Exploratory Data Analysis on Breast Cancer

1 Data Collection

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

2 Importing Dataset

[3]:
```
df=pd.read_csv("C:/Users/aadhi/Desktop/Data_analyst/Datasets_All/Datasets_All/
↪Classification/breast-cancer.csv")
```

[5]:
```
df
```

[5]:

	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	
..	…	…	…	…	…	…	
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	

	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	\
0	0.11840	0.27760	0.30010	0.14710	
1	0.08474	0.07864	0.08690	0.07017	
2	0.10960	0.15990	0.19740	0.12790	
3	0.14250	0.28390	0.24140	0.10520	
4	0.10030	0.13280	0.19800	0.10430	
..	…	…	…	…	
564	0.11100	0.11590	0.24390	0.13890	
565	0.09780	0.10340	0.14400	0.09791	

```
566          0.08455           0.10230         0.09251              0.05302
567          0.11780           0.27700         0.35140              0.15200
568          0.05263           0.04362         0.00000              0.00000

     …  radius_worst  texture_worst  perimeter_worst  area_worst  \
0    …        25.380          17.33           184.60      2019.0
1    …        24.990          23.41           158.80      1956.0
2    …        23.570          25.53           152.50      1709.0
3    …        14.910          26.50            98.87       567.7
4    …        22.540          16.67           152.20      1575.0
..   …           …              …                …          …
564  …        25.450          26.40           166.10      2027.0
565  …        23.690          38.25           155.00      1731.0
566  …        18.980          34.12           126.70      1124.0
567  …        25.740          39.42           184.60      1821.0
568  …         9.456          30.37            59.16       268.6

     smoothness_worst  compactness_worst  concavity_worst  \
0             0.16220            0.66560           0.7119
1             0.12380            0.18660           0.2416
2             0.14440            0.42450           0.4504
3             0.20980            0.86630           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            0.86810           0.9387
568           0.08996            0.06444           0.0000

     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
..                    …               …                       …
564                0.2216          0.2060                  0.07115
565                0.1628          0.2572                  0.06637
566                0.1418          0.2218                  0.07820
567                0.2650          0.4087                  0.12400
568                0.0000          0.2871                  0.07039

[569 rows x 32 columns]
```

[7]:
```
df.isnull().sum()
```

[7]:
```
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
```

The data is already clean (no null values) and ready for analysis.

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

[11]:
```
0
```

The data is already clean (no duplicate) and ready for analysis.

[13]:
```
df.info
```

[13]:
```
<bound method DataFrame.info of            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
..        …       …           …             …               …          …
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

     smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0            0.11840           0.27760         0.30010              0.14710
1            0.08474           0.07864         0.08690              0.07017
2            0.10960           0.15990         0.19740              0.12790
3            0.14250           0.28390         0.24140              0.10520
4            0.10030           0.13280         0.19800              0.10430
..               …                 …               …                    …
564          0.11100           0.11590         0.24390              0.13890
565          0.09780           0.10340         0.14400              0.09791
566          0.08455           0.10230         0.09251              0.05302
567          0.11780           0.27700         0.35140              0.15200
568          0.05263           0.04362         0.00000              0.00000

     …  radius_worst  texture_worst  perimeter_worst  area_worst  \
0    …        25.380          17.33           184.60      2019.0
1    …        24.990          23.41           158.80      1956.0
2    …        23.570          25.53           152.50      1709.0
3    …        14.910          26.50            98.87       567.7
4    …        22.540          16.67           152.20      1575.0
..   …           …               …                …          …
564  …        25.450          26.40           166.10      2027.0
565  …        23.690          38.25           155.00      1731.0
566  …        18.980          34.12           126.70      1124.0
567  …        25.740          39.42           184.60      1821.0
568  …         9.456          30.37            59.16       268.6

     smoothness_worst  compactness_worst  concavity_worst  \
0             0.16220            0.66560           0.7119
1             0.12380            0.18660           0.2416
2             0.14440            0.42450           0.4504
3             0.20980            0.86630           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            0.86810           0.9387
```

```
568           0.08996            0.06444           0.0000

     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
..                    …               …                       …
564                0.2216          0.2060                  0.07115
565                0.1628          0.2572                  0.06637
566                0.1418          0.2218                  0.07820
567                0.2650          0.4087                  0.12400
568                0.0000          0.2871                  0.07039

[569 rows x 32 columns]>
```

Let's Dive into the Breast Cancer Dataset!

This dataset is a treasure trove of information about breast cancer tumors. It's a table with 32 columns and 569 rows, each representing a patient's data.

Here's what we can find in each column:

- A unique ID for each patient
- The diagnosis result (M for malignant or B for benign)
- Various measurements of the cell nuclei, like radius, texture, and area
- Shape and texture attributes, like smoothness, compactness, and concavity
- The worst-case values for these attributes

[23]:
```
df.dtypes
```

[23]:
```
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
dtype: object
```

[17]:
```
df.describe()
```

[17]:

	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]
```

Summary statistics of the same breast cancer dataset.

- **count**: The number of non-missing values for each column.
- **mean**: The average value for each column.
- **std**: The standard deviation, showing how spread out the values are.
- **min**: The minimum value for each column.
- **25%**: The 25th percentile, or the value below which 25% of the data falls.
- **50% (Median)**: The median value, or the value below which 50% of the data falls.
- **75%**: The 75th percentile, or the value below which 75% of the data falls.
- **max**: The maximum value for eac free to ask!

[40]:
```
df.head()
```

[40]:
```
         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]
```

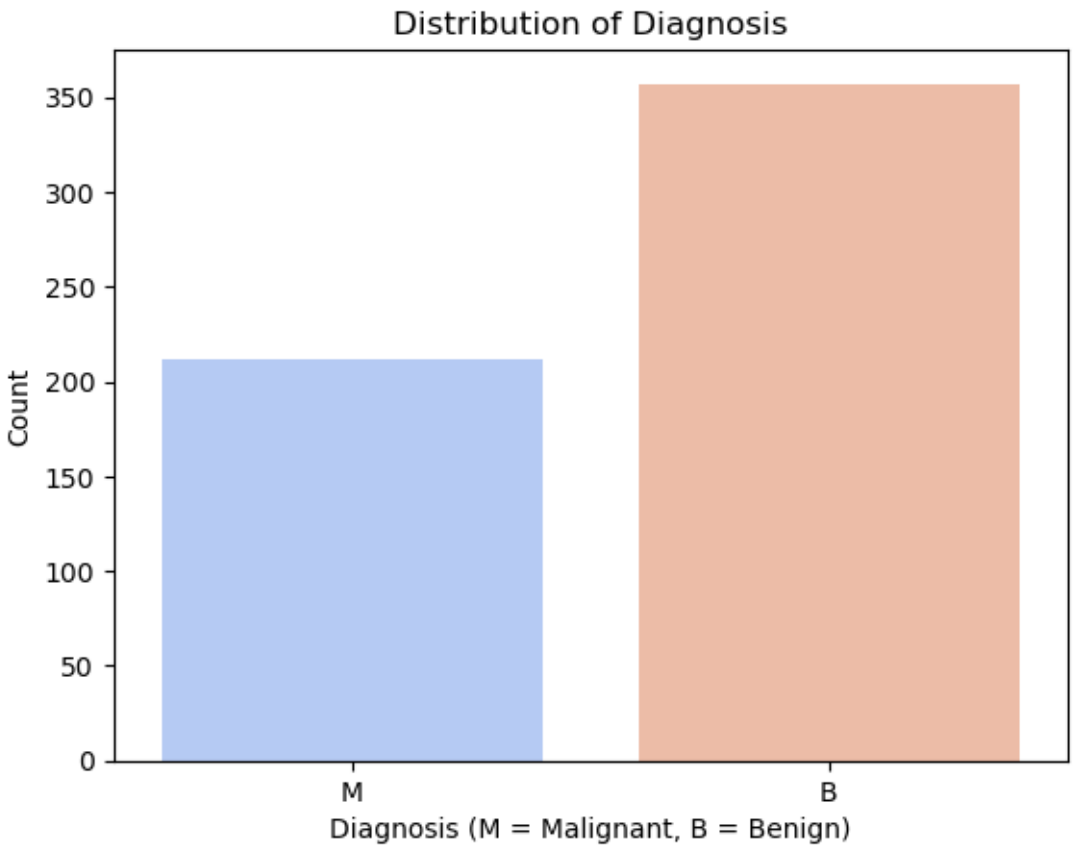
[54]:
```
df.columns
```

[54]:
```
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'],
      dtype='object')
```

3 Visualization

[38]:
```
sns.countplot(x='diagnosis', data=df, palette='coolwarm')
plt.title('Distribution of Diagnosis')
plt.xlabel('Diagnosis (M = Malignant, B = Benign)')
plt.ylabel('Count')
plt.show()
```

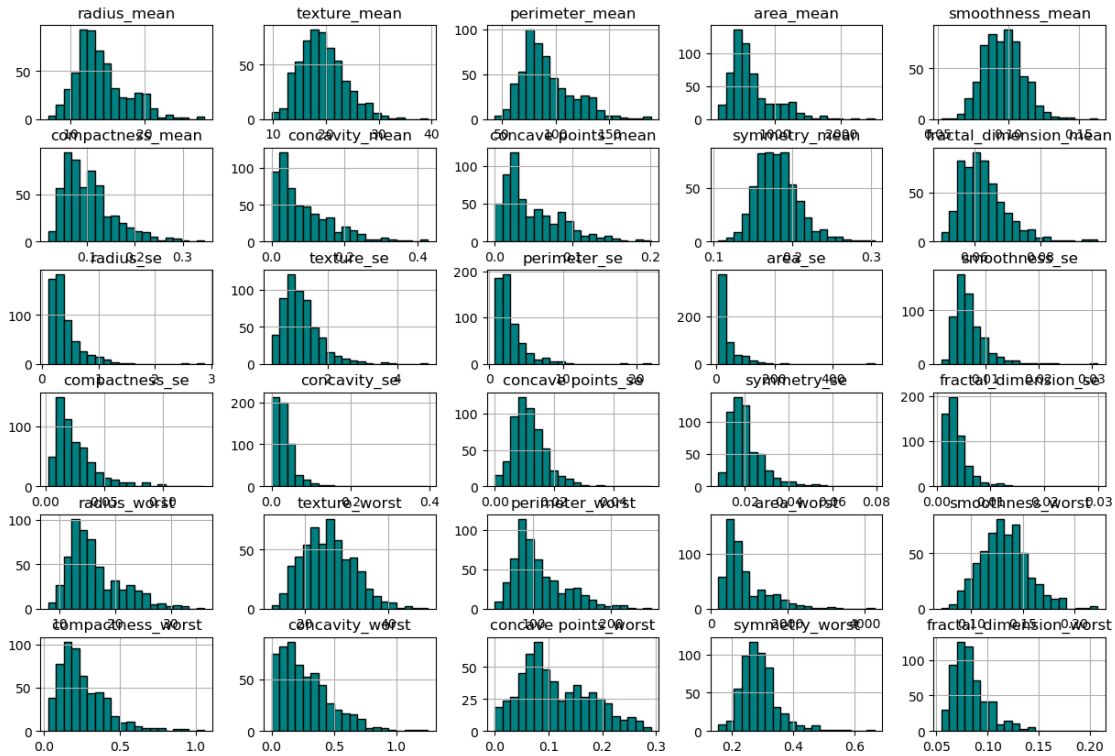
```
C:\Users\aadhi\AppData\Local\Temp\ipykernel_30016\2047936065.py:1:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.countplot(x='diagnosis', data=df, palette='coolwarm')
```



This dataset contains various attributes of breast tumors to predict malignancy or benignity, and its summary statistics provide an overview of central tendency, variability, and range for each attribute.

[52]:
```
numerical_columns = df.select_dtypes(include=['float64']).columns
df[numerical_columns].hist(bins=20, figsize=(15, 10), color='teal',␣
↪edgecolor='black')
plt.suptitle('Distributions of Numerical Features', fontsize=16)
```

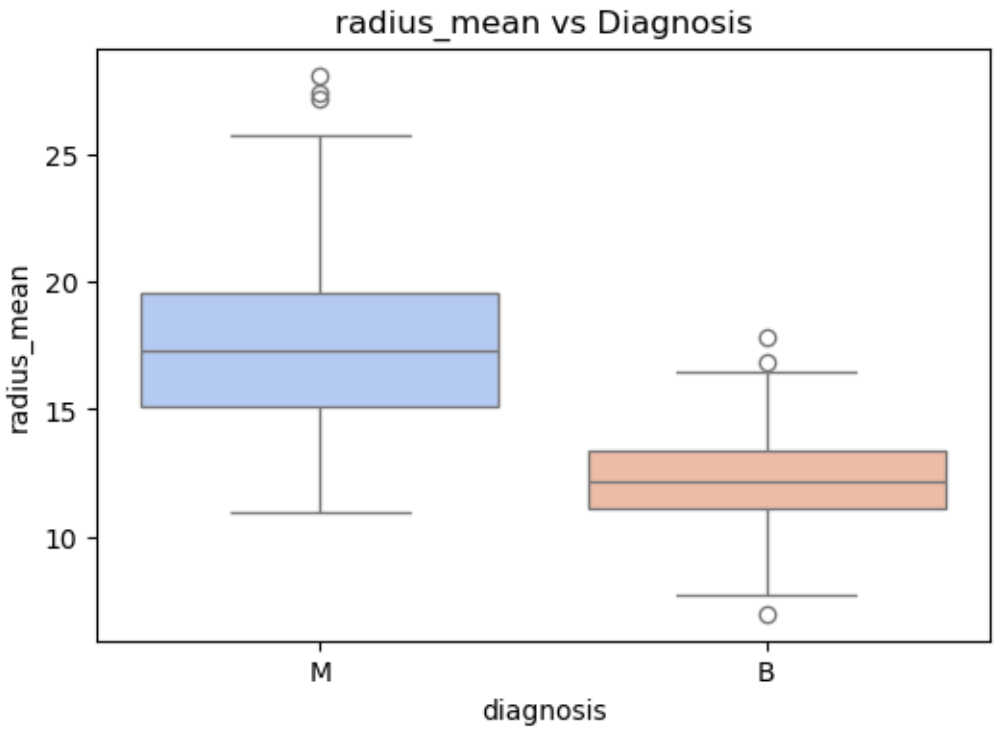
[52]:
```
Text(0.5, 0.98, 'Distributions of Numerical Features')
```



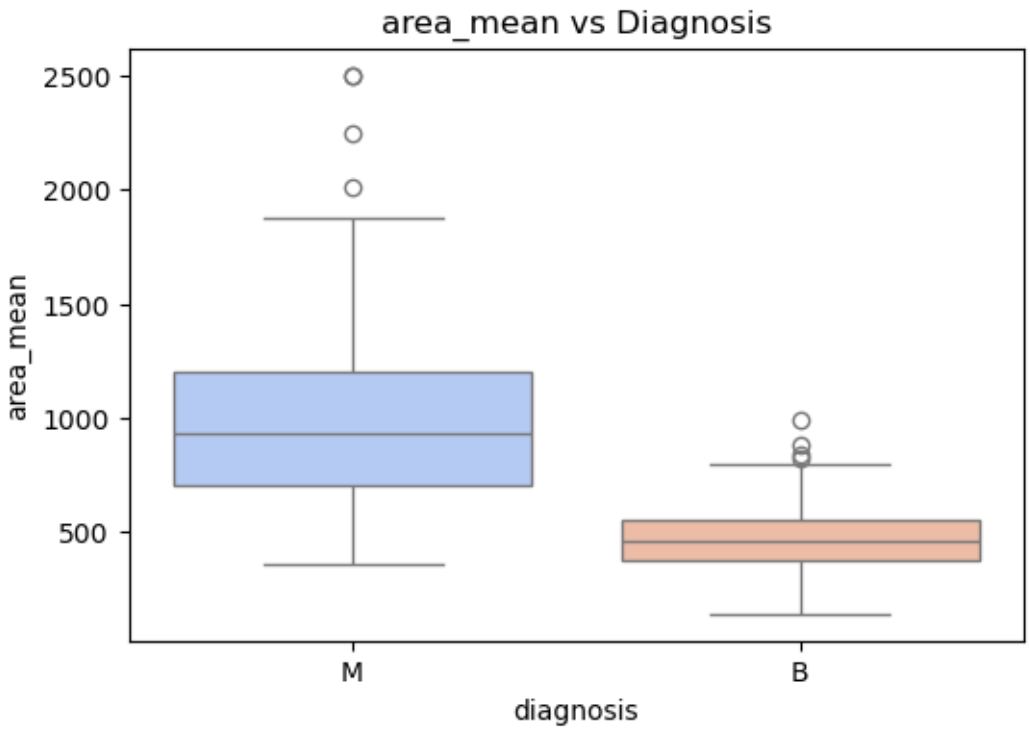
The average tumor radius is around 14 units, and it can range from about 7 to 28 units. Similarly, the texture average is approximately 19, with a variability that spreads widely. Understanding these statistics helps us grasp the overall data trends and variations, which is crucial for predicting whether tumors are malignant or benign.

[72]:
```
top_features = ['radius_mean', 'area_mean', 'perimeter_mean']
for feature in top_features:
    plt.figure(figsize=(6, 4))
    sns.boxplot(x='diagnosis', y=feature, data=df, palette='coolwarm')
    plt.title(f'{feature} vs Diagnosis')
    plt.show()
```

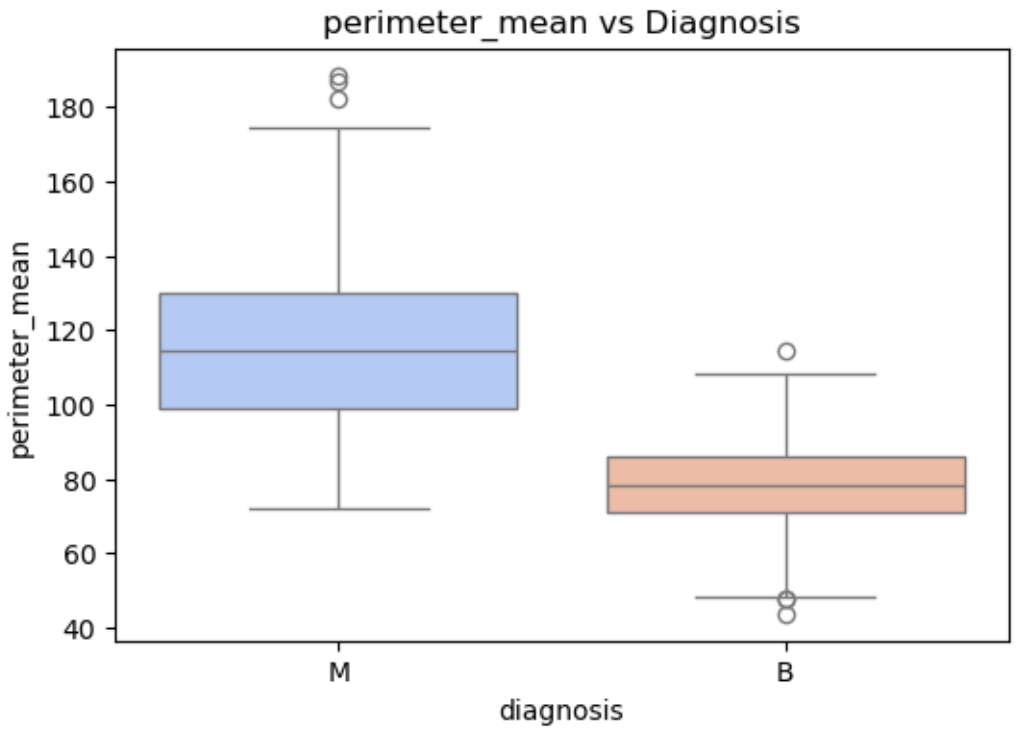
```
C:\Users\aadhi\AppData\Local\Temp\ipykernel_30016\2222074365.py:4:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(x='diagnosis', y=feature, data=df, palette='coolwarm')
```




```
C:\Users\aadhi\AppData\Local\Temp\ipykernel_30016\2222074365.py:4:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(x='diagnosis', y=feature, data=df, palette='coolwarm')
```




```
C:\Users\aadhi\AppData\Local\Temp\ipykernel_30016\2222074365.py:4:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(x='diagnosis', y=feature, data=df, palette='coolwarm')
```



Perimeter Mean: Malignant Tumors Stand Out

Malignant tumors tend to have higher perimeter values than benign ones. This boxplot shows the distribution, median, and variability of perimeter values for both groups.

Area Mean: Malignant Tumors Have More Extreme Values

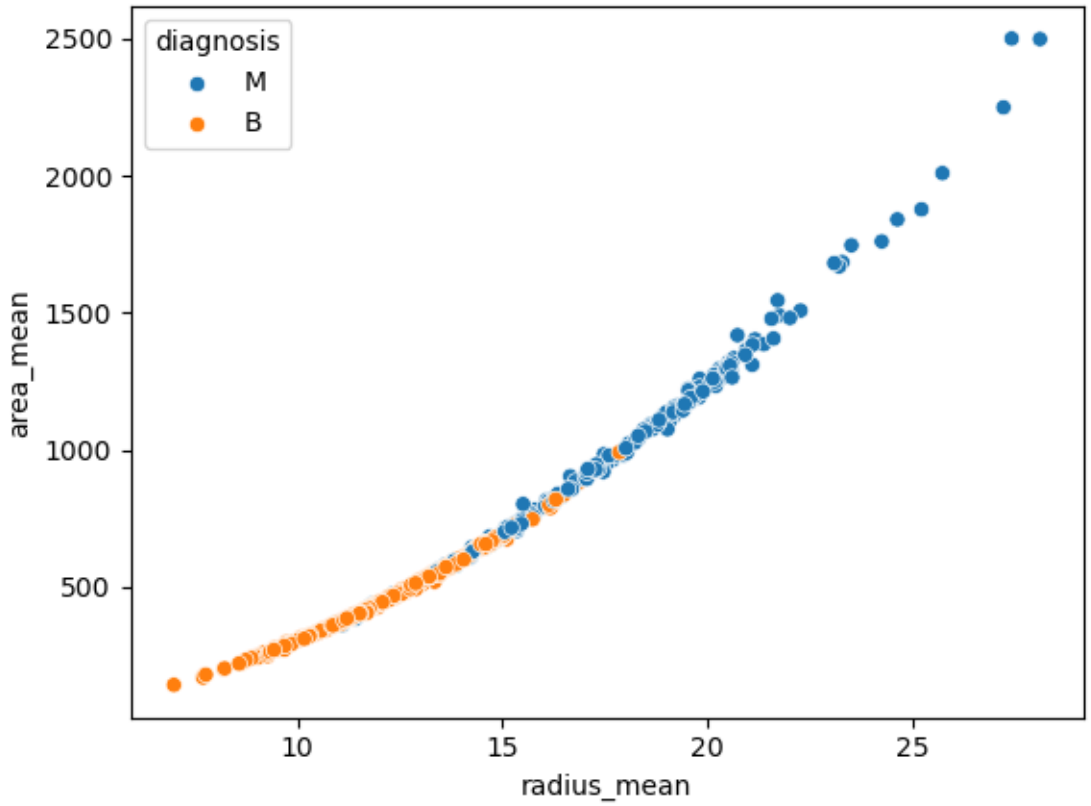
Malignant tumors generally have higher area values than benign ones. This boxplot highlights more extreme outliers in malignant cases, giving us a clearer picture of the differences between the two groups.

Radius Mean: Outliers Present in Both Groups

Malignant tumors tend to have higher radius values than benign ones. This boxplot reveals the presence of outliers in both groups, reminding us to consider these unusual cases when analyzing the data.

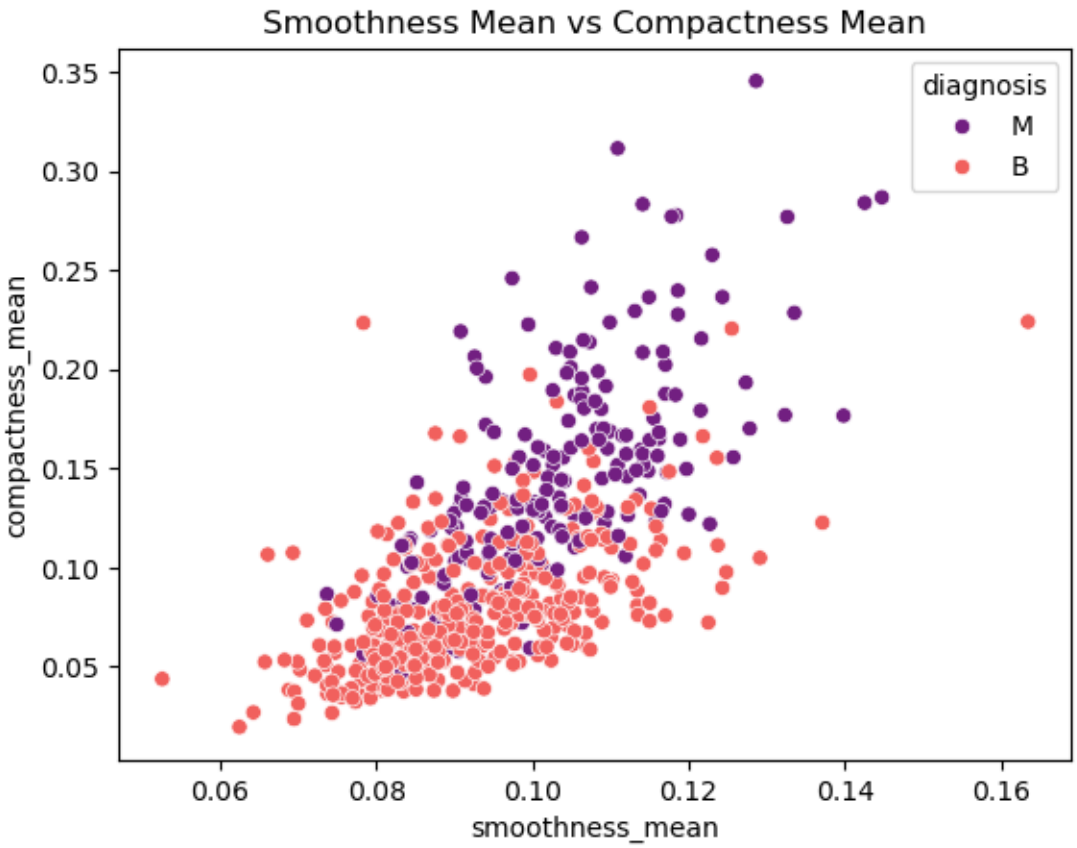
[83]:
```
sns.scatterplot(x=df['radius_mean'], y=df['area_mean'], hue=df['diagnosis'])
```

[83]:
```
<Axes: xlabel='radius_mean', ylabel='area_mean'>
```



Comparing `radius_mean` (x-axis) and `area_mean` (y-axis) for malignant (M) and benign (B) tumor diagnoses. Malignant tumors tend to have higher values for both attributes, with a noticeable separation between the two groups.

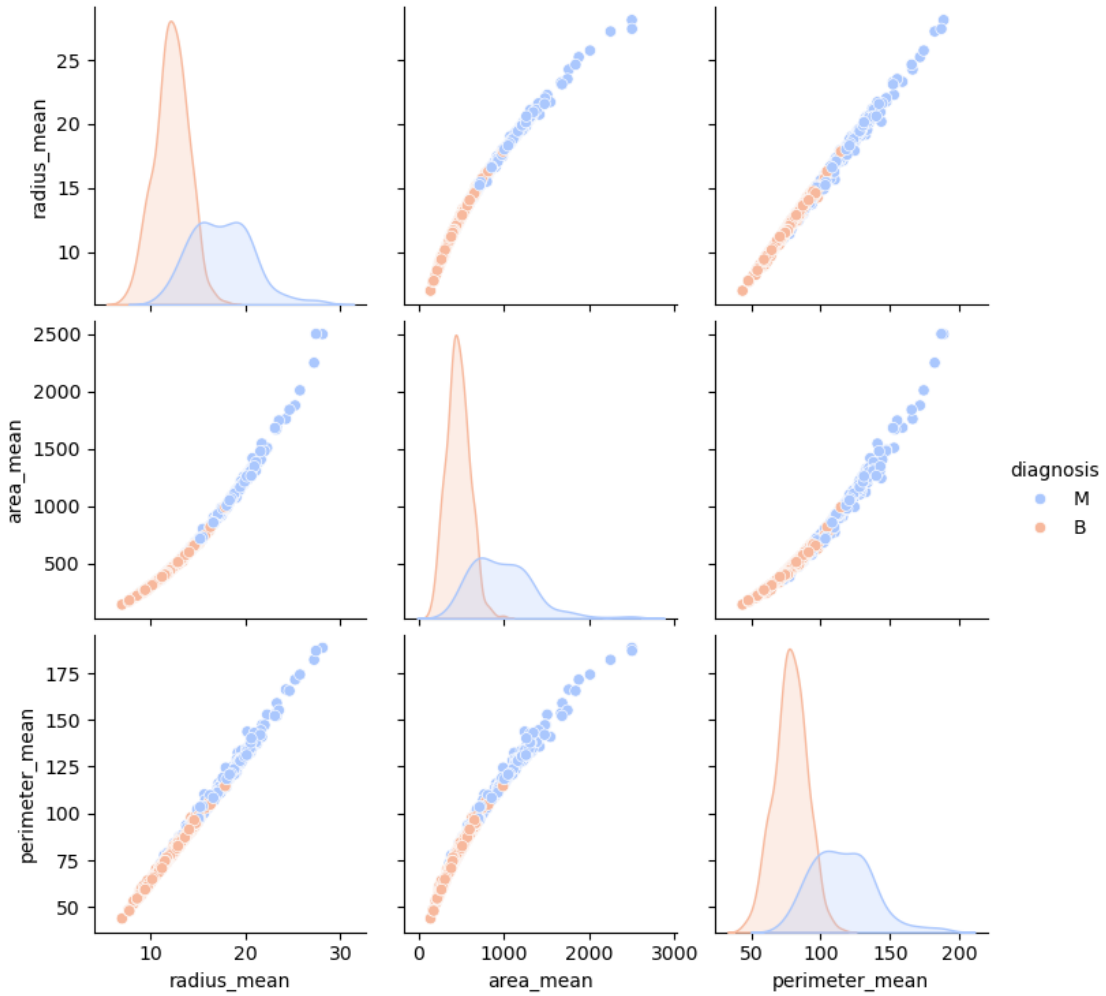
[95]:
```
sns.scatterplot(x='smoothness_mean', y='compactness_mean', hue='diagnosis',␣
↪data=df, palette='magma')
plt.title('Smoothness Mean vs Compactness Mean')
plt.show()
```



Comparing `smoothness_mean` (x-axis) and `compactness_mean` (y-axis) for malignant (M) and benign (B) tumor diagnoses. Malignant tumors generally have higher values for both attributes, with a noticeable separation between the two groups.

[87]:
```
sns.pairplot(df, vars=['radius_mean', 'area_mean', 'perimeter_mean'],␣
↪hue='diagnosis', palette='coolwarm')
```

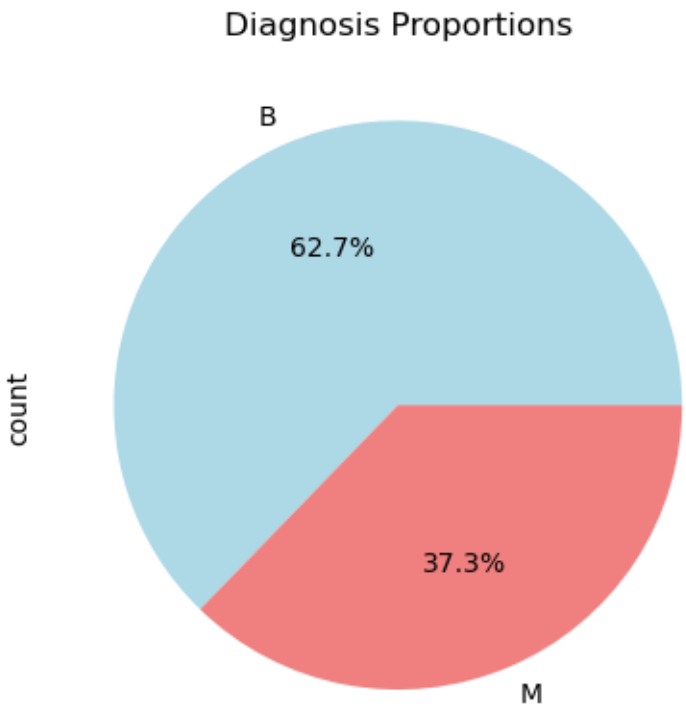
[87]:
```
<seaborn.axisgrid.PairGrid at 0x26944ab8ce0>
```



Relationships between radius_mean, area_mean, and perimeter_mean, color-coded by diagnosis (M for malignant and B for benign). Each subplot is either a scatter plot or a distribution plot, providing insights into variable distributions and relationships.

[89]:
```
df['diagnosis'].value_counts().plot.pie(autopct='%1.1f%%', colors=['lightblue',␣
↪'lightcoral'])
plt.title('Diagnosis Proportions')
```

[89]:
```
Text(0.5, 1.0, 'Diagnosis Proportions')
```



Proportions of two categories labeled “B” and “M.” The category “B” (benign) is represented in light blue and constitutes 62.7% of the total, while the category “M” (malignant) is represented in red and makes up 37.3% of the total. This chart visually represents the distribution of diagnoses, highlighting the prevalence of each diagnosis type in the dataset.

4 Unlocking Breast Cancer Secrets

We dug deep into the Breast Cancer dataset to find hidden clues that can help us predict outcomes.

What We Found

Malignant tumors tend to be bigger and have more irregular shapes. We also saw strong connections between some features.

There were some outliers, but they weren’t too far off. And, we noticed that there are more benign cases than malignant ones.

Important Features

Some features, like size and shape, are super important for predicting outcomes. We’ll make sure to focus on these.

Data Ready

We cleaned up the data and removed any unnecessary info. Now it’s ready for the next step.

What's Next 1. Split the data into training and testing groups. 2. Build models to predict outcomes. 3. Test and refine our models.

By doing this, we'll uncover more secrets about breast cancer and develop tools to help doctors make better predictions.

[ ]: