```
# Analyzing the data frame
df.head(10)
```

	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	843786	M	12.45	15.70	82.57	477.1	0.12780	0.17000	0.1578	0.08089

5 rows × 33 columns

```
# Analyzing the data using the .info() method
df.info()
```

```
<class'> RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 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
 Unnamed: 32                float64
dtypes: float64(31), int64(1), object(1)
memory usage: 147.3+ KB
```

```
# Analyzing the data using the .describe() method
df.describe()
```

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean
count	5.69e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000
mean	3.037183e+07	14.127291	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	0.181162
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	0.027414
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	0.106000
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	0.161900
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	0.179200
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	0.195700
max	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	0.304000

8 rows × 32 columns

```
# Analyzing the data using the .shape method
df.shape
```

```
(569,33)
```

```
# Analyzing the data using the .tail() method
df.tail(5)
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
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	0.10223	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

5rows × 33 columns

```
# Impact of the dependent variable on the numerical features of the dataset
from sklearn.preprocessing import PandasFeatureCorrelation
```

```
# Transformation of the dependent variable into numerical and binarizing it
model = PandasFeatureCorrelation(n_jobs=0)
df_numerical = model.fit_transform(df)
df_numerical.head(5)
# Displaying the correlation with the dependent variable
sorted_features = df_numerical['Malignant'].sort_values(ascending=False)
sorted_features.to_frame()
# Displaying the correlation with the dependent variable in a dataframe format
sorted_features = df_numerical['Malignant'].sort_values(ascending=False).to_frame
# Displaying the top 10 most correlated features with the dependent variable
top_features = sorted_features[sorted_features.abs().sort_values(ascending=False).head(10)]
top_features.T
```

	Percentage Impurities	Cumulative percentage Impurities
perimeter_worst	14.09	14.09
area_worst	10.970	25.059
radius_worst	10.969	36.028
concave points_worst	9.136	45.164
radius_mean	7.41	52.574
area_se	6.016	58.590
concavity_mean	5.177	63.767
concavity_worst	3.883	67.650
compactness_worst	3.854	71.504
perimeter_mean	3.878	75.382
area_mean	3.673	79.055
perimeter_se	3.085	82.270
compactness_mean	1.537	83.807
texture_worst	1.502	85.309
radius_se	1.419	86.728
smoothness_worst	1.326	88.054
compactness_worst	1.189	89.243
texture_mean	1.156	90.399
symmetry_worst	1.144	91.543
concave points_se	1.004	92.547
smoothness_mean	0.731	93.277
fractal_dimension_worst	0.716	93.993
concavity_se	0.636	94.629
smoothness_se	0.615	95.244
symmetry_se	0.421	95.665
fractal_dimension_se	0.401	96.066
symmetry_mean	0.330	96.396
texture_se	0.234	96.630
fractal_dimension_mean	0.064	96.694

```
# Displaying the correlation with the dependent variable
Malignant    1.000000
Benign       0.730029
Name: Malignant, dtype: float64
```

```
# Displaying the correlation with the dependent variable in a dataframe format
```