols:
    Q1 = df_clean[col].quantile(0.25)
    Q3 = df_clean[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    df_clean = df_clean[(df_clean[col] >= lower_bound) &
(df_clean[col] <= upper_bound)]

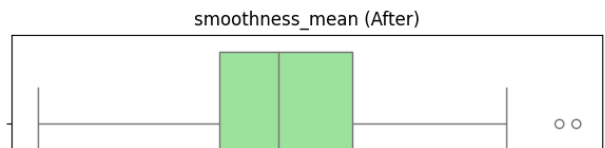
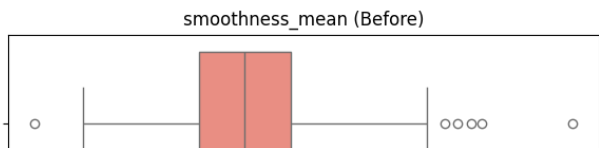
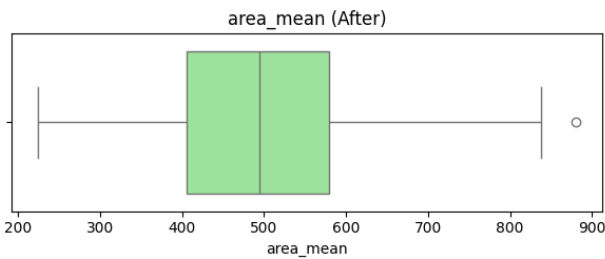
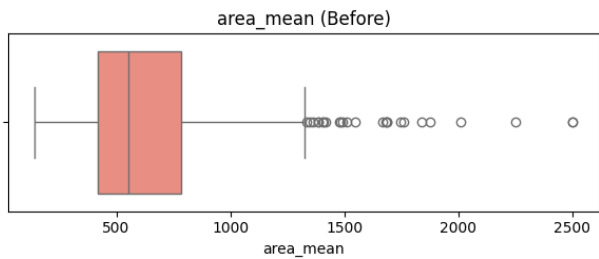
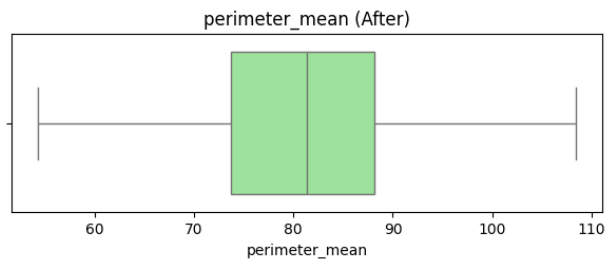
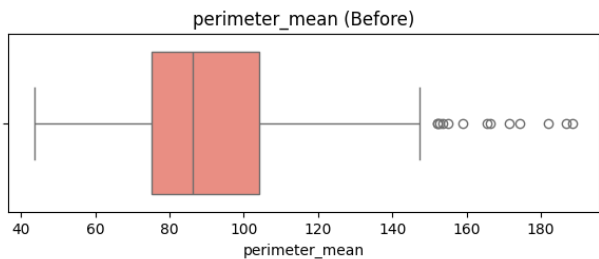
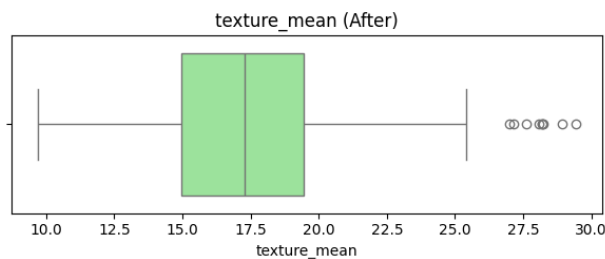
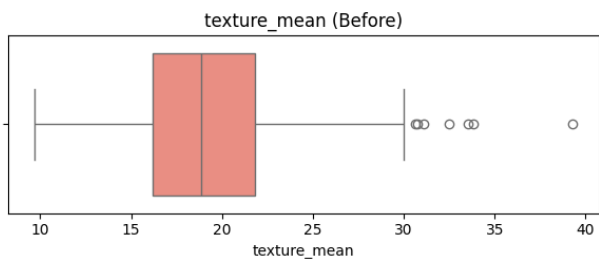
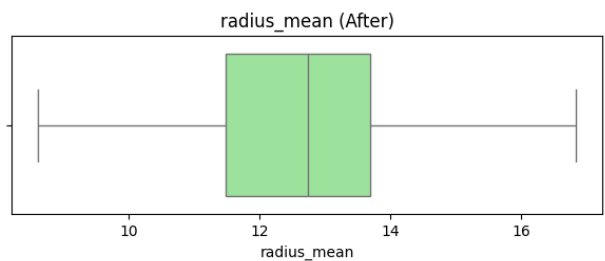
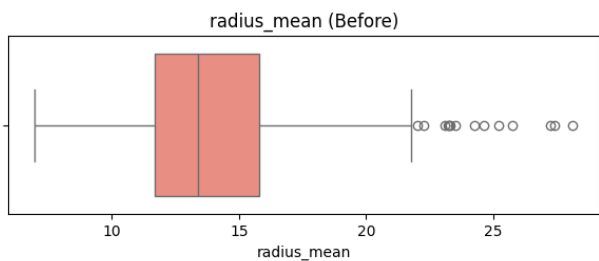
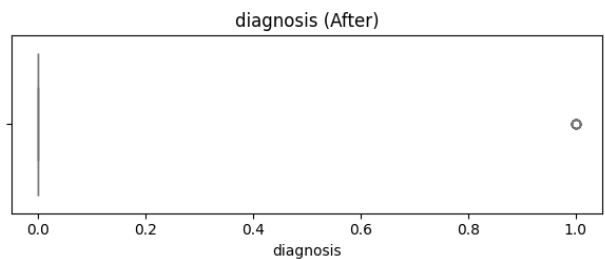
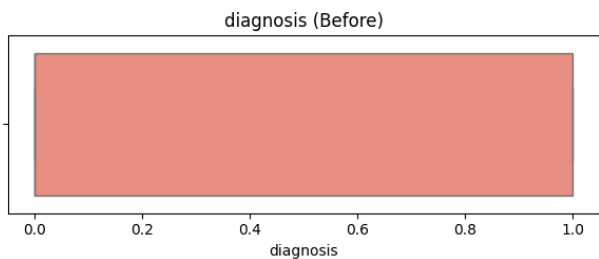
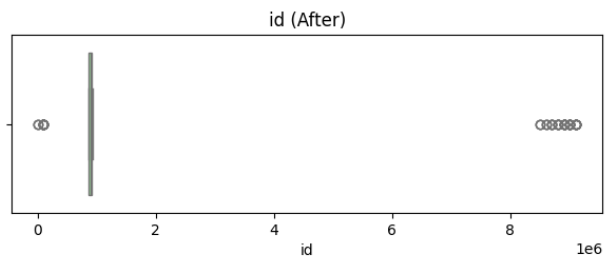
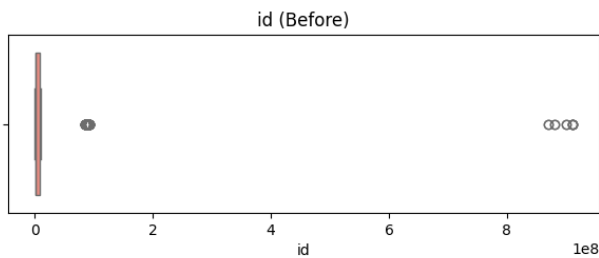
# قبل وبعد Boxplot رسم
fig, axes = plt.subplots(len(numeric_cols), 2, figsize=(12,
len(numeric_cols)*2.5))

for i, col in enumerate(numeric_cols):
    sns.boxplot(x=df[col], ax=axes[i,0], color='salmon')
    axes[i,0].set_title(f'{col} (Before)')

    sns.boxplot(x=df_clean[col], ax=axes[i,1], color='lightgreen')
    axes[i,1].set_title(f'{col} (After)')

plt.tight_layout()
plt.show()

```



```

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder

# تحميل البيانات
df = pd.read_csv('data.csv')

# حذف العمود الفارغ (إن وُجد)
if 'Unnamed: 32' in df.columns:
    df = df.drop(columns=['Unnamed: 32'])

# تحويل diagnosis إلى أرقام (M=1, B=0)
label_encoder = LabelEncoder()
df['diagnosis'] = label_encoder.fit_transform(df['diagnosis'])

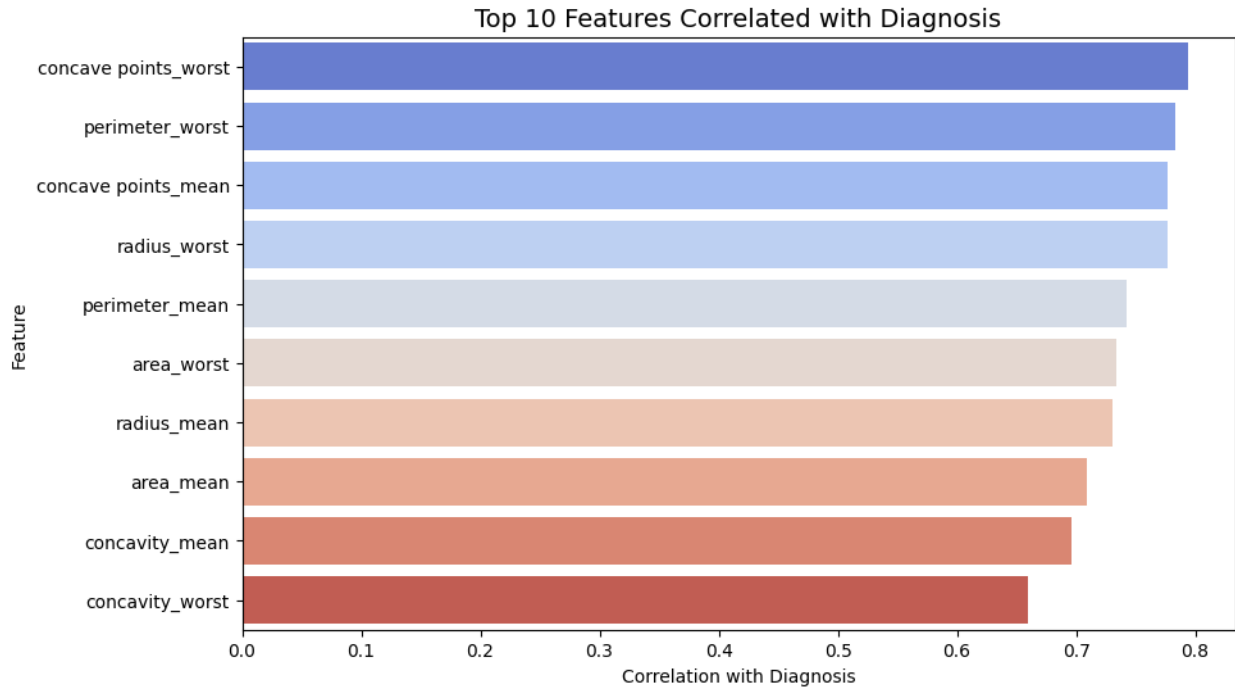
# حساب مصفوفة الارتباط
corr = df.corr(numeric_only=True)

# ترتيبه تنازليًا diagnosis استخراج الارتباط مع
corr_target = corr['diagnosis'].sort_values(ascending=False)

# نفسه diagnosis باستثناء) أخذ أعلى 10 أعمدة
top_features = corr_target[1:11]

# رسم المخطط الشريطي بدون تحذير
plt.figure(figsize=(10, 6))
sns.barplot(x=top_features.values, y=top_features.index,
            hue=top_features.index, palette='coolwarm', legend=False)
plt.title('Top 10 Features Correlated with Diagnosis', fontsize=14)
plt.xlabel('Correlation with Diagnosis')
plt.ylabel('Feature')
plt.show()

```



```
plt.figure(figsize=(8,6))
sns.scatterplot(
    x='radius_mean',
    y='area_mean',
    hue='diagnosis',
    palette='coolwarm',
    data=df
)
plt.title('Radius Mean vs Area Mean (colored by Diagnosis)')
plt.xlabel('Radius Mean')
plt.ylabel('Area Mean')
plt.legend(title='Diagnosis', labels=['Benign (0)', 'Malignant (1)'])
plt.show()
```

تلوين النقاط حسب التشخيص

