Looking at data with Python: Matplotlib and Pandas

I like python. R and Excel have their uses too for data analysis, but I just keep coming back to Python.

One of the first things I want to do once I've finally wrangled a dataset out of various APIs, websites and pieces of paper, is to have a good look at what's in it. Two python libraries are useful here: Pandas and Matplotlib.

- Pandas is Wes McKinney's library for R-style dataframe (data in rows and columns) manipulation, summary and analysis.
- Matplotlib is John D Hunter's library for Matlab-style plots of data.

Before you start, you'll need to type "pip install pandas" and "pip install matplotlib" in the terminal window. It's also convention to load the libraries into your code with these two lines:

```
import pandas as pd  import matplotlib.pyplot as plt
```

Some things in Pandas (like reading in datafiles) are wonderfully easy; others take a little longer to learn. I'll meander through a few of them here. I'll be using Nepal medical shipments data as an example.

Reading in data

Pandas makes this easy. Reading in a CSV file is as simple as:

```
df = pd.read_csv(csvfilename) # Comma-separated file  df = pd.read_csv(cs
vfilename, sep='\t') # Tab-separated file
```

There's also *pd.read_json, pd.read_html, pd.read_sas, pd.read_stata* and *pd.read_sql_table* to read in other data formats. Be careful with read_html though: it only reads in html tables, and you'll need lxml, beautifulsoup or suchlike if you want to read tables straight from a webpage.

First-look at the dataframe

I like to know what I'm dealing with before starting analysis. I usually use Tableau or R for this, but that's not always possible, and Pandas is a good alternative.

```
df.columns # List all the column headings  df.head(4) # The first 4 rows of data, same as df.head(n=4)  df[['column1','column2', 'column3']].head(10) # Just some of the columns  df.describe() # Basic statistics for every numerical column
```

That tells you what your columns are, what your first few rows look like (df.tail(4) will give the last rows) and some basic statistics for numerical columns, but you're probably more curious than that.

```
df['columnname1'].value_counts()
```

Value_counts will tell you what's in a single column. If you want to know what's in a pair or combination of columns, you'll need to start using pivot tables or group_by.

You might know pivot tables from Excel. They're ways of creating a new datatable whose rows, columns and content are defined by you. This function, for example, gives you a new table whose rows are column x values, columns are column y values, and contents are the number of rows that contained those combinations of x and y values.

```
x_by_y = df.pivot_table(index='columnx',columns='columny', values='columnz', aggfunc='count', fill_value=0)
```

Column z gets involved here because if you don't nominate a column for the values, Pandas will return an array with the counts for every combination of columns. I've included fill_value=0 because I'm counting, and Pandas would otherwise include NaN (not a number) in its counts.

x_by_y is a data frame. You can plot this, for example:

```
x_by_y.head(10).plot(kind='bar', stacked=True)  plt.show()
```

You're now using Matplotlib. And that was quite a complex plot: a stacked bar chart, with a legend. Note that .plot creates a plot object: if you want to *see* your plot, you need to type "plt.show()". This will put up a plot window and stop your code until you close the window again.

Basic data manipulation

I've had a look at the dataset, got some ideas for more things to look at in it, and some of them need calculations. Pandas handles this too. More soon. Meanwhile, here's some stuff I did with the Nepal dataset.

```
pivot1 = df.pivot_table(   index='Material Hierarchy Family',   columns='Final Recipient Name',   values='Dollar Value',   aggfunc='count').plot(kind='barh', stacked=True)    recipientsize = df.groupby('Final Recipient Name').size()    pivot2 = df.pivot_table(   columns='Material Hierarchy Family',   index='Final Recipient Name',   values='Dollar Value',   aggfunc='count')    pivot2.plot(kind='bar', stacked=True, legend=False)  plt.show()
```

Links

- Pandas:
  - http://bconnelly.net/2013/10/summarizing-data-in-python-with-pandas/
  - http://pbpython.com/excel-pandas-comp.html