Assignment 6

1. Open the file in a text editor while you read over the column names in the data file details at the end of this assignment.

Note, the extension is '.dat', and even though it is a text file your operating system may not recognize it as one. One option is to change the extension (to something like '.txt'). Another is to start your text editor and open the file through the editor. In any case, doing this will give you a chance to observe the structure of the data while you read the data file details.

2. Read the text file in as a dataframe. And then print it to make sure it read in correctly. Keep in mind this can be a tricky, error-prone problem depending on the formatting of the data file.

Use the method you used previously to read in a CSV file. There are at least two potential problems to consider. Note, if the tutorial is insufficient you can simply read the function documentation online for how to handle the following issues.

The delimiter is a space rather than a comma: One option is to open the file in a text editor, and use the Find/Replace All option to change every space to a single comma. However, that's cumbersome and error-prone. A better option is to just use the read CSV function and change the separator to a space rather than a comma. There is no header line: You have a description of each column, but the file does not have column headers. You may need to make sure the read function is set up so it does not expect column headers. However, a more common option is to supply your own 14 item list of strings to serve as column headers - you can use the one below as an example. Using names of columns will make it much easier to read your code later and is highly recommended for this step.

['age','sex','chest_pain','bp','chol','sugar','ecg','heart_rate','angina','oldpeak','slope','vessel','thal','disea

In [17]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [18]:
```
header=['age','sex','chest_pain','bp','chol','sugar','ecg',
        'heart_rate','angina','oldpeak','slope','vessel','thal','disease']
df=pd.read_csv('C:\\Users\\swapn\\OneDrive\\Desktop\\heart.dat',sep=" ",names
print(df)
```

```
      age  sex  chest_pain     bp   chol  sugar  ecg  heart_rate  angina  \
0    70.0  1.0         4.0  130.0  322.0    0.0  2.0       109.0     0.0
1    67.0  0.0         3.0  115.0  564.0    0.0  2.0       160.0     0.0
2    57.0  1.0         2.0  124.0  261.0    0.0  0.0       141.0     0.0
3    64.0  1.0         4.0  128.0  263.0    0.0  0.0       105.0     1.0
4    74.0  0.0         2.0  120.0  269.0    0.0  2.0       121.0     1.0
..    ...  ...         ...    ...    ...    ...  ...         ...     ...
265  52.0  1.0         3.0  172.0  199.0    1.0  0.0       162.0     0.0
266  44.0  1.0         2.0  120.0  263.0    0.0  0.0       173.0     0.0
267  56.0  0.0         2.0  140.0  294.0    0.0  2.0       153.0     0.0
268  57.0  1.0         4.0  140.0  192.0    0.0  0.0       148.0     0.0
269  67.0  1.0         4.0  160.0  286.0    0.0  2.0       108.0     1.0

     oldpeak  slope  vessel  thal  disease
0        2.4    2.0     3.0   3.0        2
1        1.6    2.0     0.0   7.0        1
2        0.3    1.0     0.0   7.0        2
3        0.2    2.0     1.0   7.0        1
4        0.2    1.0     1.0   3.0        1
..       ...    ...     ...   ...      ...
265      0.5    1.0     0.0   7.0        1
266      0.0    1.0     0.0   7.0        1
267      1.3    2.0     0.0   3.0        1
268      0.4    2.0     0.0   6.0        1
269      1.5    2.0     3.0   3.0        2

[270 rows x 14 columns]
```

3. Now let's get a sense of the ages of the people. First print only the ages of the patients by selecting only that column. Then calculate the mean and standard deviation of the ages.

There are two ways you can calculate the mean. The easiest is to select the age column and use the mean function that can instantly be performed on columns in a data frame. The second option is to convert the column of data into an array, then perform the operation as you would for an array. Do this for the mean and the standard deviation.

(optional) you could create a histogram with this data too. For that you may need to convert the data to a list or array after selecting that column.

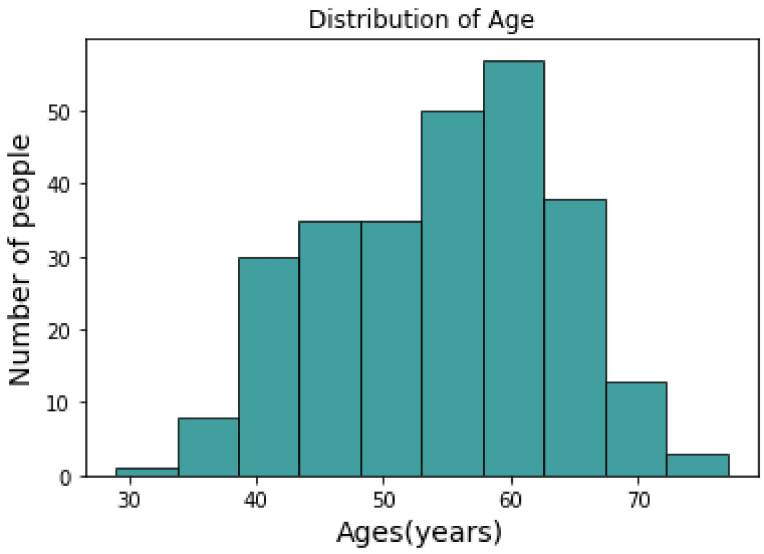
In [19]:
```
print(df['age'].mean(),"and",df['age'].std())
```

```
54.43333333333333 and 9.109066523898203
```

In [20]:
```
age=df['age'].to_numpy()
print(age)
```

```
[70. 67. 57. 64. 74. 65. 56. 59. 60. 63. 59. 53. 44. 61. 57. 71. 46. 53.
 64. 40. 67. 48. 43. 47. 54. 48. 46. 51. 58. 71. 57. 66. 37. 59. 50. 48.
 61. 59. 42. 48. 40. 62. 44. 46. 59. 58. 49. 44. 66. 65. 42. 52. 65. 63.
 45. 41. 61. 60. 59. 62. 57. 51. 44. 60. 63. 57. 51. 58. 44. 47. 61. 57.
 70. 76. 67. 45. 45. 39. 42. 56. 58. 35. 58. 41. 57. 42. 62. 59. 41. 50.
 59. 61. 54. 54. 52. 47. 66. 58. 64. 50. 44. 67. 49. 57. 63. 48. 51. 60.
 59. 45. 55. 41. 60. 54. 42. 49. 46. 56. 66. 56. 49. 54. 57. 65. 54. 54.
 62. 52. 52. 60. 63. 66. 42. 64. 54. 46. 67. 56. 34. 57. 64. 59. 50. 51.
 54. 53. 52. 40. 58. 41. 41. 50. 54. 64. 51. 46. 55. 45. 56. 66. 38. 62.
 55. 58. 43. 64. 50. 53. 45. 65. 69. 69. 67. 68. 34. 62. 51. 46. 67. 50.
 42. 56. 41. 42. 53. 43. 56. 52. 62. 70. 54. 70. 54. 35. 48. 55. 58. 54.
 69. 77. 68. 58. 60. 51. 55. 52. 60. 58. 64. 37. 59. 51. 43. 58. 29. 41.
 63. 51. 54. 44. 54. 65. 57. 63. 35. 41. 62. 43. 58. 52. 61. 39. 45. 52.
 62. 62. 53. 43. 47. 52. 68. 39. 53. 62. 51. 60. 65. 65. 60. 60. 54. 44.
 44. 51. 59. 71. 61. 55. 64. 43. 58. 60. 58. 49. 48. 52. 44. 56. 57. 67.]
```

In [21]:
```
num_bins=10
n,bins,patches=plt.hist(age,num_bins,facecolor='teal',alpha=0.75,edgecolor='b
plt.title('Distribution of Age')
plt.xlabel('Ages(years)',fontsize=14)
plt.ylabel('Number of people',fontsize=14)
plt.show()
```



4. Create a data frame called young_df with all the people below 55 years old. Similarly, create a data frame called old_df with everyone at or above 55 years old. Now calculate the mean age, blood pressure, cholesterol, and heart rate for both groups and note the trend as people age.

In [23]:
```
young_df=df[df['age']<55]
print("The average for young people: \n")
avg_young=young_df[['age','bp','chol','heart_rate']].mean()
print(avg_young)

old_df=df[df['age']>=55]
print("The average for old people: \n")
avg_old=old_df[['age','bp','chol','heart_rate']].mean()
print(avg_old)
```

```
The average for young people:

age            46.757576
bp            125.916667
chol          238.810606
heart_rate    157.636364
dtype: float64
The average for old people:

age            61.775362
bp            136.536232
chol          260.036232
heart_rate    142.065217
dtype: float64
```

5. Using all the data, let's try to see if there tend to be overall differences between people with heart disease and people without it. Find the mean age, blood pressure, cholesterol, and max heart rate for people with heart disease, and separately for the people without heart disease. This can be done easily using a grouping method for your data frame. Note the trend in each of these between people with heart disease and those without.

Such information provides indications of which features that can be used to predict who is at risk of heart disease.

Data file details (the meaning of each column)

Attribute Information by column order.

```
-- 1. age
-- 2. sex
-- 3. chest pain type  (4 values)
-- 4. resting blood pressure
  -- 5. serum cholesterol in mg/dl
-- 6. fasting blood sugar > 120 mg/dl
-- 7. resting electrocardiographic results  (values 0,1,2)
-- 8. maximum heart rate achieved
   -- 9. exercise induced angina
-- 10. oldpeak = ST depression induced by exercise relative to rest
   -- 11. the slope of the peak exercise ST segment
   -- 12. number of major vessels (0-3) colored by fluoroscopy
-- 13. thal: 3 = normal; 6 = fixed defect; 7 = reversible defect
   -- 14. Absence (1) or Presence (2) of heart disease
```

Attributes types

Real: 1,4,5,8,10,12 Ordered:11, Binary: 2,6,9,14 Nominal:7,3,13

In [25]:
```
heart_disease=df.groupby('disease')
print(heart_disease[['age','bp','chol','heart_rate']].mean())
```

```
                age          bp        chol  heart_rate
disease
1         52.706667  128.866667  244.213333  158.333333
2         56.591667  134.441667  256.466667  138.858333
```