```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
```

```
import warnings
warnings.filterwarnings('ignore')
```

```
# from google.colab import files
# uploaded = files.upload()
# import io
dataset = pd.read_csv('/data.csv')
```

```
dataset.head()
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactn
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	

5 rows × 33 columns

```
dataset.shape
```

```
(569, 33)
```

```
dataset.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
 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
```

```
dataset.describe()
```

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactnes
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.096360	0
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0
max	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	0.163400	0

8 rows × 32 columns

```
dataset.select_dtypes(include = 'object').columns
```

```
Index(['diagnosis'], dtype='object')
```

```
dataset.select_dtypes(include = ['int64','float64']).columns
```

```
Index(['id', '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')
```

```
dataset.columns
```

```
Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst', 'Unnamed: 32'],
      dtype='object')
```

```
dataset.isnull().values.any()
```

```
True
```

```
dataset.isnull().values.sum()
```

```
569
```

```
dataset.columns[dataset.isnull().any()]
```

```
Index(['Unnamed: 32'], dtype='object')
```

```
dataset = dataset.drop(columns = 'Unnamed: 32')
```

```
dataset.shape
```

```
(569, 32)
```

```
dataset.isnull().values.any()
```

```
False
```

```
dataset.select_dtypes(include = 'object')
```

	diagnosis
0	M
1	M
2	M
3	M
4	M
...	...
564	M
565	M
566	M

```
dataset.select_dtypes(include = 'object')
```

	diagnosis
0	M
1	M
2	M
3	M
4	M
...	...
564	M
565	M
566	M
567	M
568	B

569 rows × 1 columns

```
dataset.select_dtypes(include = 'object').columns
```

```
Index(['diagnosis'], dtype='object')
```

```
dataset = pd.get_dummies(data = dataset, drop_first =True)
```

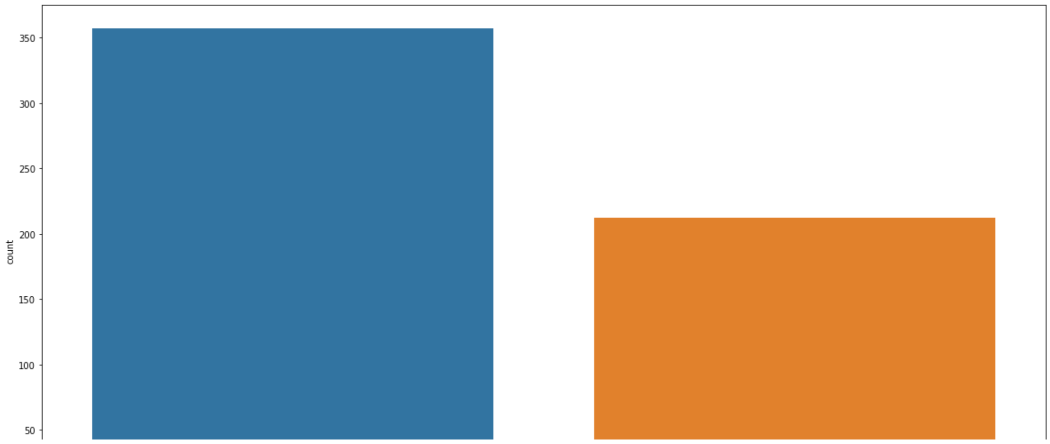
```
dataset.head()
```

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	c
0	842302	17.99	10.38	122.80	1001.0	0.11840	0.27760	
1	842517	20.57	17.77	132.90	1326.0	0.08474	0.07864	
2	84300903	19.69	21.25	130.00	1203.0	0.10960	0.15990	
3	84348301	11.42	20.38	77.58	386.1	0.14250	0.28390	
4	84358402	20.29	14.34	135.10	1297.0	0.10030	0.13280	

5 rows × 32 columns

```
sns.countplot(dataset['diagnosis_M'], label = 'count')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f76c9f5bb20>
```



```
(dataset.diagnosis_M == 0).sum()
```

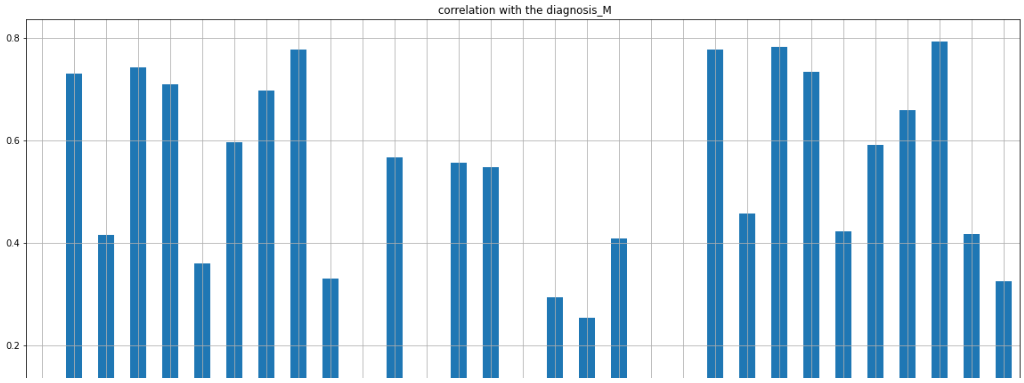
```
357
```

```
dataset_2 = dataset.drop(columns = 'diagnosis_M')
```

```
dataset_2.corrwith(dataset['diagnosis_M'])
```

```
id                         0.039769
radius_mean                0.730029
texture_mean               0.415185
perimeter_mean             0.742636
area_mean                  0.708984
smoothness_mean            0.358560
compactness_mean           0.596534
concavity_mean             0.696360
concave points_mean        0.776614
symmetry_mean              0.330499
fractal_dimension_mean    -0.012838
radius_se                  0.567134
texture_se                -0.008303
perimeter_se               0.556141
area_se                    0.548236
smoothness_se             -0.067016
compactness_se             0.292999
concavity_se               0.253730
concave points_se          0.408042
symmetry_se               -0.006522
fractal_dimension_se       0.077972
radius_worst               0.776454
texture_worst              0.456903
perimeter_worst            0.782914
area_worst                 0.733825
smoothness_worst           0.421465
compactness_worst          0.590998
concavity_worst            0.659610
concave points_worst       0.793566
symmetry_worst             0.416294
fractal_dimension_worst    0.323872
dtype: float64
```

```
dataset_2.corrwith(dataset['diagnosis_M']).plot.bar(figsize = (20,10), title = ' correlation with the diagnosis_M', rot = 45, grid = True
<matplotlib.axes._subplots.AxesSubplot at 0x7f76c98fe280>
```



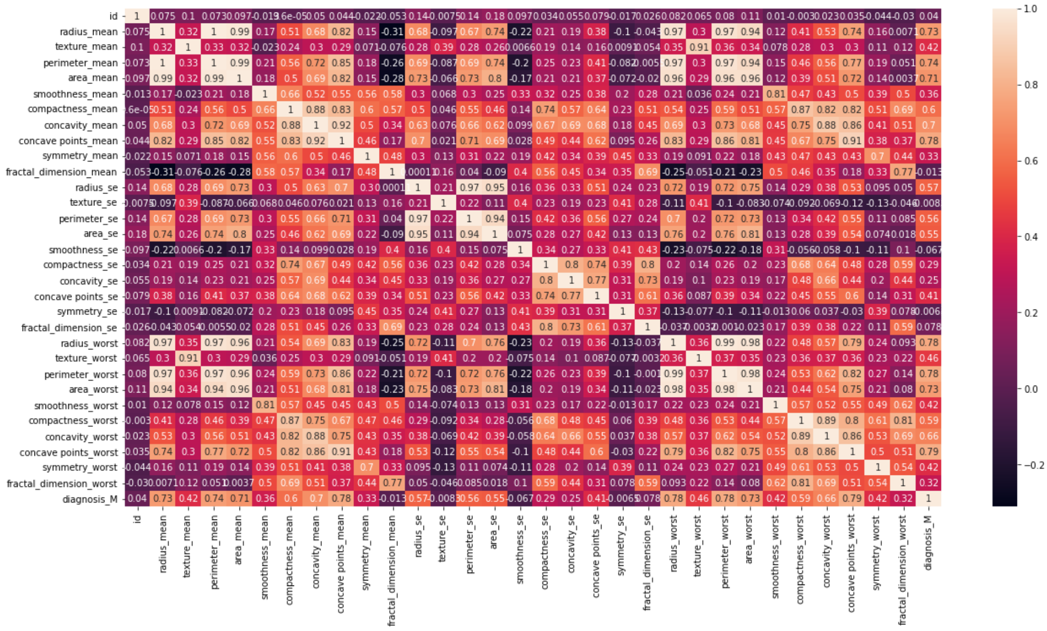
```
corr = dataset.corr()
corr
```

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean
id	1.000000	0.074626	0.099770	0.073159	0.096893	-0.012968
radius_mean	0.074626	1.000000	0.323782	0.997855	0.987357	0.170581
texture_mean	0.099770	0.323782	1.000000	0.329533	0.321086	-0.023389
perimeter_mean	0.073159	0.997855	0.329533	1.000000	0.986507	0.207278
area_mean	0.096893	0.987357	0.321086	0.986507	1.000000	0.177028
smoothness_mean	-0.012968	0.170581	-0.023389	0.207278	0.177028	1.000000
compactness_mean	0.000096	0.506124	0.236702	0.556936	0.498502	0.659123
concavity_mean	0.050080	0.676764	0.302418	0.716136	0.685983	0.521984
concave points_mean	0.044158	0.822529	0.293464	0.850977	0.823269	0.553695
symmetry_mean	-0.022114	0.147741	0.071401	0.183027	0.151293	0.557775
fractal_dimension_mean	-0.052511	-0.311631	-0.076437	-0.261477	-0.283110	0.584792
radius_se	0.143048	0.679090	0.275869	0.691765	0.732562	0.301467
texture_se	-0.007526	-0.097317	0.386358	-0.086761	-0.066280	0.068406
perimeter_se	0.137331	0.674172	0.281673	0.693135	0.726628	0.296092
area_se	0.177742	0.735864	0.259845	0.744983	0.800086	0.246552
smoothness_se	0.096781	-0.222600	0.006614	-0.202694	-0.166777	0.332375
compactness_se	0.033961	0.206000	0.191975	0.250744	0.212583	0.318943
concavity_se	0.055239	0.194204	0.143293	0.228082	0.207660	0.248396
concave points_se	0.078768	0.376169	0.163851	0.407217	0.372320	0.380676
symmetry_se	-0.017306	-0.104321	0.009127	-0.081629	-0.072497	0.200774
fractal_dimension_se	0.025725	-0.042641	0.054458	-0.005523	-0.019887	0.283607
radius_worst	0.082405	0.969539	0.352573	0.969476	0.962746	0.213120
texture_worst	0.064720	0.297008	0.912045	0.303038	0.287489	0.036072
perimeter_worst	0.079986	0.965137	0.358040	0.970387	0.959120	0.238853
area_worst	0.107187	0.941082	0.343546	0.941550	0.959213	0.206718
smoothness_worst	0.010338	0.119616	0.077503	0.150549	0.123523	0.805324
compactness_worst	-0.002968	0.413463	0.277830	0.455774	0.390410	0.472468
concavity_worst	0.023203	0.526911	0.301025	0.563879	0.512606	0.434926
concave points_worst	0.035174	0.744214	0.295316	0.771241	0.722017	0.503053
symmetry_worst	-0.044224	0.163953	0.105008	0.189115	0.143570	0.394309
fractal_dimension_worst	-0.029866	0.007066	0.119205	0.051019	0.003738	0.499316
diagnosis_M	0.039769	0.730029	0.415185	0.742636	0.708984	0.358560

32 rows × 32 columns

```
plt.figure(figsize = (20,10))
sns.heatmap(corr, annot = True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f76c9ee3940>
```



```
x =  dataset.iloc[:,1:-1].values
```

```
y = dataset.iloc[:,-1].values
```

```
x.shape
```

```
(569, 30)
```

```
y.shape
```

```
(569,)
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y, test_size = 0.33, random_state = 10)
```

```
x_test.shape
```

```
(188, 30)
```

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

```
x_test
```

```
array([[ 0.32939352, -1.70277545,  0.38002753, ...,  1.01205104,
         0.56500364,  0.94395709],
       [-1.37094822,  0.62832296, -1.3674522 , ..., -1.41478814,
        -0.80492439, -0.56996616],
       [-0.16181015, -0.73322407, -0.20902864, ..., -0.7313608 ,
         0.75268904, -0.39771335],
       ...,
       [ 0.04077667,  0.12499127,  0.05777443, ...,  0.60721731,
        -0.30852281,  0.5261881 ],
       [ 0.14900799, -1.05099569,  0.06500704, ..., -0.58178864,
        -0.98208538, -1.26960339],
       [-0.39492376,  0.49611141, -0.41837261, ..., -0.94552094,
        -0.71195873, -0.14604528]])
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import f1_score, accuracy_score, confusion_matrix, precision_score, recall_score
logreg = LogisticRegression()
logreg.fit(x_train,y_train,)
```

```
     LogisticRegression()
```

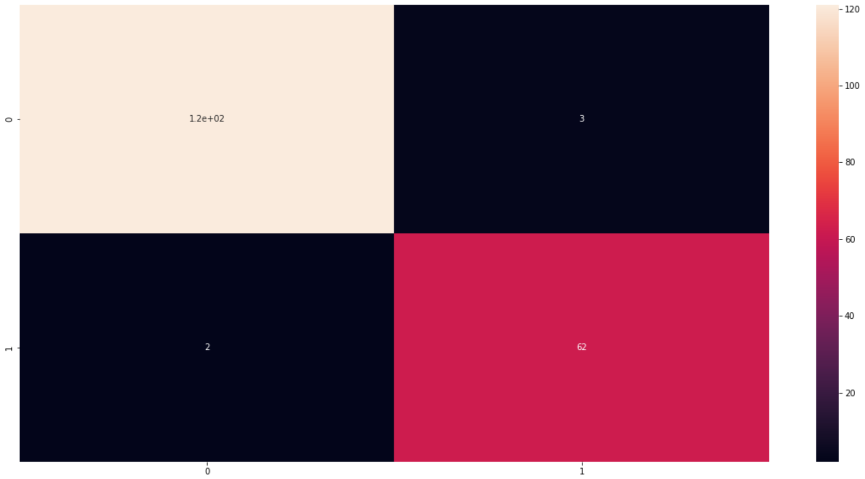
```
y_pred = logreg.predict(x_test)
acc = accuracy_score(y_test,y_pred)
f1 = f1_score(y_test,y_pred)
precision = precision_score(y_test,y_pred)
recall = recall_score(y_test,y_pred)
results = pd.DataFrame([['Logistic Regression', acc, f1, precision, recall]],
                       columns = ['Model','Accuracy', 'F1', ' Precision', 'Recall'])
```

```
results
```

	Model	Accuracy	F1	Precision	Recall
0	Logistic Regression	0.973404	0.96124	0.953846	0.96875

```
cm = confusion_matrix(y_test,y_pred)
print(cm)
sns.heatmap(cm,annot = True)
```

```
     [[121   3]
      [  2  62]]
     <matplotlib.axes._subplots.AxesSubplot at 0x7f76c344bb80>
```


1.2e+02
3
2
62
0
1
0
1
120
100
80
60
40
20