

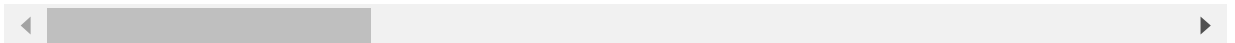
```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
```

```
In [2]: # Data Cllection
df = pd.read_csv("breast-cancer-data.csv", index_col=0)
df.head()
```

Out[2]:

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

5 rows × 31 columns



In [3]: *# Data Wrangling*

```
df_cancer = df
df_cancer.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Int64Index: 569 entries, 842302 to 92751
```

```
Data columns (total 31 columns):
```

#	Column	Non-Null Count	Dtype
0	diagnosis	569 non-null	object
1	radius_mean	569 non-null	float64
2	texture_mean	569 non-null	float64
3	perimeter_mean	569 non-null	float64
4	area_mean	569 non-null	float64
5	smoothness_mean	569 non-null	float64
6	compactness_mean	569 non-null	float64
7	concavity_mean	569 non-null	float64
8	concave points_mean	569 non-null	float64
9	symmetry_mean	569 non-null	float64
10	fractal_dimension_mean	569 non-null	float64
11	radius_se	569 non-null	float64
12	texture_se	569 non-null	float64
13	perimeter_se	569 non-null	float64
14	area_se	569 non-null	float64
15	smoothness_se	569 non-null	float64
16	compactness_se	569 non-null	float64
17	concavity_se	569 non-null	float64
18	concave points_se	569 non-null	float64
19	symmetry_se	569 non-null	float64
20	fractal_dimension_se	569 non-null	float64
21	radius_worst	569 non-null	float64
22	texture_worst	569 non-null	float64
23	perimeter_worst	569 non-null	float64
24	area_worst	569 non-null	float64
25	smoothness_worst	569 non-null	float64
26	compactness_worst	569 non-null	float64
27	concavity_worst	569 non-null	float64
28	concave points_worst	569 non-null	float64
29	symmetry_worst	569 non-null	float64
30	fractal_dimension_worst	569 non-null	float64

```
dtypes: float64(30), object(1)
```

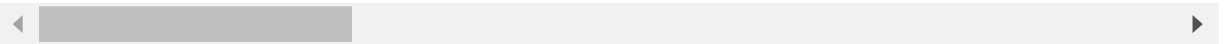
```
memory usage: 142.2+ KB
```

```
In [4]: df_cancer.drop(["diagnosis"], inplace=True, axis=1)
df_cancer.head()
```

Out[4]:

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

5 rows × 30 columns



```
In [5]: # Data Transformation
pca_model = PCA(n_components=2)
pca_model.fit(df_cancer)
transformed_data = pca_model.transform(df_cancer)
```

```
In [9]: # New Dataframe
new_df = pd.DataFrame(transformed_data)
new_df.index = df_cancer.index      #setting original index
new_df.columns = ["PC1", "PC2"]     #changing column names
new_df['diagnosis'] = df['diagnosis'] #result column
new_df.head()
```

```

-----
KeyError                                Traceback (most recent call last)
~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
    2645         try:
-> 2646             return self._engine.get_loc(key)
    2647         except KeyError:

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

```

KeyError: 'diagnosis'

During handling of the above exception, another exception occurred:

```

KeyError                                Traceback (most recent call last)
<ipython-input-9-553aad5d4d81> in <module>
      3 new_df.index = df_cancer.index      #setting original index
      4 new_df.columns = ["PC1", "PC2"]     #changing column names
----> 5 new_df['diagnosis'] = df['diagnosis'] #result column
      6 new_df.head()

~\anaconda3\lib\site-packages\pandas\core\frame.py in __getitem__(self, key)
    2798         if self.columns.nlevels > 1:
    2799             return self._getitem_multilevel(key)
-> 2800         indexer = self.columns.get_loc(key)
    2801         if is_integer(indexer):
    2802             indexer = [indexer]

~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
    2646         return self._engine.get_loc(key)
    2647         except KeyError:
-> 2648         return self._engine.get_loc(self._maybe_cast_indexer(key))
    2649         indexer = self.get_indexer([key], method=method, tolerance=tolerance)
    2650         if indexer.ndim > 1 or indexer.size > 1:

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

```

KeyError: 'diagnosis'

```
In [10]: # Variance Ratio
print(pca_model.explained_variance_ratio_)

[0.98204467 0.01617649]
```

```
In [ ]:
```