**Lab3**

201635825 오성원

Page.2

|  |
| --- |
| Source code |
| import numpy as np  import pandas as pd  import matplotlib.pyplot as plt  import seaborn as sns  from sklearn import preprocessing  df = pd.read\_csv('C:/python\_file/bmi\_data\_lab3\_1.csv')  print(df,"\n")  print("Feature names\m", df.columns, "\n")  print("Data type : ", type(df))  for j in range(0,5):  data=[]  for i in range(len(df)):  if(df['BMI'][i].item() == j):  if df['Height (Inches)'][i] < 65:  data.append(5)  elif 65 <= df['Height (Inches)'][i] < 66:  data.append(15)  elif 66 <= df['Height (Inches)'][i] < 67:  data.append(25)  elif 67 <= df['Height (Inches)'][i] < 68:  data.append(35)  elif 68 <= df['Height (Inches)'][i] < 69:  data.append(45)  elif 69 <= df['Height (Inches)'][i] < 70:  data.append(55)  elif 70 <= df['Height (Inches)'][i] < 71:  data.append(65)  elif 71 <= df['Height (Inches)'][i] < 72:  data.append(75)  elif 72 <= df['Height (Inches)'][i] < 73:  data.append(85)  elif 73 <= df['Height (Inches)'][i] :  data.append(95)  if data:  name = ['~65', '65~66', '66~67', '67~68','67~68','68~69','70~71','71~72','72~73','73~']  plt.hist(data, bins=[0,10,20,30,40,50,60,70,80,90,100], rwidth=0.7)  plt.xticks([5,15,25,35,45,55,65,75,85,95], name, fontsize=6)  if j==0:  plt.title('Height histogram-Bmi Extremely weak')  elif j==1:  plt.title('Height histogram-Bmi Weak')  elif j==2:  plt.title('Height histogram-Bmi Normal')  elif j==3:  plt.title('Height histogram-Bmi Overweight')  elif j==4:  plt.title('Height histogram-Bmi Obesity')  plt.xlabel("Height(Inches)")  plt.ylabel("Number of students")  plt.show()  #-----------------------------------------------------------------------------------------------------  for j in range(0,5):  data=[]  for i in range(len(df)):  if(df['BMI'][i].item() == j):  if df['Weight (Pounds)'][i] < 100:  data.append(5)  elif 100 <= df['Weight (Pounds)'][i] < 106:  data.append(15)  elif 106 <= df['Weight (Pounds)'][i] < 112:  data.append(25)  elif 112 <= df['Weight (Pounds)'][i] < 118:  data.append(35)  elif 118 <= df['Weight (Pounds)'][i] < 124:  data.append(45)  elif 124 <= df['Weight (Pounds)'][i] < 130:  data.append(55)  elif 130 <= df['Weight (Pounds)'][i] < 136:  data.append(65)  elif 136 <= df['Weight (Pounds)'][i] < 142:  data.append(75)  elif 142 <= df['Weight (Pounds)'][i] < 148:  data.append(85)  elif 148 <= df['Weight (Pounds)'][i]:  data.append(95)  if data:  name = ['~100', '100~106', '106~112', '112~118','118~124','124~130','130~136','136~142','142~148','148~']  plt.hist(data, bins=[0,10,20,30,40,50,60,70,80,90,100], rwidth=0.7)  plt.xticks([5,15,25,35,45,55,65,75,85,95], name, fontsize=6)  if j==0:  plt.title('Weight histogram-Bmi Extremely weak')  elif j==1:  plt.title('Weight histogram-Bmi Weak')  elif j==2:  plt.title('Weight histogram-Bmi Normal')  elif j==3:  plt.title('Weight histogram-Bmi Overweight')  elif j==4:  plt.title('Weight histogram-Bmi Obesity')  plt.xlabel("Weight(Pounds)")  plt.ylabel("Number of students")  plt.show() |

|  |
| --- |
| Screenshot |
|  |

Page2con’t)

|  |
| --- |
| Source code |
| import numpy as np  import pandas as pd  import matplotlib.pyplot as plt  import seaborn as sns  from sklearn import preprocessing  import sklearn.linear\_model as lm  df = pd.read\_csv('C:/python\_file/bmi\_data\_lab3\_1.csv')  #-----------------------------------------------------------------------------------------------------  df2 = pd.DataFrame(  {'Height (Inches)' : df['Height (Inches)'],  'Weight (Pounds)': df['Weight (Pounds)']  })  scaler = preprocessing.StandardScaler()  scaled\_df = scaler.fit\_transform(df2)  scaled\_df = pd.DataFrame(scaled\_df,columns=['Height (Inches)','Weight (Pounds)'])  fig, (ax1, ax2) = plt.subplots(ncols = 2, figsize =(6,5))  ax1.set\_title('Before Scaling')  sns.kdeplot(df ['Height (Inches)'], ax=ax1)  sns.kdeplot(df ['Weight (Pounds)'], ax=ax1)  ax2.set\_title('After Standard Scaler')  sns.kdeplot(scaled\_df ['Height (Inches)'], ax=ax2)  sns.kdeplot(scaled\_df ['Weight (Pounds)'], ax=ax2)  plt.show()  #-----------------------------------------------------------------------------------------------------  df2 = pd.DataFrame(  {'Height (Inches)' : df['Height (Inches)'],  'Weight (Pounds)': df['Weight (Pounds)']  })  scaler = preprocessing.MinMaxScaler()  scaled\_df = scaler.fit\_transform(df2)  scaled\_df = pd.DataFrame(scaled\_df,columns=['Height (Inches)','Weight (Pounds)'])  fig, (ax1, ax2) = plt.subplots(ncols = 2, figsize =(6,5))  ax1.set\_title('Before Scaling')  sns.kdeplot(df ['Height (Inches)'], ax=ax1)  sns.kdeplot(df ['Weight (Pounds)'], ax=ax1)  ax2.set\_title('After MinMax Scaler')  sns.kdeplot(scaled\_df ['Height (Inches)'], ax=ax2)  sns.kdeplot(scaled\_df ['Weight (Pounds)'], ax=ax2)  plt.show()  #-----------------------------------------------------------------------------------------------------  df2 = pd.DataFrame(  {'Height (Inches)' : df['Height (Inches)'],  'Weight (Pounds)': df['Weight (Pounds)']  })  scaler = preprocessing.RobustScaler()  scaled\_df = scaler.fit\_transform(df2)  scaled\_df = pd.DataFrame(scaled\_df,columns=['Height (Inches)','Weight (Pounds)'])  fig, (ax1, ax2) = plt.subplots(ncols = 2, figsize =(6,5))  ax1.set\_title('Before Scaling')  sns.kdeplot(df ['Height (Inches)'], ax=ax1)  sns.kdeplot(df ['Weight (Pounds)'], ax=ax1)  ax2.set\_title('After Robust Scaler')  sns.kdeplot(scaled\_df ['Height (Inches)'], ax=ax2)  sns.kdeplot(scaled\_df ['Weight (Pounds)'], ax=ax2)  plt.show() |

|  |
| --- |
| Screen shot |
|  |

Page3

|  |
| --- |
| Source code |
| import numpy as np  import pandas as pd  import matplotlib.pyplot as plt  import seaborn as sns  from sklearn import preprocessing  import sklearn.linear\_model as lm  import copy  missing\_values=[""]  df = pd.read\_csv('C:/python\_file/bmi\_data\_lab3.csv', na\_values=missing\_values)  print(df)  print("#of NaN for each row")  for i in range(len(df)):  df\_temp=df.loc[i]  print("Index",i, " of NaN: ",df\_temp.isna().sum())  print("#of NaN for each column")  print(df.isna().sum())  print("Extract all rows without NaN")  print(df.dropna(how='any'))  df\_temp = copy.deepcopy(df)  print("!-- Result of fillna mean --!")  df\_temp['Height (Inches)'] = pd.to\_numeric(df\_temp['Height (Inches)'])  df\_temp['Weight (Pounds)'] = pd.to\_numeric(df\_temp['Weight (Pounds)'])  df\_temp['Age'] = pd.to\_numeric(df\_temp['Age'])  df\_temp['BMI'] = pd.to\_numeric(df\_temp['BMI'])  mean=df\_temp.mean()  print(df\_temp.fillna(mean))  print("!-- Result of fillna median --!")  print(df.fillna(df.median()))  print("!-- Result of ffill --!")  print(df.fillna(df.fillna(axis=0, method='ffill')))  print("!-- Result of bfill --!")  print(df.fillna(df.fillna(axis=0, method='bfill'))) |

|  |
| --- |
| Screen shot |
| The result of converting dirty value to nan    #of NaN for each row(Omitted in the middle)    #of NaN for each column |

Page6

|  |
| --- |
| Source code |
| import numpy as np  import pandas as pd  import matplotlib.pyplot as plt  import seaborn as sns  from sklearn import preprocessing  import sklearn.linear\_model as lm  df = pd.read\_csv('C:/python\_file/bmi\_data\_lab3.csv')  df\_temp=df.copy()  df\_nonNa=df.dropna(how='any')  height\_temp=np.array(df\_temp ['Height (Inches)'])  weight\_temp=np.array(df\_temp ['Weight (Pounds)'])  height=np.array(df\_nonNa ['Height (Inches)'])  weight=np.array(df\_nonNa ['Weight (Pounds)'])  height\_na=[]  weight\_na=[]  for i in range(len(df)):  if np.isnan(height\_temp[i]) :  weight\_na.append(weight\_temp[i])  if np.isnan(weight\_temp[i]) :  height\_na.append(height\_temp[i])  arr\_height\_na=np.array(height\_na)  arr\_weight\_na=np.array(weight\_na)  E1 =lm.LinearRegression()  E1.fit(height[:, np.newaxis], weight)  E2 =lm.LinearRegression()  E2.fit(weight[:, np.newaxis], height)  pxh = np.array([height.min()-1, height.max()+1])  pyw = E1.predict(pxh[:, np.newaxis])  pxw = np.array([weight.min()-1, weight.max()+1])  pyh = E2.predict(pxw[:, np.newaxis])  x=E2.predict( arr\_weight\_na[ : ,np.newaxis])  y=E1.predict( arr\_height\_na[ : ,np.newaxis])  fx=np.append( arr\_height\_na, x)  fy=np.append( y, arr\_weight\_na)  plt.scatter(height, weight)  plt.scatter(fx, fy, color='r')  plt.plot(pxh, pyw, color='black')  plt.plot(pyh, pxw, color='black')  plt.xlabel('Height (Inches)')  plt.ylabel('Weight (Pounds)')  plt.show() |

|  |
| --- |
| Screen shot |
|  |

Page3(con’t)

|  |
| --- |
| Source Code |
| import numpy as np  import pandas as pd  import matplotlib.pyplot as plt  import seaborn as sns  from sklearn import preprocessing  import sklearn.linear\_model as lm  df = pd.read\_csv('C:/python\_file/bmi\_data\_lab3.csv')  df\_temp=df.copy()  df\_nonNa=df.dropna(how='any')  male\_na=df\_nonNa.loc[df\_nonNa['Sex']=='Male'].copy()  female\_na=df\_nonNa.loc[df\_nonNa['Sex']=='Female'].copy()  male = df.loc[df['Sex']=='Male'].copy()  height\_temp=np.array(male['Height (Inches)'])  weight\_temp=np.array(male['Weight (Pounds)'])  height=np.array(male\_na['Height (Inches)'])  weight=np.array(male\_na['Weight (Pounds)'])  height\_na=[]  weight\_na=[]  for i in range(len(male)):  if np.isnan(height\_temp[i]) :  weight\_na.append(weight\_temp[i])  if np.isnan(weight\_temp[i]) :  height\_na.append(height\_temp[i])  arr\_height\_na=np.array(height\_na)  arr\_weight\_na=np.array(weight\_na)  E1=lm.LinearRegression()  E1.fit(height[:, np.newaxis], weight)  E2 =lm.LinearRegression()  E2.fit(weight[:, np.newaxis], height)  pxh = np.array([height.min()-1, height.max()+1])  pyw = E1.predict(pxh[:, np.newaxis])  pxw = np.array([weight.min()-1, weight.max()+1])  pyh = E2.predict(pxw[:, np.newaxis])  x=E2.predict( arr\_weight\_na[ : ,np.newaxis])  y=E1.predict( arr\_height\_na[ : ,np.newaxis])  fx=np.append( arr\_height\_na, x)  fy=np.append( y, arr\_weight\_na)  plt.scatter(height, weight)  plt.scatter(fx, fy, color='r')  plt.plot(pxh, pyw, color='black')  plt.plot(pyh, pxw, color='black')  plt.title('Male')  plt.xlabel('Height (Inches)')  plt.ylabel('Weight (Pounds)')  plt.show()  #-----------------------------------------------------------------  female = df.loc[df['Sex']=='Female'].copy()  height\_temp=np.array(female['Height (Inches)'])  weight\_temp=np.array(female['Weight (Pounds)'])  height=np.array(female\_na['Height (Inches)'])  weight=np.array(female\_na['Weight (Pounds)'])  height\_na=[]  weight\_na=[]  for i in range(len(female)):  if np.isnan(height\_temp[i]) :  weight\_na.append(weight\_temp[i])  if np.isnan(weight\_temp[i]) :  height\_na.append(height\_temp[i])  arr\_height\_na=np.array(height\_na)  arr\_weight\_na=np.array(weight\_na)  E1=lm.LinearRegression()  E1.fit(height[:, np.newaxis], weight)  E2 =lm.LinearRegression()  E2.fit(weight[:, np.newaxis], height)  pxh = np.array([height.min()-1, height.max()+1])  pyw = E1.predict(pxh[:, np.newaxis])  pxw = np.array([weight.min()-1, weight.max()+1])  pyh = E2.predict(pxw[:, np.newaxis])  x=E2.predict( arr\_weight\_na[ : ,np.newaxis])  y=E1.predict( arr\_height\_na[ : ,np.newaxis])  fx=np.append( arr\_height\_na, x)  fy=np.append( y, arr\_weight\_na)  plt.scatter(height, weight)  plt.scatter(fx, fy, color='r')  plt.plot(pxh, pyw, color='black')  plt.plot(pyh, pxw, color='black')  plt.title('Female')  plt.xlabel('Height (Inches)')  plt.ylabel('Weight (Pounds)')  plt.show() |

|  |
| --- |
| Screen shot |
|  |