Lab – Assignment 03 – 1 fill out result note

|  |  |  |
| --- | --- | --- |
| **No.** | **Code.** | **Result Note** |
| **1** | **import numpy as np**  **import pandas as pd**  **import sklearn as sk**  **import opendatasets as od**  **import os**  **import matplotlib.pyplot as plt**  **import seaborn as sns** | **เป็นการ import library ต่างๆ ถ้าไม่มีให้ทำการติดตั้งด้วย**  **pip install แล้วตามด้วยชื่อ libary** |
| **2** | **data\_scikit = sk.datasets.load\_iris() #see more datasets from https://scikit-learn.org/stable/datasets.html**  **data\_kaggle = od.downloads(dataset url)**  **data\_local = pd.read\_csv(data path)**  **data\_git = pd.read\_csv(raw url of github .csv file)** | **เป็นการ load data iris จาก science kit learn ใส่ในตัวแปร data\_scikit** |
| **3** | **data\_iris = data\_scikit.data**  **print(data\_iris [:5])**  **print(data\_iris.size)**  **print(data\_iris.shape)**  **print(type(data\_iris))**  **print(len(data\_iris))** | **ในตัว data\_scikit มี data json object ให้เลือก จีงเลือกข้อมูลของ data ใส่ในตัวแปร data\_iris**  **แสดงข้อมูล 5 อันดับแรก**  **แสดงขนาดของข้อมูล**  **แสดงมิติของข้อมูล หรือ แกนของข้อมูล หรืออาจเทียบได้ว่า Feature**  **แสดงชนิดของข้อมูล**  **แสดงความยาวของมิติแรก** |
| **4** | **secret\_array = np.array([['Wilson', 2212], ['Thorn', 1705], ['Mulberry', 2606], ['Blaze', np.nan], ['Saria', 2304], ['Ifrit', 2207]])**  **secret\_df = pd.DataFrame(secret\_array, columns = [Column1, Column2, … ])**  **secret\_df.head(5)**  **secret\_df.sample(5)**  **secret\_df.dtypes**  **secret\_df.info()**  **secret.loc[:, Column]**  **secret\_df.isna()**  **secret\_df.dropna()**  **#Extra**  **secret\_df[Column2] = pd.to\_numeric(secret\_df[Column2], errors = ‘coerce’)**  **secret\_df.isna()**  **#** | **สร้าง array จาก numpy ที่มีมิติ 6,2**  **จากนั้นเปลี่ยนจาก array เป็น dataframe**  **ดูข้อมูล dataframe 5 ข้อมูลแรก**  **สุ่มดูข้อมูล ของ dataframe**  **ดูชนิดของ dataframe**  **ดูข้อมูล ของ dataframe**  **ดูข้อมูลแบบไม่มีตาราง และบอกชนิดของข้อมูลและชื่อของ columns**  **ดูว่าข้อมูลไม่ได้เติมไหม**  **ลบข้อมูลที่ไม่ได้เติมออก โดยลบ row ทิ้งทั้งแถว**  **Extra แปลงชนิดจาก object เป็น float แล้วจะสามารถลบ nan ได้** |
| **5** | **X = np.arange(0, 100, 5)**  **Y = np.arange(len(X))**  **plt.plot(X, Y, 'r')**  **plt.show()**  **plt.plot(X, X\*\*3, ‘g^’)**  **plt.title(‘Exponential’)**  **plt.show()**  **plt.plot(np.sin(X), ‘D-’)**  **plt.grid(True)**  **plt.show()** | **X = array ที่มีค่าเริ่มต้น 0 ถึง 100 โดยเพิ่มทีละ 5**  **Y = ค่ามิติของแต่ละตัวของ X**  **Plot เป็น เส้น**  **Plot เป็น จุด**  **Plot เป็นจุด + เป็น เส้นเชื่อมแต่ละจุด**  **และมีเส้น grid** |
| **6** | **N = 30**  **X = np.random.rand(N)**  **Y = np.random.rand(N)**  **plt.scatter(X, Y)**  **plt.show()**  **colors = np.random.rand(len(X))**  **area = (30 \* np.random.rand(len(X))) \*\* 2**  **plt.scatter(X, Y, s = area, c = colors)**  **plt.show()** | **X,Y = array เป็น random ตัวเลขจำนวน 30 ตัว**  **Plot เป็นจุด**  **Plot เป็นจุดที่มีขนาดต่างกันตาม area** |
| **7** | **X = np.linspace(0, 10, 100)**  **plt.figure(figsize=(10,10))**  **plt.subplot(3,1,1)**  **plt.title(‘Sin’)**  **plt.plot(np.sin(X))**  **plt.subplot(3,1,2)**  **plt.title(‘Cos’)**  **plt.plot(np.cos(X))**  **plt.subplot(3,1,3)**  **plt.title(‘Tan’)**  **plt.plot(np.tan(X))**  **plt.suptitle(‘Trigonometry’, fontsize=20)**  **plt.show()** | **X = สร้างตัวแปรที่มีจุดทศนิยมตั้งแต่ 0 ถึง 10 จำนวน 100 ตัว**  **สร้าง graph ขนาด 10\*10**  **แล้วสร้าง plot หลายอัน โดยกำหนดขนาดเป็น 3,1,1**  **เป็นกราฟของ sin**  **3,1,2 เป็น cos**  **3,1,3 เป็น tan**  **และสุดท้ายสร้างหัวข้อ ขนาด 20 ชื่อว่า Trigonometry** |
| **8** | **penguins = sns.load\_dataset(“penguins”)**  **sns.pairplot(penguins)** | **ใช้ lib seaborn แล้ว load data from penguins**  **Pairplot คือ การ plot หลายอันตาม feature ของ penguins**  **โดยจะมีกราฟของจุดและแท่ง** |
| **9** | **sns.pairplot(penguins, hue = “species”)** | **มีกราฟเป็นจุดหลายสีแบ่งตาก species แล้วมีกราฟการกระจายแบบละฉัง** |
| **10** | **sns.pairplot(penguins, hue = “species”, diag\_kind = “hist”)** | **เปลี่ยนจากกราฟกระจายเป็นกราฟแท่งแทน** |
| **11** | **sns.pairplot(penguins, kind = “kde”)** | **เป็นกราฟหลุดและกราฟกระจาย** |
| **12** | **sns.pairplot(penguins, kind = “hist”)** | **เป็นแท่งกับ ภาพpixcel** |
| **13** | **sns.pairplot(penguins, hue = “species”, markers = [“o”, “s”, “D”])** | **ใช้ lib seaborn แล้ว load data from penguins**  **Pairplot คือ การ plot หลายอันตาม feature ของ penguins**  **โดยจะมีกราฟของจุดและแท่ง แต่เปลี่ยนจุดตาม code** |
| **14** | **sns.pairplot(penguins, corner = True)** | **ตัดกราฟออกครึ่งหนึ่ง** |
| **15** | **sns.pairplot(penguins, plot\_kws = dict(marker = “+”, linewidth = 1), diag\_kws = dict(fill = False))** | **กราฟ เครื่องหมาย + และแท่ง** |
| **16** | **map = sns.pairplot(penguins, diag\_kind = “kde”)**  **map.map\_lower(sns.kdeplot, levels = 4, color = “.2”)** | **เป็นการรวมการ plot หลายอย่างอยู่ pairplot โดยในที่นี้มีเป็นหลุดกับจุดและแท่ง** |

Lab – Assignment 03 – 2 Following code

|  |  |
| --- | --- |
| No. | Code |
| 1 | import numpy as np  import matplotlib.pyplot as plt  import matplotlib  matplotlib.rcParams['font.sans-serif'] = "cmr10"  import random  import warnings  warnings.filterwarnings("ignore")  np.random.seed(54)  # random.seed(4959)  def dist\_function(x):      f\_x = np.sin(x)      return f\_x  def gen\_data(x):      f\_x = dist\_function(x)      b = 0.4      y = f\_x + np.random.uniform(-b, b, len(x))      sigma = 1/12\*(2\*b)\*\*2      return y, f\_x, sigma  def get\_rand\_data(len\_ratio, x, y, f\_x):      m = len(x)      m\_new = int(np.round(len\_ratio\*m))      ind = random.sample(range(m), m\_new)      ind = np.sort(ind)      x1 = x[ind]      y1 = y[ind]      f\_x1 = f\_x[ind]      return x1, y1, f\_x1  x\_orig = np.arange(0, 4 \* np.pi, .2)  y\_orig, f\_x\_orig, sigma = gen\_data(x\_orig)  p\_order = [1,3,5,7,9,11]  num\_data = 2000 |
| 2 | lpN = 2  hN = (int)(len(p\_order)/lpN)  f, ax = plt.subplots(lpN,hN, figsize=(15, 8), dpi=200, facecolor='w', edgecolor='k')  t = np.arange(0.1, 4\*np.pi, .2)  P=[]  bb=[]  vv=[]  for i, p\_val in enumerate(p\_order):      exp\_f\_x = np.zeros(len(t))      exp\_f\_x\_plot = np.zeros(len(x\_orig))      var = np.zeros(len(t))      for j in range(num\_data):          x, y, f\_x = get\_rand\_data(0.7, x\_orig, y\_orig, f\_x\_orig)          p = np.poly1d(np.polyfit(x, y, p\_val))          # ax[j, i].plot(x, y, 'o', t, p(x\_orig), '-')          P.append(p)          exp\_f\_x = exp\_f\_x + p(t)          exp\_f\_x\_plot = exp\_f\_x\_plot + p(x\_orig)      exp\_f\_x = exp\_f\_x/num\_data      exp\_f\_x\_plot = exp\_f\_x\_plot / num\_data      ax[(int)(i/hN)][i%hN].plot(x, y, 'o', x\_orig, exp\_f\_x\_plot, '-')      plt.setp(plt.gcf().get\_axes(), xticks=[], yticks=[])      ax[(int)(i/hN)][i%hN].set\_xlabel('P='+str(p\_order[i]))      bias = np.linalg.norm(exp\_f\_x-dist\_function(t))      bb.append(bias)      for j in range(num\_data):          p\_t = P.pop(0)          var = var + np.square(p\_t(t)-exp\_f\_x)      var = var/num\_data      variance = np.linalg.norm(var)      vv.append(variance)  print("bias: ", bb)  print("var: ",vv)  fig, ax1 = plt.subplots(dpi=200)  color = 'tab:orange'  ax1.set\_xlabel('Polynomial order')  ax1.set\_ylabel('Variance', color=color)  ax1.plot(p\_order, vv, color=color, label = 'Variance')  ax1.tick\_params(axis='y', labelcolor=color)  ax2 = ax1.twinx()  color = 'tab:blue'  ax2.set\_ylabel('Bias', color=color)  ax2.plot(p\_order, np.square(bb), color=color,  label = 'Bias')  ax2.tick\_params(axis='y', labelcolor=color)  fig.tight\_layout()  error = np.square(bb)+vv+sigma\*len(t)  min\_ind = np.argmin(error)  plt.plot(p\_order, error, 'k--', label = 'error')  plt.plot(p\_order[min\_ind], error[min\_ind], 'ro', label= 'Minimum Error')  leg = ax2.legend(loc='upper right')  leg = ax1.legend(loc='upper left')  plt.show() |