Assignment 1 [part 1 of 2]

For the below assignment questions, questions 1-3 are pen-and-paper exercises and do not require code. Questions 4-6 are coding exercises. For all questions, please **do make sure to show your workings** - i.e. how did you derive the result, by writing down your thinking, writing down the relevant equations or math derivations, or by showing the code that was used to generate the result that addresses the question.

1. Consider the following sales data: [3, 16, 20, 4, 2, 5, 10, 9, 13, 7, 14, 8]. Apply the following binning techniques on the data, assuming 3 bins in each case:

   A. Equal-frequency binning
   B. Smoothing by bin boundaries
   [0.5 marks out of 5]

1. Use the below methods to normalize the following data: [10, 5, 25, 50, 35]:

   A. min-max normalization with min=0 and max=1.
   B. z-score normalization
   [0.5 marks out of 5]

1. Students at two universities, University A and University B, have been provided with feedback forms on student satisfaction, with the below responses recorded. Is student satisfaction correlated with a specific university? Use a chi-square test to find out, assuming a significance level of 0.001 and a corresponding chi-square significance value of 10.828. [1 mark out of 5]

Rating/University	University A	University B
Satisfied	71	129
Dissatisfied	37	73

1. Load the CSV file country-income.csv which includes both numerical and categorical attributes. Perform data cleaning in order to replace any NaN values with the mean of the value for a given field. Then replace any categorical labels with numerical labels. Display the resulting dataset. You can use the sklearn.impute and sklearn.preprocessing packages to asssist you. [1 mark out of 5]

1. Load the CSV file shoesize.csv, which includes measurements of shoe size and height (in inches) for 408 subjects, both female and male. Plot the scatterplots of shoe size versus height for female and male subjects separately. Compute the Pearson's correlation coefficient of shoe size versus height for female and male subjects separately. What can be inferred by the scatterplots and computed correlation coefficients? You can implement your own formulation of the correlation coefficient or use the scipy.stats package to assist you. [1 mark out of 5]

1. Using the breast cancer dataset from section 1 of this notebook, perform Principal Component Analysis with 2 components. Compute the explained variance ratio for each component, and plot the scatterplot of all samples along the two principal components, color-coded according to the "Class" column (this column should not be used in the PCA analysis). Ensure that your data is normalized prior to performing PCA. What insights can you obtain by the explained variance ratio of each component, and by viewing the scatterplot of the principal components? [1 mark out of 5]

Sample Solutions [part 1 of 2]

Solution to exercise 1:

First, we sort the data: [2, 3, 4, 5, 7, 8, 9, 10, 13, 14, 16, 20].

```
(a) Bin 1: 2, 3, 4, 5
    Bin 2: 7, 8, 9, 10
    Bin 3: 13, 14, 16, 20
    (0.25 marks out of 5)
(b) Bin 1: 2, 2, 5, 5
    Bin 2: 7, 7, 10, 10
    Bin 3: 13, 13, 13, 20
    (0.25 marks out of 5)
```

Solution to exercise 2:

(a) $v' = \frac{v-5}{50-5}(1-0)+0 \Rightarrow [0.11, 0, 0.44, 1.0, 0.66]$.

$(0.25$ marks out of $5)$

(b) $v' = \frac{v-\bar{A}}{\sigma_A}$

$A = \frac{1}{5}(10+5+25+50+35) = 25$

$\sigma_A = \sqrt{\frac{1}{5}(10^2+5^2+25^2+50^2+35^2) - \bar{A}^2} = 16.43$

$\Rightarrow [-0.91, -1.21, 0, 1.52, 0.61]$.

$(0.25$ marks out of $5)$

Solution to exercise 3:

In order to calculate the χ^2 value we use the following formula: $\chi^2 = \sum_{i=1}^{c}\sum_{j=1}^{r}\frac{(o_{ij}-e_{ij})^2}{e_{ij}}$. The observed counts o_{ij} are recorded in the above table.

The expected counts e_{ij} are calculated using the following equation: $e_{ij} = \frac{\text{count}(A=a_i)\cdot\text{count}(B=b_j)}{n}$ and are as follows:

Rating/University	University A	University B
Satisfied	69.67	130.32
Dissatisfied	38.32	71.67

Then, the χ^2 value is:

$$\chi^2 = \frac{(71-69.67)^2}{69.67} + \frac{(129-130.32)^2}{130.32} + \frac{(37-38.32)^2}{38.32} + \frac{(73-71.67)^2}{71.67} = 0.1089.$$

Therefore, since 0.1089 is much smaller compared to the chi-square significance value of 10.828, the independence hypothesis is accepted with a significance level of 0.001. There is no correlation between student satisfaction levels and the two universities, in other words there is no difference in student satisfaction between the two universities.

Marking scheme: 0.25 marks for calculating the expected counts, 0.25 marks for calculating the χ^2 value, 0.5 marks for answering whether student satisfaction is correlated with a specific university.

Solution to exercise 4:

In [1]:

```python
# marking scheme:
# 0.5 marks for replacing NaN values with the mean
# 0.5 marks for replacing categorical labels with numerical labels
import pandas as pd
import numpy as np
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import LabelEncoder

# Load CSV file
country_income = pd.read_csv('country-income.csv', header='infer')


# Replace NaN string with python NaN value
country_income = country_income.replace('NaN',np.NaN)

# Replacing NaN values with the mean
X = country_income.values
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
imputer = imputer.fit(X[:, 1:3]) # impute only the numerical columns!
X[:, 1:3] = imputer.transform(X[:, 1:3])
print(X)

# Encode categorical variables into numerical labels
labelencoder_X = LabelEncoder()
X[:, 0] = labelencoder_X.fit_transform(X[:, 0])
X[:, 3] = labelencoder_X.fit_transform(X[:, 3])
print(X)
```

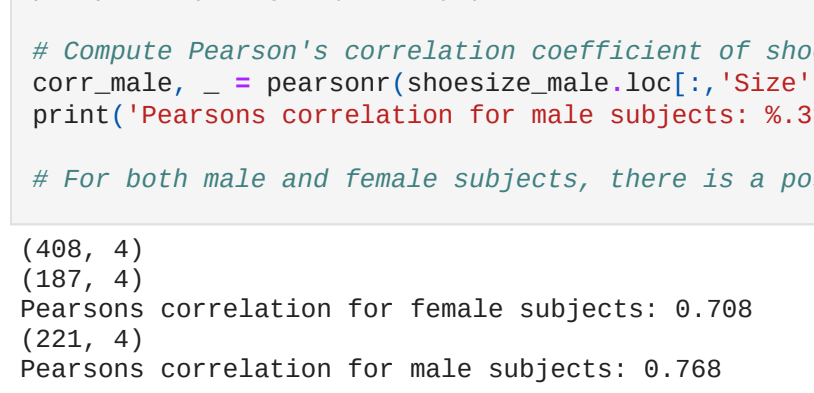
```
[['India' 49.0 86400.0 'No']
 ['Brazil' 32.0 57600.0 'Yes']
 ['USA' 35.0 64800.0 'No']
 ['Brazil' 43.0 73200.0 'No']
 ['USA' 45.0 76533.33333333333 'Yes']
 ['India' 40.0 69600.0 'Yes']
 ['Brazil' 43.77777777777778 62400.0 'No']
 ['India' 53.0 94800.0 'Yes']
 ['USA' 55.0 99600.0 'No']
 ['India' 42.0 80400.0 'Yes']]
[[1 49.0 86400.0 0]
 [0 32.0 57600.0 1]
 [2 35.0 64800.0 0]
 [0 43.0 73200.0 0]
 [2 45.0 76533.33333333333 1]
 [1 40.0 69600.0 1]
 [0 43.77777777777778 62400.0 0]
 [1 53.0 94800.0 1]
 [2 55.0 99600.0 0]
 [1 42.0 80400.0 1]]
```

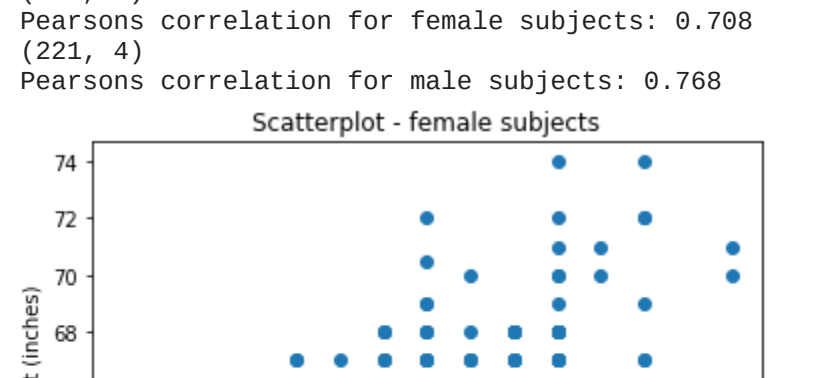
Solution to exercise 5:

In [2]:

```python
# marking scheme:
# 0.3 marks for plotting the 2 scatterplots of shoe size versus height
# 0.3 marks for computing the two correlation coefficients of shoe size versus height
# 0.4 marks for stating that there is a positive correlation between shoe size and he
import matplotlib.pyplot as plt
from scipy.stats import pearsonr

# Open CSV file
shoesize = pd.read_csv('shoesize.csv', header='infer')
shoesize.head()
print(shoesize.shape)

# Select rows for female subjects
shoesize_female = shoesize.loc[shoesize['Gender'] == 'F']
print(shoesize_female.shape)

# Print scatterplot of shoe size vs height for female subjects
plt.figure()
plt.scatter(shoesize_female.loc[:,'Size'],shoesize_female.loc[:,'Height'])
plt.title("Scatterplot - female subjects")
plt.xlabel('Shoe size')
plt.ylabel('Height (inches)')

# Compute Pearson's correlation coefficient of shoe size vs height for female subjects
corr_female, _ = pearsonr(shoesize_female.loc[:,'Size'], shoesize_female.loc[:,'Height
print('Pearsons correlation for female subjects: %.3f' % corr_female)

# Select rows for male subjects
shoesize_male = shoesize.loc[shoesize['Gender'] == 'M']
print(shoesize_male.shape)

# Print scatterplot of shoe size vs height for male subjects
plt.figure()
plt.scatter(shoesize_male.loc[:,'Size'],shoesize_male.loc[:,'Height'])
plt.title("Scatterplot - male subjects")
plt.xlabel('Shoe size')
plt.ylabel('Height (inches)')

# Compute Pearson's correlation coefficient of shoe size vs height for male subjects
corr_male, _ = pearsonr(shoesize_male.loc[:,'Size'], shoesize_male.loc[:,'Height'])
print('Pearsons correlation for male subjects: %.3f' % corr_male)

# For both male and female subjects, there is a positive linear correlation between sh
```

```
(408, 4)
(187, 4)
Pearsons correlation for female subjects: 0.708
(221, 4)
Pearsons correlation for male subjects: 0.768
```





Solution to exercise 6:

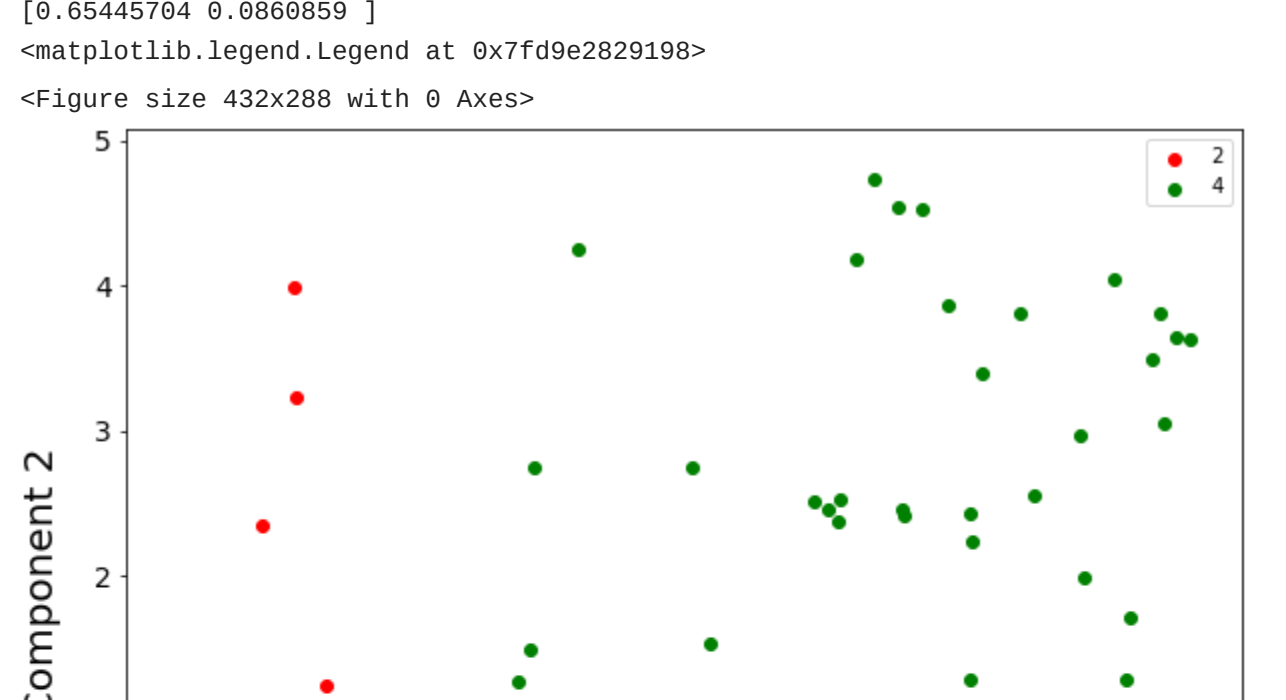
In [3]:

```python
# marking scheme:
# 0.3 marks for peforming PCA with 2 components on the breast cancer dataset;
# 0.2 marks for computing the explained variance ratio for each component;
# 0.2 marks for plotting the scatterplot of all samples along the two principal compon
# 0.3 marks for discussing insights from the explained variance ratio and the PC scatt

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
get_ipython().run_line_magic('matplotlib', 'inline')

#Import data
df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/breast-can
df.columns = ['Sample code', 'Clump Thickness', 'Uniformity of Cell Size', 'Uniformity
              'Marginal Adhesion', 'Single Epithelial Cell Size', 'Bare Nuclei', 'Bl
              'Normal Nucleoli', 'Mitoses','Class']
#Pre-process
df= df.drop(['Sample code'],axis=1)
df = df.replace('?',np.NaN)
df = df.fillna(df.median())
df['Bare Nuclei'] = pd.to_numeric(df['Bare Nuclei'])

#df.head()

col=df.columns
features=col.tolist()
feature=features[:-1]
target=features[-1]
X=df.loc[:,feature].values
y=df.loc[:,target].values

#Standard Scaling
sc=StandardScaler()
X=sc.fit_transform(X)
pd.DataFrame(X,columns=feature).head()

#PCA
pca=PCA(n_components=2)
principalComponents=pca.fit_transform(X)
principalDf=pd.DataFrame(data=principalComponents,columns=['principal component 1','pr
principalDf.head()
finalDf=pd.concat([principalDf,df[['Class']]],axis=1)
finalDf.head(100)
print(pca.explained_variance_ratio_)

plt.figure()
plt.figure(figsize=(10,10))
plt.xticks(fontsize=12)
plt.yticks(fontsize=14)
plt.xlabel('Principal Component 1',fontsize=20)
plt.ylabel('Principal Component 2',fontsize=20)
targets = [2, 4]
colors = ['r', 'g']
for target, color in zip(targets,colors):
    indicesToKeep = finalDf['Class'] == target
    plt.scatter(finalDf.loc[indicesToKeep, 'principal component 1'], finalDf.loc[indic

plt.legend(targets,prop={'size': 10})

# Discussion:
# The first principal component explains 65.4% of the data variance, however the 2nd
# principal component explains only 8.6% of the data variance.
# The scatterplot of the principal components shows that the two classes
# are (relatively) separable using the two principal components.
```

```
[0.65445704 0.0860859 ]
```

Out[3]:

```
<matplotlib.legend.Legend at 0x7fd9e2829198>
<Figure size 432x288 with 0 Axes>
```



Assignment 1 [part 2 of 2]

For your answers to the assignment, please include include your workings (e.g. equations, code) when this is relevant to the question.

1. In Section 1, what kind of relationship can be inferred from summary statistics regarding `ACT composite score` and `SAT total score`? Which visualisations make this relationship apparent? [0.5 marks out of 5]
2. Based on the box plots presented in Section 1, what is the relationship between `parental level of education` and `parental income`? Using table visualisation, find and show the entire rows that correspond to the outliers regarding `parental income` whose parents have a master's degree. [0.5 marks out of 5]
3. Using an example, explain the importance of scaling features so that their magnitudes are comparable when computing distances. [0.5 marks out of 5]
4. In Section 1, the distance matrix visualisation is not very informative. However, it is still possible to infer that the average distance between students whose parents only have some high school education and students whose parents have a master's degree is larger than the average distance between students whose parents only have some high school education. Explain how this inference is possible from the visualisation. [0.5 marks out of 5]
5. In Section 2, increase the number of evenly spaced numbers from 10 to 100 for both axes and observe the corresponding heat map created through nearest neighbour interpolation. Read about this interpolation method and explain what you observed. [0.5 marks out of 5]
6. The function `load_wine` from `sklearn.datasets` can be used to load the *wine dataset* into a `DataFrame` by using the commands `data = load_wine()`, `df = pd.DataFrame(data.data, columns=data.feature_names)`, and `df['target'] = pd.Series(data.target)`.

   6.1. Load the wine dataset. Compute the frequency of each value of the 'target' feature. [0.5 marks out of 5]

   6.2. Compute univariate and multivariate summaries for all numerical features (except from the target feature). Group observations by the target feature and compute the corresponding **median** for each numerical feature. [0.5 marks out of 5]

   6.3. Group observations by the target feature and create one box plot of `alcohol` for each group. [0.5 marks out of 5]

   6.4. Create a scatter plot for the pair of **distinct** numerical features with the highest correlation. [0.5 marks out of 5]

   6.5. Exclude the target feature, standardize the remaining numerical features, and display a projection obtained by multidimensional scaling. Color the points by the target feature. [0.5 marks out of 5]

Solution to exercises 1-5 (brief)

1. Highly correlated. Scatterplot.
2. Also correlated. The rows that should be shown can be found by sorting the parental income after filtering students whose parents have a master's degree.
3. Any example where a feature with a large range would dominate distance computations.
4. The average color in the submatrix that corresponds to distances between students whose parents only have some high school education is lighter.
5. The resolution is increased, the plot is much smoother.

Solution to exercise 6

In [4]:

```python
# Imports
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.datasets import load_wine
from sklearn.preprocessing import StandardScaler
from sklearn.manifold import MDS

# Configuring seaborn output
%config InlineBackend.figure_formats = set(['retina'])
sns.set_style('darkgrid')
```

In [5]:

```python
#6.1 Loading dataset as DataFrame - 0.2 marks
data = load_wine()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = pd.Series(data.target)

# Counting number of elements of each target (class) - 0.3 marks
print(df['target'].value_counts())
```

```
1    71
0    59
2    48
Name: target, dtype: int64
```

In [6]:

```python
#6.2 Displaying univariate and multivariate summaries for all features except target (
display(df.drop(columns='target').describe())
display(df.drop(columns='target').corr())
```

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflava
count	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	
mean	13.000618	2.336348	2.366517	19.494944	99.741573	2.295112	2.029270	
std	0.811827	1.117146	0.274344	3.339564	14.282484	0.625851	0.998859	
min	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0.340000	
25%	12.362500	1.602500	2.210000	17.200000	88.000000	1.742500	1.205000	
50%	13.050000	1.865000	2.360000	19.500000	98.000000	2.355000	2.135000	
75%	13.677500	3.082500	2.557500	21.500000	107.000000	2.800000	2.875000	
max	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000	5.080000	

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols
alcohol	1.000000	0.094397	0.211545	-0.310235	0.270798	0.289101
malic_acid	0.094397	1.000000	0.164045	0.288500	-0.054575	-0.335167
ash	0.211545	0.164045	1.000000	0.443367	0.286587	0.128980
alcalinity_of_ash	-0.310235	0.288500	0.443367	1.000000	-0.083333	-0.321113
magnesium	0.270798	-0.054575	0.286587	-0.083333	1.000000	0.214401
total_phenols	0.289101	-0.335167	0.128980	-0.321113	0.214401	1.000000
flavanoids	0.236815	-0.411007	0.115077	-0.351370	0.195784	0.864564
nonflavanoid_phenols	-0.155929	0.292977	0.186230	0.361922	-0.256294	-0.449935
proanthocyanins	0.136698	-0.220746	0.009652	-0.197327	0.236441	0.612413
color_intensity	0.546364	0.248985	0.258887	0.018732	0.199950	-0.055136
hue	-0.071747	-0.561296	-0.074667	-0.273955	0.055398	0.433681
od280/od315_of_diluted_wines	0.072343	-0.368710	0.003911	-0.276769	0.066004	0.699949
proline	0.643720	-0.192011	0.223626	-0.440597	0.393351	0.498115

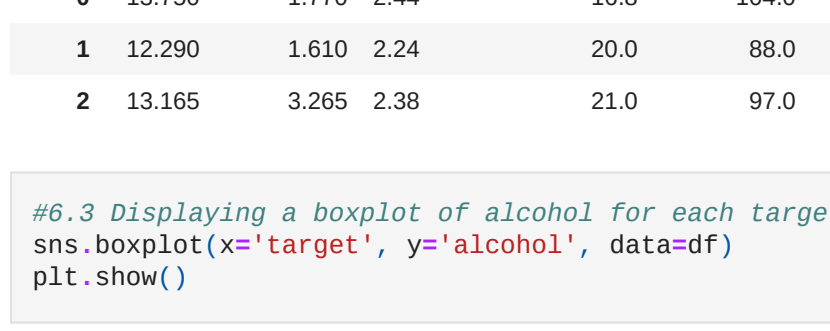
In [7]:

```python
# Grouping observations by target (class) and computing the corresponding median for e
display(df.groupby('target').median())
```

target	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phen
0	13.750	1.770	2.44	16.8	104.0	2.800	2.980	0
1	12.290	1.610	2.24	20.0	88.0	2.200	2.030	0
2	13.165	3.265	2.38	21.0	97.0	1.635	0.685	0

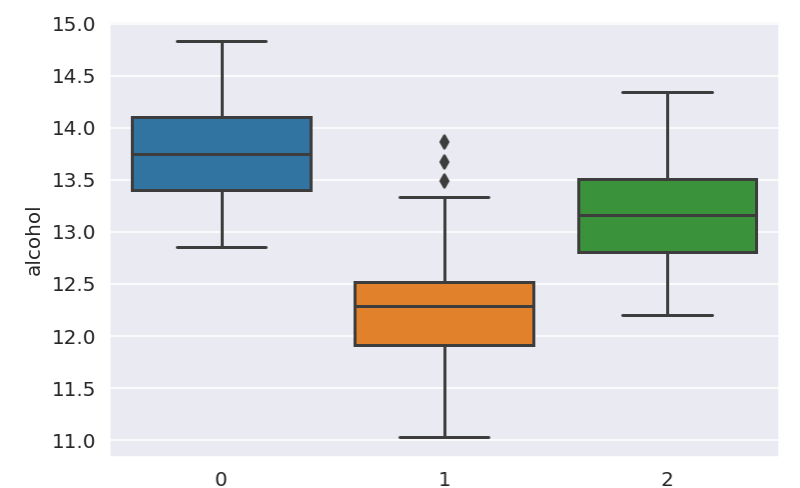
In [8]:

```python
#6.3 Displaying a boxplot of alcohol for each target (class) - 0.5 marks
sns.boxplot(x='target', y='alcohol', data=df)
plt.show()
```



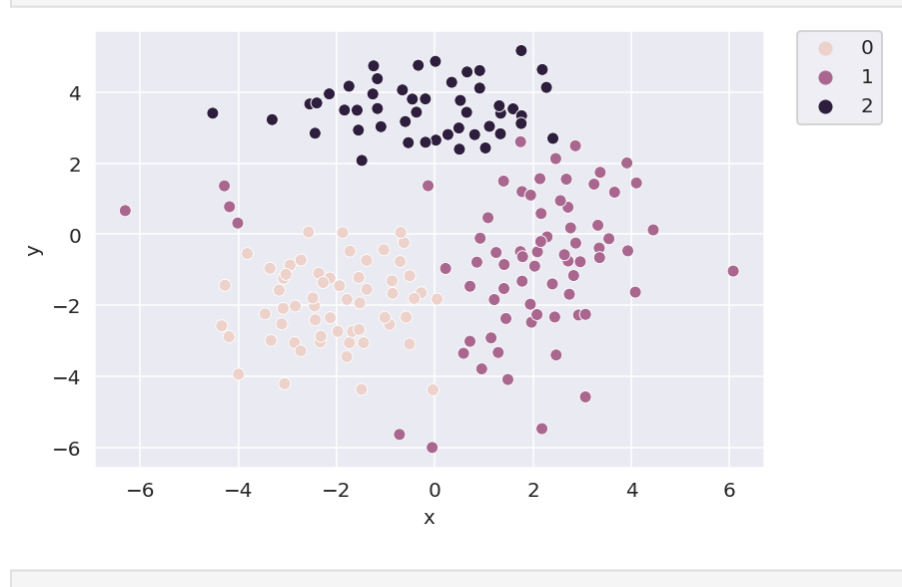
In [9]:

```python
#6.4 Displaying a scatter plot for the two distinct features with the highest correla
sns.scatterplot(x='flavanoids', y='total_phenols', data=df)
plt.show()
```



In [10]:

```python
#6.5 Standardizing features while excluding target (class) - 0.25 marks
X = df.drop(columns='target').to_numpy()
scaler = StandardScaler()
X = scaler.fit_transform(X)
# Computing a projection using MDS and displaying the resulting projection - 0.25 mark
embedding = MDS(n_components=2)
Xp = embedding.fit_transform(X)
df_projection = pd.DataFrame({'x': Xp[:, 0], 'y': Xp[:, 1],
                              'target': df['target']})
sns.scatterplot(x='x', y='y', hue='target', data=df_projection)
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.show()
```



In [ ]: