Reading Data into Dictionaries

Consider CSV data of the form:

Alabama,10,20,30,...
Alaska,32,43,56,...
.
.
.
Wyoming,2,0,78,...

Write a function `data_from_file` that takes a filename and returns a dictionary `data` where each key is a state name and each value is a list of integers.

```python
>>> data["Minnesota"]
[47, 156, 107, 193, 121, 128]
>>> data["Iowa"]
[15, 36, 52, 57, 62, 45]
```

```python
import csv
def data_from_file(filename):
    with open(filename) as fin:
        return {state: [int(x) for x in nums]
                for state, *nums in csv.reader(fin)}
```
Plotting with Matplotlib

Suppose we want to plot test takers, by state and year. Here is a nice line plot with legend and axis labels.

And here is the code.

```python
import matplotlib.pyplot as plt

def plot1(data, states, years):
    for state in states:
        plt.plot(years, data[state], label=state)
    plt.legend(loc="best")
    plt.xlabel("Year")
    plt.ylabel("No. Students Taking CS AP Exam")
    plt.title("No. Students Taking CS AP Exam by Year")
    plt.savefig("out.png")
```

3D Scatter Plot of the Iris Dataset

This 3D scatter plot visualizes the iris dataset, with each point representing a flower and its characteristics. The x, y, and z axes represent different measurements of the flowers, allowing us to identify patterns and groupings within the dataset.

And here is the code.

```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# Load the iris dataset
iris = pd.read_csv('iris.csv')

# Extract features and target
X = iris.iloc[:, :-1]
Y = iris.iloc[:, -1]

# Create a 3D scatter plot
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')

# Plot the data points
ax.scatter(X.iloc[:, 0], X.iloc[:, 1], X.iloc[:, 2], c=Y, cmap='viridis', alpha=0.7)

# Add labels
ax.set_xlabel('Feature 1')
ax.set_ylabel('Feature 2')
ax.set_zlabel('Feature 3')

# Show the plot
plt.show()
```
A more aesthetically pleasing plot might fill the areas under the curves in. However, this requires us to choose our own colors using colormaps as well as create and place our own legend.

Here is the code.

```python
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.patches as mpatches
from itertools import count

def plot2(data, states, years):
    colors = plt.cm.Paired(np.linspace(0, 1, len(states)))
    patches = []
    for state, c in zip(states, colors):
        plt.fill_between(years, data[state], color=c, alpha=0.5)
        patches.append(mpatches.Patch(color=c, label=state))
    plt.legend(handles=patches, loc="upper left")
    plt.xlabel("Year")
    plt.ylabel("No. Students Taking CS AP Exam")
    plt.title("No. Students Taking CS AP Exam by Year")
    plt.savefig("out2.png")
```

Spring Semester 2015 3 CS 135: Diving into the Deluge of Data
Suppose we wanted subplots for each of the states.

```python
12 def plot3(data, states, years):
    colors = plt.cm.Set1(np.linspace(0,1,len(states)))
    for i, state, c in zip(count(), states, colors):
        ax = plt.subplot2grid((len(states),1),(i,0))
        ax.fill_between(years, data[state], color=c)
        ax.set_ylabel("Count")
        for tick in ax.yaxis.get_major_ticks():
            tick.label.set_fontsize(8)
    plt.tight_layout()
    plt.xlabel("Year")
    plt.savefig("out3.png")
```

We can use the `subplot2grid` command to help out.