

In [3]:

```
from sklearn import datasets
iris = datasets.load_iris()
```

In [7]:

```
import numpy as np
import pandas as pd
data1 = pd.DataFrame(data = np.c_[iris['data'], iris['target']],
                     columns = iris['feature_names'] + ['target'] )
print(data1.head)
```

```
<bound method NDFrame.head of      sepal length (cm)  sepal width (cm)  petal length (cm)  petal wi
dth (cm)  \
0          5.1          3.5          1.4          0.2
1          4.9          3.0          1.4          0.2
2          4.7          3.2          1.3          0.2
3          4.6          3.1          1.5          0.2
4          5.0          3.6          1.4          0.2
..          ...          ...          ...          ...
145         6.7          3.0          5.2          2.3
146         6.3          2.5          5.0          1.9
147         6.5          3.0          5.2          2.0
148         6.2          3.4          5.4          2.3
149         5.9          3.0          5.1          1.8
```

```
      target
0         0.0
1         0.0
2         0.0
3         0.0
4         0.0
..         ...
145        2.0
146        2.0
147        2.0
148        2.0
149        2.0
```

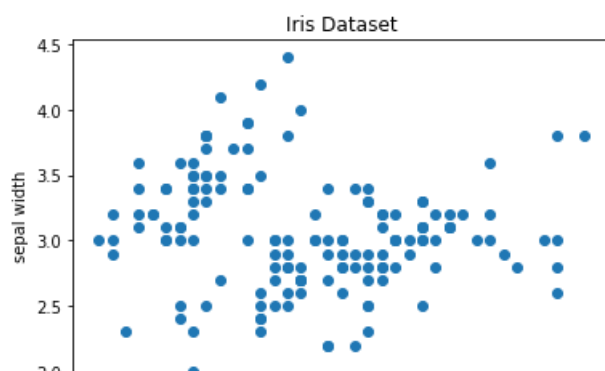
[150 rows x 5 columns]>

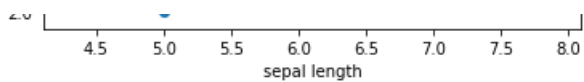
In [14]:

```
import matplotlib.pyplot as plt
fig , ax = plt.subplots()
# scatter the sepal length against the sepal width
ax.scatter(data1['sepal length (cm)'], data1['sepal width (cm)'])
#set a title and labels
ax.set_title('Iris Dataset')
ax.set_xlabel('sepal length')
ax.set_ylabel('sepal width')
```

Out[14]:

Text(0, 0.5, 'sepal width')





In [10]:

```
list(data1.columns)
```

Out[10]:

```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)',
 'target']
```

In [16]:

```
print(data1.target)
```

```
0      0.0
1      0.0
2      0.0
3      0.0
4      0.0
```

...

```
145     2.0
146     2.0
147     2.0
148     2.0
149     2.0
```

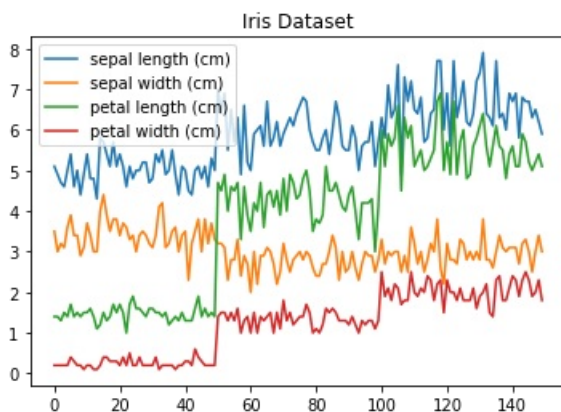
Name: target, Length: 150, dtype: float64

In [17]:

```
#get columns to plot(to plot all the columns)
columns = data1.columns.drop(['target'])
#create x data
x_data = range(0, data1.shape[0])
#create figure and axis
fig , ax =plt.subplots()
#plot each column
for column in columns:
    ax.plot(x_data, data1[column], label=column)
#set title and legend
ax.set_title('Iris Dataset')
ax.legend()
```

Out[17]:

<matplotlib.legend.Legend at 0x233b5110a88>



In [18]:

```
import pandas as pd
```

```
wine_reviews = pd.read_csv ("")
#print(df)
#create a figure and axis
fig, ax = plt.subplots()
#count the occurrences of each class
data = wine_reviews['Proline'].value_counts()
#get x and y data
points = data.index
frequency = data.values
# create bar chart
ax.bar(points, frequency)
#set title and labels
ax.set_title('Wine Review Scores')
ax.set_xlabel('Points')
ax.set_ylabel('Frequency')
```

FileNotFoundError Traceback (most recent call last)

<ipython-input-18-fe2f03d89c16> in <module>

```
1 import pandas as pd
----> 2 wine_reviews = pd.read_csv ("")
3 #print(df)
4 #create a figure and axis
5 fig, ax = plt.subplots()
```

```
~\Anaconda3\lib\site-packages\pandas\io\parsers.py in parser_f(filepath_or_buffer, sep, delimiter,
header, names, index_col, usecols, squeeze, prefix, mangle_dupe_cols, dtype, engine, converters, t
rue_values, false_values, skipinitialspace, skiprows, skipfooter, nrows, na_values,
keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format,
keep_date_col, date_parser, dayfirst, cache_dates, iterator, chunksize, compression, thousands,
decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment, encoding, dialect,
error_bad_lines, warn_bad_lines, delim_whitespace, low_memory, memory_map, float_precision)
683     )
684
```

```
--> 685         return _read(filepath_or_buffer, kwds)
686
687     parser_f.__name__ = name
```

```
~\Anaconda3\lib\site-packages\pandas\io\parsers.py in _read(filepath_or_buffer, kwds)
```

```
455
456     # Create the parser.
--> 457     parser = TextFileReader(fp_or_buf, **kwds)
458
459     if chunksize or iterator:
```

```
~\Anaconda3\lib\site-packages\pandas\io\parsers.py in __init__(self, f, engine, **kwds)
```

```
893         self.options["has_index_names"] = kwds["has_index_names"]
894
--> 895         self._make_engine(self.engine)
896
897     def close(self):
```

```
~\Anaconda3\lib\site-packages\pandas\io\parsers.py in _make_engine(self, engine)
```

```
1133     def _make_engine(self, engine="c"):
1134         if engine == "c":
--> 1135             self._engine = CParserWrapper(self.f, **self.options)
1136         else:
1137             if engine == "python":
```

```
~\Anaconda3\lib\site-packages\pandas\io\parsers.py in __init__(self, src, **kwds)
```

```
1915         kwds["usecols"] = self.usecols
1916
--> 1917         self._reader = parsers.TextReader(src, **kwds)
1918         self.unnamed_cols = self._reader.unnamed_cols
1919
```

```
pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader._cinit__()
```

```
pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader._setup_parser_source()
```

FileNotFoundError: [Errno 2] File b'' does not exist: b''

In [19]:

[https://nbconvert.readthedocs.io/en/latest/install.html#installing-tex.](https://nbconvert.readthedocs.io/en/latest/install.html#installing-tex)

```
File "<ipython-input-19-8ca5a9a9e093>", line 1
```

```
    https://nbconvert.readthedocs.io/en/latest/install.html#installing-tex.
```

```
    ^
```

```
SyntaxError: invalid syntax
```

```
In [ ]:
```