Task 1:

Create a dataframe variable 'a' with this dataset. This dataframe should have all the 569 instances, 30 features and the class of 569 instances as 0 (Malignant) or 1 (Benign). The column that contains the classes should be labeled as 'typeofcancer

```
from sklearn.datasets import load_breast_cancer

cancer = load_breast_cancer(return_X_y=True, as_frame=True)
a = cancer[0]
a["Type of Cancer"] = cancer[1]
a.shape
```

```
(569, 31)
```

b)

```
import pandas as pd
from pandas import DataFrame

cancer_type = load_breast_cancer().target

#Convert to DataFrame
b = DataFrame(a)

#Create new DataFrame
df = DataFrame()
df["Mean Radius"] = b.iloc[:,0]
df["Mean Perimeter"] = b.iloc[:,2]
df["Mean Area"] = b.iloc[:,3]
df["Type of Cancer"] = cancer_type
df.iloc[0:2, :]
```

	Mean Radius	Mean Perimeter	Mean Area	Type of Cancer
0	17.99	122.8	1001.0	0
1	20.57	132.9	1326.0	0

```
df.iloc[17:22, :]
```

	Mean Radius	Mean Perimeter	Mean Area	Type of Cancer
17	16.130	108.10	798.8	0
18	19.810	130.00	1260.0	0
19	13.540	87.46	566.3	1
20	13.080	85.63	520.0	1
21	9.504	60.34	273.9	1

Task 2:

i)

```
import matplotlib.pyplot as plt
import numpy as np

fig, ax = plt.subplots(1,3, figsize=(15,5))

## Gets the type of cancer in an array to iterate through
cancer_type_array = np.array(DataFrame(df["Type of Cancer"]))
malig_rad = []
benign_rad = []

#Iterates through and gets the mean radius of malignant and benign cases.
for index, instance in enumerate(cancer_type_array):
    if instance == 1:
        benign_rad.append(df.iloc[index,0])
    else:
        malig_rad.append(df.iloc[index,0])

ax[0].hist(malig_rad, edgecolor='blue', alpha=0.7, label='Malignant', linewidth=2, fill=False)
ax[0].hist(benign_rad, edgecolor='red', alpha=0.7, label='Benign', linewidth=2, fill=False)

ax[0].set_xlabel('Mean Radius')
ax[0].set_ylabel('Frequency')
ax[0].legend(loc='upper right')
ax[0].set_xticks(np.arange(10,30, step=5))
ax[0].set_yticks(np.arange(0,100, step=20))
```

```
malig_perim = []
benign_perim = []

#Iterates through and gets the mean perimeter of malignant and benign cases.
for index, instance in enumerate(cancer_type_array):
    if instance == 1:
        benign_perim.append(df.iloc[index,1])
    else:
        malig_perim.append(df.iloc[index,1])


ax[1].scatter(malig_perim, malig_rad, label='Malignant')
ax[1].scatter(benign_perim, benign_rad, label='Benign')


ax[1].set_xlabel("Mean Perimeter")
ax[1].set_ylabel("Mean Radius")
ax[1].legend(loc='upper left')
ax[1].set_xticks(np.arange(50,200, step=50))
ax[1].set_yticks(np.arange(10,30, step=5))
```

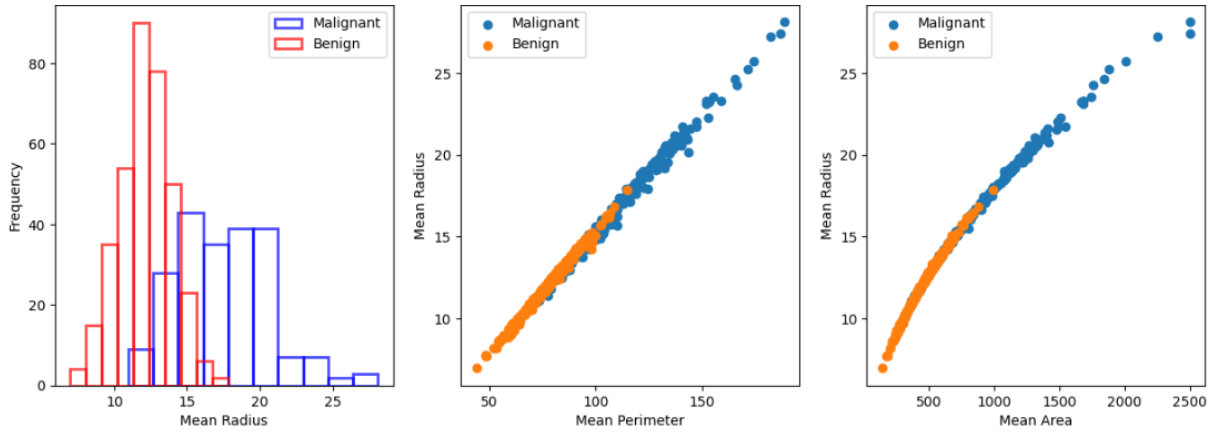
```
malig_area = []
benign_area = []

#Iterates through and gets the mean perimeter of malignant and benign cases.
for index, instance in enumerate(cancer_type_array):
    if instance == 1:
        benign_area.append(df.iloc[index,2])
    else:
        malig_area.append(df.iloc[index,2])

ax[2].scatter(malig_area, malig_rad, label='Malignant')
ax[2].scatter(benign_area, benign_rad, label='Benign')

ax[2].set_xlabel("Mean Area")
ax[2].set_ylabel("Mean Radius")
ax[2].legend(loc='upper left')
ax[2].set_xticks(np.arange(500,3000, step=500))
ax[2].set_yticks(np.arange(10,30, step=5))

fig
```




```
'''
The first scatterplot reveals that malignant cases of breast cancer have a greater mean perimeter and area of the tumours.
Also, these two pieces of data have a linear correlation given that circumference has a relationship with radius of
Circumference = 2*pi*radius

The second scatterplot also reveals that malignant cases of breast cancer have a greater mean area of tumours. However, this
relationship is more parabolic as area and radius are related by the equation Area = pi*radius^2

'''
```

Task 3:

```
# Addition of data
df["Mean Concavity"] = b.iloc[:,6]
df["Mean Concave Points"] = b.iloc[:,7]
df["Mean Symmetry"] = b.iloc[:,8]

# Creation of new subplot
fig2, ax2 = plt.subplots(1,4, figsize=(15,5))

#Addition of same first histogram
ax2[0].hist(malig_rad, edgecolor='blue', alpha=0.7, label='Malignant', linewidth=2, fill=False)
ax2[0].hist(benign_rad, edgecolor='red', alpha=0.7, label='Benign', linewidth=2, fill=False)

ax2[0].set_xlabel('Mean Radius')
ax2[0].set_ylabel('Frequency')
ax2[0].legend(loc='upper right')
ax2[0].set_xticks(np.arange(10,30, step=10))
ax2[0].set_yticks(np.arange(0,100, step=20))
```

```
malig_conc_pts = []
benign_conc_pts = []

#Iterates through and gets the mean concavity points of malignant and benign cases.
for index, instance in enumerate(cancer_type_array):
    if instance == 1:
        benign_conc_pts.append(df.iloc[index,5])
    else:
        malig_conc_pts.append(df.iloc[index,5])


ax2[2].scatter(malig_conc_pts, malig_rad, label='Malignant')
ax2[2].scatter(benign_conc_pts, benign_rad, label='Benign')


ax2[2].set_xlabel("Mean Concave Points")
ax2[2].set_ylabel("Mean Radius")
ax2[2].legend(loc='lower right')
ax2[2].set_xticks(np.arange(0.0,0.3, step=0.1))
ax2[2].set_yticks(np.arange(10,30, step=5))
```

```
malig_concavity = []
benign_concavity = []

#Iterates through and gets the mean concavity of malignant and benign cases.
for index, instance in enumerate(cancer_type_array):
    if instance == 1:
        benign_concavity.append(df.iloc[index,4])
    else:
        malig_concavity.append(df.iloc[index,4])


ax2[1].scatter(malig_concavity, malig_rad, label='Malignant')
ax2[1].scatter(benign_concavity, benign_rad, label='Benign')


ax2[1].set_xlabel("Mean concavity")
ax2[1].set_ylabel("Mean Radius")
ax2[1].legend(loc='upper left')
ax2[1].set_xticks(np.arange(0.0,0.6, step=0.2))
ax2[1].set_yticks(np.arange(10,30, step=5))
```

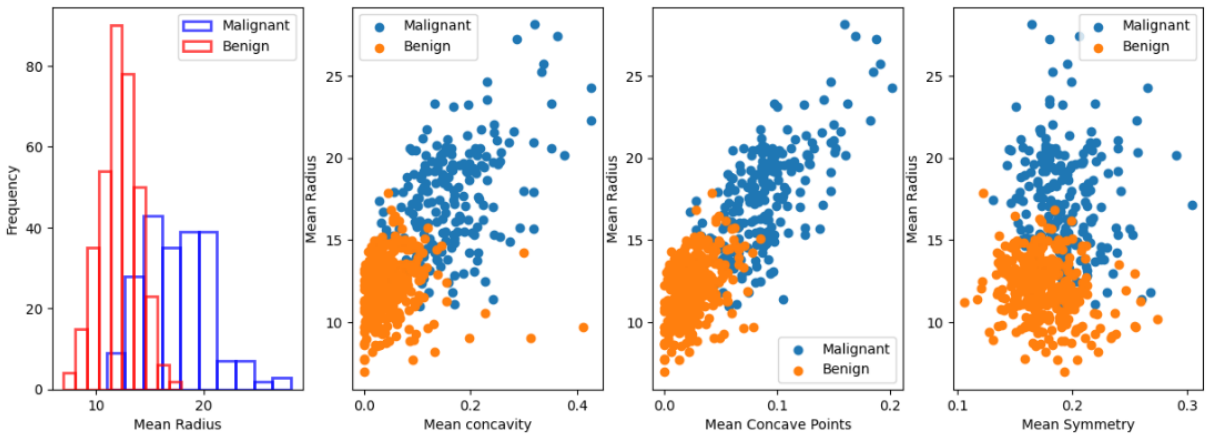
```
malig_symmetry = []
benign_symmetry = []

#Iterates through and gets the mean symmetry of malignant and benign cases.
for index, instance in enumerate(cancer_type_array):
    if instance == 1:
        benign_symmetry.append(df.iloc[index,6])
    else:
        malig_symmetry.append(df.iloc[index,6])


ax2[3].scatter(malig_symmetry, malig_rad, label='Malignant')
ax2[3].scatter(benign_symmetry, benign_rad, label='Benign')


ax2[3].set_xlabel("Mean Symmetry")
ax2[3].set_ylabel("Mean Radius")
ax2[3].legend(loc='upper right')
ax2[3].set_xticks(np.arange(0.1,0.35, step=0.1))
ax2[3].set_yticks(np.arange(10,30, step=5))

fig2
```




```
'''
The first scatterplot reveals that malignant cases of breast cancer tend to have more severe concavity points in the cells measured. This is despite there being no apparent formulaic correlation between the data on the x and y axis.

The second scatterplot also shows that malignant cases of breast cancer tend to have more concave points in the cells measured than benign cases. The correlation between the two datasets is slightly more linear, perhaps when we begin the machine learning portion we will get a clearer idea.

The final scatterplot reveals something different to those prior. It seems that regardless of the cells being malignant or benign, they seem to have the same mean symmetry. This data also seems to have little to no correlation, but perhaps the machine learning segment will show us differently.

'''
```

Task 4:

```
import pandas as pd
from pandas import DataFrame

df = DataFrame()
df["499 or Less"] = [203, 217, 237, 222, 250, 300, 309, 305, 340, 322, 110, 118, 135, 122, 150, 185, 190, 187, 205, 195]
df["500-999"] = [208,222,242,227,255,305,314,310,345,327,115,123,140,127,155,190,195,192,210,200]
df["1000-1499"] = [213, 227, 247, 232, 260, 310, 319, 315, 350, 332, 120, 128, 145, 132, 160, 195, 200, 197, 215, 205]
df["1500-1999"] = [218, 232, 252, 237, 265, 315, 324, 320, 355, 337, 125, 133, 150, 137, 165, 200, 205, 202, 220, 210]
df["2000-2999"] = [221, 235, 255, 240, 268, 318, 327, 323, 358, 340, 128, 136, 153, 140, 168, 203, 208, 205, 223, 213]
df["3000-3999"] = [223, 237, 257, 242, 270, 320, 329, 325, 360, 342, 130, 138, 155, 142, 170, 205, 210, 207, 225, 215]
df["4000-4999"] = [226, 240, 260, 245, 273, 323, 332, 328, 363, 345, 133, 141, 158, 145, 173, 208, 213, 210, 228, 218]
df["5000+"] = [228, 242, 262, 247, 275, 325, 334, 330, 365, 347, 135, 143, 160, 147, 175, 210, 215, 212, 230, 220]
df["Status"] = ["A","A","A","A","A","P","P","P","P","P","A","A","A","A","A","P","P","P","P","P"]
df.index = ["John Smith (D)", "Tiger Aldrin (D)", "Jeremy Cole (D)", "Lee West (D)", "Warren Buff (D)", "Waldo Where (D)",
            "Patrick Reel (D)", "Vijay Love (D)", "Greta Lindstrom (D)","Jeffrey Bautista (D)", "John Smith (7)",
            "Tiger Aldrin (7)", "Jeremy Cole (7)", "Lee West (7)", "Warren Buff (7)", "Waldo Where (7)", "Patrick Reel (7)",
            "Vijay Love (7)", "Greta Lindstrom (7)","Jeffrey Bautista (7)"]

df
```

	499 or Less	500-999	1000-1499	1500-1999	2000-2999	3000-3999	4000-4999	5000+	Status
John Smith (D)	203	208	213	218	221	223	226	228	A
Tiger Aldrin (D)	217	222	227	232	235	237	240	242	A
Jeremy Cole (D)	237	242	247	252	255	257	260	262	A
Lee West (D)	222	227	232	237	240	242	245	247	A
Warren Buff (D)	250	255	260	265	268	270	273	275	A
Waldo Where (D)	300	305	310	315	318	320	323	325	P
Patrick Reel (D)	309	314	319	324	327	329	332	334	P
Vijay Love (D)	305	310	315	320	323	325	328	330	P
Greta Lindstrom (D)	340	345	350	355	358	360	363	365	P
Jeffrey Bautista (D)	322	327	332	337	340	342	345	347	P
John Smith (7)	110	115	120	125	128	130	133	135	A
Tiger Aldrin (7)	118	123	128	133	136	138	141	143	A
Jeremy Cole (7)	135	140	145	150	153	155	158	160	A
Lee West (7)	122	127	132	137	140	142	145	147	A
Warren Buff (7)	150	155	160	165	168	170	173	175	A
Waldo Where (7)	185	190	195	200	203	205	208	210	P
Patrick Reel (7)	190	195	200	205	208	210	213	215	P
Vijay Love (7)	187	192	197	202	205	207	210	212	P
Greta Lindstrom (7)	205	210	215	220	223	225	228	230	P
Jeffrey Bautista (7)	195	200	205	210	213	215	218	220	P

```
'''
This dataframe analyzes the distance that golfers of different skill levels hit the ball(in yards) at different elevations
(in feet) with their driver and 7-iron.

Number of instances: 20
Number of attributes: 8
List of attributes:
    Driving/7-iron distance at:
        - 499 feet or less
        - 500-999 feet
        - 1000-1499 feet
        - 1500-1999 feet
        - 2000-2999 feet
        - 3000-3999 feet
        - 4000-4999 feet
        - 5000 feet or more
List of classes:
    Various skilled golfers determine by status:
        - Professional (P)
        - Amateur (A)
Creator: Sam Laquerre
'''
```

Data Resource: Nickel, Chris. 2023a. "ARCCOS 2021 Distance Report." MyGolfSpy, February. https://mygolfspy.com/news-opinion/arccos-2021-distance-report/. Data not used, but inspiration for the dataset.