Python Data Analysis Guide

Ayman Ali

Contents

List of Figures

List of Tables

1 Dataframe Manipulation with Pandas

1. Create dataframes based on value in column:

```
new_df = df.loc[df['Column'] == value]
```

2. Drop columns:

```
drop_columns = ['Column 1', 'Column 2', 'Column 3']
new_df = df.drop(drop_columns, axis = 1)
```

3. Select rows by index (works with slicing as well):

```
rows = df.iloc[index]
```

4. Select row and column by index (can use slicing on both):

```
new_df = df.iloc[index_of_row, index_of_column]
```

5. Replace values:

```
new_df = df.replace(to_replace = value_to_replace,
                    value = value_to_replace_with)
```

2 Graphing with Matplotlib

2.1 General Setup

1. For the imports:

```
import matplotlib.pyplot as plt
# if you want to see what styles you can pick from
plt.style.available
# personal preference
plt.style.use('bmh')
```

2. Set up a graph that you're about to plot with:

```
'''
If both columns and rows are greater than 1, then ax will be a (row, column)
dimensional array. Otherwise, it'll be a one-dimensional array, with either
(rows) or (columns) depending on which one is greater than 1. You can access
each axis with standard indexing (i.e. ax[1][2] will give you the axes of
the graph in row 2 and column 3)
'''
fig, ax = plt.subplots(nrows = num_rows, ncols = num_cols,
                       sharex = boolean, sharey = boolean, dpi = integer,
                       figsize = (width, height))
```

3. Titles:

```
# Individual axis titles with
ax.set_title('string')
# For a general graph title, use text boxes (next section)
```

4. Adding text boxes to the figure:

```
# Find all kwargs at the following URL:
# https://matplotlib.org/api/text_api.html#matplotlib.text.Text
# Some of the more useful kwargs that I use are:
# verticalalignment ('center', 'top', 'bottom', 'baseline') and
# horizontalalignment ('center', 'right', 'left')
fig.text(double_xloc, double_yloc, 'text', **kwargs)
```

5. Finishing the plot:

```
# If you had a label anywhere then make sure to call it
plt.legend(fontsize = integer or None)
plt.savefig(file_path)
plt.close()
plt.clf()
```

2.2 Plot Types

1. Standard plot:

```
# fill in
```

2. Scatter:

```
# fill in
```

3. Bar graphs:

```
spacing = double_val
plt.bar(x_values +/-/nothing spacing, y_vals, width = spacing,
        align = 'center', color = any matplotlib color)
```

4. Errorbars:

```
axis.errorbar(x_vals, y_vals, xerr = error_x, yerr = error_y,
              markersize = integer,
              lolims / uplims / xlolims / xuplims = optional booleans)
```

2.3 Misc.

1. Custom tick marks and steps:

```
plt.xticks(np.arange(start = start_val, stop = stop_val, step = step_val))
plt.yticks(np.arange(start = start_val, stop = stop_val, step = step_val))
```