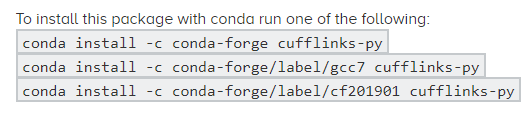


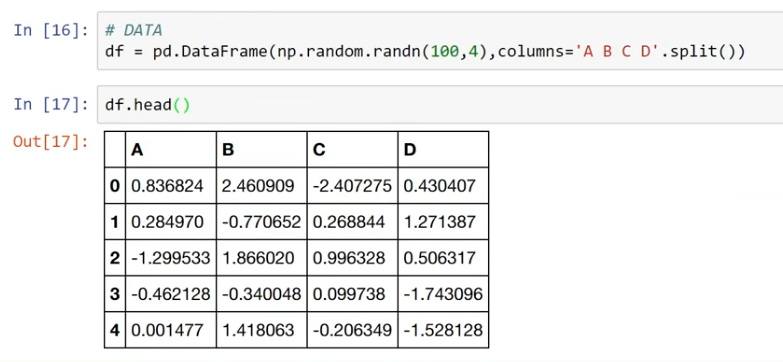
First Install Cufflinks

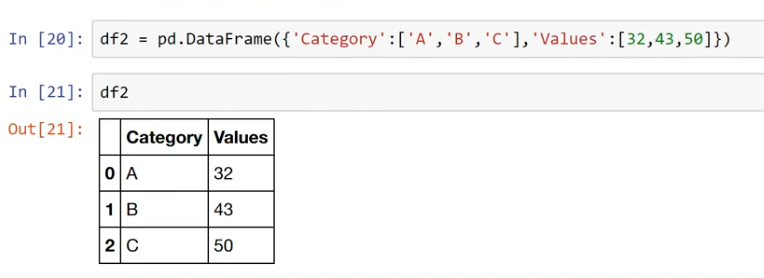


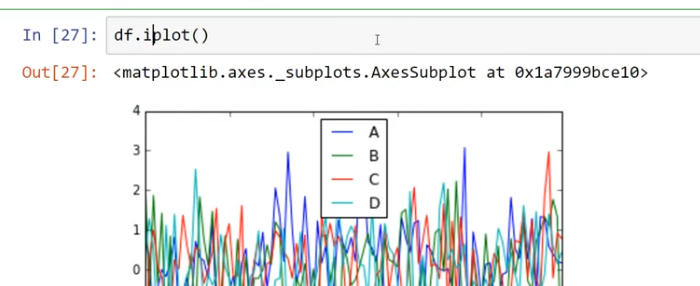
Next Instal Plotly

conda install -c plotly plotly=4.1.0

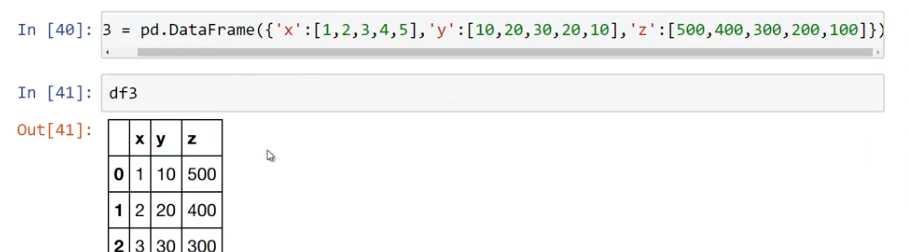










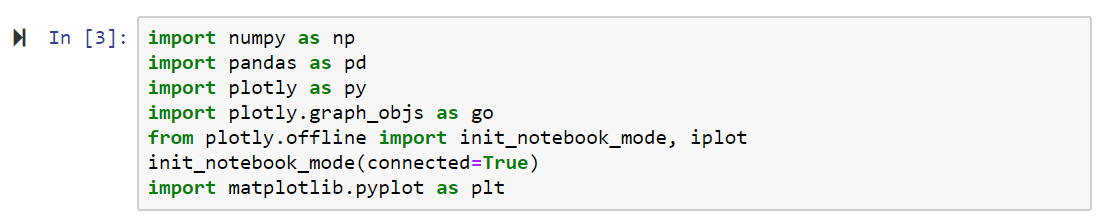


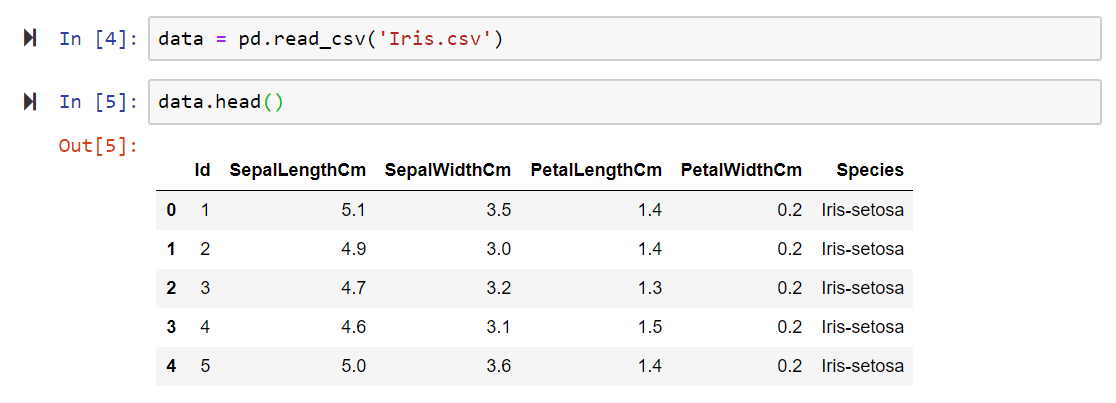


Different canged value



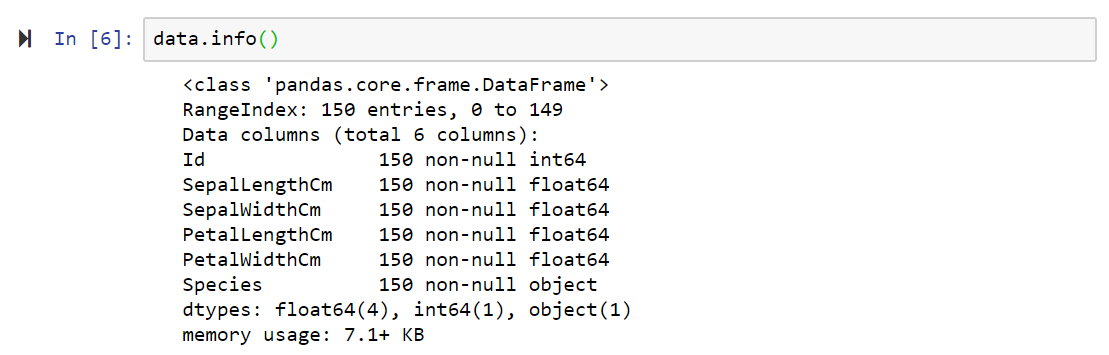
**Handling With IRIS DataSet:::**

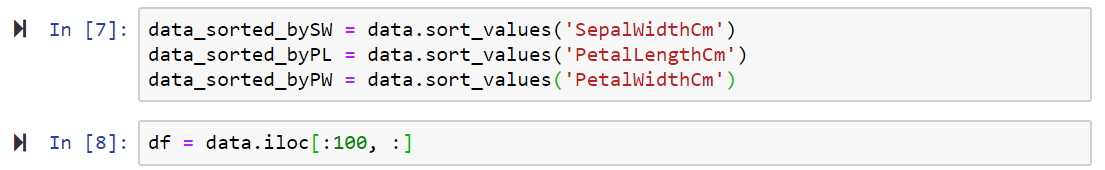
****

****

Scatter Plot

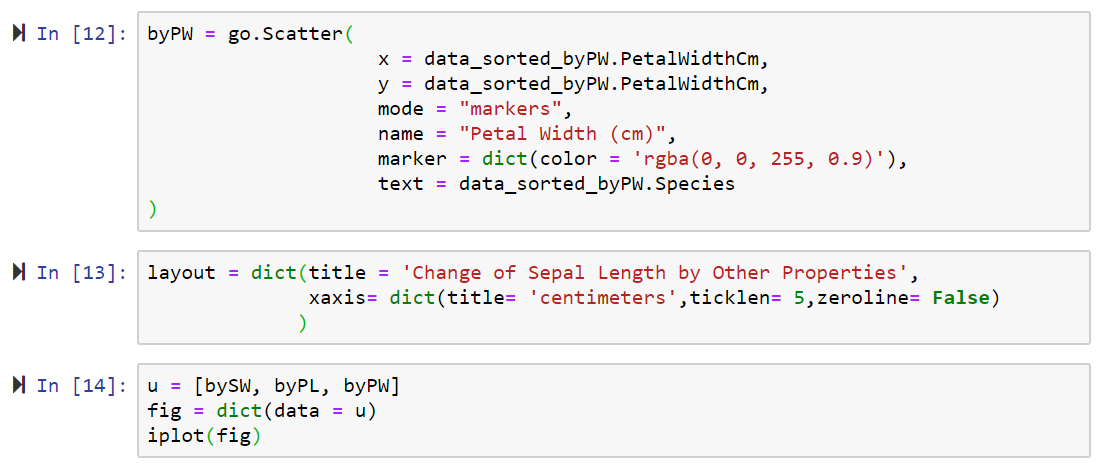
Scatter plot is a good way to visualize the correlations among features. I will be examining the correlation of SepalLengthCm with other features. So Sepal Length will be our y-axis, others will be laying on the x-axis. And I sorted and kept them in distinct dataframes to see correlations clearly.

****

****

**import plotly.graph\_objects as go**

****

****

* Seems like Petal Width and Sepal Length has a very strong correlation.
* We can say there is a correlation between Petal Length and Sepal Length, but not like the one above.
* There is no correlation between the Sepal Length and Sepal Width.

## Bar Plot

Let's visualize each species' average lengths, so we will be able to see how features change as genre of the flower changes.

data1 = data.groupby(data.Species).mean()

data1['Species'] = data1.index

t1 = go.Bar(

x = data1.Species,

y = data1.SepalLengthCm,

name = "Sepal Length (cm)",

marker = dict(color = 'rgba(160, 55, 0, 0.8)', line = dict(color = 'rgb(0,0,0)', width = 1.5)),

text = data1.Species

)

t2 = go.Bar(

x = data1.Species,

y = data1.SepalWidthCm,

name = "Sepal Width (cm)",

marker = dict(color = 'rgba(0, 55, 160, 0.8)', line = dict(color = 'rgb(0,0,0)', width = 1.5)),

text = data1.Species

)

t3 = go.Bar(

x = data1.Species,

y = data1.PetalLengthCm,

name = "Petal Length (cm)",

marker = dict(color = 'rgba(20, 55, 30, 0.8)', line = dict(color = 'rgb(0,0,0)', width = 1.5)),

text = data1.Species

)

t4 = go.Bar(

x = data1.Species,

y = data1.PetalWidthCm,

name = "Petal Width (cm)",

marker = dict(color = 'rgba(70, 55, 160, 0.8)', line = dict(color = 'rgb(0,0,0)', width = 1.5)),

text = data1.Species

)

b = [t1,t2,t3,t4]

layout\_bar = go.Layout(barmode = "group")

fig\_bar = go.Figure(data = b, layout = layout\_bar)

iplot(fig\_bar)

As we can clearly observe from the barplot above; Sepal Length, Petal Length and Petal Width features grow as we walk in species Iris-setosa, Iris-versicolor and Iris-virginica, respectively.

## Bubble Chart

Now I want to visualize all four attributes in one chart. Bubble chart is an appropriate way of visualization for this kind of purposes.  
Let's decide what each parameter corresponds to:  
x : PetalLengthCm  
y : PetalWidthCm  
color : SepalWidthCm  
size : SepalLengthCm

fig\_bubble = [

{

'x' : data.PetalLengthCm,

'y' : data.PetalWidthCm,

'mode' : 'markers',

'marker' : {

'color' : data.SepalWidthCm,

'size' : data.SepalLengthCm,

'showscale' : True

},

'text' : data.Species

}

]

iplot(fig\_bubble)

## Boxplot

Boxplot is always the best choice, if we want to get some statistical information from the data.

t1\_box = go.Box(

name = 'Sepal Length (cm)',

y = data.SepalLengthCm,

marker = dict(color = 'rgba(160,160,50,0.7)')

)

t2\_box = go.Box(

name = 'Sepal Width (cm)',

y = data.PetalWidthCm,

marker = dict(color = 'rgba(50,160,150,0.7)')

)

t3\_box = go.Box(

name = 'Petal Length (cm)',

y = data.PetalLengthCm,

marker = dict(color = 'rgba(160,60,150,0.7)')

)

t4\_box = go.Box(

name = 'Petal Width (cm)',

y = data.SepalWidthCm,

marker = dict(color = 'rgba(150,160,150,0.7)')

)

fig\_box = [t1\_box, t2\_box, t3\_box, t4\_box]

iplot(fig\_box)

Petal Width has a short range from 2 to 4.4, with some outliers.

Petal Length has a wide range between 1 cm and 6.9 cm, but without outliers.

## Scatterplot Matrix

Now let's examine the relations by crosschecking each feature by scatterplot matrix.

import plotly.figure\_factory as ff

data\_ff = data.loc[:, ['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]

data\_ff['index'] = np.arange(1, len(data\_ff)+1)

fig\_ff = ff.create\_scatterplotmatrix(data\_ff, diag = 'box', index = 'index', colormap = 'Blues', colormap\_type = 'cat', height = 800, width = 800)

iplot(fig\_ff)

## 3D Scatter

Let's jump into third dimension, and decide what each variable correspond to:  
x : SepalLengthCm  
y : SepalWidthCm  
z : PetalLengthCm  
color : PetalWidthCm

trace\_3d = go.Scatter3d(

x = data.SepalLengthCm,

y = data.SepalWidthCm,

z = data.PetalLengthCm,

mode = 'markers',

opacity = 0.7,

*#name = data.Species,*

marker = dict(

size = 5,

color = data.PetalWidthCm

)

)

list\_3d = [trace\_3d]

fig\_3d = go.Figure(data = list\_3d)

iplot(fig\_3d)

## Iris-setosa : pink Iris-versicolor : lime Iris-virginica : blue

i\_setosa = data[data['Species'] == 'Iris-setosa']

i\_versicolor = data[data['Species'] == 'Iris-versicolor']

i\_virginica = data[data['Species'] == 'Iris-virginica']

*# Iris-setosa*

trace\_setosa = go.Scatter3d(

x = i\_setosa.SepalLengthCm,

y = i\_setosa.SepalWidthCm,

z = i\_setosa.PetalLengthCm,

mode = 'markers',

opacity = 0.7,

name = "Iris-setosa",

marker = dict(

size = 5,

color = 'rgba(255,102, 255,0.8)'

)

)

*# Iris-versicolor*

trace\_versicolor = go.Scatter3d(

x = i\_versicolor.SepalLengthCm,

y = i\_versicolor.SepalWidthCm,

z = i\_versicolor.PetalLengthCm,

mode = 'markers',

opacity = 0.7,

name = "Iris-versicolor",

marker = dict(

size = 5,

color = 'rgba(102, 255, 51, 0.8)'

)

)

*# Iris-virginica*

trace\_virginica = go.Scatter3d(

x = i\_virginica.SepalLengthCm,

y = i\_virginica.SepalWidthCm,

z = i\_virginica.PetalLengthCm,

mode = 'markers',

opacity = 0.7,

name = "Iris-virginica",

marker = dict(

size = 5,

color = 'rgba(51, 102, 255, 0.8)'

)

)

list\_3d = [trace\_setosa, trace\_versicolor, trace\_virginica]

fig\_3d = go.Figure(data = list\_3d)

iplot(fig\_3d)