# Problem 1:

- Define knowledge discovery in databases
    - The process of extracting knowledge/patterns from datasets
- Briefly describe the steps of the knowledge discovery in the database process
    - Data selection
        - Selecting relevant data from sources
    - Data preprocessing
        - Filter out bad data, remove unwanted fields and select the correct attributes
    - Data transformation
        - Normalization, aggregation, and making composite attributes might occur here.
- Define the term, data mining
    - Datamining is a way to find implicit, unknown patterns in a dataset.

# Problem 2

1) Missing GDPPC value

- One way would be to calculate the average and insert it into the missing data field. That value would be 18.2. The issue with this is that the missing value could be vastly different from the average, meaning absolutely nothing to the dataset as a whole.
- Another way would be to find this data in another dataset, normalize it accordingly, and merge it into the previous data. The issue is that new data may not have been recorded in the same manner, or measured at a different point in time leading to an incorrect and irrelevant result to our data mining.
- A final way to do this would be to generate the new value in a sort of weighted sense. Generate a correlation between SWL and GDPPC and draw an estimated prediction from there.

2) keeping country name

- Keeping the country name in the dataset while data mining wouldn't necessarily make sense - the name of the country does not provide any useful data that aids in the discovery of patterns.

3) Transformation of country attribute (max 4 values)

- This attribute could be changed into which continent each country is a part of, bundling the Americas into one.

4) Discretize the AC-S-ED attribute by binning it into 4 equi-width intervals using unsupervised discretization.- sort the data: - 5.2, 49.9, 73.4, 79, 81.9, 94.6, 99, 99.9, 102.1, 102.6, 103.2, 108.7

- calculate the range

  - max - min: 108.7 - 5.2 = 103.5
  - Width = range/4 : 103.5/4 = 25.875
- Intervals should be:
  - [5.2, 31.075)
  - [31.075, 56.95)
  - [56.95, 82.825)
  - [82.825, 108.7]
- Explanation:
  - Each interval was calculated with the same width of 25.875.

5) Discretize the AC-S-ED attribute by binning it into 4 equi-depth (= equal- frequency) intervals using unsupervised discretization

- sort the data:
  - 5.2, 49.9, 73.4, 79, 81.9, 94.6, 99, 99.9, 102.1, 102.6, 103.2, 108.7
- calculate the frequency
  - Total number of points = 12
  - Threshold = total / intervals: 12/4 = 3
- Intervals should be:
  - [5.2, 79)
  - [79, 99.9)
  - [99.9, 102.6)
  - [102.6, 108.7]
- Each interval was calculated to have 3 entries.

6) Mean = 83, stdv = 30; [mean − (k + 1) × sd, mean − k × sd) for k = -4 to 2

- -1: [83 - (-1 + 1) x 30, 83 - (-1) x 30) = [83, 113)
- 0 : [83 - (0 + 1) x 30, 83 - (0) x 30) = [53, 83)
- 1 : [83 - (1 + 1) x 30, 83 - (1) x 30) = [23, 53)
- 2 : [83 - (2 + 1) x 30, 83 - (2) x 30) = [-7, 23)

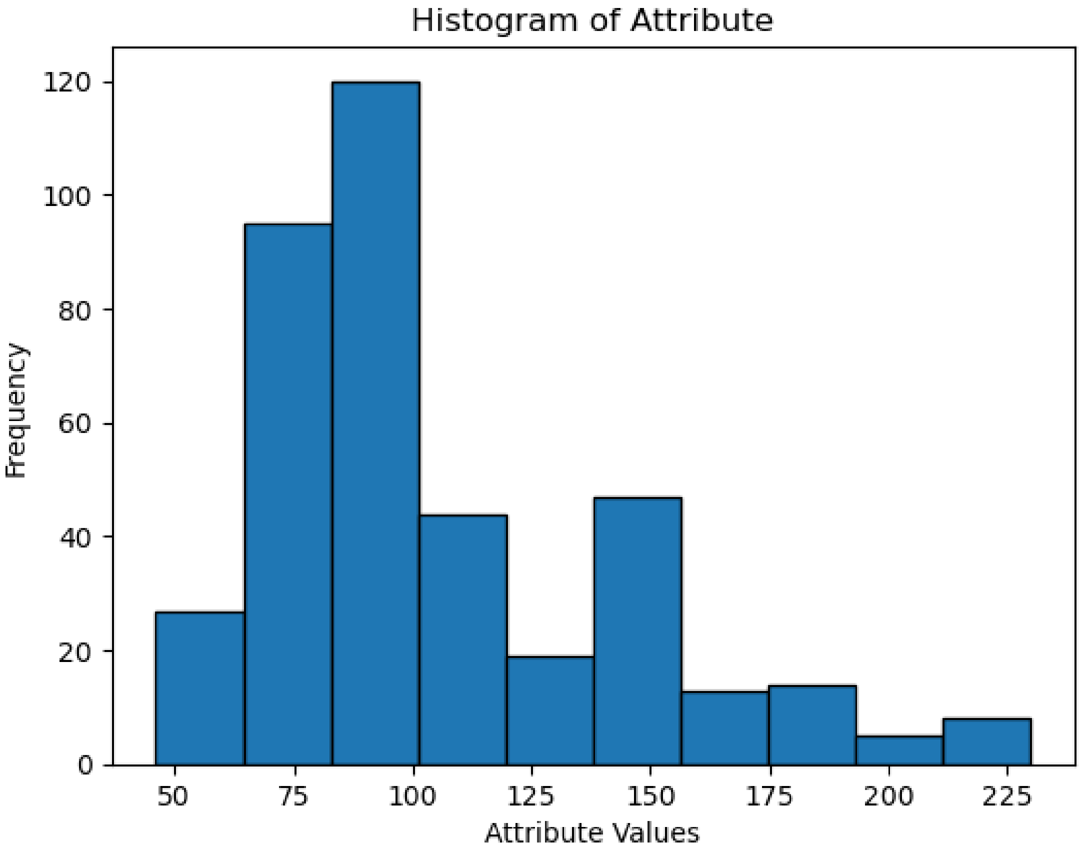
# Problem 3.1

- Good:
  - The data is already normalized even though it is from multiple sources
  - The data is already sorted from oldest to newest, allowing for some human-interpretable patterns to emerge.
- Bad:
  - Some of the data is missing, left with a '?'.
  - Some data from different sources is being mislabeled. For example, one of the sources uses Volkswagen, and the other abbreviates it as VW

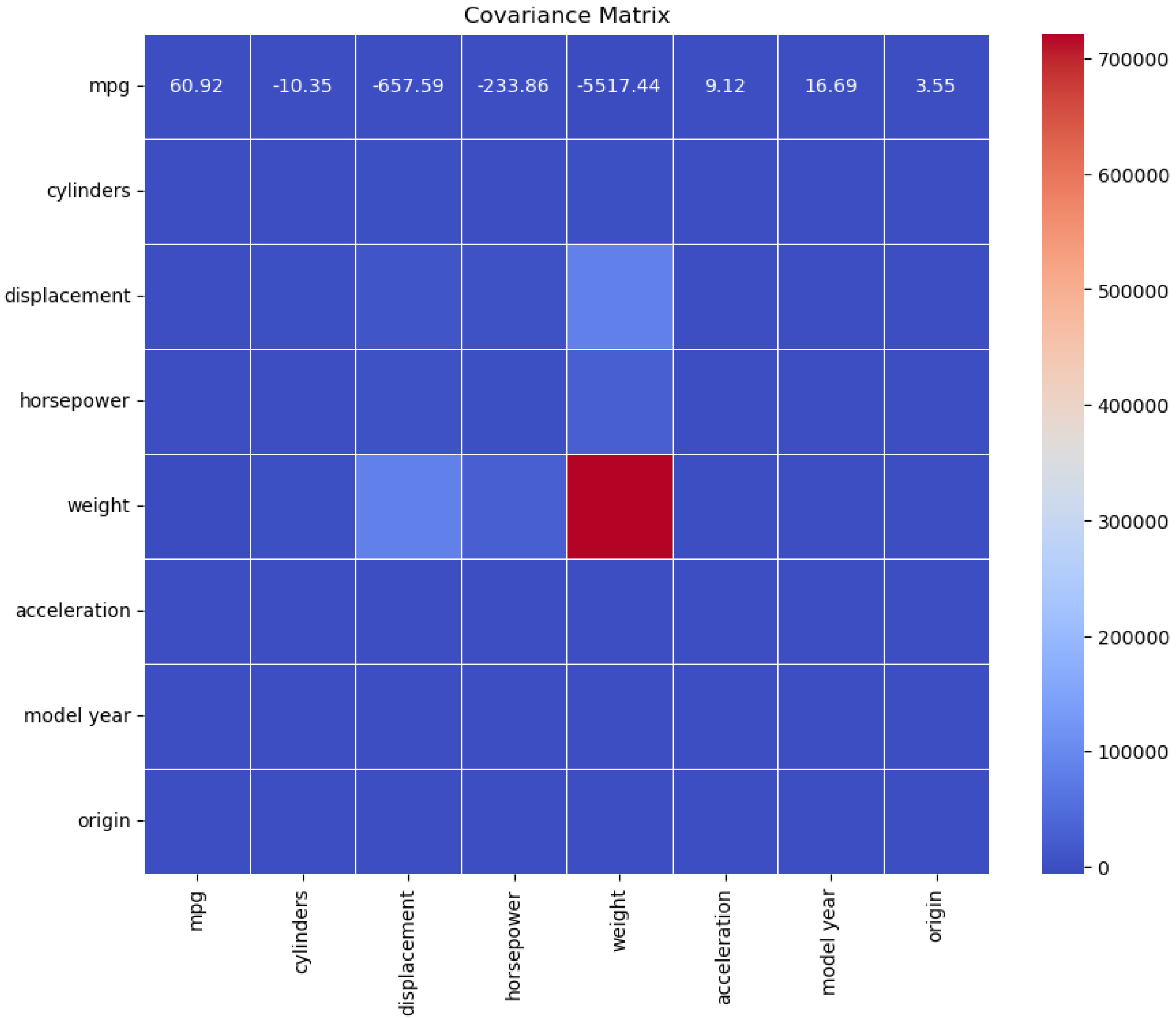
```
# Problem 3.2
    # Calculate the percentiles in increments of 10, the mean, median,
# range, and variance.
```

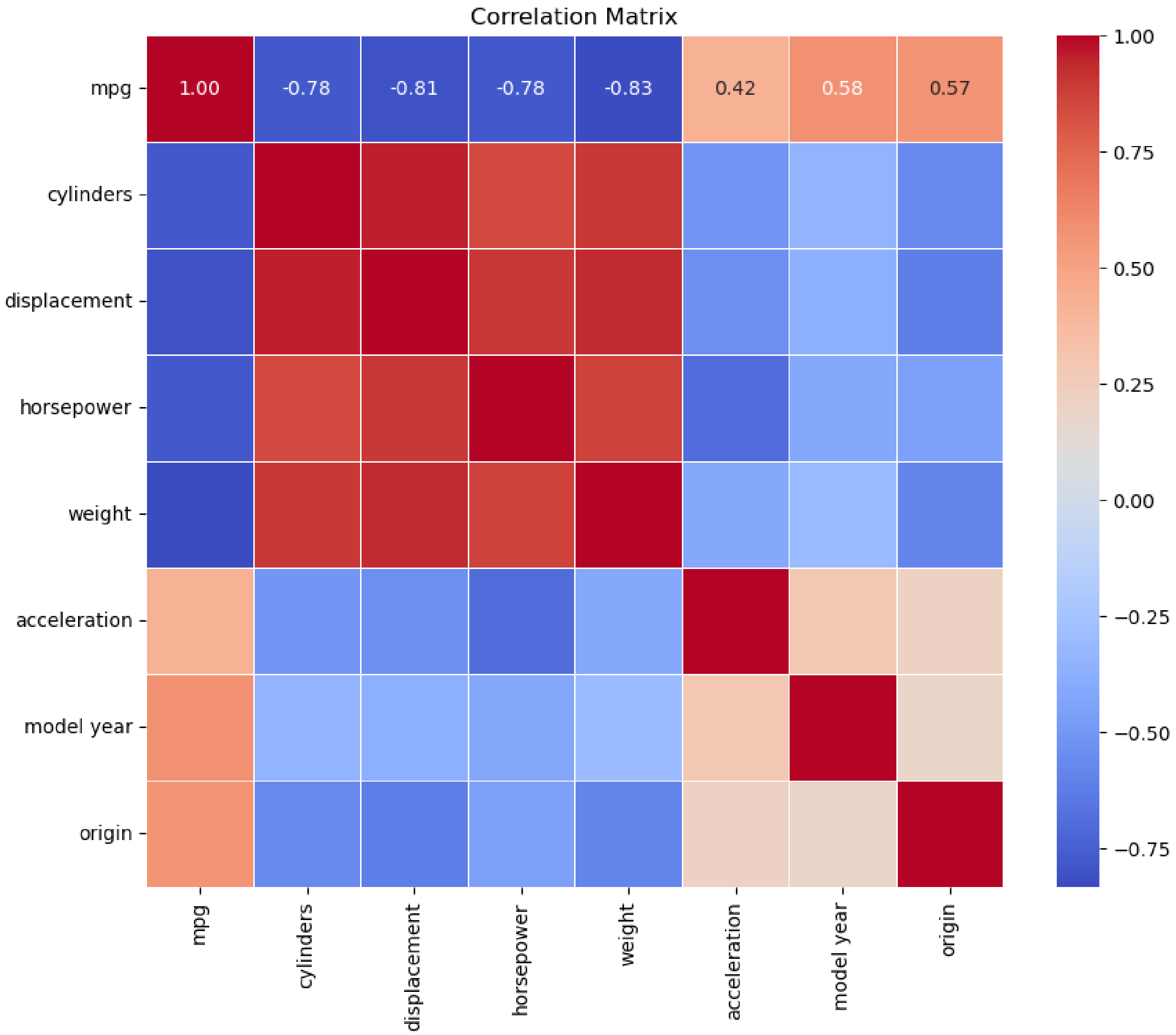
```
    # Plot a histogram of the attribute using 10 or 20 bins.

import pandas as PD
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv("auto-mpg.csv")
attribute_values = data["horsepower"].tolist()
attribute_values = list(filter(lambda i : i != '?', attribute_values))
attribute_values = [int(i) for i in attribute_values]

percentiles = np.percentile(attribute_values, np.arange(0, 100, 10))

mean_value = np.mean(attribute_values)
median_value = np.median(attribute_values)
data_range = np.ptp(attribute_values)
variance_value = np.var(attribute_values)

print("Percentiles (in increments of 10):", percentiles)
print("Mean:", mean_value)
print("Median:", median_value)
print("Range:", data_range)
print("Variance:", variance_value)

plt.hist(attribute_values, bins=10, edgecolor='black')
plt.xlabel("Attribute Values")
plt.ylabel("Frequency")
plt.title("Histogram of Attribute")
plt.show()
```

```
Percentiles (in increments of 10): [ 46.   67.   72.   80.   88.
 93.5 100.  110.  140.  157.7]
Mean: 104.46938775510205
Median: 93.5
Range: 184
Variance: 1477.789879216993
```



```
# Problem 3.3
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv('auto-mpg.csv', na_values='?')
df_subset = df.drop(columns=['car name'])

df_subset = df_subset.dropna()

cov_matrix = df_subset.cov()

corr_matrix = df_subset.corr()

plt.figure(figsize=(10, 8))
sns.heatmap(cov_matrix, annot=True, cmap='coolwarm', fmt=".2f",
linewidths=.5)
plt.title('Covariance Matrix')
plt.show()

plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f",
linewidths=.5)
plt.title('Correlation Matrix')
plt.show()
```





# Problem 3.4

- If I had to remove 2 attributes, one would be the origin and the other would be the acceleration. Neither of these values directly gives useful information to any overall conclusion

```
# Problem 3.5
import pandas as pd
from sklearn.decomposition import PCA
import numpy as np

df = pd.read_csv('auto-mpg.csv', na_values='?')
df_subset = df.drop(columns=['car name'])
df_subset = df_subset.dropna()

pca = PCA()
```

```
pca.fit(df_subset)

original_dimensions = df_subset.shape[1]
dimensions_after_pca = pca.components_.shape[0]

variance_explained = pca.explained_variance_ratio_
total_variance_explained = np.sum(variance_explained)

print("Original dimensions:", original_dimensions)
print("Dimensions after PCA:", dimensions_after_pca)
print("Variance explained by each component:", variance_explained)
print("Total variance explained:", total_variance_explained)

first_component = pca.components_[0]
print("First principal component:", first_component)
```

```
Original dimensions: 8
Dimensions after PCA: 8
Variance explained by each component: [9.97536268e-01 2.06325558e-03
3.56501852e-04 3.16851311e-05
 7.55318054e-06 3.89264255e-06 4.91057912e-07 3.52144547e-07]
Total variance explained: 1.0
First principal component: [-7.59590581e-03  1.79257460e-03
1.14338202e-01  3.89660894e-02
  9.92644743e-01 -1.35281211e-03 -1.33689886e-03 -5.51527155e-04]
```

# Problem 3.5 explanation

- In most cases, the PCA decreased the variance of the entries

```
# Problem 3.6
import pandas as pd
from sklearn.decomposition import PCA
import numpy as np

df = pd.read_csv('auto-mpg.csv', na_values='?')

df['car brand'] = df['car name'].apply(lambda x: x.split()[0])
df = df.drop(columns=['car name'])
df = df.dropna()
df['car brand'] = pd.Categorical(df['car brand'])
df = pd.get_dummies(df, columns=['car brand'])

pca = PCA()
pca.fit(df)

original_dimensions = df.shape[1]
dimensions_after_pca = pca.components_.shape[0]
```

```
variance_explained = pca.explained_variance_ratio_
total_variance_explained = np.sum(variance_explained)

print("Original dimensions:", original_dimensions)
print("Dimensions after PCA:", dimensions_after_pca)
print("Variance explained by each component:", variance_explained)
print("Total variance explained:", total_variance_explained)
```

```
Original dimensions: 45
Dimensions after PCA: 45
Variance explained by each component: [9.97535015e-01 2.06326072e-03
3.56515337e-04 3.16956755e-05
 7.55992595e-06 3.89769963e-06 5.34896606e-07 3.62794992e-07
 1.58064537e-07 1.29354705e-07 1.01812961e-07 8.95163838e-08
 8.08844566e-08 6.60031912e-08 5.64723769e-08 5.55717681e-08
 4.74813955e-08 4.14300021e-08 3.74111239e-08 3.49013108e-08
 2.74155547e-08 2.52921133e-08 2.48812489e-08 2.20073547e-08
 1.98076395e-08 1.82807371e-08 1.58994665e-08 1.37893312e-08
 1.31709085e-08 1.11063654e-08 1.04581674e-08 7.82170969e-09
 7.43157125e-09 6.76337469e-09 6.58918954e-09 4.19340501e-09
 4.06227877e-09 3.48291798e-09 3.47513218e-09 3.43560728e-09
 3.39168778e-09 3.33182451e-09 3.12228845e-09 6.90616825e-33
 6.02331432e-33]
Total variance explained: 1.0
```

# Problem 3.6 explanation

- There was a lot more dimensions present here now that the model of the cars has been removed.