- In your own words, define data mining. Describe the steps involved in data mining

  Data mining is the process of extracting and manipulating various forms of datasets

  The steps consit of the following

  - Cleaning (delete/ filter only needed metrics)

  - Integrate (Combine and create uniformity in the retrieved datasets)

  - Filter (Select releted metrics based on the project objective)

  - Transformation (Wrangle and sort consplidate)

  - Review Patterns for insights

- Explain the difference between discrimination and classification, characterization and clustering, and classification and regression. Support your answers with clear examples.

  - Difference between discrimination and classification: Comparing general features for a population

  - Similarities between discrimination and classification: Both measure nominal data types and analyze objects.

  - Difference between characterization and clustering: Models or functions to describe or distinguish data classes to model and predict. Summary of general characteristics or features of the target population or sample size.

  - Similarities between characterization and clustering: Grouping of objects or related data to compare against data set values.

  - Difference between classification and regression: classification is the process of finding a set of models at of the population or sample.

  - Regression predicts data that it isn't available at the time of the analysis this data is often numerical data values.

  - Similarities between classification and regression; both are used to predict possible trends; one object data type the other numerical values.

- Suppose that the data for analysis includes the feature age. The age values for the data tuples are: 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 36, 40, 45, 46, 52, 70

In [4]:

```
ages = [13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 36, 40, 45, 46, 52, 70]
import numpy as np

# This is the number that lies in the middle of a list of ordered numbers
np.median(ages)
```

Out[4]:

```
25.0
```

In [7]:

```
# This is the average of all the numbers
np.mean(ages)
```

Out[7]:

```
29.56
```

In [10]:

```
from scipy import stats
# This is the most frequent number in the list
stats.mode(ages)
```

Out[10]:

```
ModeResult(mode=array([25]), count=array([4]))
```

In [13]:

```
# Mid Range
(min(ages) + max(ages)) / 2
```

Out[13]:

```
41.5
```

In [17]:

```
# Five Number Summary
from scipy import stats
stats.describe(ages)
```

Out[17]:

```
DescribeResult(nobs=25, minmax=(13, 70), mean=29.56, variance=179.17333333333332, skewness=1.2
404417486852026, kurtosis=1.4671174178082689)
```

the sample minimum (smallest observation)

the lower quartile or first quartile

the median (the middle value)

the upper quartile or third quartile

the sample maximum (largest observation)

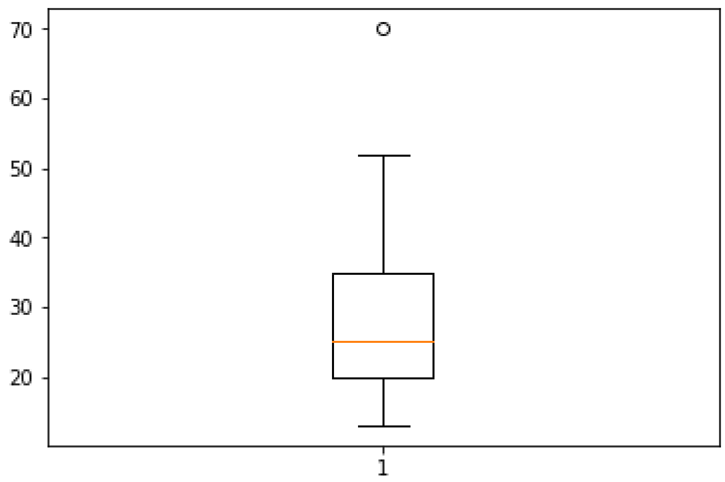
In [19]:

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

# Box Plot of Ages
plt.boxplot(ages)
plt.figure()
```

Out[19]:

```
<Figure size 432x288 with 0 Axes>
```



```
<Figure size 432x288 with 0 Axes>
```

- Briefly outline how to compute the dissimilarity between objects described by the following (include mathematical notation)
    - Nominal attributes
        - A categorical variable is a generalization of the binary variable in that it can take on more than two states.

          The dissimilarity between two objects i and j can be computed based on the ratio of mismatches: $d(i, j) = (p - m)/P$ where m is the number of matches (i.e., the number of variables for which i and j are in the same state), and p is the total number of variables. Alternatively, we can use a large number of binary variables by creating a new binary variable for each of the M nominal states. For an object with a given state value, the binary variable representing that state is set to 1, while the remaining binary variables are set to 0.
        - Asymmetric binary attributes
            - In computing the dissimilarity between asymmetric binary variables, the number of negative matches, t, is considered unimportant and thus is ignored in the computation, that is,
        - Numeric attributes
        - Use Euclidean distance, Manhattan distance, or supremum distance. Euclidean distance is defined as $d(i, j) = \sqrt{}(xi1 - xj1)2 + (xi2 - xj2)2 + \cdots + (xin - xjn)2$

where $i = (xi1, xi2, \ldots, xin)$, and $j = (xj1, xj2, \ldots, xjn)$, are two n-dimensional data objects. The Manhattan (or city block) distance, is defined as $d(i, j) = |xi1 - xj1| + |xi2 - xj2| + \cdots + |xin - xjn|$

(d) Term-frequency vectors To measure the distance between complex objects represented by vectors, it is often easier to abandon traditional metric distance computation and introduce a nonmetric similarity function For example, the similarity between two vectors, x and y, can be defined as a cosine measure, as follows:

$s(x, y) = xt \cdot y \, ||x||||y||$

where xt is a transposition of vector x, ||x|| is the Euclidean norm of vector x,1 ||y|| is the Euclidean norm of vector y, and s is essentially the cosine of the angle between vectors x and y.