BREAST CANCER PREDICTION

1. DATA COLLECTION

In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
data=pd.read_csv("data.csv")
```

In [3]:
```
data
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	radius_worst	texture_worst	perimeter_worst	area_worst	smoothness_worst
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	...	25.38	17.33	184.60	2019.0	0.1622
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	...	24.99	23.41	158.80	1956.0	0.1238
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	...	23.57	25.53	152.50	1709.0	0.1444
3	84358401	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	...	14.910	26.50	98.87	567.7	0.2098
4	843786412	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	...	22.540	16.67	152.20	1575.0	0.1374
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	...	25.450	26.40	166.10	2027.0	0.1410
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	...	23.690	38.25	155.00	1731.0	0.1166
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10223	0.09251	0.05302	...	18.980	34.12	126.70	1124.0	0.1139
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	...	25.740	39.42	184.60	1821.0	0.1650
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	...	9.456	30.37	59.16	268.6	0.0892

569 rows × 32 columns

In [4]:
```
len(data.index),len(data.columns)
```

Out[4]:
```
(569, 32)
```

In [5]:
```
data.shape
```

Out[5]:
```
(569, 32)
```

In [6]:
```
data.head()
```

Out[6]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	radius_worst	texture_worst	perimeter_worst	area_worst	smoothness_worst
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	25.38	17.33	184.60	2019.0	0.1622
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	24.99	23.41	158.80	1956.0	0.1238
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	23.57	25.53	152.50	1709.0	0.1444
3	84358401	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	14.91	26.50	98.87	567.7	0.2098
4	843786412	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10400	...	22.54	16.67	152.20	1575.0	0.1374

5 rows × 32 columns

In [7]:
```
data.tail()
```

Out[7]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	radius_worst	texture_worst	perimeter_worst	area_worst	smoothness_worst
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.2439	0.13890	...	25.450	26.40	166.10	2027.0	0.1410
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.1440	0.09791	...	23.690	38.25	155.00	1731.0	0.1166
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10223	0.09251	0.05302	...	18.980	34.12	126.70	1124.0	0.1139
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.3514	0.15200	...	25.740	39.42	184.60	1821.0	0.1650
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	...	9.456	30.37	59.16	268.6	0.0892

5 rows × 32 columns

2. EXPLORING DATA ANALYSIS

In [8]:
```
data.info()
```

Out[8]:
```
<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
dtype: int64, memory usage: 142.8 KB
```

In [9]:
```
data.tail()
```

Out[9]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	radius_worst	texture_worst	perimeter_worst	area_worst	smoothness_worst
564	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
565	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
566	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
567	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
568	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False

569 rows × 32 columns

In [10]:
```
data.isna().any()
```

Out[10]:
```
id                         False
diagnosis                  False
radius_mean                False
texture_mean               False
perimeter_mean             False
area_mean                  False
smoothness_mean            False
compactness_mean           False
concavity_mean             False
concave points_mean        False
symmetry_mean              False
fractal_dimension_mean     False
radius_se                  False
texture_se                 False
perimeter_se               False
area_se                    False
smoothness_se              False
compactness_se             False
concavity_se               False
concave points_se          False
symmetry_se                False
fractal_dimension_se       False
radius_worst               False
texture_worst              False
perimeter_worst            False
area_worst                 False
smoothness_worst           False
compactness_worst          False
concavity_worst            False
concave points_worst       False
symmetry_worst             False
fractal_dimension_worst    False
dtype: bool
```

In [11]:
```
data.isna().sum()
```

Out[11]:
```
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
```

Get Object Features

In [12]:
```
data.describe()
```

Out[12]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	...	radius_worst	texture_worst	perimeter_worst	area_worst	smoothness_worst
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	...	569.000000	569.000000	569.000000	569.000000	569.000000
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	0.181162	...	16.269190	25.677223	107.261213	880.583	
std	1.250516e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	0.027414	...	4.833242	6.146258	33.602542	569.356	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	0.106000	...	7.930000	12.020000	50.410000	185.200	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	0.161900	...	13.010000	21.080000	84.110000	515.300	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.096070	0.092630	0.061540	0.033500	0.179200	...	14.970000	25.410000	97.660000	686.500	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	0.195700	...	18.790000	29.720000	125.400000	1084.000	
max	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	0.304000	...	36.040000	49.540000	251.200000	4254.000	

8 rows × 31 columns

In [13]:
```
data.describe(include='O')
```

Out[13]:

	diagnosis
count	569
unique	2
top	B
freq	357

To Know how many unique values

In [14]:
```
data.diagnosis.value_counts()
```

Out[14]:
```
B    357
M    212
Name: diagnosis, dtype: int64
```

In [15]:
```
data.head()
```

Out[15]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	radius_worst	texture_worst	perimeter_worst	area_worst	smoothness_w
0	842302	M	17.99	10.38	122.8	1001.0	0.11840	0.27760	0.3001	0.14710	...	25.38	17.33	184.6	2019.0	0.
1	842517	M	20.57	17.77	132.9	1326.0	0.08474	0.07864	0.0869	0.07017	...	24.99	23.41	158.8	1956.0	0.

2 rows × 32 columns

In [16]:
```
diagnosis_unique=data.diagnosis.unique()
```

In [17]:
```
diagnosis_unique
```

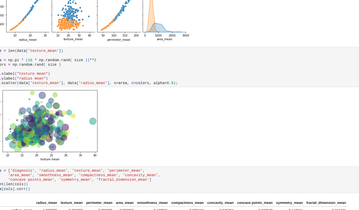
Out[17]:
```
array(['M', 'B'], dtype=object)
```

3. DATA VISUALIZATION

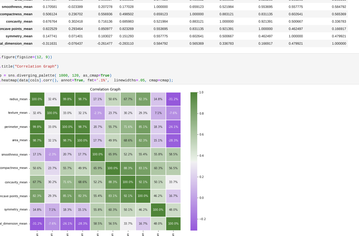
In [18]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

In [19]:
```
plt.figure(figsize=(15,5))

plt.subplot(1,2,1)
plt.hist(data.diagnosis)
plt.title('Count of Diagnosis')
plt.xlabel('Diagnosis')

plt.subplot(1,2,2)
sns.countplot('Diagnosis',data=data);

C:\Users\ASUS\AppDataSlim\Local\Temp\ipykernel_10404\2873.py:20: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional
argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

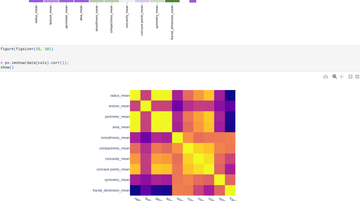
In [20]:
```
px.histogram(data, x='diagnosis')
```



In [21]:
```
sns.pairplot(data[cols], hue="diagnosis")
plt.show()
```



In [22]:
```
size = len(data['texture_mean'])

area = np.p[ * 10 / np.random.rand(* size )**2
colors = np.random.rand( size )

plt.xlabel("texture_mean")
plt.ylabel("radius_mean")
plt.scatter(data['texture_mean'], data['radius_mean'], s=area, c=colors, alpha=0.5);
```



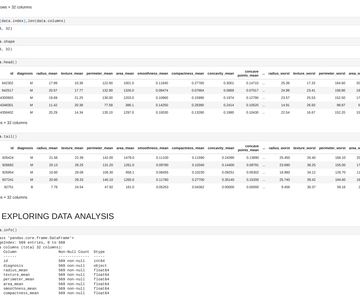
In [23]:
```
cols = ['diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
        'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
        'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean']
print(len(cols))
df=data[cols].dropna()
```

Out[23]:
```
11
```

	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	fractal_dimension_mean
radius_mean	1.000000	0.323782	0.997855	0.987357	0.170581	0.506124	0.676764	0.822529	0.147741	-0.311631
texture_mean	0.323782	1.000000	0.329533	0.321086	-0.023389	0.236702	0.302418	0.293464	0.071401	-0.076437
perimeter_mean	0.997855	0.329533	1.000000	0.986507	0.207278	0.556936	0.716136	0.850977	0.183027	-0.261477
area_mean	0.987357	0.321086	0.986507	1.000000	0.177028	0.498502	0.685983	0.823268	0.151293	-0.283102
smoothness_mean	0.170581	-0.023389	0.207278	0.177028	1.000000	0.659123	0.521984	0.553696	0.557775	0.584792
compactness_mean	0.506124	0.236702	0.556936	0.498502	0.659123	1.000000	0.883121	0.831135	0.602641	0.565369
concavity_mean	0.676764	0.302418	0.716136	0.685983	0.521984	0.883121	1.000000	0.921391	0.500667	0.336783
concave points_mean	0.822529	0.293464	0.850977	0.823268	0.553696	0.831135	0.921391	1.000000	0.462497	0.166617
symmetry_mean	0.147741	0.071401	0.183027	0.151293	0.557775	0.602641	0.500667	0.462497	1.000000	0.479921
fractal_dimension_mean	-0.311631	-0.076437	-0.261477	-0.283102	0.584792	0.565369	0.336783	0.166617	0.479921	1.000000

In [24]:
```
plt.figure(figsize=(15, 10))

plt.title('Correlation Graph')

cmap = sns.diverging_palette( 1000, 120, as_cmap=True)
sns.heatmap(corr,linewidths=.1,cmap=cmap, fmt='.2f', [,cbar=False, ax=ax, cmap=cmap];
```


In [25]:
```
plt.figure(figsize=(15, 10))

fig = px.imshow(data[cols].corr())
fig.show()
```




```
<Figure size 1080x720 with 0 Axes>
```