

mpl_styles

March 23, 2016

1 What styles have you got?

```
In [2]: %matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
```

```
/usr/local/lib/python3.4/dist-packages/matplotlib/_init_.py:872: UserWarning: axes.color_cycle is deprecated and
warnings.warn(self.msg_depr % (key, alt_key))
```

Next we want to check the various styles. We will look at creating your own style later on. The built in ones are good starting points. To print the available ones run

```
In [3]: print(plt.style.available)
```

```
['seaborn-paper', 'seaborn-dark-palette', 'fivethirtynine', 'seaborn-whitegrid', 'seaborn-white', 'seaborn-dark
```

You can also set things like we did earlier today, with the rcParams.

```
In [18]: plt.rcParams
```

```
Out[18]: RcParams({'agg.path.chunksize': 0,
                  'animation.avconv_args': [],
                  'animation.avconv_path': 'avconv',
                  'animation.bitrate': -1,
                  'animation.codec': 'mpeg4',
                  'animation.convert_args': [],
                  'animation.convert_path': 'convert',
                  'animation.ffmpeg_args': [],
                  'animation.ffmpeg_path': 'ffmpeg',
                  'animation.frame_format': 'png',
                  'animation.html': 'none',
                  'animation.mencoder_args': [],
                  'animation.mencoder_path': 'mencoder',
                  'animation.writer': 'ffmpeg',
                  'axes.axisbelow': False,
                  'axes.edgecolor': 'k',
                  'axes.facecolor': 'w',
                  'axes.formatter.limits': [-7, 7],
                  'axes.formatter.use_locale': False,
                  'axes.formatter.use_mathtext': False,
                  'axes.formatter.useoffset': True,
                  'axes.grid': False,
                  'axes.grid.axis': 'both',
                  'axes.grid.which': 'major',
```

```

'axes.hold': True,
'axes.labelcolor': 'k',
'axes.labelpad': 5.0,
'axes.labelsize': 'medium',
'axes.labelweight': 'normal',
'axes.linewidth': 1.0,
'axes.prop_cycle': cycler('color', ['b', 'g', 'r', 'c', 'm', 'y', 'k']),
'axes.spines.bottom': True,
'axes.spines.left': True,
'axes.spines.right': True,
'axes.spines.top': True,
'axes.titlesize': 'large',
'axes.titleweight': 'normal',
'axes.unicode_minus': True,
'axes.xmargin': 0.0,
'axes.ymargin': 0.0,
'axes3d.grid': True,
'backend': 'module://ipykernel.pylab.backend_inline',
'backend.qt4': 'PyQt4',
'backend.qt5': 'PyQt5',
'backend_fallback': True,
'boxplot.bootstrap': None,
'boxplot.boxprops.color': 'b',
'boxplot.boxprops.linestyle': '-',
'boxplot.boxprops.linewidth': 1.0,
'boxplot.capprops.color': 'k',
'boxplot.capprops.linestyle': '-',
'boxplot.capprops.linewidth': 1.0,
'boxplot.flierprops.color': 'b',
'boxplot.flierprops.linestyle': 'none',
'boxplot.flierprops.linewidth': 1.0,
'boxplot.flierprops.marker': '+',
'boxplot.flierprops.markeredgecolor': 'k',
'boxplot.flierprops.markerfacecolor': 'b',
'boxplot.flierprops.markersize': 6.0,
'boxplot.meanline': False,
'boxplot.meanprops.color': 'r',
'boxplot.meanprops.linestyle': '-',
'boxplot.meanprops.linewidth': 1.0,
'boxplot.medianprops.color': 'r',
'boxplot.medianprops.linestyle': '-',
'boxplot.medianprops.linewidth': 1.0,
'boxplot.notch': False,
'boxplot.patchartist': False,
'boxplot.showbox': True,
'boxplot.showcaps': True,
'boxplot.showfliers': True,
'boxplot.showmeans': False,
'boxplot.vertical': True,
'boxplot.whiskerprops.color': 'b',
'boxplot.whiskerprops.linestyle': '--',
'boxplot.whiskerprops.linewidth': 1.0,
'boxplot.whiskers': 1.5,
'contour.corner_mask': True,

```

```

'contour.negative_linestyle': 'dashed',
'datapath': '/usr/local/lib/python3.4/dist-packages/matplotlib/mpl-data',
'docstring.hardcopy': False,
'errorbar.capsize': 3.0,
'examples.directory': '',
'figure.autolayout': False,
'figure.dpi': 80.0,
'figure.edgecolor': (1, 1, 1, 0),
'figure.facecolor': (1, 1, 1, 0),
'figure.figsize': [6.0, 4.0],
'figure.frameon': True,
'figure.max_open_warning': 20,
'figure.subplot.bottom': 0.125,
'figure.subplot.hspace': 0.2,
'figure.subplot.left': 0.125,
'figure.subplot.right': 0.9,
'figure.subplot.top': 0.9,
'figure.subplot.wspace': 0.2,
'figure.titlesize': 'medium',
'figure.titleweight': 'normal',
'font.cursive': ['Apple Chancery',
                 'Textile',
                 'Zapf Chancery',
                 'Sand',
                 'Script MT',
                 'Felipa',
                 'cursive'],
'font.family': ['sans-serif'],
'font.fantasy': ['Comic Sans MS',
                 'Chicago',
                 'Charcoal',
                 'ImpactWestern',
                 'Humor Sans',
                 'fantasy'],
'font.monospace': ['Bitstream Vera Sans Mono',
                   'DejaVu Sans Mono',
                   'Andale Mono',
                   'Nimbus Mono L',
                   'Courier New',
                   'Courier',
                   'Fixed',
                   'Terminal',
                   'monospace'],
'font.sans-serif': ['Bitstream Vera Sans',
                   'DejaVu Sans',
                   'Lucida Grande',
                   'Verdana',
                   'Geneva',
                   'Lucid',
                   'Arial',
                   'Helvetica',
                   'Avant Garde',
                   'sans-serif'],
'font.serif': ['Bitstream Vera Serif',

```

```

        'DejaVu Serif',
        'New Century Schoolbook',
        'Century Schoolbook L',
        'Utopia',
        'ITC Bookman',
        'Bookman',
        'Nimbus Roman No9 L',
        'Times New Roman',
        'Times',
        'Palatino',
        'Charter',
        'serif'],
'font.size': 10.0,
'font.stretch': 'normal',
'font.style': 'normal',
'font.variant': 'normal',
'font.weight': 'normal',
'grid.alpha': 1.0,
'grid.color': 'k',
'grid.linestyle': ':',
'grid.linewidth': 0.5,
'image.aspect': 'equal',
'image.cmap': 'jet',
'image.composite_image': True,
'image.interpolation': 'bilinear',
'image.lut': 256,
'image.origin': 'upper',
'image.resample': False,
'interactive': True,
'keymap.all_axes': ['a'],
'keymap.back': ['left', 'c', 'backspace'],
'keymap.forward': ['right', 'v'],
'keymap.fullscreen': ['f', 'ctrl+f'],
'keymap.grid': ['g'],
'keymap.home': ['h', 'r', 'home'],
'keymap.pan': ['p'],
'keymap.quit': ['ctrl+w', 'cmd+w'],
'keymap.save': ['s', 'ctrl+s'],
'keymap.xscale': ['k', 'L'],
'keymap.yscale': ['l'],
'keymap.zoom': ['o'],
'legend.borderaxespad': 0.5,
'legend.borderpad': 0.4,
'legend.columnspacing': 2.0,
'legend.edgecolor': 'inherit',
'legend.facecolor': 'inherit',
'legend.fancybox': False,
'legend.fontsize': 'large',
'legend.framealpha': None,
'legend.frameon': True,
'legend.handleheight': 0.7,
'legend.handlelength': 2.0,
'legend.handletextpad': 0.8,
'legend.isaxes': True,

```

```

'legend.labelsplacement': 0.5,
'legend.loc': 'upper right',
'legend.markerscale': 1.0,
'legend.numpoints': 2,
'legend.scatterpoints': 3,
'legend.shadow': False,
'lines.antialiased': True,
'lines.color': 'b',
'lines.dash_capstyle': 'butt',
'lines.dash_joinstyle': 'round',
'lines.linestyle': '-',
'lines.linewidth': 3.0,
'lines.marker': 'None',
'lines.markeredgewidth': 0.5,
'lines.markersize': 6.0,
'lines.solid_capstyle': 'projecting',
'lines.solid_joinstyle': 'round',
'markers.fillstyle': 'full',
'mathtext.bf': 'serif:bold',
'mathtext.cal': 'cursive',
'mathtext.default': 'it',
'mathtext.fallback_to_cm': True,
'mathtext.fontset': 'cm',
'mathtext.it': 'serif:italic',
'mathtext.rm': 'serif',
'mathtext.sf': 'sans\\-serif',
'mathtext.tt': 'monospace',
'nbgg.transparent': True,
'patch.antialiased': True,
'patch.edgecolor': 'k',
'patch.facecolor': 'b',
'patch.linewidth': 1.0,
'path.effects': [],
'path.simplify': True,
'path.simplify_threshold': 0.1111111111111111,
'path.sketch': None,
'path.snap': True,
'pdf.compression': 6,
'pdf.fonttype': 3,
'pdf.inheritcolor': False,
'pdf.use14corefonts': False,
'pgf.debug': False,
'pgf.preamble': [],
'pgf.rcfonts': True,
'pgf.texsystem': 'xelatex',
'plugins.directory': '.matplotlib_plugins',
'polaraxes.grid': True,
'ps.distiller.res': 6000,
'ps.fonttype': 3,
'ps.papersize': 'letter',
'ps.useafm': False,
'ps.usedistiller': False,
'savefig.bbox': None,
'savefig.directory': '~',

```

```

'savefig.dpi': 72.0,
'savefig.edgecolor': 'w',
'savefig.facecolor': 'w',
'savefig.format': 'png',
'savefig.frameon': True,
'savefig.jpeg_quality': 95,
'savefig.orientation': 'portrait',
'savefig.pad_inches': 0.1,
'savefig.transparent': False,
'svg.fonttype': 'path',
'svg.image_inline': True,
'svg.image_noscale': False,
'text.antialiased': True,
'text.color': 'k',
'text.dvipnghack': None,
'text.hinting': 'auto',
'text.hinting_factor': 8,
'text.latex.preamble': [],
'text.latex.preview': False,
'text.latex.unicode': False,
'text.usetex': False,
'timezone': 'UTC',
'tk.window_focus': False,
'toolbar': 'toolbar2',
'verbose.fileo': 'sys.stdout',
'verbose.level': 'silent',
'webagg.open_in_browser': True,
'webagg.port': 8988,
'webagg.port_retries': 50,
'xtick.color': 'k',
'xtick.direction': 'in',
'xtick.labelsize': 'medium',
'xtick.major.pad': 4.0,
'xtick.major.size': 4.0,
'xtick.major.width': 0.5,
'xtick.minor.pad': 4.0,
'xtick.minor.size': 2.0,
'xtick.minor.visible': False,
'xtick.minor.width': 0.5,
'ytick.color': 'k',
'ytick.direction': 'in',
'ytick.labelsize': 'medium',
'ytick.major.pad': 4.0,
'ytick.major.size': 4.0,
'ytick.major.width': 0.5,
'ytick.minor.pad': 4.0,
'ytick.minor.size': 2.0,
'ytick.minor.visible': False,
'ytick.minor.width': 0.5})

```

```

In [4]: plt.rcParams['figure.facecolor'] = 'w'
        plt.rcParams['font.size'] = 15
        plt.rcParams['lines.linewidth'] = 3

```

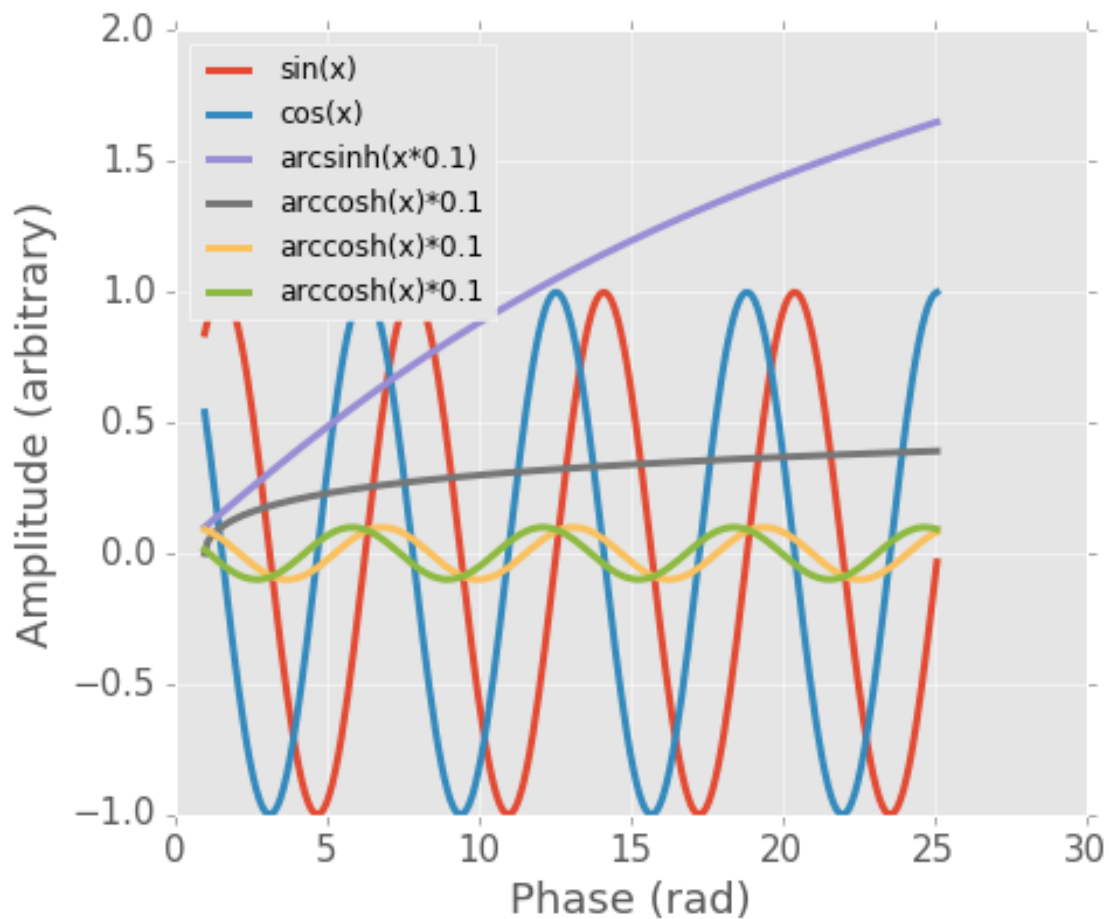
Now I just create a function to plot some stuff just to see how the styles handle colors, lines axes,

background etc.

```
In [5]: def draw_it(style):
plt.style.reload_library()
with plt.style.context(style):
    x = np.arange(1,8*np.pi,0.1)
    plt.figure(figsize=(6.5,5.5))
    plt.plot(x,np.sin(x), label='sin(x)')
    plt.plot(x,np.cos(x), label='cos(x)')
    plt.plot(x,np.arcsinh(x*0.1), label='arcsinh(x*0.1)')
    plt.plot(x,np.arccosh(x)*0.1, label='arccosh(x)*0.1')
    plt.plot(x,np.sin(x+1.)*0.1, label='arccosh(x)*0.1')
    plt.plot(x,np.sin(x+2.)*0.1, label='arccosh(x)*0.1')
    plt.legend(loc=2, fontsize=12)
    plt.xlabel('Phase (rad)')
    plt.ylabel('Amplitude (arbitrary)')
    plt.tight_layout();
```

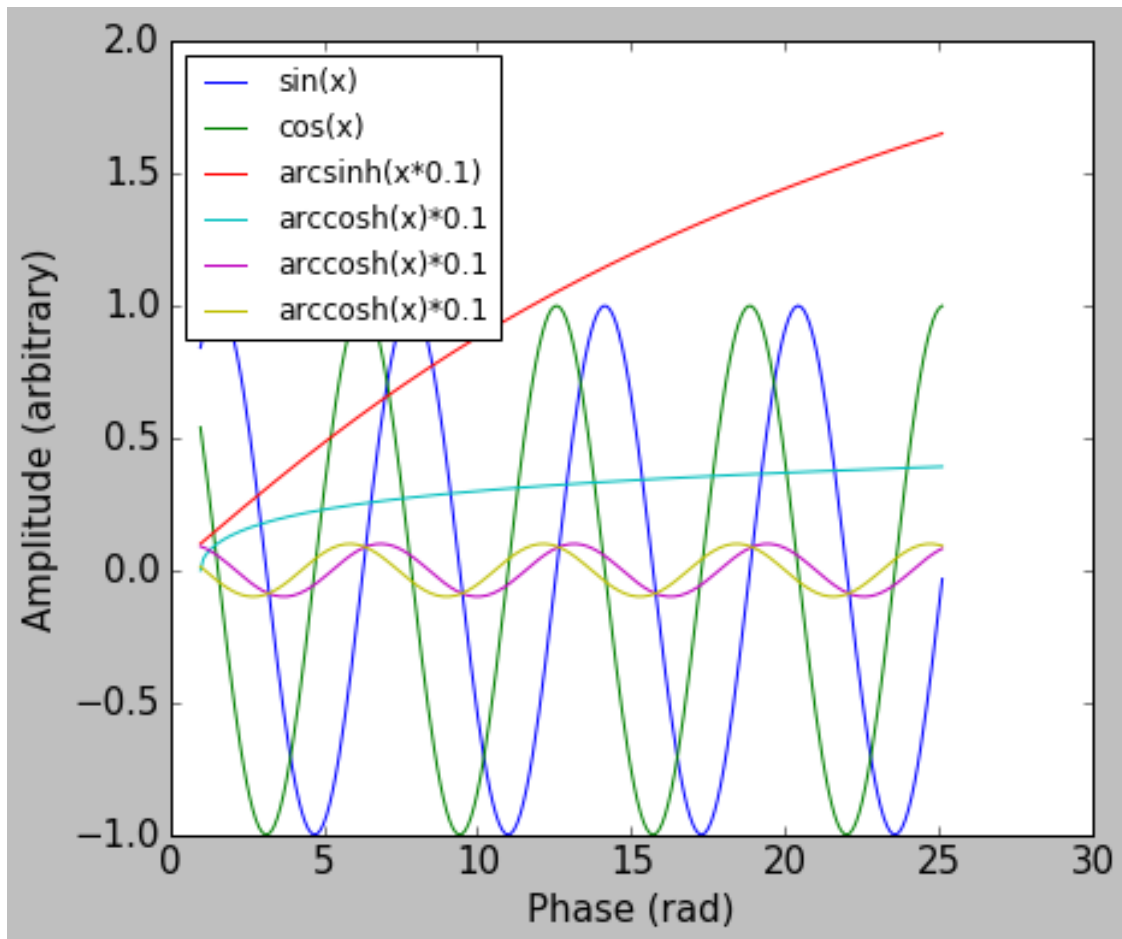
```
In [8]: draw_it('ggplot')
```

/usr/local/lib/python3.4/dist-packages/matplotlib/_init_.py:872: UserWarning: axes.color_cycle is deprecated and ignored.
warnings.warn(self.msg_depr % (key, alt_key))

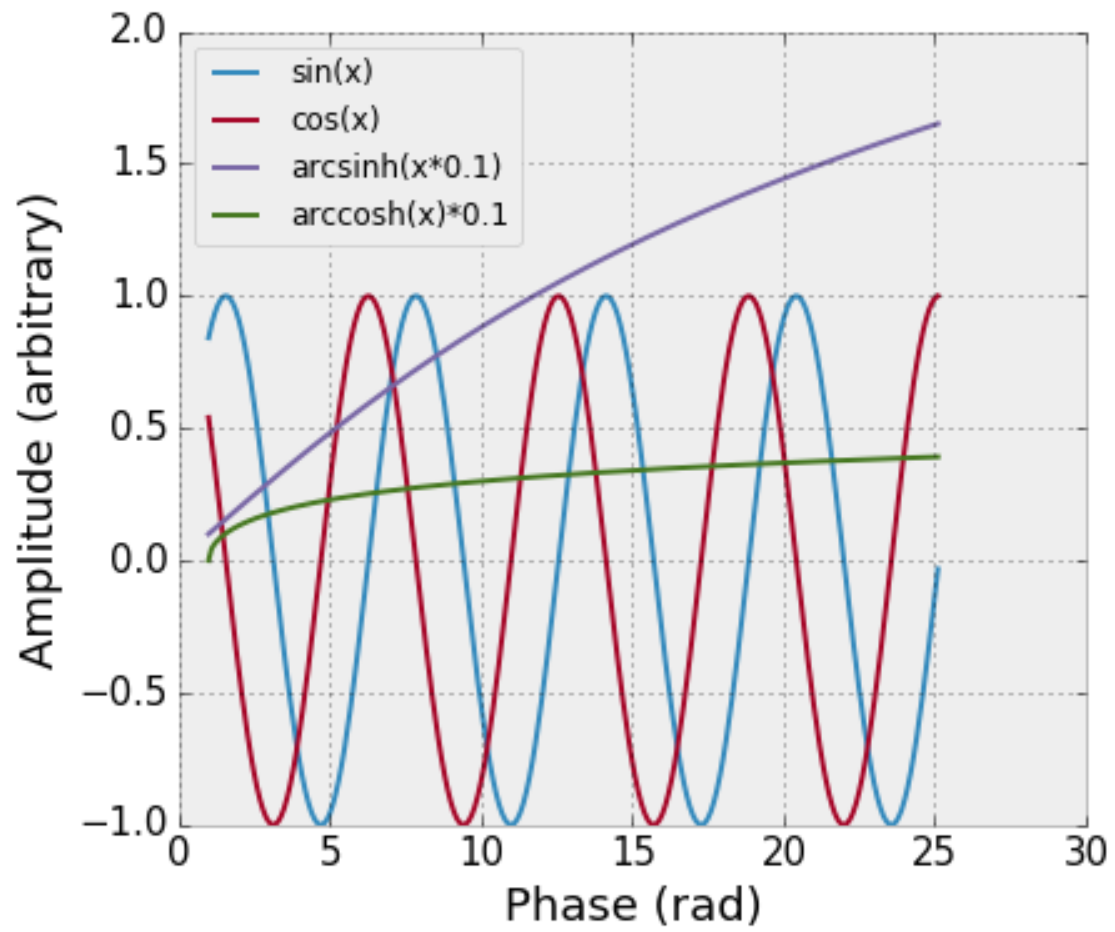


```
In [7]: draw_it('classic')
```

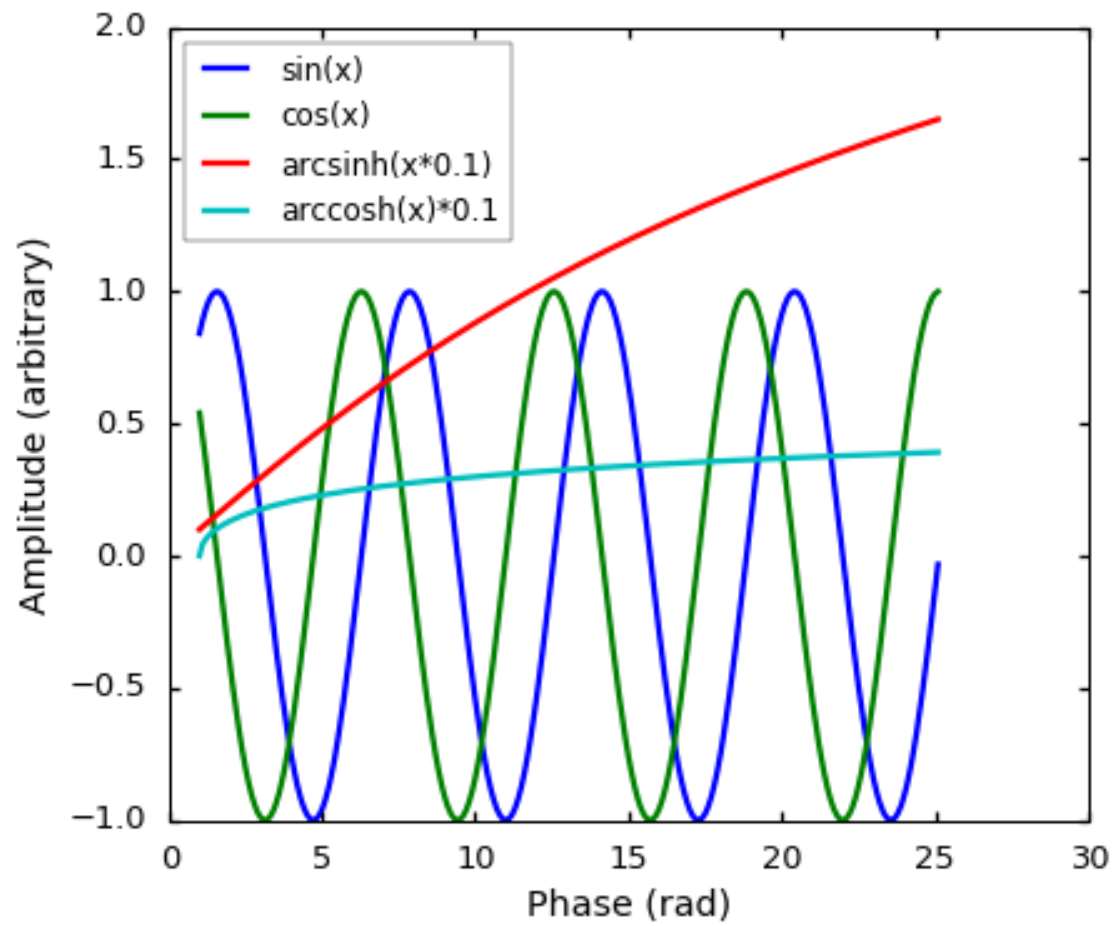
```
/usr/local/lib/python3.4/dist-packages/matplotlib/__init__.py:872: UserWarning: axes.color_cycle is deprecated and axes.patches.color_cycle will be removed in a future version
  warnings.warn(self.msg_depr % (key, alt_key))
```



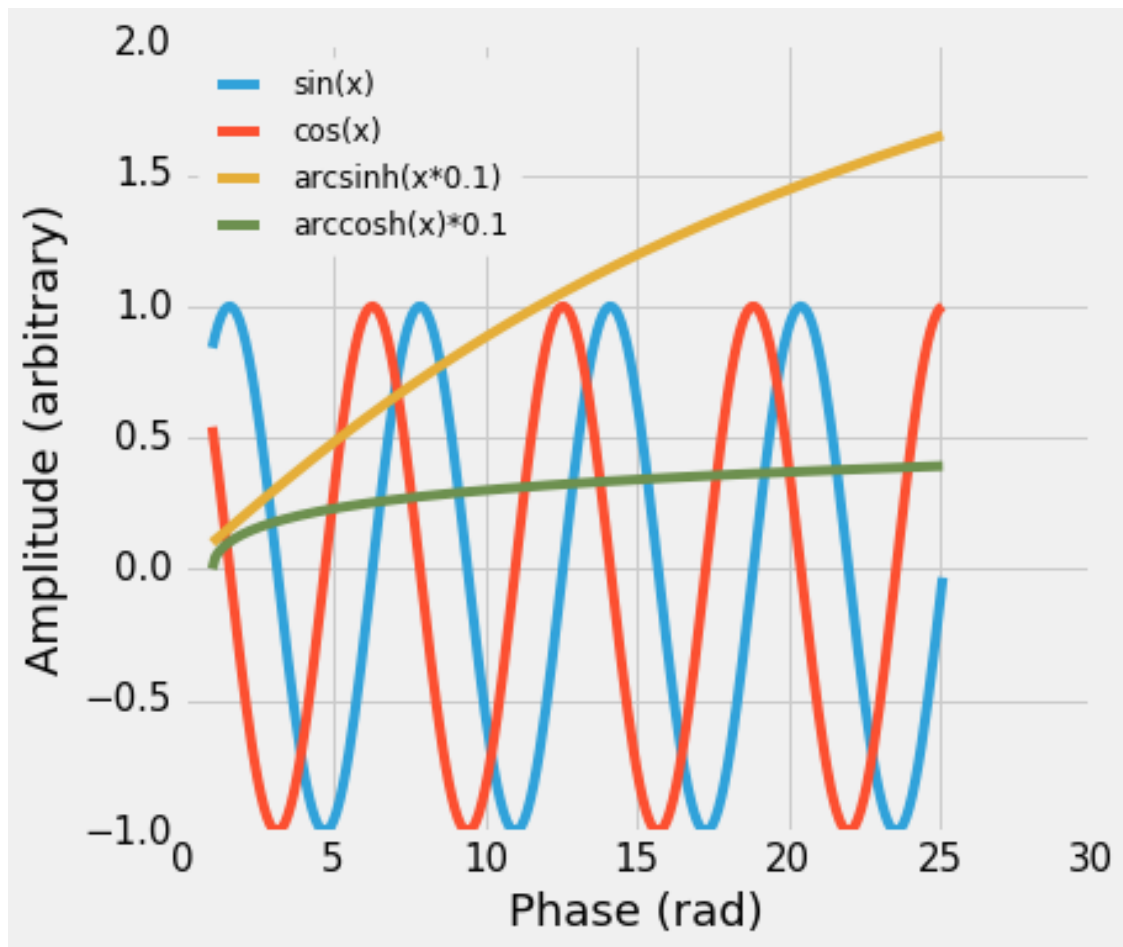
```
In [6]: draw_it('bmh')
```

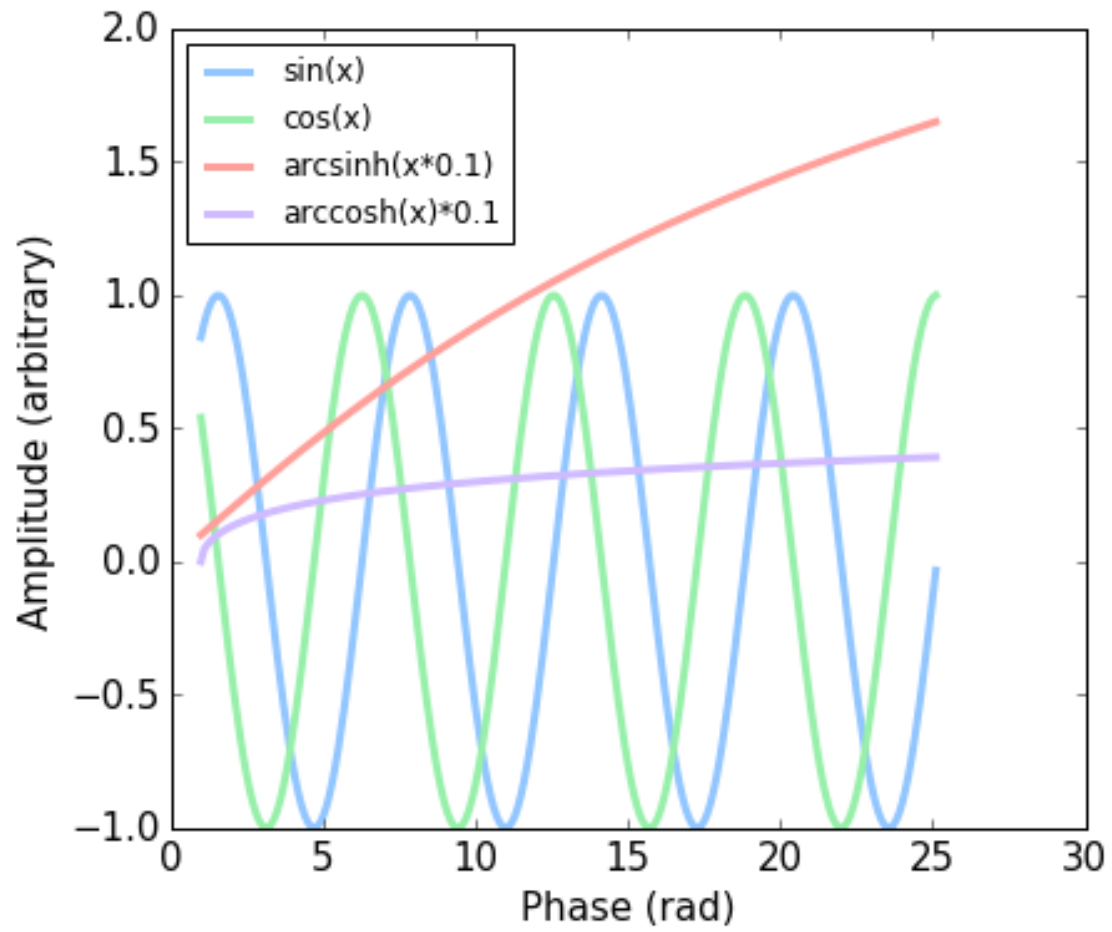
```
In [7]: draw_it('seaborn-talk')
```



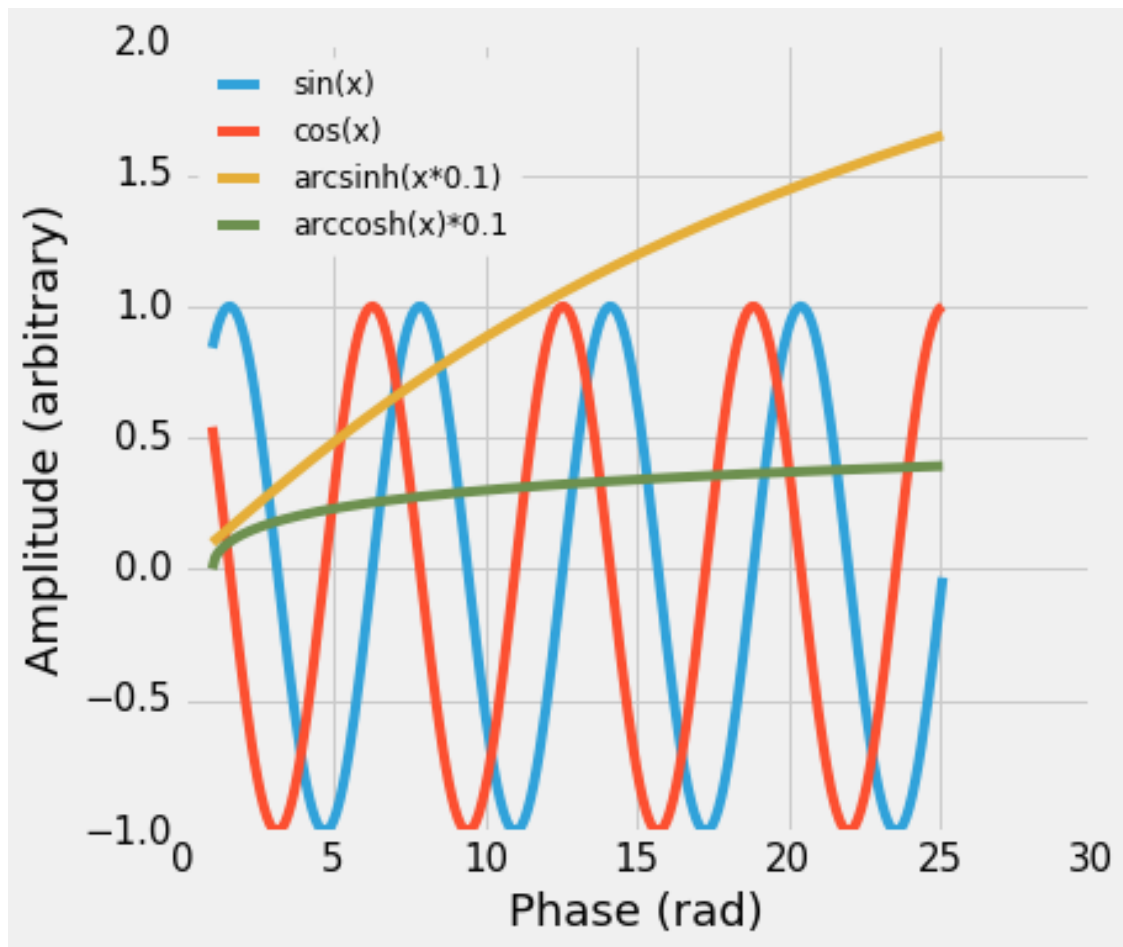
```
In [8]: draw_it('fivethirtyeight')
```



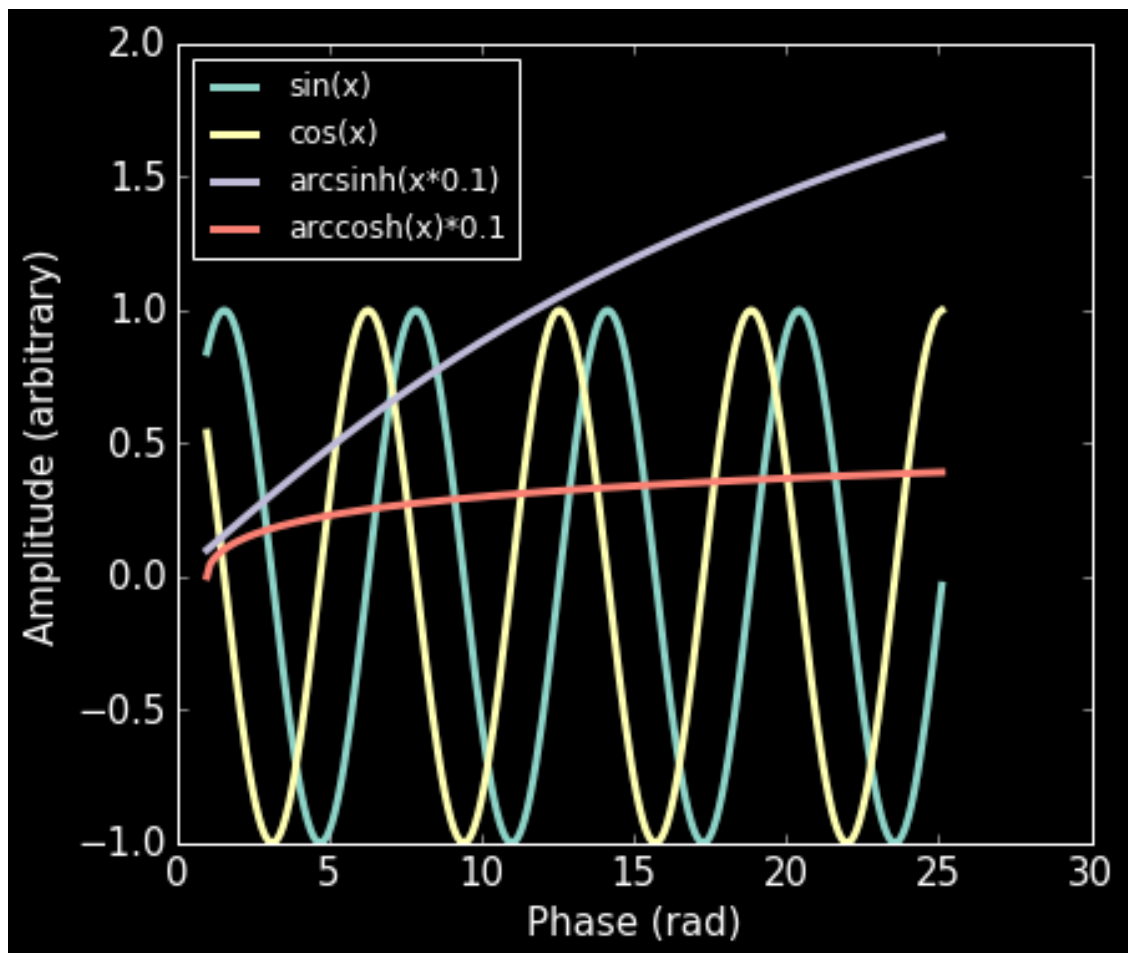
```
In [9]: draw_it('seaborn-pastel')
```



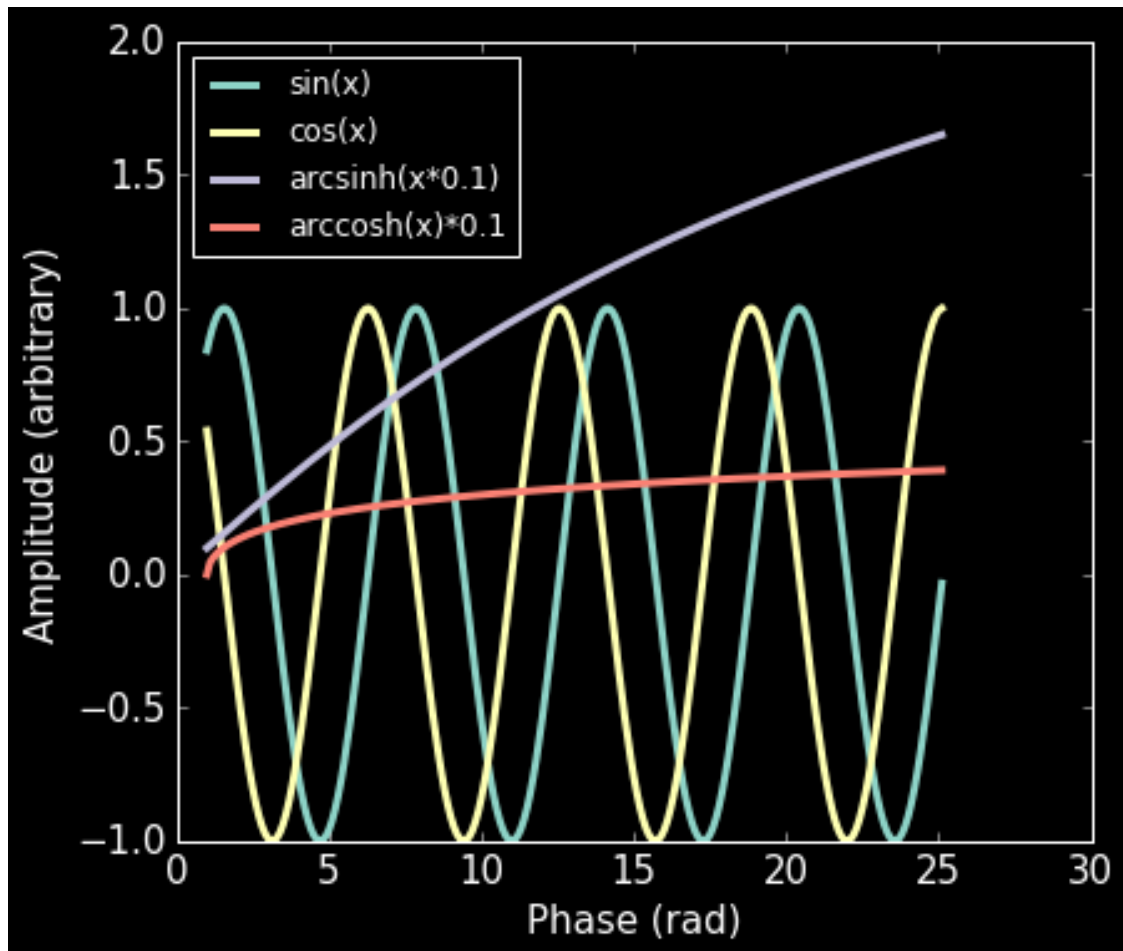
```
In [10]: draw_it(['seaborn-bright', 'fivethirtyeight'])
```



```
In [11]: draw_it('dark_background')
```



```
In [12]: draw_it('dark')
```



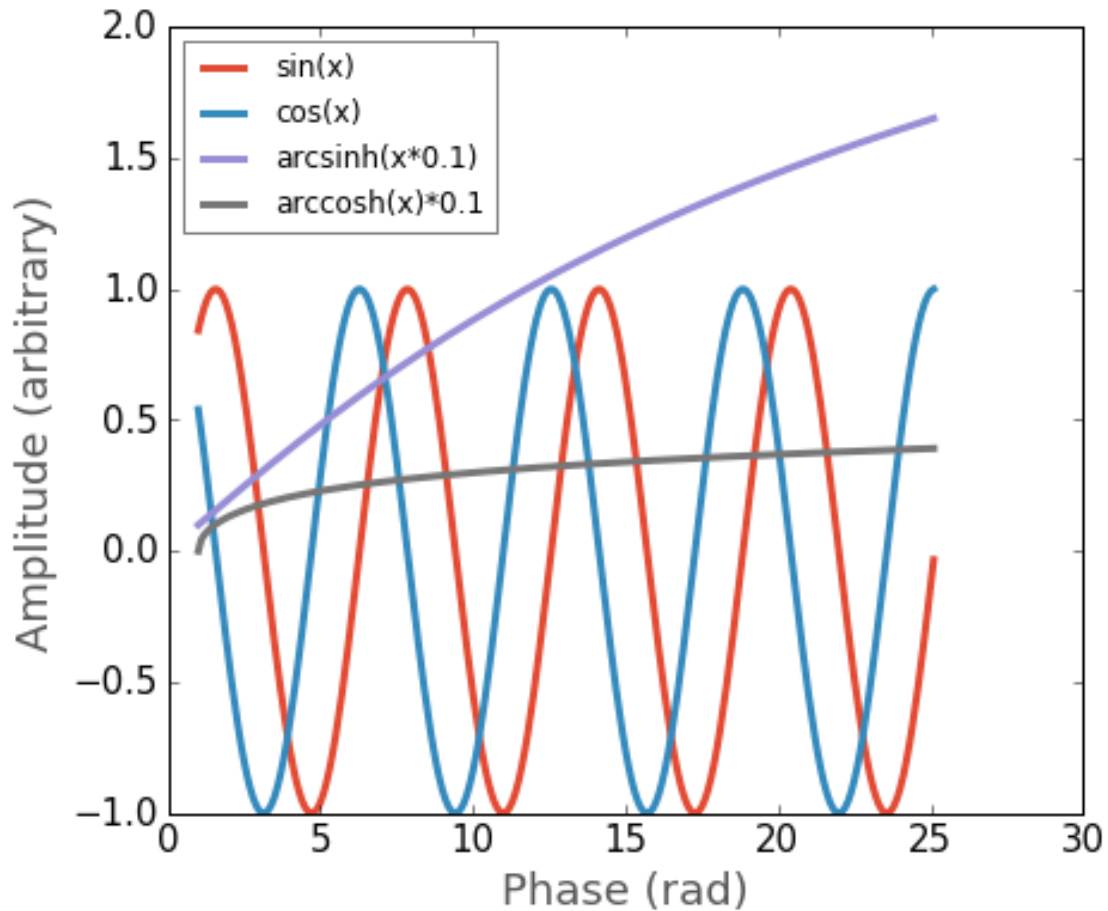
You can create your own styles, i.e. line width, colors, grid lines etc. This file is stored in the matplotlib user config directory.

2 Linux: \$HOME/.config/matplotlib/stylelib/mystyle.mplstyle

You can also read them in directly, I have supplied one in the sub-directory “styles”.

```
In [13]: import os
         cwd = os.getcwd()
         draw_it(os.path.join(cwd, 'styles/mystyle.mplstyle'))
```

```
/usr/local/lib/python3.4/dist-packages/matplotlib/_init_.py:872: UserWarning: axes.color_cycle is deprecated
  warnings.warn(self.msg_depr % (key, alt_key))
```



2.0.1 Exercise - Create your own style file

Now you should create your own matplotlib style file. Copy an existing one that you like and use that, or use the one that we supplied.

3 xkcd style plots (for fun)

```
In [15]: with plt.xkcd():
    plt.figure(figsize=(5.5,4))
    y1 = lambda x: x/20.
    y2 = lambda x: np.ones_like(x)
    y3 = lambda x: x/-10+5
    x1 = np.arange(0,20,0.5)
    x2 = np.arange(20,40,0.5)
    x3 = np.arange(40,50,0.5)
    y1obs = y1(x1) + np.random.randn(len(x1))/7
    plt.plot(x1,y1obs,'k.')
    y2obs = y2(x2) + np.random.randn(len(x2))/10
    plt.plot(x2,y2obs,'k.')
    y3obs = y3(x3) + np.random.randn(len(x3))/7
```

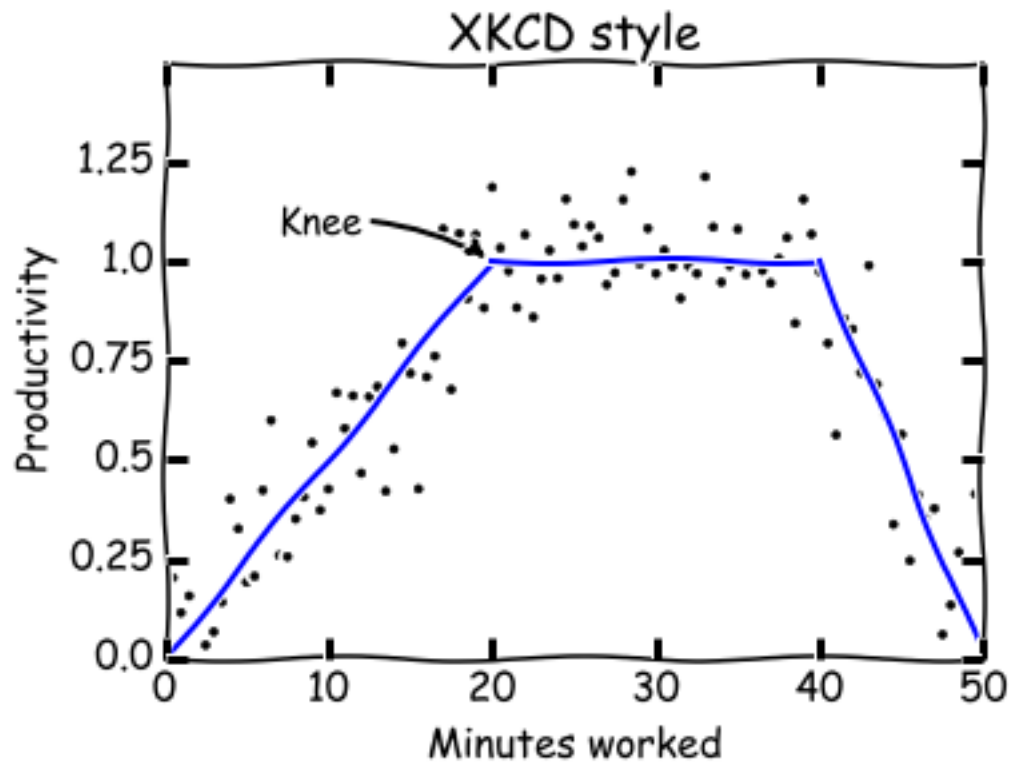


```

plt.plot(x3,y3obs,'k.')
plt.plot([0,20],y1(np.array([0., 20.])), 'b')
plt.plot([20,40],y2(np.array([20.,40.])), 'b')
plt.plot([40,50],y3(np.array([40.,50.])), 'b')
arrowprops=dict(arrowstyle="->",
                 connectionstyle="arc3,rad=-0.2",
                 fc="k", color='k', lw=2)

plt.annotate(s='Knee',
            xy=[20,1],
            xytext=[-80,10],
            textcoords='offset points',
            arrowprops=arrowprops)
yticks = np.arange(0,1.4,0.25)
plt.yticks(yticks, [str(i) for i in yticks])
plt.ylabel('Productivity')
plt.xlabel('Minutes worked')
plt.ylim((0,1.5))
plt.title('XKCD style')
y3(np.array([40., 50.]));

```



In [16]: `import antigravity`

4 HTML color names

All of the HTML color names can be used in plots as " color='name' ".
One example

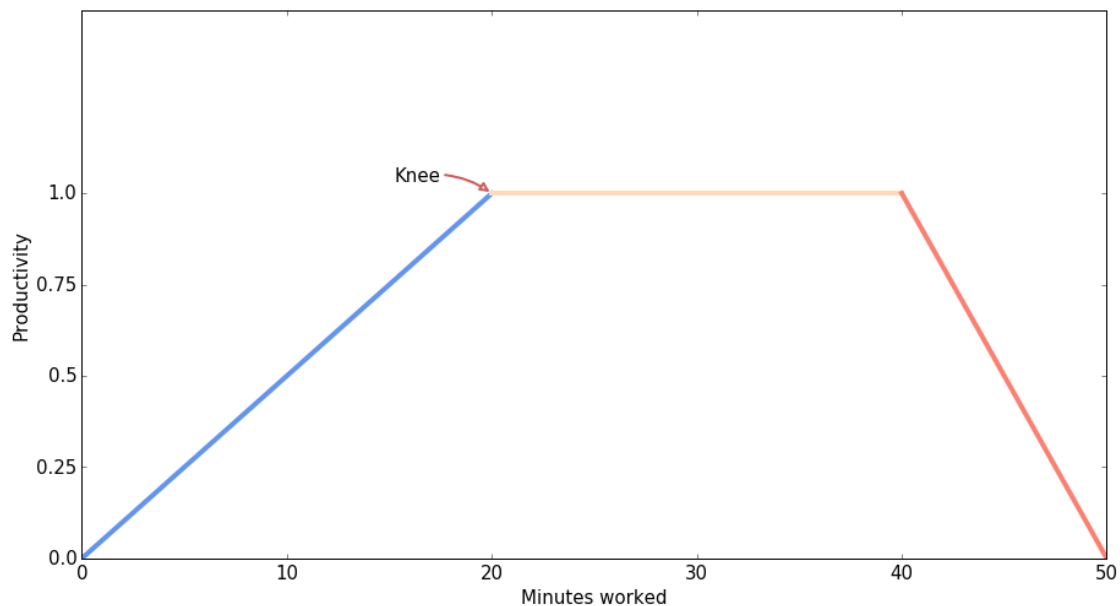
```

In [18]: plt.figure(figsize=(15,8))
plt.plot([0,20],[0,1], 'CornflowerBlue', lw=4)
plt.plot([20,40],[1,1], 'PeachPuff', lw=4)
plt.plot([40,50],[1,0], 'Salmon', lw=4)
arrowprops=dict(arrowstyle="->",
                  connectionstyle="arc3,rad=-0.2",
                  fc="Linen", color='IndianRed', lw=2)

plt.annotate(s='Knee',
             xy=[20,1],
             xytext=[-80,10],
             textcoords='offset points',
             arrowprops=arrowprops)

plt.ylim((0,1.5))
yticks = np.arange(0,1.2,0.25)
plt.yticks(yticks, [str(i) for i in yticks])
plt.ylabel('Productivity')
plt.xlabel('Minutes worked');

```



In []: