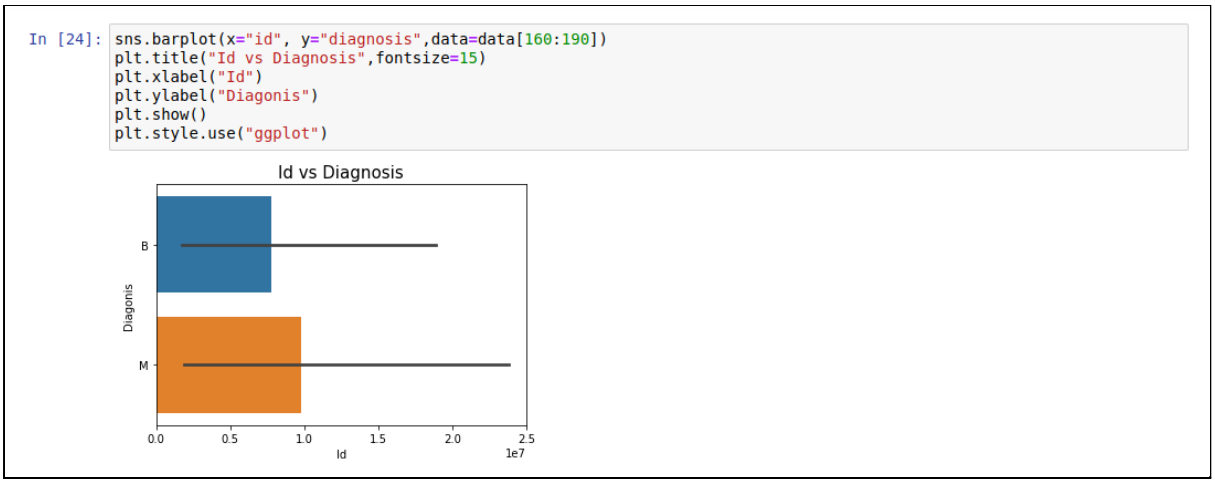
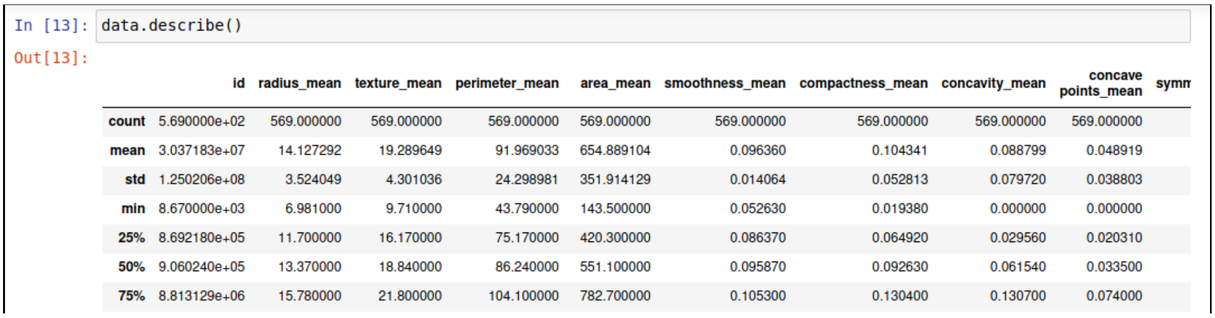


```
In [24]: sns.barplot(x="id", y="diagnosis",data=data[160:190])
         plt.title("Id vs Diagnosis",fontsize=15)
         plt.xlabel("Id")
         plt.ylabel("Diagonis")
         plt.show()
         plt.style.use("ggplot")
```



```
In [13]: data.describe()
Out[13]:
                   id  radius_mean  texture_mean  perimeter_mean   area_mean  smoothness_mean  compactness_mean  concavity_mean     concave  symm
                                                                                                                                points_mean
count    5.690000e+02   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
  std    1.250206e+08     3.524049      4.301036       24.298981  351.914129         0.014064          0.052813        0.079720    0.038803
  min    8.670000e+03     6.981000      9.710000       43.790000  143.500000         0.052630          0.019380        0.000000    0.000000
  25%    8.692180e+05    11.700000     16.170000       75.170000  420.300000         0.086370          0.064920        0.029560    0.020310
  50%    9.060240e+05    13.370000     18.840000       86.240000  551.100000         0.095870          0.092630        0.061540    0.033500
  75%    8.813129e+06    15.780000     21.800000      104.100000  782.700000         0.105300          0.130400        0.130700    0.074000
```

Project title: Analysis and prediction of breast cancer

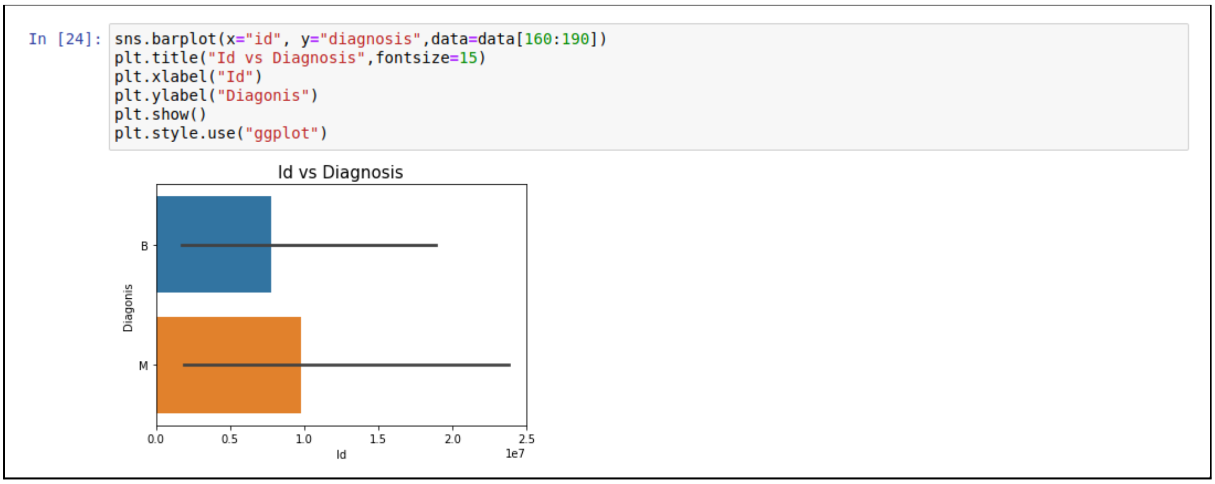
Team Name: Patterns n Parameters

Team member Name	SRN
Sanjana Murthy	PES2UG19CS364
T. Sunaina	PES2UG19CS427
Susan Mathew K	PES2UG19CS416
Toshani Rungta	PES2UG19CS433

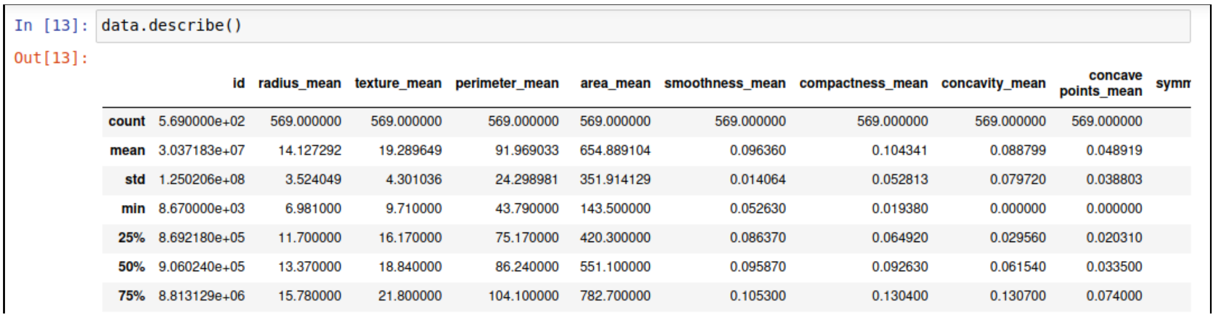
1. Dataset Name and Description.

Our dataset is the Breast Cancer Wisconsin(Diagnostic) Data Set. Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. The attribute Information includes ID number, Diagnosis (M = malignant, B = benign), ten real-valued features are computed for each cell nucleus: radius (mean of distances from center to points on the perimeter); texture (standard deviation of gray-scale values); perimeter; area; smoothness (local variation in radius lengths); compactness (perimeter^2 / area - 1.0); concavity (severity of concave portions of the contour); concave points (number of concave portions of the contour; symmetry; fractal dimension ("coastline approximation" - 1)

The mean, standard error and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius. All feature values are recorded with four significant digits. There are no missing attribute values in the dataset and it has a class distribution of 357 benign and 212 malignant masses.



```
In [24]: sns.barplot(x="id", y="diagnosis",data=data[160:190])
         plt.title("Id vs Diagnosis",fontsize=15)
         plt.xlabel("Id")
         plt.ylabel("Diagonis")
         plt.show()
         plt.style.use("ggplot")
```



```
In [13]: data.describe()
```

Out[13]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symm
count	5.690000e+02	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	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	

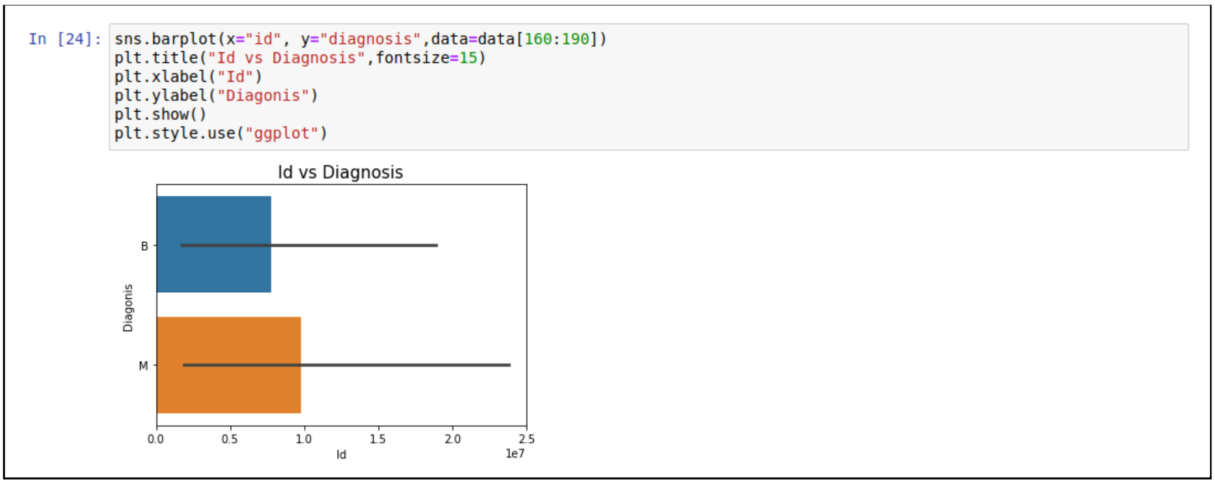
2. Problem statement:

Breast cancer is a malignant tumor that occurs in females with the highest incidence, which has serious adverse effects on a woman's health. Therefore, early and accurate diagnosis of breast cancer patients is extremely crucial to recovery.
We chose the Wisconsin Breast Cancer(Diagnostic) Data Set for our study to classify and predict if the patient has breast cancer.

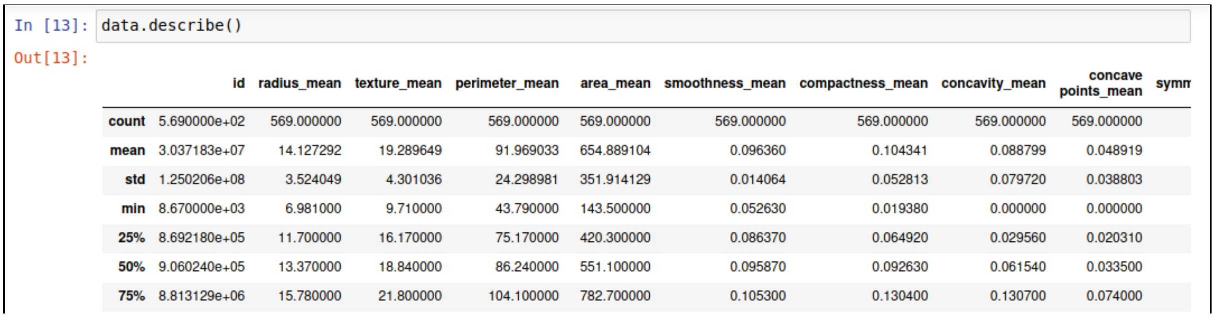
3. EDA and Visualization

- How many rows and attributes?
  569 rows and 33 attributes
- How many missing data and outliers?
  Zero missing values
- Any inconsistent, incomplete, duplicate or incorrect data?
  No
- Are the variables correlated to each other?
  Yes
- Are any of the preprocessing techniques needed: dimensionality reduction, range transformation, standardization, etc.?

  No
- Does PCA help visualize the data? Do we get any insights from histograms/ bar charts/ line plots, etc.?
  we did not have to do PCA for our dataset



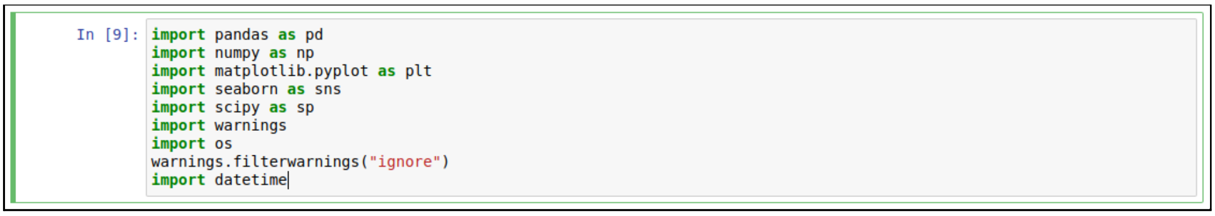
```
In [24]: sns.barplot(x="id", y="diagnosis",data=data[160:190])
         plt.title("Id vs Diagnosis",fontsize=15)
         plt.xlabel("Id")
         plt.ylabel("Diagonis")
         plt.show()
         plt.style.use("ggplot")
```



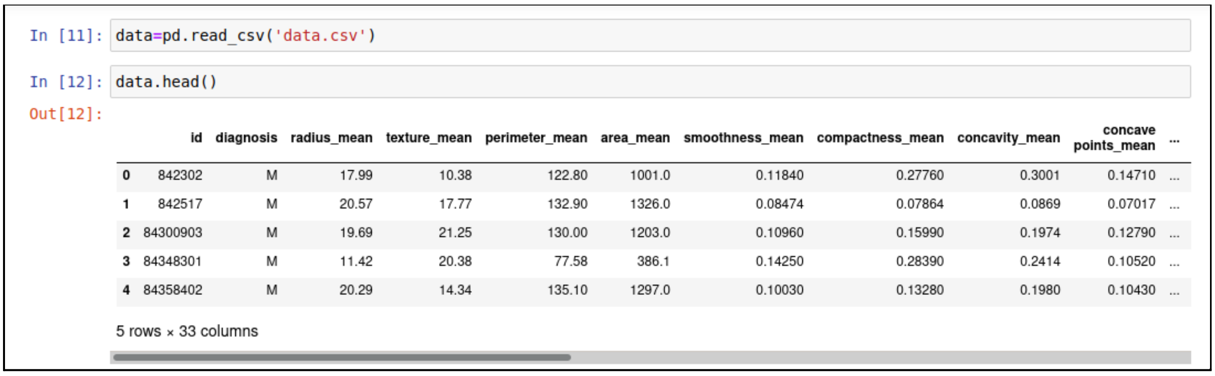
```
In [13]: data.describe()
```

Out[13]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symm
count	5.690000e+02	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	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	



```
In [9]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        import scipy as sp
        import warnings
        import os
        warnings.filterwarnings("ignore")
        import datetime
```

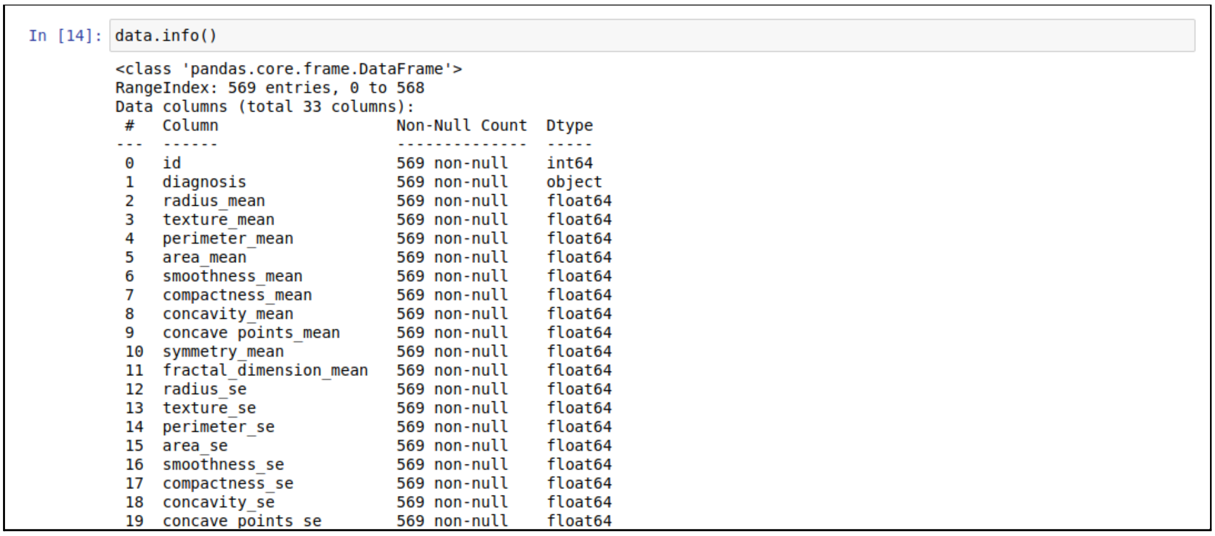


```
In [11]: data=pd.read_csv('data.csv')

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

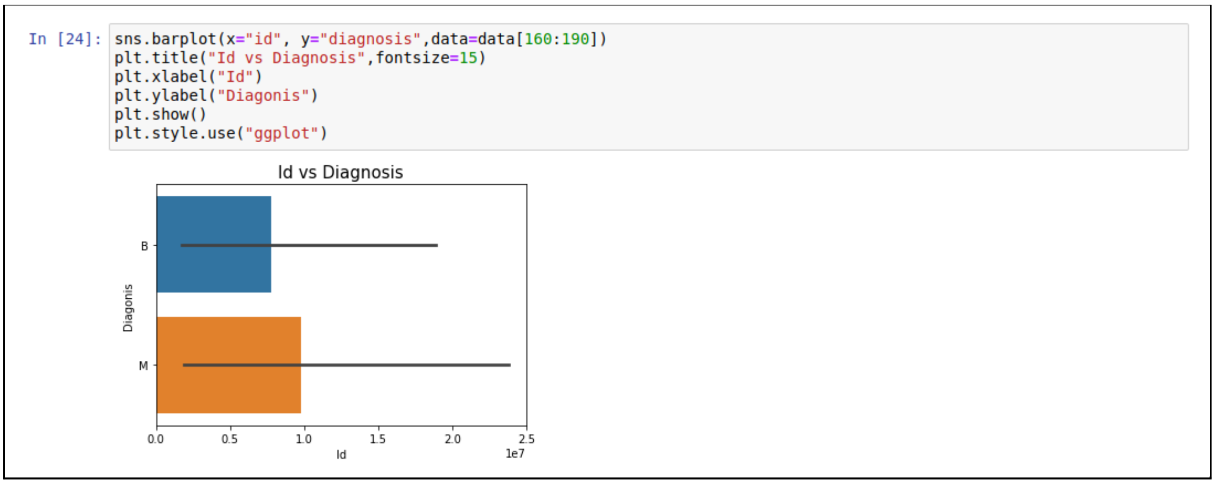
Out[12]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...

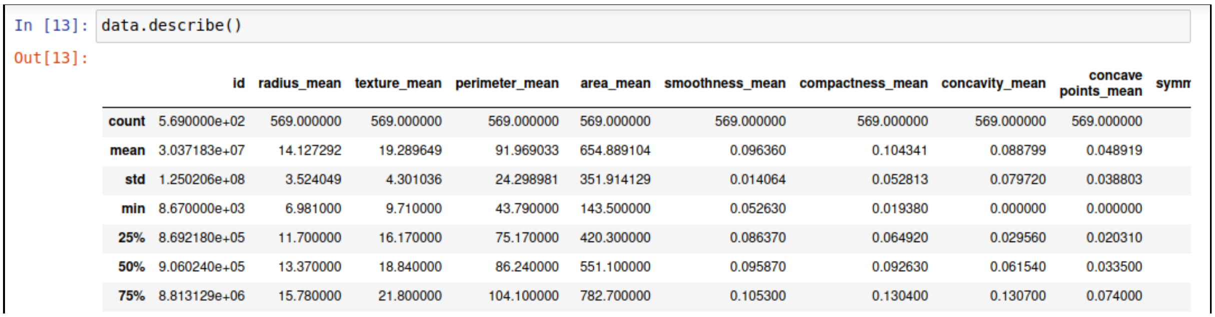
5 rows × 33 columns



```
In [14]: data.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
```

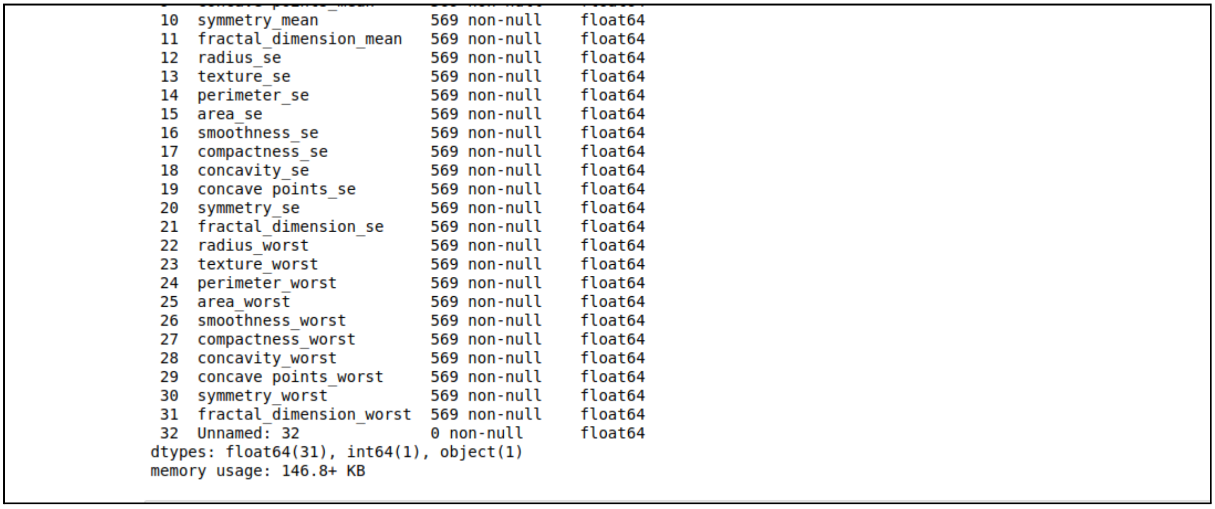


```
In [24]: sns.barplot(x="id", y="diagnosis",data=data[160:190])
         plt.title("Id vs Diagnosis",fontsize=15)
         plt.xlabel("Id")
         plt.ylabel("Diagonis")
         plt.show()
         plt.style.use("ggplot")
```

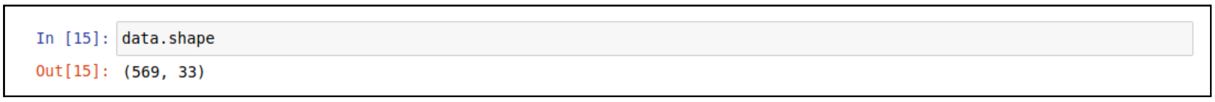


```
In [13]: data.describe()
Out[13]:
```

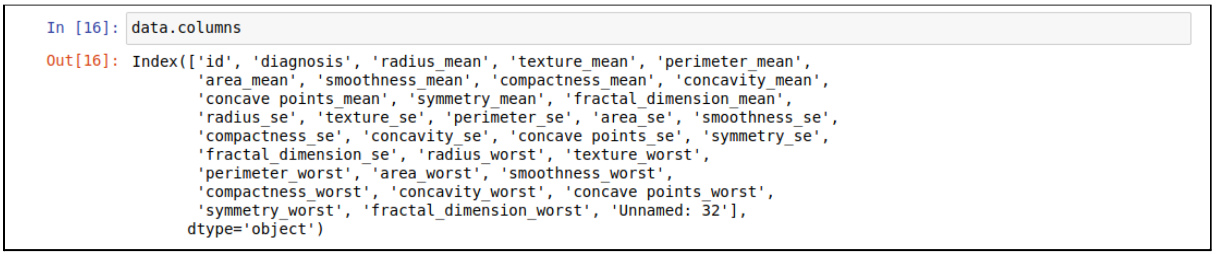
	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symm
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	



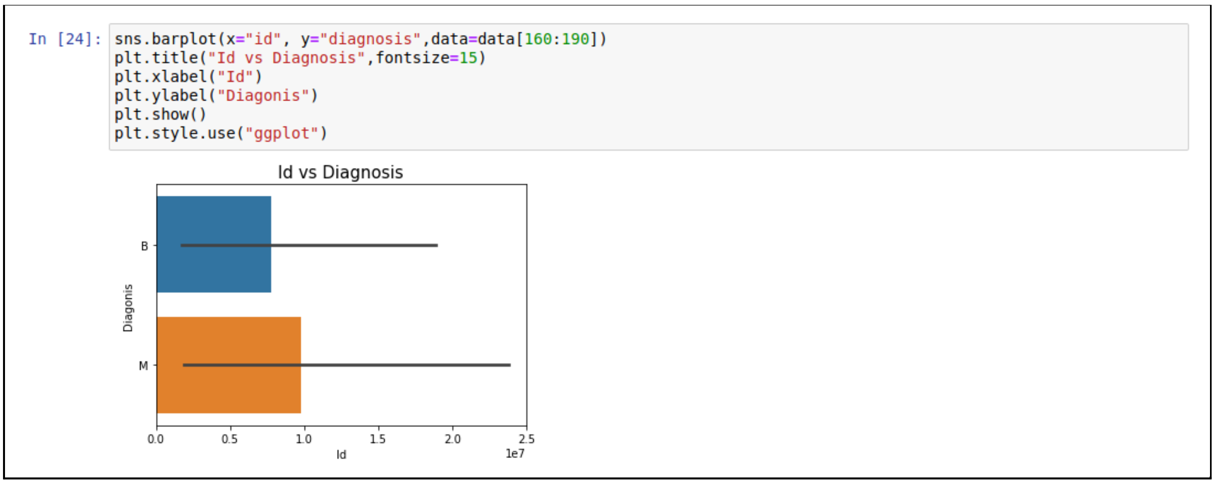
```
    10  symmetry_mean            569 non-null    float64
    11  fractal_dimension_mean   569 non-null    float64
    12  radius_se                569 non-null    float64
    13  texture_se               569 non-null    float64
    14  perimeter_se             569 non-null    float64
    15  area_se                  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
```



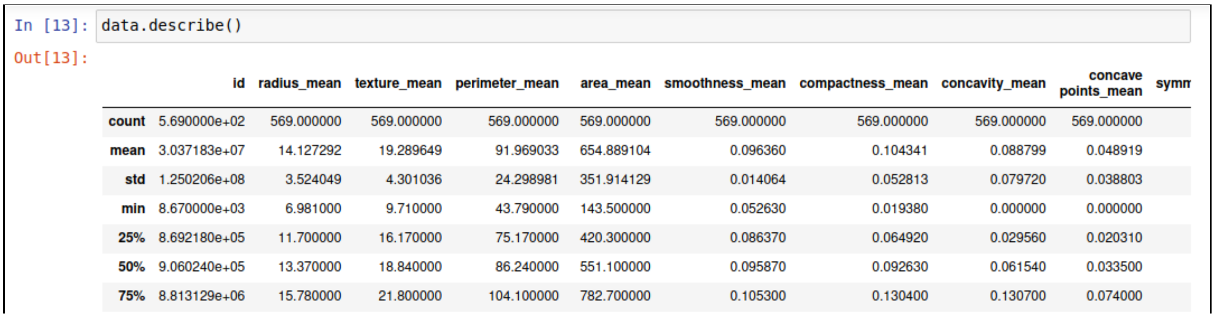
```
In [15]: data.shape
Out[15]: (569, 33)
```



```
In [16]: data.columns
Out[16]: 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')
```



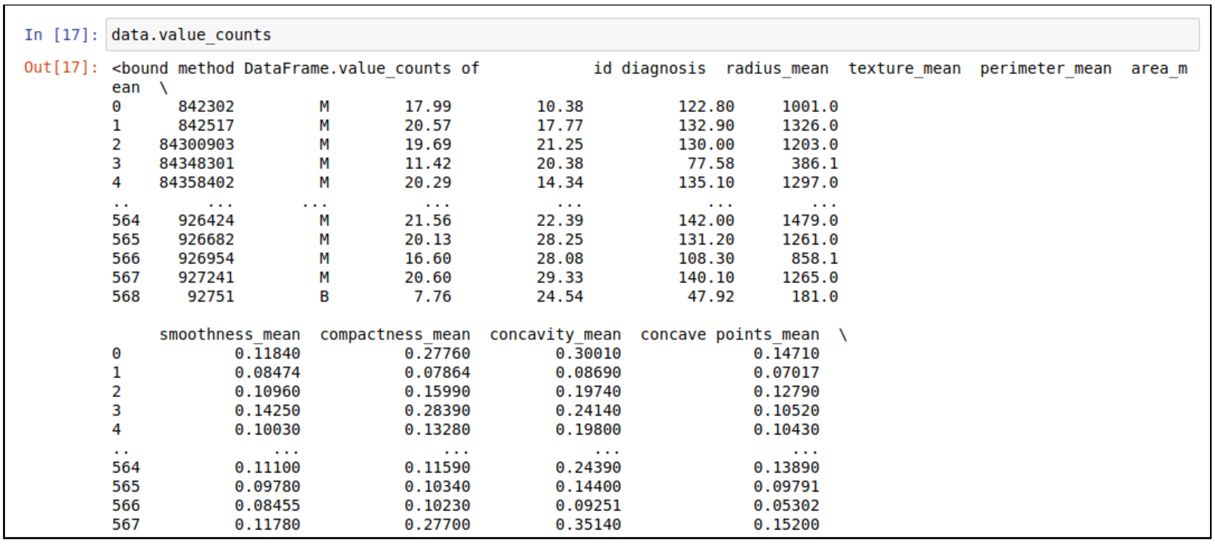
```
In [24]: sns.barplot(x="id", y="diagnosis",data=data[160:190])
         plt.title("Id vs Diagnosis",fontsize=15)
         plt.xlabel("Id")
         plt.ylabel("Diagonis")
         plt.show()
         plt.style.use("ggplot")
```



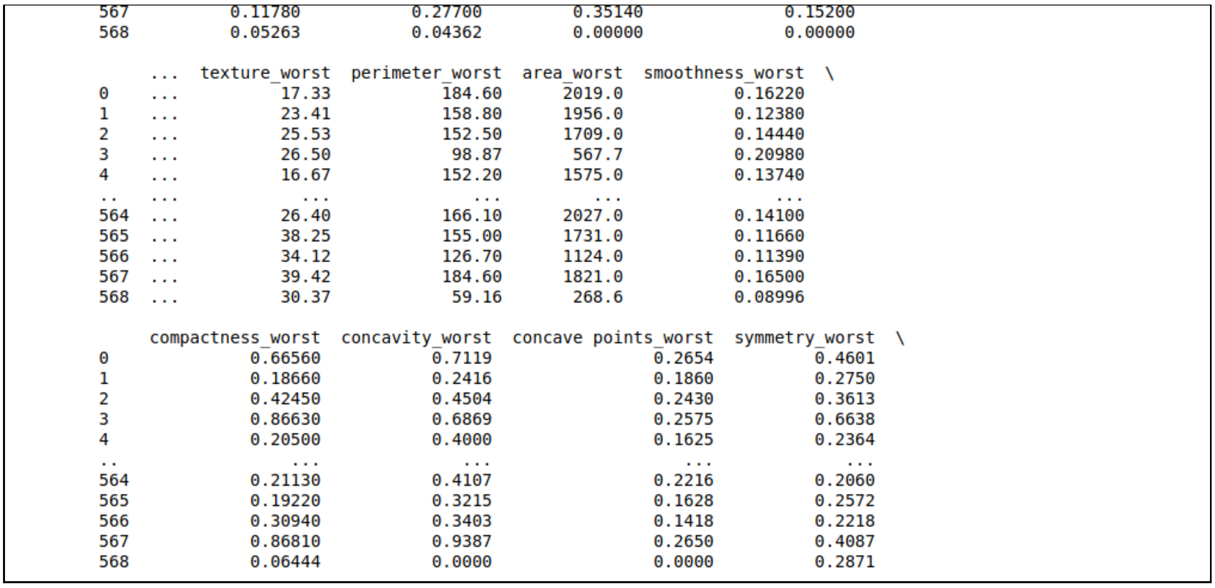
```
In [13]: data.describe()
```

Out[13]:

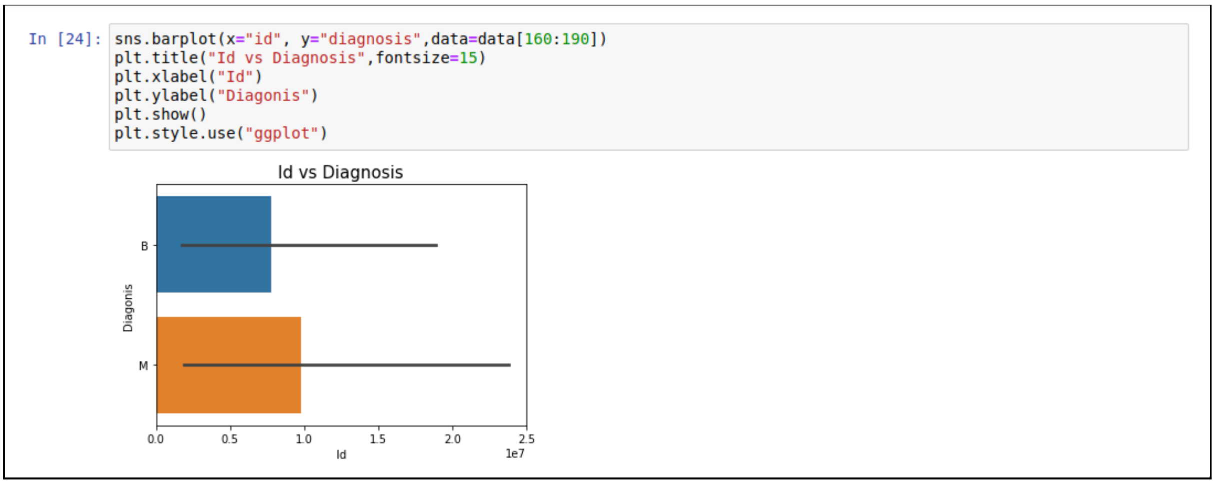
	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symm
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	



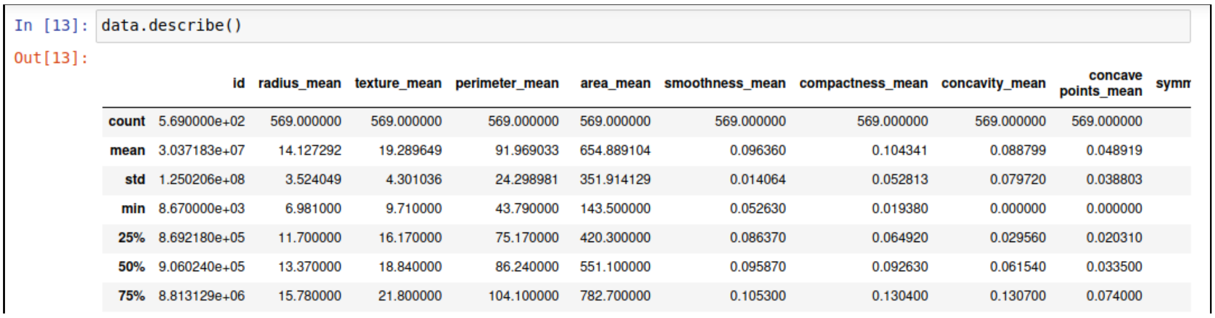
```
In [17]: data.value_counts

Out[17]: <bound method DataFrame.value_counts of            id diagnosis  radius_mean  texture_mean  perimeter_mean  area_m
         ean  \
         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
```



```
567          0.11780           0.27700         0.35140              0.15200
568          0.05263           0.04362         0.00000              0.00000

     ...  texture_worst  perimeter_worst  area_worst  smoothness_worst  \
0    ...          17.33           184.60      2019.0           0.16220
1    ...          23.41           158.80      1956.0           0.12380
2    ...          25.53           152.50      1709.0           0.14440
3    ...          26.50            98.87       567.7           0.20980
4    ...          16.67           152.20      1575.0           0.13740
..   ...            ...              ...         ...               ...
564  ...          26.40           166.10      2027.0           0.14100
565  ...          38.25           155.00      1731.0           0.11660
566  ...          34.12           126.70      1124.0           0.11390
567  ...          39.42           184.60      1821.0           0.16500
568  ...          30.37            59.16       268.6           0.08996

     compactness_worst  concavity_worst  concave points_worst  symmetry_worst  \
0              0.66560           0.7119                0.2654          0.4601
1              0.18660           0.2416                0.1860          0.2750
2              0.42450           0.4504                0.2430          0.3613
3              0.86630           0.6869                0.2575          0.6638
4              0.20500           0.4000                0.1625          0.2364
..                 ...              ...                   ...             ...
564            0.21130           0.4107                0.2216          0.2060
565            0.19220           0.3215                0.1628          0.2572
566            0.30940           0.3403                0.1418          0.2218
567            0.86810           0.9387                0.2650          0.4087
568            0.06444           0.0000                0.0000          0.2871
```



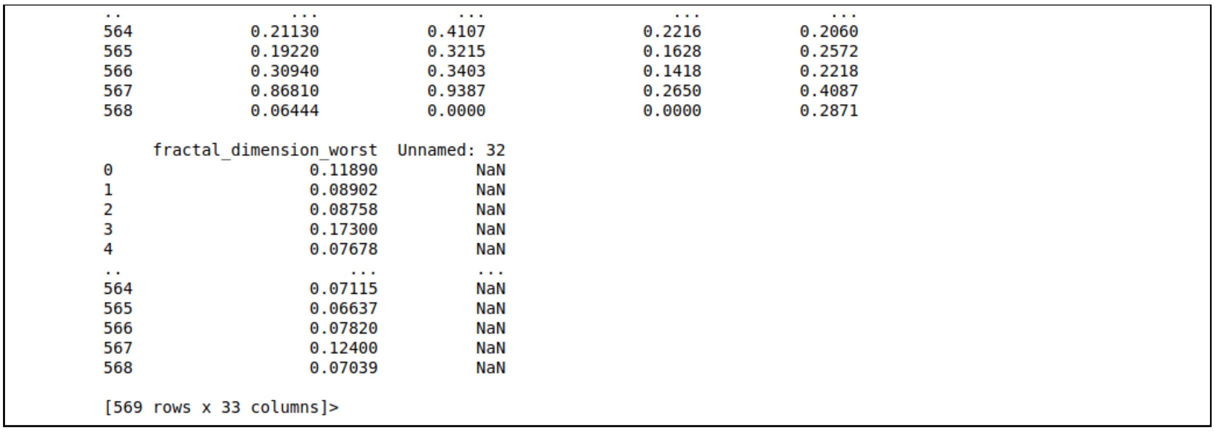
```
In [24]: sns.barplot(x="id", y="diagnosis",data=data[160:190])
         plt.title("Id vs Diagnosis",fontsize=15)
         plt.xlabel("Id")
         plt.ylabel("Diagonis")
         plt.show()
         plt.style.use("ggplot")
```



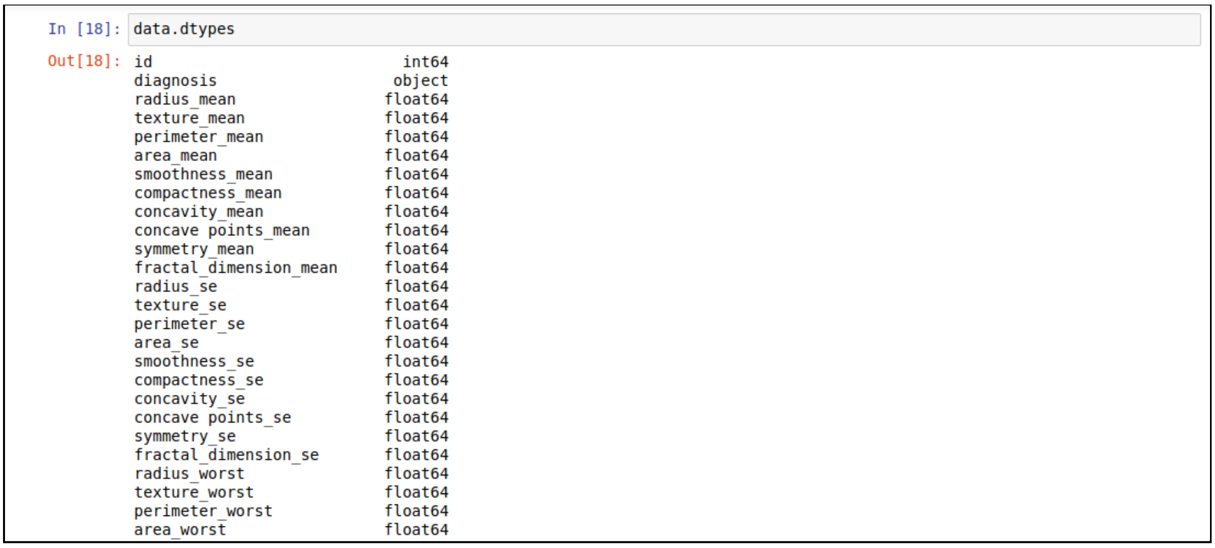
```
In [13]: data.describe()
```

Out[13]:

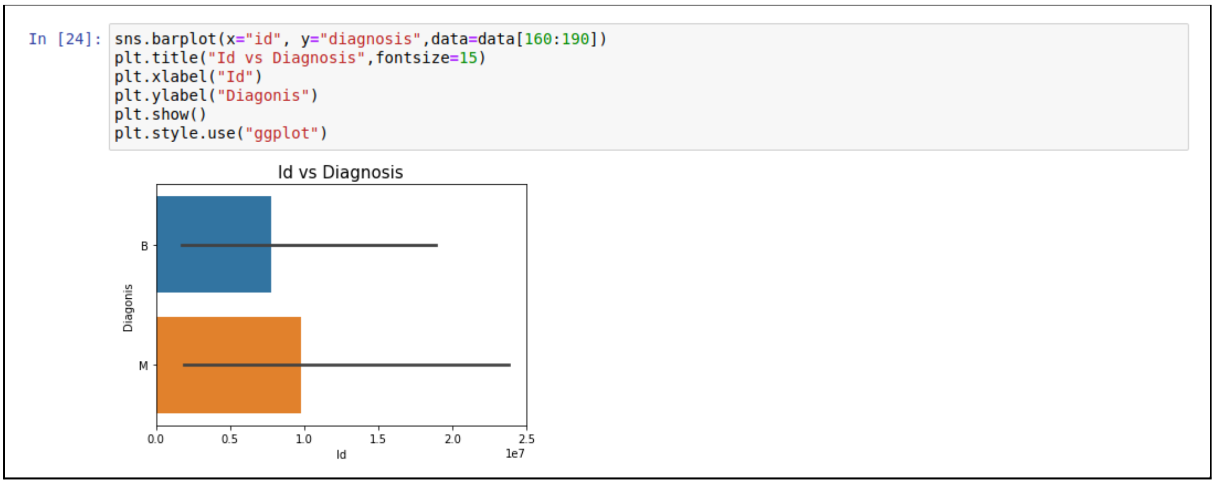
	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symm
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	



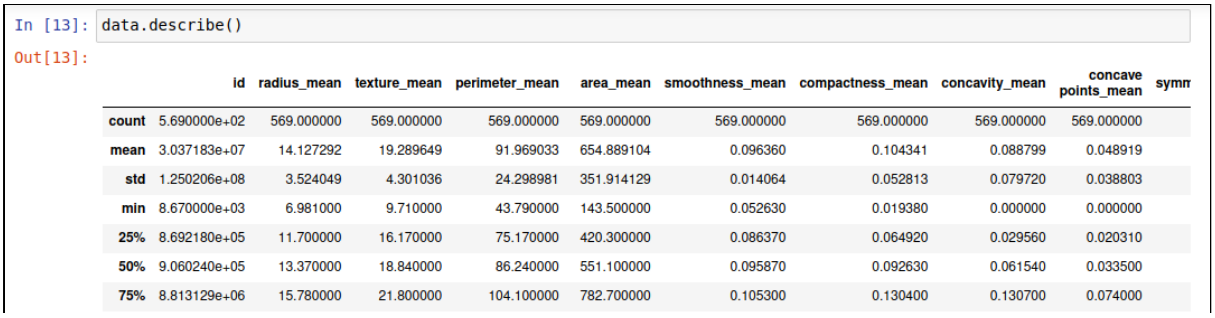
```
..                ...              ...                    ...              ...
564           0.21130           0.4107                0.2216           0.2060
565           0.19220           0.3215                0.1628           0.2572
566           0.30940           0.3403                0.1418           0.2218
567           0.86810           0.9387                0.2650           0.4087
568           0.06444           0.0000                0.0000           0.2871

     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
..                       ...          ...
564                  0.07115          NaN
565                  0.06637          NaN
566                  0.07820          NaN
567                  0.12400          NaN
568                  0.07039          NaN

[569 rows x 33 columns]>
```



```
In [18]: data.dtypes

Out[18]: 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
```

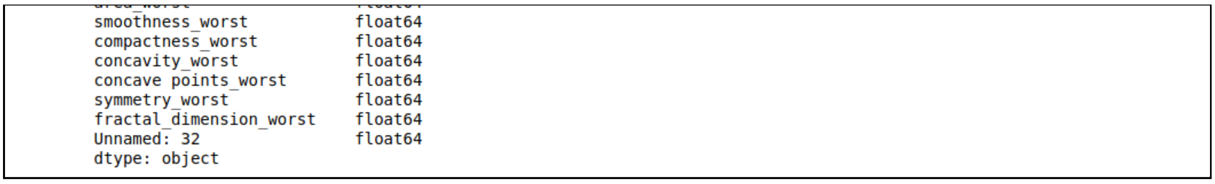


```
In [24]: sns.barplot(x="id", y="diagnosis",data=data[160:190])
         plt.title("Id vs Diagnosis",fontsize=15)
         plt.xlabel("Id")
         plt.ylabel("Diagonis")
         plt.show()
         plt.style.use("ggplot")
```

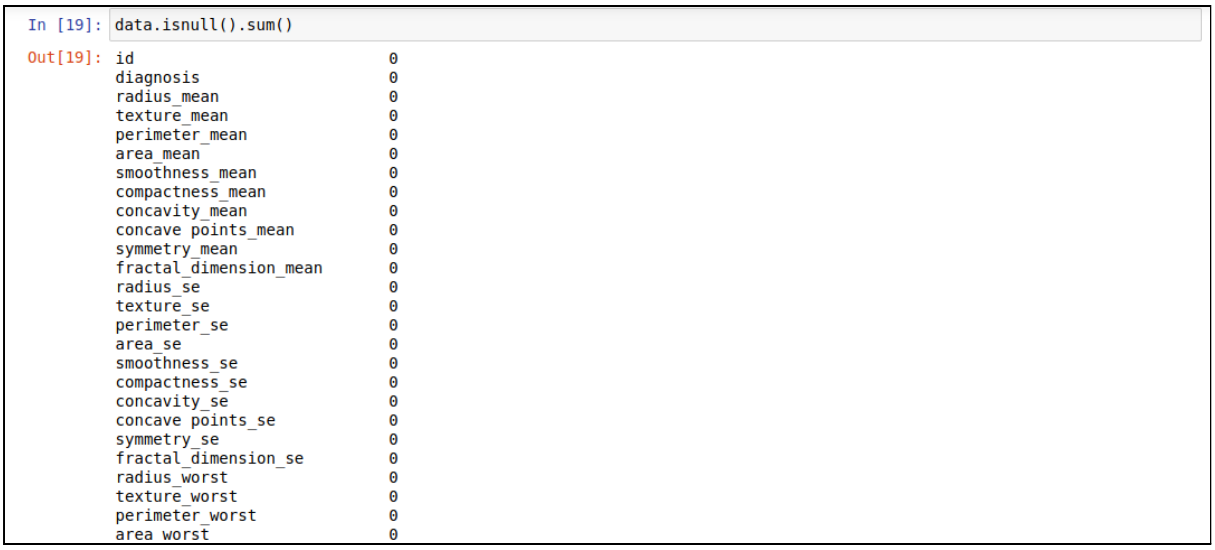


```
In [13]: data.describe()
Out[13]:
```

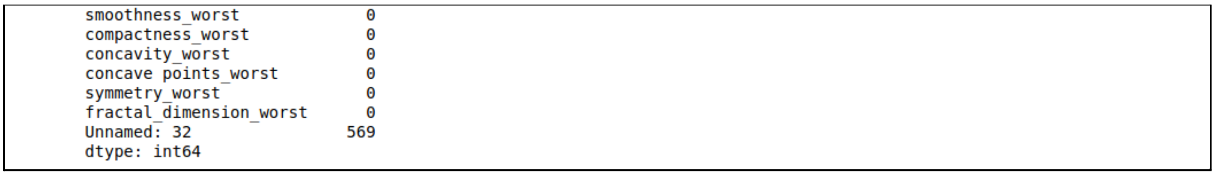
	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symm
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	



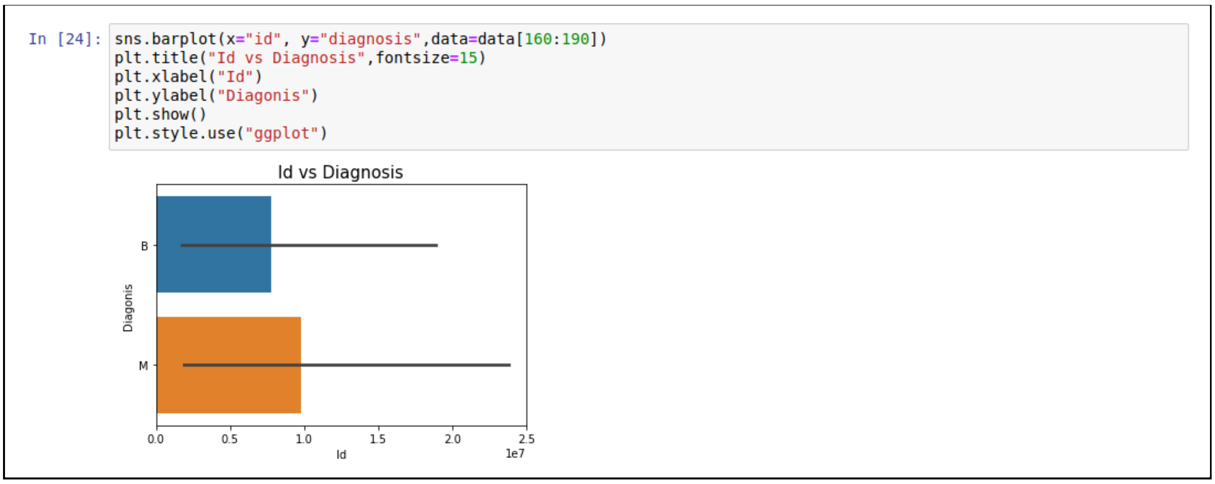
```
area_worst                 float64
smoothness_worst           float64
compactness_worst          float64
concavity_worst            float64
concave points_worst       float64
symmetry_worst             float64
fractal_dimension_worst    float64
Unnamed: 32                float64
dtype: object
```



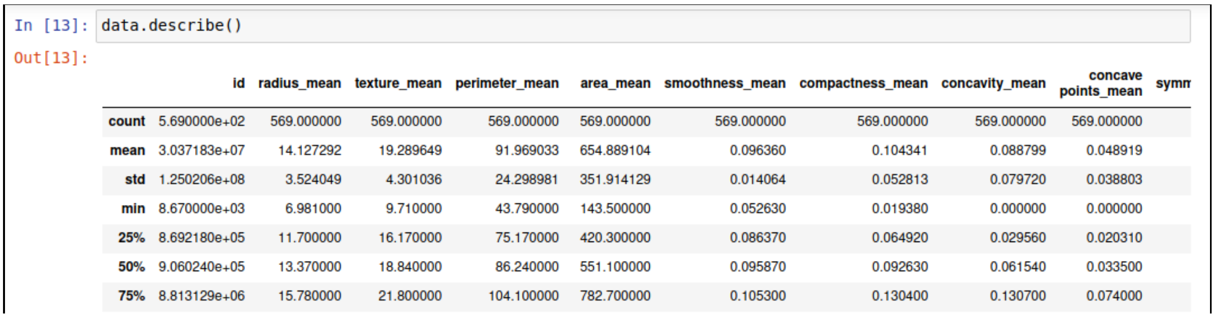
```
In [19]: data.isnull().sum()
Out[19]: id                         0
         diagnosis                  0
         radius_mean                0
         texture_mean               0
         perimeter_mean             0
         area_mean                  0
         smoothness_mean            0
         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
```



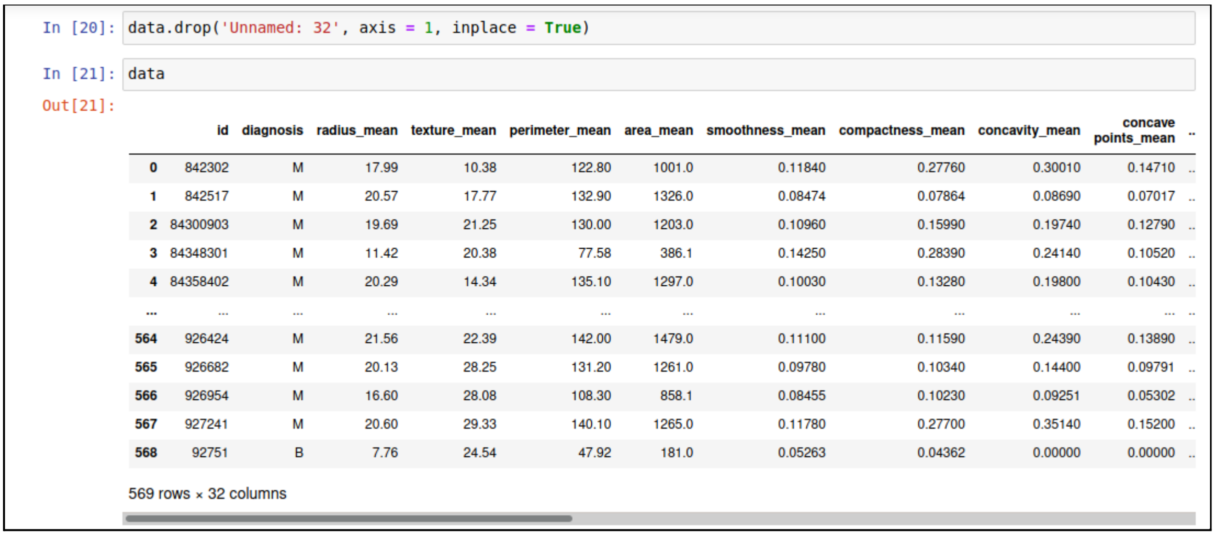
In [24]:
```
sns.barplot(x="id", y="diagnosis",data=data[160:190])
plt.title("Id vs Diagnosis",fontsize=15)
plt.xlabel("Id")
plt.ylabel("Diagonis")
plt.show()
plt.style.use("ggplot")
```



In [13]:
```
data.describe()
```

Out[13]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symm
count	5.690000e+02	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	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	



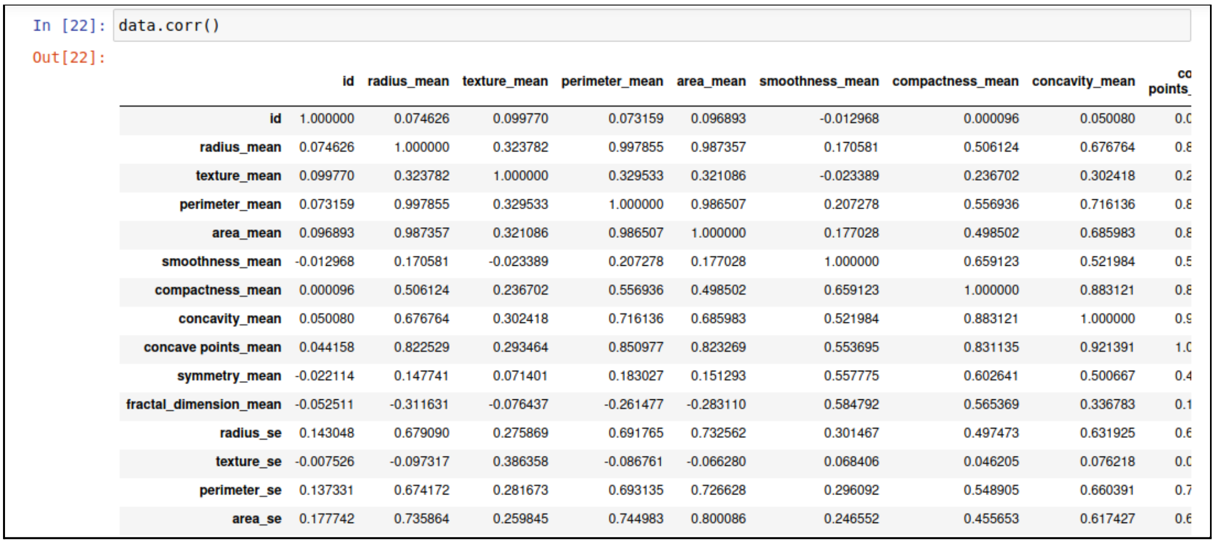
In [20]:
```
data.drop('Unnamed: 32', axis = 1, inplace = True)
```

In [21]:
```
data
```

Out[21]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	..
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	..
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	..
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	..
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	..
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	..
...	...	...	...	...	...	...	...	...	...	...	..
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	..
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	..
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	..
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	..
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	..

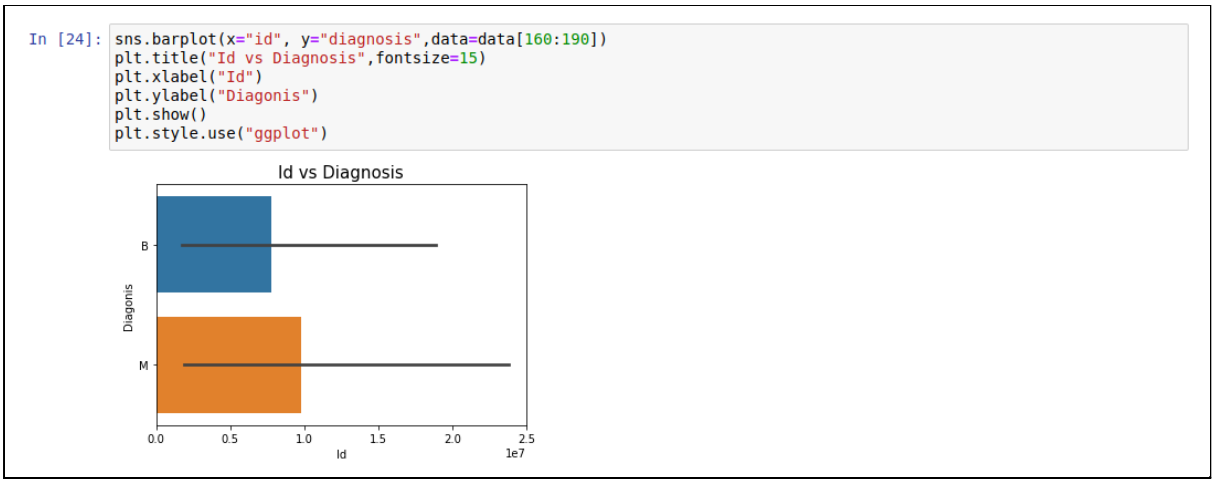
569 rows × 32 columns



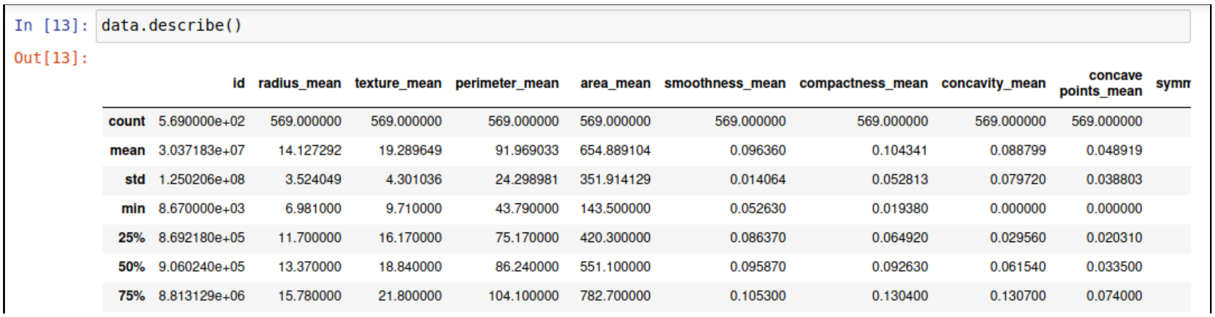
In [22]:
```
data.corr()
```

Out[22]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	co points_
id	1.000000	0.074626	0.099770	0.073159	0.096893	-0.012968	0.000096	0.050080	0.0
radius_mean	0.074626	1.000000	0.323782	0.997855	0.987357	0.170581	0.506124	0.676764	0.8
texture_mean	0.099770	0.323782	1.000000	0.329533	0.321086	-0.023389	0.236702	0.302418	0.2
perimeter_mean	0.073159	0.997855	0.329533	1.000000	0.986507	0.207278	0.556936	0.716136	0.8
area_mean	0.096893	0.987357	0.321086	0.986507	1.000000	0.177028	0.498502	0.685983	0.8
smoothness_mean	-0.012968	0.170581	-0.023389	0.207278	0.177028	1.000000	0.659123	0.521984	0.5
compactness_mean	0.000096	0.506124	0.236702	0.556936	0.498502	0.659123	1.000000	0.883121	0.8
concavity_mean	0.050080	0.676764	0.302418	0.716136	0.685983	0.521984	0.883121	1.000000	0.9
concave points_mean	0.044158	0.822529	0.293464	0.850977	0.823269	0.553695	0.831135	0.921391	1.0
symmetry_mean	-0.022114	0.147741	0.071401	0.183027	0.151293	0.557775	0.602641	0.500667	0.4
fractal_dimension_mean	-0.052511	-0.311631	-0.076437	-0.261477	-0.283110	0.584792	0.565369	0.336783	0.1
radius_se	0.143048	0.679090	0.275869	0.691765	0.732562	0.301467	0.497473	0.631925	0.6
texture_se	-0.007526	-0.097317	0.386358	-0.086761	-0.066280	0.068406	0.046205	0.076218	0.0
perimeter_se	0.137331	0.674172	0.281673	0.693135	0.726628	0.296092	0.548905	0.660391	0.7
area_se	0.177742	0.735864	0.259845	0.744983	0.800086	0.246552	0.455653	0.617427	0.6



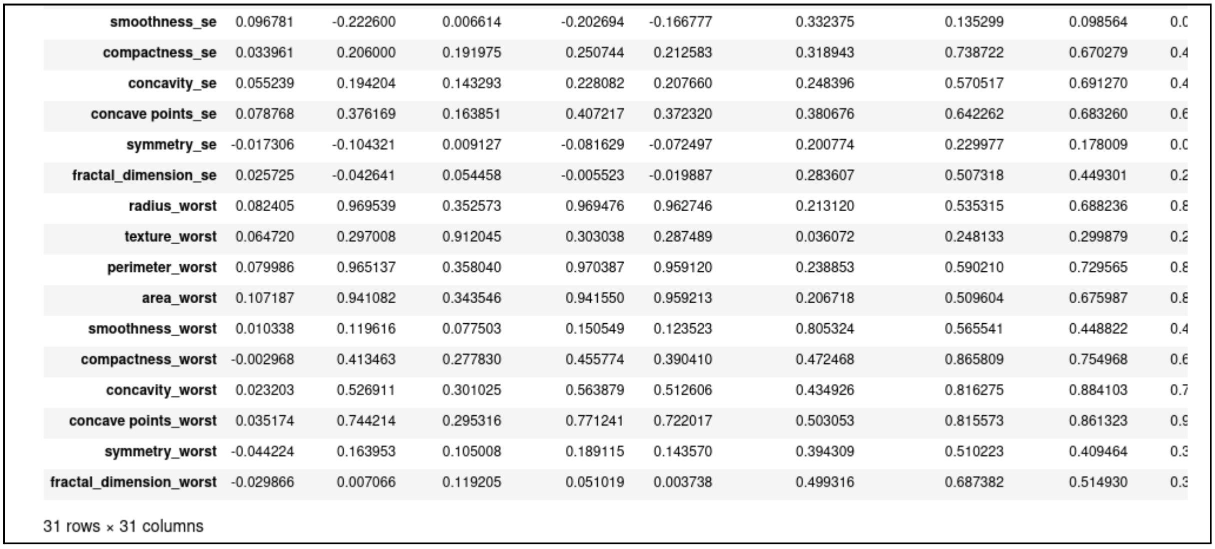
```
In [24]: sns.barplot(x="id", y="diagnosis",data=data[160:190])
         plt.title("Id vs Diagnosis",fontsize=15)
         plt.xlabel("Id")
         plt.ylabel("Diagonis")
         plt.show()
         plt.style.use("ggplot")
```



```
In [13]: data.describe()
```

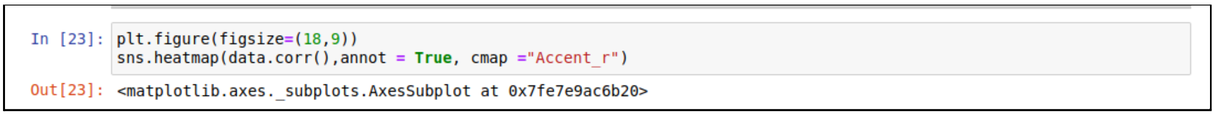
Out[13]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symm
count	5.690000e+02	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	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	

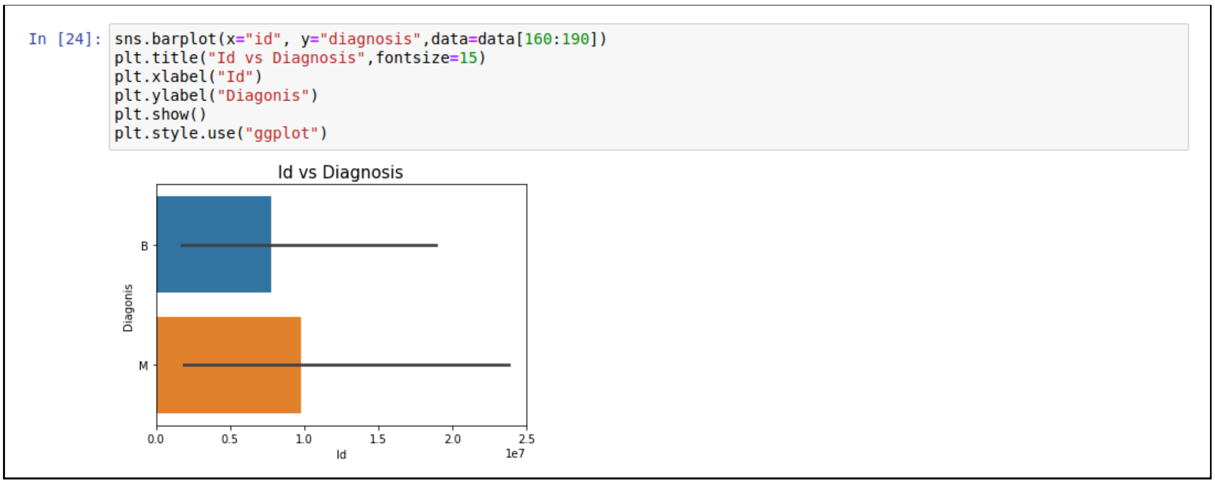


smoothness_se	0.096781	-0.222600	0.006614	-0.202694	-0.166777	0.332375	0.135299	0.098564	0.0
compactness_se	0.033961	0.206000	0.191975	0.250744	0.212583	0.318943	0.738722	0.670279	0.4
concavity_se	0.055239	0.194204	0.143293	0.228082	0.207660	0.248396	0.570517	0.691270	0.4
concave points_se	0.078768	0.376169	0.163851	0.407217	0.372320	0.380676	0.642262	0.683260	0.6
symmetry_se	-0.017306	-0.104321	0.009127	-0.081629	-0.072497	0.200774	0.229977	0.178009	0.0
fractal_dimension_se	0.025725	-0.042641	0.054458	-0.005523	-0.019887	0.283607	0.507318	0.449301	0.2
radius_worst	0.082405	0.969539	0.352573	0.969476	0.962746	0.213120	0.535315	0.688236	0.8
texture_worst	0.064720	0.297008	0.912045	0.303038	0.287489	0.036072	0.248133	0.299879	0.2
perimeter_worst	0.079986	0.965137	0.358040	0.970387	0.959120	0.238853	0.590210	0.729565	0.8
area_worst	0.107187	0.941082	0.343546	0.941550	0.959213	0.206718	0.509604	0.675987	0.8
smoothness_worst	0.010338	0.119616	0.077503	0.150549	0.123523	0.805324	0.565541	0.448822	0.4
compactness_worst	-0.002968	0.413463	0.277830	0.455774	0.390410	0.472468	0.865809	0.754968	0.6
concavity_worst	0.023203	0.526911	0.301025	0.563879	0.512606	0.434926	0.816275	0.884103	0.7
concave points_worst	0.035174	0.744214	0.295316	0.771241	0.722017	0.503053	0.815573	0.861323	0.9
symmetry_worst	-0.044224	0.163953	0.105008	0.189115	0.143570	0.394309	0.510223	0.409464	0.3
fractal_dimension_worst	-0.029866	0.007066	0.119205	0.051019	0.003738	0.499316	0.687382	0.514930	0.3

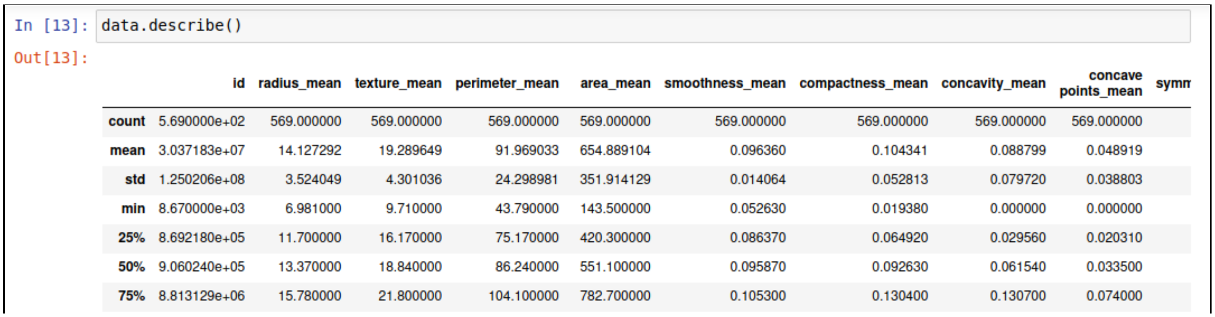
31 rows × 31 columns



```
In [23]: plt.figure(figsize=(18,9))
         sns.heatmap(data.corr(),annot = True, cmap ="Accent_r")

Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe7e9ac6b20>
```



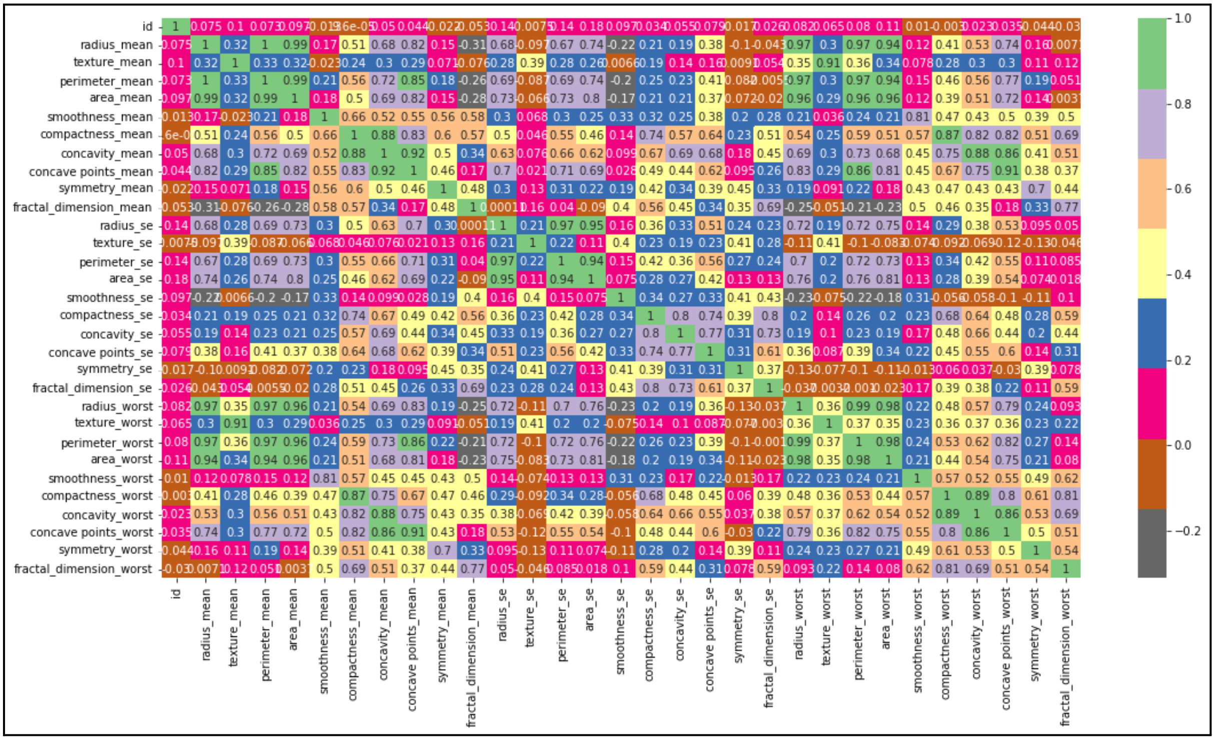
In [24]:
```
sns.barplot(x="id", y="diagnosis",data=data[160:190])
plt.title("Id vs Diagnosis",fontsize=15)
plt.xlabel("Id")
plt.ylabel("Diagonis")
plt.show()
plt.style.use("ggplot")
```

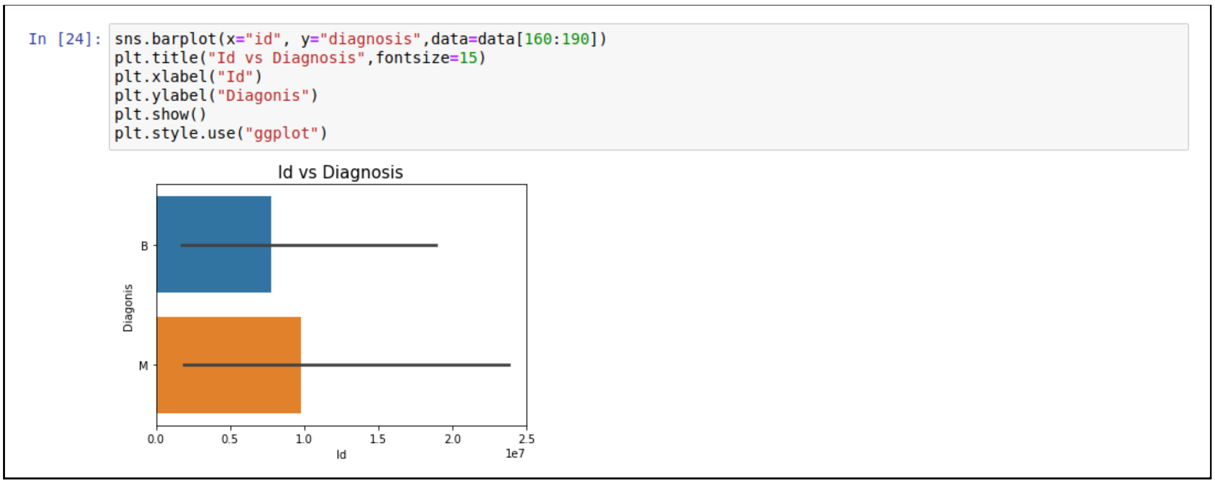


In [13]:
```
data.describe()
```

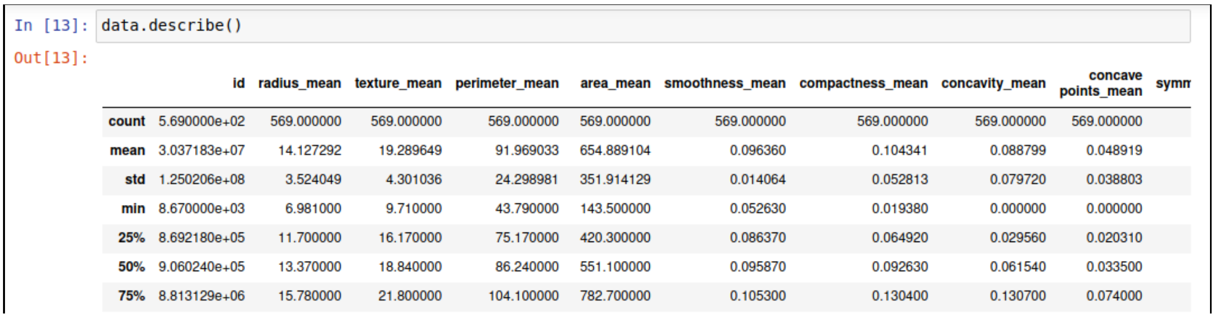
Out[13]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symm
count	5.690000e+02	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	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	



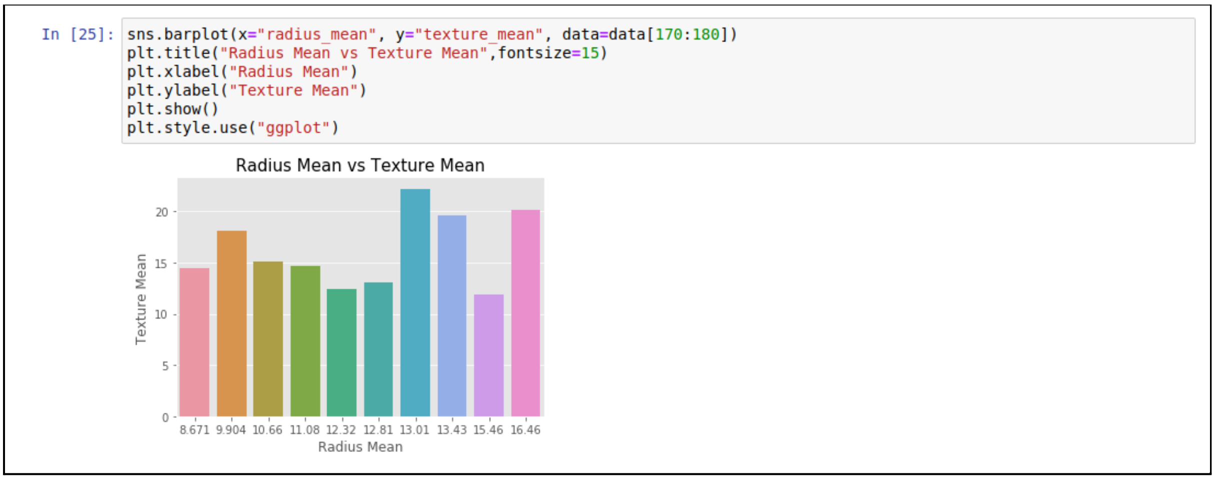


```
In [24]: sns.barplot(x="id", y="diagnosis",data=data[160:190])
         plt.title("Id vs Diagnosis",fontsize=15)
         plt.xlabel("Id")
         plt.ylabel("Diagonis")
         plt.show()
         plt.style.use("ggplot")
```

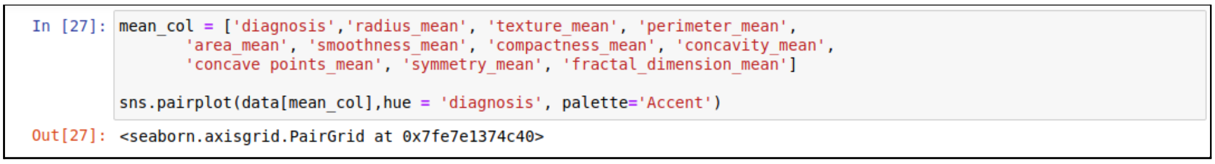


```
In [13]: data.describe()

Out[13]:
```

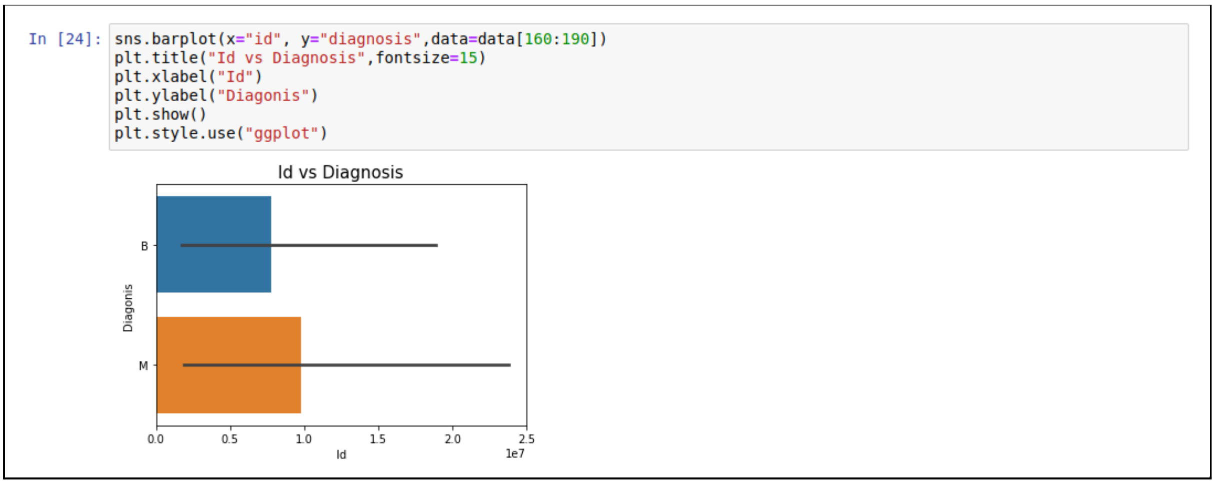
	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symm
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	



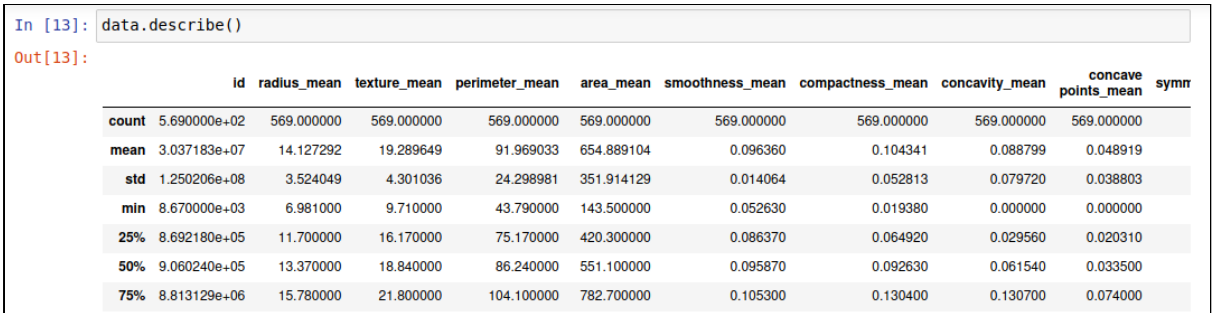
```
In [25]: sns.barplot(x="radius_mean", y="texture_mean", data=data[170:180])
         plt.title("Radius Mean vs Texture Mean",fontsize=15)
         plt.xlabel("Radius Mean")
         plt.ylabel("Texture Mean")
         plt.show()
         plt.style.use("ggplot")
```



```
In [27]: mean_col = ['diagnosis','radius_mean', 'texture_mean', 'perimeter_mean',
                 'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
                 'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean']

         sns.pairplot(data[mean_col],hue = 'diagnosis', palette='Accent')

Out[27]: <seaborn.axisgrid.PairGrid at 0x7fe7e1374c40>
```



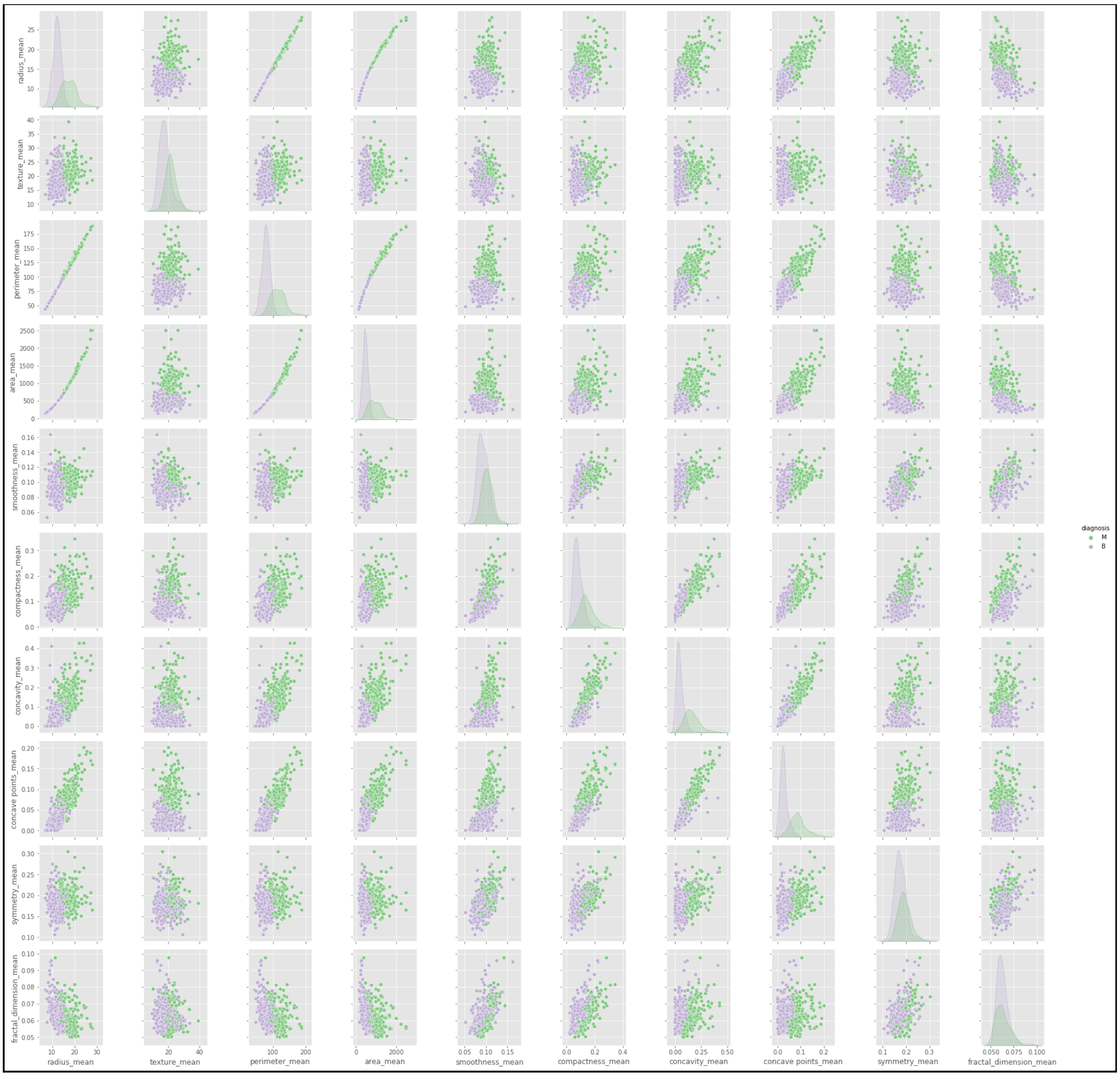
```
In [24]: sns.barplot(x="id", y="diagnosis",data=data[160:190])
         plt.title("Id vs Diagnosis",fontsize=15)
         plt.xlabel("Id")
         plt.ylabel("Diagonis")
         plt.show()
         plt.style.use("ggplot")
```

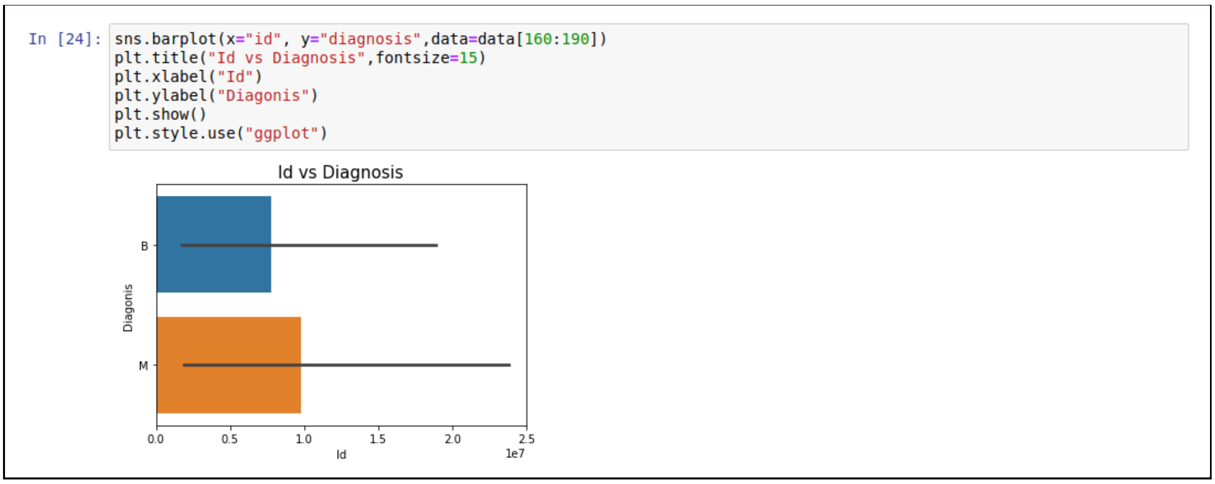


```
In [13]: data.describe()
```

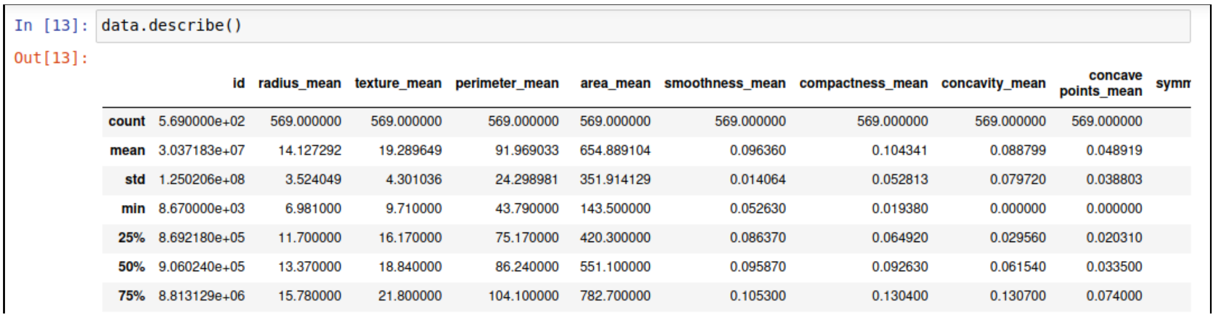
Out[13]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symm
count	5.690000e+02	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	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	





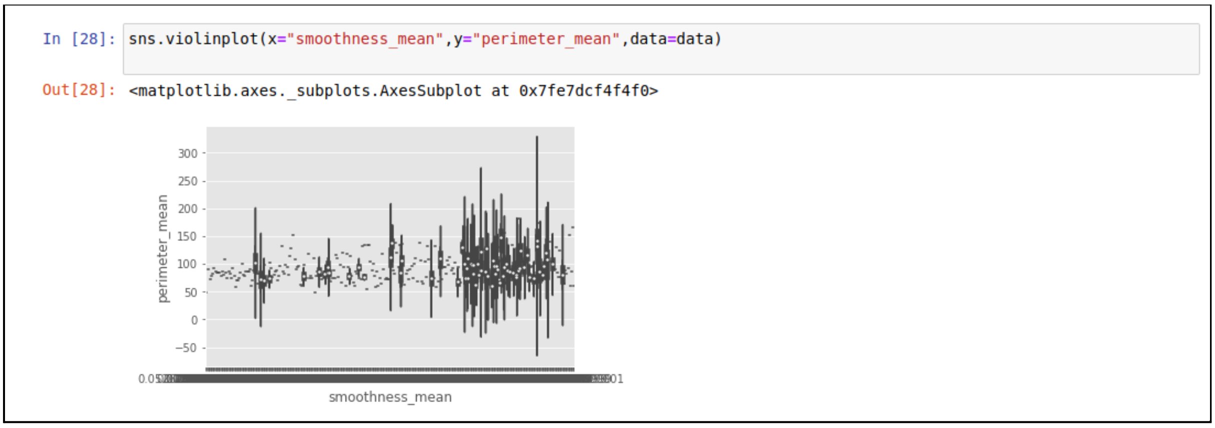
```
In [24]: sns.barplot(x="id", y="diagnosis",data=data[160:190])
         plt.title("Id vs Diagnosis",fontsize=15)
         plt.xlabel("Id")
         plt.ylabel("Diagonis")
         plt.show()
         plt.style.use("ggplot")
```



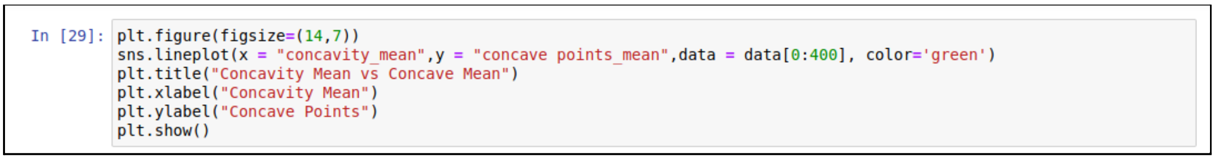
```
In [13]: data.describe()
```

Out[13]:

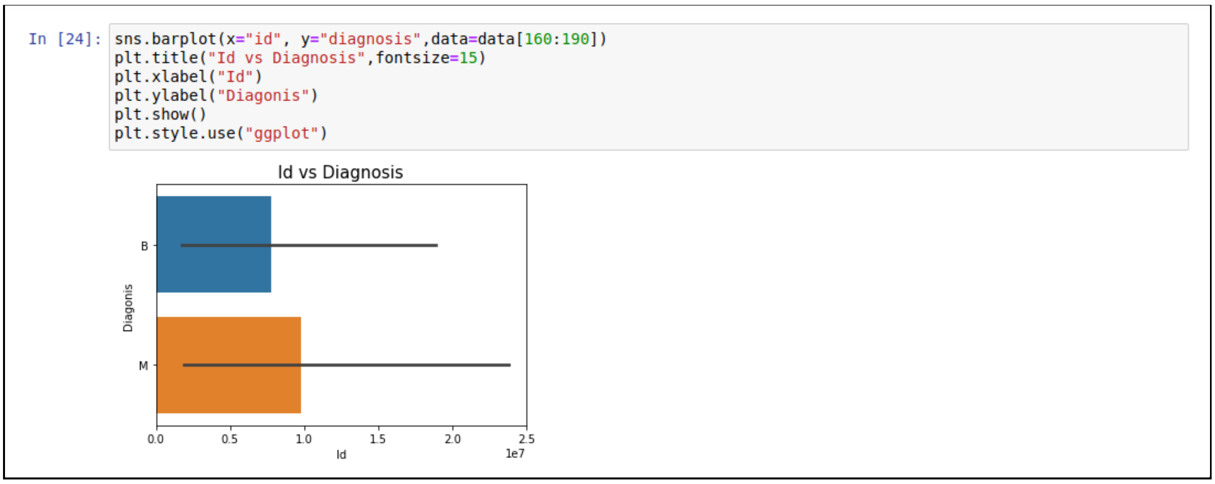
	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symm
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	



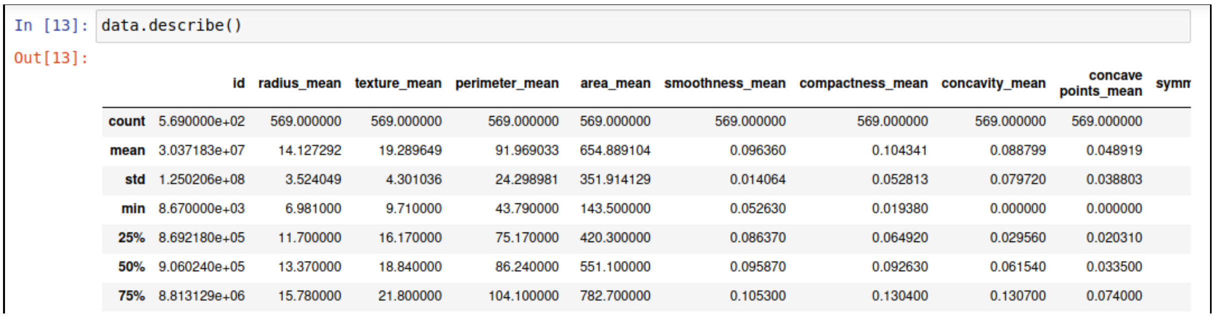
```
In [28]: sns.violinplot(x="smoothness_mean",y="perimeter_mean",data=data)

Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe7dcf4f4f0>
```



```
In [29]: plt.figure(figsize=(14,7))
         sns.lineplot(x = "concavity_mean",y = "concave points_mean",data = data[0:400], color='green')
         plt.title("Concavity Mean vs Concave Mean")
         plt.xlabel("Concavity Mean")
         plt.ylabel("Concave Points")
         plt.show()
```

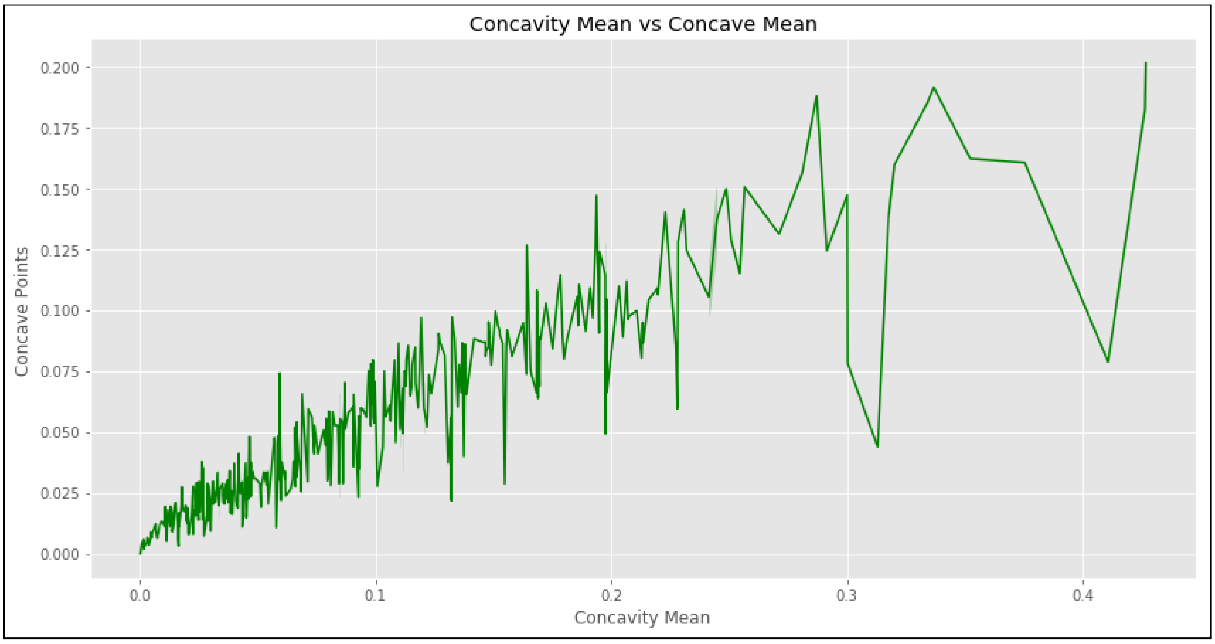


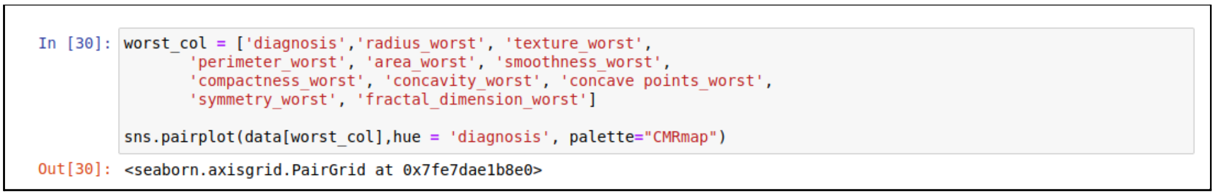
```
In [24]: sns.barplot(x="id", y="diagnosis",data=data[160:190])
         plt.title("Id vs Diagnosis",fontsize=15)
         plt.xlabel("Id")
         plt.ylabel("Diagonis")
         plt.show()
         plt.style.use("ggplot")
```



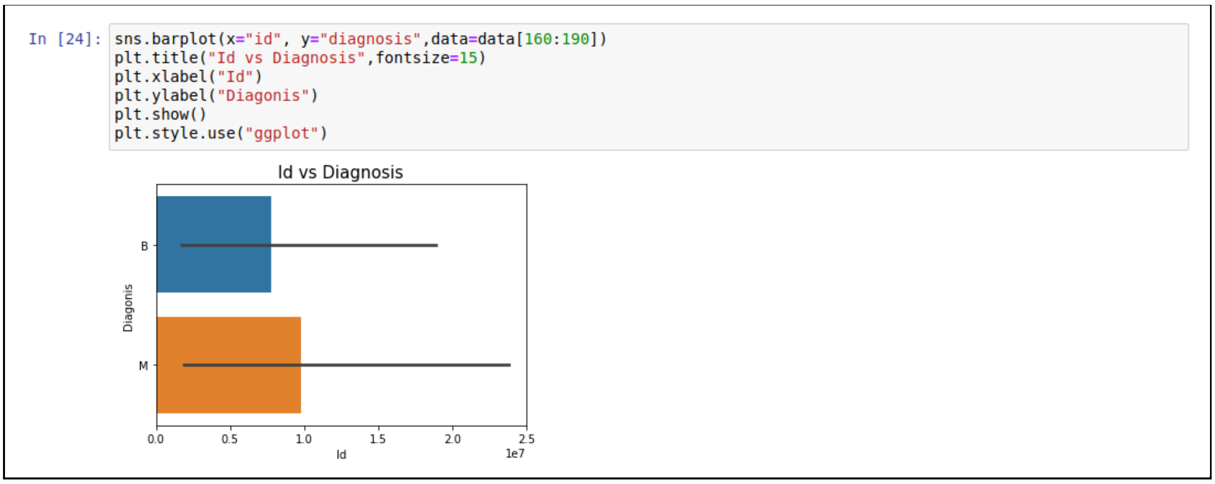
```
In [13]: data.describe()
Out[13]:
```

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symm
count	5.690000e+02	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	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	

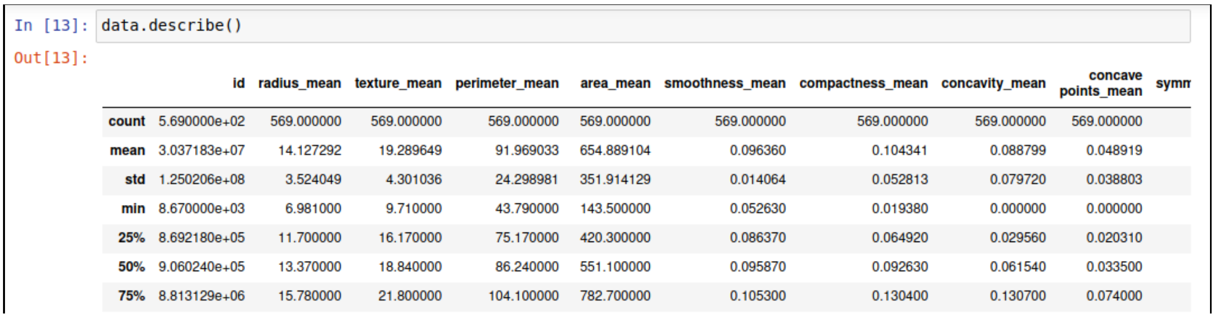




```
In [30]: worst_col = ['diagnosis','radius_worst', 'texture_worst',
                'perimeter_worst', 'area_worst', 'smoothness_worst',
                'compactness_worst', 'concavity_worst', 'concave points_worst',
                'symmetry_worst', 'fractal_dimension_worst']

         sns.pairplot(data[worst_col],hue = 'diagnosis', palette="CMRmap")
Out[30]: <seaborn.axisgrid.PairGrid at 0x7fe7dae1b8e0>
```



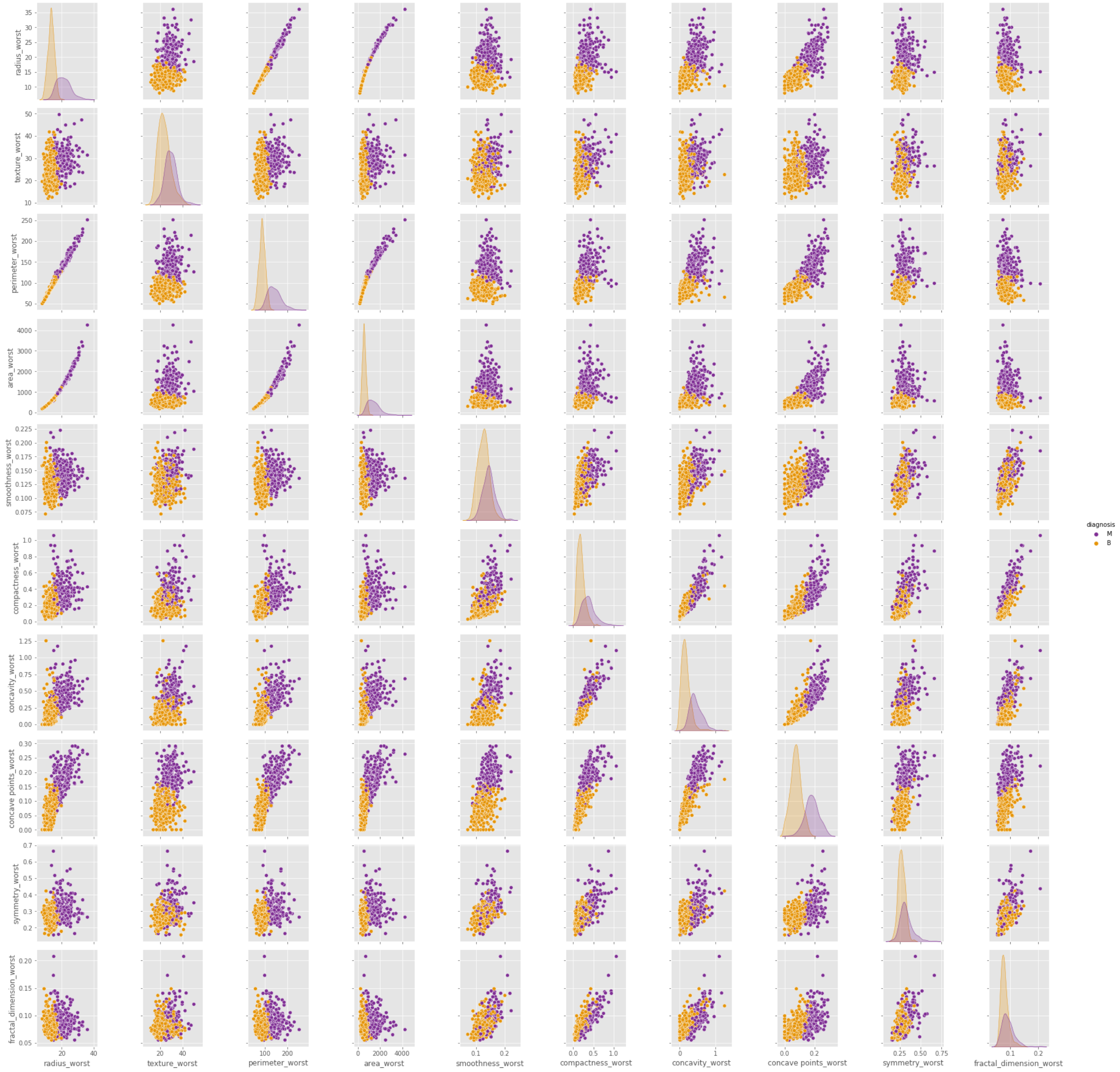
```
In [24]: sns.barplot(x="id", y="diagnosis",data=data[160:190])
         plt.title("Id vs Diagnosis",fontsize=15)
         plt.xlabel("Id")
         plt.ylabel("Diagonis")
         plt.show()
         plt.style.use("ggplot")
```

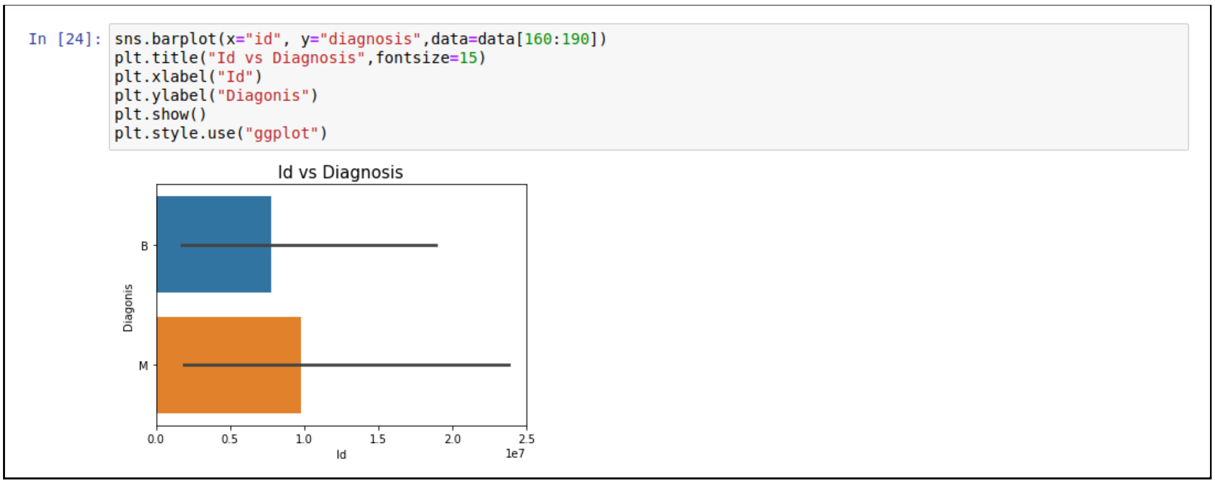


```
In [13]: data.describe()
```

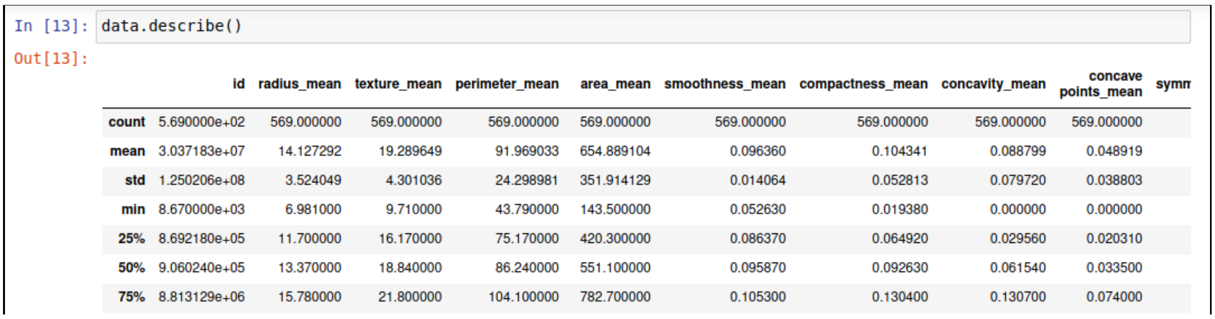
Out[13]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symm
count	5.690000e+02	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	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	





```
In [24]: sns.barplot(x="id", y="diagnosis",data=data[160:190])
         plt.title("Id vs Diagnosis",fontsize=15)
         plt.xlabel("Id")
         plt.ylabel("Diagonis")
         plt.show()
         plt.style.use("ggplot")
```



```
In [13]: data.describe()
Out[13]:
```

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symm
count	5.690000e+02	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	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	

4. Link for google sheet: ⊞ Data Analytics - Literature survey

5. Literature Survey (Summarize): ☰ Patterns n Parameters_ literature survey

6. Your Plan:We'll implement various algorithms such as Support Vector Machine, Logistic Regression, K-Nearest Neighbor, Decision Tree algorithms and compare the results.We will conclude with whichever individual/ensemble model gives the highest accuracy

7. References: https://ieeexplore.ieee.org/document/9445847
https://ieeexplore.ieee.org/document/9421338
https://ieeexplore.ieee.org/document/6016771
https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0226765
https://ieeexplore.ieee.org/document/8965528
https://ieeexplore.ieee.org/abstract/document/8820378
https://ieeexplore.ieee.org/document/5703994
http://ijiepr.iust.ac.ir/article-1-1069-fa.pdf
https://ieeexplore.ieee.org/document/8605180
https://pubs.rsna.org/doi/full/10.1148/radiol.2019182716
https://academic.oup.com/jnci/article/98/17/1204/2521747?login=true
https://core.ac.uk/download/pdf/295538238.pdf