MATPLOTLIB

In [32]:
```
import matplotlib as plt
import numpy as np
%matplotlib inline
x= np.linspace(1,10,20)
x
```

Out[32]:
```
array([ 1.        ,  1.47368421,  1.94736842,  2.42105263,  2.89473684,
        3.36842105,  3.84210526,  4.31578947,  4.78947368,  5.26315789,
        5.73684211,  6.21052632,  6.68421053,  7.15789474,  7.63157895,
        8.10526316,  8.57894737,  9.05263158,  9.52631579, 10.        ])
```

In [8]:
```
from numpy.random import randint
y=randint(1,50,20)
y
```

Out[8]:
```
array([16, 19, 29, 49, 29, 24, 26, 16, 39, 38, 47, 24, 39, 21, 19,  4, 46,
       27, 10, 48])
```

In [10]:
```
y.size
```

Out[10]: 20

In [25]:
```
plt
```

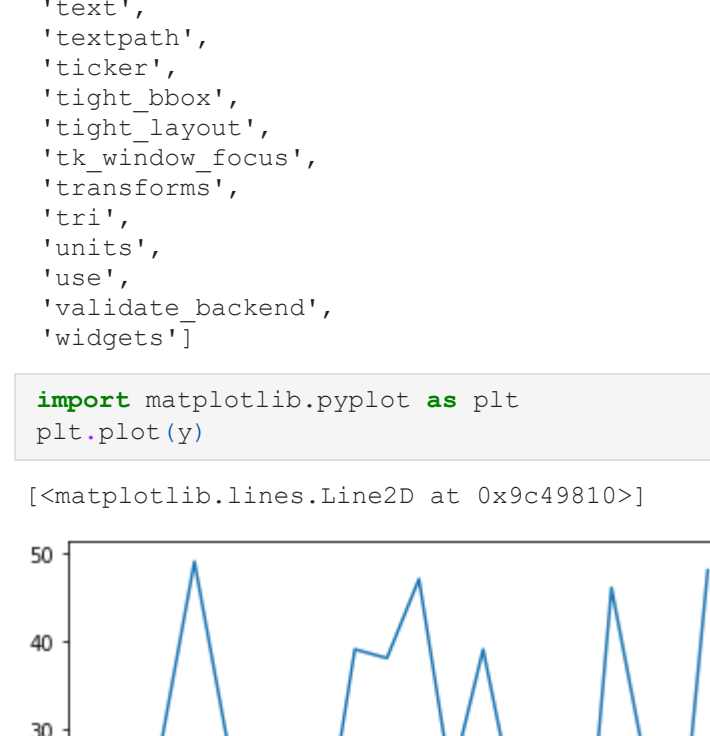
Out[25]: <module 'matplotlib' from 'c:\\program files (x86)\\python37-32\\lib\\site-packages\\matplotlib__init__.py'>

In [23]:
```
dir(plt)
```

Out[23]:
```
['LooseVersion',
 'MatplotlibDeprecationWarning',
 'MutableMapping',
 'Parameter',
 'Path',
 'RcParams',
 'URL_REGEX',
 '_DATA_DOC_APPENDIX',
 '_DATA_DOC_TITLE',
 '_ExecInfo',
 '__bibtex__',
 '__builtins__',
 '__cached__',
 '__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__path__',
 '__spec__',
 '__version__',
 '_add_data_doc',
 '_all_deprecated',
 '_check_versions',
 '_cm',
 '_cm_listed',
 '_color_data',
 '_constrained_layout',
 '_create_tmp_config_or_cache_dir',
 '_deprecated_ignore_map',
 '_deprecated_map',
 '_deprecated_remain_as_none',
 '_ensure_handler',
 '_error_details_fmt',
 '_get_config_or_cache_dir',
 '_get_data_path',
 '_get_executable_info',
 '_get_xdg_cache_dir',
 '_get_xdg_config_dir',
 '_image',
 '_init_tests',
 '_label_from_arg',
 '_layoutbox',
 '_log',
 '_logged_cached',
 '_mathtext_data',
 '_open_file_or_url',
 '_path',
 '_preprocess_data',
 '_pylab_helpers',
 '_rc_params_in_file',
 '_replacer',
 '_version',
 'afm',
 'artist',
 'atexit',
 'axes',
 'axis',
 'backend_bases',
 'backend_tools',
 'backends',
 'bezier',
 'blocking_input',
 'category',
 'cbook',
 'checkdep_dvipng',
 'checkdep_ghostscript',
 'checkdep_inkscape',
 'checkdep_pdftops',
 'checkdep_ps_distiller',
 'checkdep_usetex',
 'cm',
 'collections',
 'colorbar',
 'colors',
 'compare_versions',
 'container',
 'contextlib',
 'contour',
 'cycler',
 'dates',
 'dedent',
 'defaultParams',
 'default_test_modules',
 'docstring',
 'dviread',
 'figure',
 'font_manager',
 'fontconfig_pattern',
 'ft2font',
 'functools',
 'get_backend',
 'get_cachedir',
 'get_configdir',
 'get_data_path',
 'get_home',
 'get_label',
 'get_py2exe_datafiles',
 'gridspec',
 'image',
 'importlib',
 'inspect',
 'interactive',
 'is_interactive',
 'is_url',
 'legend',
 'legend_handler',
 'lines',
 'locale',
 'logging',
 'markers',
 'mathtext',
 'matplotlib_fname',
 'mlab',
 'mplDeprecation',
 'namedtuple',
 'numpy',
 'offsetbox',
 'os',
 'patches',
 'path',
 'pprint',
 'projections',
 'pyplot',
 'quiver',
 'rc',
 'rcParams',
 'rcParamsDefault',
 'rcParamsOrig',
 'rc_context',
 'rc_file',
 'rc_file_defaults',
 'rc_params',
 'rc_params_from_file',
 'rcdefaults',
 'rcsetup',
 're',
 'sanitize_sequence',
 'scale',
 'set_loglevel',
 'shutil',
 'spines',
 'stackplot',
 'streamplot',
 'style',
 'subprocess',
 'sys',
 'table',
 'tempfile',
 'test',
 'texmanager',
 'text',
 'textpath',
 'ticker',
 'tight_bbox',
 'tight_layout',
 'tk_window_focus',
 'transforms',
 'tri',
 'units',
 'use',
 'validate_backend',
 'widgets']
```

In [27]:
```
import matplotlib.pyplot as plt
plt.plot(y)
```

Out[27]: [<matplotlib.lines.Line2D at 0x9c49810>]

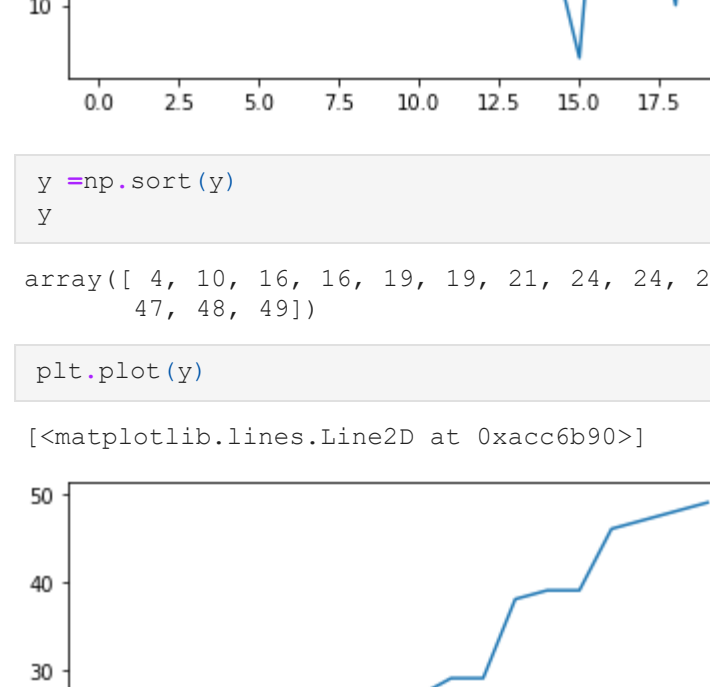


In [29]:
```
y =np.sort(y)
y
```

Out[29]:
```
array([ 4, 10, 16, 16, 19, 19, 21, 24, 24, 26, 27, 29, 29, 38, 39, 39, 46,
       47, 48, 49])
```

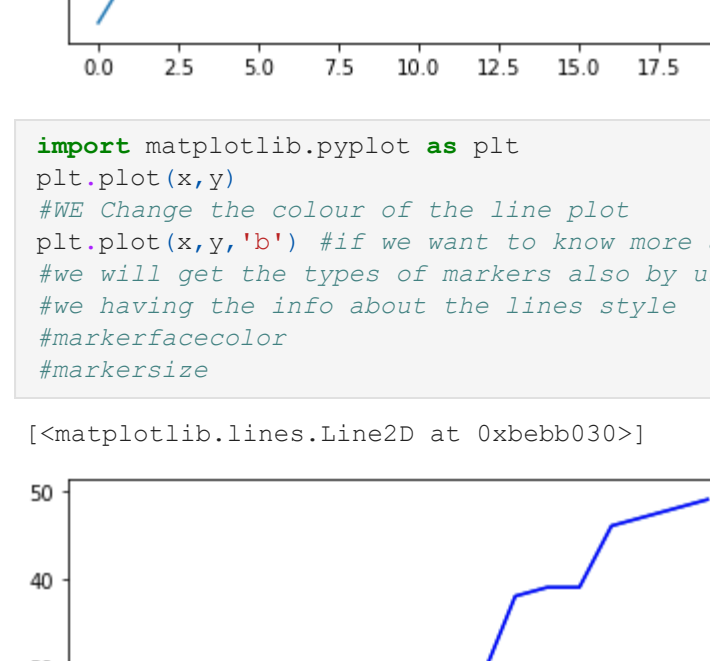
In [30]:
```
plt.plot(y)
```

Out[30]: [<matplotlib.lines.Line2D at 0xacc6b90>]



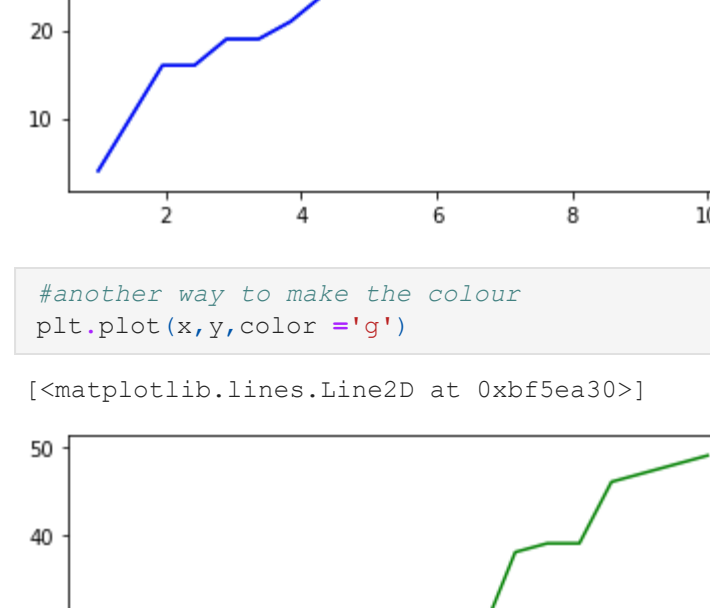
In [42]:
```
import matplotlib.pyplot as plt
plt.plot(x,y)
#WE Change the colour of the line plot
plt.plot(x,y,'b') #if we want to know more about it(colour) then we have to press (shift + tab)
#we will get the types of markers also by using the same (shift + tab)
#we having the info about the lines style
#markerfacecolor
#markersize
```

Out[42]: [<matplotlib.lines.Line2D at 0xbebb030>]

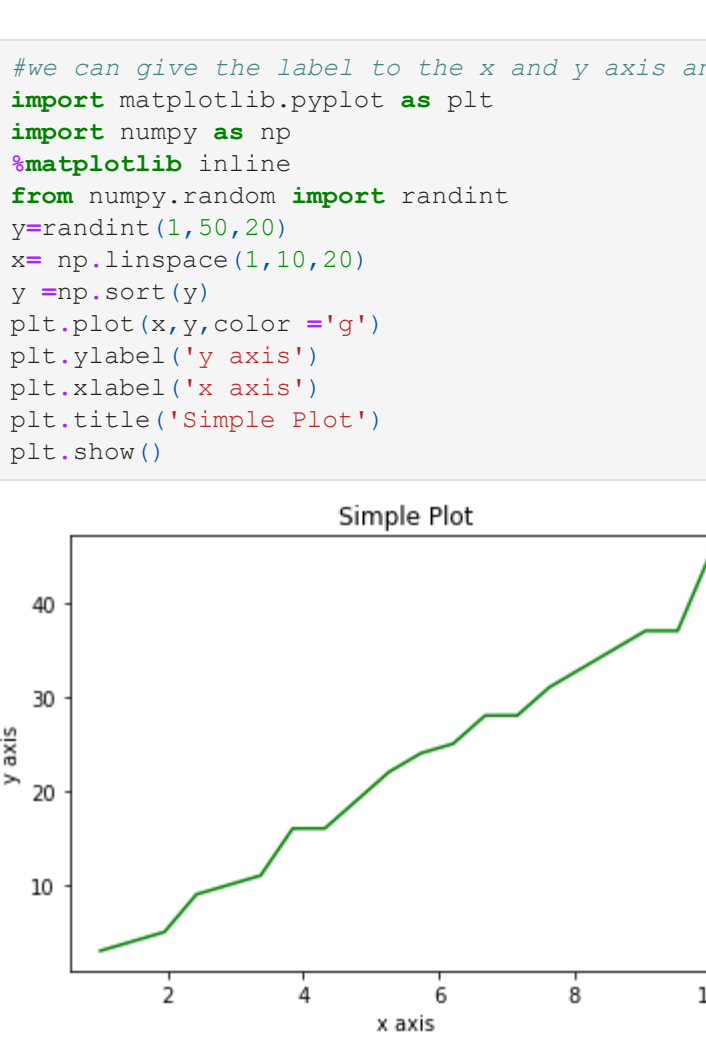


In [46]:
```
#another way to make the colour
plt.plot(x,y,color ='g')
```

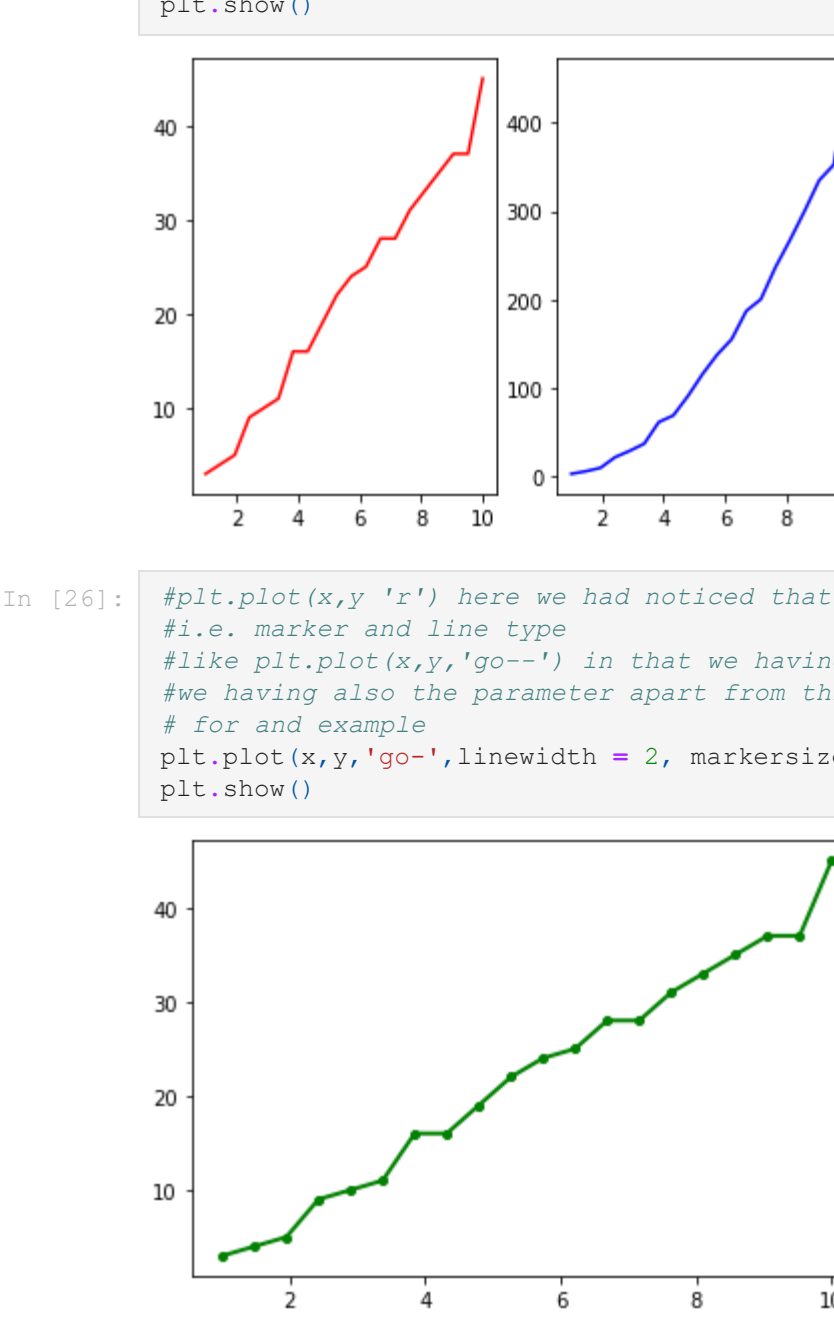
Out[46]: [<matplotlib.lines.Line2D at 0xbf5ea30>]



In [7]:
```
#we can give the label to the x and y axis and we can give the title to our plot
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
from numpy.random import randint
y=randint(1,50,20)
x= np.linspace(1,10,20)
y =np.sort(y)
plt.plot(x,y,color ='g')
plt.ylabel('y axis')
plt.xlabel('x axis')
plt.title('Simple Plot')
plt.show()
```

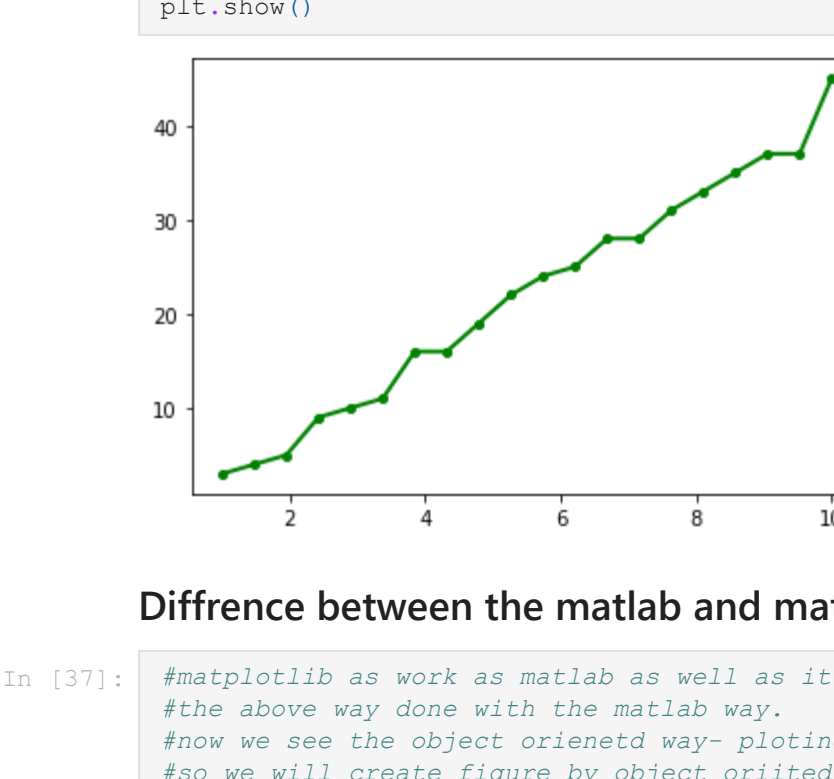


In [13]:
```
#so how use the multiple plots on the same canvas
plt.plot()
#in that plt we having single canvas and by default it having the single axis
plt.subplot(1,2,1)#here we having the subplot command in that we are referring the 1 row and 2 columns and refe
plt.plot(x,y,'r')
plt.subplot(1,2,2)#here we having the subplot command in that we are referring the 1 row and 2 columns and refe
plt.plot(x,y,'b')
plt.show()
```

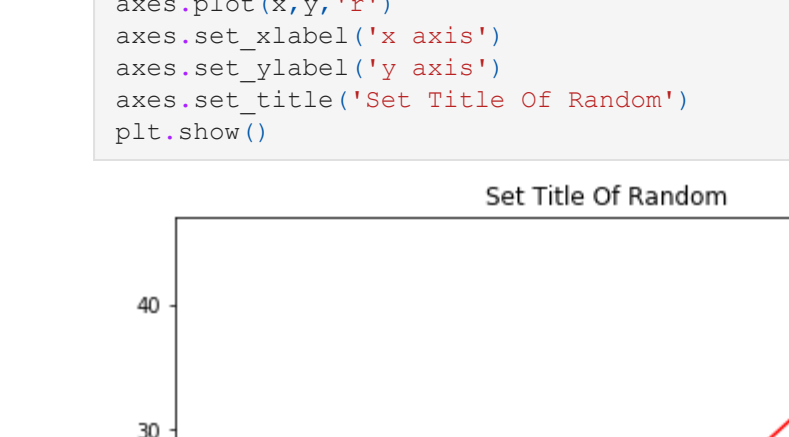


In [ ]:
```
#this is about the subplot(multiple plots on the same canvas) on the same canvas
```

In [15]:
```
y1 = y**x
plt.subplot(1,2,1)
plt.plot(x,y,'r')
plt.subplot(1,2,2)
plt.plot(x,y1,'b')
plt.show()
```

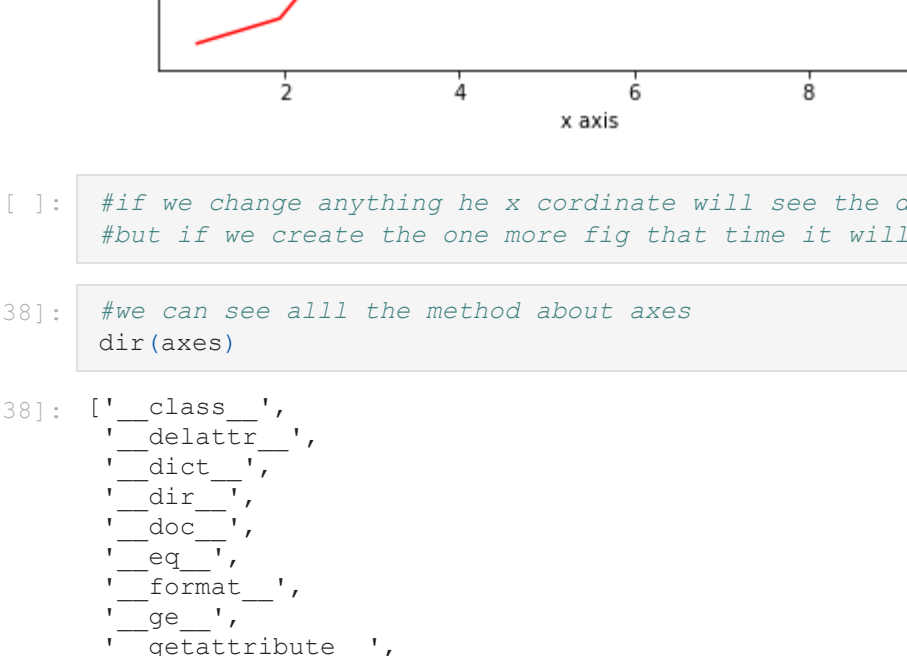


In [28]:
```
#plt.plot(x,y,'r') here we had noticed that there is data that is x and y and then colour and then we have als
#i.e. marker and line type
#like plt.plot(x,y,'go--') in that we are having the marker that is o and -- is line type
#we having also the parameter apart from this one i.e.linewidth and markersize in the close bracket
# for ex example
plt.plot(x,y,'go--',linewidth = 2, markersize = 6)
plt.show()
```



Diffrence between the matlab and matplotlib

In [59]:
```
#matplotlib as work as matlab as well as it works as objet oriented language or method,its called to figure obj
#the above way done with the matlab way.
#now we see the object oriented way: plotting figure and we will get the control of statics of that fig.
#so we will create figure by object orited way
fig= plt.figure()
axes = fig.add_axes([0.1,0.1,1,1])
#here x = 0.1,y=0.1 and it will go upto the 1 and 1
#left bottom corner treated as 0.1 corner and top right hand corner treated as 1.
#if we apply the method on it like plot
axes.plot(x,y,'r')
axes.set_xlabel('x axis')
axes.set_ylabel('y axis')
axes.set_title('Set Title Of Random')
```

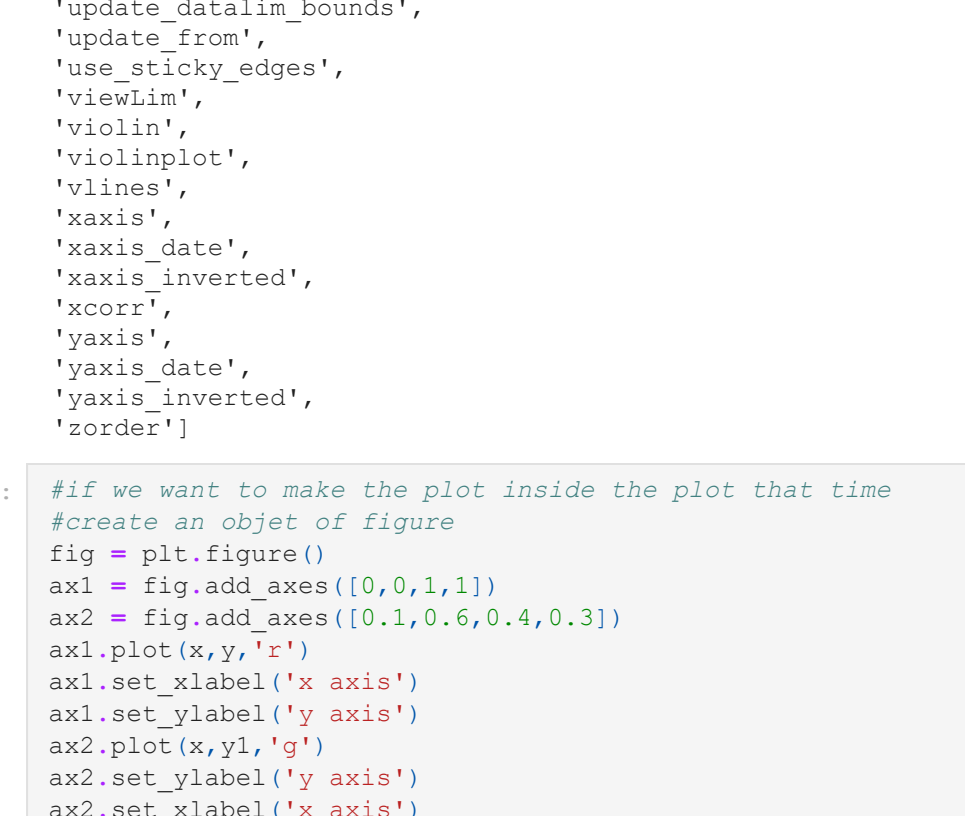


In [ ]:
```
#if we change anything he a coordinate will see the diffrence in the fig
#but if we create the one more fig than time it will show the diffrence.
```

In [38]:
```
#we can see all the method above axes
dir(axes)
```

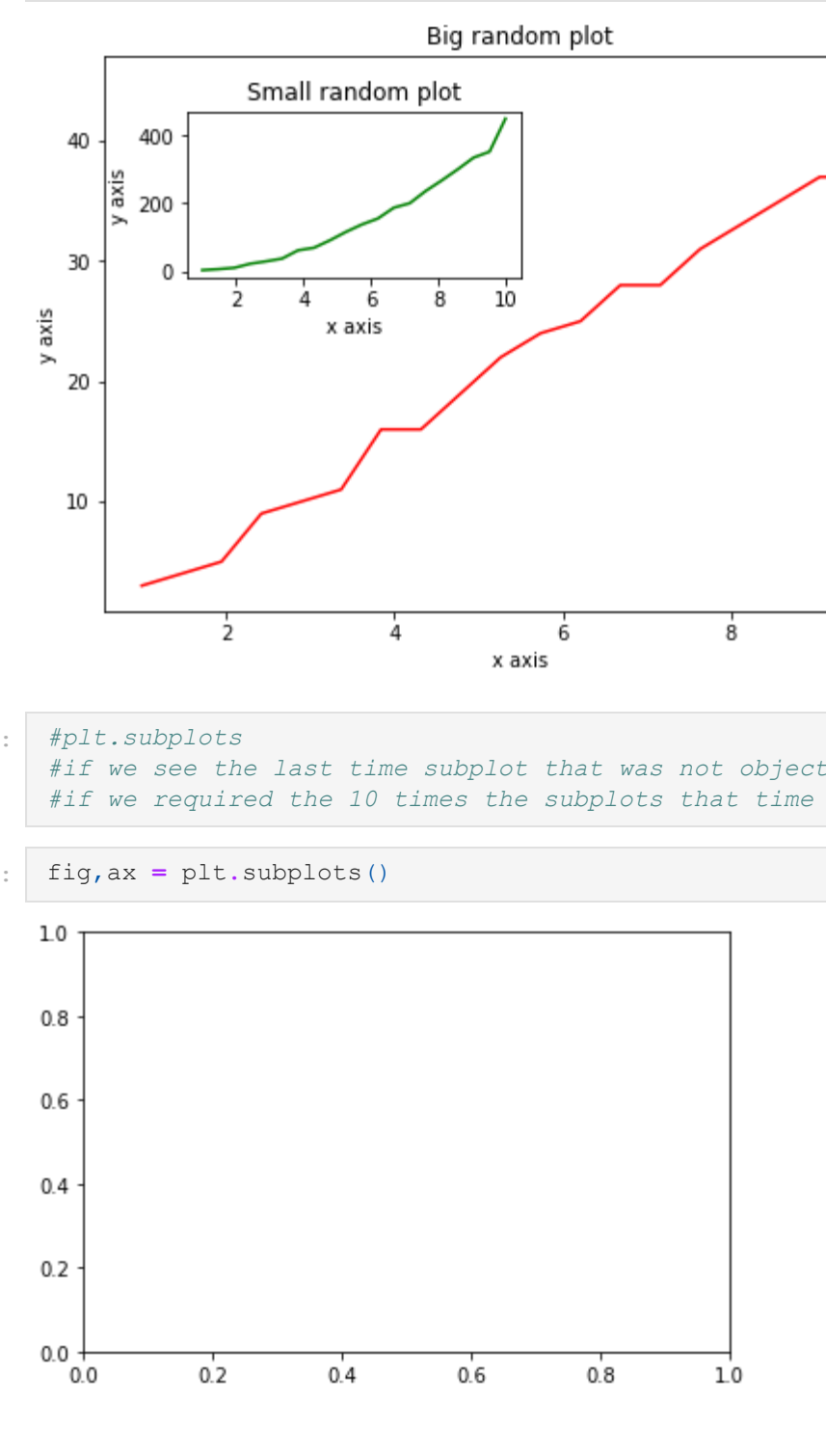
Out[38]:
```
['__class__',
 '__delattr__',
 '__dict__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getstate__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__le__',
 '__lt__',
 '__module__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__setstate__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 '__weakref__',
 '_add_text',
 '_adjustable',
 '_agg_filter',
 '_alpha',
 '_anchor',
 '_animated',
 '_aspect',
 '_autoscaleXon',
 '_autoscaleYon',
 '_axes',
 '_axes_locator',
 '_axisbelow',
 '_clipon',
 '_clippath',
 '_connected',
 '_contains',
 '_convert_dx',
 '_current_image',
 '_facecolor',
 '_frameon',
 '_gci',
 '_gen_axes_patch',
 '_gen_axes_spines',
 '_get_axis_list',
 '_get_clipping_extent_bbox',
 '_get_lines',
 '_get_patches_for_fill',
 '_get_view',
 '_gid',
 '_gridOn',
 '_in_layout',
 '_init_axis',
 '_label',
 '_layoutbox',
 '_left_title',
 '_make_twin_axes',
 '_mouseover',
 '_mouseover_set',
 '_navigate',
 '_navigate_mode',
 '_oid',
 '_on_units_changed',
 '_originalPosition',
 '_parse_scatter_color_args',
 '_path_effects',
 '_pcolorargs',
 '_picker',
 '_position',
 '_poslayoutbox',
 '_process_unit_info',
 '_prop_order',
 '_propobservers',
 '_quiver_units',
 '_rasterization_zorder',
 '_rasterized',
 '_remove_legend',
 '_remove_method',
 '_right_title',
 '_sci',
 '_set_artist_props',
 '_set_gc_clip',
 '_set_lim_and_transforms',
 '_set_position',
 '_set_title_offset_trans',
 '_set_view',
 '_set_view_from_bbox',
 '_shared_x_axes',
 '_shared_y_axes',
 '_sharex',
 '_sharey',
 '_sketch',
 '_snap',
 '_stale',
 '_sticky_edges',
 '_tight',
 '_transform',
 '_transformSet',
 '_twinned_axes',
 '_update_image_limits',
 '_update_line_limits',
 '_update_patch_limits',
 '_update_title_position',
 '_update_transScale',
 '_url',
 '_use_sticky_edges',
 '_validate_converted_limits',
 '_visible',
 '_xaxis_transform',
 '_xcid',
 '_xmargin',
 '_yaxis_transform',
 '_ycid',
 '_ymargin',
 'acorr',
 'add_artist',
 'add_callback',
 'add_child_axes',
 'add_collection',
 'add_container',
 'add_image',
 'add_line',
 'add_patch',
 'add_table',
 'aname',
 'angle_spectrum',
 'annotate',
 'apply_aspect',
 'arrow',
 'artists',
 'autoscale',
 'autoscale_view',
 'axes',
 'axhline',
 'axhspan',
 'axis',
 'axison',
 'axvline',
 'axvspan',
 'bar',
 'barbs',
 'barh',
 'bbox',
 'boxplot',
 'broken_barh',
 'bxp',
 'callbacks',
 'can_pan',
 'can_zoom',
 'child_axes',
 'cla',
 'clabel',
 'clear',
 'clipbox',
 'cohere',
 'collections',
 'containers',
 'contains',
 'contains_point',
 'contour',
 'contourf',
 'convert_xunits',
 'convert_yunits',
 'csd',
 'dataLim',
 'drag_pan',
 'draw',
 'draw_artist',
 'end_pan',
 'errorbar',
 'eventplot',
 'eventson',
 'figure',
 'fill',
 'fill_between',
 'fill_betweenx',
 'findobj',
 'fmt_xdata',
 'fmt_ydata',
 'format_coord',
 'format_cursor_data',
 'format_xdata',
 'format_ydata',
 'get_adjustable',
 'get_agg_filter',
 'get_alpha',
 'get_anchor',
 'get_animated',
 'get_aspect',
 'get_autoscale_on',
 'get_autoscalex_on',
 'get_autoscaley_on',
 'get_axes_locator',
 'get_axisbelow',
 'get_children',
 'get_clip_box',
 'get_clip_on',
 'get_clip_path',
 'get_contains',
 'get_cursor_data',
 'get_data_ratio',
 'get_data_ratio_log',
 'get_default_bbox_extra_artists',
 'get_facecolor',
 'get_fc',
 'get_figure',
 'get_frame_on',
 'get_gid',
 'get_images',
 'get_in_layout',
 'get_label',
 'get_legend',
 'get_legend_handles_labels',
 'get_lines',
 'get_navigate',
 'get_navigate_mode',
 'get_path_effects',
 'get_picker',
 'get_position',
 'get_rasterization_zorder',
 'get_rasterized',
 'get_renderer_cache',
 'get_shared_x_axes',
 'get_shared_y_axes',
 'get_sketch_params',
 'get_snap',
 'get_tightbbox',
 'get_title',
 'get_transform',
 'get_transformed_clip_path_and_affine',
 'get_url',
 'get_visible',
 'get_window_extent',
 'get_xaxis',
 'get_xaxis_text1_transform',
 'get_xaxis_text2_transform',
 'get_xaxis_transform',
 'get_xbound',
 'get_xgridlines',
 'get_xlabel',
 'get_xlim',
 'get_xmajorticklabels',
 'get_xminorticklabels',
 'get_xscale',
 'get_xticklabels',
 'get_xticklines',
 'get_xticks',
 'get_yaxis',
 'get_yaxis_text1_transform',
 'get_yaxis_text2_transform',
 'get_yaxis_transform',
 'get_ybound',
 'get_ygridlines',
 'get_ylabel',
 'get_ylim',
 'get_ymajorticklabels',
 'get_yminorticklabels',
 'get_yscale',
 'get_yticklabels',
 'get_yticklines',
 'get_yticks',
 'get_zorder',
 'grid',
 'has_data',
 'have_units',
 'hexbin',
 'hist',
 'hist2d',
 'hlines',
 'ignore_existing_data_limits',
 'images',
 'imshow',
 'in_axes',
 'indicate_inset',
 'indicate_inset_zoom',
 'inset_axes',
 'invert_xaxis',
 'invert_yaxis',
 'is_transform_set',
 'legend',
 'legend_',
 'lines',
 'locator_params',
 'loglog',
 'magnitude_spectrum',
 'margins',
 'matshow',
 'minorticks_off',
 'minorticks_on',
 'mouseover',
 'mouseover_set',
 'name',
 'patch',
 'patches',
 'pcolor',
 'pcolorfast',
 'pcolormesh',
 'phase_spectrum',
 'pick',
 'pickable',
 'pie',
 'plot',
 'plot_date',
 'properties',
 'psd',
 'quiver',
 'quiverkey',
 'redraw_in_frame',
 'relim',
 'remove',
 'remove_callback',
 'reset_position',
 'scatter',
 'secondary_xaxis',
 'secondary_yaxis',
 'semilogx',
 'semilogy',
 'set',
 'set_adjustable',
 'set_agg_filter',
 'set_alpha',
 'set_anchor',
 'set_animated',
 'set_aspect',
 'set_autoscale_on',
 'set_autoscalex_on',
 'set_autoscaley_on',
 'set_axes_locator',
 'set_axis_off',
 'set_axis_on',
 'set_axisbelow',
 'set_clip_box',
 'set_clip_on',
 'set_clip_path',
 'set_contains',
 'set_facecolor',
 'set_fc',
 'set_figure',
 'set_frame_on',
 'set_gid',
 'set_in_layout',
 'set_label',
 'set_navigate',
 'set_navigate_mode',
 'set_path_effects',
 'set_picker',
 'set_position',
 'set_prop_cycle',
 'set_rasterization_zorder',
 'set_rasterized',
 'set_sketch_params',
 'set_snap',
 'set_title',
 'set_transform',
 'set_url',
 'set_visible',
 'set_xbound',
 'set_xlabel',
 'set_xlim',
 'set_xmargin',
 'set_xscale',
 'set_xticklabels',
 'set_xticks',
 'set_ybound',
 'set_ylabel',
 'set_ylim',
 'set_ymargin',
 'set_yscale',
 'set_yticklabels',
 'set_yticks',
 'set_zorder',
 'specgram',
 'spines',
 'spy',
 'stackplot',
 'stale',
 'stale_callback',
 'start_pan',
 'stem',
 'step',
 'sticky_edges',
 'streamplot',
 'table',
 'tables',
 'text',
 'texts',
 'tick_params',
 'ticklabel_format',
 'title',
 'titleOffsetTrans',
 'transAxes',
 'transData',
 'transLimits',
 'transScale',
 'tricontour',
 'tricontourf',
 'tripcolor',
 'triplot',
 'twinx',
 'twiny',
 'update',
 'update_datalim',
 'update_datalim_bounds',
 'update_from',
 'use_sticky_edges',
 'viewLim',
 'violin',
 'violinplot',
 'vlines',
 'xaxis',
 'xaxis_date',
 'xaxis_inverted',
 'xcorr',
 'yaxis',
 'yaxis_date',
 'yaxis_inverted',
 'zorder']
```

In [49]:
```
#if we want to make the plot inside the plot that time
#create as objet of figure
fig = plt.figure()
ax1 = fig.add_axes([0,0,1,1])
ax2 = fig.add_axes([0.1,0.6,0.4,0.3])
ax1.plot(x,y,'r')
ax1.set_xlabel('x axis')
ax1.set_ylabel('y axis')
ax2.plot(x,y1,'g')
ax2.set_ylabel('y axis')
ax2.set_xlabel('x axis')
ax1.set_title('Big random plot')
ax2.set_title('Small random plot')
plt.show()
```

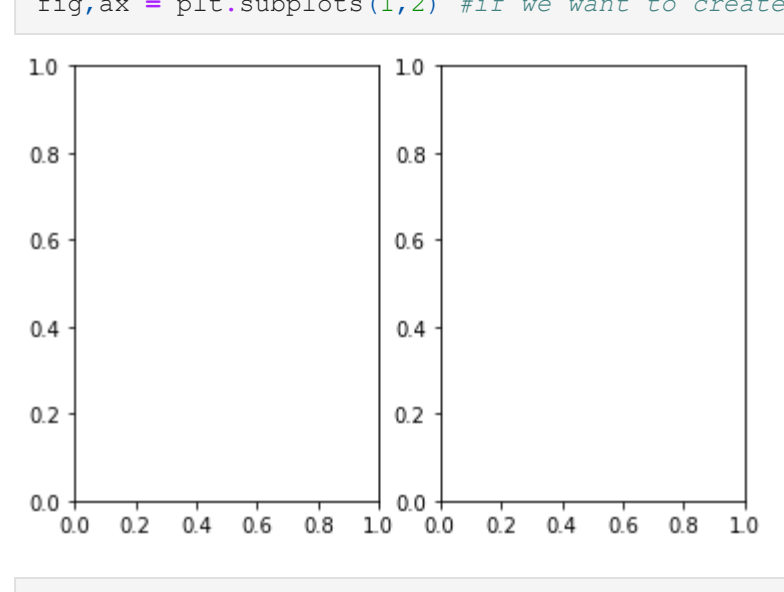


In [ ]:
```
#plt.subplots
#if we see the last time subplot that was not object oriented and also it wasn't gives the full control over th
#if we required the 10 times the subplots that time we have write the same command for the 10 times and it beco
```

In [51]:
```
fig,ax = plt.subplots()
```



In [52]:
```
fig,ax = plt.subplots(1,2) #if we want to create the 2 subplots in one canvas so for that we have to use the i
```

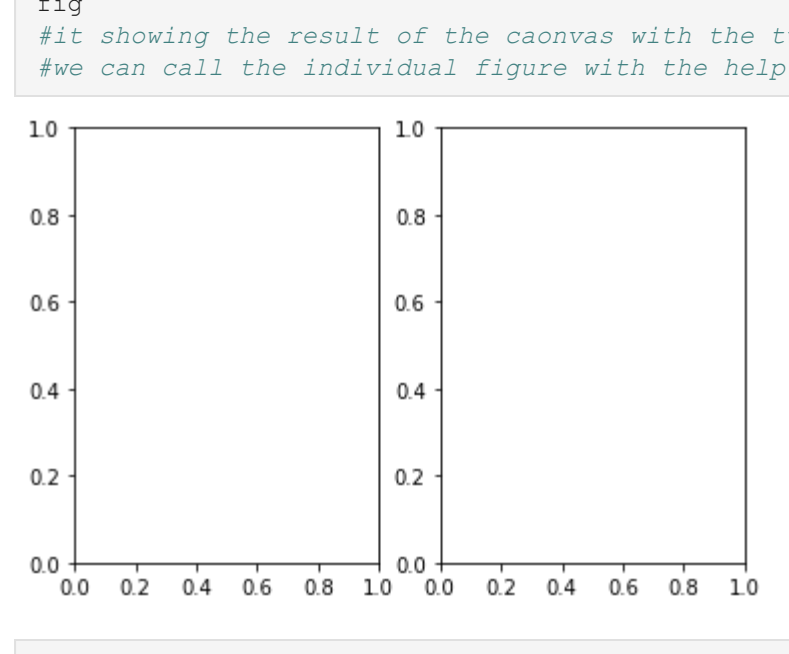


In [53]:
```
#if see here about an axis
ax
#here we got the output as axis= 0 and axis= 1
#that means we having the two axis for the two plots.
```

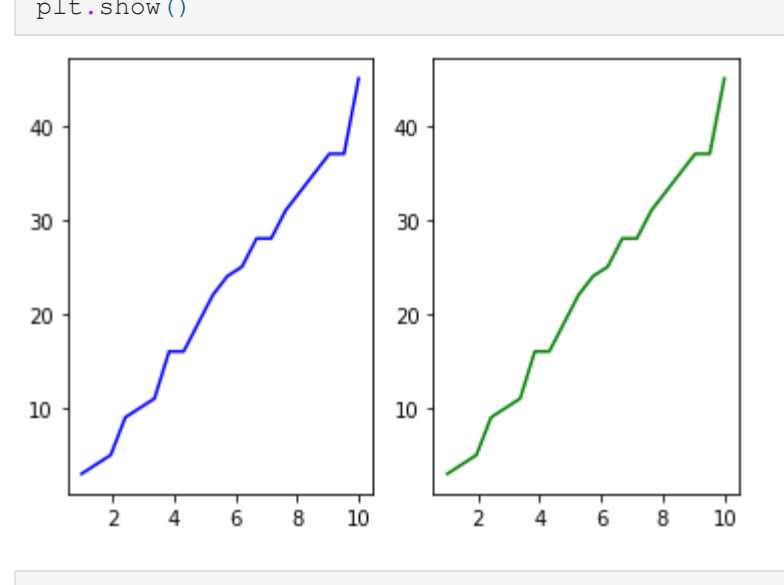
Out[53]:
```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x0C6E6F90>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x0C7AB770>],
      dtype=object)
```

In [54]:
```
#if we see the figure by using the
fig
#it showing the result of the canvas with the two figures.
#we can call the individual figure with the help of the that is ax[0] and ax[1]
```

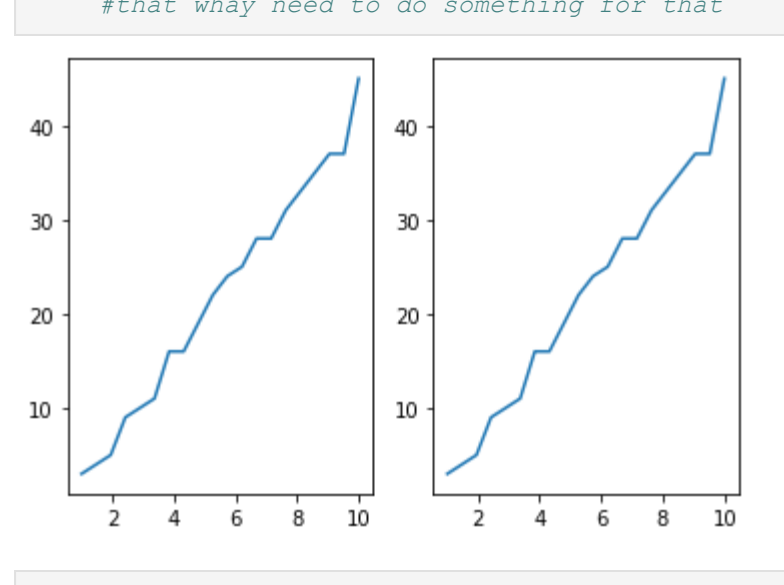
Out[54]:



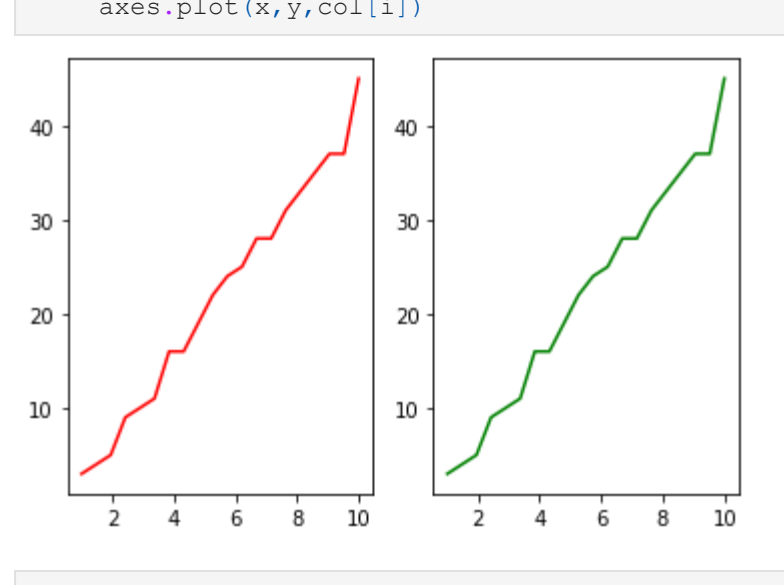
In [60]:
```
#cont force way.
fig,ax = plt.subplots(1,2)
ax[0].plot(x,y,'b')
ax[1].plot(x,y,'g')
plt.show()
```



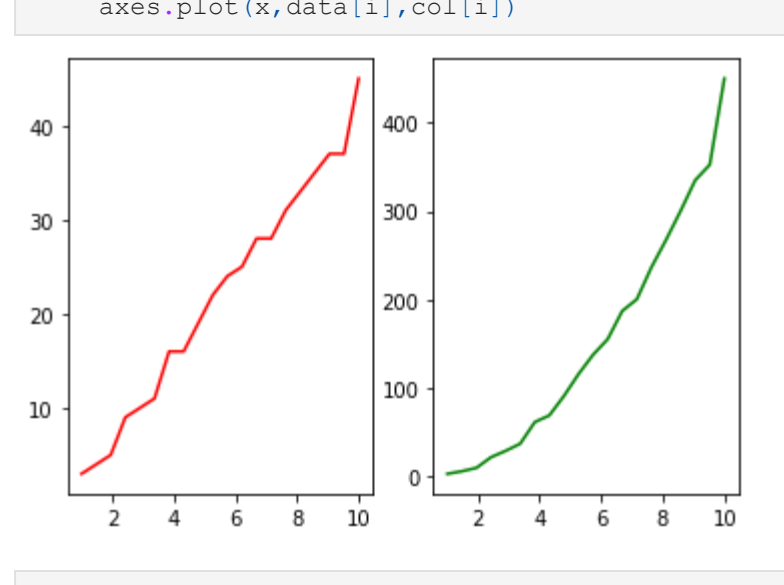
In [64]:
```
#for loop way
fig,ax = plt.subplots(1,2)
for axes in ax:
    axes.plot(x,y)
    #here if we tried to get the new colour that time we both colour will change
    #that whay need to do something for that
```



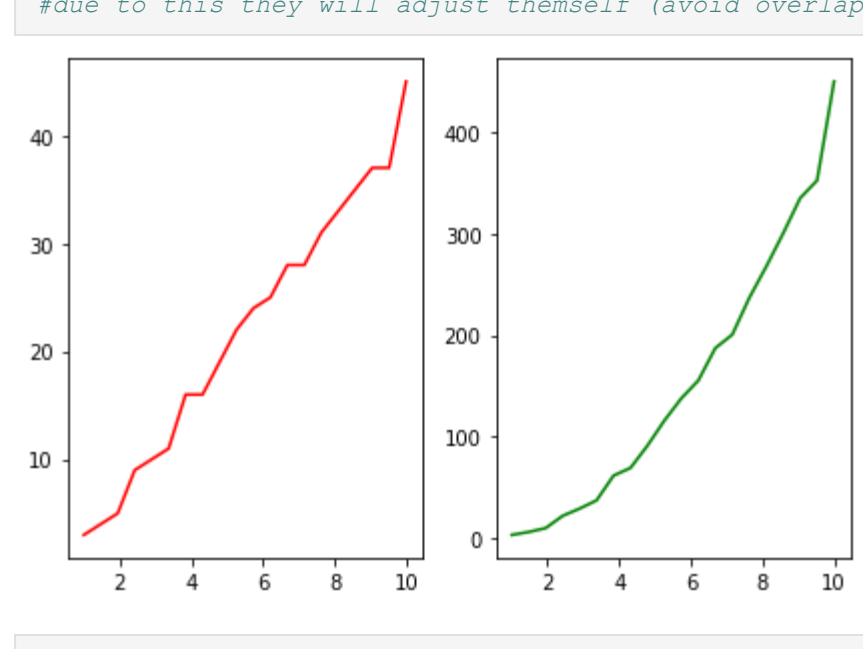
In [68]:
```
col=['r','g']
fig,ax = plt.subplots(1,2)
for i,axes in enumerate(ax):
    axes.plot(x,y,col[i])
```



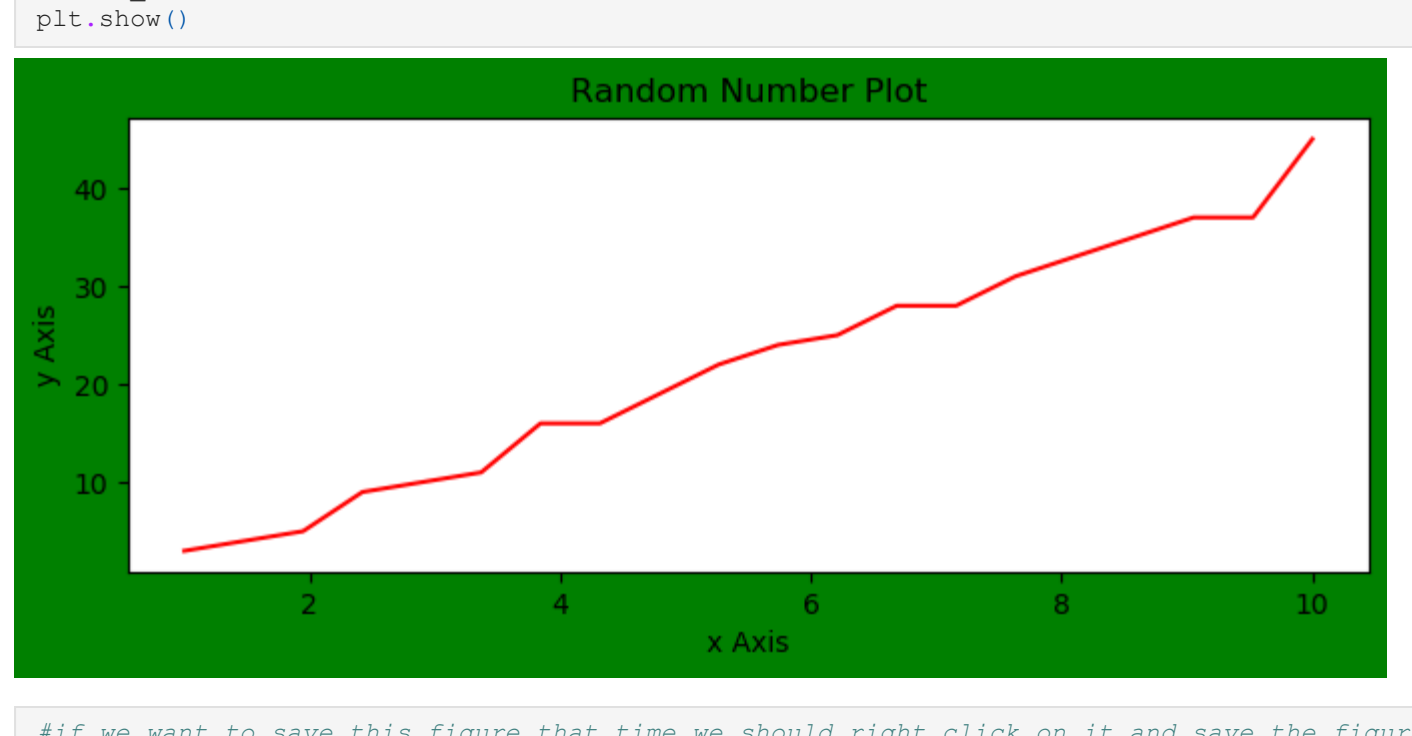
In [70]:
```
#if we want to pass some data that time
col=['r','g']
data=[y,y1]
fig,ax = plt.subplots(1,2)
for i,axes in enumerate(ax):
    axes.plot(x,data[i],col[i])
```



In [71]:
```
#if we want to pass some data that time
col=['r','g']
data=[y,y1]
fig,ax = plt.subplots(1,2)
for i,axes in enumerate(ax):
    axes.plot(x,data[i],col[i])
fig.tight_layout()
#due to this they will adjust themself (avoid overlap of any text and another things)
```



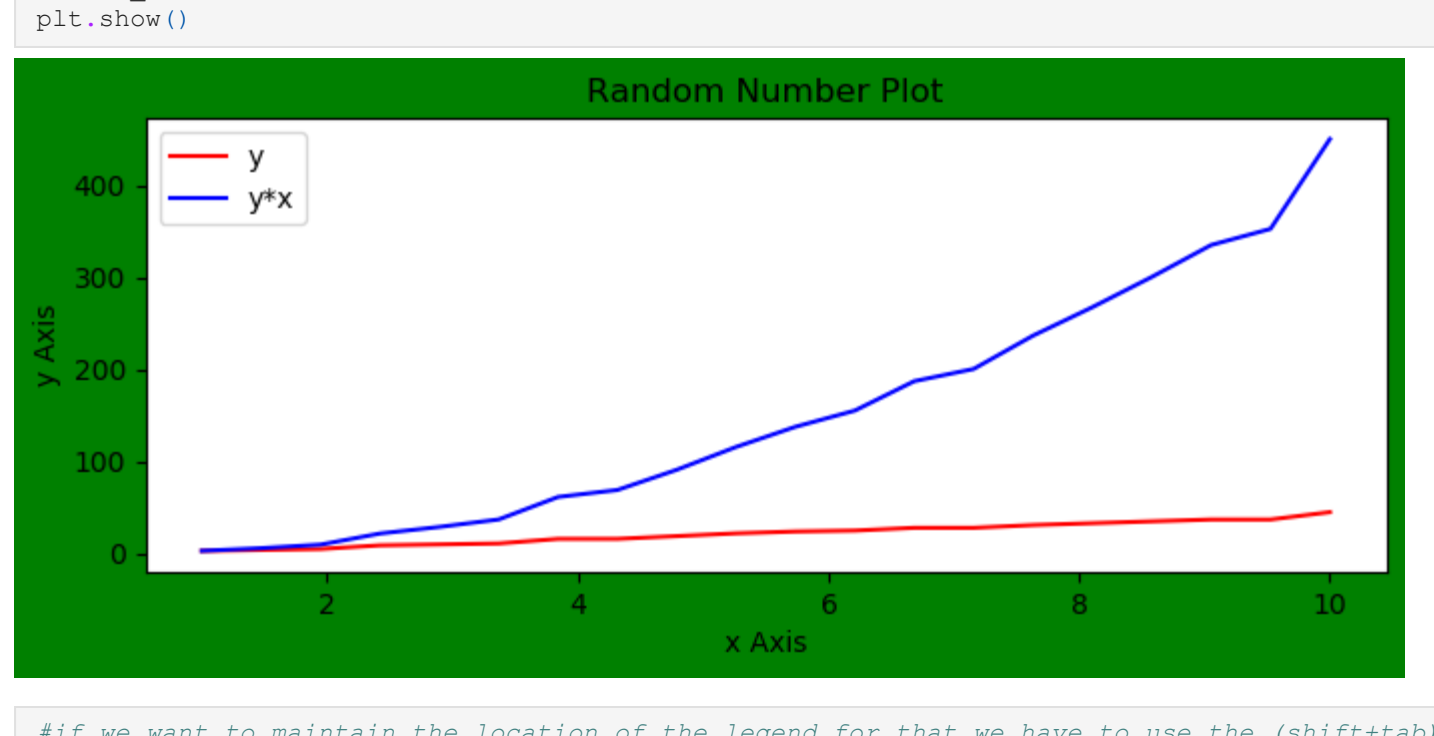
In [89]:
```
#we chane the figuresize by using figuresize function and dpi (dots per inch)
fig,ax= plt.subplots(figsize= (8,3),dpi = 100,facecolor ='g')
ax.plot(x,y,'r')
ax.set_xlabel('x Axis')
ax.set_ylabel('y Axis')
ax.set_title('Random Number Plot')
plt.show()
```



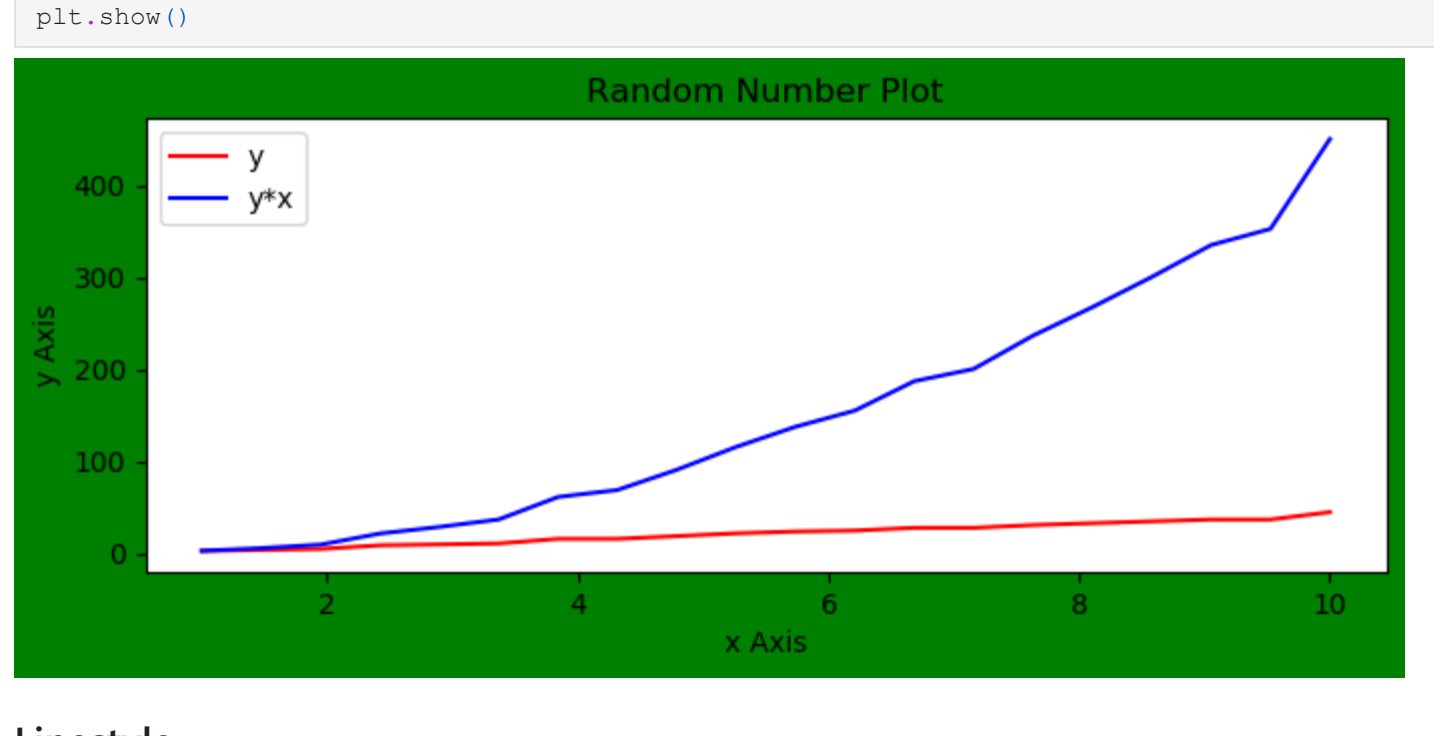
In [91]:
```
#if we want to save this figure that time we should right click on it and save the figure.
#another way that is
fig.savefig('random plot.png') #here we can get the more detail by pressing the(shift + lab in the braket) we c
```

In [ ]:
```
#legend
#this is used to recognization of the line or plot which had drawn on the figure.
```

In [110]:
```
fig,ax= plt.subplots(figsize= (8,3),dpi = 100,facecolor ='g')
ax.plot(x,y,'r',label= 'y')#we can use in this way
ax.plot(x,y1,'b',label='y*x') #or in this way also
ax.set_xlabel('x Axis')
ax.set_ylabel('y Axis')
ax.set_title('Random Number Plot')
ax.legend()
plt.show()
```



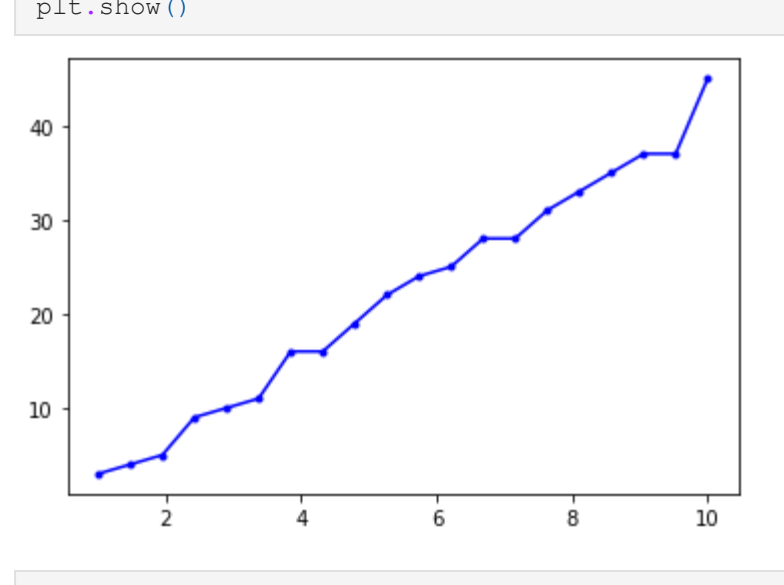
In [124]:
```
#if we want to maintain the location of the legend for that we have to use the (shift+tab) in the ax.legend() t
fig,ax= plt.subplots(figsize= (8,3),dpi = 100,facecolor ='g')
ax.plot(x,y,'r',label= 'y')#we can use in this way
ax.plot(x,y1,'b',label='y*x') #or in this way also
ax.set_xlabel('x Axis')
ax.set_ylabel('y Axis')
ax.legend(loc = 0)
ax.set_title('Random Number Plot')
plt.show()
```



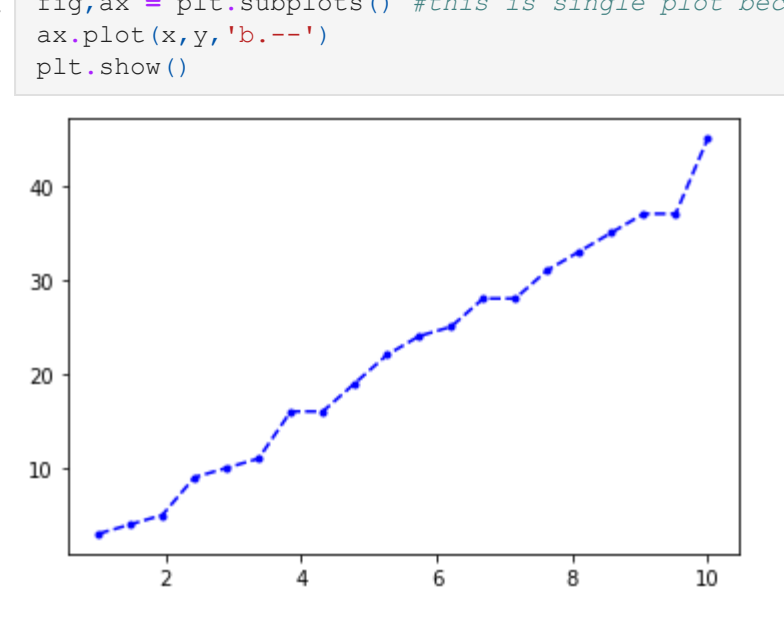
Linestyle

In [ ]:
```
#see how can set the color, linewidth and linetype
```

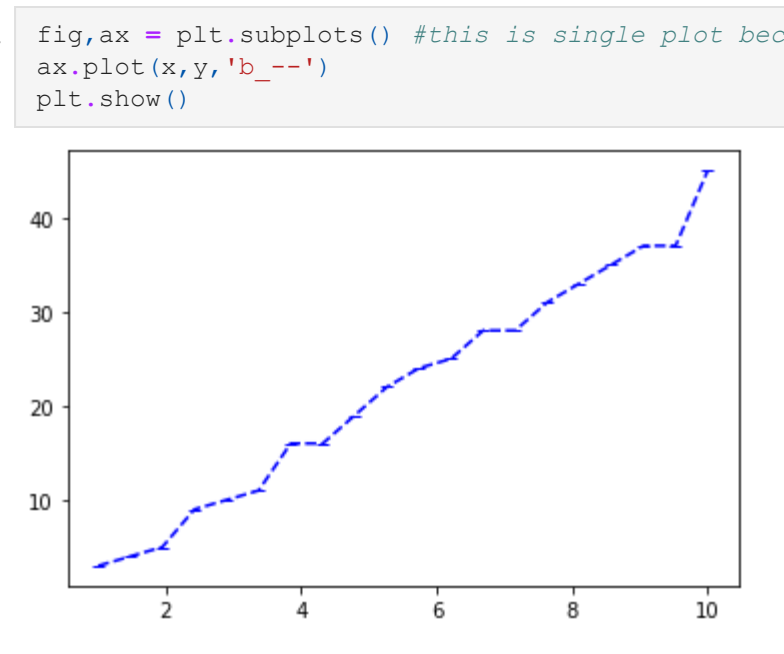
In [127]:
```
fig,ax = plt.subplots() #this is single plot because we write anything in the bracket about about how many plo
ax.plot(x,y,'b.-')
plt.show()
```



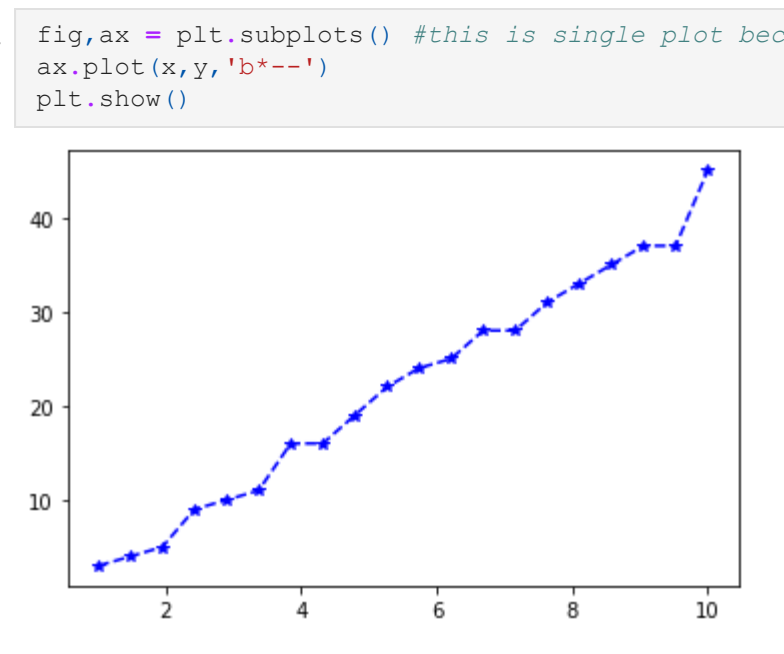
In [128]:
```
fig,ax = plt.subplots() #this is single plot because we write anything in the bracket about about how many plo
ax.plot(x,y,'b.--')
plt.show()
```



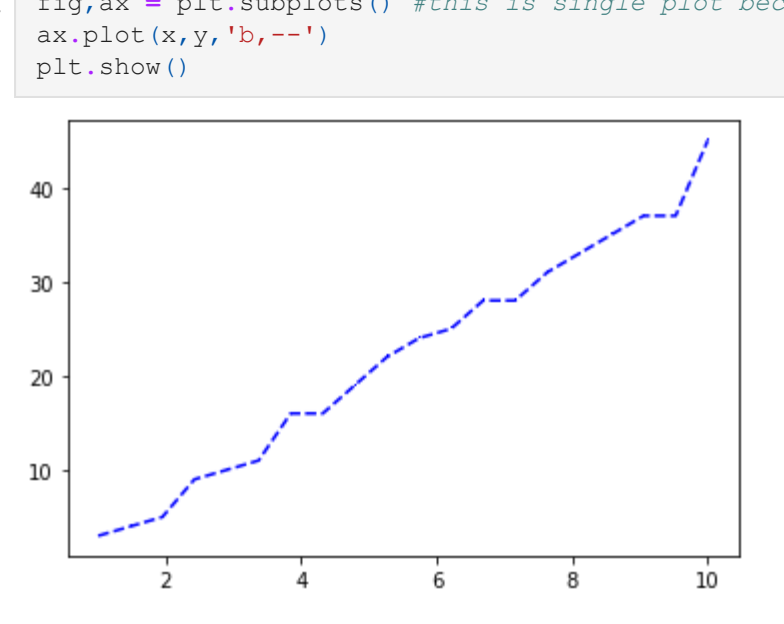
In [130]:
```
fig,ax = plt.subplots() #this is single plot because we write anything in the bracket about about how many plo
ax.plot(x,y,'b_--')
plt.show()
```



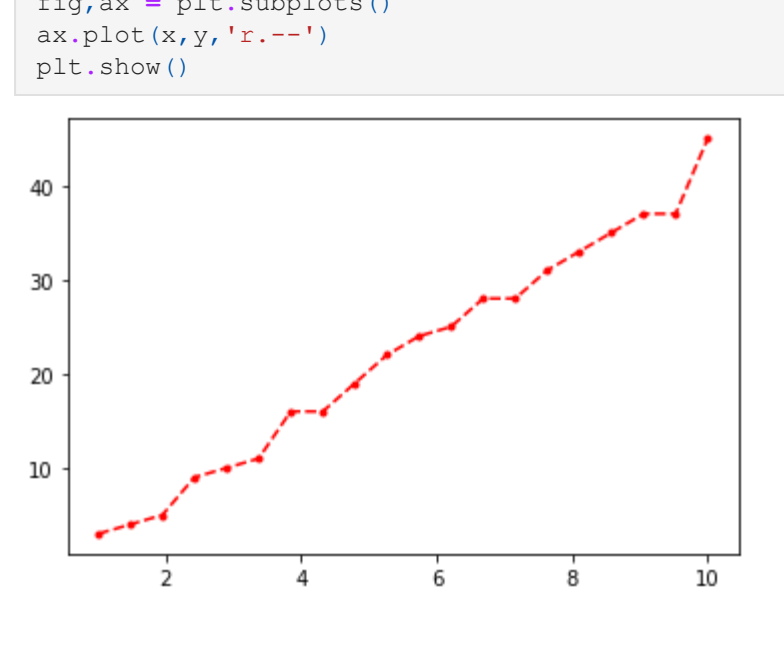
In [131]:
```
fig,ax = plt.subplots() #this is single plot because we write anything in the bracket about about how many plo
ax.plot(x,y,'b*--')
plt.show()
```



In [132]:
```
fig,ax = plt.subplots() #this is single plot because we write anything in the bracket about about how many plo
ax.plot(x,y,'b,--')
plt.show()
```



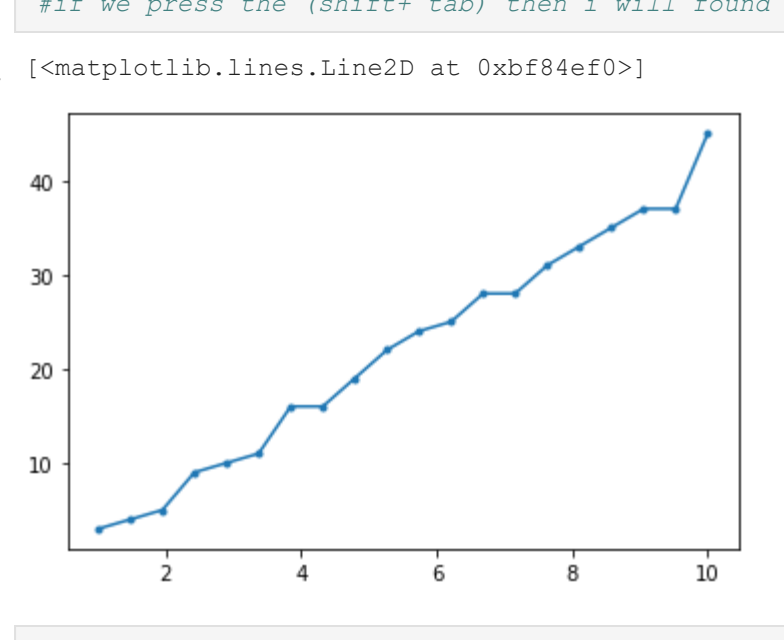
In [133]:
```
#if we press the (shift and tab)
fig,ax = plt.subplots()
ax.plot(x,y,'r.--')
plt.show()
```



Marker

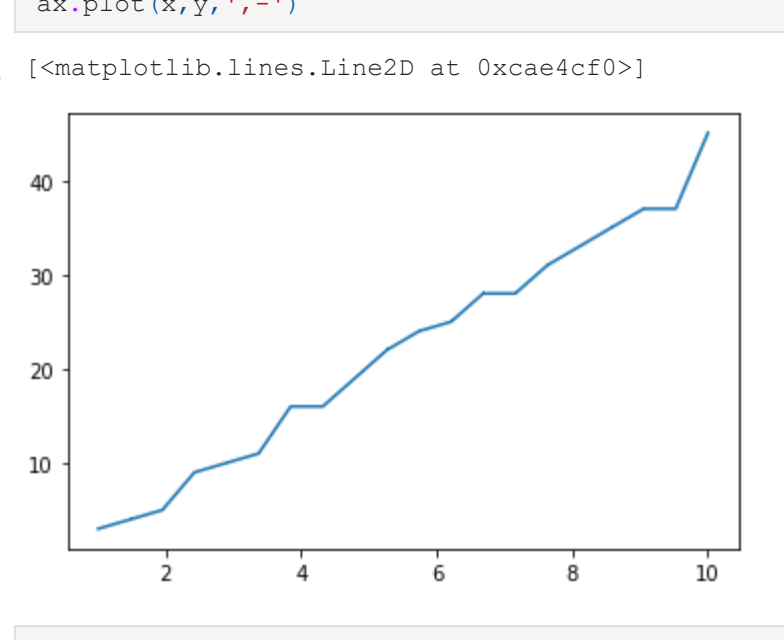
In [134]:
```
fig, ax = plt.subplots()
ax.plot(x,y,',-')
#if we press the (shift+ tab) then i will found the detailed info about the marker
```

Out[134]: [<matplotlib.lines.Line2D at 0xbf84ef0>]



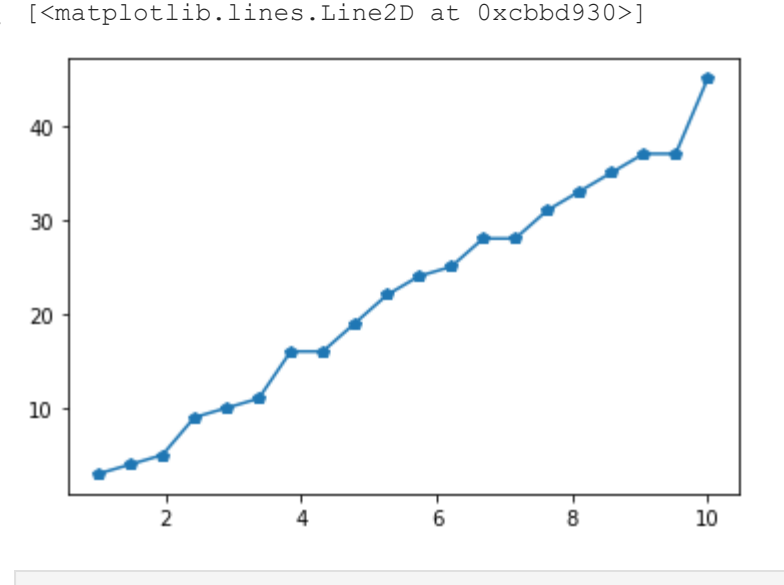
In [161]:
```
fig, ax = plt.subplots()
ax.plot(x,y,',-')
```

Out[161]: [<matplotlib.lines.Line2D at 0xcae4cf0>]



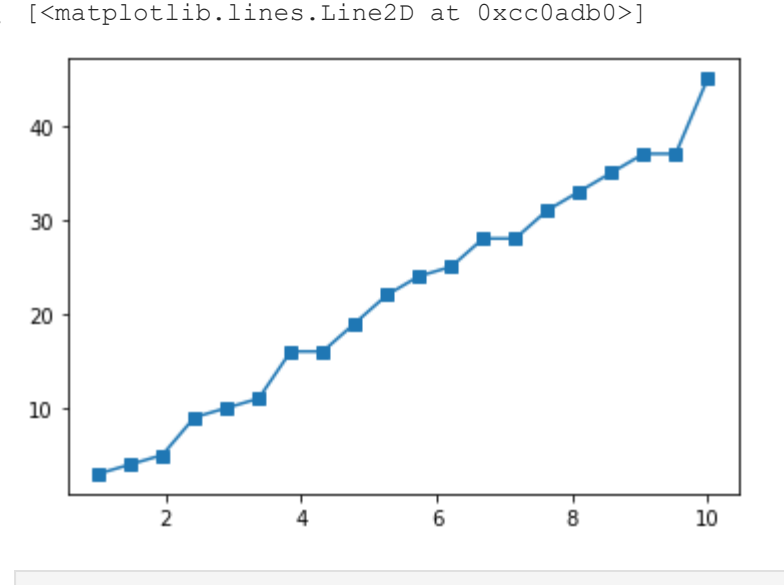
In [166]:
```
fig, ax = plt.subplots()
ax.plot(x,y,'p-')
```

Out[166]: [<matplotlib.lines.Line2D at 0xcbbd930>]



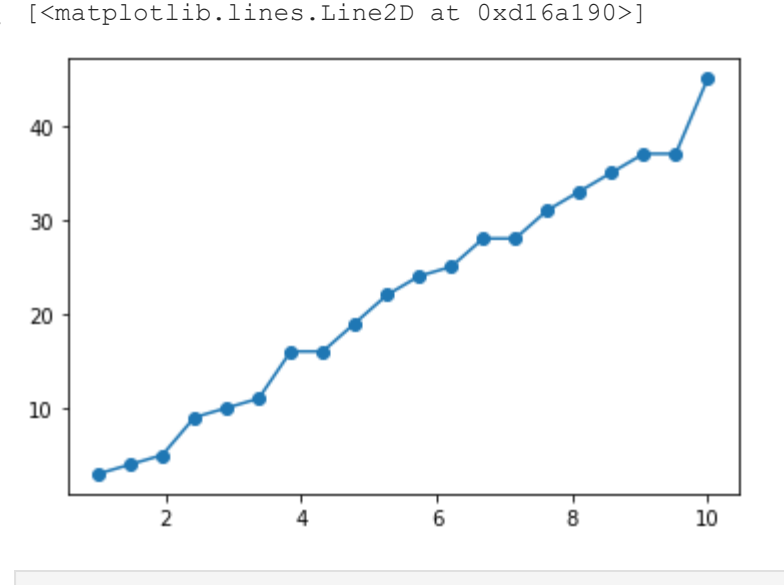
In [169]:
```
fig, ax = plt.subplots()
ax.plot(x,y,'s-')
```

Out[169]: [<matplotlib.lines.Line2D at 0xcc0adb0>]



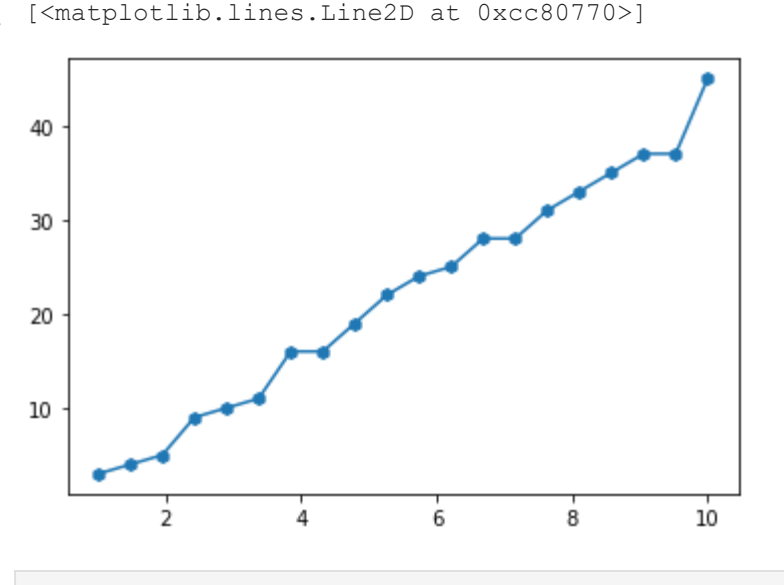
In [171]:
```
fig, ax = plt.subplots()
ax.plot(x,y,'o-')
```

Out[171]: [<matplotlib.lines.Line2D at 0xd16a190>]



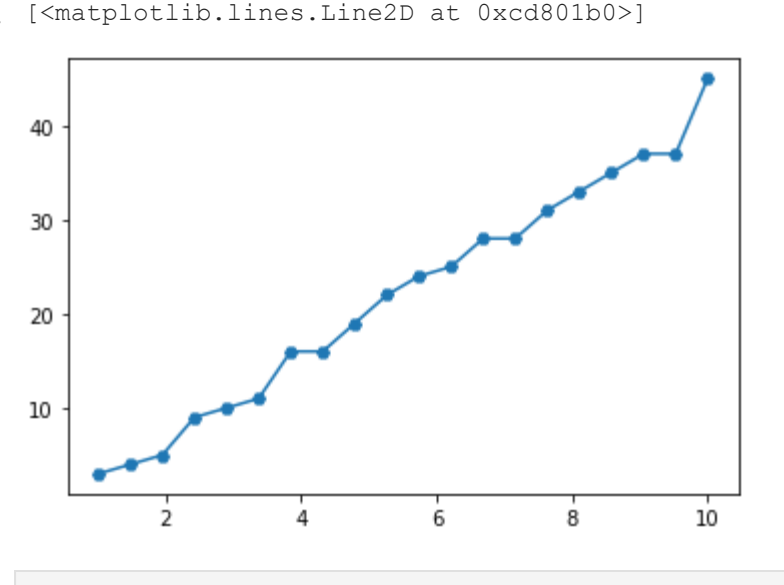
In [172]:
```
fig, ax = plt.subplots()
ax.plot(x,y,'h-')
```

Out[172]: [<matplotlib.lines.Line2D at 0xcc80770>]



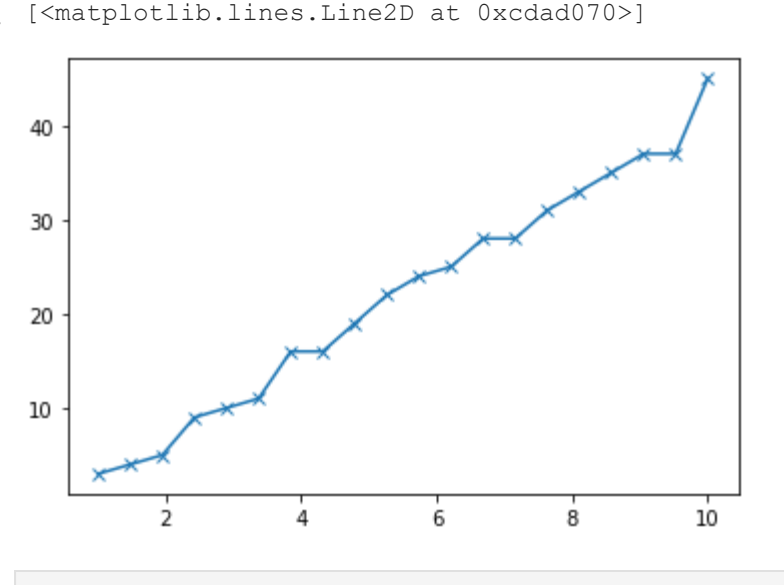
In [173]:
```
fig, ax = plt.subplots()
ax.plot(x,y,'H-')
```

Out[173]: [<matplotlib.lines.Line2D at 0xcd801b0>]



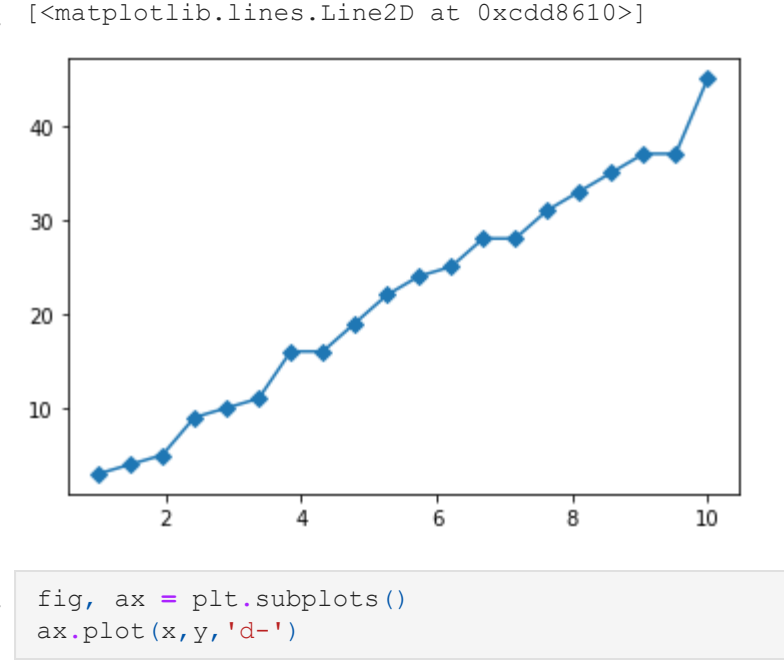
In [174]:
```
fig, ax = plt.subplots()
ax.plot(x,y,'x-')
```

Out[174]: [<matplotlib.lines.Line2D at 0xcdad070>]



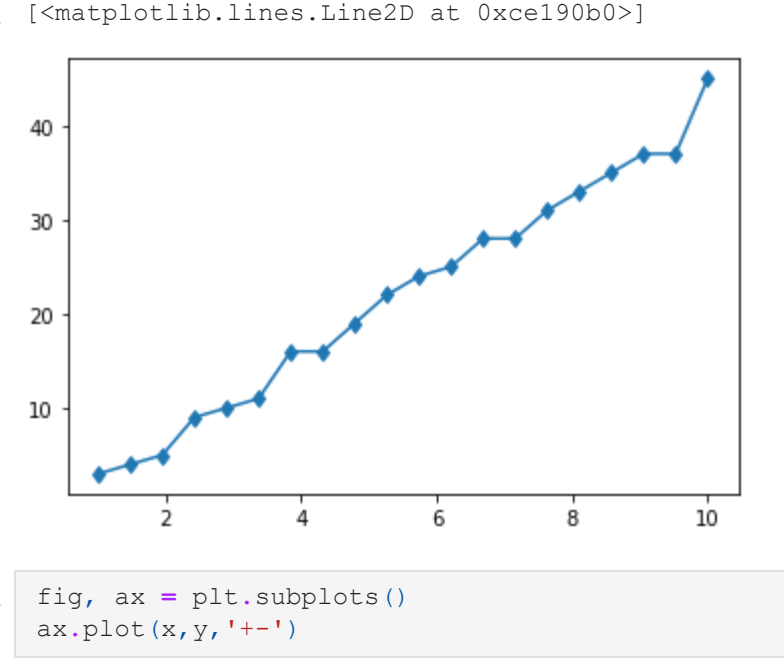
In [175]:
```
fig, ax = plt.subplots()
ax.plot(x,y,'D-')
```

Out[175]: [<matplotlib.lines.Line2D at 0xcdd8610>]



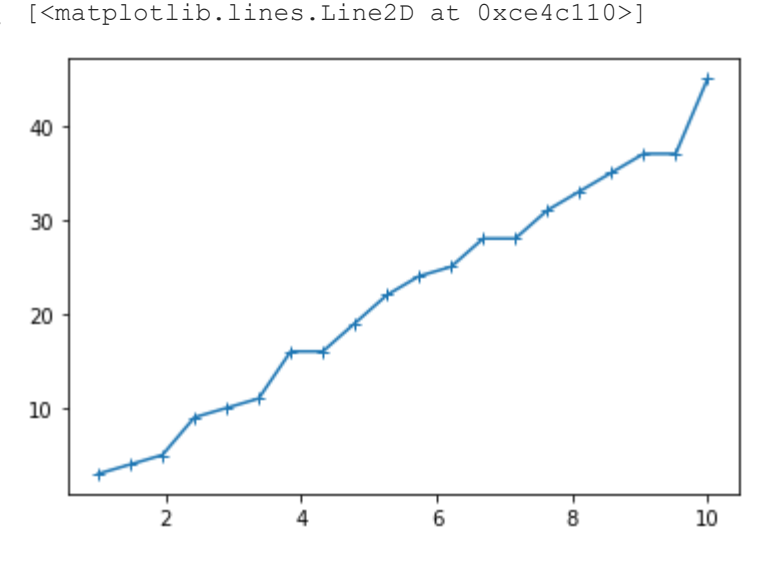
In [176]:
```
fig, ax = plt.subplots()
ax.plot(x,y,'d-')
```

Out[176]: [<matplotlib.lines.Line2D at 0xce190b0>]



In [177]:
```
fig, ax = plt.subplots()
ax.plot(x,y,'+-')
```

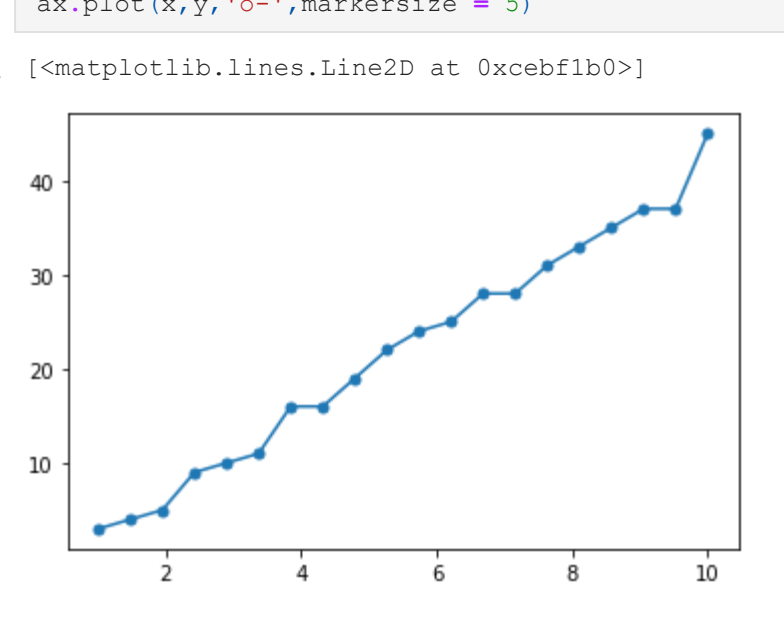
Out[177]: [<matplotlib.lines.Line2D at 0xce4c110>]



Markersize

In [180]:
```
#we change the size of the marker by using the function that is markersize = ?
fig,ax = plt.subplots()
ax.plot(x,y,'o-',markersize = 5)
```

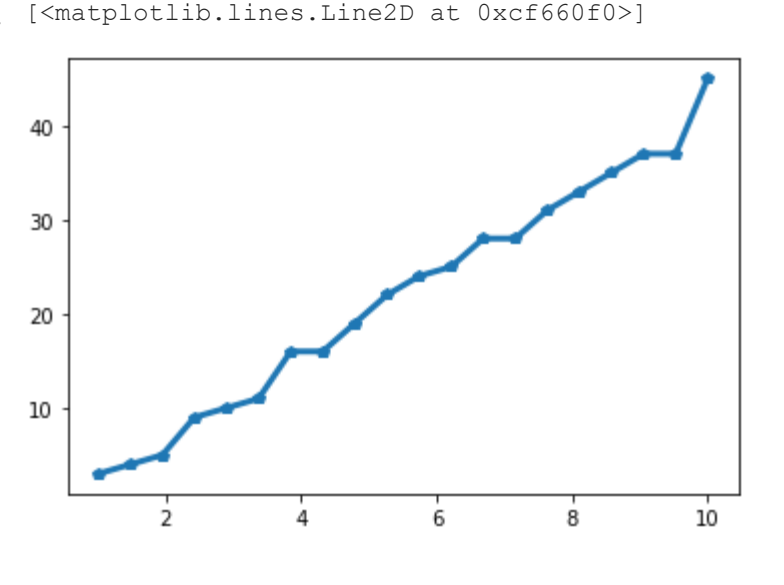
Out[180]: [<matplotlib.lines.Line2D at 0xcebf1b0>]



Linewidth

In [183]:
```
fig, ax = plt.subplots()
ax.plot(x,y,'p-',linewidth = 3)
```

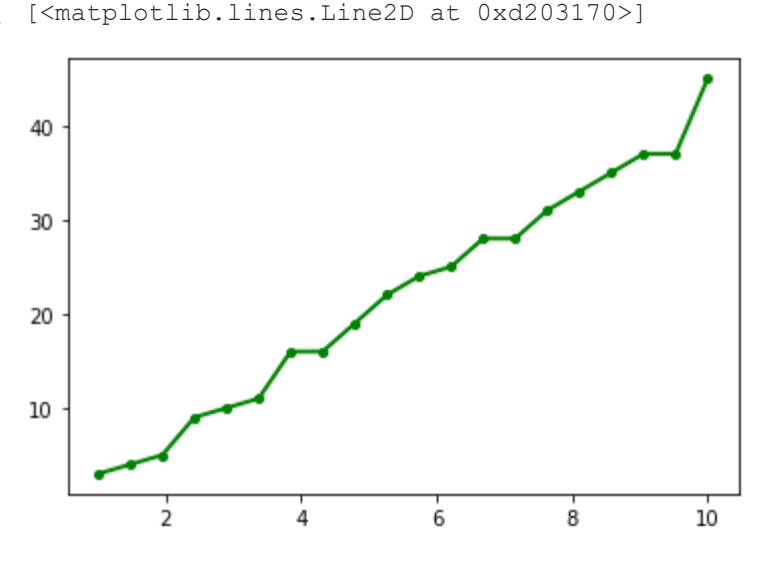
Out[183]: [<matplotlib.lines.Line2D at 0xcf660f0>]



Color

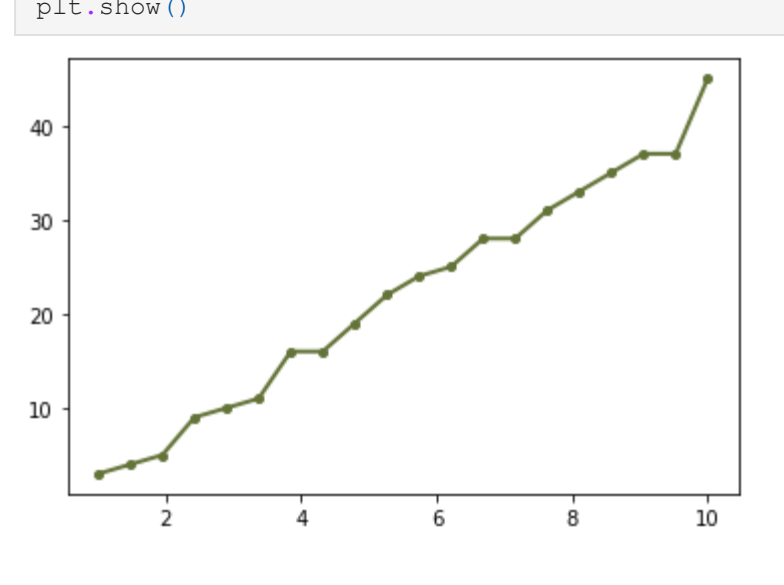
In [191]:
```
fig, ax = plt.subplots()
ax.plot(x,y,'o-',markersize = 4 ,linewidth= 2,color ='g')
```

Out[191]: [<matplotlib.lines.Line2D at 0xd203170>]



Hex Color

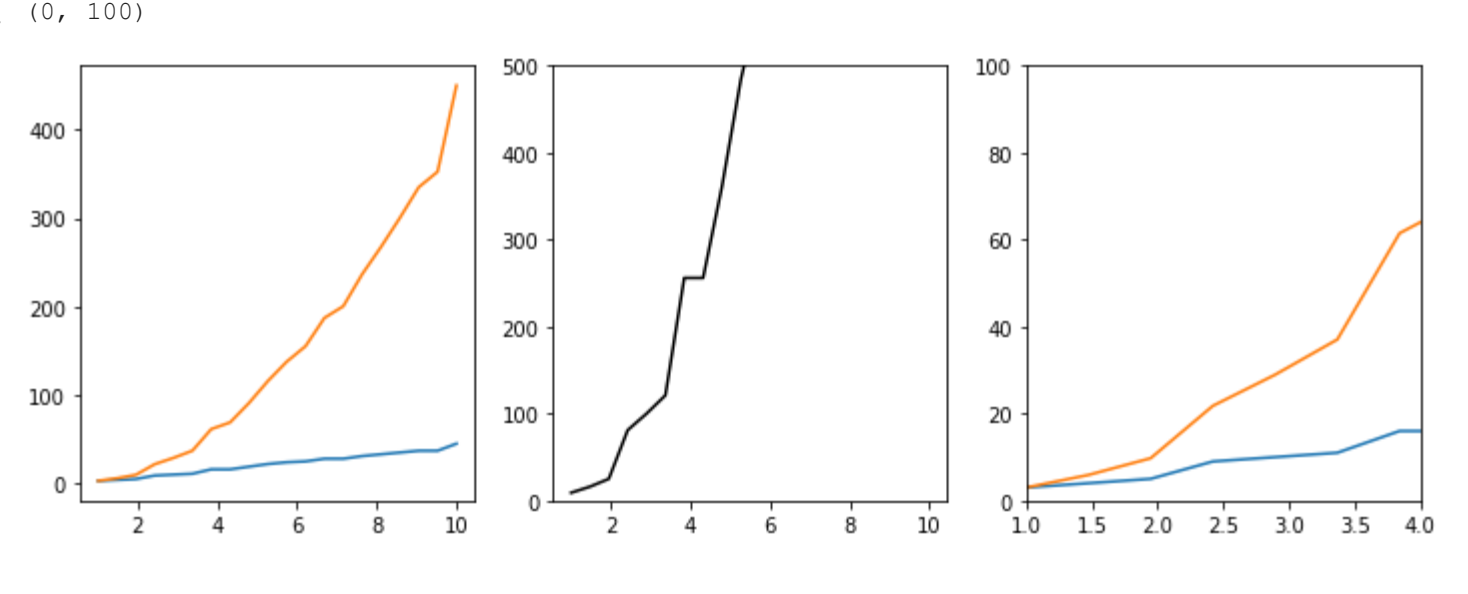
In [195]:
```
#we can use the hex colour like below
fig,ax = plt.subplots()
ax.plot(x,y,'o-',markersize = 4 ,linewidth= 2,color ='#647338')
plt.show()
```



Plot-range

In [209]:
```
fig, ax = plt.subplots(1,3,figsize =(12,4))
ax[0].plot(x,y,x,y1)
ax[1].plot(x,y**2,'k')
#in this pic we are founding the graphs in that we having the on e graph that is initial in that iam getting go
#but the another graph that is 1st which is extremly reach to the y limit nearby the 2000 so if we want to keep
#y axis upto the 500 that time we should have to use the ax.set_ylim([0,500])
ax[1].set_ylim([0,500])
ax[2].plot(x,y,x,y1)
ax[2].set_xlim([1,4])
ax[2].set_ylim([0,100])
```

Out[209]: (0, 100)



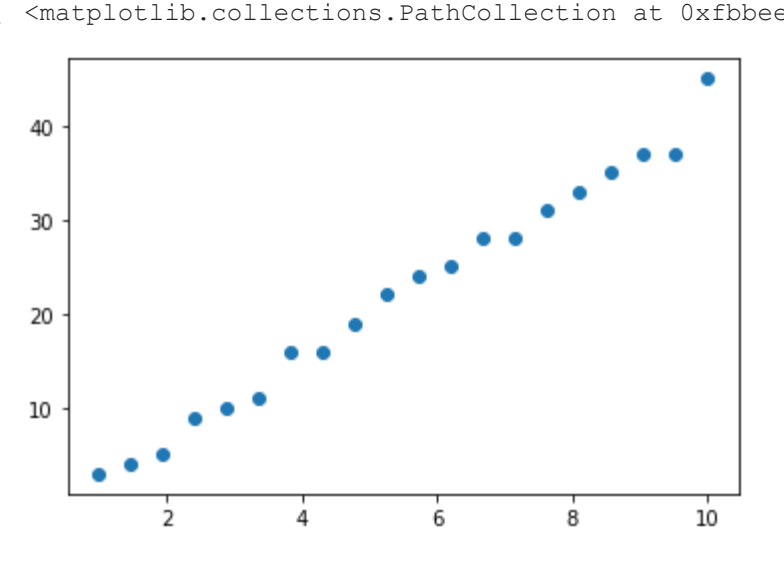
Other type of plots

In [ ]:
```
#bar,(horizontalbar)bar,pie,scatter,hist
```

Scatter Plots

In [212]:
```
plt.scatter(x,y)
```

Out[212]: <matplotlib.collections.PathCollection at 0xfbbeed0>

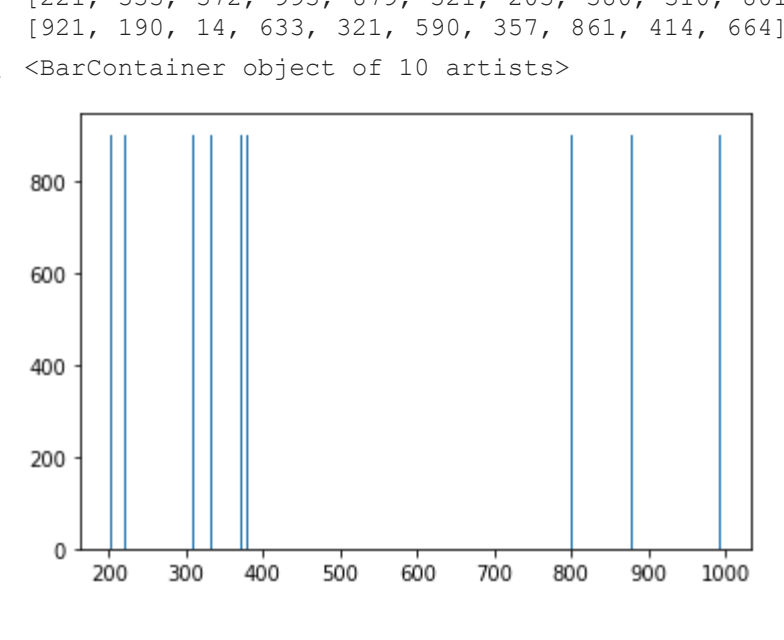


Bar Plot

In [227]:
```
from random import sample
data = sample(range(1,1000),10)
print(data)
data2 =sample(range(1,1000),10)
print(data2)
plt.bar(data,height =900,width =0.8)
```

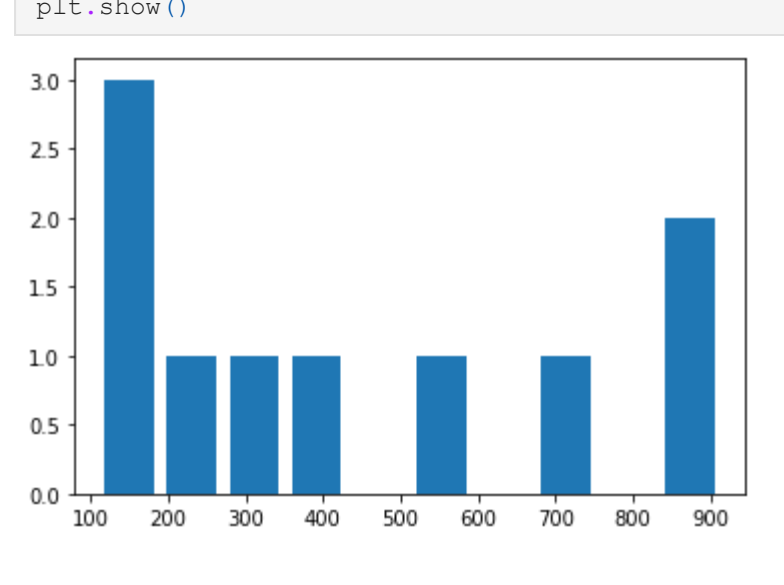
```
[221, 333, 372, 993, 879, 321, 203, 380, 310, 801]
[921, 190, 14, 633, 321, 590, 357, 861, 414, 664]
```

Out[227]: <BarContainer object of 10 artists>



Histogram plot

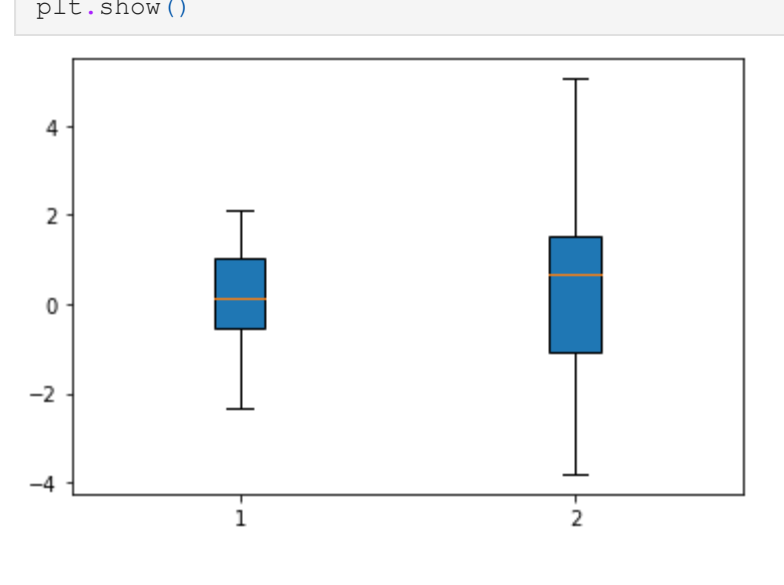
In [217]:
```
plt.hist(data,rwidth = 0.8)
#this plot showing the occurance of the number in the list
#number nearby the 700 occure one time only.
plt.show()
```



Boxplot

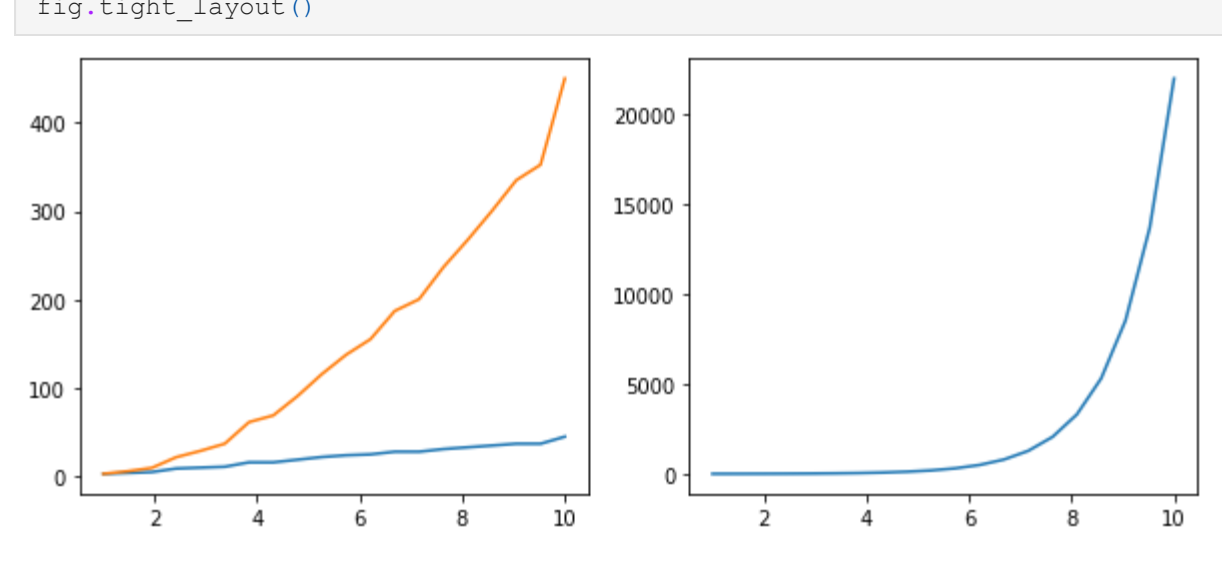
In [231]:
```
data =[np.random.normal(0,std,50) for std in range(1,3)]
```

In [232]:
```
plt.boxplot(data,vert=True,patch_artist = True)
plt.show()
```



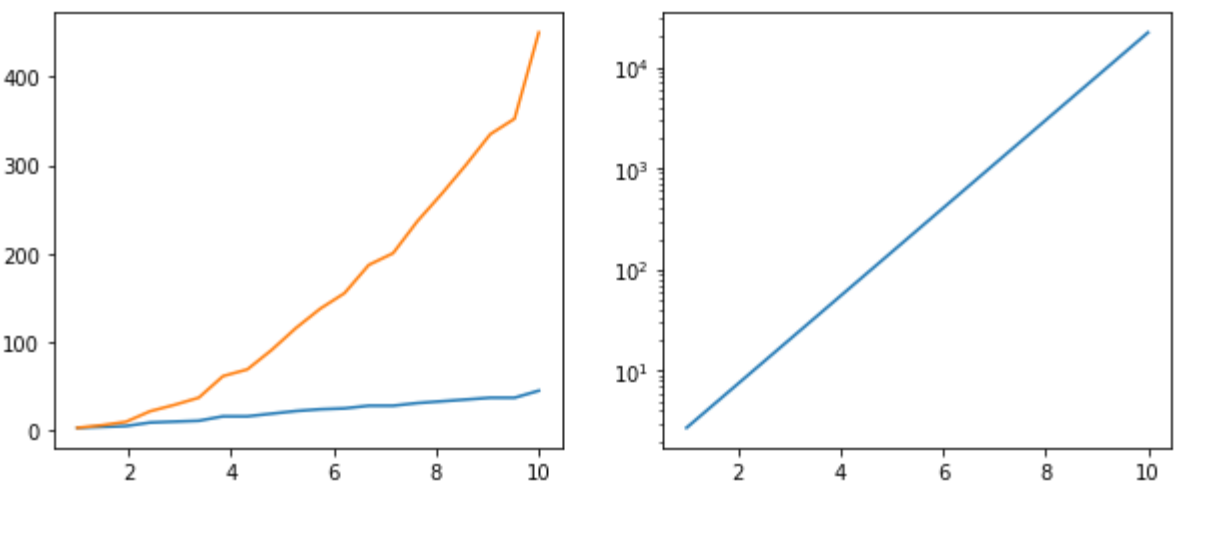
Log-axis-plot

In [237]:
```
fig,ax = plt.subplots(1,2,figsize = (10,4))
ax[0].plot(x,y,x,y1)
ax[1].plot(x,np.exp(x))
plt.show()
fig.tight_layout()
```



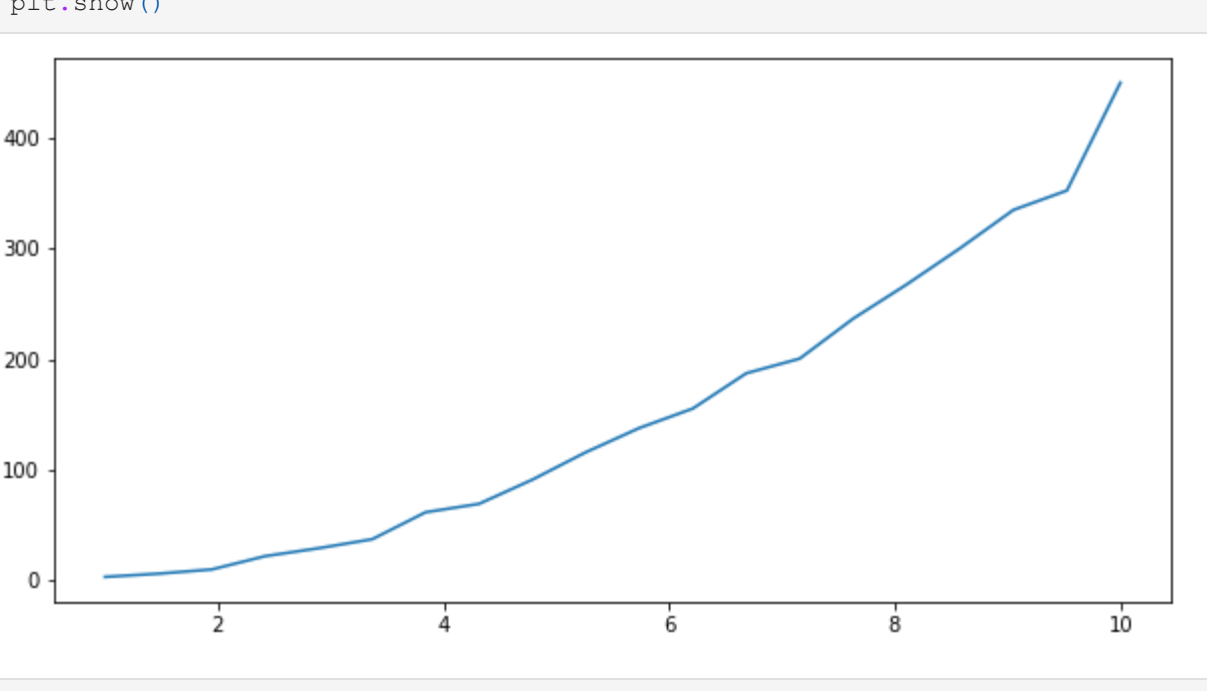
Set-Scale

In [238]:
```
#if we check that(y axis) is based on the linear scale so how we can change the linear scale to log sacle of th
fig,ax = plt.subplots(1,2,figsize = (10,4))
ax[0].plot(x,y,x,y1)
ax[1].plot(x,np.exp(x))
ax[1].set_yscale('log')
plt.show()
fig.tight_layout()
```

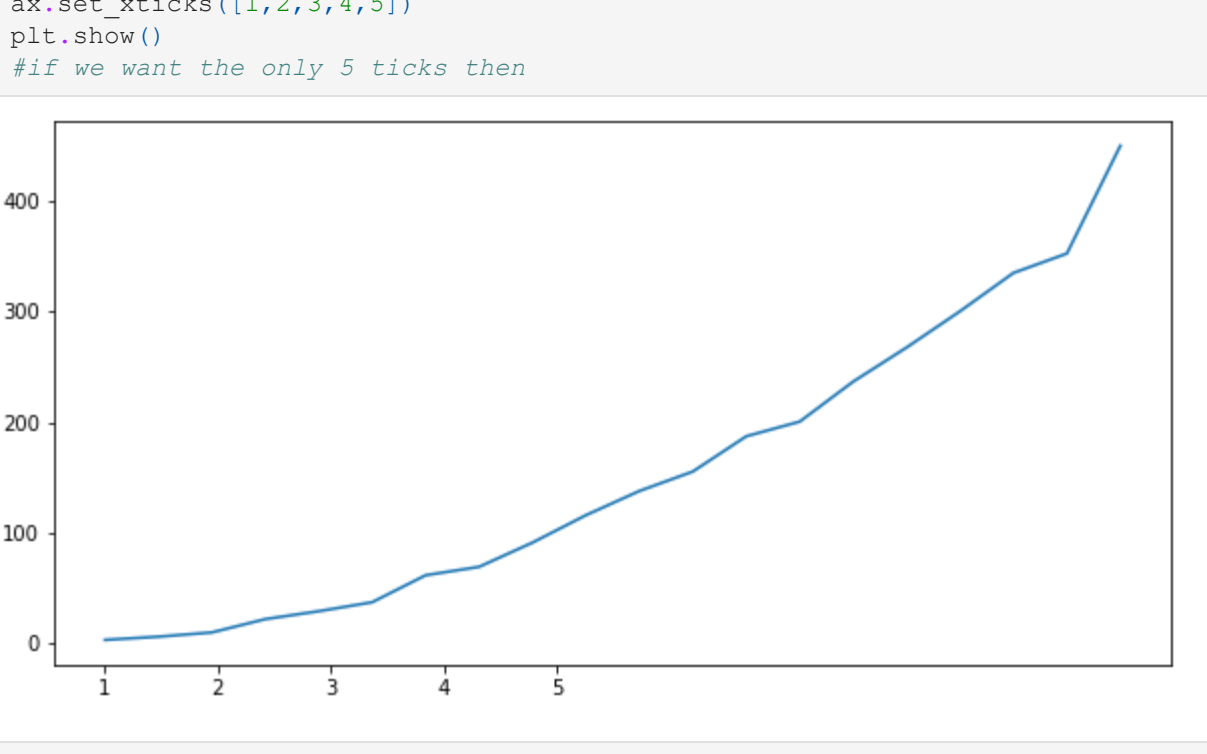


Ticks

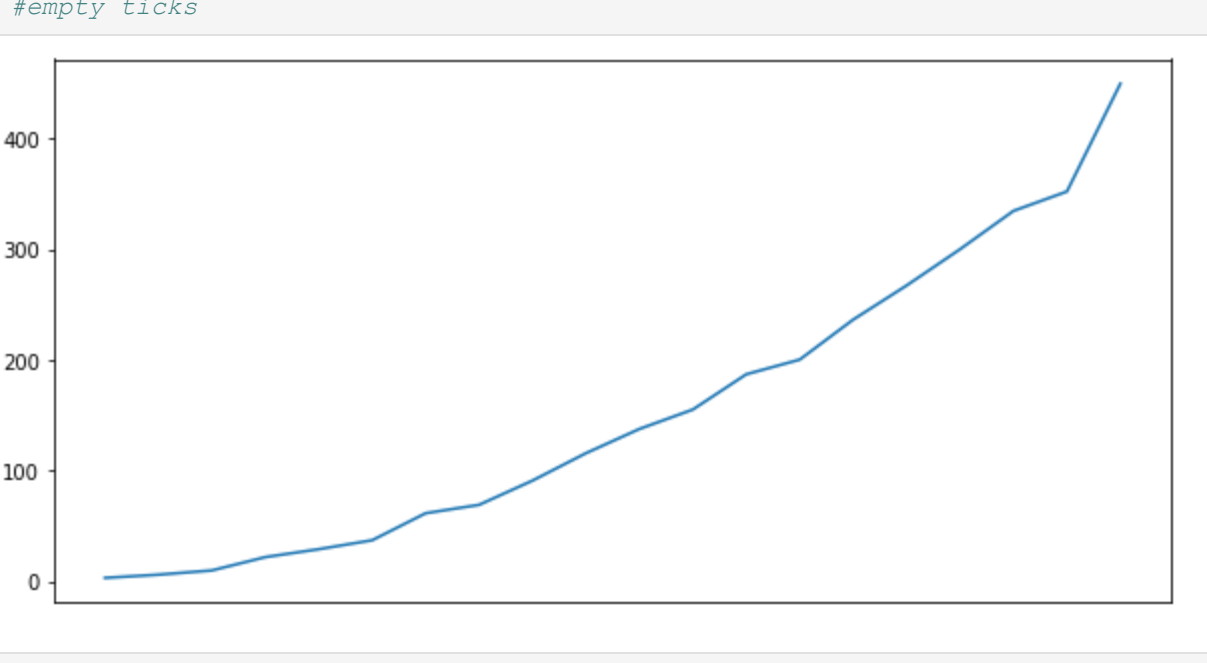
In [240…:
```
#how we chnge the ticks ,lable of the ticks,and everhing about the ticks
fig,ax = plt.subplots(figsize = (10,5))
ax.plot(x,y1)
plt.show()
```



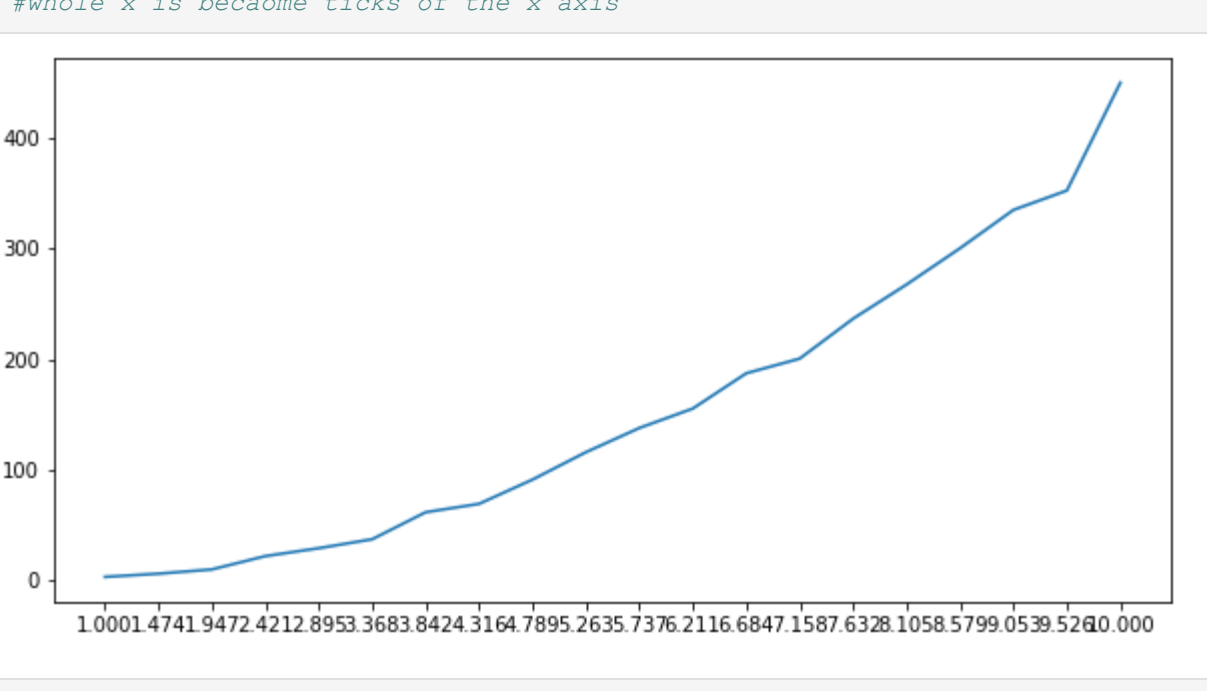
In [242…:
```
fig,ax = plt.subplots(figsize = (10,5))
ax.plot(x,y1)
ax.set_xticks([1,2,3,4,5])
plt.show()
#if we want the only 5 ticks then
```



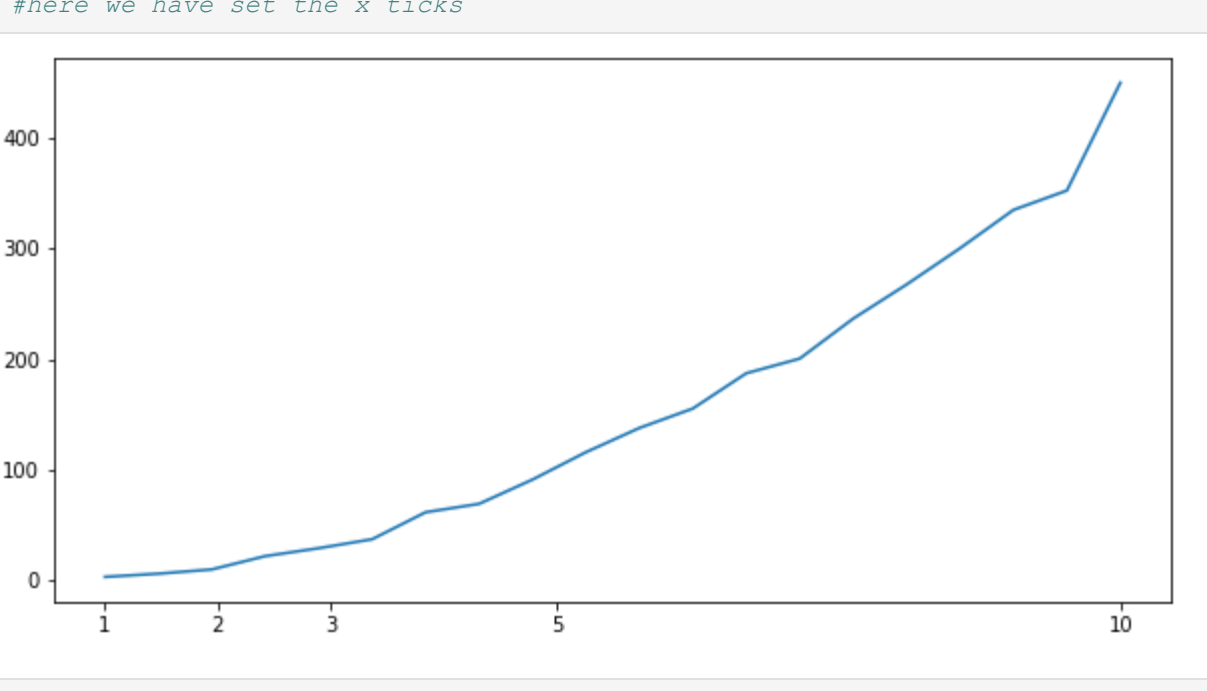
In [243…:
```
fig,ax = plt.subplots(figsize = (10,5))
ax.plot(x,y1)
ax.set_xticks([ ])
plt.show()
#empty ticks
```



In [245…:
```
fig,ax = plt.subplots(figsize = (10,5))
ax.plot(x,y1)
ax.set_xticks(x)
plt.show()
#whole x is becaome ticks of the x axis
```

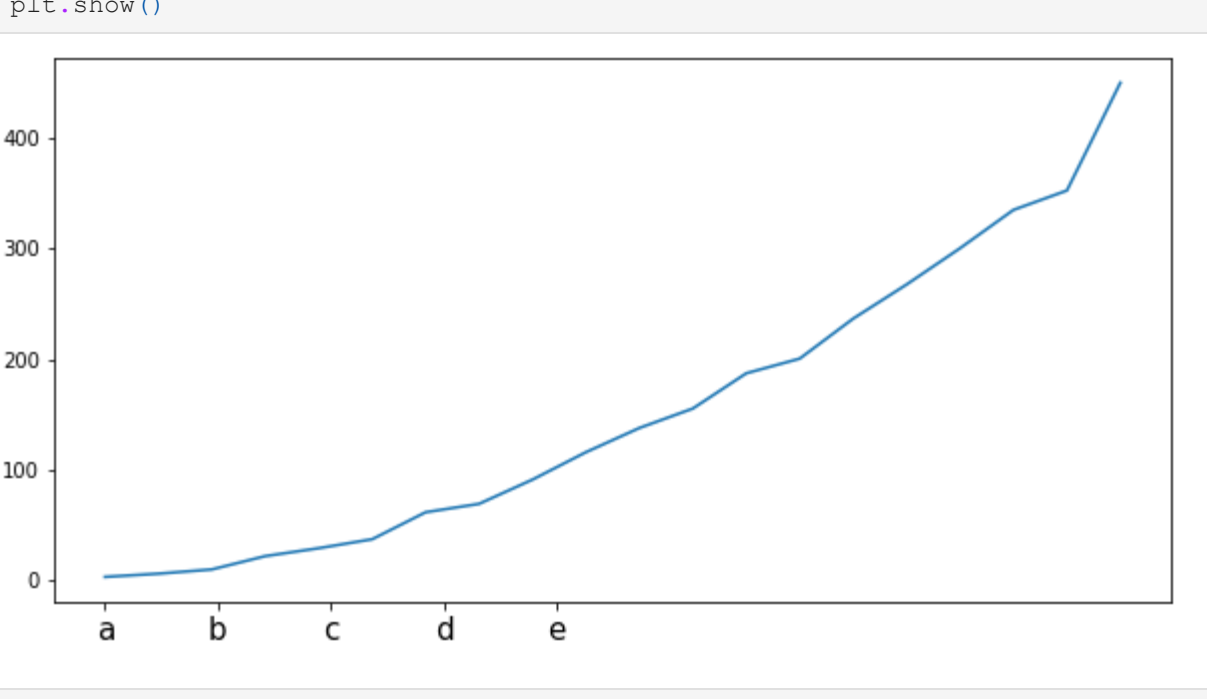


In [246…:
```
fig,ax = plt.subplots(figsize = (10,5))
ax.plot(x,y1)
ax.set_xticks([1,2,3,5,10])
plt.show()
#here we have set the x ticks
```

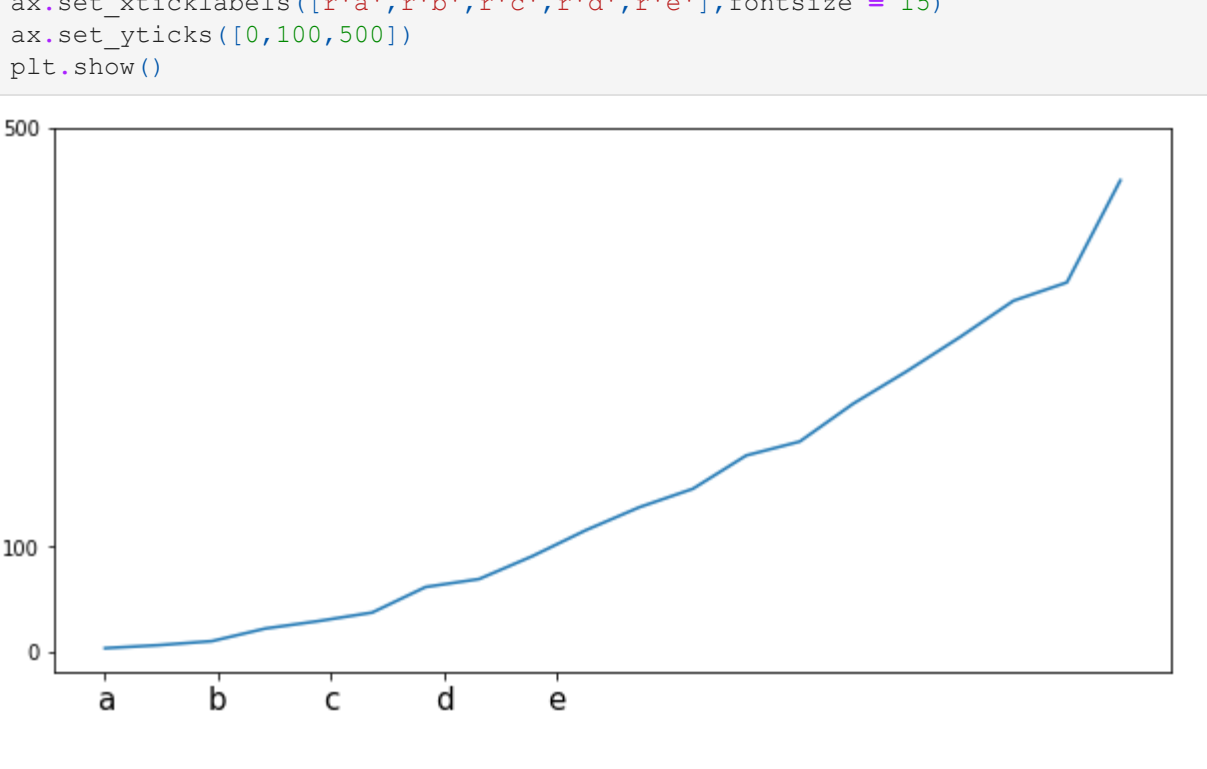


In [ ]:
```
#we can set the label to each of the ticks
```

In [255…:
```
fig,ax = plt.subplots(figsize = (10,5))
ax.plot(x,y1)
ax.set_xticks([1,2,3,4,5])
ax.set_xticklabels([r'a',r'b',r'c',r'd',r'e'],fontsize = 15)
plt.show()
```



In [257…:
```
#here we have set the y ticks labels
fig,ax = plt.subplots(figsize = (10,5))
ax.plot(x,y1)
ax.set_xticks([1,2,3,4,5])
ax.set_xticklabels([r'a',r'b',r'c',r'd',r'e'],fontsize = 15)
ax.set_yticks([0,100,500])
plt.show()
```

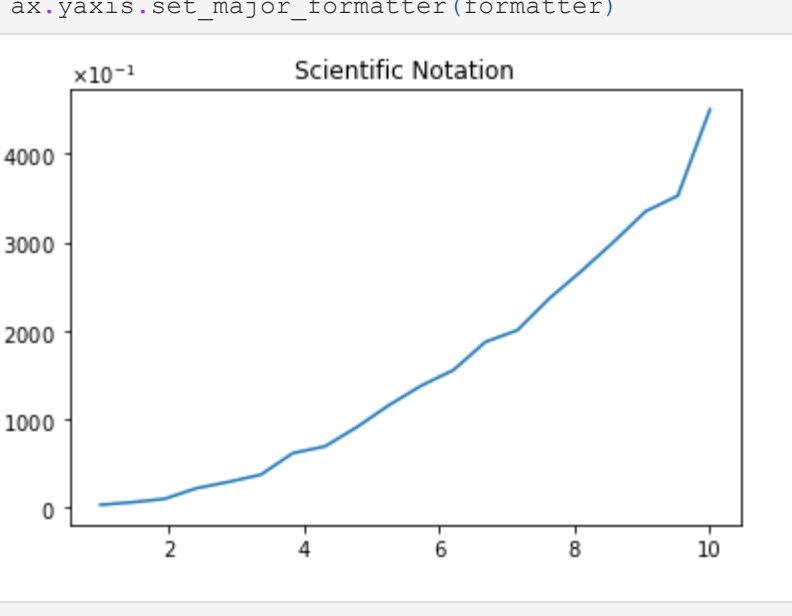


Scientific Notation

In [ ]:
```
from matplotlib import ticker
```

In [262…:
```
from matplotlib import ticker
fig,ax = plt.subplots()
ax.plot(x,y1)
ax.set_title('Scientific Notation')

formatter =ticker.ScalarFormatter(useMathText= True)
formatter.set_scientific(True)
formatter.set_powerlimits((-1,-1))
ax.yaxis.set_major_formatter(formatter)
```



In [ ]:

In [ ]:

In [ ]:

In [ ]: