Data Visualization (Matplotlib)

Contents

Data Visualization (Matplotlib)

Matplotlib သည် Python programming language ၏ အခြေခံ အကျဆုံးသော data visualization tool တစ်ခု ဖြစ်သည်။ Matplotlib object hierarchy ၊ plot အမျိုးအစားများ၊ customization techniques တို့ကို ရှင်းပြထားသည်။

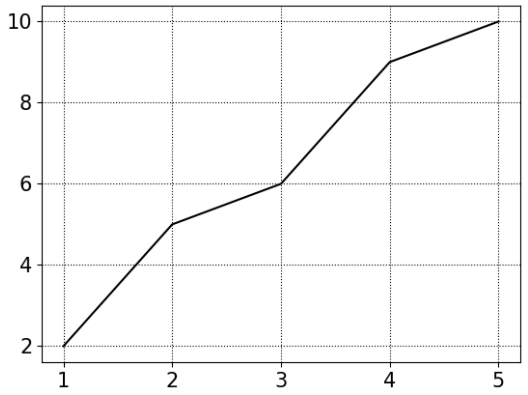
1. Introduction

ပထမဆုံး လက်နဲ့ ဂရပ်စာရွက်ပေါ်မှာ ဆွဲကြရအောင်။ ပုံမှန်အားဖြင့် ဂရပ်တစ်ခုမှာ X ဝင်ရိုးနှင့် Y ဝင်ရိုး ဆိုပြီး (၂)ခုရှိသည်။ ဂရပ်ဆွဲတဲ့အခါ X တန်ဖိုးတွေကို X ဝင်ရိုးပေါ်မှာ ချပြီးရင် Y တန်ဖိုးတွေကို Y ဝင်ရိုးပေါ်မှာ ချပြီးတော့ ဖြတ်မှတ်တွေကို ဆက်ရင် ဂရပ်လိုင်း ရသည်။

Matplot နဲ့ဆွဲရင်လည်း X တန်ဖိုးတွေ နှင့် Y တန်ဖိုးတွေ ရှိရမည်။ ဂရပ်စာရွက်ပေါ်မှာလို လက်နဲ့ လိုက်ချစရာ မလိုဘူး။ plt.plot(x,y) ဆိုရင် ရပြီ။ plt.plot(x,y) မှ plt ရယ် .plot() ရယ် x,y ရယ် ဆိုပြီးတော့ (၃)ခု ပါသည်။ plt ကတော့ matplotlib.pyplot ရဲ့ အတိုခေါက်ပါ။ ပုံဆွဲပေးဘို့ လိုအပ်တဲ့ Python package သို့မဟုတ် library ပါ။ .plot() ဆိုတာ ပုံဆွဲပေးတဲ့ function ပါ။ x, y ဆိုတာက X တန်ဖိုးတွေ နှင့် Y တန်ဖိုးတွေပါ။ X တန်ဖိုးတွေ နှင့် Y တန်ဖိုးတွေကို .plot()function ထဲ ထည့်ပေးလိုက်ပြီး ပုံဆွဲတာပါ။ plt ပုံဆွဲပေးတဲ့ Python library ပါ။ plt.show() ကတော့ ဆွဲပြီးသာ ပုံကို ပြပေးဘို့ ထည့်ပေးရတာပါ။ Python မှ **.show()** ဆိုမှာ ဖော်ပြပေးသည်။

`plt.plot(x,y)` တစ်ကြောင်းဖြင့် အကြမ်းဆွဲပေးပါလိမ့်မည်။ သို့သော် စနစ်တကျ ပုံနှိပ် ထုတ်ဝေသည့်အဆင့် ဂရပ်ဖြစ်ရန် တခြားသော အချက် အလက်များ ထည့်သွင်းပေးရပါမည်။

```
import matplotlib.pyplot as plt
x = [1,2,3,4,5]
y = [2,5,6,9,10]
plt.plot(x,y)
plt.show()
```



Data Visualization သည် ဂရပ်များ ရေးဆွဲခြင်းသာမက အချက်အလက်များကို ခွဲခြမ်းစိတ်ဖြာလေ့လာသည့် အလုပ်များလည်း ပါဝင်ပါသည်။ Data Visualization တွင် ကိန်း ဂဏန်းအချက်အလက်(numerical data)များကို input အဖြစ် ထည့်သွင်းပြီး chart များ figure များနှင့် ဇယားများ(table)ကို output အဖြစ် ထုတ်ပေးခြင်းတွေလည်း ပါဝင်သည်။ Data Visualization သည် ဒေတာများကို ပုံဖော်ကြည့်ပြီး တိကျမှန်ကန်သည့် ဆုံးဖြတ်ချက်များ ချမှတ်နိုင်သည့်အဆင့်အထိ ကျယ်ပြန့်ပါသည်။ ဆုံးဖြတ်ချက်များ ချမှတ်နိုင်အောင် ကောင်းစွာ အထောက်အကူ ပြုနိုင်သည့် အဆင့်အထိ ပါဝင်သည်။

အနည်းသော အားထုတ်မှုဖြင့် ရှင်းလင်းပြည့်စုံသည့် ဂရပ်ပုံများ ကားချပ်များ ဇယားများကို ပြုစုထုတ်လုပ်ပေးနိုင်သည့် **Matplotlib** အကြောင်း လေ့လာကြပါစို့။ **Matplotlib သည်** Python programming language တွင် data visualization tool တစ်ခုဖြစ်သည်။ Python တွင် အမျိုးမျိုးသော ကိစ္စရပ်များအတွက် အသုံးပြုရန် visualization tool များစုံရှိသည်။ ဤစာအုပ်တွင် **Matplotlib** အကြောင်းကိုသာ ရေးသား ဖော်ပြထားသည်။

2. Python Visualization Tools

Python သည် data scientist များ language အကြိုက်ဆုံးသော ဖြစ်သည်။ . Python တွင် data visualization လုပ်ရန်အတွက် ရွေးချယ်စရာများစွာ ရှိသည်။ Python တွင် အသုံးပြုနိုင်သည့် data visualization tools များမှာ

Matplotlib	Seaborn	pandas
Bokeh	Plotly	ggplot
pygal		

Data visualization tool တစ်ခုဖြစ်သည့် Matplotlib အကြောင်းကို အဓိက ရှင်းပြပါမည်။

3. Matplotlib

Matplotlib သည် Python programming language တွင် အခြေခံအကျဆုံးသော plotting library တစ်ခုဖြစ်သည်။ Python visualization package အတွင် မရှိမဖြစ် အသုံးပြုရသည့် plotting library ဖြစ်သည်။

Matplotlib ဂရပ်ပုံအမျိုးမျိုးကို ရေးဆွဲရန်အတွက် အလွန်သင့်လျော် လွယ်ကူသည်။ ပုံနှိပ်ထုတ်ဝေနိုင်သည့် အဆင့် အရည်အသွေး ရရှိနိုင်သည်။ like PDF, SVG, JPG, PNG, BMP and GIF စသည့် အသုံးများသည့် format အမျိုးမျိုးကိုလည်း ထုတ်ပေးနိုင်သည်။

Line plot, scatter plot, histogram, bar chart, error charts, pie chart, box plot, တို့ အပြင် တခြား အသုံးများသော visualization ဂရပ်များကိုလည်း ဆွဲနိုင်သည်။ 3D ဂရပ်များကိုလည်း ဆွဲနိုင်သည်။ Matplotlib ပေါ်တွင် အခြေခံ၍ pandas ၊ Seabor စသည့် Python library များကို တည်ဆောက်ထားသည်။ ကုဒ်အနည်းရေးရုံဖြင့် Matplotlib ကို access လုပ်နိုင်သည်။

၂၀၀၂ ခုနှစ်တွင် **Matplotlib** ကို John Hunter က စတင်ခဲ့သည်။ အစပိုင်းတွင် Neurobiology ဘွဲ့လွန် သုတေသန ကျောင်းသားများ အတွက် ဝက်ရူးပြန်ရောဂါသည်များ၏ Electrocorticography (ECoG)ပုံများကို ပုံဖော်ပေး(visualize လုပ်)ရန်အတွက် ဖြစ်သည်။ Python programming language နှင့် ပေါင်း၍ open-source tool Matplotlib ဖြစ်ပေါ်လာသည်။ ၂၀၀၈ ခုနှစ်နောက်ပိုင်းတွင် data visualization အတွက် စတင် အသုံးပြုလာကြသည်။

4. Import Matplotlib

Python တွင် data visualization လုပ်ရန်(ဂရပ်ဆွဲရန်)အတွက် ပထမဦးစွာ Matplotlib ကို import လုပ်ရန်လိုအပ်သည်။ Import မလုပ်ခင် matplotlib.pyplot ရှိ မရှိ စစ်ရန်လိုသည်။ Terninal တွင်pip list ဖြစ် စစ်နိုင်သည်။ မရှိပါက pip install matplotlib ဖြင့် install လုပ်ပါ။ Requirement already satisfied ပြလျှင် install လုပ်ပြီး ဖြစ်သည်။ `import matplotlib`

ဂရပ်များဆွဲရန် Matplotlib ထဲတွင် ပါဝင်သည့် **pyplot** ကို အများဆုံး အသုံးပြုသည်။ ထို့ကြောင့် **pyplot** ဟု တိတိကျကျ ရေးသား၍လည်း import လုပ်နိုင်သည်။

```
import matplotlib.pyplot
```

pyplot ကို import လုပ်သည့်ပြီး သုံးသည့်အခါတိုင်း နာမည် အပြည့်အစုံ ရေးရသည်။ နာမည် အပြည့်အစုံ ရေးလျှင် ရှည်လွန်းလှသောကြောင့် နာမည်အတိုခေါက်ရေး ရန်တွက် **as** keyword ကို ၍ နာမည်ပြောင်းသည်။

```
import matplotlib.pyplot as plt

import numpy as np          # ဂရပ်ဆွဲရန်အတွက် ဒေတာများ ထုတ်ယူရန်
import pandas as pd         # ဂရပ်ဆွဲရန်အတွက် ဒေတာများ ထုတ်ယူရန်
```

5. Matplotlib ဖြင့် ဂရပ်ပုံများ ပုံဖော်ခြင်း

Matplotlib တွင် ဆွဲပြီးသည့် ဂရပ်ပုံများကို ကြည့်ရန် **plt.show()** command ကို အသုံးပြုရသည်။ မသုံးပါ အလိုအလျောက် မဖော်ပြပေးပါ။ Matplotlib plot ကို နည်း(၃)မျိုးဖြင့် ရေးနိုင်သည်။ နည်း(၃)နည်းဖြင့် ဂရပ်ပုံများကို ကြည့်ရန် command ပေးရသည်။

- plotting from a script,
- plotting from an IPython shell or
- plotting from a Jupyter notebook.

5.1 Plotting from a script

Python script များအကြားတွင် Matplotlib ကို ထည့်သုံးပါက **plt.show()** command ကို သုံးရသည်။ Python session တစ်ခုတိုင်းအတွက် **plt.show()** command ထည့်ပေးရသည်။ ပုံမှန်အားဖြင့် plot script များ၏ အောက်ဆုံးလိုင်းတွင် ထည့်ရေးလေ့ ရှိသည်။

5.2 Plotting from an IPython shell

IPython shell တွင် Matplotlib ကို ချဲ့ခြင်း ချုံ့ခြင်း zoon ဆွဲခြင်း (interactively)တို့လုပ်နိုင်သည်။ IPython တွင် %matplotlib magic command ကို သုံး၍ Matplotlib mode အဖြစ် သုံးနိုင်သည်။

5.3 Plotting from a Jupyter notebook

Jupyter Notebook (အရင်က IPython Notebook ဟုခေါ်သည်။)တွင် ဂရပ်ဆွဲလိုပါက သို့မဟုတ် ဆွဲထားသည့် ဂရပ်များ web brower ထဲတွင် ပေါ်လိုပါက %matplotlib command ကို ထည့်ပေးရသည်။ နည်း(၂)မျိုး ရှိသည်။ Input၊ output လိုင်းများအကြားတွင် ဆွဲထားသည့် ဂရပ်များကို ထည့်သွင်း ဖော်ပြ လိမ့်မည်။

• **%matplotlib notebook** – သည် interactive plot ကို web brower ထဲတွင် embedded လုပ်ပေးသည်။ ချဲ့ခြင်း ချုံ့ခြင်း လိုအပ်သည်နေရာကို zoon ဆွဲကြည့်ခြင်း စသည်တို့ ပြုလုပ်နိုင်သည်။

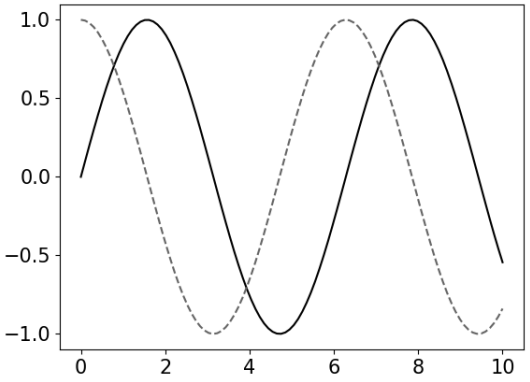
%matplotlib inline – သည် static image ကိုသာ output အဖြစ်ထုတ်ပေးသည်။ static image ဆိုသည်မှာ ချဲ့ခြင်း ချုံ့ခြင်း လိုအပ်သည်နေရာကို zoon ဆွဲကြည့်ခြင်း စသည်တို့ မပြုလုပ်နိုင်ပါ။ **%matplotlib inline** ထည့်ပေးရန်လိုသည်။

6. Matplotlib Object Hierarchy

Matplotlib ကို အသုံးပြုသည့်အခါ Object Hierarchy ဆိုသည့် ဝေါဟာရကို သိထားရန်လိုသည်။ Matplotlib တွင် Python object များသည် အလွှာလိုက် အထပ်လိုက်(nested)ရှိနေသည်။ Hierarchy ဆိုသည်မှာ တစ်ခုပေါ်တွင် တစ်ခု အဆင့်ဆင့် ရှိနေခြင်းကို ဆိုလိုသည်။ ဥပမာ ပန်းချီကားတစ်ချပ် ဆွဲရန်အတွက် ဘောင်ရှိရမည်။ ထိုဘောင်ပေါ်တွင် ကင်းဘတ်စ်စကို တင်းနေအောင် တင်ထားရမည်။ ထို ကင်းဘတ်စ်စပေါ်တွင် ခံဆေးသုတ်ရန်

လိုသည့်အခါ ခံဆေးသုတ်ရသည်။ အားလုံးအဆင်သင့် ဖြစ်သည့်အခါမှာ ပန်းချီ စဆွဲနိုင်သည်။ ထို့အတူ ဂရပ်တစ်ခု ဆွဲရန်အတွက် လိုအပ်သည့် ပြင်ဆင်မှုအဆင့်ဆင့်ရှိသည်။ ထိုအဆင့်များသည် Hierarchy ဖြစ်သည်။

```
import matplotlib.pyplot as plt
%matplotlib inline
x1 = np.linspace(0, 10, 100)

# create a plot figure
fig = plt.figure()

plt.plot(x1, np.sin(x1), '-')
plt.plot(x1, np.cos(x1), '--')
plt.show()
```



Matplotlib ကို သုံးမည့် သူတိုင်း **Figure** object နှင့် **Axes** objects အကြောင်းကို ကွဲပြားစွာ နားလည် သဘောပေါက်ထားသင့်သည်။

Figure object သည် ကင်းဘတ်စ်စ နှင့်တူသည်။ Matplotlib plot တွင် အခြေခံအကျဆုံး ဖြစ်သည်။

Axes object များသည် ပန်းချီ ရုပ်ပုံများနှင့်တူသည်။ တစ်ခါတစ်ရံ ရုပ်ပုံများစွာ ရေးဆွဲသည့်အခါ **Axes** တစ်ခုသည် ရုပ်ပုံတစ်ခု ဖြစ်သည်။ ထို့ကြောင့် Figure object ဆိုသည့် ကင်းဘတ်စ်စ ပေါ်တွင် Axes object ဆိုသည့် ရုပ်ပုံ များစွာ ရှိနေနိုင်သည်။ Matplot တွင် Axes များကို ဂရပ်ပုံ အငယ်များ(subplot) ဟုခေါ်သည်။ **Subplot** နှင့် **Axes** တို့သည့် အပြန်အလှန် ပြောင်းလဲ ခေါ်နိုင်သည့် နာမည်များ ဖြစ်သည်။

Matplot ဝေါဟာရ ဖြင့်ပြောလျှင် Axes ဆိုသည်မှာ ဂရပ်ငယ်ကလေးများ ဖြစ်သည်။ (The Axes represent an individual plot.)။ Figure ထဲတွင် Axes တစ်ခု သို့မဟုတ် တစ်ခုထက်ပိုများသည့် ဂရပ်ငယ်ကလေး (subplot)များ ရှိနေနိုင်သည်။ (Figure object as a box-like container containing one or more Axes.)

လူရုပ်ပုံ ငှက်ပုံများတွင် ခေါင်း၊ ခြေ၊ လက် စသည်တို့ ရှိသကဲ့သို့ Axes တစ်ခုတွင်လည်း tick marks, lines, legends, title and text-boxes စသည်တို့ ပါဝင်သည်။

7. Matplotlib API

Matplotlib တွင် API (၃)မျိုးရှိသည်။ MATLAB-style state-based interface နှင့် object-oriented (OO) interface တို့ဖြစ်သည်။ MATLAB-style ကို **pyplot interface** ဟု ခေါ်သည်။ **Object-Oriented** interface ဟု ခေါ်သည်။ တတိယ အမျိုးအစားသည် **pylab** interface ဖြစ်သည်။

8. Pyplot API

Matplotlib.pyplot သည် MATLAB-style, procedural, state-machine interface ဖြစ်သည်။ **Pyplot** ထဲတွင် ဂရပ်ပုံစံများစွာ ဆွဲနိုင်သည့် function များ ရှိသည်။ Pyplot function ထဲတွင် figure များ ရေးဆွဲ ပေးရန် function များ ၊ ပုံဆွဲရမည့်နေရာကို သတ်မှတ်ပေးသည့် plotting area function များ ပါဝင်ကြသည်။

Matplotlib.pyplot ကို သုံးသည့်အခါ object reference များဖြင့် မညွှန်းဘဲ pyplot command သက်သက်ဖြင့်လည်း အသုံးပြုနိုင်သည်။ Object reference များဖြင့် မညွှန်းသောကြောင့် မရှုပ်ထွေးဘဲ ရှင်းလင်း

သည်ဟု ယူဆနိုင်သည်။ memory ပေါ်တွင် လက်ရှိ figure နှင့် axes (current figure and axes) များကို အောက်တွင် ဖော်ပြထားသည့် commands ဖြင့် ရနိုင်သည်။ ကြည့်နိုင်သည်။

```
plt.gcf ( ) # get current figure
plt.gca ( ) # get current axes
```

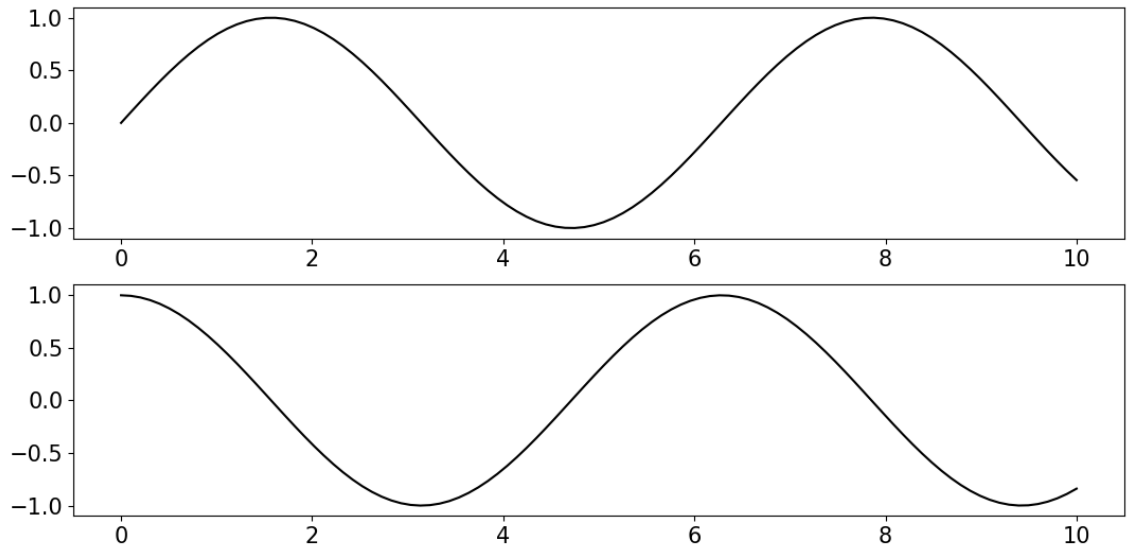
```
Figure(640x480)
```

```
AxesSubplot(0.125,0.11;0.775x0.77)
```

Matplotlib.pyplot သည် MATLAB အတိုင်း ဂရပ်များ ဆွဲနိုင်သည့် tool တစ်ခု ဖြစ်သည်။ ထို MATLAB-style tools တွင် pyplot (plt) interface ပါဝင်သည်။ **Matplotlib.pyplot** ဖြင့် interactive plotting လုပ်သည့်အခါ ထည့်ပေးလိုက်သည့် command အလုပ် လုပ်ပုံကို ချက်ချင်း မြင်နိုင်သည်။ သို့သော် ဒေတာ အရမ်းများပြီး ခက်ခဲ ရှုပ်ထွေးသည့် ဂရပ်များအတွက် မသင့်လျော်ပေ။ **Matplotlib.pyplot** နှင့် မတူသည် တခြားသော interface တစ်မျိုးမှာ **Object-Oriented** interface ဖြစ်သည်။ (နောက်ပိုင်းတွင် ဖော်ပြထားသည်။)

အောက်တွင် ဖော်ပြထားသည့် code သည် Pyplot API ကို အသုံးပြု၍ sine and cosine curve များကို ဆွဲပေးသည့် code ဖြစ်သည်။

```
# create a plot figure
plt.figure()

# create the first of two panels and set current axis
plt.subplot(2, 1, 1)   # (rows, columns, panel number)
plt.plot(x1, np.sin(x1))

# create the second of two panels and set current axis
plt.subplot(2, 1, 2)   # (rows, columns, panel number)
plt.plot(x1, np.cos(x1));
```



gcf() ဆိုတာ လက်ရှိ ဂရပ်ရဲ့ အချက်အလက်(get current figure)ပါ။ အပေါ်မှာက ဂရပ်ရဲ့ gcf() ကို ကြည့်ချင်ရင် plt.gcf() နှင့် ကြည့်နိုင်ပါသည်။

```
# get current figure information
print(plt.gcf())
```

```
Figure(432x288)
<Figure size 432x288 with 0 Axes>
```

gca() ဆိုတာ လက်ရှိ ဂရပ်ရဲ့ အချက်အလက်(get current axis)ပါ။ အပေါ်မှာက ဂရပ်ရဲ့ gcf() ကို ကြည့်ချင်ရင် plt.gcf() နှင့် ကြည့်နိုင်ပါသည်။

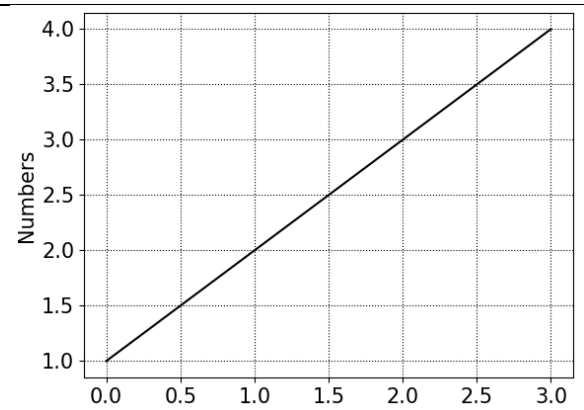
```
# get current axis information
print(plt.gca())
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```

Pyplot နှင့် ဂရပ်ပုံတွေ ထုတ်ရတာ မခက်ပါဘူး။ X ဝင်ရိုး(x-axis)က 0 ကနေ 3အထိ ၊ Y ဝင်ရိုး (x-axis)မှာ 1 ကနေ 4 အထိ ထားပြီးတော့ x data တွေက [0,1,2,3] နှင့် y data တွေက [1,2,3,4] ဖြစ်တဲ့ ပုံဆွဲကြည့်ရအောင်။

ဂရပ်ဆွဲတဲ့အခါ X တန်ဖိုးတွေကို X ဝင်ရိုးပေါ်မှာ ချ ပြီးရင် Y တန်ဖိုးတွေကို Y ဝင်ရိုး ပေါ်မှာ ချပြီးတော့ ဖြတ်မှတ်တွေကို ဆက်ရင် ဂရပ်ပုံ ရသည်။

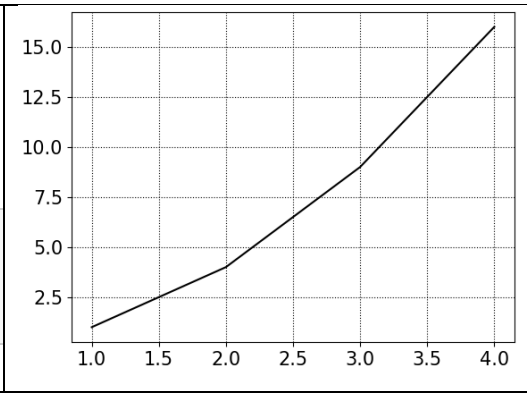
```
import matplotlib.pyplot as plt
x = [0,1,2,3]
y = [1, 2, 3, 4]
plt.plot(x, y)
plt.show()
```



plot() - A versatile command

.plot() သည် နည်းမျိုးစုံနှင့် အသုံးပြုလို့ရသည့် command ဖြစ်သည်။ X တန်ဖိုးတွေ နှင့် Y တန်ဖိုး များထည့် ပေးလျှင်လည်း လက်ခံသည်။ သို့မဟုတ် number များ၊ list များ၊ tuple များ ထည့်ပေး၍လည်း plot လုပ်နိုင်သည်။ ဥပမာ အောက်တွင် list နှစ်ခုထည့်၍လည်း plot လုပ်နိုင်ပုံကို ဖော်ပြထားသည်။

```
plt.plot([1, 2, 3, 4],
         [1, 4, 9, 16])
plt.show()
```



State-machine interface

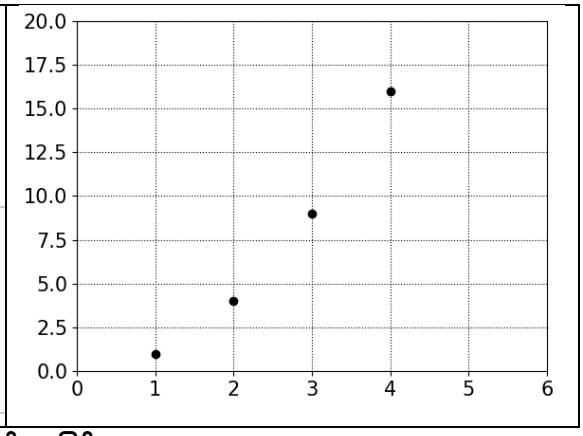
ဂရပ်ဆွဲသူက ဘာမှ မပြောလျှင် (မထည့်ပေးလျှင်) Pyplot သည် သင်လျော့်သည့် figures နှင့် axes များ တည်ဆောက်သွားသည်။ လိုအပ်သည့် စကေးများကိုလည်း အလိုအလျောက်သင့်လျော်အောင် ထည့်ပေးသွားသည်။

8.1 Formatting the style of plot

plt.plot() ထဲတွင် X တန်ဖိုးနှင့် Y တန်ဖိုးများ(x and y arguments)သာမက နောက်ထပ် လိုင်းအမျိုးအစား (line type)၊ လိုင်းအရောင်(line color) စသည့် argument များ ထည့်ပေးနိုင်သည်။ လိုင်းအမျိုးအစား (line type) ကို လိုင်းစတိုင်ဟုလည်းခေါ်သည်။ လိုင်းစတိုင်နှင့် အရောင်ကို ပေါင်း၍လည်း ရေးနိုင်သည်။ (concatenate a color string with a line style string)။

ဥပမာ- 'b-' သည် solid blue line ဖြစ်သည်။ b သည် အပြာရောင်() ဖြစ်ပြီး - သည် solid line ဖြစ်သည်။ "ko" သည် အနက်ရောင် အစက်(black circle) ဖြစ်သည်။ k သည် အနက်ရောင် ဖြစ်သည်။

```
plt.plot([1, 2, 3, 4],
         [1, 4, 9, 16], 'ko')

plt.axis([0, 6, 0, 20])
plt.show()
```

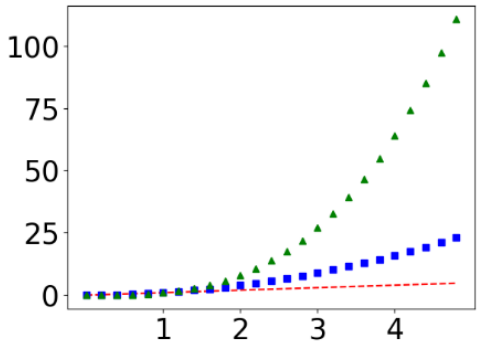


8.2 ဂရပ်စကေး ၏ အများဆုံး နှင့် အနည်းဆုံးကို သတ်မှတ်ပေးခြင်း

axis() command ဖြင့် ဂရပ်၏ စကေး အနည်းဆုံး အများဆုံး သတ်မှတ်ပေးနိုင်သည်။ X ဝင်ရိုးနှင့် Y ဝင်ရိုး နှစ်ခု ရှိသောကြောင့် [xmin, xmax, ymin, ymax] ဖြင့် သတ်မှတ်ပေးနိုင်သည်။

Matplot တွင် NumPy arrays နှင့် တွဲ၍ အသုံးပြုနိုင်သည်။ အောက်တွင် ဂရပ်လိုင်းများကို လိုင်းစတိုင် အမျိုးမျိုးဖြင့် ဆွဲပြထားသည်။

```
# evenly sampled time at 200ms
# intervals
t = np.arange(0., 5., 0.2)
# red dashes, blue squares and green
triangles
plt.plot(t, t, 'r--',
         t, t**2, 'bs',
         t, t**3, 'g^')
plt.show()
```



`plt.plot(t, t, 'r--')` သည် linear curve ကို အနီရောင် dashes လိုင်းဖြင့် ဖော်ပြရန် ဖြစ်သည်။

`plt.plot(t, t**2, 'bs')` သည် quadratic curve လိုင်း အပြာရောင်လေးထောင့်တုံး(blue squares) ဖြင့် ဖော်ပြရန် ဖြစ်သည်။

`plt.plot(t, t**3, 'g^')` သည် cubic curve ကို အစိမ်းရောင် တြိဂံ(green triangles)ဖြင့် ဖော်ပြရန် ဖြစ်သည်။

အပေါ်မှ လိုင်း(၃)လိုင်းကို plt.plot() ထဲတွင် အားလုံးကို ပေါင်း၍ ရေးနိုင်သည်။

```
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
```

9. Object-Oriented API

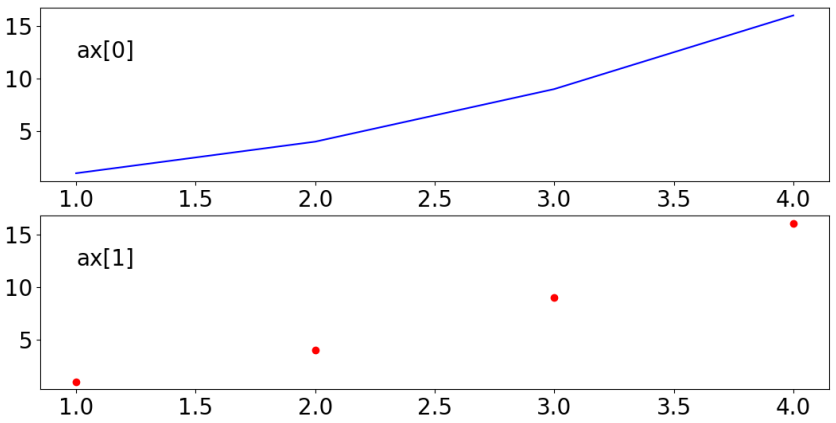
ခက်ခဲ ရှုပ်ထွေးသည့် ဂရပ်များအတွက် **Object-Oriented API** ကို အသုံးပြုသည်။ Pyplot API တွင် "active" figure or axes ပေါ်တွင် အလုပ်လုပ်သည်။

Object-Oriented API တွင် plotting functions များသည် Figure and Axes objects ကို တိတိကျကျ သတ်မှတ်ပြီး method များကို သုံးသည်။

Figure သည် ဂရပ်ပုံထဲတွင်ပါဝင်မည့်အရာများ(plot elements) ကို ထည့်ရမည့် top level container ဖြစ်သည်။ **Figure** object သည် ဂရပ်ငယ်(**Axes** or subplot)များ ထည့်ရန် သေတ္တာအဖြစ်(box-like container) မှတ်ယူနိုင်သည်။

Axes ဆိုသည်မှာ **Figure** ထဲတွင် ရှိနေသည့် ဂရပ်ငယ်ကလေးများ တစ်ခုချင်း(individual plot) ကို ဆိုလိုသည်။ **Axes** object ထဲတွင် ဂရပ်ငယ်ကလေးများ၏ axis, tick marks, lines, legends, title and text-boxes စသည်တို့ ပါဝင်ကြသည်။ အောက်တွင် Object-Oriented API နည်းဖြင့် ရေးဆွဲထားသည့် sine and cosine curve များ ဖြစ်သည်။

```
import matplotlib.pyplot as plt
# ဂရပ်(၂)ပုံဆွဲရန် fig တည်ဆောက်သည်။ ax[0]သည် ပထမ subplots ဖြစ်သည်။
fig, ax = plt.subplots(2)

# ပထမ subplots အတွက် plot() method ဖြစ်သည်။ 'b-'သည် solid blue လိုင်းဖြစ်သည်။
ax[0].plot([1, 2, 3, 4], [1, 4, 9, 16], "b-")

# ax[1]သည် ဒုတိယ subplots ဖြစ်သည်။ "ro" သည် အနီရောင်အစက်(red circle) ဖြစ်သည်။
ax[1].plot([1, 2, 3, 4], [1, 4, 9, 16], "ro")
plt.show()
```



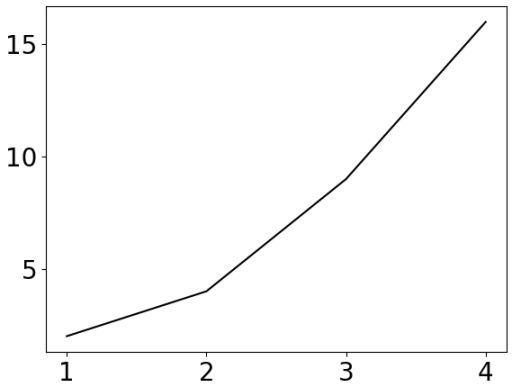
9.1 Objects and Reference

Object Oriented API တွင် အရာအားလုံးကို object များ အဖြစ်ပြုလုပ်ရခြင်း၏ အဓိက အကြောင်းမှာ object များ ဖြစ်သွားလျှင် function များ method များကို အလွယ်တကူ အသုံးပြုနိုင်သောကြောင့် ဖြစ်သည်။ figure တစ်ခုတွင် subplot များ ပိုများနေပါက object များအဖြစ် ပြုလုပ်ပြီး function များ method များကို လွယ်ကူလျှင်မြန်စွာ ပြုလုပ်နိုင်သည်။ **Object Oriented API** ၏ figure နှင့် axes တို့ စတင် အသုံးပြု ကြည့်ရအောင်။

add_axes method (Object-Oriented API)

add_axes method ဆိုသည်မှာ **Figure** object အတွင်းသို့ ဂရပ်ငယ်(subplot) များ ထည့်ခြင်းဖြစ်သည်။ ဂရပ်ငယ်(subplot)ထည့်ရန် လေးထောင်ဘောင်၏ position [left, bottom] နှင့် width, height တို့ကို ထည့်ပေးရသည်။ `fig.add_axes([left, bottom, width, height]`

```
import matplotlib.pyplot as plt
fig = plt.figure()
x2 = [1, 2, 3, 4]
y2 = [2, 4, 9, 16]
#(0.1, 0.1) နေရာတွင်ထည့်မည်။
axes = fig.add_axes
         ([0.1, 0.1, 0.8, 0.8])

axes.plot(x2, y2)
plt.show()
```



`axes = fig.add_axes([0.1, 0.1, 0.8, 0.8])` တွင် (0.1, 0.1)သည် နေရာဖြစ်သည်။ 0.8, 0.8 သည် ဂရပ်ငယ်၏ width နှင့် height ဖြစ်သည်။

9.2 Figure and Axes

figure နှင့် axes ကို အောက်ပါအတိုင်းလည်း ပြုလုပ်နိုင်သည်။

`fig = plt.figure()` နှင့်
`ax = plt.axes()`

Matplotlib တွင် `figure` သည် plt.Figure class ၏ instance တစ်ခု ဖြစ်သည်။ `figure` သည် axes object များ, graphic object များ, text object များ နှင့် label object များ ပါဝင်သည့် ကွန်တိန်နာ တစ်ခုဖြစ်သည်။

axes သည် plt.Axes class ၏ instance တစ်ခု ဖြစ်သည်။ **axes** သည် suplot များ ရေးဆွဲမည့် လေးထောင့်နေရာတစ်ခု ဖြစ်သည်။ **axes** ထဲတွင် ဂရပ်ပုံနှင့် tick object များ နှင့် label object များ ပါဝင်သည်။ ပုံမှန်အားဖြင့် တည်ဆောက်လိုက်သည့် figure instance ကို fig သို့မဟုတ် my_fig ဟု နာမည်ပေးလေ့ရှိသည်။

axes instance များကို ax1, ax2 သို့မဟုတ် ax[1], ax[1], ax[3] စသည်ဖြင့် နာမည်ပေးလေ့ရှိသည်။

```
fig = plt.figure()
ax = plt.axes()
```

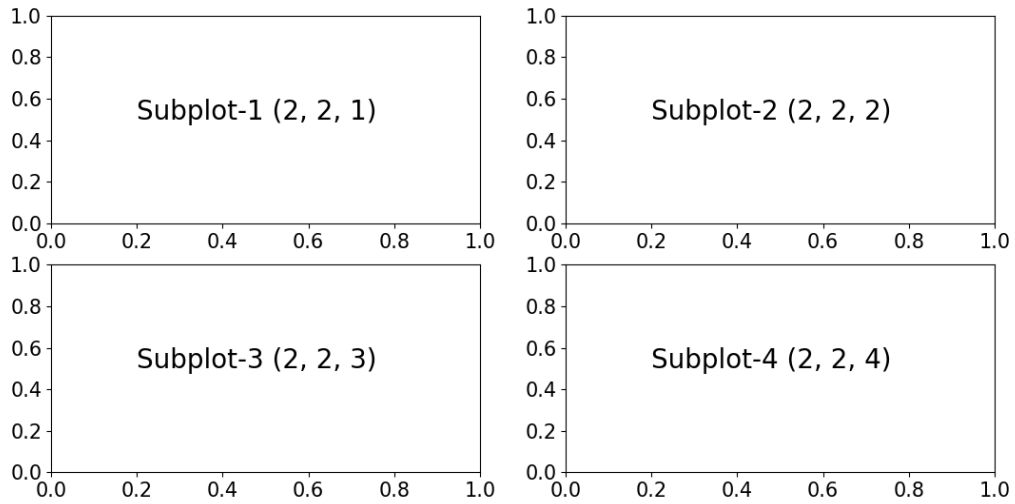
10. Figure and Subplots

Matplotlib တွင် ဂရပ်ပုံ(plot)များသည် Figure object အထဲတွင် တည်ရှိကြသည်။ အောက်ပါအတိုင်း figure instance တစ်ခုကို ပြုလုပ်နိုင်သည်။

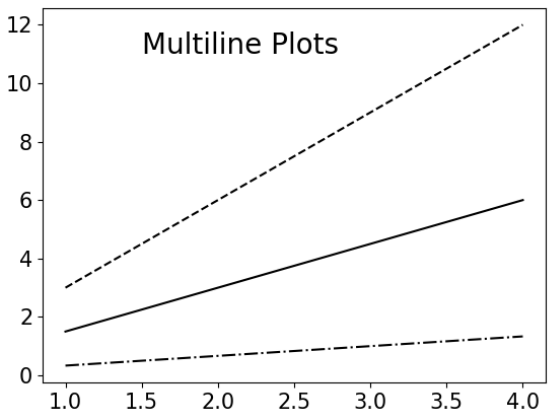
```
fig = plt.figure()
```

fig.add_subplot() ကို သုံး၍ fig ထဲသို့ subplot များ ထည့်မည်။ .add_subplot() ၏ လက်သည်းကွင်း() ထဲတွင် ဂရပ်ငယ်(subplot)များ၏ အချက်အလက်များ ထည့်ပေးရသည်။

```
import matplotlib.pyplot as plt
fig = plt.figure()
ax1 = fig.add_subplot(2, 2, 1)  # 22-1 --> 2 row, 2 column မှ Subplot-1
ax2 = fig.add_subplot(2, 2, 2)  # 22-1 --> 2 row, 2 column မှ Subplot-2
ax3 = fig.add_subplot(2, 2, 3)  # 22-1 --> 2 row, 2 column မှ Subplot-3
ax4 = fig.add_subplot(2, 2, 4)  # 22-1 --> 2 row, 2 column မှ Subplot-4
```




```
import matplotlib.pyplot as plt
import numpy as np

x4 = np.array(range(1, 5))

plt.plot(x4,  x4* 1.5, "k-")
plt.plot(x4, x4*3 ,"k--")
plt.plot(x4, x4/3.0,"k-.")

plt.show()
```

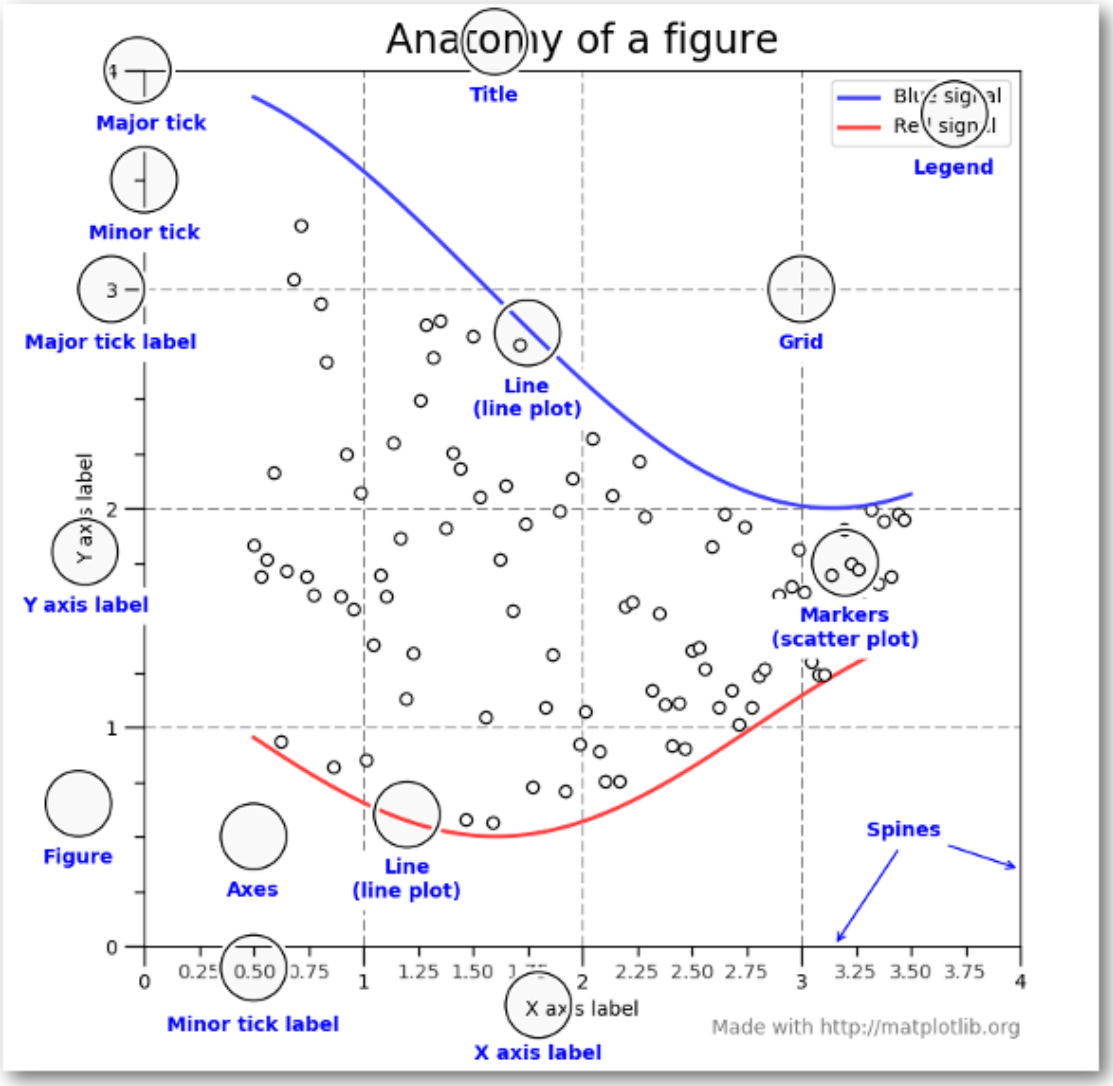


12. Multiline Plots

Multiline plot ဆိုသည် ဂရပ်တစ်ခုပေါ်တွင် လိုင်းများစွာထည့်သွင်း ဖော်ပြခြင်း ဖြစ်သည်။ အောက်တွင် multiline plot တစ်ခုကို ဥပမာ အဖြစ် ဖော်ပြထားသည်။

13. Parts of a Plot

Plot တစ်ခု တွင် title, legend, grid, axis and labels စသည့် အစိတ်အပိုင်းများစွာ ပါဝင်သည်။ အစိတ်အပိုင်းတစ်မျိုးစီနှင့် ၄င်းတို့၏ နာမည်များကို အောက်တွင် ဖော်ပြထားသည်။



14. Saving the Plot

ဆွဲထားသည့်ဂရပ်(figure)ကို **savefig()** command ကို အသုံးပြု၍ format အမျိုးမျိုးဖြင့် သိမ်းဆည်း(save)နိုင်သည်။ `fig.savefig('fig1.png')` သည် .png format ဖြင့်သိမ်းဆည်းသည်။

```
# Saving the figure
fig.savefig('plot1.png')
```

14.1 Supported file formats

ဖိုင်သိမ်းဆည်းနိုင်သည့် format များကို `fig.canvas.get_supported_filetypes()` ဖြင့် ဖတ်ယူနိုင်သည်။

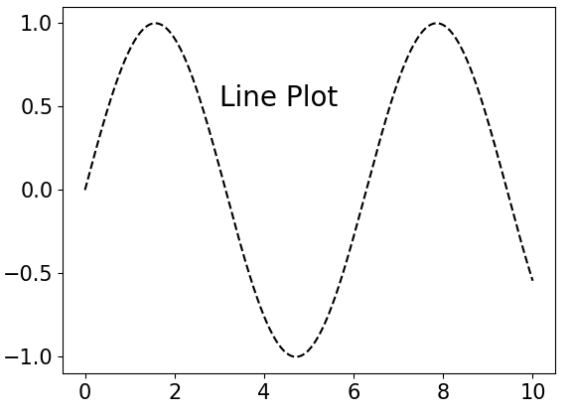
```
# Explore supported file formats
fig.canvas.get_supported_filetypes()
```

```
{'ps': 'Postscript',
 'eps': 'Encapsulated Postscript',
 'pdf': 'Portable Document Format',
 'pgf': 'PGF code for LaTeX',
 'png': 'Portable Network Graphics',
 'raw': 'Raw RGBA bitmap',
 'rgba': 'Raw RGBA bitmap',
 'svg': 'Scalable Vector Graphics',
 'svgz': 'Scalable Vector Graphics',
 'jpg': 'Joint Photographic Experts Group',
 'jpeg': 'Joint Photographic Experts Group',
 'tif': 'Tagged Image File Format',
 'tiff': 'Tagged Image File Format'}
```

15. Line Plot

simple sinusoid line plot တစ်ခုကို နမူနာအဖြစ်ဖော်ပြထားသည်။

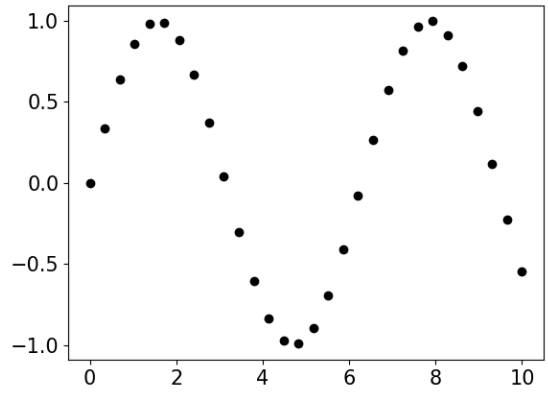
```
import matplotlib.pyplot as plt
import numpy as np
# Create figure and axes first
fig = plt.figure()
ax = plt.axes()

# Declare a variable x 5
x5 = np.linspace(0, 10, 1000)
# Plot the sinusoid function
ax.plot(x5, np.sin(x5), 'b--');
plt.show()
```



16. Scatter Plot

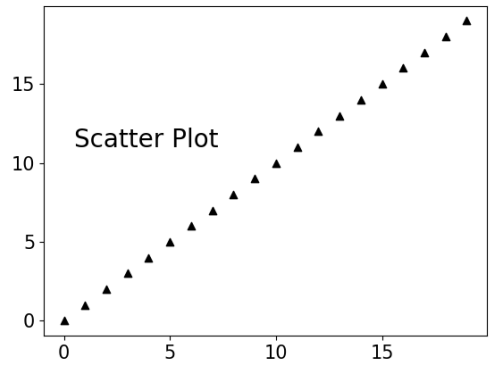
အသုံးများသည့် နောက်ဂရပ်တစ်မျိုးမှာ scatter plot ဖြစ်သည်။ `plt.plot()` ကို သုံး၍လည်း Scatter Plot ဆွဲနိုင်သည်။

```
import matplotlib.pyplot as plt
import numpy as np
x7 = np.linspace(0, 10, 30)

y7 = np.sin(x7)

plt.plot(x7, y7, 'o',
         color = 'black')
plt.show()
```



.scatter() ဖြင့် scatter plot ဆွဲခြင်း

```
import matplotlib.pyplot as plt
x7 = range(0,20)
x8 = range(0,20)

plt.scatter(x7, x8, marker='^',
            color = 'k' )
plt.show()
```

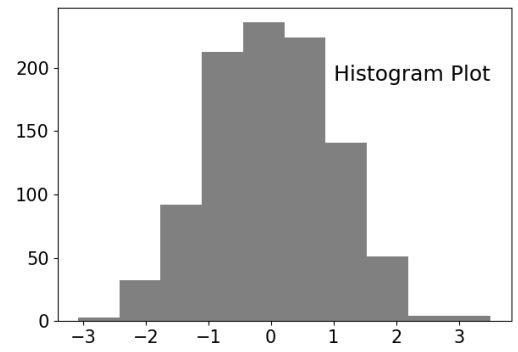


marker='^' သည် တြိဂံပုံ marker များကို ဆိုလိုသည်။ color = 'k' သည် အနက်ရောင် ဖြစ်သည်။

17. Histogram

ဖြစ်ပေါ်သည့် အကြိမ်အရေအတွက် သို့မဟုတ် ပါဝင်နေသည့် အရေအတွက်ကို ဖော်ပြရန်(graphical display of frequencies)အတွက် Histogram chart ကို သုံးသည်။ ဘားဂရပ်မှာကဲ့သို့ bar များဖြင့် ဖော်ပြသည်။ Histogram chart ကို ဆွဲရန်အတွက် bins ဆိုသည့် ဝေါဟာရတစ်ခုကို နားလည်ထားရန်လိုသည်။ bin ဆိုသည်မှာ ဖော်ပြလိုသည့် အတိုင်းအတာ("intervals") အမျိုးအစား("classes") သို့မဟုတ် "buckets" ဖြစ်သည်။ ဥပမာ - အသက်(၁၀)နှစ်အောက်၊ (၁၀) မှ အသက်(၂၀)နှစ်အတွင်း၊ (၂၀) မှ အသက်(၃၀) အတွင်း စသည်တို့သည် " age intervals" bin ဖြစ်သည်။ အောက်တွင် **plt.hist()** function ဖြင့် histogram ဂရပ်တစ်ခုကို ရေးဆွဲထားသည်။

```
import matplotlib.pyplot as plt
import numpy as np

data1 = np.random.randn(1000)

plt.hist(data1, color = 'k')
plt.show()
```

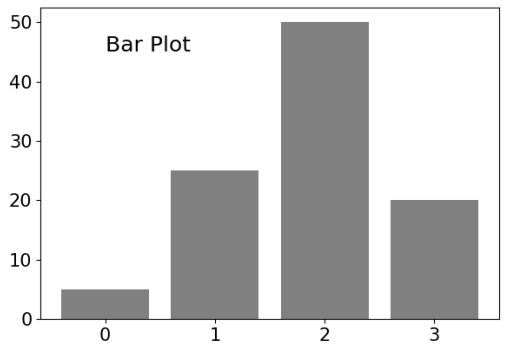


18. Bar Chart

တန်ဖိုးများကို အလွယ်တကူ ကြည့်ရှုနိုင်ရန် ထင်သာ မြင်သာ ရှိရန်အတွက် ဘား(bar)များ၏ အရှည်ဖြင့် ဖော်ပြထားသည့် ဂရပ်ကို ဘားဂရပ်ဟုခေါ်သည်။ ဘားအရှည်သည် တစ်မျိုးချင်းစီ၏ ပမာဏ ဖြစ်သည်။ တစ်မျိုးချင်းစီ တွင်လည်း ယခင်တန်ဖိုး နှင့် ယခုတန်ဖိုး စသည်ဖြင့် နှိုင်းယှဉ်ရန် ဘားဂရပ်ကို အသုံးပြုသည်။ vertical bar chart နှင့် horizontal bar chart ဘားဂရပ် နှစ်မျိုးရှိသည်။

plt.bar() function ကို အသုံးပြု၍ အောက်တွင် ဘားဂရပ် တစ်ခုကို နမူနာဆွဲပြထားသည်။

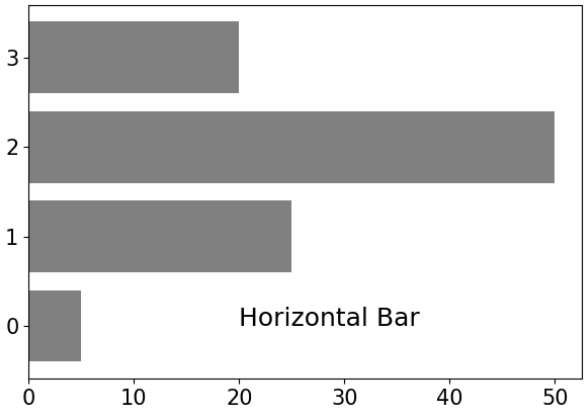
```
import matplotlib.pyplot as plt

data2 = [5., 25., 50., 20.]

plt.bar(range(len(data2)), data2)
plt.show()
```



19. Horizontal Bar Chart

plt.barh() function ကို အသုံးပြု၍ horizontal bar chart ရေးဆွဲနိုင်သည်။ **plt.bar()** function နှင့် အတူတူပင်ဖြစ်သည်။ အလျားလိုက်(horizontal)နှင့် ဒေါင်လိုက်သာ ကွာခြားသည်။

```
import matplotlib.pyplot as pl
t

data2 = [5., 25., 50., 20.]

plt.barh(range(len(data2)),
         data2, color = 'k')

plt.show()
```

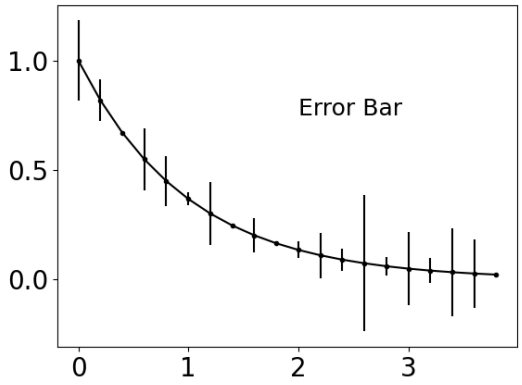


20. Error Bar Chart

လက်တွေ့စမ်းသပ်မှုများ ပြုလုပ်သည့်အခါ ဖြစ်ပေါ်တတ်သည့်အမှားများကို ဖော်ပြရန်အတွက် ကို အသုံးပြုသည်။ လက်တွေ့စမ်းသပ်မှု တစ်ကြိမ်နှင့်တစ်ကြိမ် ဖြစ်ပေါ်သည့် အမှား()ပမာဏ လည်း မတူညီနိုင်ပေ။

errorbar() function ကို အသုံးပြု၍ Error Bar Chart ကို ရေးဆွဲသည်။

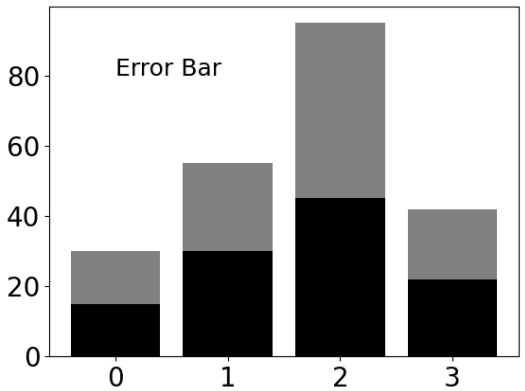
```
import matplotlib.pyplot as plt
import numpy as np
x9 = np.arange(0, 4, 0.2)
y9 = np.exp(-x9)

e1 = 0.1 * np.abs(np.random.randn
                  (len(y9)))

plt.errorbar(x9, y9, yerr = e1,
             fmt = 'k.-')
plt.show();
```



21. Stacked Bar Chart

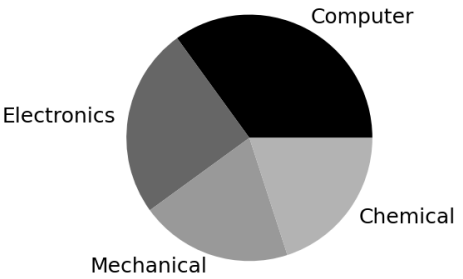
stacked bar chart သည် bar chart တစ်မျိုးဖြစ်သည်။ ထူးခြားချက်မှာ ဘားတစ်ခုနှင့်တစ်ခု ထပ်ထားခြင်းဖြစ်သည်။ stacked bar chart တွင် **bottom** ဟုခေါ်သည့် special parameter တစ်ခုပါဝင်သည်။ stacked bar chart ကို ဘားဂရပ်ဆွဲသည့် **plt.bar()** function ဖြင့်သာ ဆွဲသည်။ ဒေတာနှစ်မျိုးကို ထပ်ထား သောကြောင့် အောက်တွင် အထိုင်ထားမည့် data set ကို bottom keyword ဖြင့် ရွေးချယ်ပေးရမည်။ **bottom** ဟုခေါ်သည့် special parameter သည် အောက်တွင် အထိုင်ထားမည့် data set ကို ရည်ညွှန်းသည်။

```
import matplotlib.pyplot as plt
A = [15., 30., 45., 22.]
B = [15., 25., 50., 20.]
z2 = range(4)

plt.bar(z2, A, color = 'k')
plt.bar(z2, B, color = 'gray',
        bottom = A)

plt.show()
```



22. Pie Chart

Pie chart သည် စက်ဝိုင်းပုံသဏ္ဌာန်ထဲတွင် အစိတ်အပိုင်းများဖြင့် ဖော်ပြသည် ဂရပ်ဖြစ်သည်။ arc length သည် ဖော်ပြလိုသည် အမျိုးအစား၏ ပမာဏ ဖြစ်သည်။ arc length ရှည်လေ ပမာဏ များလေ ဖြစ်သည်။ အမျိုးအစားများ တစ်ခုနှင့် တစ်ခု နှိုင်းယှဉ်ပြီး ဖော်လိုသည့်အခါ အသုံးပြုသည်။ အထူး သဖြင့် ရာခိုင်နှုန်း ဖော်ပြလိုသည့်အခါတွင် သုံးသည်။



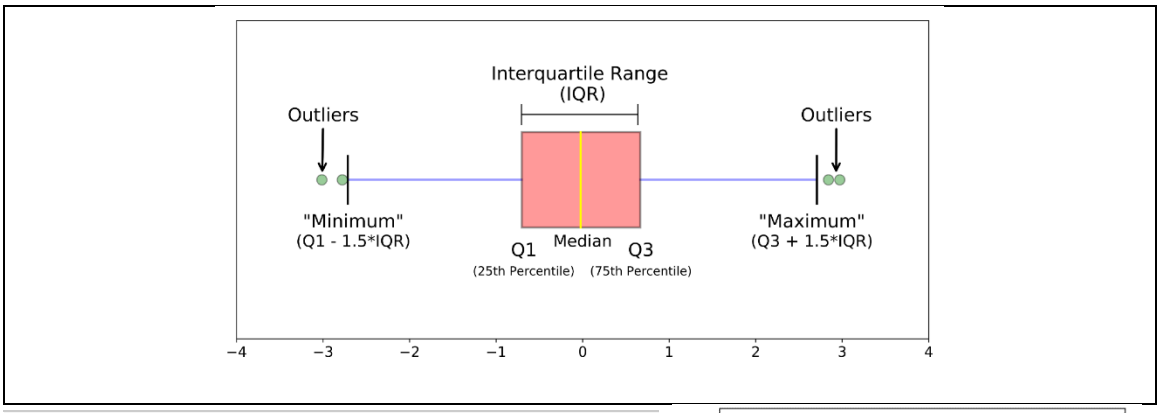
pie chart ဆွဲရန်အတွက် **pie()** function ကို အသုံးပြုသည်။

```
import matplotlib.pyplot as plt
plt.figure(figsize=(7,7))
x10 = [35, 25, 20, 20]
```

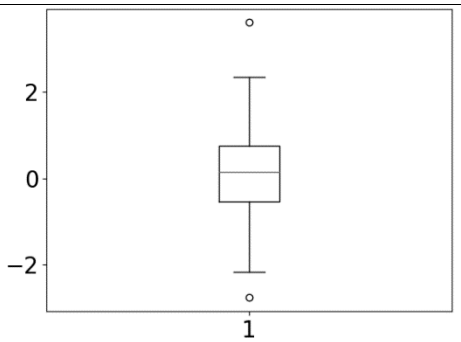
```
labels = ['Computer', 'Electronics', 'Mechanical', 'Chemical']
plt.pie(x10, labels=labels);

plt.show()
```

23. Boxplot

ဒေတာများ median တန်ဖိုး, quartiles တန်ဖိုး, maximum တန်ဖိုး နှင့် minimum တန်ဖိုး တို့ကို ဖော်ပြရန် အတွက် Boxplot chart ကို အသုံးပြုသည်။ **boxplot()** function ကို အသုံးပြု၍ ရေးဆွဲသည်။




```
import matplotlib.pyplot as plt
data3 = np.random.randn(100)

plt.boxplot(data3)

plt.show();
```

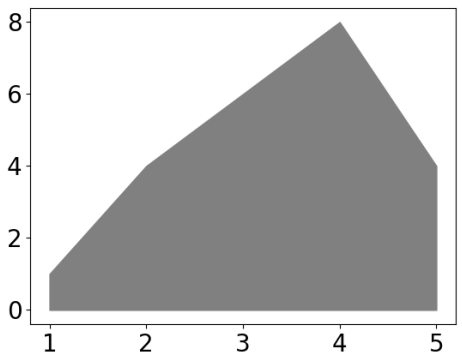


boxplot() function သည် ထည့်ပေးသည့် ဒေတာများကို မူတည်၍ mean, median and other statistical quantities များကို တွက်၍ ဖော်ပြပေးသည်။

24. Area Chart

Area Chart သည် **Line Chart** နှင့် ခပ်ဆင်ဆင်တူညီသည်။ x-axis နှင့် ဂရပ်လိုင်း(line) နှစ်ခုအကြားကို အရောင်ချယ်ထားသည်။

```
import matplotlib.pyplot as plt

# Create some data
x12 = range(1, 6)
y12 = [1, 4, 6, 8, 4]

# Area plot
plt.fill_between(x12, y12)
plt.show()
```



stackplot function ကို သုံး၍လည်း Area chart ကို ရေးဆွဲနိုင်သည်။ ဥပမာ -`plt.stackplot(x12, y12)` ။ **fill_between()** function သည် မိမိလိုသလို customization လုပ်နိုင်သောကြောင့် ပိုအဆင်ပြေသည်။

25. Contour Plot

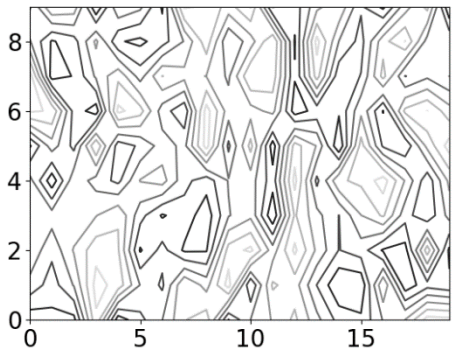
three-dimensional data များကို two dimensions contours သို့မဟုတ် color-coded region များဖြင့် ဖော်ပြနိုင်သည်။ three-dimensional data များကို two dimension တွင် ဖော်ပြလိုသည့်အခါ Contour plot များကို အသုံးပြုကြသည်။

ကွန်တိုလိုင်း(Contour lines)များကို လယ်ဗယ်လိုင်း(level line)များ သို့မဟုတ် အိုင်ဆိုလိုင်း(isoline) များဟုလည်း ခေါ်သည်။

Contour lines ပေါ်တွင် ရှိသည့်တန်ဖိုးများသည် တူကြသည်။ (constant value များ ဖြစ်ကြသည်။) မိုးလေဝသ မြေပုံများတွင် ပူချိန်တူညီသည့်လိုင်းများ၊ လေဖိအားတူညီသည့်လိုင်းများ(pressure) မိုးရေချိန် လက္ခဏာတူညီသည့် လိုင်းများ၊ လေတိုက်နှုန်းတူညီသည့် လိုင်းများ စသည်တို့ကို ကွန်တိုလိုင်းများ ဖြင့် ဖော်ပြလေ့ရှိသည်။

plt.contour() function ကို အသုံးပြု၍ **Contour plot** ကို တည်ဆောက်နိုင်သည်။ **contour()** function သည် ကွန်တိုလိုင်း(contour line)များကို ဆွဲပေးသည်။

```
import matplotlib.pyplot as plt
import numpy as np

# Create a matrix
matrix1 = np.random.rand(10, 20)

cp = plt.contour(matrix1)

plt.show()
```



26. Styles with Matplotlib Plots

Matplotlib တွင် ဂရပ်များကို စတိုင်အမျိုးမျိုးဖြင့် ဆွဲ(plot)နိုင်သည်။ ရရှိနိုင်သည့် စတိုင်(available style)များကို `print(plt.style.availabe)` command ဖြင့် ဖတ်နိုင်သည်။

```
# ရရှိနိုင်သည့် စတိုင်(available style)များကို ကြည့်ရန်
print(plt.style.available)
```

```
['classic', 'seaborn-poster', 'dark_background', 'seaborn-ticks',
'seaborn-muted', 'seaborn-deep', 'fivethirtyeight', 'grayscale',
'seaborn-notebook', 'ggplot', 'bmh', 'seaborn-paper', 'fast', 'tableau-
colorblind10', 'seaborn-bright', 'seaborn-colorblind', 'seaborn-pastel',
'seaborn', 'seaborn-talk', 'seaborn-white', 'seaborn-dark', 'seaborn-
whitegrid', 'seaborn-dark-palette', 'Solarize_Light2', '_classic_test',
'seaborn-darkgrid']
```

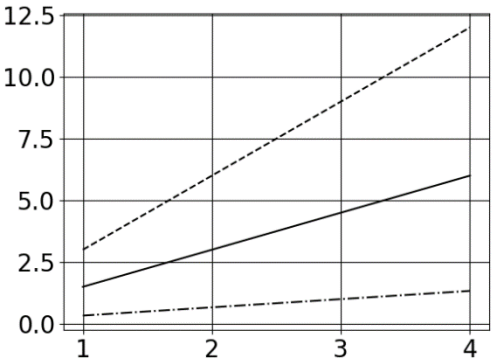
'seaborn-bright' စတိုင်ကို သုံးရန် ဤကဲ့သို့ `plt.style.use('seaborn-bright')` ရေးသားနိုင်သည်။ ဥပမာ -

```
# Set styles for plots
plt.style.use('seaborn-bright')
```

27. Grid လိုင်းများ ထည့်ခြင်း(Adding a grid)

plot တွင် reference system လိုအပ်ပါက grid လိုင်းများ ထည့်ပေးရသည်။ grid လိုင်းများကြောင့် ဂရပ်သည် ဖတ်ရန်အတွက် ပိုလွယ်ကူလာသည်။ **grid()** function ကို သုံး၍ grid လိုင်းများ ထည့်နိုင်သည်။ လက်သည်းကွင်းထဲတွင် True သို့မဟုတ် False ဆိုသည့် Boolean value ထည့်ပေးရသည်။ True သည် grid လိုင်းများကို ဖော်ပြပေးရန် ဖြစ်သည်။

```
import matplotlib.pyplot as plt
import numpy as np
x15 = np.arange(1, 5)

plt.plot(x15,  x15* 1.5, "k-")
plt.plot(x15, x15*3 ,"k--")
plt.plot(x15, x15/3.0,"k-.")
plt.grid(True)
plt.show()
```



28. Handling axes

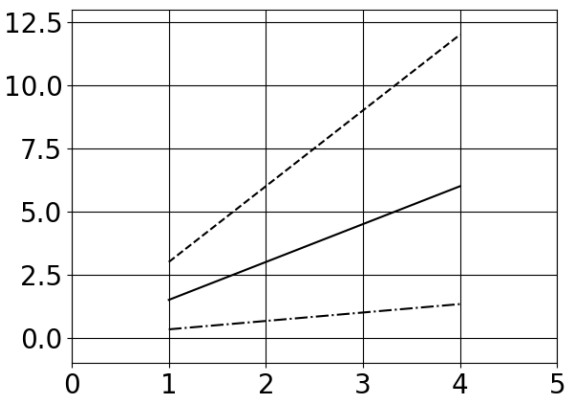
Matplotlib သည် ဒေတာများအပေါ်မူတည်၍ X ဝင်ရိုးနှင့် Y ဝင်ရိုး limit များကို အလိုအလျောက် သတ်မှတ်ပေး(set လုပ်ပေး)သည်။ axes limit ဆိုသည်မှာ X ဝင်ရိုးနှင့် Y ဝင်ရိုး တို့၏ အများဆုံး(max)နှင့် အနည်းဆုံး (min) တန်ဖိုးများ ဖြစ်သည်။ **axis()** function ကို အသုံးပြု၍ မိမိကြိုက်နှစ်သက်သည့် ပမာဏကို set လုပ်နိုင်သည်။

```
plt.axis([xmin, xmax, ymin, ymax]
```

အောက်တွင် X ဝင်ရိုး၏ အနည်းဆုံး တန်ဖိုး (min=0) နှင့် အများဆုံး တန်ဖိုး (max = 5) ၊ Y ဝင်ရိုး အနည်းဆုံး တန်ဖိုး(min= -1)နှင့် အများဆုံး တန်ဖိုး(max = 13) အသီးသီး သတ်မှတ်(set)ထားသည်။

ဥပမာ - `plt.axis([0, 5, -1, 13])`

```
import matplotlib.pyplot as plt
import numpy as np
x15 = np.arange(1, 5)

plt.plot(x15,  x15* 1.5, "k-")
plt.plot(x15, x15*3 ,"k--")
plt.plot(x15, x15/3.0,"k-.")
plt.grid(True)
# [xmin, xmax, ymin, ymax]
plt.axis([0, 5, -1, 13])
plt.show()
```



Matplot က သတ်မှတ်ပေးထားသည့် axis limit များကို သိလိုပါက **plt.axis()** ဖြင့် print ထုတ်နိုင်သည်။ (If we execute **axis()** without parameters, it returns the actual axis limits.)

```
print(plt.axis())   # shows the current axis limits values
```

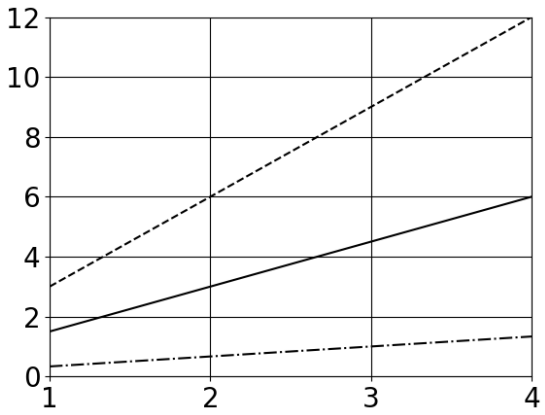
```
(0.85, 4.15, -0.25000000000000006, 12.583333333333334)
```

xmin = minimum limits of X axis ymin = minimum limits of Y axis

xmax = maximum limits of X axis ymin = maximum limits of Y axis

X ဝင်ရိုးနှင့် Y ဝင်ရိုး limit များကို သီးခြား သတ်မှတ်()လိုပါက `xlim()` နှင့် `ylim()` function ကို အသုံးပြု နိုင်သည်။ အောက်တွင် `plt.xlim([1.0, 4.0])` နှင့် `plt.ylim([0.0, 12.0])` တို့ကို ဥပမာ အဖြစ် ပြထားသည်။

```
import matplotlib.pyplot as plt
import numpy as np
x15 = np.arange(1, 5)

plt.plot(x15,  x15* 1.5, "k-")
plt.plot(x15, x15*3 ,"k--")
plt.plot(x15, x15/3.0,"k-.")
plt.grid(True)
plt.xlim([1.0, 4.0])
plt.ylim([0.0, 12.0])
plt.show()
```



29. X ဝင်ရိုးနှင့် Y ဝင်ရိုး အပေါ်က အမှတ်ကလေးတွေ(Handling X and Y ticks)

X ဝင်ရိုးနှင့် Y ဝင်ရိုးအတွင် ရှိနေသည့် အမှတ်ကလေးများကို horizontal ticks နှင့် Vertical ticks ဟုခေါ်သည်။ grid line များ ၏ တန်ဖိုးများကို ဖော်ပြသည့် အမှတ်များလည်း ဖြစ်သည်။ တစ်နည်းအားဖြင့် ဂရပ်၏ reference system ဖြစ်သည်။

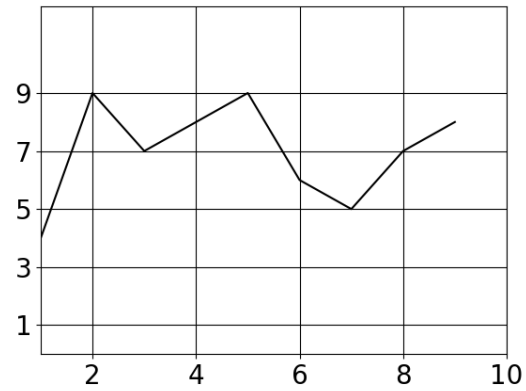
Matplotlib တွင် **xticks()** နှင့် **yticks()** functions ဖြင့် X ဝင်ရိုးနှင့် Y ဝင်ရိုးအတွင် ရှိနေသည့် အမှတ် ကလေးများကို စီမံနိုင်သည်။

Executing with no arguments, the tick function returns the current ticks' locations and the labels corresponding to each of them.

ticks function အတွက် argument (၂)ခု ထည့်ပေး(pass လုပ်)ရသည်။ tick ဖော်ပြပေးရမည့် တည်နေရာ(Locations of the ticks) နှင့် ထိုနေရာတွင် ဖော်ပြရမည့် လေဘယ်(Labels to draw at these locations)တို့ ဖြစ်သည်။ အောက်တွင် ticks function ဥပမာကို ဖော်ပြထားသည်။

`plt.xticks([2, 4, 6, 8, 10])` နှင့် `plt.yticks([1, 3, 5, 7, 9])`

```
import matplotlib.pyplot as plt
u = [5, 4, 9, 7, 8,
     9, 6, 5, 7, 8]
plt.plot(u)
plt.xlim([1.0, 4.0])
plt.ylim([0.0, 12.0])
plt.xticks([2, 4, 6, 8, 10])
plt.yticks([2, 4, 6, 8, 10])
plt.grid(True)
plt.show()
```

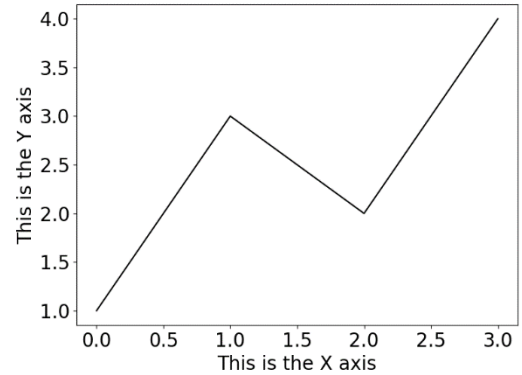


Xticks နှင့် yticks ကို လိုက်၍ grid လိုင်းများ လိုက်ပြောင်းသွားပုံကို သတိပြုပါ။

30. X ဝင်ရိုးနှင့် Y ဝင်ရိုး တို့တွင် လေဘယ်ထိုးခြင်း(Adding labels)

X ဝင်ရိုးနှင့် Y ဝင်ရိုးတို့သည် မည်သည့် အမျိုးအစားများ ဖြစ်ကြောင်းနှင့် ၄င်းတို့၏ တိုင်းတာသည့် ယူနစ်များကို ဖော်ပြပေးရန်လိုသည်။ ဒေတာများသည် မည်သည့်ဒေတာများ ဖြစ်ကြောင်း လေဘယ်များထိုး၍ ဖော်ပြ ပေးသည်။

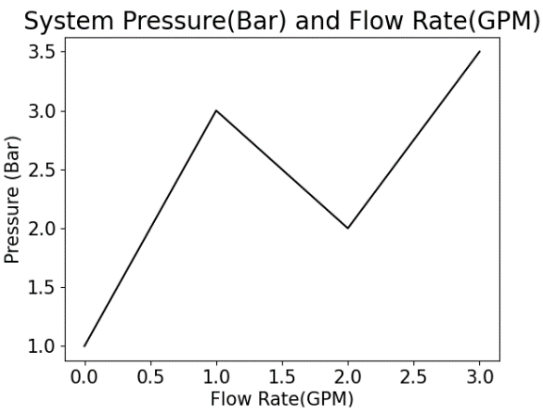
```
import matplotlib.pyplot as plt

plt.plot([1, 3, 2, 4])

plt.xlabel('This is the X axis')
plt.ylabel('This is the Y axis')

plt.show()
```



31. ဂရပ် ခေါင်းစဉ် ထည့်ခြင်း(Adding Title)

ဂရပ်တိုင်းကို ခေါင်းစဉ်(title of a plot) ထည့်ပေးရန် လိုအပ်သည်။ `title()` ကို အသုံးပြုနိုင်သည်။

```
import matplotlib.pyplot as plt
plt.plot([1, 3, 2, 4])

plt.xlabel('Flow Rate(GPM)')
plt.ylabel('Pressure (Bar)')
plt.title('System Pressure(Bar)
           and Flow Rate(GPM)')

plt.show()
```

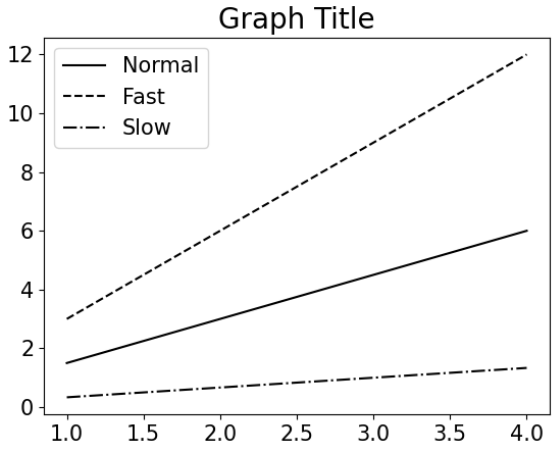


Xlabel နှင့် ylabel တို့ကို ပြောင်းလိုက်သည်။ `plt.title()` ဖြင့် 'System Pressure(Bar) and Flow Rate(GPM) ' ဟုသည် ဂရပ်ခေါင်းစဉ် ထည့်သည်။

32. Legend ထည့်ခြင်း

ဂရပ်တွင် လိုင်းများစွာ ရှိနေလျှင် ထိုလိုင်း(line or curve)များကို **legend** ထိုး၍ ဖော်ပြပေးရသည်။ Legends ထည့်နိုင်သည့် နည်း (၂)မျိုးရှိသည်။ axis object တွင် **legend** method ကို သုံးသည့်နည်း နှင့် ထည့်မည် legend များကို list သို့မဟုတ် tuple တစ်ခု အဖြစ် ပြင်ဆင်ပြီး ထည့်သွင်းနည်း ဖြစ်သည်။

```
import matplotlib.pyplot as plt
x15 = np.arange(1, 5)
fig, ax = plt.subplots()

ax.plot(x15, x15*1.5, "k-")
ax.plot(x15, x15*3.0,"k--")
ax.plot(x15, x15/3.0,"k-.")

plt.title('Graph Title')
ax.legend(['Normal','Fast',
           'Slow'])
plt.show()
```



အပေါ်တွင် ဖော်ပြထားသည့် method သည် MATLAB API နည်း ဖြစ်သည်။ ဤနည်းသည် ဂရပ်လိုင်း(curve)များ ထည့်ခြင်း ဖယ်ထုတ်ခြင်း ပြုလုပ်သည့်အခါ error ဖြစ်နိုင်သည်။

Subplot များကို figure ထဲသို့ ထည့်သည့်အခါ **label** keyword argument ကို သုံးသည့် နည်းသည် ပိုသင့်လျော်သည်။ ဂရပ်လိုင်းများတွင် label ထိုး၍ **legend** ထည့်လျှင် ဂရပ်လိုင်းများထပ်ထည့်သည့်အခါ ဖယ်ထုတ်သည့်အခါ အမှားအယွင်း မဖြစ်နိုင်ပေ။

```
x15 = np.arange(1, 5)

fig, ax = plt.subplots()

ax.plot(x15, x15*1.5, label='Normal')              # 'Normal' legend
ax.plot(x15, x15*3.0, label='Fast')                 # 'Fast' legend
ax.plot(x15, x15/3.0, label='Slow')                 #  'Slow' legend

ax.legend();
```

Legend ကို မည်သည့်နေရာတွင် ထည့်မည်နည်း? ထည့်မည့် နေရာကို ax.legend() ၏ လက်သည်းကွင်း()ထဲတွင် ထည့်ပေးရမည်။ တစ်နည်းအားဖြင့် legend function တွင် ထည့်မည့် နေရာ(loc)ကို optional keyword argument အဖြစ် ထည့်ပေးနိုင်သည်။ ထည့်မည့် နေရာ(loc)ကို 0,1,2, စသည် numerical code များဖြင့် ထည့်ပေးရသည်။ အောက်တွင် အသုံးများသည့် နေရာ(loc)ကို ဖော်ပြထားသည်။

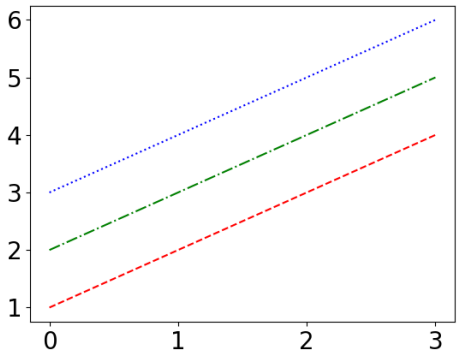
Location String	Location Code	Location String	Location Code
'best'	ax.legend(loc=0)	'center left'	ax.legend(loc=6)
'upper right'	ax.legend(loc=1)	'center right'	ax.legend(loc=7)

Location String	Location Code	Location String	Location Code
'upper left'	ax.legend(loc=2)	'lower center'	ax.legend(loc=8)
'lower left'	ax.legend(loc=3)	'upper center'	ax.legend(loc=9)
'lower right'	ax.legend(loc=4)	'center'	ax.legend(loc=10)
'right'	ax.legend(loc=5)		

33. Control Colours

ဂရပ်လိုင်းများ၏ အရောင်များ နှင့် စတိုင်များကို မိမိ အလိုရှိသလို ရေးဆွဲနိုင်သည်။

```
import matplotlib.pyplot as plt
x16 = np.arange(1, 5)

plt.plot(x16, 'r')
plt.plot(x16+1, 'g')
plt.plot(x16+2, 'b')

plt.show()
```



အောက်တွင်အရောင်(colour names)များနှင့် အရောင်များ၏ အတိုခေါက်နာမည်(colour abbreviations) များကို ဖော်ပြထားသည်။

33.1 Colour abbreviation Colour name

မိမိ ဖော်ပြလိုသည့် ဂရပ်လိုင်းအရောင်များကို နည်းအမျိုးမျိုးဖြင့် ထည့်သွင်းပေးနိုင်သည်။ ဂရပ်လိုင်း အရောင်များကို အတိုခေါက်(colour abbreviations) ထည့်သွင်းရေးသားနိုင်သည်။

b = blue	g = green	m = magenta	w = white
c = cyan	k = black	r = red	y = yellow

အတိုခေါက်(colour abbreviations) မရေးလိုပါက အရောင်နာမည် အပြည့်အစုံကို ရေးသားနိုင်သည်။ ဥပမာ yellow, green, black

အရောင်များကို Hexadecimal string ဖြင့်လည်း ရေးသားနိုင်သည်။ ဥပမာ- ##FF00FF

အရောင်များကို RGB tuples ဖြင့်လည်း ရေးသားနိုင်သည်။ ဥပမာ-(1, 0, 1)

အရောင်များကို Grayscale intensity ဖြင့်လည်း ရေးသားနိုင်သည်။ ဥပမာ '0.7'

34. Control Line Styles

Matplotlib တွင် လိုင်းစတိုင် အမျိုးမျိုးကို ကုဒ်(code)များ ဖြင့် ထည့်သွင်းနိုင်သည်။

Blue dashed line လိုင်းကို “b—“ ဟုလည်းကောင်း green dash-dotted line ကို “g -. “ဟု လည်းကောင်း red dotted line ကို “r: “ ရေးသားနိုင်သည်။

34.1 Style Styles

solid line style	-. dash-dot line style
-- dashed line style	: dotted line style

Single line plot အတွက် 'b-' သည် default format string ဖြစ်သည်။ တစ်နည်းအားဖြင့် ဘာမှ မထည့်ပေးလျှင် အပြာရောင်လိုင်း()'b-' ဖြင့် ဖော်ပြပေးမည်။

34.2 Marker Styles

character	description	character	description
'.'	point marker	**'s'**	square marker
','	pixel marker	**'p'**	pentagon marker
'o'	circle marker	**'*'**	star marker
'v'	triangle_down marker	**'h'**	hexagon1 marker
'^'	triangle_up marker	**'H'**	hexagon2 marker
'<'	triangle_left marker	**'+'**	plus marker
'>'	triangle_right marker	**'x'**	x marker
'1'	tri_down marker	**'D'**	diamond marker
'2'	tri_up marker	**'d'**	thin_diamond marker
'3'	tri_left marker	**'|'**	vline marker
'4'	tri_right marker	**'_'**	hline marker

-End –