

Pandas		Aggregation and Grouping		Matplotlib		Matplotlib (cont)	
import pandas as pd	pd.series([values]) ad = {} area = pd.series(ad)	Aggregation Functions	count()   Total number of items first(), last()   First and last item mean(), median()   Mean and median min(), max()   Minimum and maximum std(), var()   Standard deviation and variance mad()   Mean absolute deviation prod()   Product of all items sum()   Sum of all items	Line Plots	Set linspace x = np.linspace(0, 10, 100)	Other ways for plt	plt.xlabel() → ax.set_xlabel() plt.ylabel() → ax.set_ylabel() plt.xlim() → ax.set_xlim() plt.ylim() → ax.set_ylim() plt.title() → ax.set_title()
Retrieving Values	area["a"] To see all keys: area.keys() data.items()	Grouping	name.groupby("-key")	Creating figure and axis	fig = plt.figure() ax = plt.axes()	Histograms	fig = plt.figure() ax = plt.axes() ax.hist(data);
Dataframe as dictionary	area = pd.series-as({...}) data = pd.DataFrame({"area":area,})	Pivot Tables		Add graph and x,y	x = np.linspace(0, 10, 1000) y = np.sin(x) plt.plot(x,y) plt.show()		
Opening data	import pandas as pd import numpy as np  dat = np.genfromtxt('phoneBook.csv',delimiter=',',skip_header=1, dtype='<U16')	Pivot Tables by Hand	Require groupby  name.pivot_table("what is taking the action", index = "groupby row" , columns = "groupbycol")	Changing linestyle and color	plt.plot(x,y,linestyle='--', color='c')		
Grouping	index = pd.MultiIndex.from_tuples(index) index  pop = pop.reindex(index) pop  pop[:, 2010]	Aggregation Functions	name.pivot_table(index = "groupby row" , columns = "-groupbycol" , aggfunc={'taking action':sum, 'taking action':'mean'})	Multile curves and a legend	plt.plot(x,np.sin(x-.5),color='g',label="sin(x-0.5)")  plt.plot(x,np.sin(x-1),color='pink', label = "sin(x-1)")  plt.plot(x,np.cos(x-0.5),color='c',linestyle='-',label = "-cos(x-0.5)") plt.legend() plt.show()		
Merging and Joining				Adding limits	plt.xlim(-5,12) plt.ylim(-2,2)		
Merging	pd.merge()  df = pd.merge()			Scatter Plot	x = np.random.randint(-1000,100-0,150) y = np.random.randint(-1000,100-0,150) plt.scatter(x,y)  or plt.plot(x,y,'o');		
Many to one	Duplicate entries display('df3', 'df4', 'pd.merge(df3, df4)')						
Merge Key	Add on = "key column name"						
Drop	.drop('name', axis=1)						

