Ch02_The importance of data visualization in business

September 5, 2018

```
In [ ]: import matplotlib.pyplot as plt
        import numpy as np
        import pandas as pd
        import seaborn as sns
        import pygal
        from mayavi import mlab

In [5]: try:
            import matplotlib
        except:
            import pip
            pip.main(['install', 'matplotlib'])
            import matplotlib
```

1 Matplotlib

```
In [23]: import numpy as np
         import matplotlib.pyplot as plt
         %matplotlib inline
         plt.style.use('seaborn-whitegrid')

         X = [590,540,740,130,810,300,320,230,470,620,770,250]
         Y = [32,36,39,52,61,72,77,75,68,57,48,48]

         plt.scatter(X,Y)
         plt.xlim(0,1000)
         plt.ylim(0,100)

         #scatter plot color
         plt.scatter(X, Y, s=800, c='red', marker='+')

         #change axes ranges
         plt.xlim(0,1000)
         plt.ylim(0,100)

         #add title
```
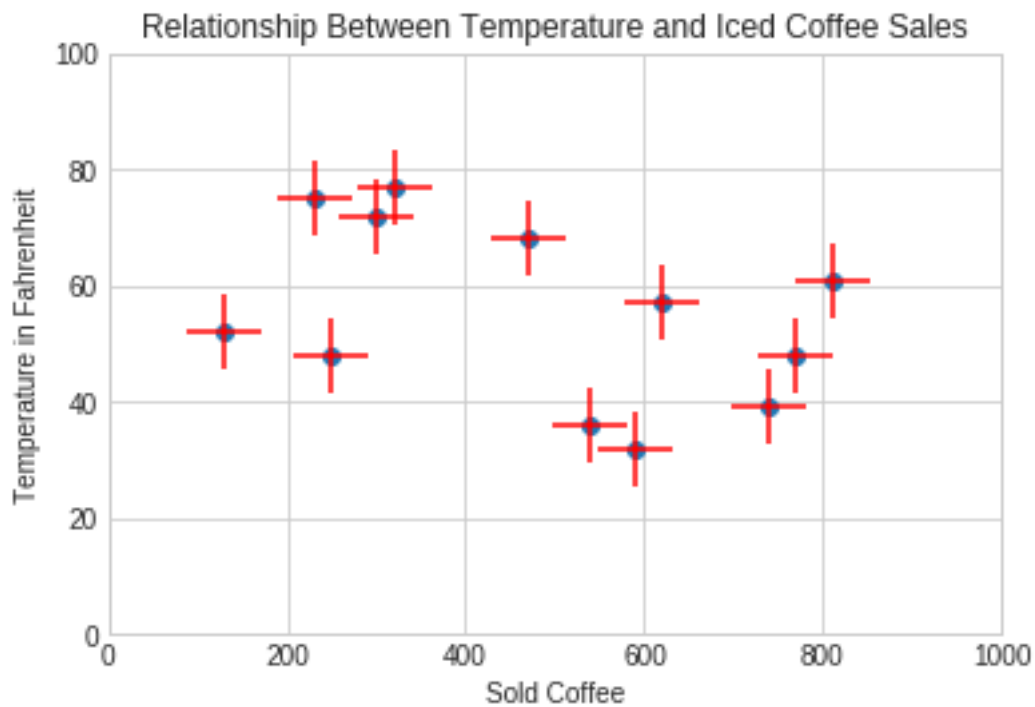
```python
plt.title('Relationship Between Temperature and Iced Coffee Sales')

#add x and y labels
plt.xlabel('Sold Coffee')
plt.ylabel('Temperature in Fahrenheit')

#show plot
plt.show()
```


Relationship Between Temperature and Iced Coffee Sales

```python
In [20]: %matplotlib inline
         import matplotlib.pyplot as plt
         plt.style.use('seaborn-whitegrid')
         import numpy as np

         # Create empty figure
         fig = plt.figure()
         ax = plt.axes()

         x = np.linspace(0, 10, 1000)
         ax.plot(x, np.sin(x));

         plt.plot(x, np.sin(x))
         plt.plot(x, np.cos(x))
         plt.xlim(0, 11)
```
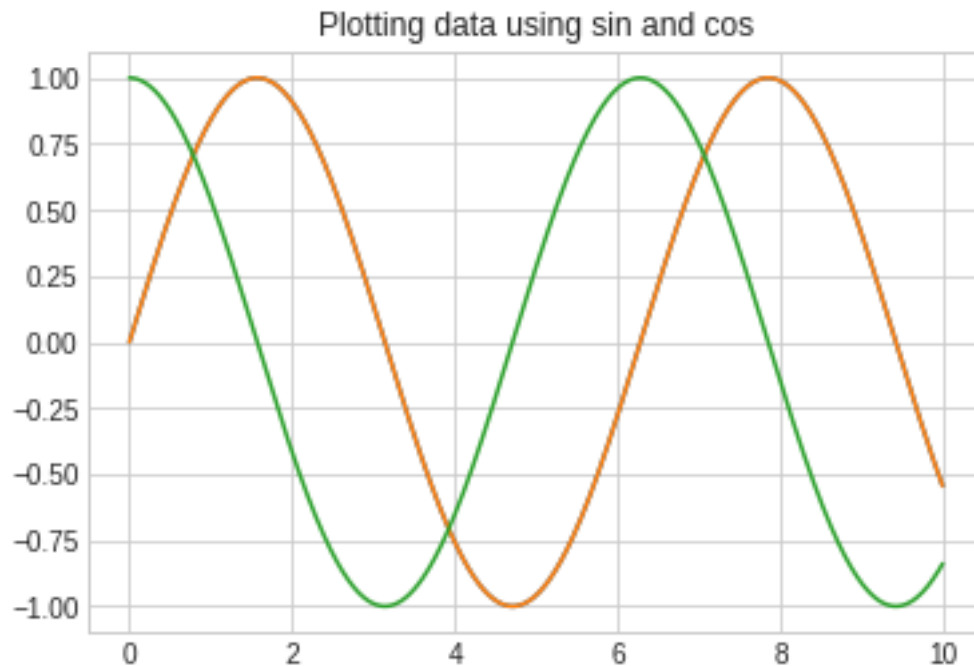
```
        plt.ylim(-2, 2)
        plt.axis('tight')
        #add title
        plt.title('Plotting data using sin and cos')
```

Out[20]: Text(0.5,1,'Plotting data using sin and cos')

Plotting data using sin and cos


```
In [18]: plt.plot(x, np.sin(x - 0), color='blue')        # specify color by name
         plt.plot(x, np.sin(x - 1), color='g')           # short color code (rgbcmyk)
         plt.plot(x, np.sin(x - 2), color='0.75')        # Grayscale between 0 and 1
         plt.plot(x, np.sin(x - 3), color='#FFDD44')     # Hex code (RRGGBB from 00 to FF)
         plt.plot(x, np.sin(x - 4), color=(1.0,0.2,0.3)) # RGB tuple, values 0 to 1
         plt.plot(x, np.sin(x - 5), color='chartreuse'); # all HTML color names supported
```
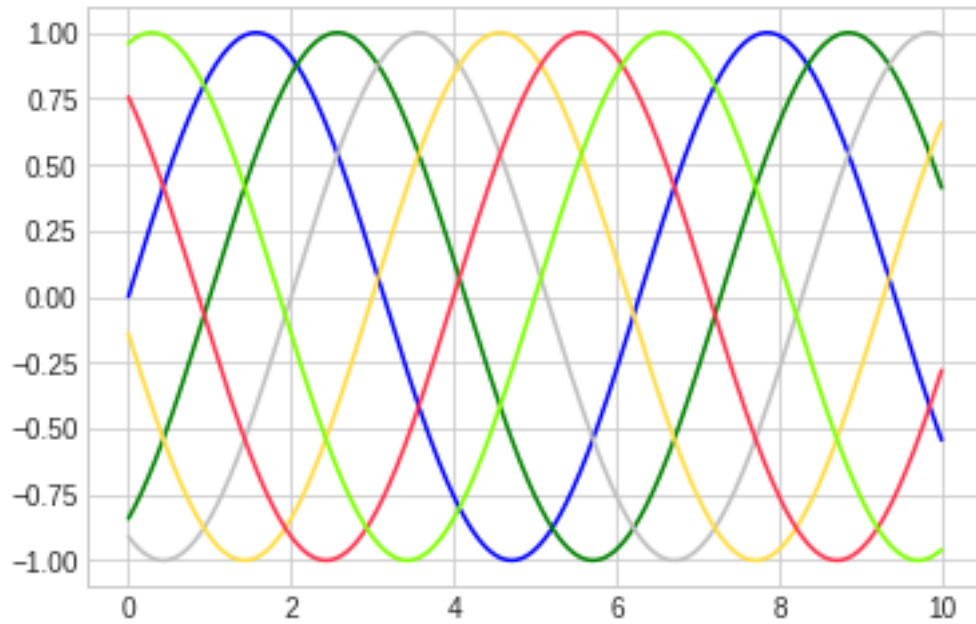
2 Seaborn

```
In [34]: import matplotlib.pyplot as plt
         %matplotlib inline
         import numpy as np
         import pandas as pd
         import seaborn as sns
         plt.style.use('classic')
         plt.style.use('seaborn-whitegrid')

         # Create some data
         data = np.random.multivariate_normal([0, 0], [[5, 2], [2, 2]], size=2000)
         data = pd.DataFrame(data, columns=['x', 'y'])

         # Plot the data with seaborn
         sns.distplot(data['x'])
         sns.distplot(data['y']);
```
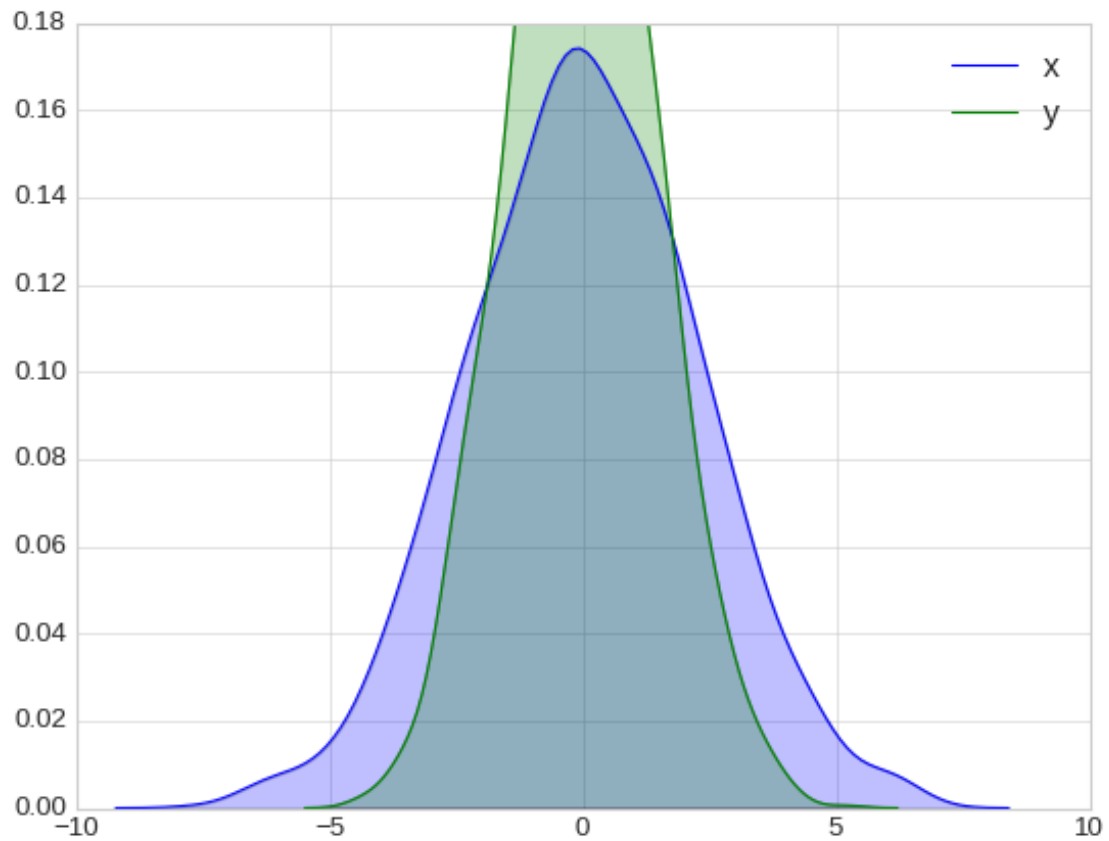
```
/home/nbuser/anaconda3_501/lib/python3.6/site-packages/scipy/stats/stats.py:1633: FutureWarning:
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```
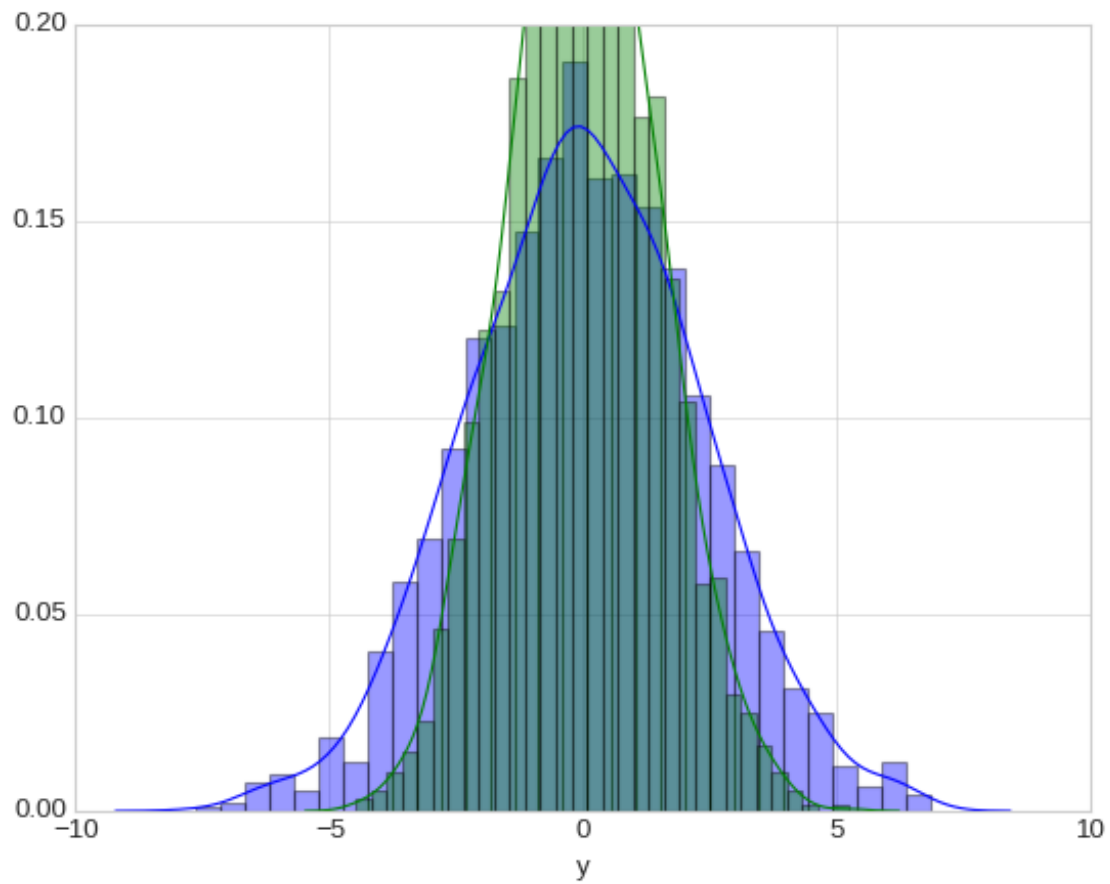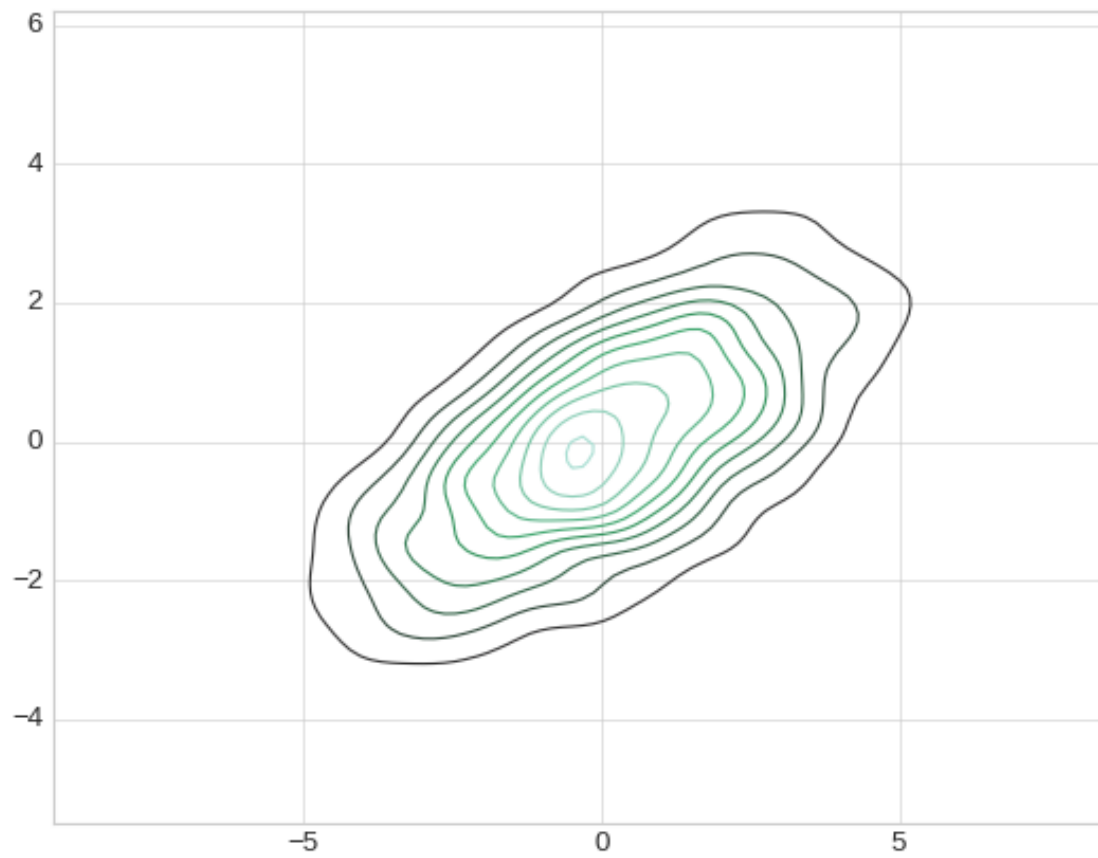
```
In [35]: for col in 'xy':
            sns.kdeplot(data[col], shade=True)
```

/home/nbuser/anaconda3_501/lib/python3.6/site-packages/scipy/stats/stats.py:1633: FutureWarning:
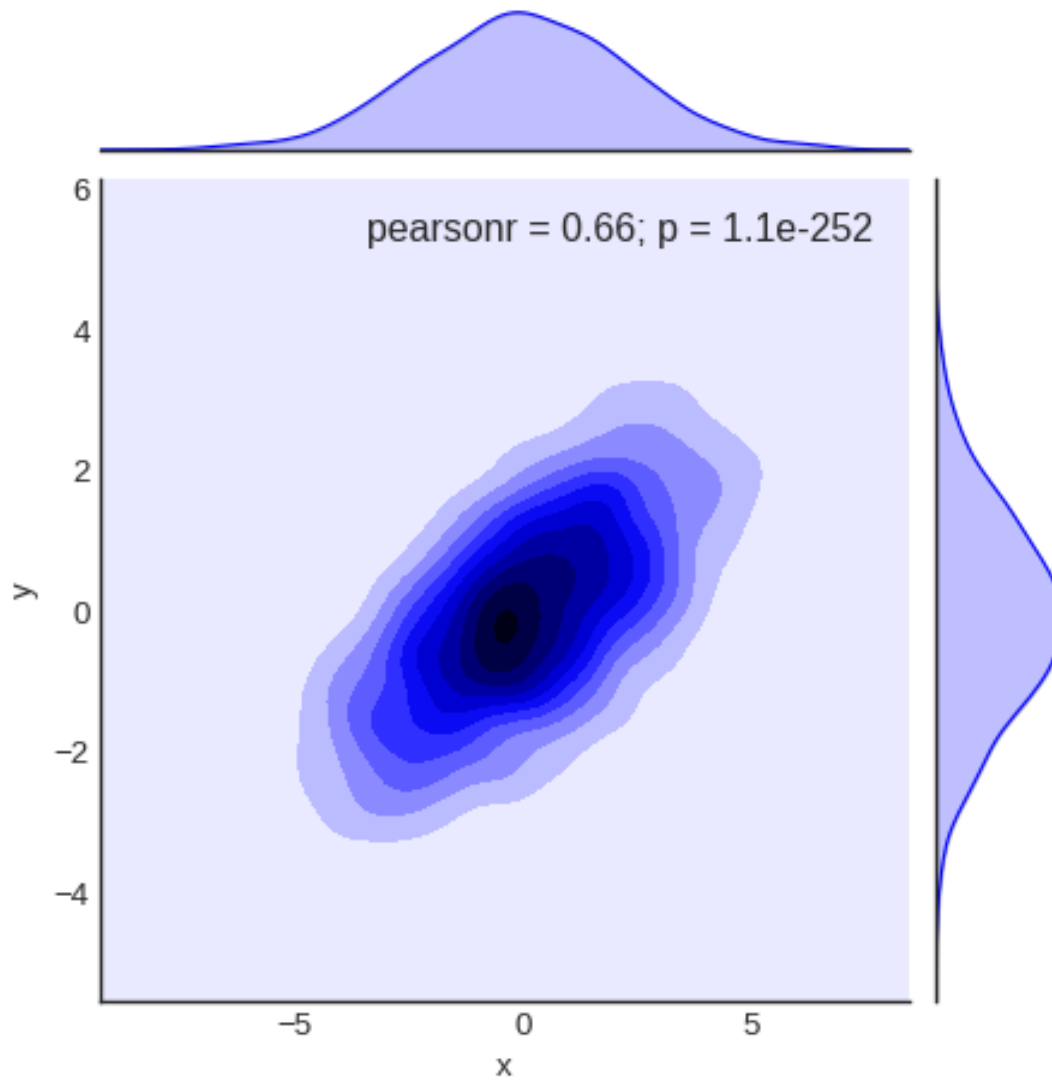 return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

In [36]: sns.kdeplot(data);

/home/nbuser/anaconda3_501/lib/python3.6/site-packages/seaborn/distributions.py:630: UserWarning
  warnings.warn(warn_msg, UserWarning)
/home/nbuser/anaconda3_501/lib/python3.6/site-packages/scipy/stats/stats.py:1633: FutureWarning:
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

In [37]: with sns.axes_style('white'):
             sns.jointplot("x", "y", data, kind='kde');

/home/nbuser/anaconda3_501/lib/python3.6/site-packages/scipy/stats/stats.py:1633: FutureWarning:
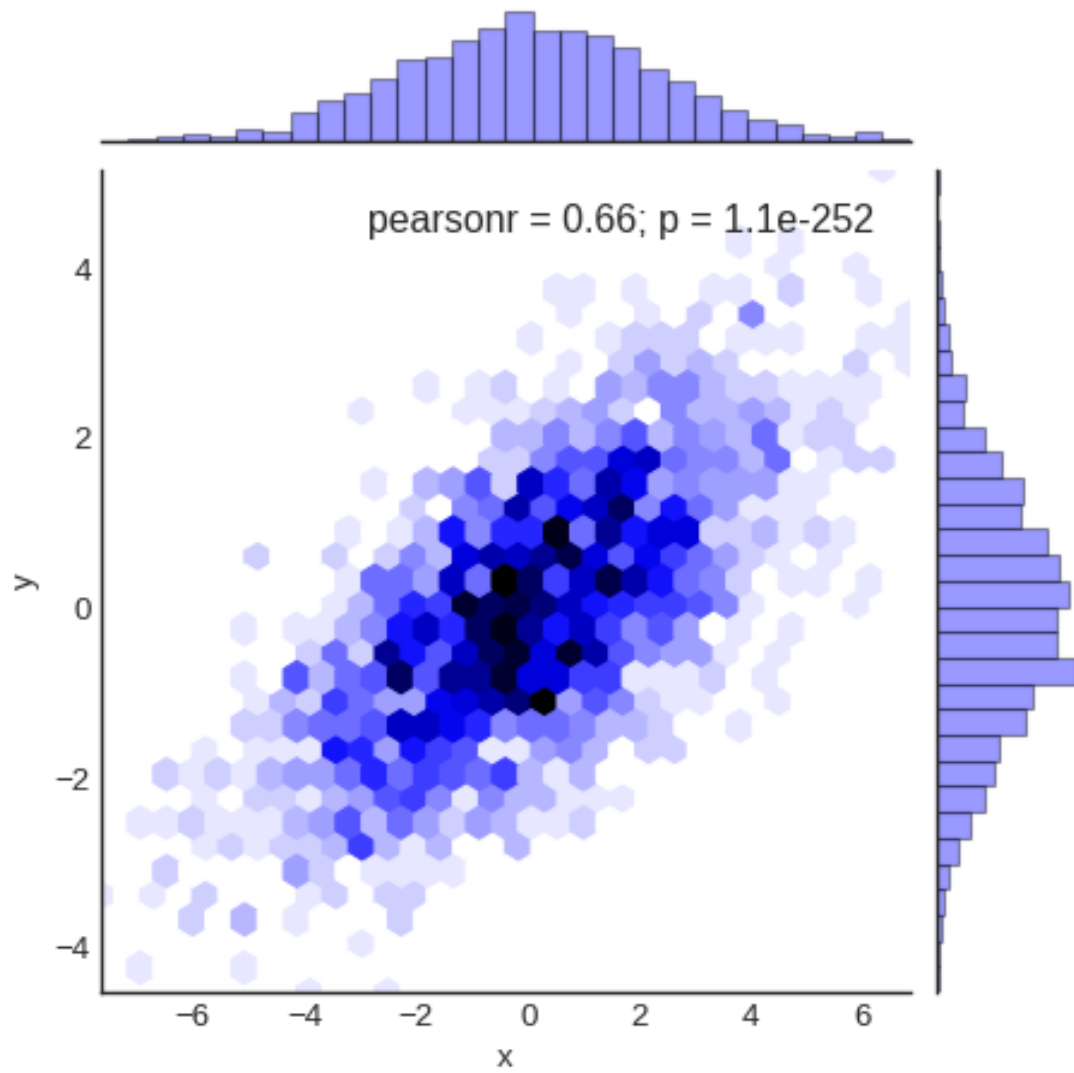 return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

pearsonr = 0.66; p = 1.1e-252

```
In [38]: with sns.axes_style('white'):
             sns.jointplot("x", "y", data, kind='hex')

/home/nbuser/anaconda3_501/lib/python3.6/site-packages/scipy/stats/stats.py:1633: FutureWarning:
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```
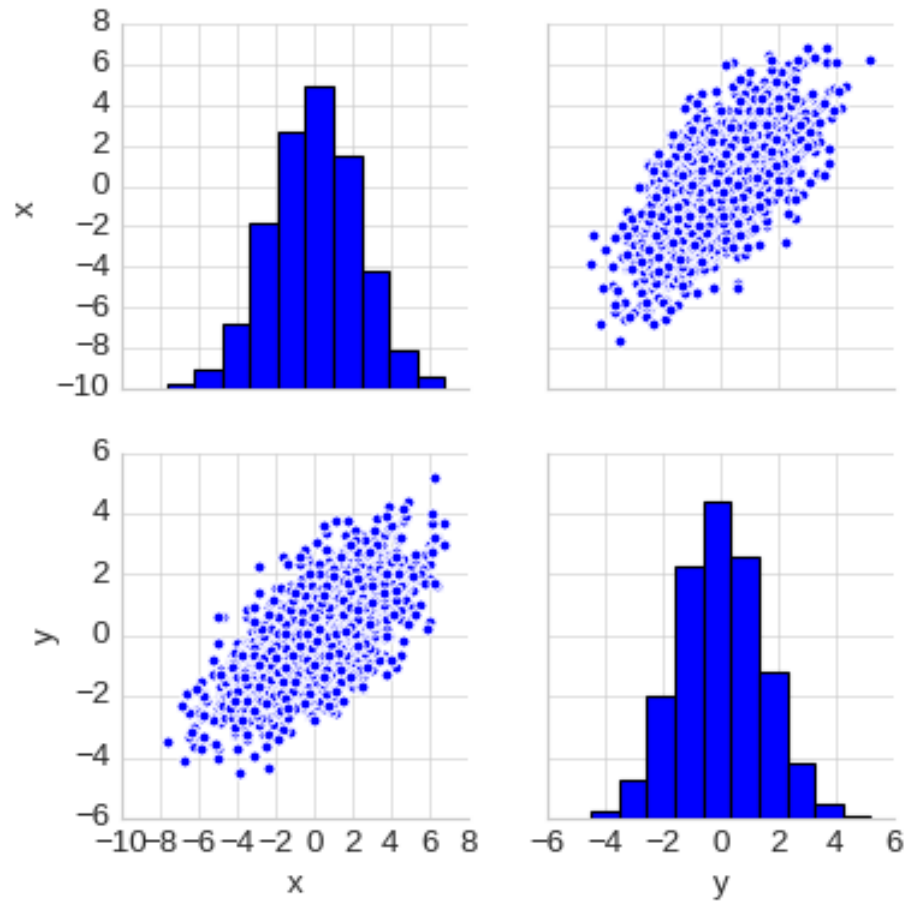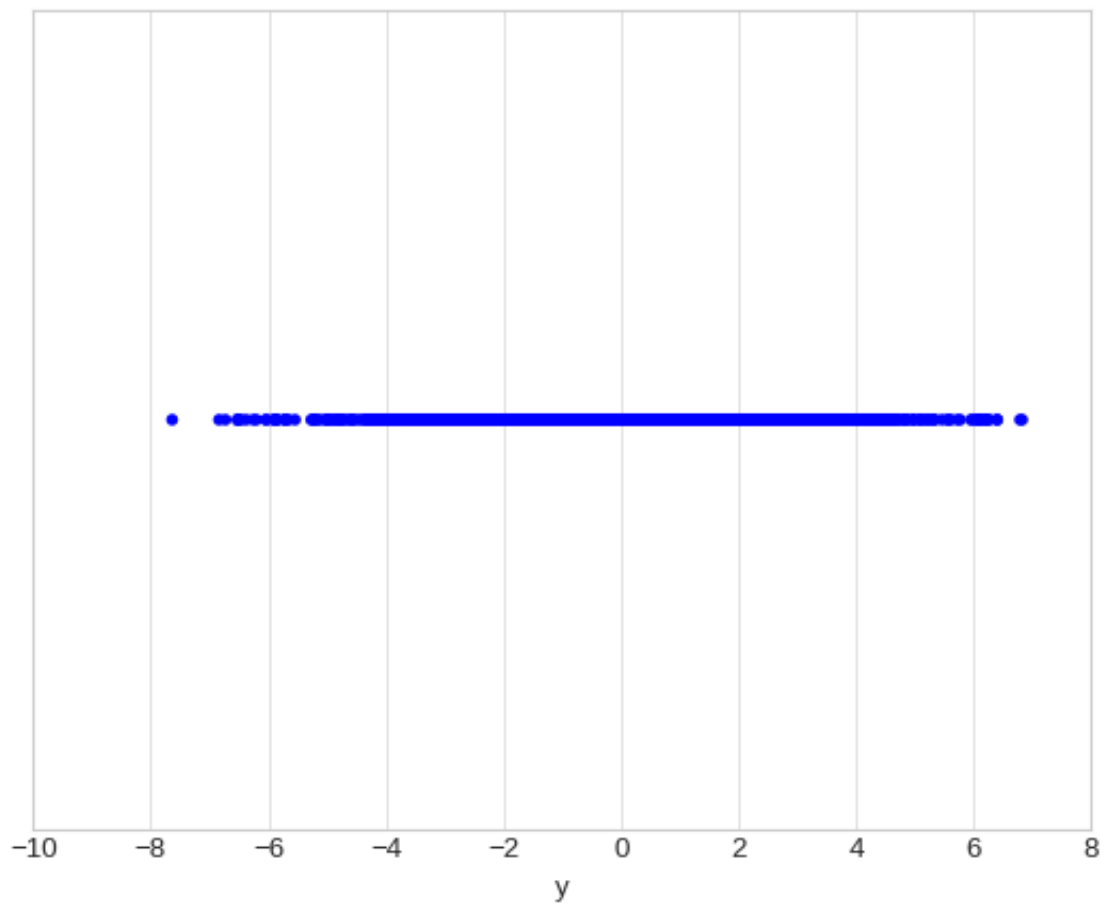
pearsonr = 0.66; p = 1.1e-252

In [41]: sns.pairplot(data);

```
In [45]: sns.stripplot( x = data['x'])
         sns.stripplot( x = data['y'])

Out[45]: <matplotlib.axes._subplots.AxesSubplot at 0x7f309cbe4ac8>
```
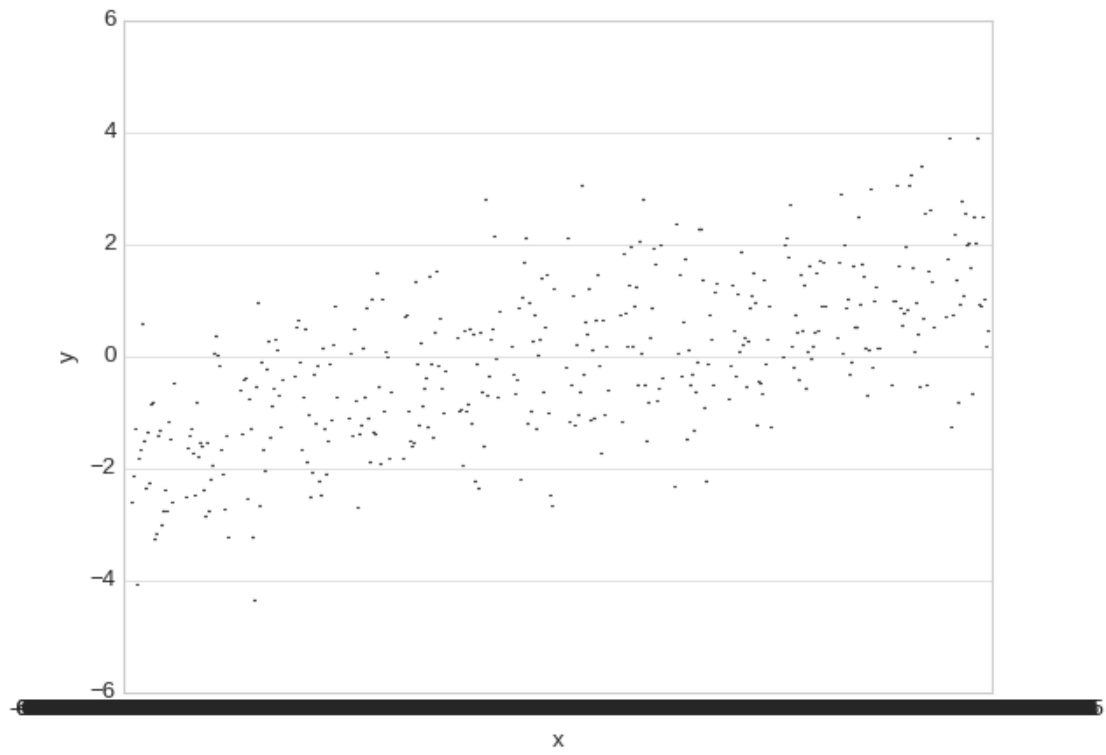
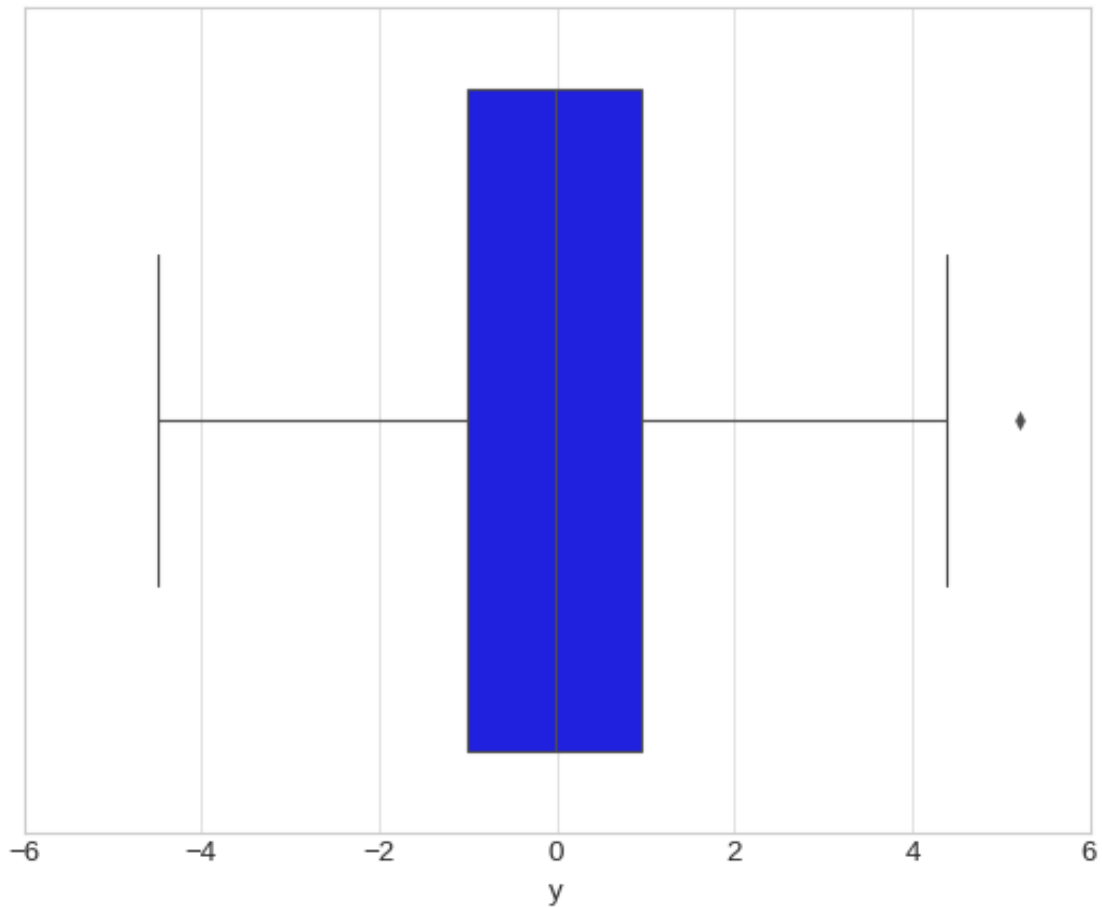In [47]: # box plot per rank
         sns.boxplot(x = 'x', y = 'y', data=data)

Out[47]: <matplotlib.axes._subplots.AxesSubplot at 0x7f309fd5eba8>

```
In [50]:  # box plot salaries
          sns.boxplot(x = data['y'], whis=2)

Out[50]:  <matplotlib.axes._subplots.AxesSubplot at 0x7f3083f15898>
```

3 Plotly

```
In [64]: from plotly import __version__
         from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
         init_notebook_mode(connected=True)
         print (__version__)
```

3.1.0

```
In [91]: import plotly.graph_objs as go

         plot([go.Scatter(x=[95, 77, 84], y=[75, 67, 56])])
```

Out[91]: 'file:///home/nbuser/library/temp-plot.html'

```
In [67]: import plotly.graph_objs as go
         import numpy as np
```

```
        x = np.random.randn(2000)
        y = np.random.randn(2000)
        iplot([go.Histogram2dContour(x=x, y=y, contours=dict(coloring='heatmap')),
               go.Scatter(x=x, y=y, mode='markers', marker=dict(color='white', size=3, opacity=
```

In [90]:
```
        import plotly.offline as offline
        import plotly.graph_objs as go

        offline.plot({'data': [{'y': [14, 22, 30, 44]}],
                      'layout': {'title': 'Offline Plotly', 'font': dict(size=16)}}, image='pn
```

Out[90]: 'file:///home/nbuser/library/temp-plot.html'

In [88]:
```
        import plotly.plotly as py
        import plotly.graph_objs as go
        import plotly
        import plotly.offline as offline


        df = pd.read_csv("https://raw.githubusercontent.com/plotly/datasets/master/school_earni

        schools = df.School

        data = [go.Bar(x=df.School,y=df.Gap)]

        py.iplot(data, filename='jupyter-basic_bar')
```
Aw, snap! We didn't get a username with your request.

Don't have an account? https://plot.ly/api_signup

Questions? accounts@plot.ly

        PlotlyError   Traceback (most recent call last)

        <ipython-input-88-64385519e16b> in <module>()
         13 data = [go.Bar(x=df.School,y=df.Gap)]
         14
    ---> 15 py.iplot(data, filename='jupyter-basic_bar')

        ~/anaconda3_501/lib/python3.6/site-packages/plotly/plotly/plotly.py in iplot(figure_or_d

                                            14

```
      171              embed_options['height'] = str(embed_options['height']) + 'px'
      172
--> 173        return tools.embed(url, **embed_options)
      174
      175


      ~/anaconda3_501/lib/python3.6/site-packages/plotly/tools.py in embed(file_owner_or_url,
      393            else:
      394                url = file_owner_or_url
--> 395            return PlotlyDisplay(url, width, height)
      396        else:
      397            if (get_config_defaults()['plotly_domain']


      ~/anaconda3_501/lib/python3.6/site-packages/plotly/tools.py in __init__(self, url, width
     1440        def __init__(self, url, width, height):
     1441            self.resource = url
->  1442            self.embed_code = get_embed(url, width=width, height=height)
     1443            super(PlotlyDisplay, self).__init__(data=self.embed_code)
     1444


      ~/anaconda3_501/lib/python3.6/site-packages/plotly/tools.py in get_embed(file_owner_or_u
      298                "'{1}'."
      299                "\nRun help on this function for more information."
--> 300                "".format(url, plotly_rest_url))
      301            urlsplit = six.moves.urllib.parse.urlparse(url)
      302            file_owner = urlsplit.path.split('/')[1].split('~')[1]


      PlotlyError: Because you didn't supply a 'file_id' in the call, we're assuming you're tr
     Run help on this function for more information.
```

4 geoplotlib

```
In [ ]: import geoplotlib
        from geoplotlib.utils import read_csv

        data = read_csv('bus.csv')
        geoplotlib.dot(data)
        geoplotlib.show()
```
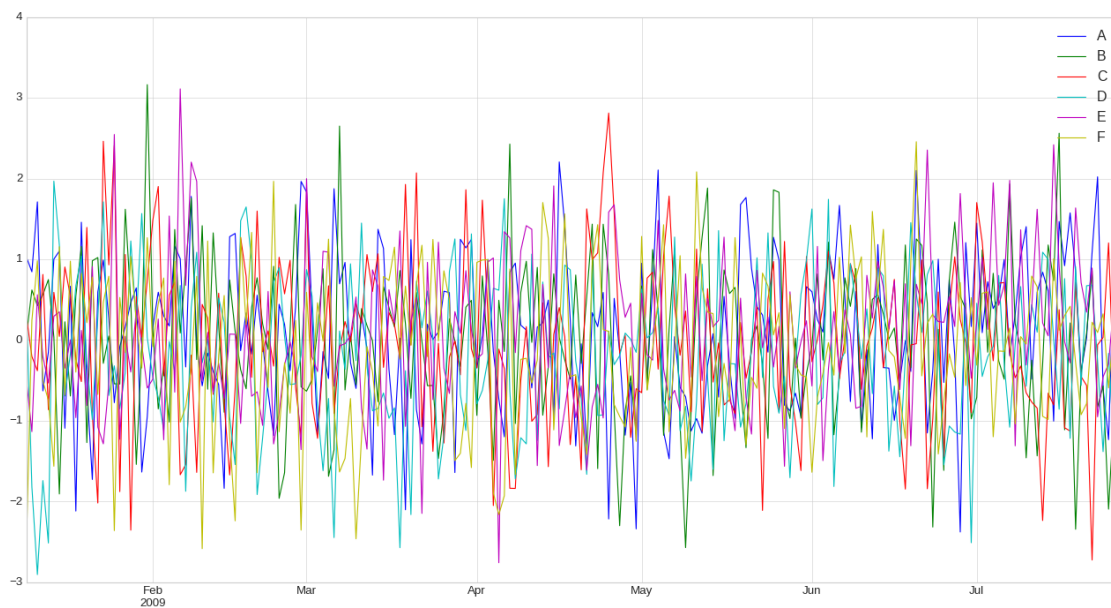
5 Direct plotting

```
In [116]: import pandas as pd
          import numpy as np

          df = pd.DataFrame(np.random.randn(200,6),index=pd.date_range('1/9/2009',
             periods=200), columns=list('ABCDEF'))


          df.plot(figsize=(20, 10)).legend(bbox_to_anchor=(1, 1))
          #Shape of passed values is (10, 200), indices imply (4, 10)
```
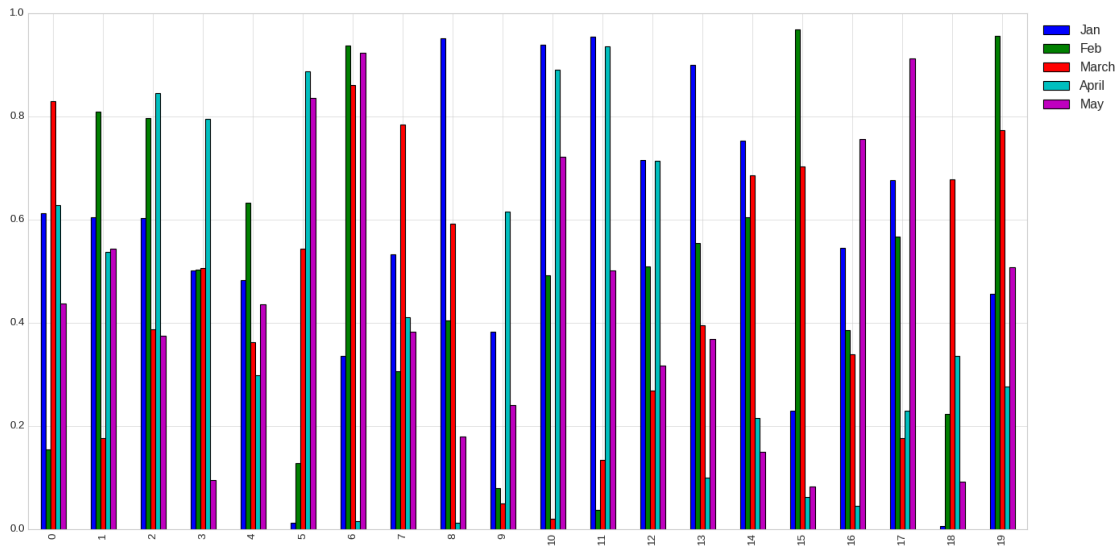
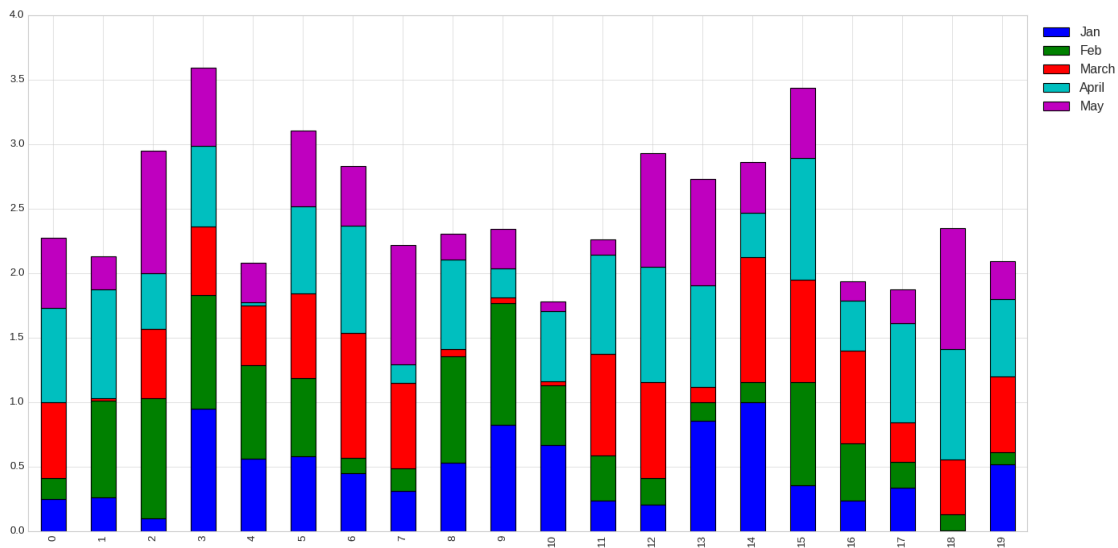Out[116]: <matplotlib.legend.Legend at 0x7f307a610780>


```
In [123]: import pandas as pd
          import numpy as np
          df = pd.DataFrame(np.random.rand(20,5),columns=['Jan','Feb','March','April', 'May'])
          df.plot.bar(figsize=(20, 10)).legend(bbox_to_anchor=(1.1, 1))
```

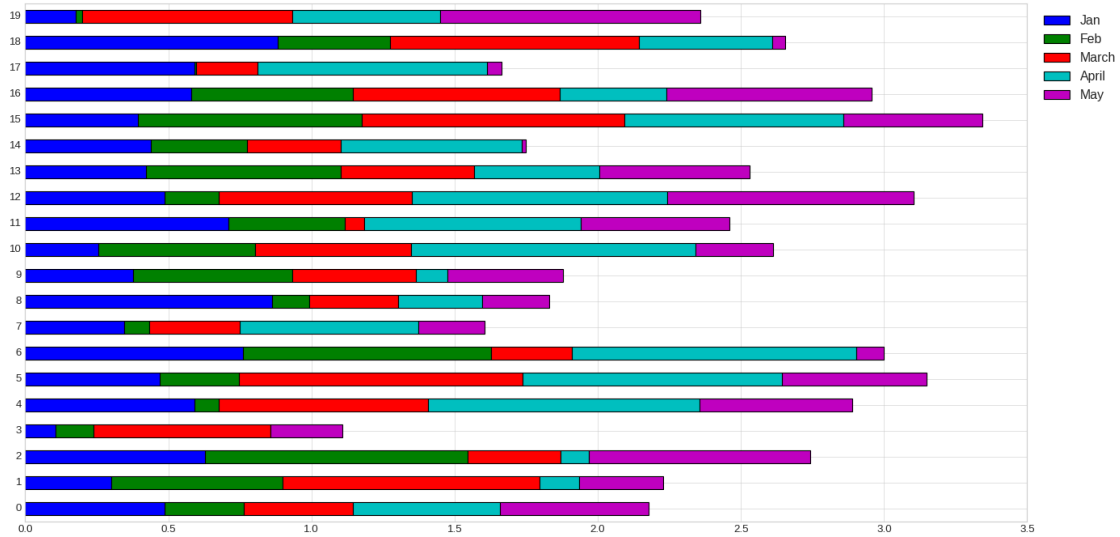Out[123]: <matplotlib.legend.Legend at 0x7f3079eef048>

16
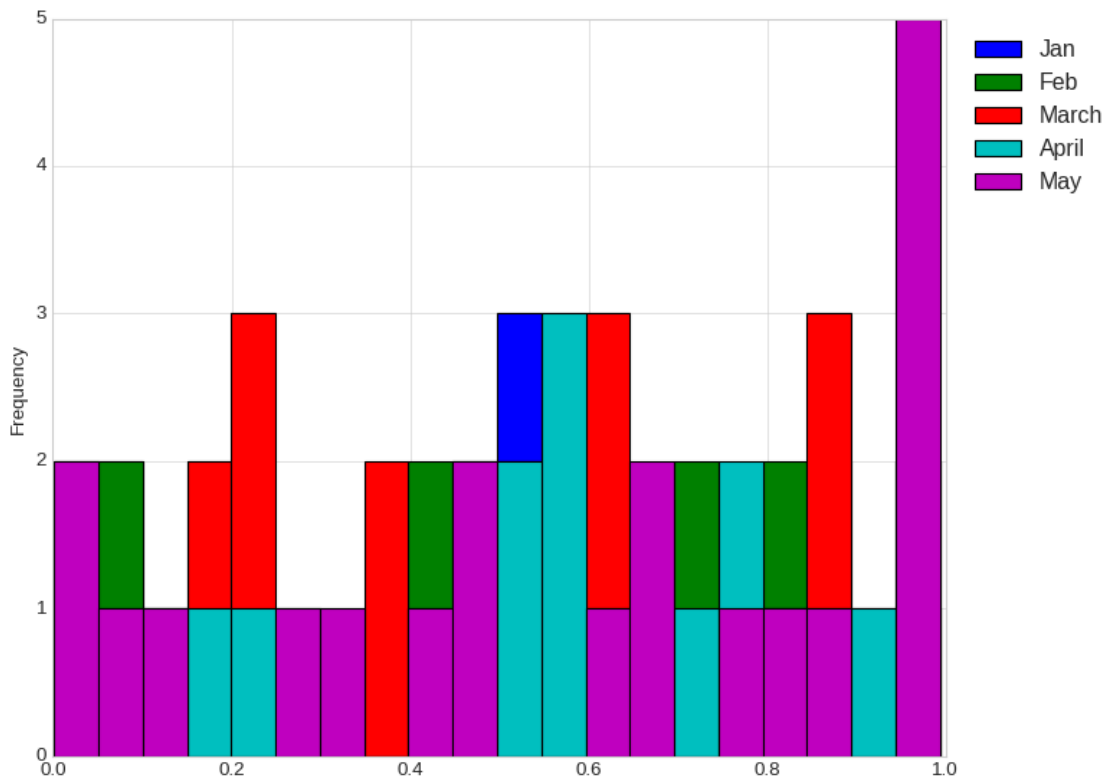
```
In [124]: import pandas as pd
          df = pd.DataFrame(np.random.rand(20,5),columns=['Jan','Feb','March','April', 'May'])
          df.plot.bar(stacked=True, figsize=(20, 10)).legend(bbox_to_anchor=(1.1, 1))

Out[124]: <matplotlib.legend.Legend at 0x7f307b422208>
```



```
In [126]: import pandas as pd
          df = pd.DataFrame(np.random.rand(20,5),columns=['Jan','Feb','March','April', 'May'])
          df.plot.barh(stacked=True, figsize=(20, 10)).legend(bbox_to_anchor=(1.1, 1))

Out[126]: <matplotlib.legend.Legend at 0x7f307b681860>
```
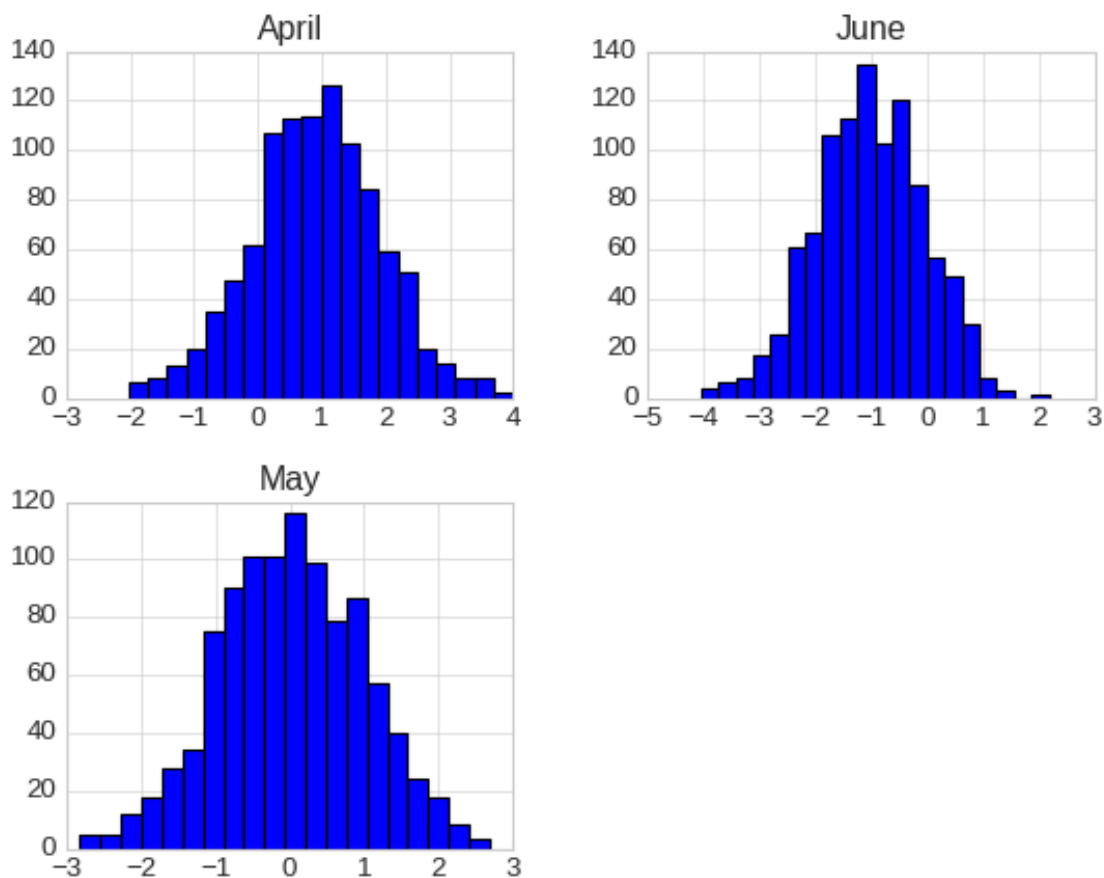
17

```
In [139]: import pandas as pd
          import numpy as np

          df=pd.DataFrame({'April':np.random.randn(1000)+1,'May':np.random.randn(1000),'June':
          np.random.randn(1000) - 1}, columns=['April', 'May', 'June'])

          df.hist(bins=20)
```
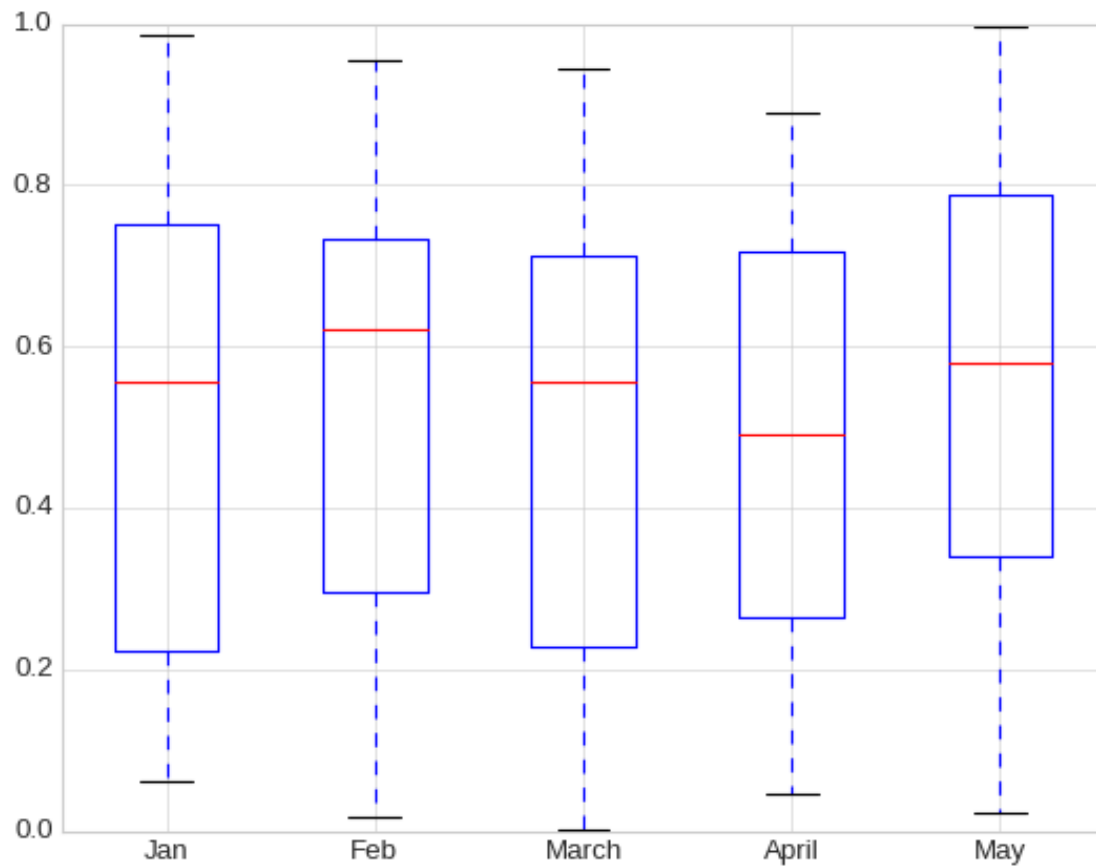
Out[139]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f307be000b8>,
                  <matplotlib.axes._subplots.AxesSubplot object at 0x7f30802bcda0>],
                 [<matplotlib.axes._subplots.AxesSubplot object at 0x7f30805ca400>,
                  <matplotlib.axes._subplots.AxesSubplot object at 0x7f30808adac8>]],
                dtype=object)


```
In [140]: import pandas as pd
          import numpy as np
          df = pd.DataFrame(np.random.rand(20,5),columns=['Jan','Feb','March','April', 'May'])
          df.plot.box()
```

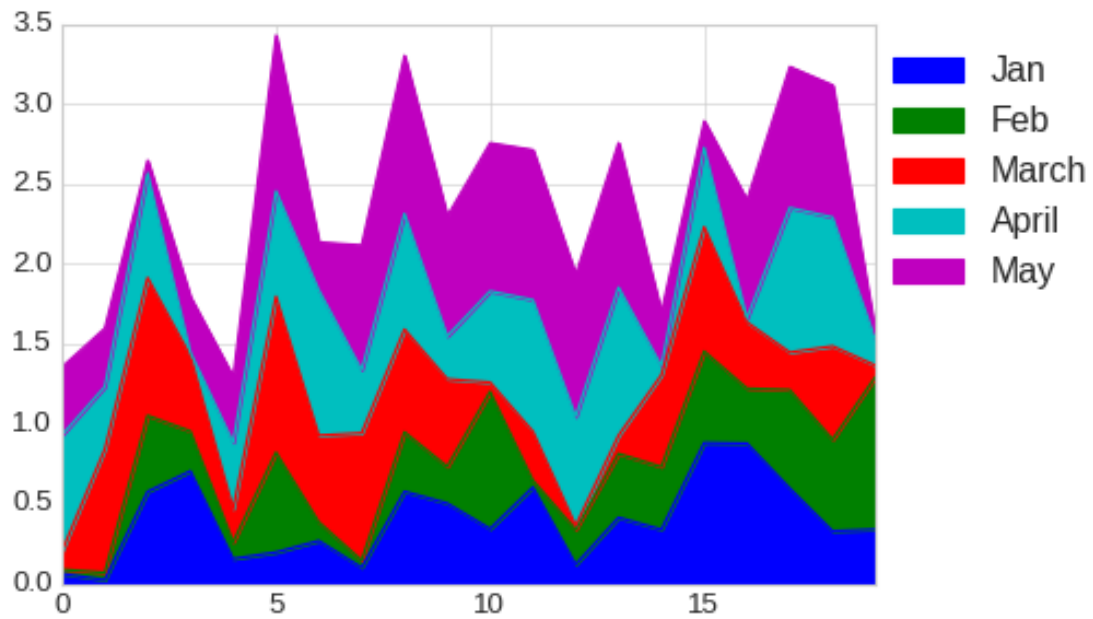Out[140]: <matplotlib.axes._subplots.AxesSubplot at 0x7f307bf553c8>

In [145]: import pandas as pd
          import numpy as np
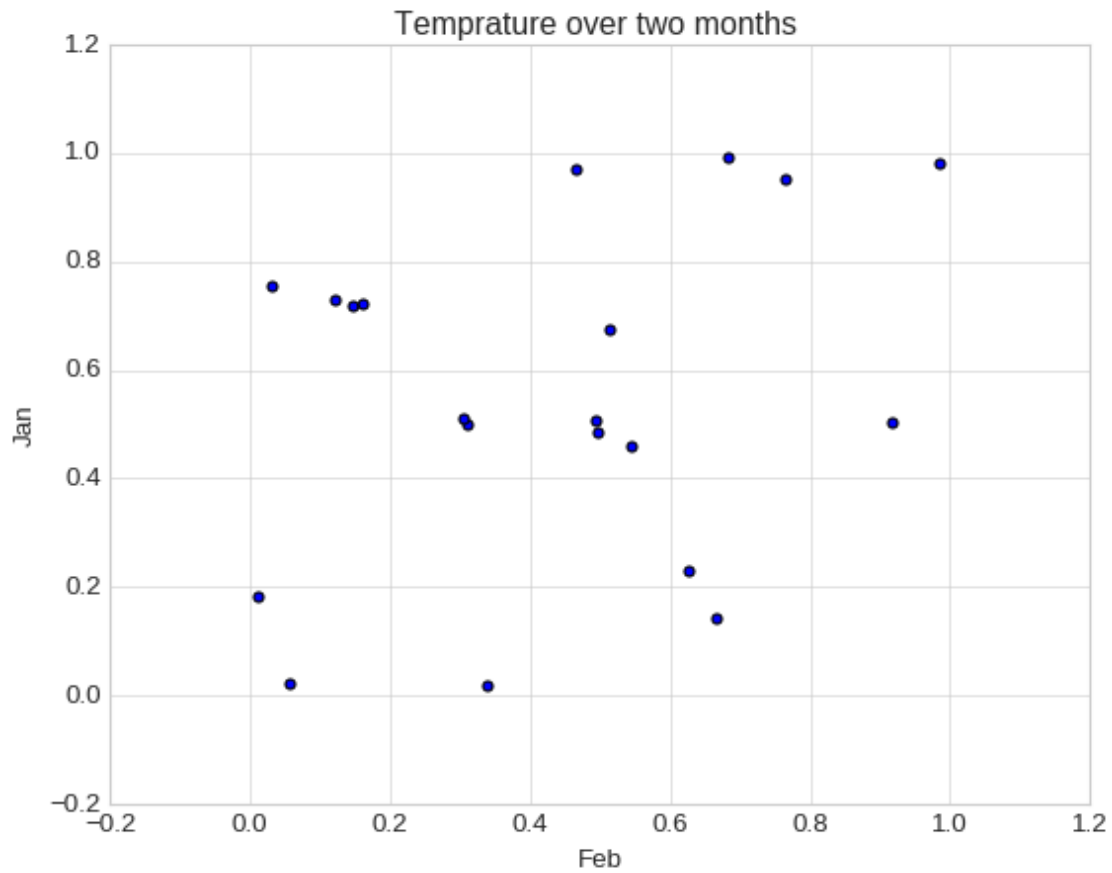
          df = pd.DataFrame(np.random.rand(20,5),columns=['Jan','Feb','March','April', 'May'])
          df.plot.area(figsize=(6, 4)).legend(bbox_to_anchor=(1.3, 1))

Out[145]: <matplotlib.legend.Legend at 0x7f30803d7a90>

```
In [150]: import pandas as pd
          import numpy as np
          df = pd.DataFrame(np.random.rand(20,5),columns=['Jan','Feb','March','April', 'May'])
          df.plot.scatter(x='Feb', y='Jan', title='Temprature over two months ')
```
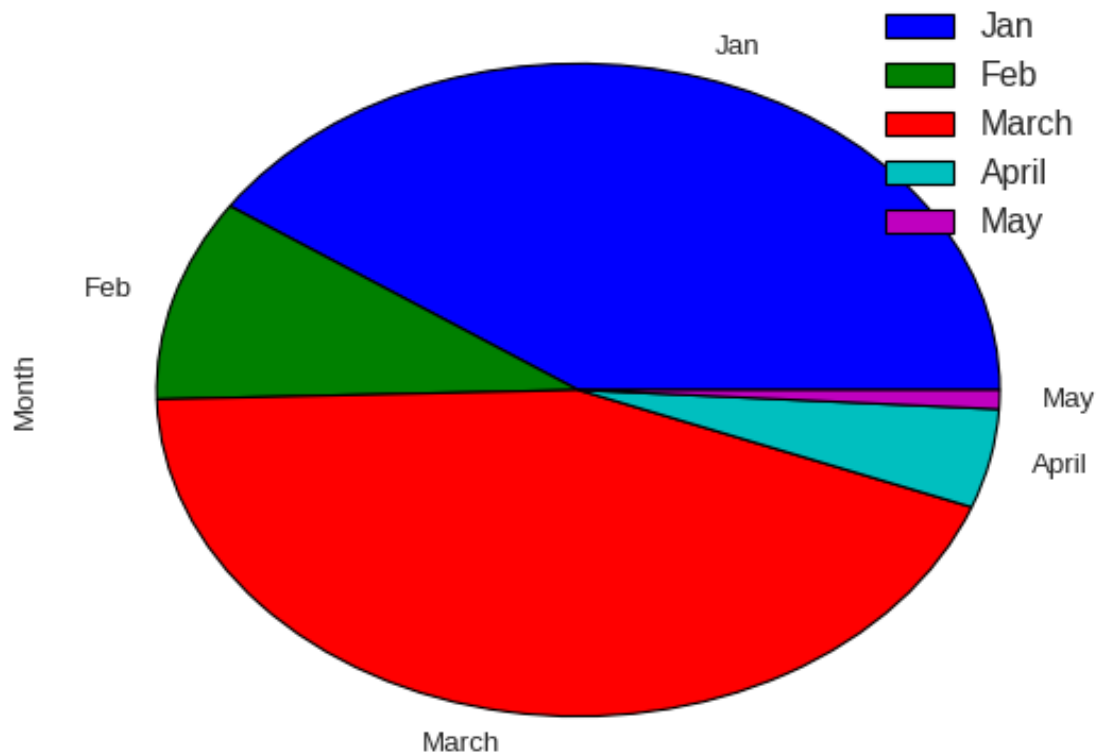
```
Out[150]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3080185ef0>
```

Temprature over two months

```
In [155]: import pandas as pd
          import numpy as np

          df = pd.DataFrame(10 * np.random.rand(5), index=['Jan','Feb','March','April', 'May'],
          df.plot.pie(subplots=True)

Out[155]: array([<matplotlib.axes._subplots.AxesSubplot object at 0x7f30802f8128>],
              dtype=object)
```

6 Exercise

```
In [14]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt

         salesMen = ['Ahmed', 'Omar', 'Ali', 'Ziad', 'Salwa', 'Lila']
         Mobile_Sales  = [2540, 1370, 1320, 2000, 2100, 2150]
         TV_Sales  = [2200, 1900, 2150, 1850, 1770, 2000]

         df = pd.DataFrame()
         df ['Name']  =salesMen
         df ['Mobile_Sales'] =  Mobile_Sales
         df ['TV_Sales'] =  TV_Sales
         df.set_index("Name",drop=True,inplace=True)

In [15]: df

Out[15]:          Mobile_Sales   TV_Sales
         Name
```

```
Ahmed            2540        2200
Omar             1370        1900
Ali              1320        2150
Ziad             2000        1850
Salwa            2100        1770
Lila             2150        2000
```
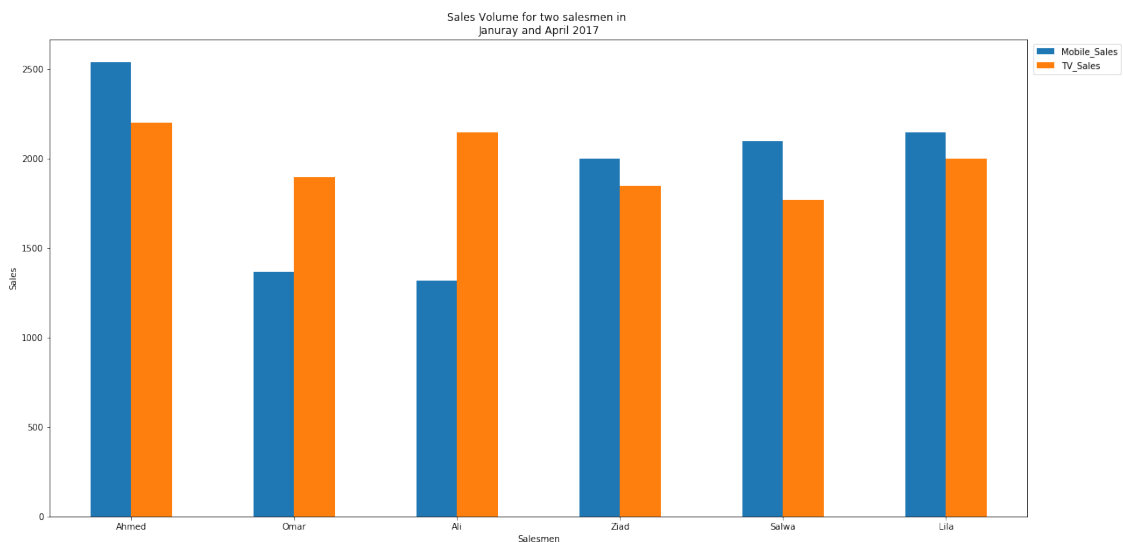
In [16]: df.plot.bar(figsize=(20, 10), rot=0).legend(bbox_to_anchor=(1.1, 1))
         plt.xlabel('Salesmen')
         plt.ylabel('Sales')
         plt.title('Sales Volume for two salesmen in \nJanuray and April 2017')
         plt.show()

In [17]: import pandas as pd
         import numpy as np
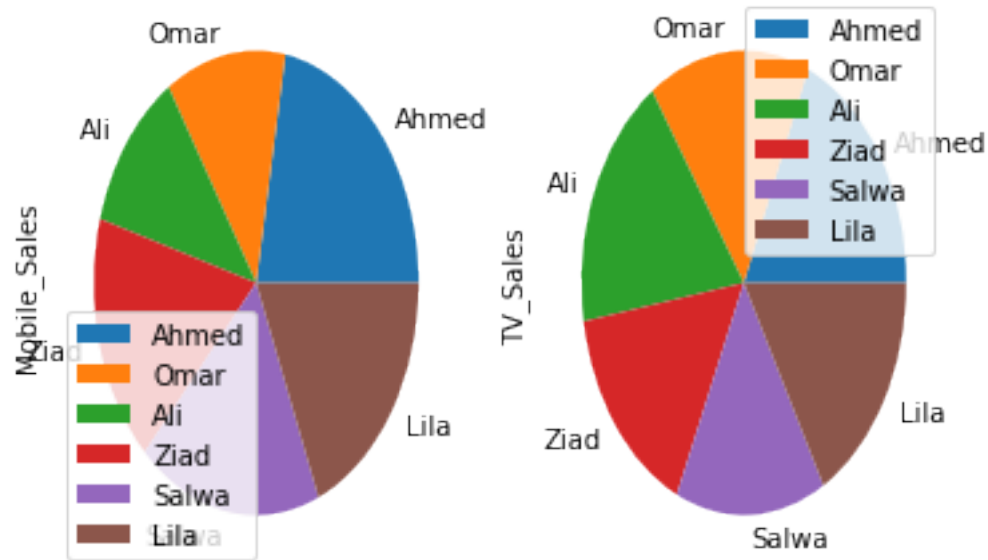   import matplotlib.pyplot as plt

         salesMen = ['Ahmed', 'Omar', 'Ali', 'Ziad', 'Salwa', 'Lila']
         Mobile_Sales = [2540, 1370, 1320, 2000, 2100, 2150]
         TV_Sales = [2200, 1900, 2150, 1850, 1770, 2000]

         df = pd.DataFrame()
         df ['Name'] =salesMen
         df ['Mobile_Sales'] = Mobile_Sales
         df ['TV_Sales'] = TV_Sales
         df.set_index("Name",drop=True,inplace=True)

         df.plot.pie(subplots=True)

24
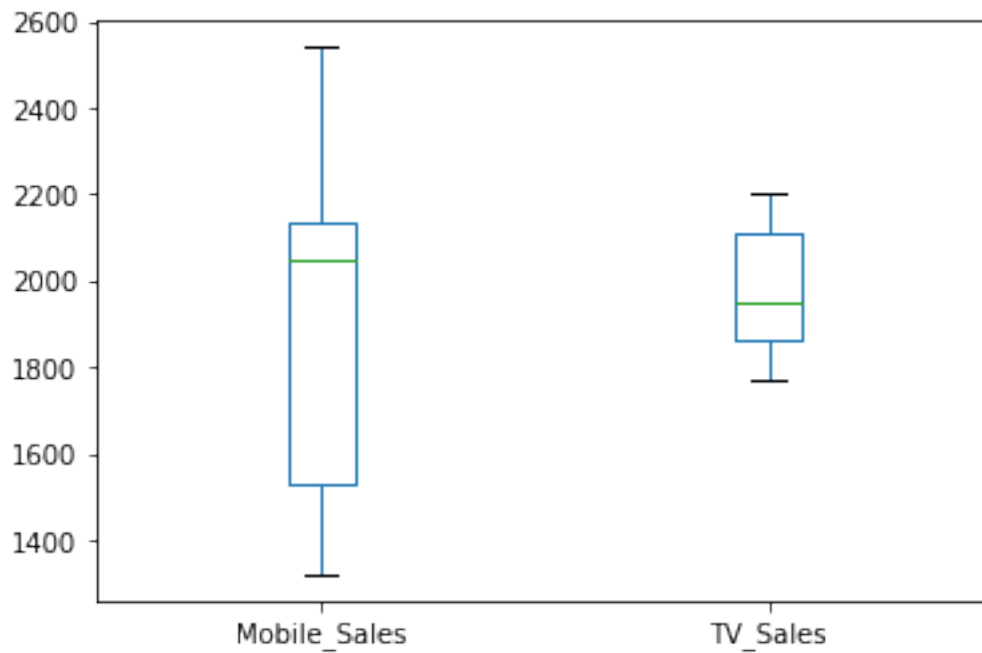
```
Out[17]: array([<matplotlib.axes._subplots.AxesSubplot object at 0x7f18625b8eb8>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7f1862515f60>],
                dtype=object)
```
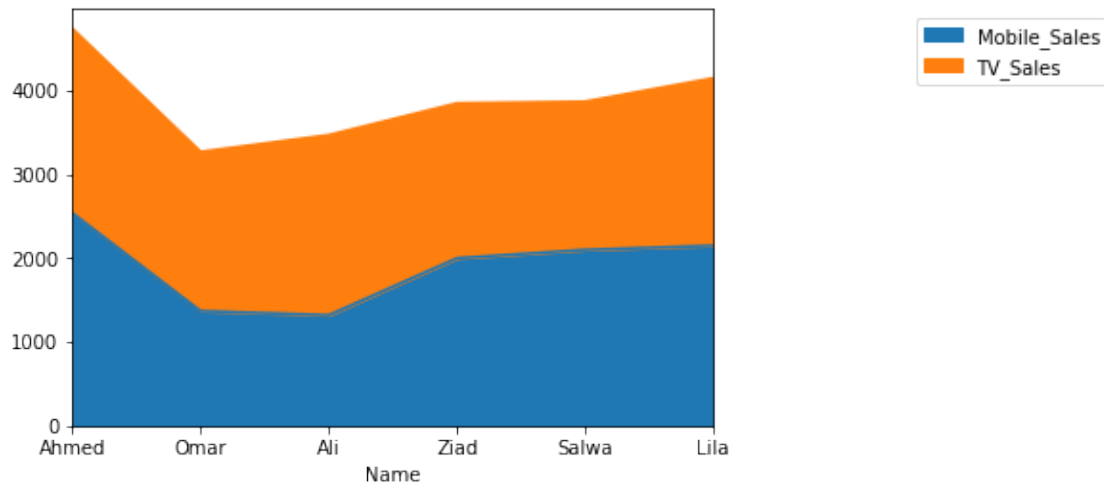


```
In [18]: df.plot.box()
```

```
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1862580550>
```

In [19]: df.plot.area(figsize=(6, 4)).legend(bbox_to_anchor=(1.3, 1))

Out[19]: <matplotlib.legend.Legend at 0x7f18621ab748>

In [20]: df.plot.bar(stacked=True, figsize=(20, 10)).legend(bbox_to_anchor=(1.1, 1))

Out[20]: <matplotlib.legend.Legend at 0x7f18620d6e48>