matplotlib

Fork me on GitHub

Version 3.1.1

home | examples | tutorials | API | contents » User's Guide » previous | next | modules | index

Tutorials »

# Customizing Matplotlib with style sheets and rcParams

Tips for customizing the properties and default styles of Matplotlib.

## Using style sheets

The `style` package adds support for easy-to-switch plotting "styles" with the same parameters as a matplotlib rc file (which is read at startup to configure matplotlib).

There are a number of pre-defined styles provided by Matplotlib. For example, there's a pre-defined style called "ggplot", which emulates the aesthetics of ggplot (a popular plotting package for R). To use this style, just add:

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
plt.style.use('ggplot')
data = np.random.randn(50)
```

To list all available styles, use:

```
print(plt.style.available)
```

Out:

```
['seaborn-ticks', 'ggplot', 'dark_background', 'bmh', 'seab
```

## Defining your own style

You can create custom styles and use them by calling `style.use` with the path or URL to the style sheet. Additionally, if you add your `<style-name>.mplstyle` file to `mpl_configdir/stylelib`, you can reuse your custom style sheet with a call to `style.use(<style-name>)`. By default `mpl_configdir` should be `~/.config/matplotlib`, but you can check where yours is with `matplotlib.get_configdir()`; you may need to create this directory. You also can change the directory where matplotlib looks for the stylelib/ folder by setting the MPLCONFIGDIR environment variable, see matplotlib configuration and cache directory locations.

Quick search

Go

Table of Contents

Related Topics

Documentation overview

- User's Guide
  - Tutorials
    - Previous: The Lifecycle of a Plot
    - Next: Artist tutorial

Show Page Source

Note that a custom style sheet in `mpl_configdir/stylelib` will override a style sheet defined by matplotlib if the styles have the same name.

For example, you might want to create `mpl_configdir/stylelib/presentation.mplstyle` with the following:

```
axes.titlesize : 24
axes.labelsize : 20
lines.linewidth : 3
lines.markersize : 10
xtick.labelsize : 16
ytick.labelsize : 16
```

Then, when you want to adapt a plot designed for a paper to one that looks good in a presentation, you can just add:

```
>>> import matplotlib.pyplot as plt
>>> plt.style.use('presentation')
```

## Composing styles

Style sheets are designed to be composed together. So you can have a style sheet that customizes colors and a separate style sheet that alters element sizes for presentations. These styles can easily be combined by passing a list of styles:

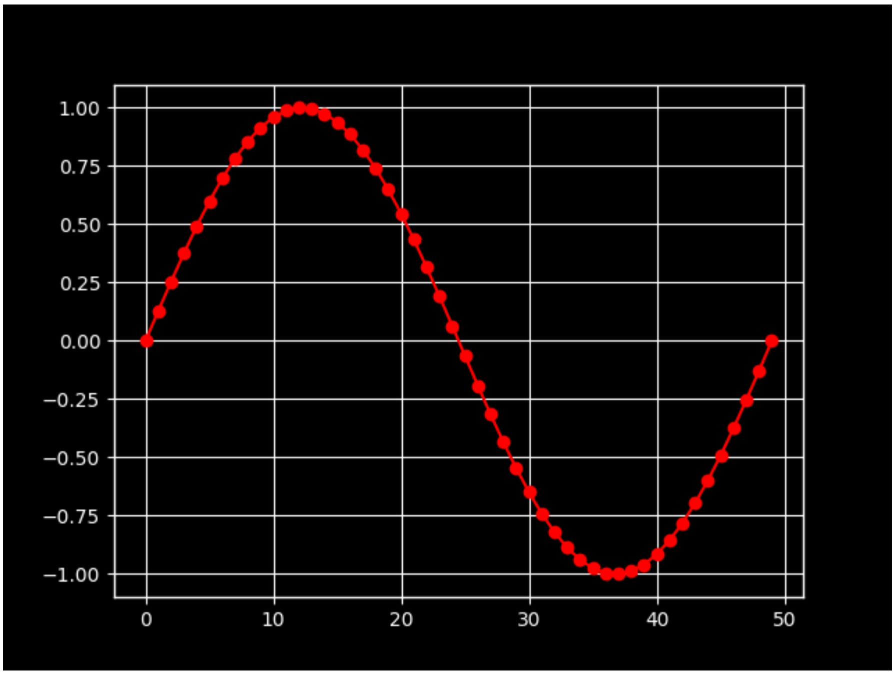
```
>>> import matplotlib.pyplot as plt
>>> plt.style.use(['dark_background', 'presentation'])
```

Note that styles further to the right will overwrite values that are already defined by styles on the left.

## Temporary styling

If you only want to use a style for a specific block of code but don't want to change the global styling, the style package provides a context manager for limiting your changes to a specific scope. To isolate your styling changes, you can write something like the following:

```
with plt.style.context('dark_background'):
    plt.plot(np.sin(np.linspace(0, 2 * np.pi)), 'r-o')
plt.show()
```

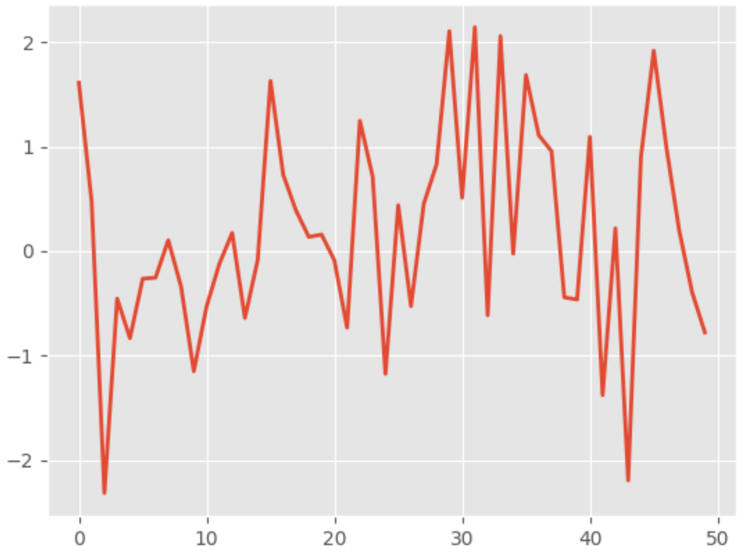


# matplotlib rcParams

## Dynamic rc settings

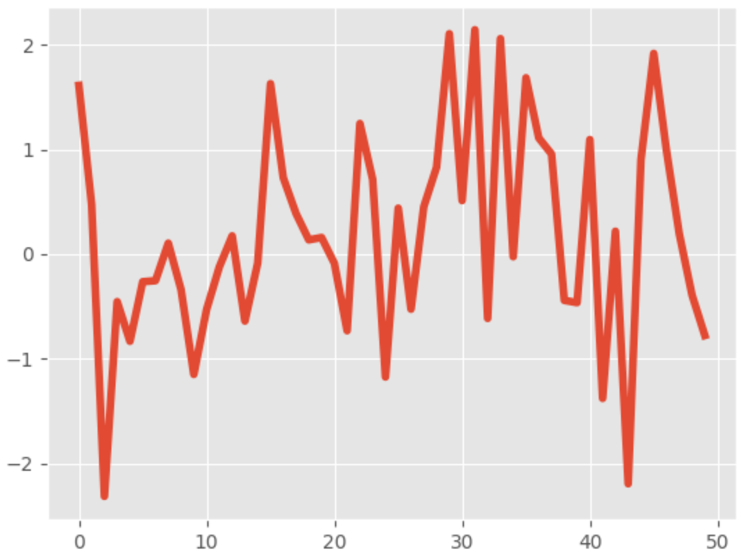
You can also dynamically change the default rc settings in a python script or interactively from the python shell. All of the rc settings are stored in a dictionary-like variable called `matplotlib.rcParams`, which is global to the matplotlib package. rcParams can be modified directly, for example:

```
mpl.rcParams['lines.linewidth'] = 2
mpl.rcParams['lines.color'] = 'r'
plt.plot(data)
```



Matplotlib also provides a couple of convenience functions for modifying rc settings. The `matplotlib.rc()` command can be used to modify multiple settings in a single group at once, using keyword arguments:

```
mpl.rc('lines', linewidth=4, color='g')
plt.plot(data)
```



The `matplotlib.rcdefaults()` command will restore the standard matplotlib default settings.

There is some degree of validation when setting the values of rcParams, see `matplotlib.rcsetup` for details.

## The `matplotlibrc` file

matplotlib uses `matplotlibrc` configuration files to customize all kinds of properties, which we call `rc settings` or `rc parameters`. You can control the defaults of almost every property in matplotlib: figure size and dpi, line width, color and style, axes, axis and grid properties, text and font properties and so on. matplotlib looks for `matplotlibrc` in four locations, in the following order:

1. `matplotlibrc` in the current working directory, usually used for specific customizations that you do not want to apply elsewhere.
2. `$MATPLOTLIBRC` if it is a file, else `$MATPLOTLIBRC/matplotlibrc`.
3. It next looks in a user-specific place, depending on your platform:
    - On Linux and FreeBSD, it looks in `.config/matplotlib/matplotlibrc` (or `$XDG_CONFIG_HOME/matplotlib/matplotlibrc`) if you've customized your environment.
    - On other platforms, it looks in `.matplotlib/matplotlibrc`.

    See matplotlib configuration and cache directory locations.
4. *`INSTALL`*`/matplotlib/mpl-data/matplotlibrc`, where *`INSTALL`* is something like `/usr/lib/python3.7/site-packages` on Linux, and maybe `C:\Python37\Lib\site-packages` on Windows. Every time you install matplotlib, this file will be overwritten, so if you want

your customizations to be saved, please move this file to your user-specific matplotlib directory.

Once a `matplotlibrc` file has been found, it will *not* search any of the other paths.

To display where the currently active `matplotlibrc` file was loaded from, one can do the following:

```
>>> import matplotlib
>>> matplotlib.matplotlib_fname()
'/home/foo/.config/matplotlib/matplotlibrc'
```

See below for a sample matplotlibrc file.

## A sample matplotlibrc file

```
#### MATPLOTLIBRC FORMAT

## This is a sample matplotlib configuration file - you can fi
## of it on your system in
## site-packages/matplotlib/mpl-data/matplotlibrc.  If you edit
## there, please note that it will be overwritten in your next
## If you want to keep a permanent local copy that will not be
## overwritten, place it in the following location:
## unix/linux:
##      $HOME/.config/matplotlib/matplotlibrc or
##      $XDG_CONFIG_HOME/matplotlib/matplotlibrc (if $XDG_CONF
## other platforms:
##      $HOME/.matplotlib/matplotlibrc
##
## See http://matplotlib.org/users/customizing.html#the-matplot
## more details on the paths which are checked for the configu
##
## This file is best viewed in a editor which supports python 
## syntax highlighting. Blank lines, or lines starting with a 
## symbol, are ignored, as are trailing comments.  Other lines
## have the format
##     key : val ## optional comment
##
## Colors: for the color values below, you can either use - a
## matplotlib color string, such as r, k, or b - an rgb tuple,
## (1.0, 0.5, 0.0) - a hex string, such as ff00ff - a scalar
## grayscale intensity such as 0.75 - a legal html color name,
## blue, darkslategray

##### CONFIGURATION BEGINS HERE

## The default backend. If you omit this parameter, the first
## working backend from the following list is used:
## MacOSX Qt5Agg Qt4Agg Gtk3Agg TkAgg WxAgg Agg
##
## Other choices include:
## Qt5Cairo Qt4Cairo GTK3Cairo TkCairo WxCairo Cairo Wx PS PDF
##
## You can also deploy your own backend outside of matplotlib 
## referring to the module name (which must be in the PYTHONPA
## 'module://my_backend'.
#backend      : Agg
```

```
## Note that this can be overridden by the environment variable
## QT_API used by Enthought Tool Suite (ETS); valid values are
## "pyqt" and "pyside".  The "pyqt" setting has the side effect
## forcing the use of Version 2 API for QString and QVariant.

## The port to use for the web server in the WebAgg backend.
#webagg.port : 8988

## The address on which the WebAgg web server should be reachab
#webagg.address : 127.0.0.1

## If webagg.port is unavailable, a number of other random port
## be tried until one that is available is found.
#webagg.port_retries : 50

## When True, open the webbrowser to the plot that is shown
#webagg.open_in_browser : True

## if you are running pyplot inside a GUI and your backend choi
## conflicts, we will automatically try to find a compatible on
## you if backend_fallback is True
#backend_fallback: True

#interactive  : False
#toolbar      : toolbar2   ## None | toolbar2  ("classic" is de
#timezone     : UTC        ## a pytz timezone string, e.g., US/

## Where your matplotlib data lives if you installed to a non-d
## location.  This is where the matplotlib fonts, bitmaps, etc
#datapath : /home/jdhunter/mpldata


#### LINES
## See http://matplotlib.org/api/artist_api.html#module-matplot
## information on line properties.
#lines.linewidth   : 1.5     ## line width in points
#lines.linestyle   : -       ## solid line
#lines.color       : C0      ## has no affect on plot(); see ax
#lines.marker      : None    ## the default marker
#lines.markerfacecolor  : auto    ## the default markerfacecolo
#lines.markeredgecolor  : auto    ## the default markeredgecolo
#lines.markeredgewidth  : 1.0     ## the line width around the
#lines.markersize  : 6            ## markersize, in points
#lines.dash_joinstyle : round        ## miter|round|bevel
#lines.dash_capstyle : butt          ## butt|round|projecting
#lines.solid_joinstyle : round       ## miter|round|bevel
#lines.solid_capstyle : projecting   ## butt|round|projecting
#lines.antialiased : True         ## render lines in antialiase

## The three standard dash patterns.  These are scaled by the l
#lines.dashed_pattern : 3.7, 1.6
#lines.dashdot_pattern : 6.4, 1.6, 1, 1.6
#lines.dotted_pattern : 1, 1.65
#lines.scale_dashes : True

#markers.fillstyle: full ## full|left|right|bottom|top|none

#### PATCHES
## Patches are graphical objects that fill 2D space, like polyg
## circles.  See
## http://matplotlib.org/api/artist_api.html#module-matplotlib.
## information on patch properties
#patch.linewidth        : 1        ## edge width in points.
```

```
#patch.facecolor        : C0
#patch.edgecolor        : black   ## if forced, or patch is not
#patch.force_edgecolor  : False   ## True to always use edgecol
#patch.antialiased      : True    ## render patches in antialia

#### HATCHES
#hatch.color     : black
#hatch.linewidth : 1.0

#### Boxplot
#boxplot.notch       : False
#boxplot.vertical    : True
#boxplot.whiskers    : 1.5
#boxplot.bootstrap   : None
#boxplot.patchartist : False
#boxplot.showmeans   : False
#boxplot.showcaps    : True
#boxplot.showbox     : True
#boxplot.showfliers  : True
#boxplot.meanline    : False

#boxplot.flierprops.color           : black
#boxplot.flierprops.marker          : o
#boxplot.flierprops.markerfacecolor : none
#boxplot.flierprops.markeredgecolor : black
#boxplot.flierprops.markeredgewidth : 1.0
#boxplot.flierprops.markersize      : 6
#boxplot.flierprops.linestyle       : none
#boxplot.flierprops.linewidth       : 1.0

#boxplot.boxprops.color     : black
#boxplot.boxprops.linewidth : 1.0
#boxplot.boxprops.linestyle : -

#boxplot.whiskerprops.color     : black
#boxplot.whiskerprops.linewidth : 1.0
#boxplot.whiskerprops.linestyle : -

#boxplot.capprops.color     : black
#boxplot.capprops.linewidth : 1.0
#boxplot.capprops.linestyle : -

#boxplot.medianprops.color     : C1
#boxplot.medianprops.linewidth : 1.0
#boxplot.medianprops.linestyle : -

#boxplot.meanprops.color           : C2
#boxplot.meanprops.marker          : ^
#boxplot.meanprops.markerfacecolor : C2
#boxplot.meanprops.markeredgecolor : C2
#boxplot.meanprops.markersize      :  6
#boxplot.meanprops.linestyle       : --
#boxplot.meanprops.linewidth       : 1.0


#### FONT

## font properties used by text.Text.  See
## http://matplotlib.org/api/font_manager_api.html for more
## information on font properties.  The 6 font properties used
## matching are given below with their default values.
##
## The font.family property has five values: 'serif' (e.g., Tin
## 'sans-serif' (e.g., Helvetica), 'cursive' (e.g., Zapf-Chance
## 'fantasy' (e.g., Western), and 'monospace' (e.g., Courier).
## these font families has a default list of font names in decr
## order of priority associated with them.  When text.usetex is
## font.family may also be one or more concrete font names.
##
## The font.style property has three values: normal (or roman),
## or oblique.  The oblique style will be used for italic, if i
## present.
##
## The font.variant property has two values: normal or small-ca
## TrueType fonts, which are scalable fonts, small-caps is equi
## to using a font size of 'smaller', or about 83%% of the curr
## size.
##
## The font.weight property has effectively 13 values: normal,
## bolder, lighter, 100, 200, 300, ..., 900.  Normal is the sam
## 400, and bold is 700.  bolder and lighter are relative value
## respect to the current weight.
##
## The font.stretch property has 11 values: ultra-condensed,
## extra-condensed, condensed, semi-condensed, normal, semi-exp
## expanded, extra-expanded, ultra-expanded, wider, and narrowe
## property is not currently implemented.
##
## The font.size property is the default font size for text, gi
## 10 pt is the standard value.

#font.family         : sans-serif
#font.style          : normal
#font.variant        : normal
#font.weight         : normal
#font.stretch        : normal
## note that font.size controls default text sizes.  To configu
## special text sizes tick labels, axes, labels, title, etc, se
## settings for axes and ticks. Special text sizes can be defin
## relative to font.size, using the following values: xx-small,
## small, medium, large, x-large, xx-large, larger, or smaller
#font.size           : 10.0
#font.serif          : DejaVu Serif, Bitstream Vera Serif, Comp
#font.sans-serif     : DejaVu Sans, Bitstream Vera Sans, Comput
#font.cursive        : Apple Chancery, Textile, Zapf Chancery,
#font.fantasy        : Comic Sans MS, Chicago, Charcoal, Impact
#font.monospace      : DejaVu Sans Mono, Bitstream Vera Sans Mo

#### TEXT
## text properties used by text.Text.  See
## http://matplotlib.org/api/artist_api.html#module-matplotlib.
## information on text properties
#text.color          : black

#### LaTeX customizations. See http://wiki.scipy.org/Cookbook/M
#text.usetex         : False  ## use latex for all text handlin
                              ## are supported through the usua
                              ## new century schoolbook, bookma
                              ## zapf chancery, charter, serif,
                              ## avant garde, courier, monospac
                              ## computer modern sans serif, co
                              ## If another font is desired whi
                              ## LaTeX \usepackage command, ple
                              ## matplotlib mailing list
#text.latex.preamble :      ## IMPROPER USE OF THIS FEATURE WIL
                            ## AND IS THEREFORE UNSUPPORTED. PL
                            ## IF THIS FEATURE DOES NOT DO WHA
                            ## text.latex.preamble is a single
                            ## will be passed on to the LaTeX s
                            ## any code that is valid for the L
                            ## between the "\documentclass" and
                            ## statements.
                            ## Note that it has to be put on a
                            ## become quite long.
                            ## The following packages are alway
                            ## beware of package collisions: co
                            ## type1cm, textcomp.
                            ## Adobe Postscript (PSSNFS) font p
                            ## loaded, depending on your font s
#text.latex.preview : False

#text.hinting : auto   ## May be one of the following:
                       ##   none: Perform no hinting
                       ##   auto: Use FreeType's autohinter
                       ##   native: Use the hinting information
                       #              font file, if available,
                       #              FreeType library supports
                       ##   either: Use the native hinting info
                       #              or the autohinter if none
                       ## For backward compatibility, this valu
                       ## True === 'auto' or False === 'none'.
#text.hinting_factor : 8 ## Specifies the amount of softness fo
                         ## horizontal direction.  A value of 1
                         ## pixels.  A value of 2 will hint to
#text.antialiased : True ## If True (default), the text will be
                         ## This only affects the Agg backend.

## The following settings allow you to select the fonts in math
## They map from a TeX font name to a fontconfig font pattern.
## These settings are only used if mathtext.fontset is 'custom
## Note that this "custom" mode is unsupported and may go away
## future.
#mathtext.cal : cursive
#mathtext.rm  : sans
#mathtext.tt  : monospace
#mathtext.it  : sans:italic
#mathtext.bf  : sans:bold
#mathtext.sf  : sans
#mathtext.fontset : dejavusans ## Should be 'dejavusans' (defau
                               ## 'dejavuserif', 'cm' (Computer
                               ## 'stixsans' or 'custom'
#mathtext.fallback_to_cm : True  ## When True, use symbols from
                                 ## fonts when a symbol can not
                                 ## the custom math fonts.
#mathtext.default : it ## The default font to use for math.
                       ## Can be any of the LaTeX font names, i
                       ## the special name "regular" for the sa
                       ## used in regular text.

#### AXES
## default face and edge color, default tick sizes,
## default fontsizes for ticklabels, and so on.  See
## http://matplotlib.org/api/axes_api.html#module-matplotlib.ax
#axes.facecolor      : white   ## axes background color
#axes.edgecolor      : black   ## axes edge color
#axes.linewidth      : 0.8     ## edge linewidth
#axes.grid           : False   ## display grid or not
#axes.grid.axis      : both    ## which axis the grid should ap
#axes.grid.which     : major   ## gridlines at major, minor or
```

```
#axes.titlesize      : large   ## fontsize of the axes title
#axes.titleweight    : normal  ## font weight of title
#axes.titlepad       : 6.0     ## pad between axes and title in
#axes.labelsize      : medium  ## fontsize of the x any y label
#axes.labelpad       : 4.0     ## space between label and axis
#axes.labelweight    : normal  ## weight of the x and y labels
#axes.labelcolor     : black
#axes.axisbelow      : line    ## draw axis gridlines and ticks
                               ## patches (True); above patches
                               ## lines ('line'); or above all
#axes.formatter.limits : -7, 7 ## use scientific notation if lo
                               ## of the axis range is smaller
                               ## first or larger than the seco
#axes.formatter.use_locale : False ## When True, format tick lo
                                   ## according to the user's l
                                   ## For example, use ',' as a
                                   ## separator in the fr_FR lo
#axes.formatter.use_mathtext : False ## When True, use mathtext
                                     ## notation.
#axes.formatter.min_exponent: 0 ## minimum exponent to format i
#axes.formatter.useoffset      : True    ## If True, the tick l
                                         ## will default to lab
                                         ## to an offset when t
                                         ## small compared to t
                                         ## value of the data.
#axes.formatter.offset_threshold : 4     ## When useoffset is T
                                         ## will be used when i
                                         ## at least this numbe
                                         ## digits from tick lo
#axes.spines.left   : True   ## display axis spines
#axes.spines.bottom : True
#axes.spines.top    : True
#axes.spines.right  : True
#axes.unicode_minus  : True    ## use unicode for the minus sym
                               ## rather than hyphen.  See
                               ## http://en.wikipedia.org/wiki/
#axes.prop_cycle    : cycler('color', ['1f77b4', 'ff7f0e', '2ca
                      ## color cycle for plot lines  as list of
                      ## colorspecs: single letter, long name,
                                         ## Note the use of str
                      ## as opposed to the rest of this file.
#axes.autolimit_mode : data ## How to scale axes limits to the
                            ## Use "data" to use data limits, p
                            ## Use "round_number" move to the n
#axes.xmargin        : .05  ## x margin.  See `axes.Axes.margin
#axes.ymargin        : .05  ## y margin See `axes.Axes.margins`
#polaraxes.grid      : True    ## display grid on polar axes
#axes3d.grid         : True    ## display grid on 3d axes

#### DATES
## These control the default format strings used in AutoDateFor
## Any valid format datetime format string can be used (see the
## `datetime` for details).  For example using '%%x' will use t
## '%%X' will use the locale time representation and '%%c' will
## representation.
## These values map to the scales:
##     {'year': 365, 'month': 30, 'day': 1, 'hour': 1/24, 'minu

#date.autoformatter.year     : %Y
#date.autoformatter.month    : %Y-%m
#date.autoformatter.day      : %Y-%m-%d
#date.autoformatter.hour     : %m-%d %H
#date.autoformatter.minute   : %d %H:%M
```

```
#date.autoformatter.second   : %H:%M:%S
#date.autoformatter.microsecond   : %M:%S.%f

#### TICKS
## see http://matplotlib.org/api/axis_api.html#matplotlib.axis.
#xtick.top            : False  ## draw ticks on the top side
#xtick.bottom         : True   ## draw ticks on the bottom side
#xtick.labeltop       : False  ## draw label on the top
#xtick.labelbottom    : True   ## draw label on the bottom
#xtick.major.size     : 3.5    ## major tick size in points
#xtick.minor.size     : 2      ## minor tick size in points
#xtick.major.width    : 0.8    ## major tick width in points
#xtick.minor.width    : 0.6    ## minor tick width in points
#xtick.major.pad      : 3.5    ## distance to major tick label
#xtick.minor.pad      : 3.4    ## distance to the minor tick la
#xtick.color          : black  ## color of the tick labels
#xtick.labelsize      : medium ## fontsize of the tick labels
#xtick.direction      : out    ## direction: in, out, or inout
#xtick.minor.visible  : False  ## visibility of minor ticks on
#xtick.major.top      : True   ## draw x axis top major ticks
#xtick.major.bottom   : True   ## draw x axis bottom major tick
#xtick.minor.top      : True   ## draw x axis top minor ticks
#xtick.minor.bottom   : True   ## draw x axis bottom minor tick
#xtick.alignment      : center ## alignment of xticks

#ytick.left           : True   ## draw ticks on the left side
#ytick.right          : False  ## draw ticks on the right side
#ytick.labelleft      : True   ## draw tick labels on the left
#ytick.labelright     : False  ## draw tick labels on the right
#ytick.major.size     : 3.5    ## major tick size in points
#ytick.minor.size     : 2      ## minor tick size in points
#ytick.major.width    : 0.8    ## major tick width in points
#ytick.minor.width    : 0.6    ## minor tick width in points
#ytick.major.pad      : 3.5    ## distance to major tick label
#ytick.minor.pad      : 3.4    ## distance to the minor tick la
#ytick.color          : black  ## color of the tick labels
#ytick.labelsize      : medium ## fontsize of the tick labels
#ytick.direction      : out    ## direction: in, out, or inout
#ytick.minor.visible  : False  ## visibility of minor ticks on
#ytick.major.left     : True   ## draw y axis left major ticks
#ytick.major.right    : True   ## draw y axis right major ticks
#ytick.minor.left     : True   ## draw y axis left minor ticks
#ytick.minor.right    : True   ## draw y axis right minor ticks
#ytick.alignment      : center_baseline ## alignment of yticks

#### GRIDS
#grid.color       :   b0b0b0    ## grid color
#grid.linestyle   :   -         ## solid
#grid.linewidth   :   0.8       ## in points
#grid.alpha       :   1.0       ## transparency, between 0.0 an

#### Legend
#legend.loc           : best
#legend.frameon       : True     ## if True, draw the legend on
#legend.framealpha    : 0.8      ## legend patch transparency
#legend.facecolor     : inherit  ## inherit from axes.facecolor
#legend.edgecolor     : 0.8      ## background patch boundary c
#legend.fancybox      : True     ## if True, use a rounded box
                                 ## legend background, else a r
#legend.shadow        : False    ## if True, give background a
#legend.numpoints     : 1        ## the number of marker points
#legend.scatterpoints : 1        ## number of scatter points
#legend.markerscale   : 1.0      ## the relative size of legend
```

```
#legend.fontsize      : medium
#legend.title_fontsize    : None ## None sets to the same as th
## Dimensions as fraction of fontsize:
#legend.borderpad     : 0.4      ## border whitespace
#legend.labelspacing  : 0.5      ## the vertical space between
#legend.handlelength  : 2.0      ## the length of the legend li
#legend.handleheight  : 0.7      ## the height of the legend ha
#legend.handletextpad : 0.8      ## the space between the legen
#legend.borderaxespad : 0.5      ## the border between the axes
#legend.columnspacing : 2.0      ## column separation

#### FIGURE
## See http://matplotlib.org/api/figure_api.html#matplotlib.fig
#figure.titlesize : large      ## size of the figure title (Fig
#figure.titleweight : normal   ## weight of the figure title
#figure.figsize   : 6.4, 4.8   ## figure size in inches
#figure.dpi       : 100        ## figure dots per inch
#figure.facecolor : white      ## figure facecolor
#figure.edgecolor : white      ## figure edgecolor
#figure.frameon : True         ## enable figure frame
#figure.max_open_warning : 20  ## The maximum number of figures
                               ## the pyplot interface before e
                               ## If less than one this feature
## The figure subplot parameters.  All dimensions are a fractio
#figure.subplot.left    : 0.125  ## the left side of the subplo
#figure.subplot.right   : 0.9    ## the right side of the subpl
#figure.subplot.bottom  : 0.11   ## the bottom of the subplots
#figure.subplot.top     : 0.88   ## the top of the subplots of
#figure.subplot.wspace  : 0.2    ## the amount of width reserve
                                 ## expressed as a fraction of
#figure.subplot.hspace  : 0.2    ## the amount of height reserv
                                 ## expressed as a fraction of

## Figure layout
#figure.autolayout : False     ## When True, automatically adju
                               ## parameters to make the plot f
                               ## using `tight_layout`
#figure.constrained_layout.use: False ## When True, automatical
                                      ## elements fit on the fi
                                      ## with `autolayout`, abo
#figure.constrained_layout.h_pad : 0.04167 ## Padding around ax
#figure.constrained_layout.w_pad : 0.04167 ##  inches. Default
#figure.constrained_layout.hspace : 0.02   ## Space between sub
#figure.constrained_layout.wspace : 0.02   ##  a fraction of th

#### IMAGES
#image.aspect : equal             ## equal | auto | a number
#image.interpolation  : nearest   ## see help(imshow) for optio
#image.cmap   : viridis           ## A colormap name, gray etc.
#image.lut    : 256               ## the size of the colormap l
#image.origin : upper             ## lower | upper
#image.resample  : True
#image.composite_image : True     ## When True, all the images
                                  ## combined into a single com
                                  ## saving a figure as a vecto
                                  ## such as a PDF.

#### CONTOUR PLOTS
#contour.negative_linestyle : dashed ## string or on-off ink se
#contour.corner_mask        : True   ## True | False | legacy

#### ERRORBAR PLOTS
#errorbar.capsize : 0             ## length of end cap on error
#### HISTOGRAM PLOTS
#hist.bins : 10                   ## The default number of hist
                                  ## If Numpy 1.11 or later is
                                  ## installed, may also be `au

#### SCATTER PLOTS
#scatter.marker : o               ## The default marker type fo
#scatter.edgecolors : face        ## The default edgecolors for

#### Agg rendering
#### Warning: experimental, 2008/10/10
#agg.path.chunksize : 0           ## 0 to disable; values in th
                                  ## 10000 to 100000 can improv
                                  ## and prevent an Agg rendere
                                  ## when plotting very large d
                                  ## especially if they are ver
                                  ## It may cause minor artifac
                                  ## A value of 20000 is probab
                                  ## starting point.
#### PATHS
#path.simplify : True   ## When True, simplify paths by removin
                        ## points to reduce file size and incre
                        ## speed
#path.simplify_threshold : 0.111111111111  ## The threshold of
                                           ## vertices will be
                                           ## simplification pr
#path.snap : True ## When True, rectilinear axis-aligned paths
                  ## the nearest pixel when certain criteria ar
                  ## paths will never be snapped.
#path.sketch : None ## May be none, or a 3-tuple of the form (s
                    ## randomness).
                    ## *scale* is the amplitude of the wiggle
                    ## perpendicular to the line (in pixels).
                    ## is the length of the wiggle along the li
                    ## pixels).  *randomness* is the factor by
                    ## the length is randomly scaled.
#path.effects : []  ##

#### SAVING FIGURES
## the default savefig params can be different from the display
## e.g., you may want a higher resolution, or to make the figur
## background white
#savefig.dpi         : figure   ## figure dots per inch or 'fig
#savefig.facecolor   : white    ## figure facecolor when saving
#savefig.edgecolor   : white    ## figure edgecolor when saving
#savefig.format      : png      ## png, ps, pdf, svg
#savefig.bbox        : standard ## 'tight' or 'standard'.
                                ## 'tight' is incompatible with
                                ## backends but will workd with
                                ## e.g. setting animation.write
                                ## use ffmpeg_file instead
#savefig.pad_inches  : 0.1      ## Padding to be used when bbox
#savefig.jpeg_quality: 95       ## when a jpeg is saved, the de
#savefig.directory   : ~        ## default directory in savefig
                                ## leave empty to always use cu
#savefig.transparent : False    ## setting that controls whethe
                                ## transparent background by de
#savefig.orientation : portrait        ## Orientation of saved

### tk backend params
#tk.window_focus   : False    ## Maintain shell focus for TkAgg
```

```
### ps backend params
#ps.papersize      : letter   ## auto, letter, legal, ledger, A
#ps.useafm         : False    ## use of afm fonts, results in s
#ps.usedistiller   : False    ## can be: None, ghostscript or x
                                          ## Experimental: may
                                          ## xpdf intended for
                                          ## but requires ghost
#ps.distiller.res  : 6000      ## dpi
#ps.fonttype       : 3         ## Output Type 3 (Type3) or Type

### pdf backend params
#pdf.compression   : 6   ## integer from 0 to 9
                         ## 0 disables compression (good for de
#pdf.fonttype       : 3         ## Output Type 3 (Type3) or Typ
#pdf.use14corefonts : False
#pdf.inheritcolor : False

### svg backend params
#svg.image_inline : True       ## write raster image data direc
#svg.fonttype :   path         ## How to handle SVG fonts:
   ##     none: Assume fonts are installed on the machine where
   ##     path: Embed characters as paths -- supported by most
#svg.hashsalt : None           ## if not None, use this string
                               ## instead of uuid4
### pgf parameter
#pgf.rcfonts : True
#pgf.preamble :            ## see text.latex.preamble for docum
#pgf.texsystem : xelatex

### docstring params
##docstring.hardcopy = False  ## set this when you want to gene

## Event keys to interact with figures/plots via keyboard.
## Customize these settings according to your needs.
## Leave the field(s) empty if you don't need a key-map. (i.e.,
#keymap.fullscreen : f, ctrl+f       ## toggling
#keymap.home : h, r, home            ## home or reset mnemonic
#keymap.back : left, c, backspace, MouseButton.BACK  ## forward
#keymap.forward : right, v, MouseButton.FORWARD      ## for qu
#keymap.pan : p                      ## pan mnemonic
#keymap.zoom : o                     ## zoom mnemonic
#keymap.save : s, ctrl+s             ## saving current figure
#keymap.help : f1                    ## display help about act
#keymap.quit : ctrl+w, cmd+w, q      ## close the current figur
#keymap.quit_all : W, cmd+W, Q       ## close all figures
#keymap.grid : g                     ## switching on/off major
#keymap.grid_minor : G               ## switching on/off minor
#keymap.yscale : l                   ## toggle scaling of y-axe
#keymap.xscale : k, L                ## toggle scaling of x-axe
#keymap.all_axes : a                 ## enable all axes
#keymap.copy : ctrl+c, cmd+c         ## Copy figure to clipboar

###ANIMATION settings
#animation.html :  none            ## How to display the animat
                                   ## the IPython notebook. 'ht
                                   ## HTML5 video tag; 'jshtml
                                   ## Javascript animation
#animation.writer : ffmpeg         ## MovieWriter 'backend' to
#animation.codec : h264            ## Codec to use for writing
#animation.bitrate: -1             ## Controls size/quality tro
                                   ## -1 implies let utility au
#animation.frame_format:  png      ## Controls frame format use
#animation.html_args:              ## Additional arguments to p
```

```
#animation.ffmpeg_path:  ffmpeg    ## Path to ffmpeg binary. Wi
                                   ## $PATH is searched
#animation.ffmpeg_args:            ## Additional arguments to p
#animation.avconv_path:  avconv    ## Path to avconv binary. Wi
                                   ## $PATH is searched
#animation.avconv_args:            ## Additional arguments to p
#animation.convert_path:  convert  ## Path to ImageMagick's con
                                   ## On Windows use the full p
                                   ## is also the name of a sys
#animation.convert_args:           ## Additional arguments to p
#animation.embed_limit : 20.0      ## Limit, in MB, of size of
                                   ## animation in HTML (i.e. I
```

**Download Python source code: customizing.py**

**Download Jupyter notebook: customizing.ipynb**

Last updated on Aug 26, 2019. Created using Sphinx 1.8.5. Doc version v3.1.1-39-gcb432858e.