

In [2]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

matplotlib ¶

In [78]:

```
x = [1,2,3,4,5,10,15]
```

In [79]:

```
y = np.random.randint(1,10,7)
```

In [9]:

```
y
```

Out[9]:

```
array([7, 3, 2, 5, 1, 4, 8])
```

In [10]:

```
x = np.array(x)
```

In [11]:

```
x
```

Out[11]:

```
array([ 1,  2,  3,  4,  5, 10, 15])
```

Functional method of creating plots

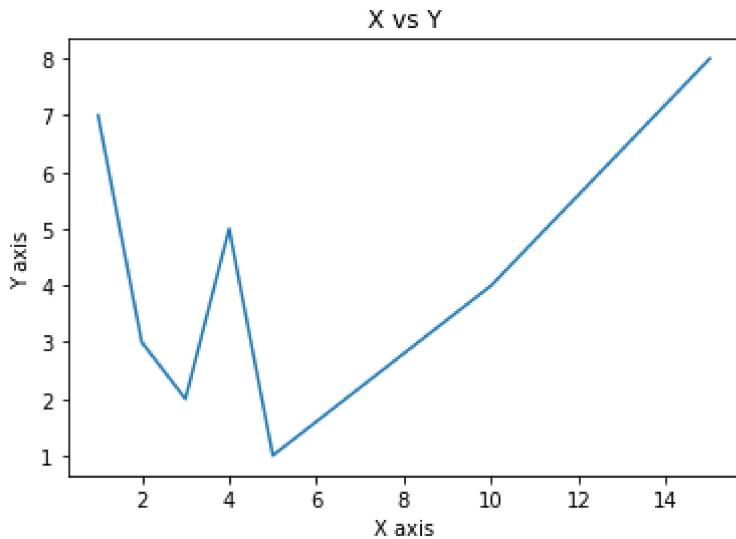
For the functional method directly use the plt library

In [12]:

```
plt.plot(x,y)
plt.xlabel('X axis')
plt.ylabel('Y axis')
plt.title('X vs Y')
```

Out[12]:

```
<matplotlib.text.Text at 0x28d85122630>
```



Creating graphs on the same campus

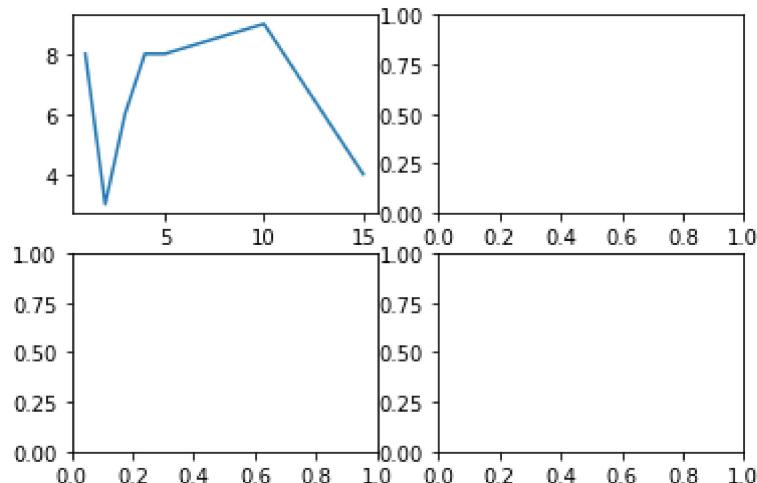
In [24]:

```
plt.subplot(2,2,1)
plt.plot(x,y)

plt.subplot(2,2,2)
plt.subplot(2,2,3)
plt.subplot(2,2,4)
```

Out[24]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c58c7027b8>
```



Object Oriented Method

In this we create an blank object first and then we add our analysis.

In [120]:

```
x = np.linspace(0, 5, 11)
y = x ** 2
z = x ** 3
```

In [41]:

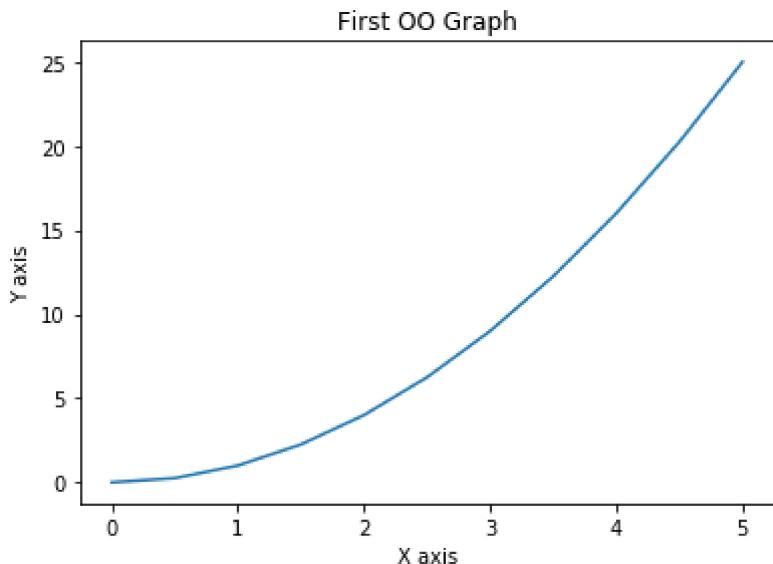
```
fig = plt.figure() # creating a black workspace (canvas)
axes = fig.add_axes([0.1,0.1,0.8,0.8]) #creating the axes by giving dimensions

# The plotting has to be done together with the creation of canvas else the graph won't
# be seen

axes.plot(x,y)
axes.set_xlabel('X axis')
axes.set_ylabel('Y axis')
axes.set_title('First OO Graph')
```

Out[41]:

```
<matplotlib.text.Text at 0x28d868de9e8>
```

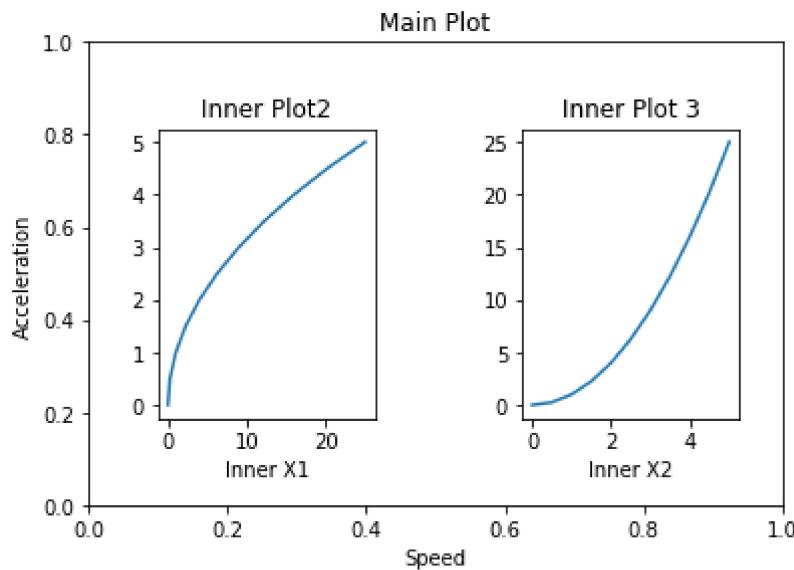


In [77]:

```
canvas = plt.figure()
axes1 = canvas.add_axes([0.1,0.1,0.8,0.8])
axes2 = canvas.add_axes([0.18,0.25,0.25,0.5])
axes3 = canvas.add_axes([0.60,0.25,0.25,0.5])
axes3.plot(x,y)
axes2.plot(y,x)
axes1.set_title('Main Plot')
axes3.set_title('Inner Plot 3')
axes2.set_title('Inner Plot2')
axes1.set_xlabel('Speed')
axes1.set_ylabel('Acceleration')
axes2.set_xlabel('Inner X1')
axes3.set_xlabel('Inner X2')
```

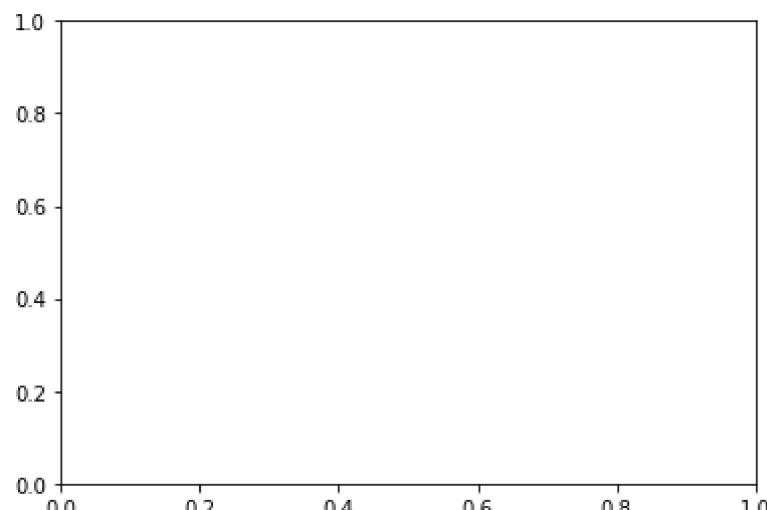
Out[77]:

```
<matplotlib.text.Text at 0x28d8b615320>
```



In [84]:

```
# Manually creating canvas and Label
canvas = plt.figure()
axes = canvas.add_axes([0.1,0.1,0.8,0.8])
```

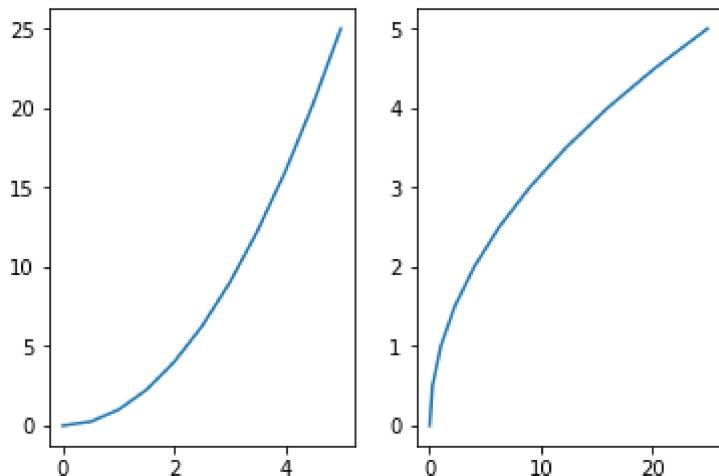


In [115]:

```
# Automatically creating the axes and printing
canvas, axes = plt.subplots(nrows = 1, ncols = 2)
axes[0].plot(x,y)
axes[1].plot(y,x)
# this means that subplots creates an array of numbers(id's) which can be used to plot
# multi graphs
```

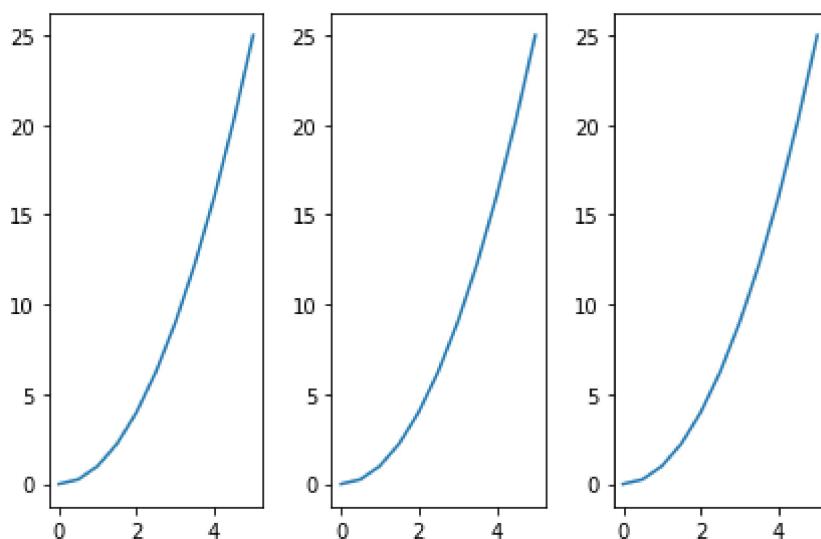
Out[115]:

```
[<matplotlib.lines.Line2D at 0x28d8d125c88>]
```



In [136]:

```
canvas2, axes = plt.subplots(nrows = 1, ncols = 3)
for loop in axes:
    loop.plot(x,y)
    continue
    loop.plot(y,x)
    continue
    loop.plot(x,z)
plt.tight_layout() # use to fix the space issues
```

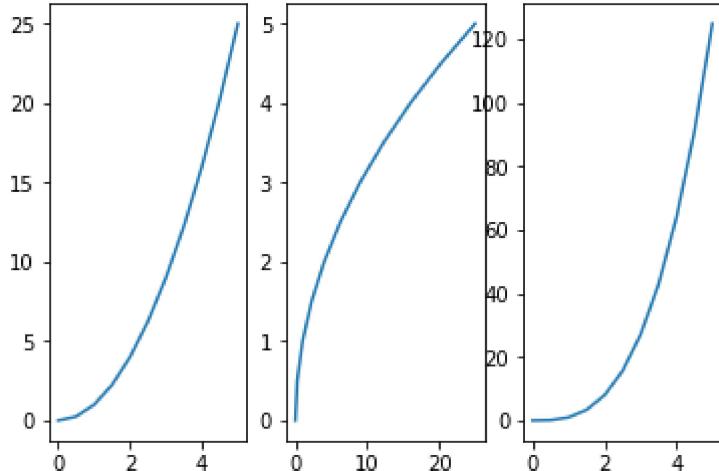


In [137]:

```
canvas2, axes = plt.subplots(nrows = 1,ncols = 3)
canvas2
axes[0].plot(x,y)
axes[1].plot(y,x)
axes[2].plot(x,z)
```

Out[137]:

```
[<matplotlib.lines.Line2D at 0x28d8d2f29e8>]
```



All in one

In [206]:

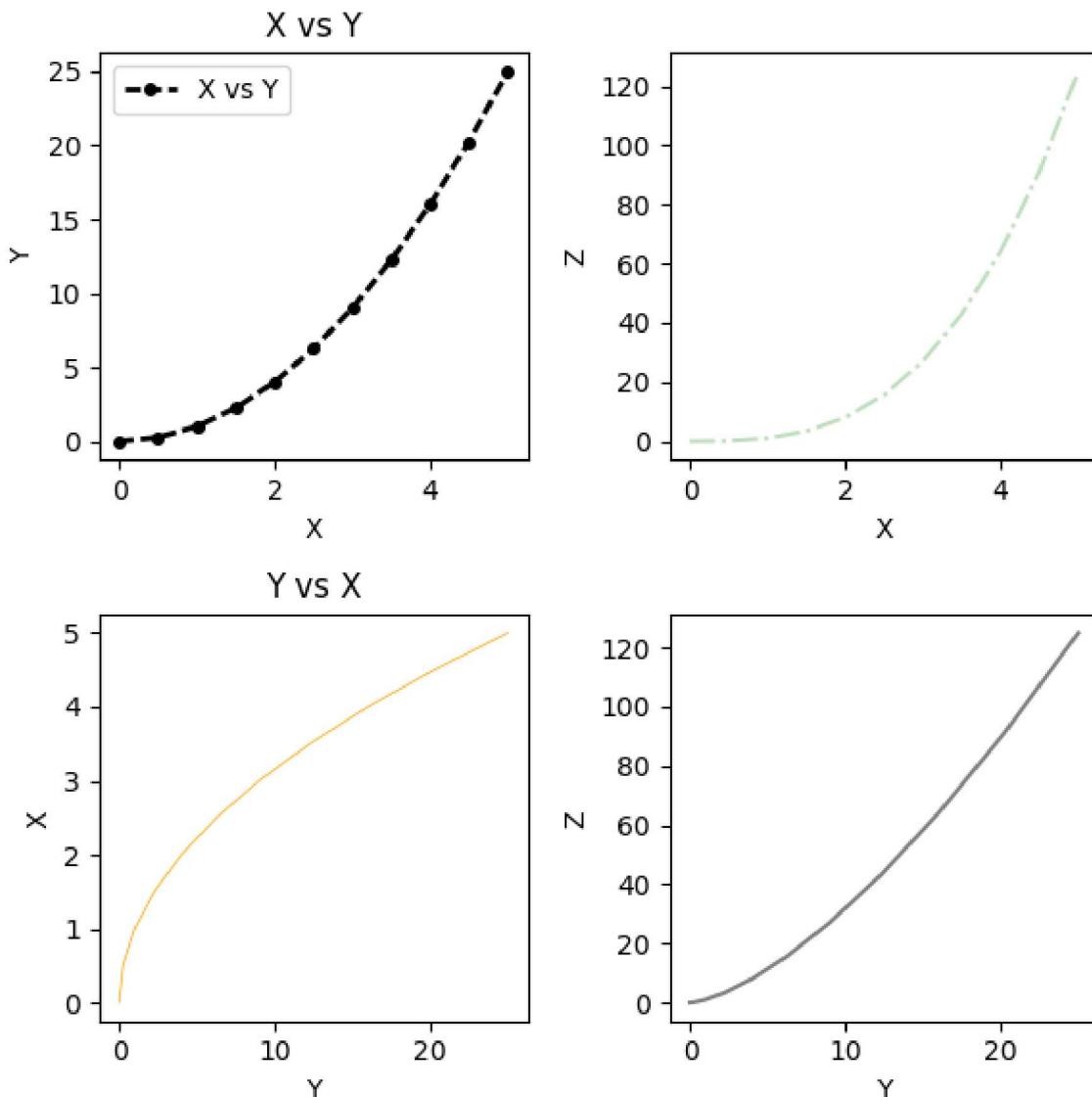
```
# figsize is very important to have good visualization
canvas, axes = plt.subplots(nrows = 2, ncols = 2, figsize=(6,6), dpi = 100)

axes[0,0].plot(x,y, label= 'X vs Y', color = 'black', linewidth=2, linestyle='--', marker='o', markersize=4)
axes[0,0].set_xlabel('X')
axes[0,0].set_ylabel('Y')
axes[0,0].set_title('X vs Y')
axes[0,0].legend()

axes[1,0].plot(y,x, color='orange', lw=0.5)
axes[1,0].set_xlabel('Y')
axes[1,0].set_ylabel('X')
axes[1,0].set_title('Y vs X')

axes[0,1].plot(x,z, color='green',alpha=0.25, linestyle='-.') #alpha is for transparency
axes[0,1].set_xlabel('X')
axes[0,1].set_ylabel('Z')

axes[1,1].plot(y,z, color='grey')
axes[1,1].set_xlabel('Y')
axes[1,1].set_ylabel('Z')
plt.tight_layout()
```



In [154]:

```
canvas.savefig('FirstPlotSave3.png',dpi=350)
```

Legend

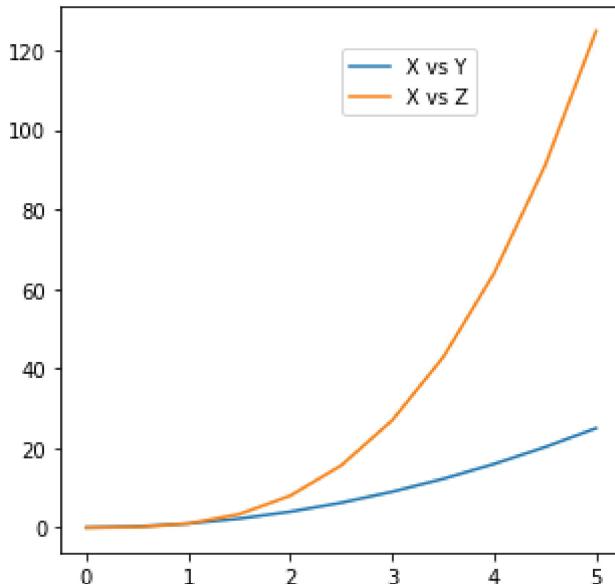
In [167]:

```
canvas, axes = plt.subplots(1,1, figsize=(5,5))
axes.plot(x,y, label = 'X vs Y')
axes.plot(x,z, label = 'X vs Z')

# For the Location of Legend, use a tuple or you can also use the default preset locations given in the doc page.
plt.legend(loc = (0.5,0.8))
# plt.legend(loc = 0) ; 0 is for the best position
```

Out[167]:

```
<matplotlib.legend.Legend at 0x28d91c50c88>
```

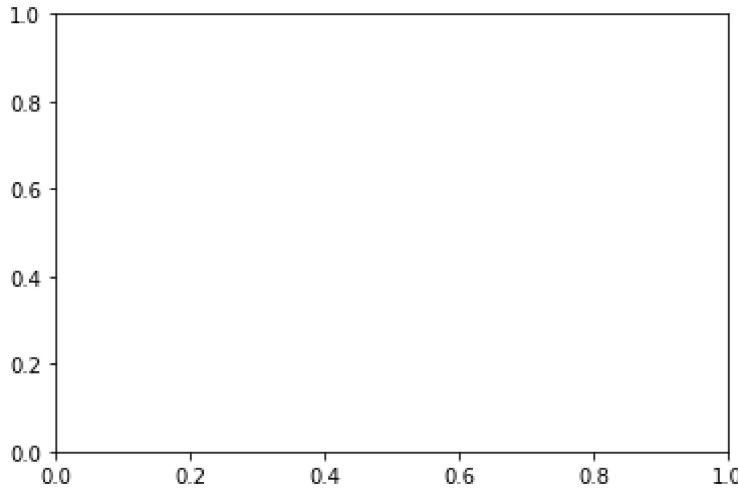


In [1]:

```
#This is the first code that I am writing using my new Surface Notebook
```

In [9]:

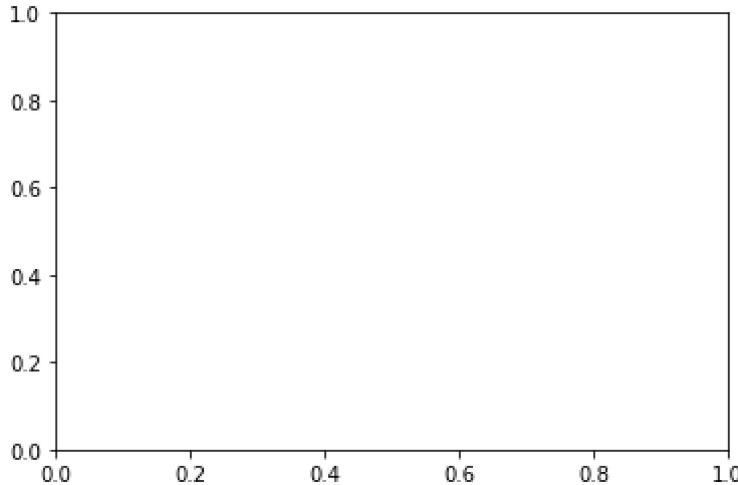
```
canvas,axes = plt.subplots(nrows=1,ncols=1)
```



In [26]:

```
axes.plot(x,y, label="X vs Y")
plt.legend()
```

```
C:\Users\Rutanshu Desai\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:545: UserWarning: No labelled objects found. Use label='...' kwarg on individual plots.
  warnings.warn("No labelled objects found. "
```



In [17]:

```
axes.set_title('X vs Y')
axes.set_xlabel('X axis')
axes.set_ylabel('Y axis')
```

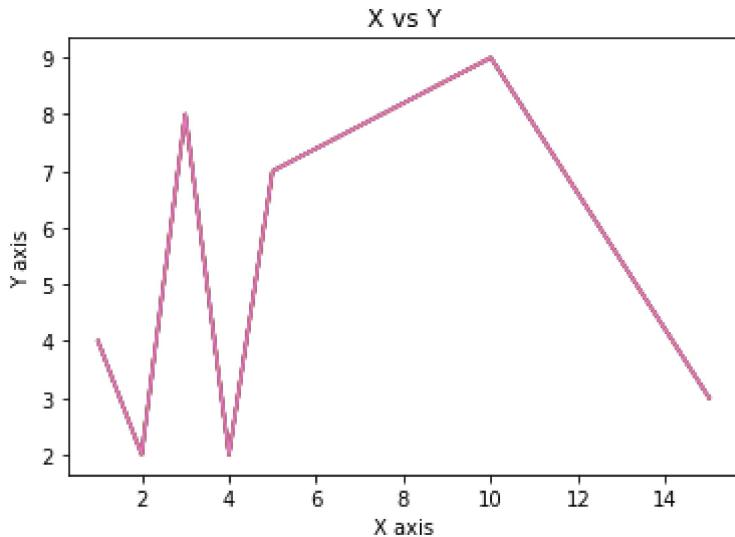
Out[17]:

```
<matplotlib.text.Text at 0x13feb2ab128>
```

In [25]:

```
canvas
```

Out[25]:



In [39]:

```
axes.plot(x,y, label="X vs Y")
axes.plot(x,x**2, label="X vs X2")
canvas.legend()
```

```
-----
-
TypeError                                Traceback (most recent call last)
t)
<ipython-input-39-e1d4a7e0c05e> in <module>()
      1 axes.plot(x,y, label="X vs Y")
----> 2 axes.plot(x,x**2, label="X vs X2")
      3 canvas.legend()
```

TypeError: unsupported operand type(s) for ** or pow(): 'list' and 'int'

In [29]:

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

```
<matplotlib.figure.Figure at 0x13fec5fa630>
```

In [33]:

```
pl=fig.add_axes([0.1,0.1,0.5,0.5])
```

In [46]:

```
pl.plot(x,y,label="x vs y")
pl.plot(x,x2,label="x vs x2")
pl.legend()
```

NameError

Traceback (most recent call last)

t)

```
<ipython-input-46-89c28af9f8e4> in <module>()
    1 pl.plot(x,y,label="x vs y")
----> 2 pl.plot(x,x2,label="x vs x2")
```

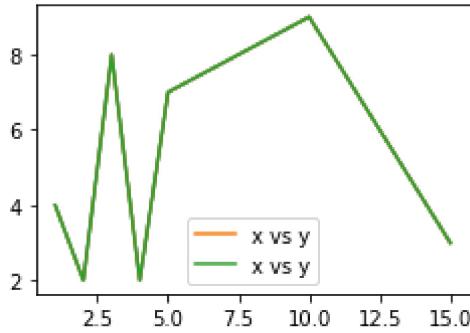
```
  3 pl.legend()
```

NameError: name 'x2' is not defined

In [38]:

```
fig
```

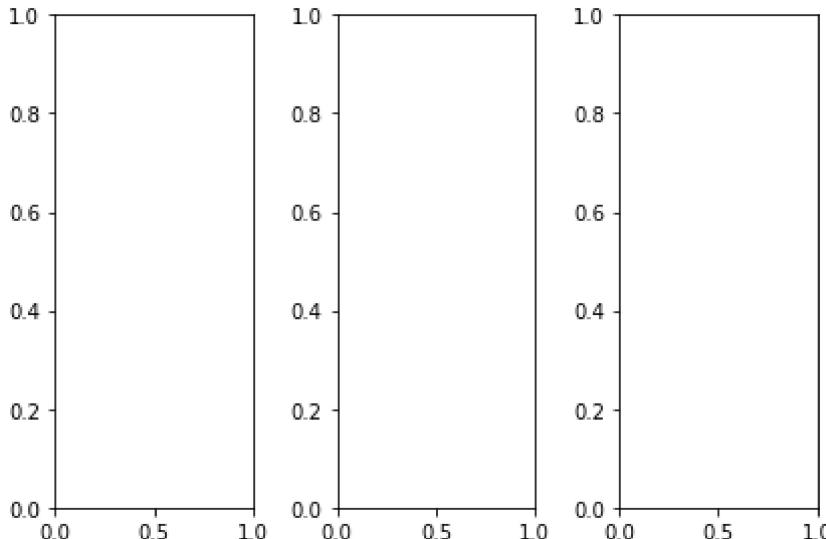
Out[38]:



In [73]:

```
canvas.tight_layout()
canvas
```

Out[73]:

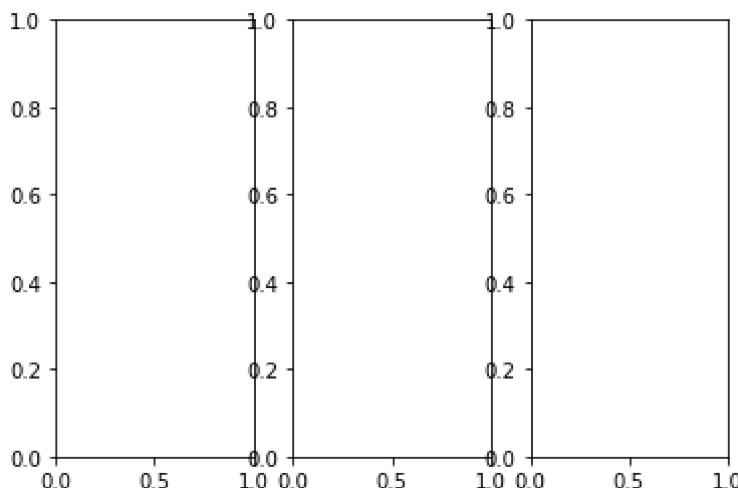


In [80]:

```
canvas, axes = plt.subplots(1,3)
i=np.array([1,2,3])
for pl in i:
    if pl==1:
        axes=axes.plot(x,y)
    if pl==2:
        axes=axes.plot(y,x)
    if pl==3:
        axes=axes.plot(x,x**2)
canvas.tight_layout()
canvas
```

```
-----
-
AttributeError                               Traceback (most recent call last)
t)
<ipython-input-80-46879856cbdd> in <module>()
      3 for pl in i:
      4     if pl==1:
----> 5         axes=axes.plot(x,y)
      6     if pl==2:
      7         axes=axes.plot(y,x)

AttributeError: 'numpy.ndarray' object has no attribute 'plot'
```



Seaborn

In [1]:

```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [7]:

```
tips.head()
```

Out[7]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [3]:

```
tips = sns.load_dataset('tips')
tips.count()
```

Out[3]:

```
total_bill    244
tip          244
sex          244
smoker       244
day          244
time         244
size          244
dtype: int64
```

In [25]:

```
tips[(tips['total_bill']>15) & (tips['total_bill']<25) ].count()
```

Out[25]:

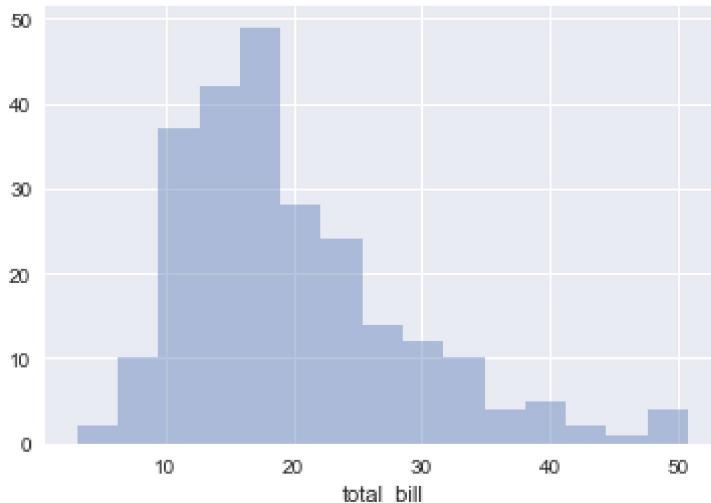
```
total_bill    108
tip          108
sex          108
smoker       108
day          108
time         108
size          108
dtype: int64
```

In [5]:

```
# A distplot is basocially a histogram  
# This is a basic univariate data  
sns.distplot(tips['total_bill'], kde=False,bins=15)
```

Out[5]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2cc50f9c860>
```

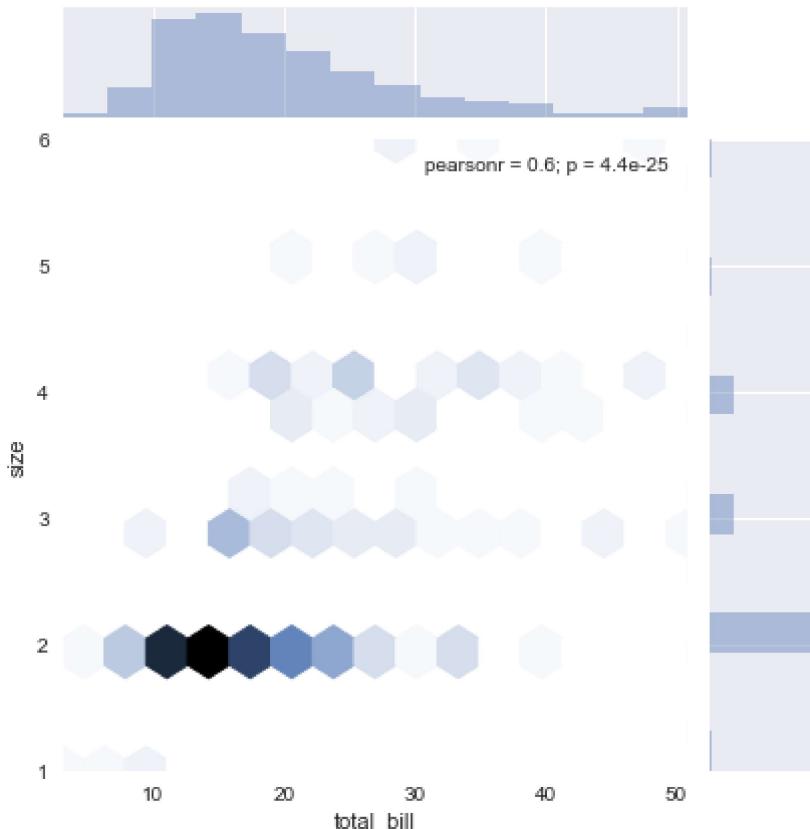


In [13]:

```
# jointplot is a bivariate data bar charts where it takes min 3 parameters, x=one var,
y=second var, data=dataset
# it has a parameter called kind, which allows to plot many kinds of bivariate plots like Scatter, hex and regression.
sns.jointplot(x='total_bill',y='size',data=tips, kind='hex')
```

Out[13]:

```
<seaborn.axisgrid.JointGrid at 0x2cc51ca4240>
```



Categorical Plots

In [18]:

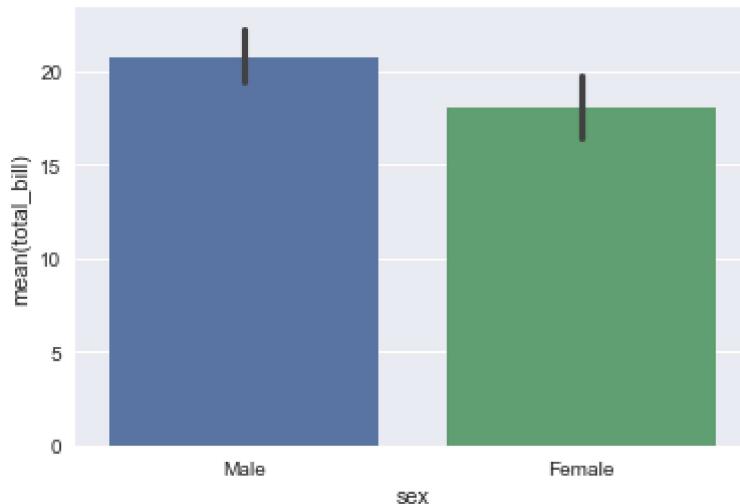
```
import numpy as np
```

In [27]:

```
sns.barplot(x='sex',y='total_bill',data=tips) #by default the value is for the mean
```

Out[27]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2cc51c651d0>
```

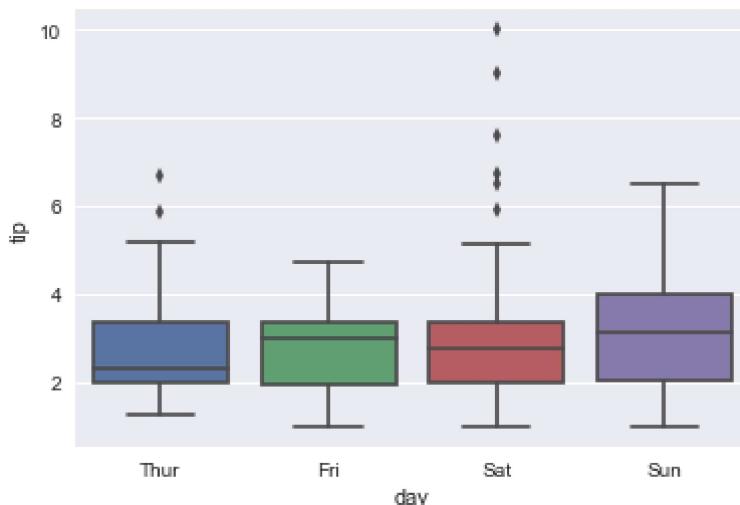


In [36]:

```
sns.boxplot(x='day',y='tip',data=tips)
```

Out[36]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2cc53398b00>
```



In [3]:

```
import seaborn as sns
tips = sns.load_dataset('tips')
```

In [4]:

```
flights = sns.load_dataset('flights')
```

In [5]:

```
flights.head()
```

Out[5]:

	year	month	passengers
0	1949	January	112
1	1949	February	118
2	1949	March	132
3	1949	April	129
4	1949	May	121

In [6]:

```
import pandas as pd
```

In [8]:

```
pivot=pd.pivot_table(data=flights,values='passengers',index='month',columns='year')
```

In [9]:

```
pivot
```

Out[9]:

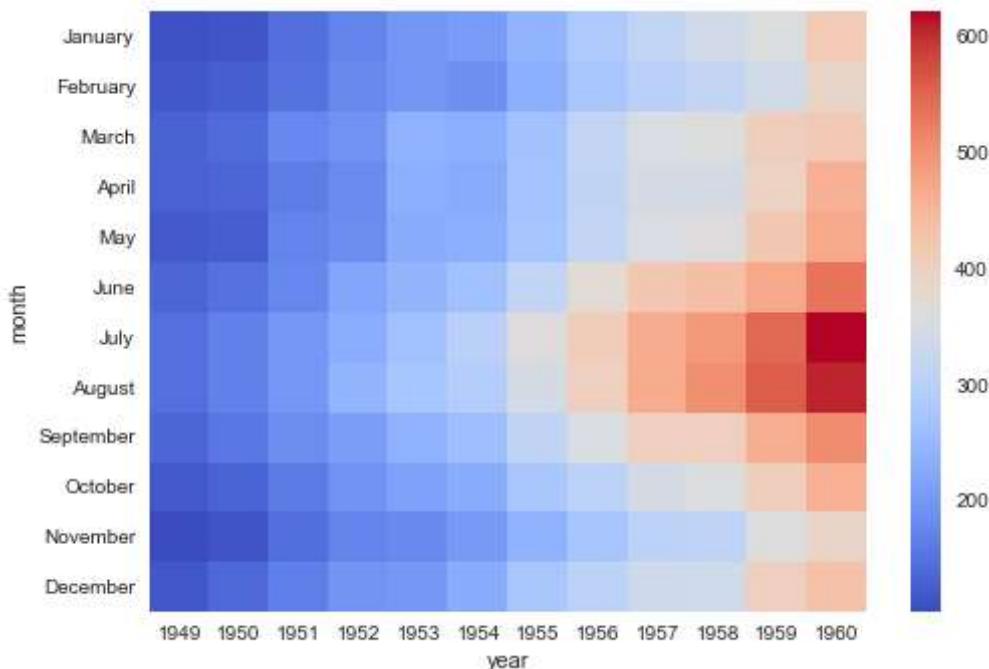
year	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960
month												
January	112	115	145	171	196	204	242	284	315	340	360	417
February	118	126	150	180	196	188	233	277	301	318	342	391
March	132	141	178	193	236	235	267	317	356	362	406	419
April	129	135	163	181	235	227	269	313	348	348	396	461
May	121	125	172	183	229	234	270	318	355	363	420	472
June	135	149	178	218	243	264	315	374	422	435	472	535
July	148	170	199	230	264	302	364	413	465	491	548	622
August	148	170	199	242	272	293	347	405	467	505	559	606
September	136	158	184	209	237	259	312	355	404	404	463	508
October	119	133	162	191	211	229	274	306	347	359	407	461
November	104	114	146	172	180	203	237	271	305	310	362	390
December	118	140	166	194	201	229	278	306	336	337	405	432

In [11]:

```
sns.heatmap(data=pivot, cmap='coolwarm')
```

Out[11]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1875fc44da0>
```



In [13]:

```
tips.head()
```

Out[13]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [25]:

```
pv=pd.pivot_table(data=tips,values='total_bill',index='day',columns='time')
```

In [17]:

```
import numpy as np
```

In [38]:

```
a=pd.Series(tips['time'])
```

In [40]:

```
a=tips['time']
tips[tips[a.lower()]=='lunch']
```

```
-----
-
AttributeError                                Traceback (most recent call last)
t)
<ipython-input-40-e2e0169ef1dc> in <module>()
      1 a=tips['time']
----> 2 tips[tips[a.lower()]=='lunch']
```

```
C:\Users\Rutanshu Desai\Anaconda3\lib\site-packages\pandas\core\generic.py
in __getattr__(self, name)
  2742         if name in self._info_axis:
  2743             return self[name]
-> 2744         return object.__getattribute__(self, name)
  2745
  2746     def __setattr__(self, name, value):
```

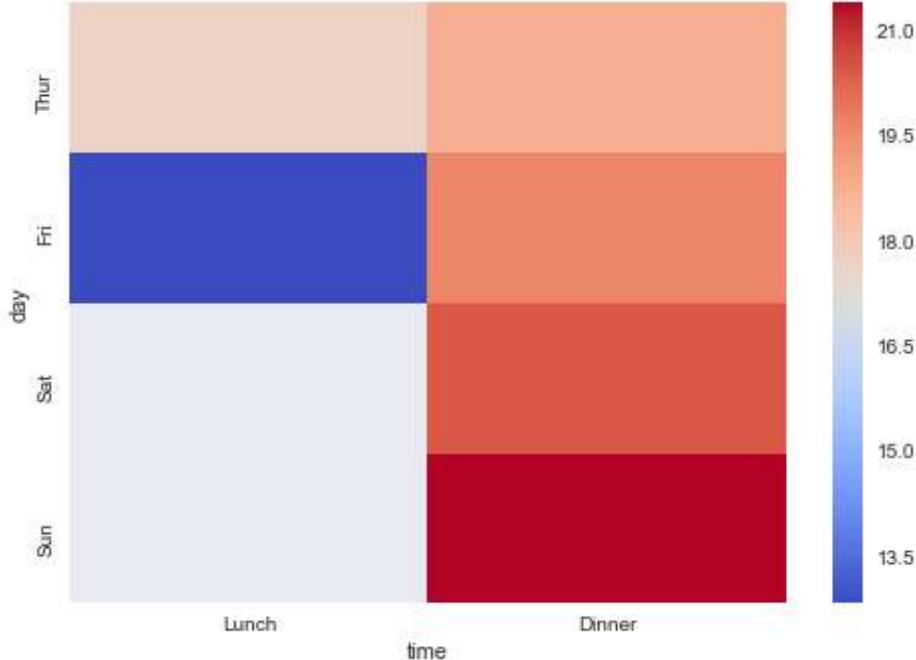
AttributeError: 'Series' object has no attribute 'lower'

In [27]:

```
sns.heatmap(data=pv,cmap='coolwarm')
```

Out[27]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1876033cf60>
```



creating regression plot using Implot of seaborn

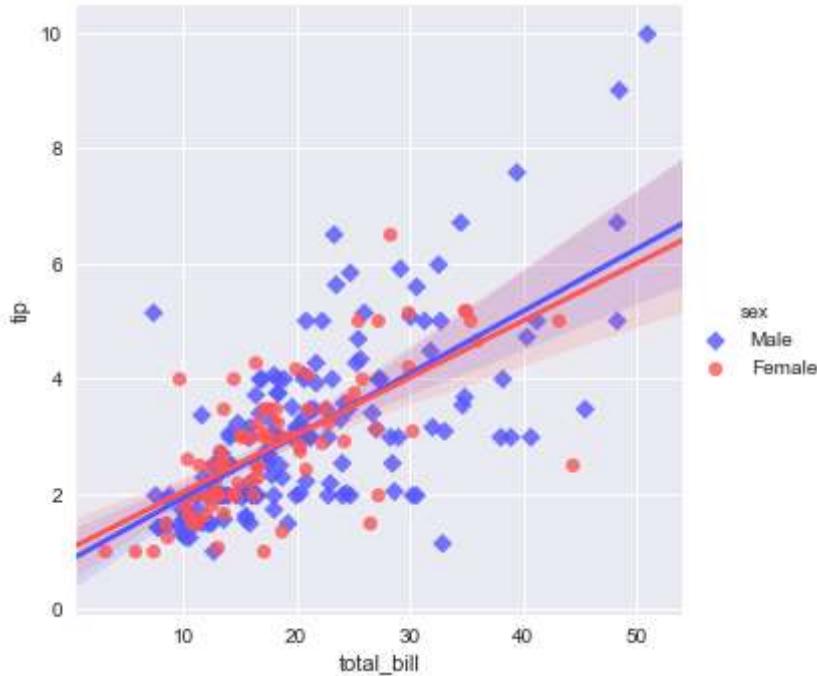
In [72]:

```
plt.figure(figsize=(15,5))
sns.lmplot(x='total_bill',y='tip',data=tips,markers=[ 'D', 'o' ],hue='sex',palette='seismic')
```

Out[72]:

```
<seaborn.axisgrid.FacetGrid at 0x2096ad61978>
```

```
<matplotlib.figure.Figure at 0x2096ad98e48>
```



In []: