In [3]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
```

In [4]:
```
from sklearn.datasets import load_breast_cancer
```

In [6]:
```
cancer = load_breast_cancer()
```

In [12]:
```
cancer.keys()
```

Out[12]:
```
dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filenam
e'])
```

In [13]:
```
df = pd.DataFrame(cancer['data'],columns=cancer['feature_names'])
```

In [14]:
```
df.describe()
```

Out[14]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	me conca poi
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.0000
mean	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.0489
std	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.0388
min	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.0000
25%	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.0203
50%	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.0335
75%	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.0740
max	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.2012

8 rows × 30 columns

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

Out[15]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	m fra dimen
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	m fra dimen
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.0
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.0
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.0

5 rows × 30 columns

In [18]:
```
cor=df.corr()
cor
```

Out[18]:

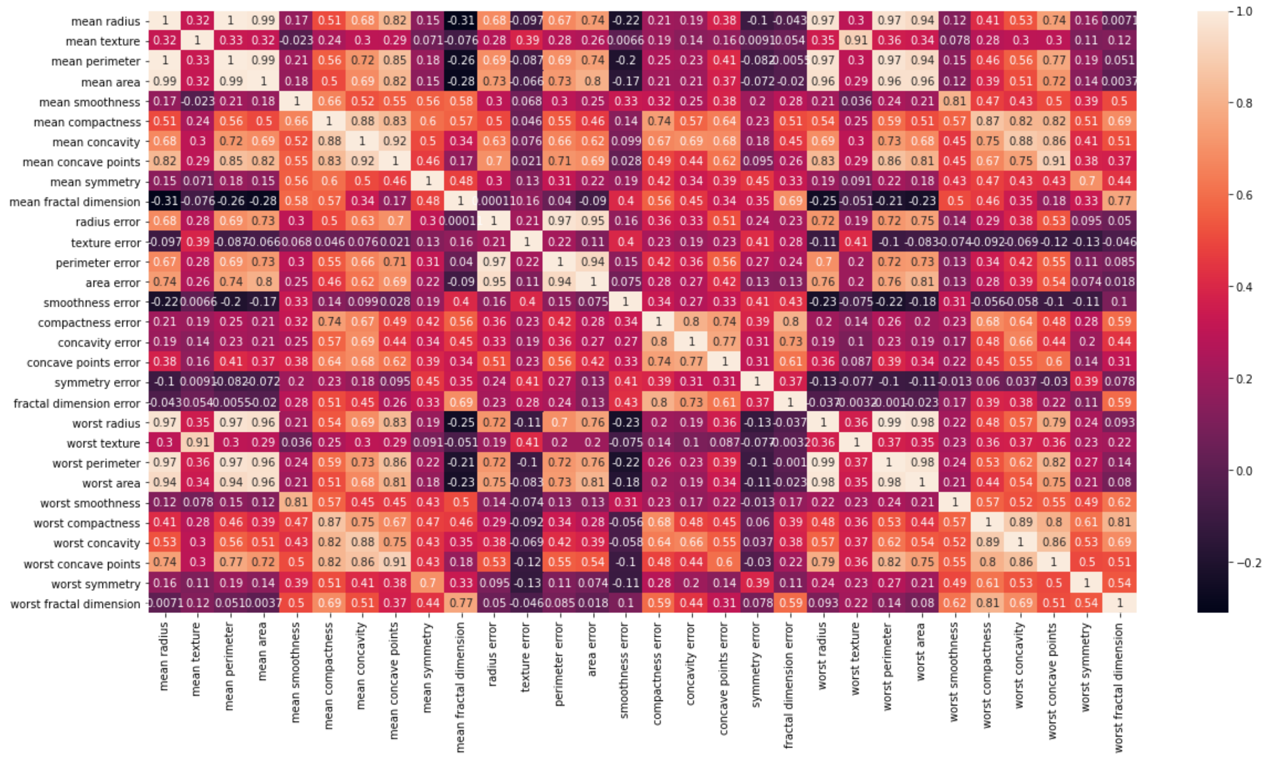
	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points
mean radius	1.000000	0.323782	0.997855	0.987357	0.170581	0.506124	0.676764	0.822529
mean texture	0.323782	1.000000	0.329533	0.321086	-0.023389	0.236702	0.302418	0.293464
mean perimeter	0.997855	0.329533	1.000000	0.986507	0.207278	0.556936	0.716136	0.850977
mean area	0.987357	0.321086	0.986507	1.000000	0.177028	0.498502	0.685983	0.823269
mean smoothness	0.170581	-0.023389	0.207278	0.177028	1.000000	0.659123	0.521984	0.553695
mean compactness	0.506124	0.236702	0.556936	0.498502	0.659123	1.000000	0.883121	0.831135
mean concavity	0.676764	0.302418	0.716136	0.685983	0.521984	0.883121	1.000000	0.921391
mean concave points	0.822529	0.293464	0.850977	0.823269	0.553695	0.831135	0.921391	1.000000
mean symmetry	0.147741	0.071401	0.183027	0.151293	0.557775	0.602641	0.500667	0.462497
mean fractal dimension	-0.311631	-0.076437	-0.261477	-0.283110	0.584792	0.565369	0.336783	0.166917
radius error	0.679090	0.275869	0.691765	0.732562	0.301467	0.497473	0.631925	0.698050
texture error	-0.097317	0.386358	-0.086761	-0.066280	0.068406	0.046205	0.076218	0.021480
perimeter error	0.674172	0.281673	0.693135	0.726628	0.296092	0.548905	0.660391	0.710650
area error	0.735864	0.259845	0.744983	0.800086	0.246552	0.455653	0.617427	0.690299
smoothness error	-0.222600	0.006614	-0.202694	-0.166777	0.332375	0.135299	0.098564	0.027653

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points
compactness error	0.206000	0.191975	0.250744	0.212583	0.318943	0.738722	0.670279	0.490424
concavity error	0.194204	0.143293	0.228082	0.207660	0.248396	0.570517	0.691270	0.439167
concave points error	0.376169	0.163851	0.407217	0.372320	0.380676	0.642262	0.683260	0.615634
symmetry error	-0.104321	0.009127	-0.081629	-0.072497	0.200774	0.229977	0.178009	0.095351
fractal dimension error	-0.042641	0.054458	-0.005523	-0.019887	0.283607	0.507318	0.449301	0.257584
worst radius	0.969539	0.352573	0.969476	0.962746	0.213120	0.535315	0.688236	0.830318
worst texture	0.297008	0.912045	0.303038	0.287489	0.036072	0.248133	0.299879	0.292752
worst perimeter	0.965137	0.358040	0.970387	0.959120	0.238853	0.590210	0.729565	0.855923
worst area	0.941082	0.343546	0.941550	0.959213	0.206718	0.509604	0.675987	0.809630
worst smoothness	0.119616	0.077503	0.150549	0.123523	0.805324	0.565541	0.448822	0.452753
worst compactness	0.413463	0.277830	0.455774	0.390410	0.472468	0.865809	0.754968	0.667454
worst concavity	0.526911	0.301025	0.563879	0.512606	0.434926	0.816275	0.884103	0.752399
worst concave points	0.744214	0.295316	0.771241	0.722017	0.503053	0.815573	0.861323	0.910155
worst symmetry	0.163953	0.105008	0.189115	0.143570	0.394309	0.510223	0.409464	0.375744
worst fractal dimension	0.007066	0.119205	0.051019	0.003738	0.499316	0.687382	0.514930	0.368661

30 rows × 30 columns

In [35]:
```
plt.figure(figsize = (20,10))
sns.heatmap(df.corr(),annot = True)
```

Out[35]:
```
<AxesSubplot:>
```



In [23]:

```
df.transpose()
```

Out[23]:

	0	1	2	3	4	5	6
mean radius	17.990000	20.570000	19.690000	11.420000	20.290000	12.450000	18.250000
mean texture	10.380000	17.770000	21.250000	20.380000	14.340000	15.700000	19.980000
mean perimeter	122.800000	132.900000	130.000000	77.580000	135.100000	82.570000	119.600000
mean area	1001.000000	1326.000000	1203.000000	386.100000	1297.000000	477.100000	1040.000000
mean smoothness	0.118400	0.084740	0.109600	0.142500	0.100300	0.127800	0.094630
mean compactness	0.277600	0.078640	0.159900	0.283900	0.132800	0.170000	0.109000
mean concavity	0.300100	0.086900	0.197400	0.241400	0.198000	0.157800	0.112700
mean concave points	0.147100	0.070170	0.127900	0.105200	0.104300	0.080890	0.074000
mean symmetry	0.241900	0.181200	0.206900	0.259700	0.180900	0.208700	0.179400
mean fractal dimension	0.078710	0.056670	0.059990	0.097440	0.058830	0.076130	0.057420
radius error	1.095000	0.543500	0.745600	0.495600	0.757200	0.334500	0.446700
texture error	0.905300	0.733900	0.786900	1.156000	0.781300	0.890200	0.773200

	0	1	2	3	4	5	6
perimeter error	8.589000	3.398000	4.585000	3.445000	5.438000	2.217000	3.180000
area error	153.400000	74.080000	94.030000	27.230000	94.440000	27.190000	53.910000
smoothness error	0.006399	0.005225	0.006150	0.009110	0.011490	0.007510	0.004314
compactness error	0.049040	0.013080	0.040060	0.074580	0.024610	0.033450	0.013820
concavity error	0.053730	0.018600	0.038320	0.056610	0.056880	0.036720	0.022540
concave points error	0.015870	0.013400	0.020580	0.018670	0.018850	0.011370	0.010390
symmetry error	0.030030	0.013890	0.022500	0.059630	0.017560	0.021650	0.013690
fractal dimension error	0.006193	0.003532	0.004571	0.009208	0.005115	0.005082	0.002179
worst radius	25.380000	24.990000	23.570000	14.910000	22.540000	15.470000	22.880000
worst texture	17.330000	23.410000	25.530000	26.500000	16.670000	23.750000	27.660000
worst perimeter	184.600000	158.800000	152.500000	98.870000	152.200000	103.400000	153.200000
worst area	2019.000000	1956.000000	1709.000000	567.700000	1575.000000	741.600000	1606.000000
worst smoothness	0.162200	0.123800	0.144400	0.209800	0.137400	0.179100	0.144200
worst compactness	0.665600	0.186600	0.424500	0.866300	0.205000	0.524900	0.257600
worst concavity	0.711900	0.241600	0.450400	0.686900	0.400000	0.535500	0.378400
worst concave points	0.265400	0.186000	0.243000	0.257500	0.162500	0.174100	0.193200
worst symmetry	0.460100	0.275000	0.361300	0.663800	0.236400	0.398500	0.306300
worst fractal dimension	0.118900	0.089020	0.087580	0.173000	0.076780	0.124400	0.083680

30 rows × 569 columns

In [24]:
```
from sklearn.preprocessing import StandardScaler
```

In [25]:
```
scaler = StandardScaler()
```

```
scaler.fit(df)
```

Out[25]:
```
StandardScaler()
```

In [26]:
```
scaled_data = scaler.transform(df)
```

In [27]:
```
from sklearn.decomposition import PCA
```

In [28]:
```
pca = PCA(n_components=2)
```

In [29]:
```
pca.fit(scaled_data)
```

Out[29]:
```
PCA(n_components=2)
```

In [30]:
```
x_pca = pca.transform(scaled_data)
```

In [31]:
```
scaled_data.shape
```

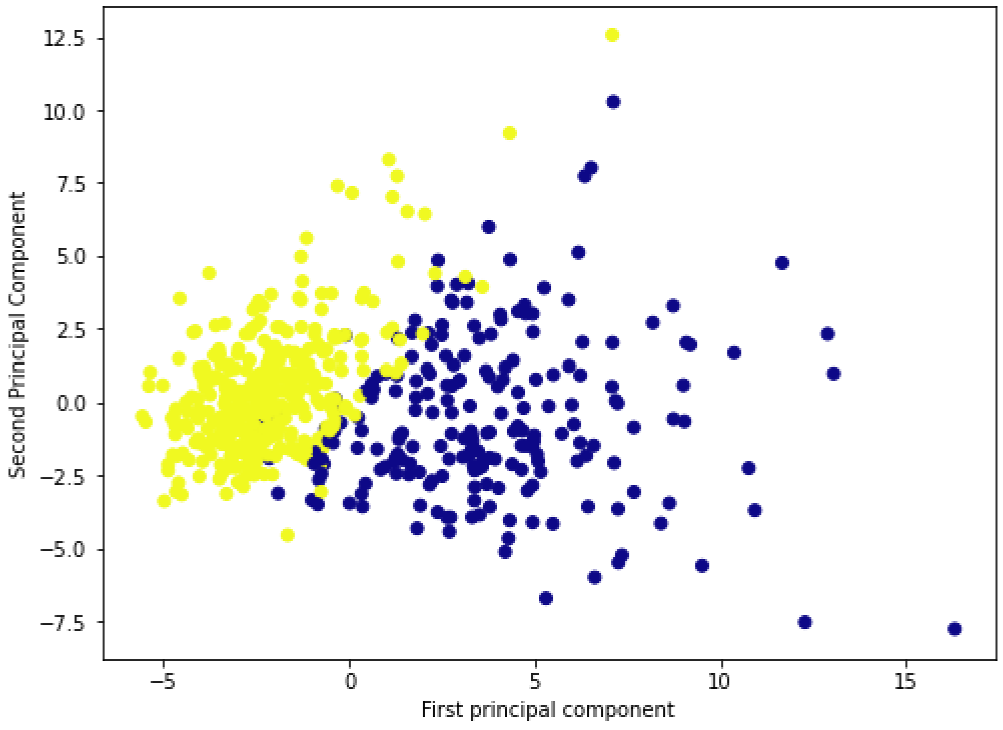
Out[31]:
```
(569, 30)
```

In [32]:
```
x_pca.shape
```

Out[32]:
```
(569, 2)
```

In [33]:
```
plt.figure(figsize=(8,6))
plt.scatter(x_pca[:,0],x_pca[:,1],c=cancer['target'],cmap='plasma')
plt.xlabel('1 Target')
plt.ylabel('2 Target')
```

Out[33]:
```
Text(0, 0.5, 'Second Principal Component')
```


Second Principal Component
First principal component
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−5
0
5
10
15

```
In [ ]:
```