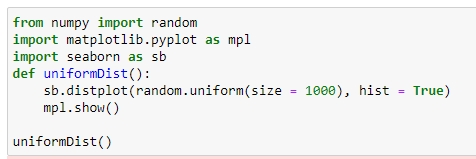
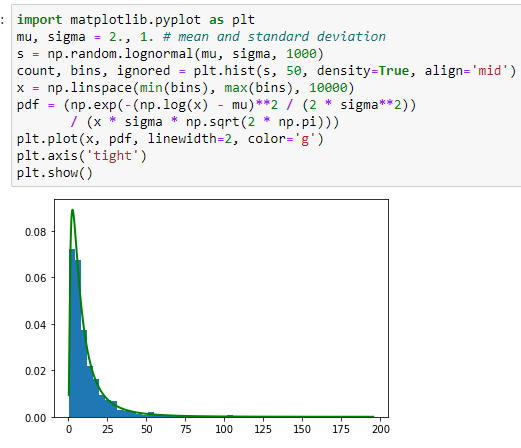
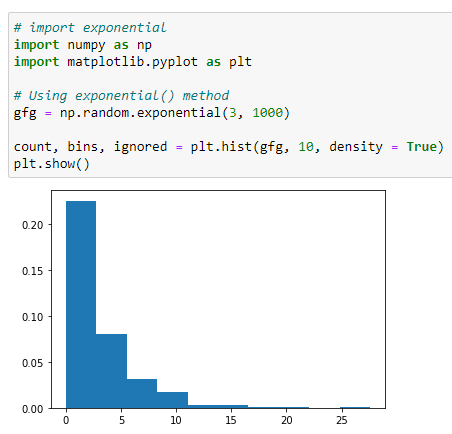


The below [Python](https://www.simplilearn.com/learn-the-basics-of-python-article) code is a simple example of continuous distribution taking 1000 samples of random variables.







# importing pandas as pd

import pandas as pd

# importing numpy as np

import numpy as np

# dictionary of lists

dict = {'First Score':[100, 90, np.nan, 95],

        'Second Score': [30, 45, 56, np.nan],

        'Third Score':[np.nan, 40, 80, 98]}

# creating a dataframe from list

df = pd.DataFrame(dict)

# using isnull() function

df.isnull()

# importing pandas package

import pandas as pd

# making data frame from csv file

data = pd.read\_csv("employees.csv")

# creating bool series True for NaN values

bool\_series = pd.isnull(data["Gender"])

# filtering data

# displaying data only with Gender = NaN

data[bool\_series]

# importing pandas as pd

import pandas as pd

# importing numpy as np

import numpy as np

# dictionary of lists

dict = {'First Score':[100, 90, np.nan, 95],

        'Second Score': [30, 45, 56, np.nan],

        'Third Score':[np.nan, 40, 80, 98]}

# creating a dataframe using dictionary

df = pd.DataFrame(dict)

# using notnull() function

df.notnull()

# importing pandas package

import pandas as pd

# making data frame from csv file

data = pd.read\_csv("employees.csv")

# creating bool series True for NaN values

bool\_series = pd.notnull(data["Gender"])

# filtering data

# displaying data only with Gender = Not NaN

data[bool\_series

# importing pandas as pd

import pandas as pd

# importing numpy as np

import numpy as np

# dictionary of lists

dict = {'First Score':[100, 90, np.nan, 95],

        'Second Score': [30, 45, 56, np.nan],

        'Third Score':[np.nan, 40, 80, 98]}

# creating a dataframe from dictionary

df = pd.DataFrame(dict)

# filling missing value using fillna()

df.fillna(0)

# importing pandas as pd

import pandas as pd

# importing numpy as np

import numpy as np

# dictionary of lists

dict = {'First Score':[100, 90, np.nan, 95],

        'Second Score': [30, 45, 56, np.nan],

        'Third Score':[np.nan, 40, 80, 98]}

# creating a dataframe from dictionary

df = pd.DataFrame(dict)

# filling a missing value with

# previous ones

df.fillna(method ='pad')

# importing pandas as pd

import pandas as pd

# importing numpy as np

import numpy as np

# dictionary of lists

dict = {'First Score':[100, 90, np.nan, 95],

        'Second Score': [30, 45, 56, np.nan],

        'Third Score':[np.nan, 40, 80, 98]}

# creating a dataframe from dictionary

df = pd.DataFrame(dict)

# filling  null value using fillna() function

df.fillna(method ='bfill')

# importing pandas package

import pandas as pd

# making data frame from csv file

data = pd.read\_csv("employees.csv")

# Printing the first 10 to 24 rows of

# the data frame for visualization

data[10:25]

# importing pandas package

import pandas as pd

# making data frame from csv file

data = pd.read\_csv("employees.csv")

# filling a null values using fillna()

data["Gender"].fillna("No Gender", inplace = True)

data

# importing pandas package

import pandas as pd

# making data frame from csv file

data = pd.read\_csv("employees.csv")

# Printing the first 10 to 24 rows of

# the data frame for visualization

data[10:25]

# importing pandas package

import pandas as pd

# making data frame from csv file

data = pd.read\_csv("employees.csv")

# will replace  Nan value in dataframe with value -99

data.replace(to\_replace = np.nan, value = -99)

# importing pandas as pd

import pandas as pd

# Creating the dataframe

df = pd.DataFrame({"A":[12, 4, 5, None, 1],

                   "B":[None, 2, 54, 3, None],

                   "C":[20, 16, None, 3, 8],

                   "D":[14, 3, None, None, 6]})

# Print the dataframe

Df

# to interpolate the missing values

df.interpolate(method ='linear', limit\_direction ='forward')

# importing pandas as pd

import pandas as pd

# importing numpy as np

import numpy as np

# dictionary of lists

dict = {'First Score':[100, 90, np.nan, 95],

        'Second Score': [30, np.nan, 45, 56],

        'Third Score':[52, 40, 80, 98],

        'Fourth Score':[np.nan, np.nan, np.nan, 65]}

# creating a dataframe from dictionary

df = pd.DataFrame(dict)

# using dropna() function

df.dropna()

# importing pandas as pd

import pandas as pd

# importing numpy as np

import numpy as np

# dictionary of lists

dict = {'First Score':[100, np.nan, np.nan, 95],

        'Second Score': [30, np.nan, 45, 56],

        'Third Score':[52, np.nan, 80, 98],

        'Fourth Score':[np.nan, np.nan, np.nan, 65]}

df = pd.DataFrame(dict)

# using dropna() function

df.dropna(how = 'all')

# importing pandas as pd

import pandas as pd

# importing numpy as np

import numpy as np

# dictionary of lists

dict = {'First Score':[100, np.nan, np.nan, 95],

        'Second Score': [30, np.nan, 45, 56],

        'Third Score':[52, np.nan, 80, 98],

        'Fourth Score':[60, 67, 68, 65]}

# creating a dataframe from dictionary

df = pd.DataFrame(dict)

# using dropna() function

df.dropna(axis = 1)

# importing pandas module

import pandas as pd

# making data frame from csv file

data = pd.read\_csv("employees.csv")

# making new data frame with dropped NA values

new\_data = data.dropna(axis = 0, how ='any')

new\_data

print("Old data frame length:", len(data))

print("New data frame length:", len(new\_data))

print("Number of rows with at least 1 NA value: ", (len(data)-len(new\_data)))

***Syntax:****DataFrame.isna()*

**Example #1:** Use isna() function to detect the missing values in a dataframe.

|  |
| --- |
| # importing pandas as pd  import pandas as pd    # Creating the dataframe  df = pd.read\_csv("nba.csv")    # Print the dataframe  df |
| # importing pandas as pd  import pandas as pd    # Creating the series  sr = pd.Series([12, 5, None, 5, None, 11])    # Print the series  sr |

# to detect the missing values

sr.isna()

1)Load the Iris dataset as a list of lists (each of the 150 lists should have 5 elements). Compute and print the mean and the standard deviation for each of the 4 measurement columns (i.e. sepal length and width, petal length and width). Compute and print the mean and the standard deviation for each of the 4 measurement columns, separately for each of the three Iris species (Versicolor, Virginica and Setose). Which measurement would you consider “best”, if you were to guess the Iris species based

only on those four values?

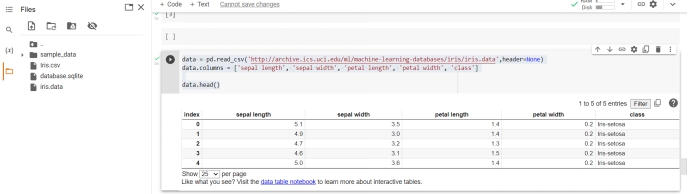
First, download the [Iris dataset](http://archive.ics.uci.edu/ml/datasets/Iris) from the UCI machine learning repository.

import pandas as pd

data = pd.read\_csv('http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data',header=None)

data.columns = ['sepal length', 'sepal width', 'petal length', 'petal width', 'class']

data.head()



from pandas.api.types import is\_numeric\_dtype

for col in data.columns:

if is\_numeric\_dtype(data[col]):

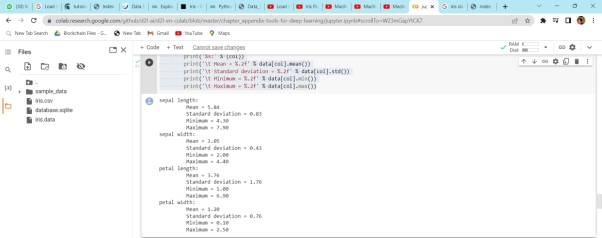
print('%s:' % (col))

print('\t Mean = %.2f' % data[col].mean())

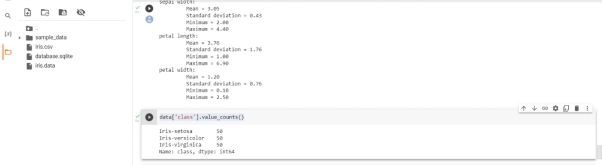
print('\t Standard deviation = %.2f' % data[col].std())

print('\t Minimum = %.2f' % data[col].min())

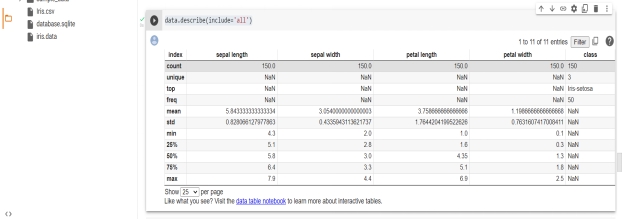
print('\t Maximum = %.2f' % data[col].max())



data['class'].value\_counts()

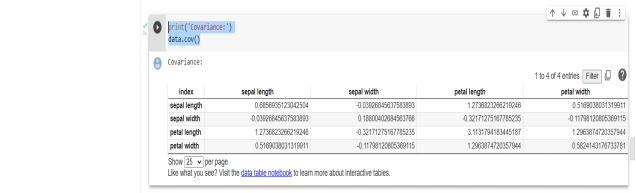


data.describe(include='all')



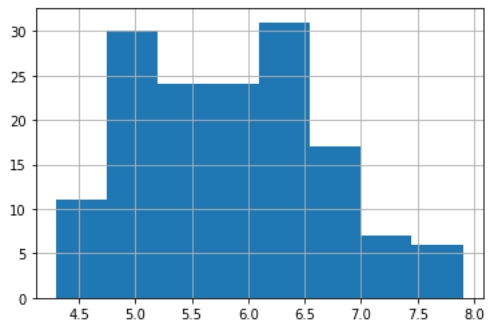
print('Covariance:')

data.cov()

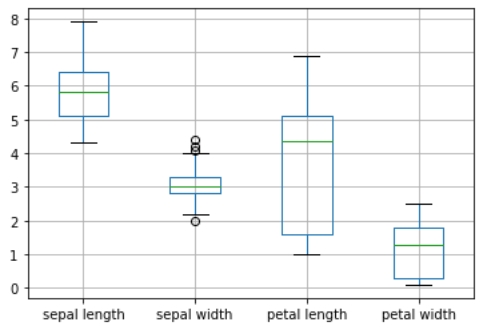


%matplotlib inline

data['sepal length'].hist(bins=8)



data.boxplot()



import matplotlib.pyplot as plt

fig, axes = plt.subplots(3, 2, figsize=(12,12))

index = 0

for i in range(3):

for j in range(i+1,4):

ax1 = int(index/2)

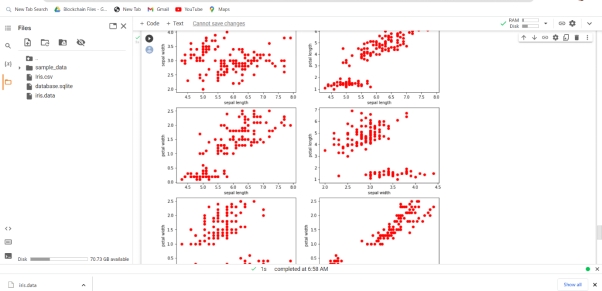
ax2 = index % 2

axes[ax1][ax2].scatter(data[data.columns[i]], data[data.columns[j]], color='red')

axes[ax1][ax2].set\_xlabel(data.columns[i])

axes[ax1][ax2].set\_ylabel(data.columns[j])

index = index + 1



from pandas.plotting import parallel\_coordinates

%matplotlib inline

parallel\_coordinates(data, 'class')

