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

# Load dataset
df = pd.read_csv("cancer_data_set - cancer_data_set.csv")
df
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	p
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	
...	...	...	...	...	...	...	...	...	...	
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	

569 rows × 32 columns

```
#drop id
df.drop(columns=['id'], inplace=True)

#  (0 = Benign, 1 = Malignant)
df['diagnosis'] = df['diagnosis'].map({'B': 'Benign', 'M': 'Malignant'})
df.head()
```

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
0	Malignant	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710
1	Malignant	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017
2	Malignant	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790
3	Malignant	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520
4	Malignant	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430

5 rows × 31 columns

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

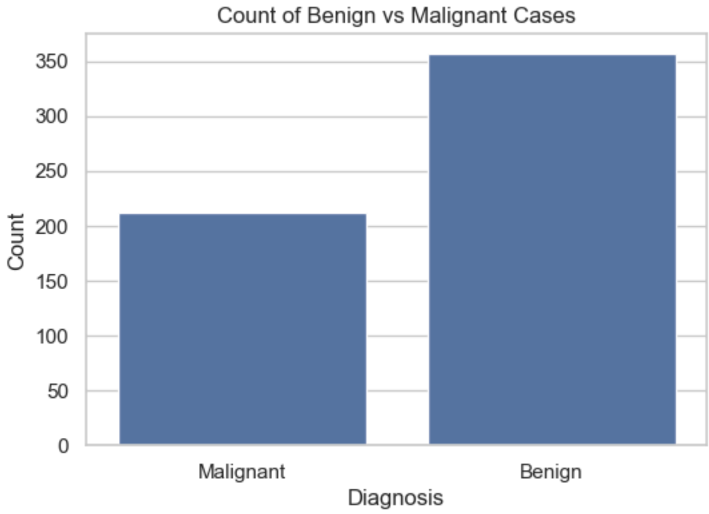
```
diagnosis                  0
radius_mean                0
texture_mean               0
perimeter_mean             0
area_mean                  2
smoothness_mean            0
compactness_mean           1
concavity_mean             1
concave points_mean        0
symmetry_mean              0
fractal_dimension_mean     1
radius_se                  2
texture_se                 0
perimeter_se               0
area_se                    1
smoothness_se              0
compactness_se             0
concavity_se               1
concave points_se          2
symmetry_se                0
fractal_dimension_se       0
radius_worst               0
texture_worst              0
perimeter_worst            0
area_worst                 0
```

```
smoothness_worst           0
compactness_worst          1
concavity_worst            0
concave points_worst       0
symmetry_worst             2
fractal_dimension_worst    0
dtype: int64
```

```
df.fillna(df.select_dtypes(include=['number']).mean(), inplace=True)
```

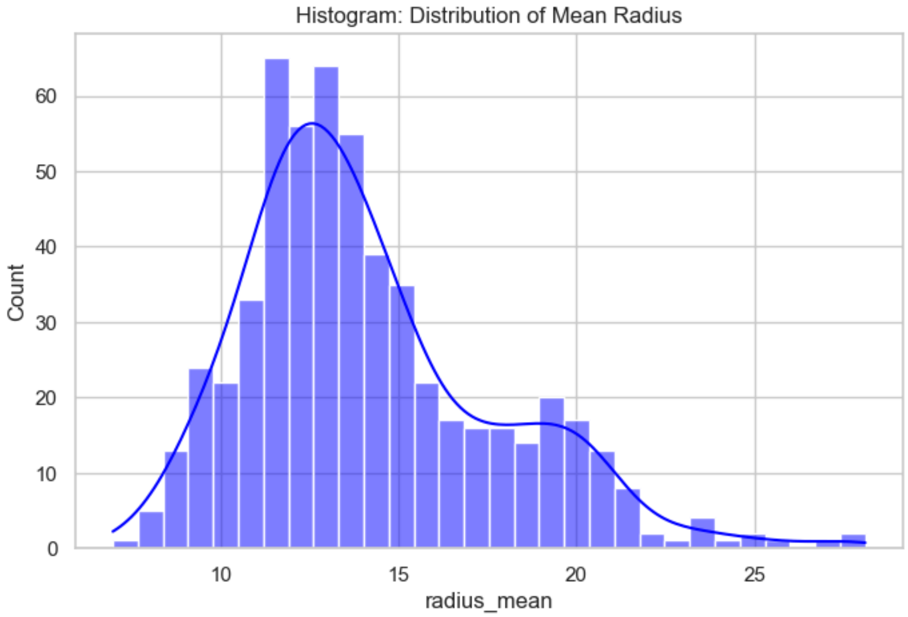
```
# Set Seaborn style
sns.set_theme(style="whitegrid")

plt.figure(figsize=(6, 4))
sns.countplot(data=df, x='diagnosis')
plt.title('Count of Benign vs Malignant Cases')
plt.xlabel('Diagnosis')
plt.ylabel('Count')
plt.show()
print(" Inference: Dataset has more benign cases than malignant cases.")
```



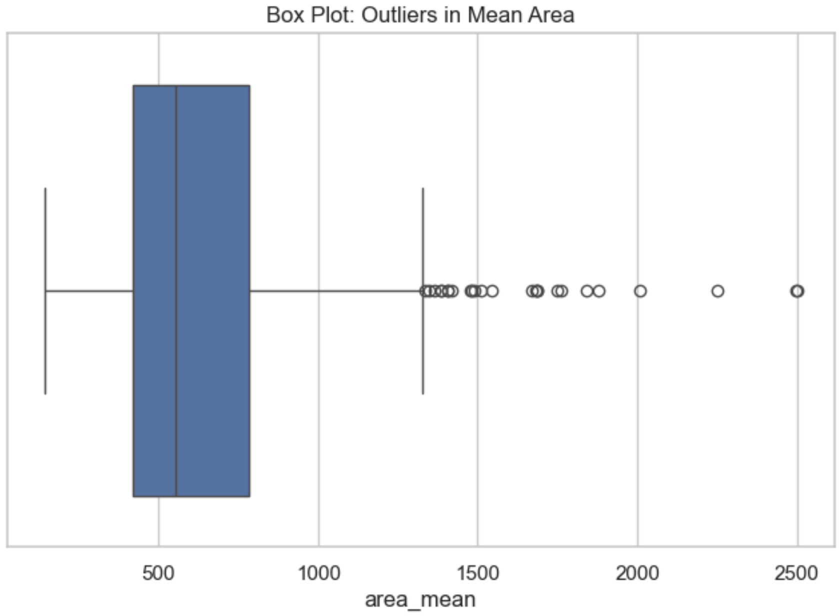
```
Inference: Dataset has more benign cases than malignant cases.
```

```
plt.figure(figsize=(8, 5))
sns.histplot(df['radius_mean'], bins=30, kde=True, color='blue')
plt.title('Histogram: Distribution of Mean Radius')
plt.show()
print("📌 Inference: The distribution is slightly right-skewed, meaning some tumors have significantly larger radii.")
```



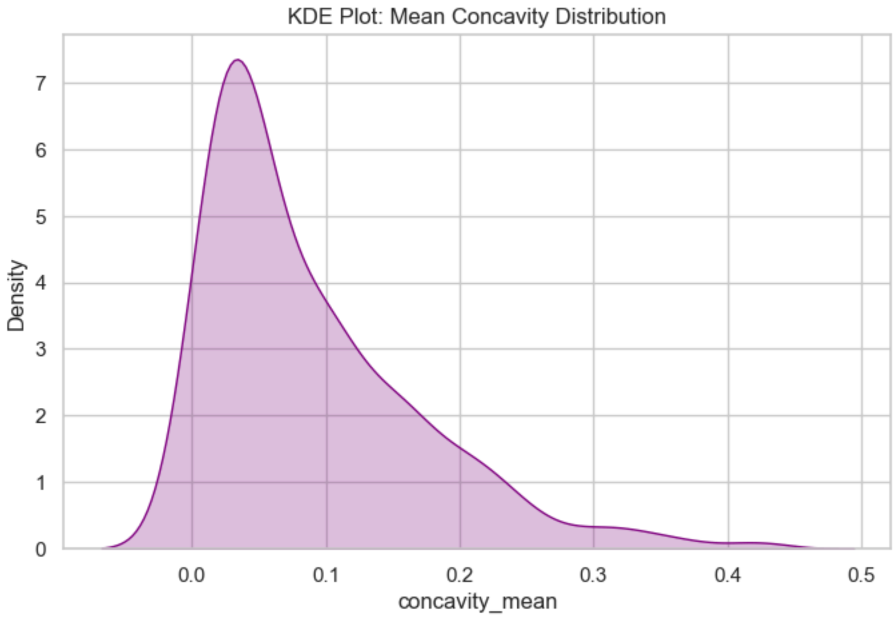
```
📌 Inference: The distribution is slightly right-skewed, meaning some tumors have significantly larger radii.
```

```
plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x='area_mean')
plt.title('Box Plot: Outliers in Mean Area')
plt.show()
print("📌 Inference: A few data points lie above the upper whisker, indicating potential outliers.")
```



```
📌 Inference: A few data points lie above the upper whisker, indicating potential outliers.
```

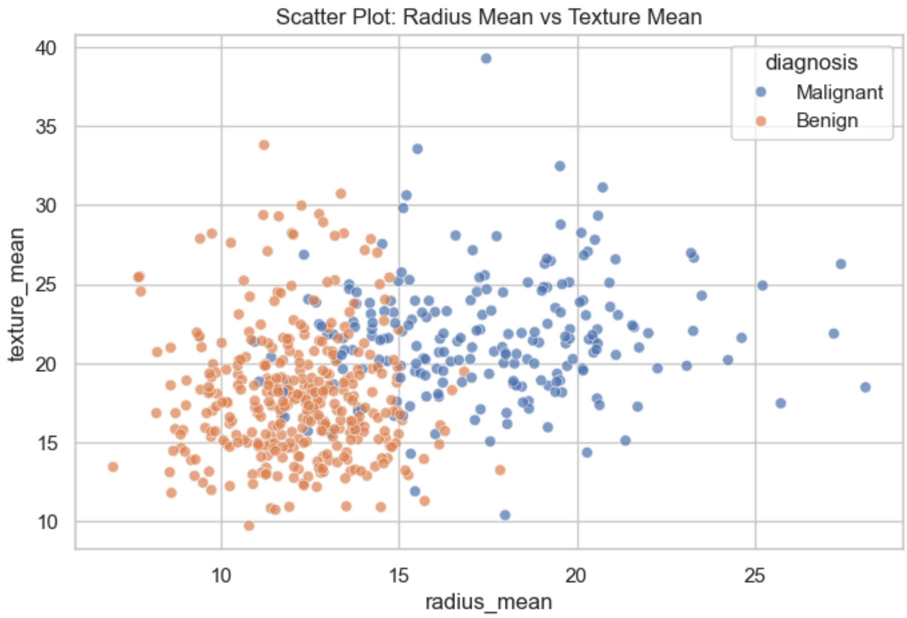
```
plt.figure(figsize=(8, 5))
sns.kdeplot(df['concavity_mean'], fill=True, color='purple')
plt.title('KDE Plot: Mean Concavity Distribution')
plt.show()
print(" Inference: Most tumors have low concavity values, with a long tail indicating some extreme cases.")
```



```
 Inference: Most tumors have low concavity values, with a long tail indicating some extreme cases.
```

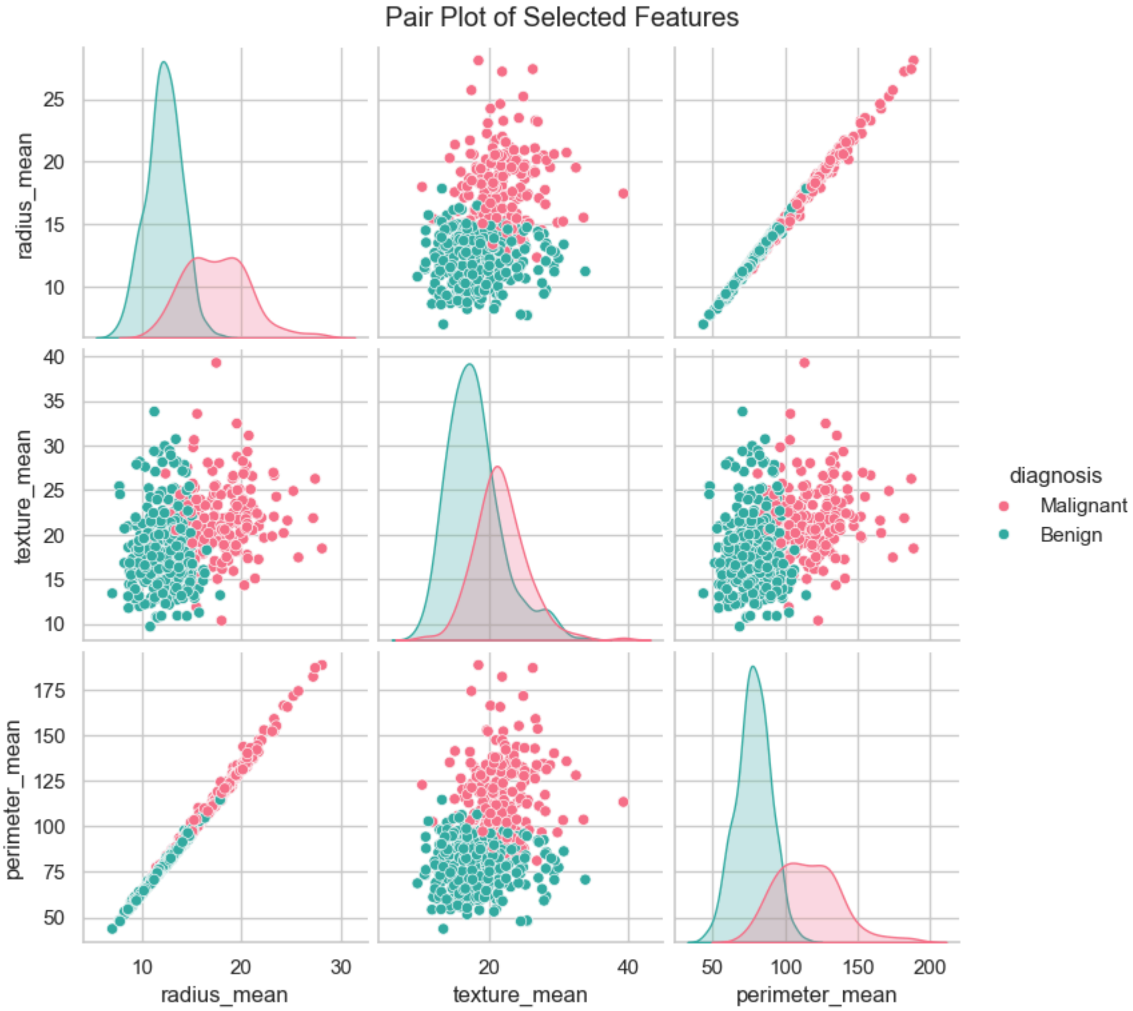
---------------- More than 2 features(relations)----------------

```
plt.figure(figsize=(8, 5))
sns.scatterplot(data=df, x='radius_mean', y='texture_mean', hue='diagnosis', alpha=0.7)
plt.title('Scatter Plot: Radius Mean vs Texture Mean')
plt.show()
print("Inference: Malignant tumors tend to have larger radii and textures compared to benign ones.")
```



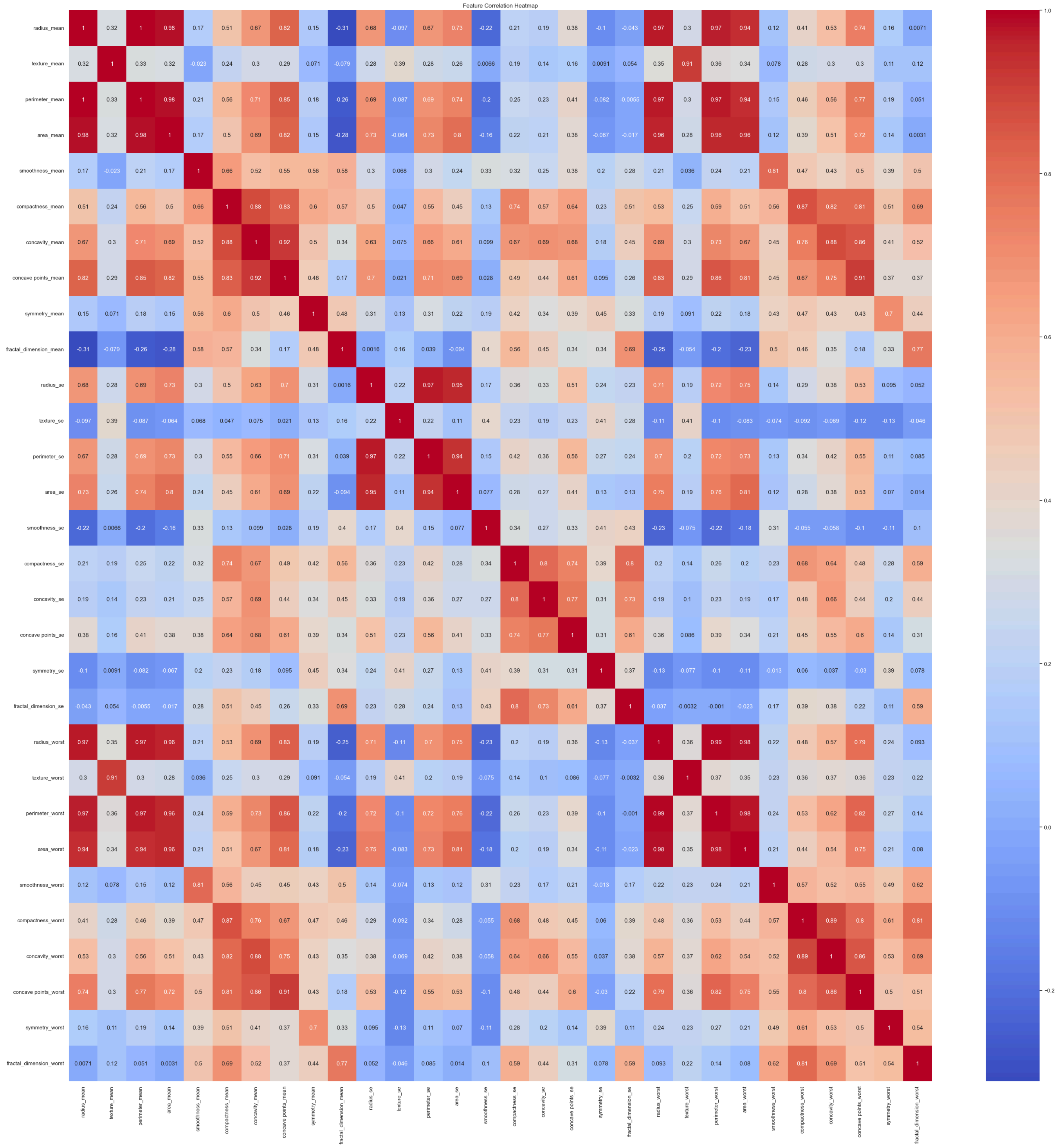
```
Inference: Malignant tumors tend to have larger radii and textures compared to benign ones.
```

```
sns.pairplot(df[['radius_mean', 'texture_mean', 'perimeter_mean', 'diagnosis']], hue='diagnosis', palette='husl')
plt.suptitle('Pair Plot of Selected Features', y=1.02)
plt.show()
print(" Inference: Malignant cases generally have higher values across all three selected features.")
```



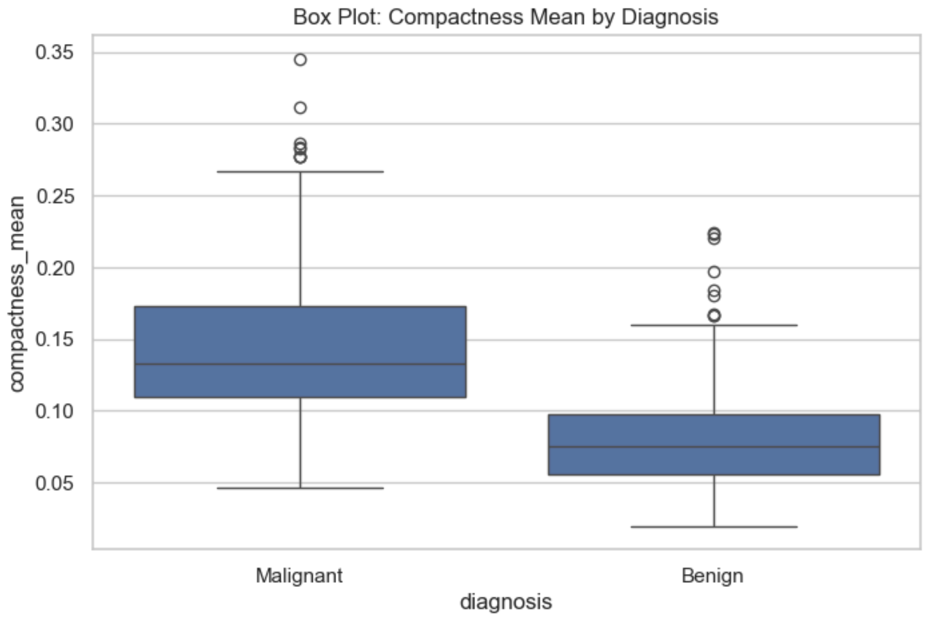
```
Inference: Malignant cases generally have higher values across all three selected features.
```

```
plt.figure(figsize=(40,40)) #12, 8 and annot= False
sns.heatmap(df.drop(columns=['diagnosis']).corr(), cmcmapcmcmapapap='coolwarm', annot=True)
plt.title('Feature Correlation Heatmap')
plt.show()
print("Inference: Strong correlation between radius, perimeter, and area, indicating redundancy in some features.")
```



```
Inference: Strong correlation between radius, perimeter, and area, indicating redundancy in some features.
```

```
plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x='diagnosis', y='compactness_mean')
plt.title('Box Plot: Compactness Mean by Diagnosis')
plt.show()
print("Inference: Malignant cases exhibit a wider range of compactness values compared to benign ones.")
```

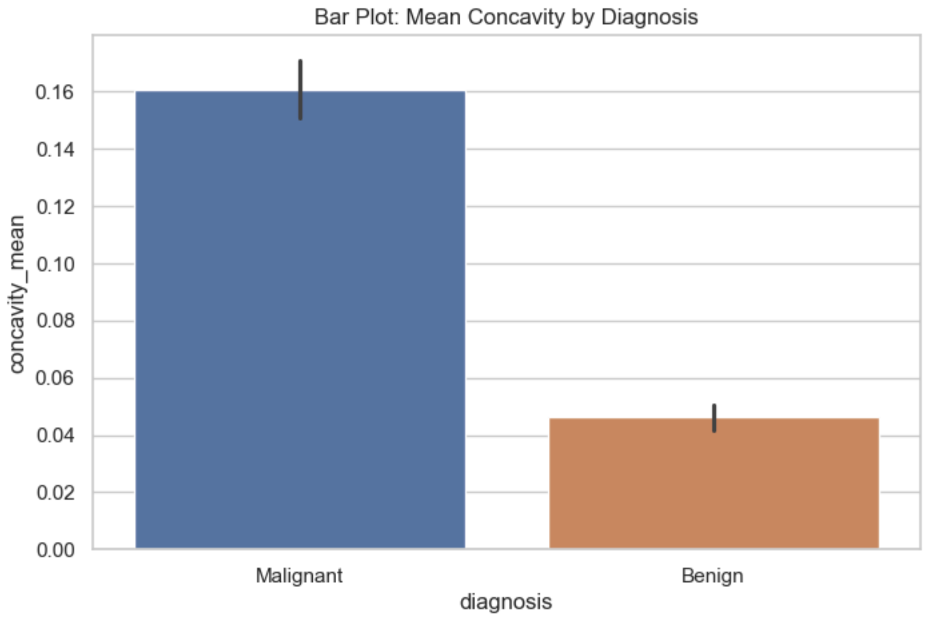


```
Inference: Malignant cases exhibit a wider range of compactness values compared to benign ones.
```

```
df.groupby('diagnosis').concavity_mean.mean()
```

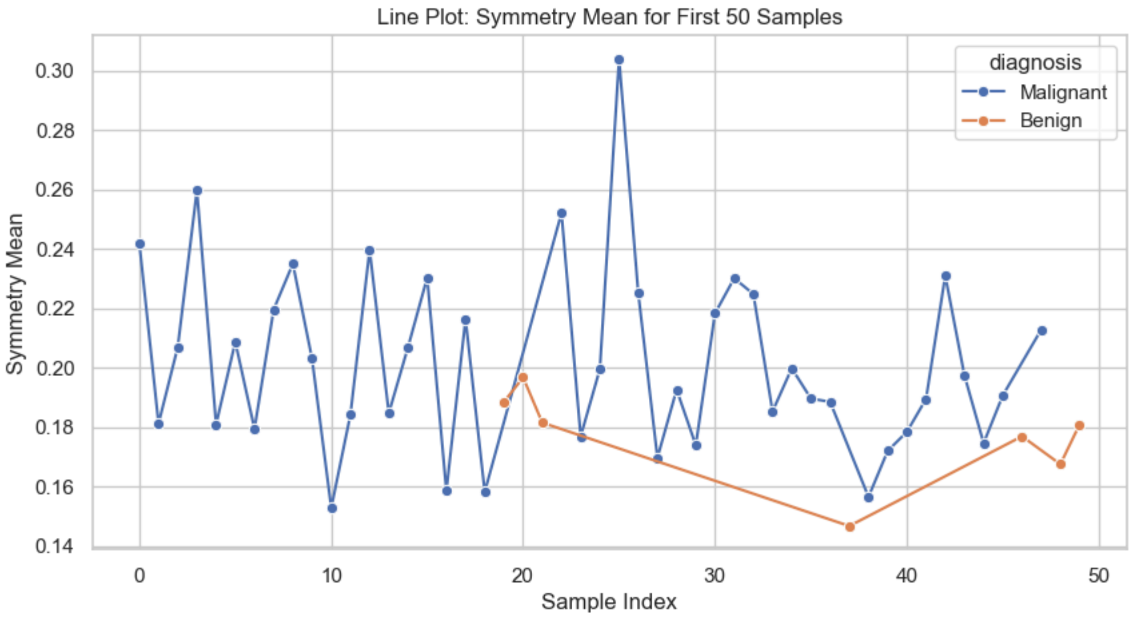
```
diagnosis
Benign       0.046235
Malignant    0.160775
Name: concavity_mean, dtype: float64
```

```
plt.figure(figsize=(8, 5))
sns.barplot(data=df, x='diagnosis', y='concavity_mean', hue='diagnosis')
plt.title('Bar Plot: Mean Concavity by Diagnosis')
plt.show()
print('Inference: On average, malignant tumors have a higher concavity value than benign ones.')
```

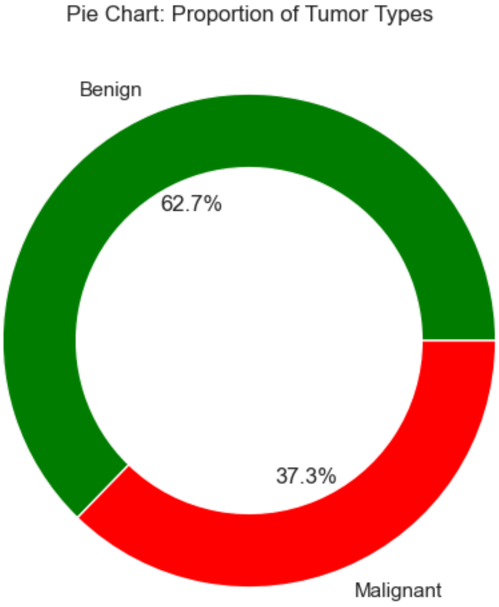


```
Inference: On average, malignant tumors have a higher concavity value than benign ones.
```

```
plt.figure(figsize=(10, 5))
sns.lineplot(data=df.iloc[:50], x=df.index[:50], y='symmetry_mean', marker='o', hue=df['diagnosis'].iloc[:50])
plt.title('Line Plot: Symmetry Mean for First 50 Samples')
plt.xlabel('Sample Index')
plt.ylabel('Symmetry Mean')
plt.show()
print("Inference: Fluctuations in symmetry values are visible across different samples.")
```



```
# Pie Chart: Distribution of Benign vs Malignant Cases
plt.figure(figsize=(6, 6))
sizes = df['diagnosis'].value_counts()
labels = sizes.index
plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=['green','red' ], wedgeprops=dict(width=0.3))
plt.title('Pie Chart: Proportion of Tumor Types')
plt.show()
print("Inference: More benign cases than malignant ones in the dataset.Benign (non-cancerous)and Malignant (cancerous) tumors.")
```



```
Inference: More benign cases than malignant ones in the dataset.Benign (non-cancerous)and Malignant (cancerous) tumors.
```

Start coding or generate with AI.