ESCI 386 – Scientific Programming, Analysis and Visualization with Python

Lesson 12 - Multiple-Panel Plots
Multi-panel Plots Using `subplot()`

- To create multi panel plots we use `subplot(rcp)`, which is either a pyplot function or a figure method.

- `subplot(rcp)` creates an axes object on the figure for which it is called, or on the current figure if called as a pyplot function.

- The arguments are:
  - $r$ the number of rows for the subplots
  - $c$ the number of columns for the subplots
  - $p$ the subplot number
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(0, 100.0)
y1 = np.cos(2*np.pi*x/50.0)
y2 = np.sin(2*np.pi*x/50.0)
ax1 = plt.subplot(211) # creates first axis
    ax1.plot(x, y1, 'k-')
ax2 = plt.subplot(212) # creates second axis
    ax2.plot(x, y2, 'k--')
plt.show()
subplot() Result
Adjusting Subplot Alignment

- The `subplots_adjust()` `pyplot` function or figure method is used for controlling the spacing between and around subplots.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>left</td>
<td>The position of the left side of the subplots in axes coordinates (0 to 1.0)</td>
</tr>
<tr>
<td>bottom</td>
<td>The position of the bottom of the subplots in axes coordinates (0 to 1.0)</td>
</tr>
<tr>
<td>right</td>
<td>The position of the right side of the subplots in axes coordinates (0 to 1.0)</td>
</tr>
<tr>
<td>top</td>
<td>The position of the top of the subplots in axes coordinates (0 to 1.0)</td>
</tr>
<tr>
<td>wspace</td>
<td>The spacing between columns in points</td>
</tr>
<tr>
<td>hspace</td>
<td>The spacing between rows in points</td>
</tr>
</tbody>
</table>
Adjusting Subplot Alignment

• Subplots can also be adjusted interactively once the plot is created.
• In the plot window, click on the icon shown.
Multi-panel Plots Using `subplots()`

- An alternate method for creating subplots is with the `pyplot.subplots()` function (note the plural here!).

- This function creates the figure and all axes at one time.
Using `subplots()`

- The usage is
  
  ```python
  fig, ax = plt.subplots(r,c)
  ```
  
  - `r` is the number of rows
  - `c` is the number of columns

- `fig` is a reference to the figure that is created.

- `ax` is an $r \times c$ NumPy array containing all the individual axes for the subplots.
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(0,10.0, 0.1)
fig, ax = plt.subplots(2,2)
ax[0,0].plot(x, x)
ax[0,1].plot(x, x**2)
ax[1,0].plot(x, np.sqrt(x))
ax[1,1].plot(x, 1/x)
plt.show()
subplots() Result
Sharing Axis Labels

- Using the keywords `sharex` or `sharey` the x or y axes can be shared among subplots.
- For example, using `sharex = True` in the previous example yields the plot shown.
Iterating Over Subplots

• Since subplots() returns an array containing all the axes, we can iterate over it if we want to.

```python
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(0, 10.0, 0.1)
fig, ax = plt.subplots(2, 2)
for i, a in enumerate(ax.flatten()):
    a.plot(x, x**i)
plt.show()
```

flatten() creates 1-D array from multi-dimensional array

File: subplots-iterate.py