Setup for Notebooks

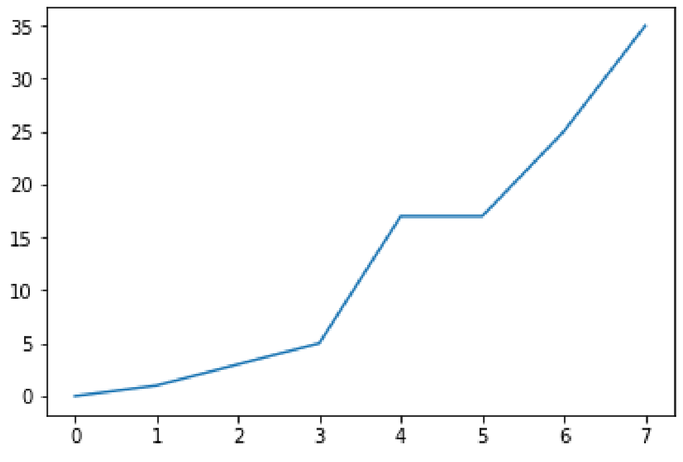
In [1]:
```
%matplotlib inline
```

In [2]:
```
# We will just use that a lot in general
import numpy as np
```

My First Plot

In [3]:
```
from matplotlib import pyplot as plt

plt.plot([0, 1, 3, 5, 17, 17, 25, 35])
# plt.show() # Not needed in notebooks
```

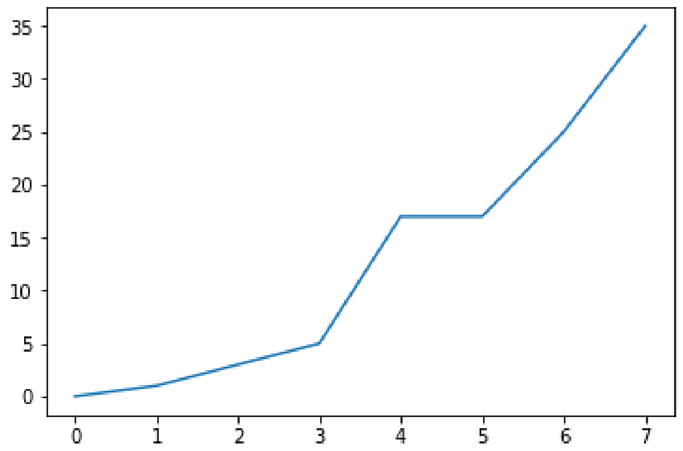
Out[3]: [<matplotlib.lines.Line2D at 0x7f4d03080ef0>]



Clean Version

In [4]:
```
from matplotlib import pyplot as plt

figure, axis = plt.subplots()
ys = [0, 1, 3, 5, 17, 17, 25, 35]
xs = np.arange(len(ys))
axis.plot(xs, ys)

# axis.show() # Not needed in notebooks
```

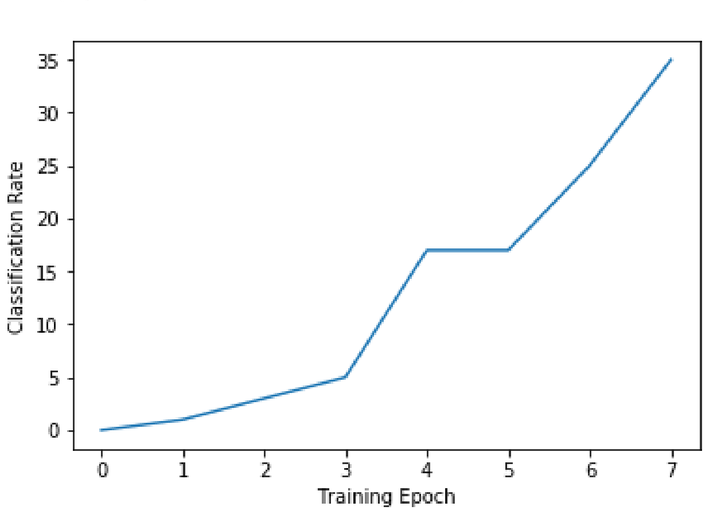
Out[4]: [<matplotlib.lines.Line2D at 0x7f4d05137128>]



Adding Labels

In [5]:
```
from matplotlib import pyplot as plt

figure, axis = plt.subplots()
ys = [0, 1, 3, 5, 17, 17, 25, 35]
xs = np.arange(len(ys))
axis.plot(xs, ys)
axis.set_xlabel("Training Epoch")
axis.set_ylabel("Classification Rate")
```

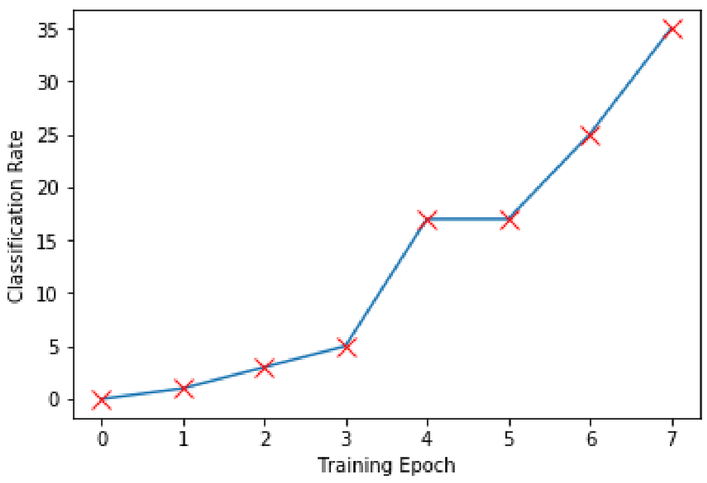
Out[5]: Text(0,0.5,'Classification Rate')



Adding Markers

In [6]:
```
from matplotlib import pyplot as plt

figure, axis = plt.subplots()
ys = [0, 1, 3, 5, 17, 17, 25, 35]
xs = np.arange(len(ys))
axis.plot(xs, ys,
          marker="x",
          markersize=10,
          markeredgecolor="red")
axis.set_xlabel("Training Epoch")
axis.set_ylabel("Classification Rate")
```

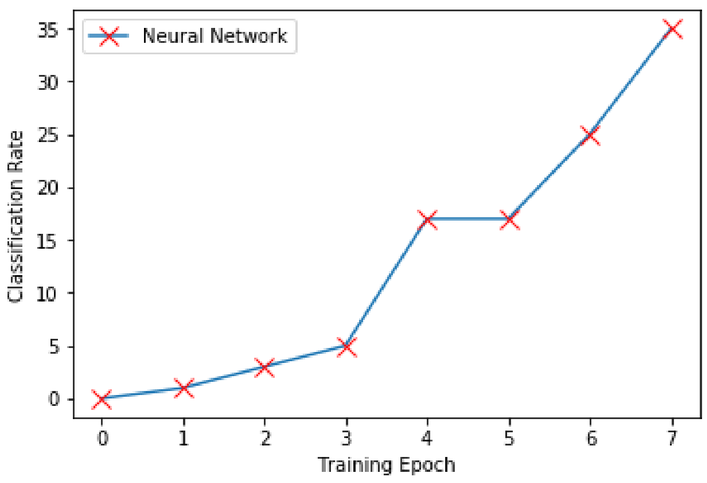
Out[6]: Text(0,0.5,'Classification Rate')



Adding a Legend

In [7]:
```
from matplotlib import pyplot as plt

figure, axis = plt.subplots()
ys = [0, 1, 3, 5, 17, 17, 25, 35]
xs = np.arange(len(ys))
axis.plot(xs, ys,
          marker="x",
          markersize=10,
          markeredgecolor="red",
          label="Neural Network")
axis.set_xlabel("Training Epoch")
axis.set_ylabel("Classification Rate")
axis.legend()
```

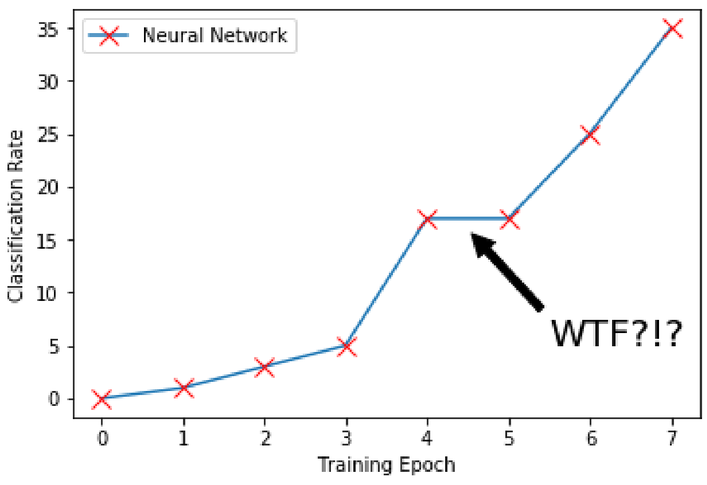
Out[7]: <matplotlib.legend.Legend at 0x7f4d02f0a470>



Adding an Annotation

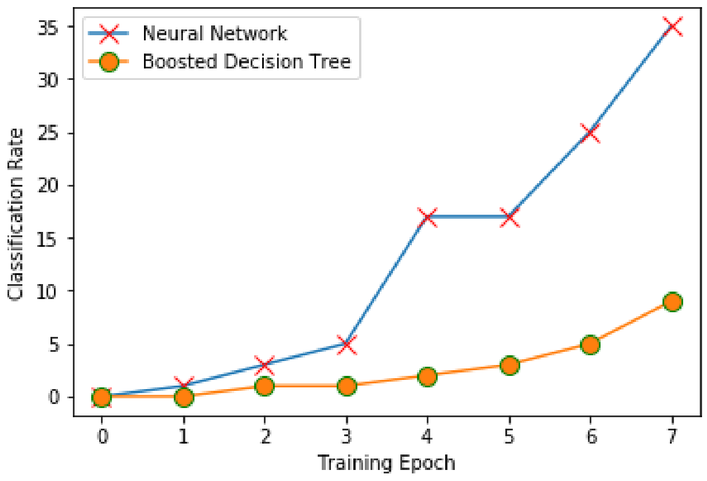
In [8]:
```
from matplotlib import pyplot as plt

figure, axis = plt.subplots()
ys = [0, 1, 3, 5, 17, 17, 25, 35]
xs = np.arange(len(ys))
axis.plot(xs, ys,
          marker="x",
          markersize=10,
          markeredgecolor="red",
          label="Neural Network")
axis.annotate('WTF?!?',
              xy=(4.5, 16),
              xytext=(5.5, 5),
              arrowprops=dict(facecolor='black',
                              shrink=0.05),
              fontsize=20 )
axis.set_xlabel("Training Epoch")
axis.set_ylabel("Classification Rate")
axis.legend()
```

Out[8]: <matplotlib.legend.Legend at 0x7f4d02f03278>



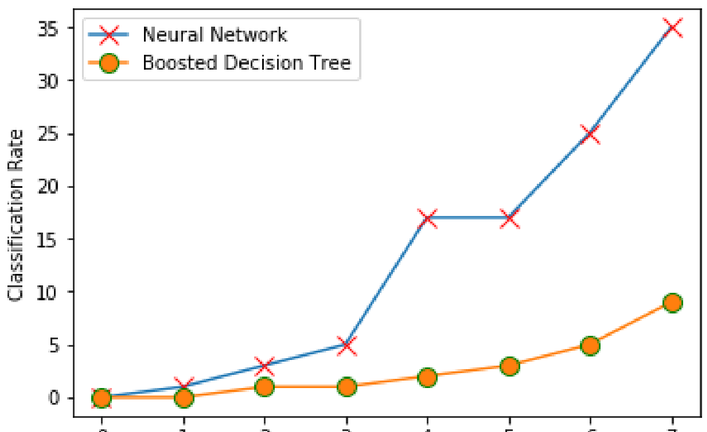
Multiple Lines

In [13]:
```
from matplotlib import pyplot as plt

figure, axis = plt.subplots()
ys = [0, 1, 3, 5, 17, 17, 25, 35]
xs = np.arange(len(ys))
axis.plot(xs, ys,
          marker="x",
          markersize=10,
          markeredgecolor="red",
          label="Neural Network")

axis.plot(xs, [0, 0, 1, 1, 2, 3, 5, 9],
          marker="o",
          markersize=10,
          markeredgecolor="green",
          label="Boosted Decision Tree")
axis.set_xlabel("Training Epoch")
axis.set_ylabel("Classification Rate")
axis.legend()
```



Saving Plots

In [14]:
```
from matplotlib import pyplot as plt

figure, axis = plt.subplots()
ys = [0, 1, 3, 5, 17, 17, 25, 35]
xs = np.arange(len(ys))
axis.plot(xs, ys,
          marker="x",
          markersize=10,
          markeredgecolor="red",
          label="Neural Network")

axis.plot(xs, [0, 0, 1, 1, 2, 3, 5, 9],
          marker="o",
          markersize=10,
          markeredgecolor="green",
          label="Boosted Decision Tree")
axis.set_xlabel("Training Epoch")
axis.set_ylabel("Classification Rate")
axis.legend()
figure.savefig("rate_over_epochs.png",
               transparent=True,
               bbox_inches="tight")
figure.savefig("rate_over_epochs.pdf",
               transparent=True,
               bbox_inches="tight")
```

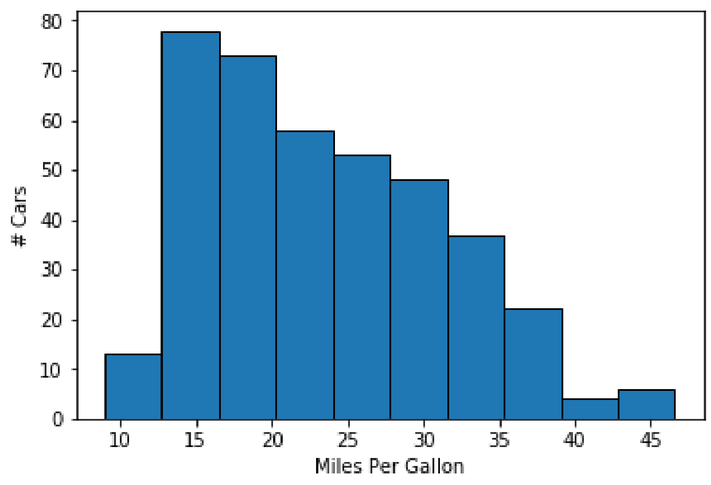


Other Plot Types

Histogram

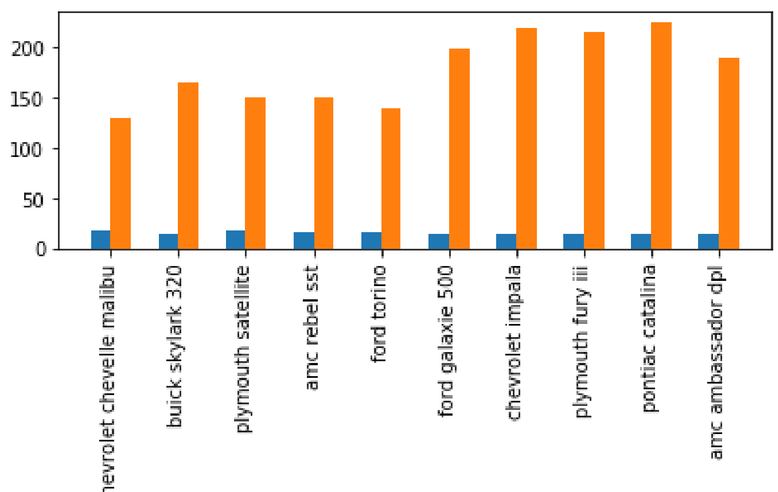
In [45]:
```
from bokeh.sampledata.autompg import autompg as car_data
```

In [86]:
```
mpgs = list(car_data['mpg'])
figure, axis = plt.subplots()
_ = axis.hist(mpgs, edgecolor="black")
axis.set_xlabel("Miles Per Gallon")
axis.set_ylabel("$\#$ Cars")
figure.savefig("histogram.pdf", transparent=True, bbox_inches="tight")
```



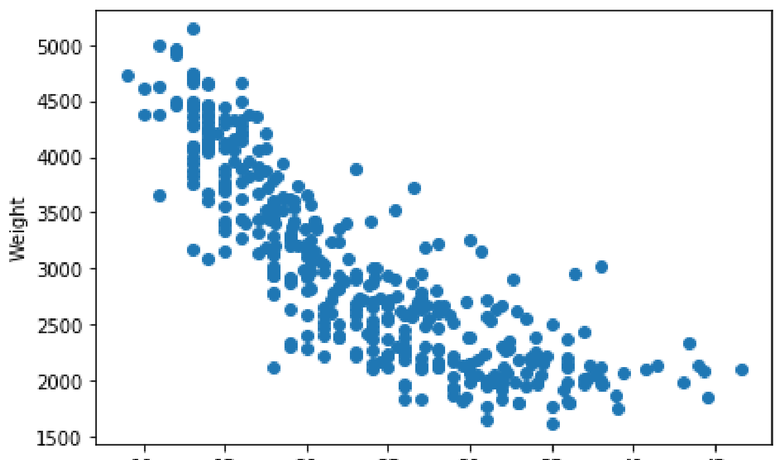
Bar Chart

In [73]:
```
mpgs = list(car_data['mpg'])[0:10]
hp = list(car_data['hp'])[0:10]
names = list(car_data['name'])[0:10]
indices = np.arange(len(mpgs))
figure, axis = plt.subplots()
bar_width = 0.3
axis.bar(indices, mpgs, bar_width)
axis.bar(indices + bar_width, hp, bar_width)
axis.set_xticks(indices + bar_width / 2)
axis.set_xticklabels(labels=names, rotation="vertical")
figure.tight_layout()
axis.get_xticklabels()
```



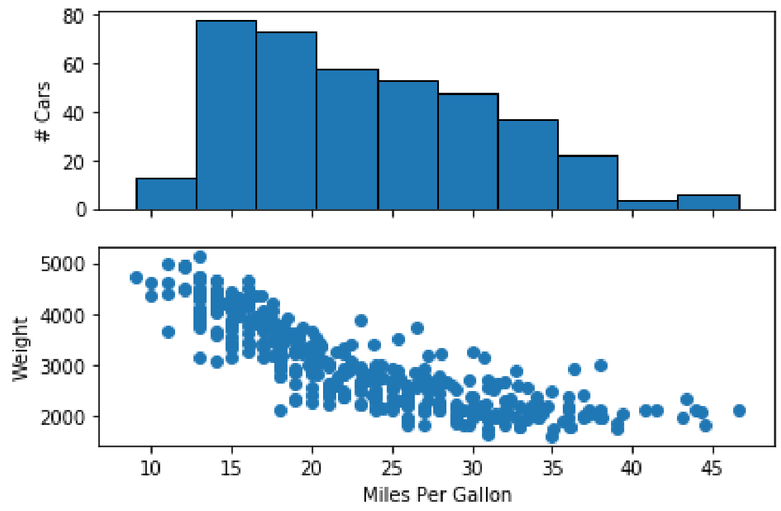
Scatter

In [77]:
```
mpgs = list(car_data['mpg'])
weights = list(car_data['weight'])
figure, axis = plt.subplots()
_ = axis.scatter(mpgs, weights)
axis.set_xlabel("MPG")
axis.set_ylabel("Weight")
figure.tight_layout()
```

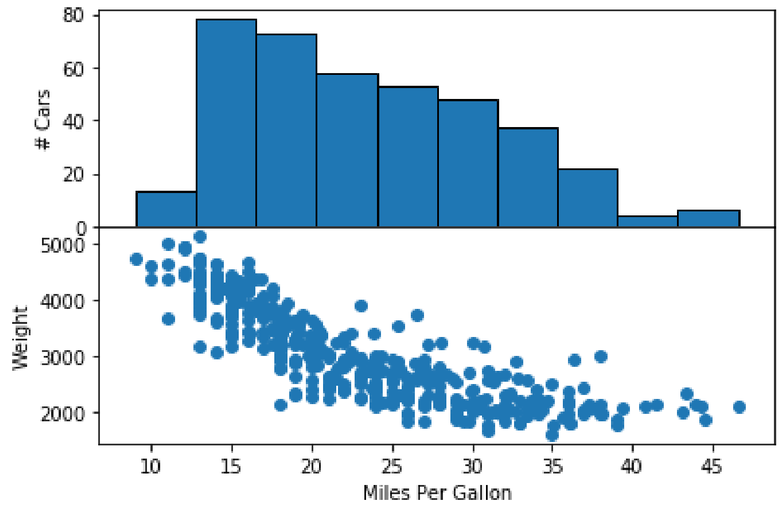


Subplots

In [87]:
```
figure, axes = plt.subplots((2),
                            sharex=True)
axes[0].hist(mpgs, edgecolor="black")
axes[0].set_ylabel("$\#$ Cars")

axes[1].scatter(mpgs, weights)
axes[1].set_xlabel("Miles Per Gallon")
axes[1].set_ylabel("Weight")
figure.tight_layout()
```



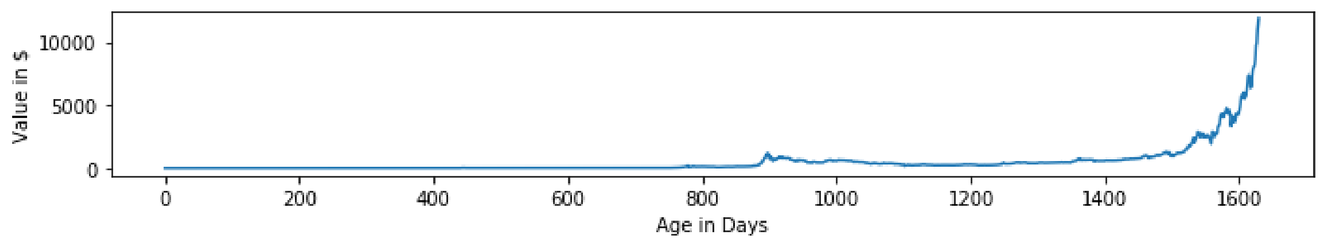
In [88]:
```
figure, axes = plt.subplots((2),
                            sharex=True)
axes[0].hist(mpgs, edgecolor="black")
axes[0].set_ylabel("$\#$ Cars")

axes[1].scatter(mpgs, weights)
axes[1].set_xlabel("Miles Per Gallon")
axes[1].set_ylabel("Weight")
figure.tight_layout()
figure.subplots_adjust(hspace=0)
```



Scaling Normal vs Log

In [ ]:
```
import csv
with open('market-price.csv', "r") as csvfile:
    csv_data = list(csv.reader(csvfile, delimiter=","))
timestamps = [x[0] for x in csv_data]
values = [float(x[1]) for x in csv_data]
```

In [128]:
```
figure, axis = plt.subplots(figsize=(10, 2))
axis.plot(values)
axis.set_xlabel("Age in Days")
axis.set_ylabel("Value in $")
figure.tight_layout()
```



In [130]:
```
figure, axis = plt.subplots(figsize=(10, 2))
axis.plot(values)
axis.set_yscale("symlog")
axis.set_xlabel("Age in Days")
axis.set_ylabel("$log(Value\ in\ \$)$")
figure.tight_layout()
```