NumPy / SciPy / Pandas Cheat Sheet

python™

NumPy / SciPy

`arr = array([])`	Create numpy array.
`arr.shape`	Shape of an array.
`convolve(a,b)`	Linear convolution of two sequences.
`arr.reshape()`	Reshape array.
`sum(arr)`	Sum all elements of array.
`mean(arr)`	Compute mean of array.
`std(arr)`	Compute standard deviation of array.
`dot(arr1,arr2)`	Compute inner product of two arrays.
`vectorize()`	Turn a scalar function into one which accepts & returns vectors.

Pandas

Create Structures

`s = Series (data, index)`	Create a Series.
`df = DataFrame (data, index, columns)`	Create a Dataframe.
`p = Panel (data, items, major_axis, minor_axis)`	Create a Panel.

DataFrame commands

`df[col]`	Select column.
`df.iloc[label]`	Select row by label.
`df.index`	Return DataFrame index.
`df.drop()`	Delete given row or column. Pass axis=1 for columns.
`df1 = df1.reindex_like(df1,df2)`	Reindex df1 with index of df2.
`df.reset_index()`	Reset index, putting old index in column named index.
`df.reindex()`	Change DataFrame index, new indecies set to NaN.
`df.head(n)`	Show first n rows.
`df.tail(n)`	Show last n rows.
`df.sort()`	Sort index.
`df.sort(axis=1)`	Sort columns.
`df.pivot(index,column,values)`	Pivot DataFrame, using new conditions.
`df.T`	Transpose DataFrame.
`df.stack()`	Change lowest level of column labels into innermost row index.
`df.unstack()`	Change innermost row index into lowest level of column labels.

`df.applymap()`	Apply function to every element in DataFrame.
`df.apply()`	Apply function along a given axis.
`df.dropna()`	Drops rows where any data is missing.
`df.count()`	Returns Series of row counts for every column.
`df.min()`	Return minimum of every column.
`df.max()`	Return maximum of every column.
`df.describe()`	Generate various summary statistics for every column.
`concat()`	Merge DataFrame or Series objects.

Groupby

`groupby()`	Split DataFrame by columns. Creates a GroupBy object (gb).
`gb.agg()`	Apply function (single or list) to a GroupBy object.
`gb.transform()`	Applies function and returns object with same index as one being grouped.
`gb.filter()`	Filter GroupBy object by a given function.
`gb.groups`	Return dict whose keys are the unique groups, and values are axis labels belonging to each group.

I/O

`df.to_csv('foo.csv')`	Save to CSV.
`read_csv('foo.csv')`	Read CSV into DataFrame.
`to_excel('foo.xlsx', sheet_name)`	Save to Excel.
`read_excel('foo.xlsx','sheet1', index_col = None, na_values = ['NA'])`	Read exel into DataFrame.

Pandas Time Series

Any Structure with a datetime index

`date_range(start, end, freq)`	Create a time series index.

Freq has many options including:

B	Business Day
D	Calender day
W	Weekly
M	Monthly
Q	Quarterly
A	Annual
H	Hourly

`ts.resample()`	Resample data with new frequency.

`ts.ix[start:end]`	Return data for nearest time interval.
`ts[]`	Return data for specific time.
`ts.between_time()`	Return data between specific interval.
`to_pydatetime()`	Convert Pandas DatetimeIndex to datetime.datetime object.
`to_datetime()`	Conver a list of date-like objects (strings, epochs, etc.) to a DatetimeIndex.

Plotting

Matplotlib is an extremely powerful module.
See www.matplotlib.org for complete documentation.

`plot()`	Plot data or plot a function against a range.
`xlabel()`	Label the x-axis.
`ylabel()`	Label the y-axis.
`title()`	Title the graph.
`subplot(n,x,y)`	Create multiple plots; n- number of plots, x - number horizontally displayed, y- number vertically displayed.
`xticks([],[])`	Set tick values for x-axis. First array for values, second for labels.
`yticks([],[])`	Set tick values for y-axis. First array for values, second for labels.
`ax=gca()`	Select current axis.
`ax.spines[].set_color()`	Change axis color, none to remove.
`ax.spines[].set_position()`	Change axis position. Can change coordinate space.
`legend(loc=' ')`	Create legend. Set to 'best' for auto placement.
`savefig('foo.png')`	Save plot.

Quandl

The Quandl package enables Quandl API access from within Python, which makes acquiring and manipulating numerical data as quick and easy as possible.

In your first Quandl function call you should specifiy your authtoken (found on Quandl's website after signing up) to avoid certain API call limits.

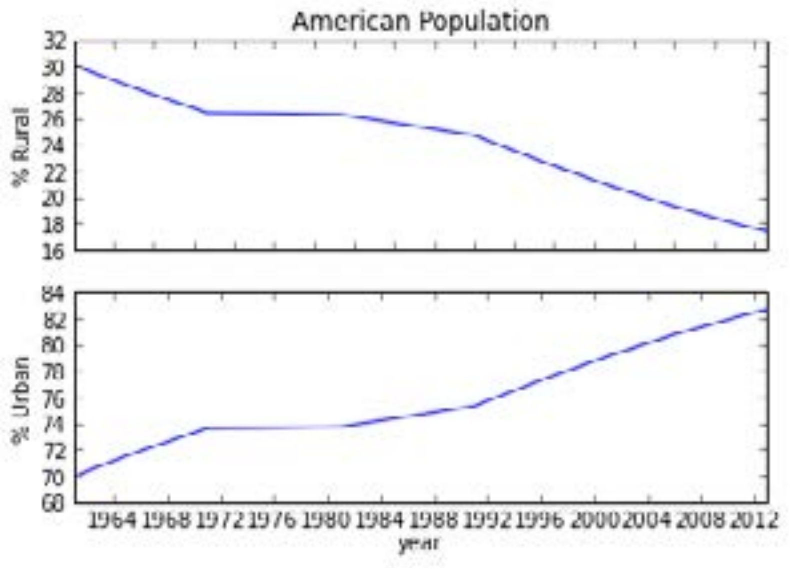
See www.quandl.com/help/packages/python for more.

Quandl is a search engine for numerical data, allowing easy access to financial, social, and demographic data from hundreds of sources.

`authtoken = ‘YOURTOKENHERE’`	Add the following to any function call.
`get(‘QUANDL/CODE’)`	Download Quandl data for a certain Quandl code as a Dataframe.
`search(‘searchterm’)`	Search Quandl. Outputs first 4 results.
`push(data, code, name)`	Upload a Pandas DataFrame (with a time series index) to Quandl. Code must be all capital alphanumeric.

Plotting Example

```
import Quandl as q
import matplotlib.pyplot as plt
rural =  q.get(‘WORLDBANK/USA_SP_RUR_TOTL_ZS’)
urban = q.get(‘WORLDBANK/USA_SP_URB_TOTL_IN_ZS’)

plt.subplot(2, 1, 1)
plt.plot(rural.index,rural)

plt.xticks(rural.index[0::3],[])

plt.title(‘American Population’)

plt.ylabel(‘% Rural’)
plt.subplot(2, 1, 2)
plt.plot(urban.index,urban)

plt.xlabel(‘year’)
plt.ylabel(‘% Urban’)
plt.show()
```


American Population
% Rural
% Urban
year