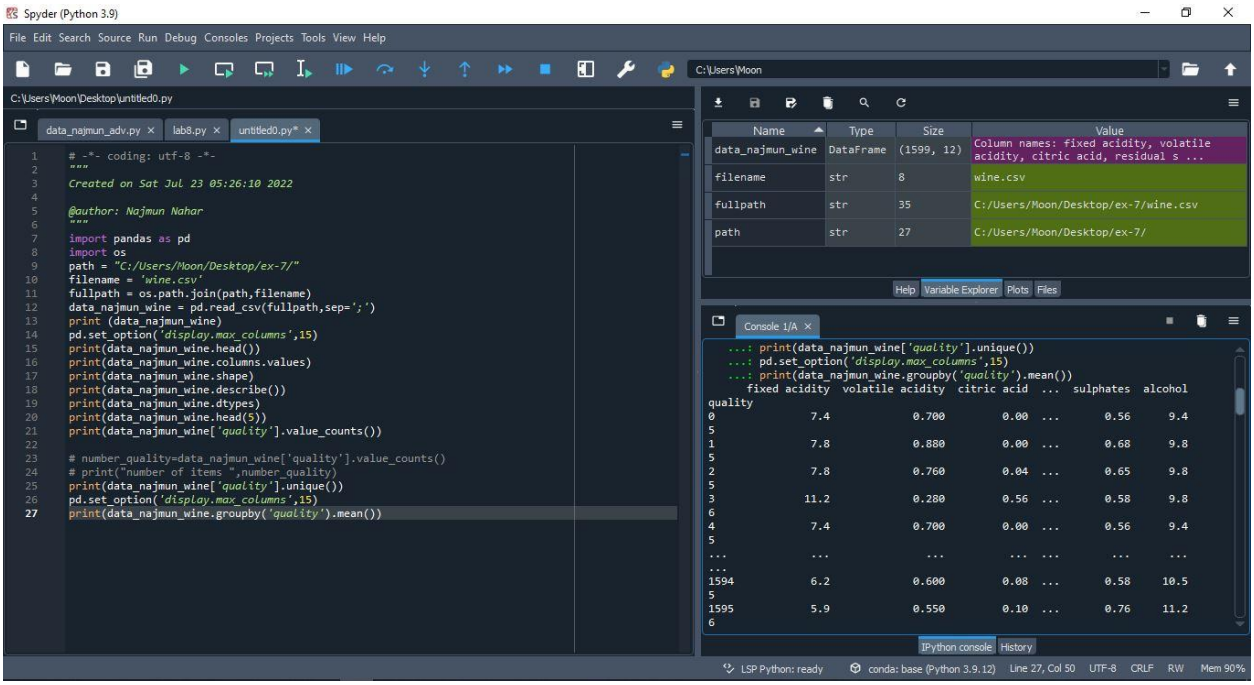
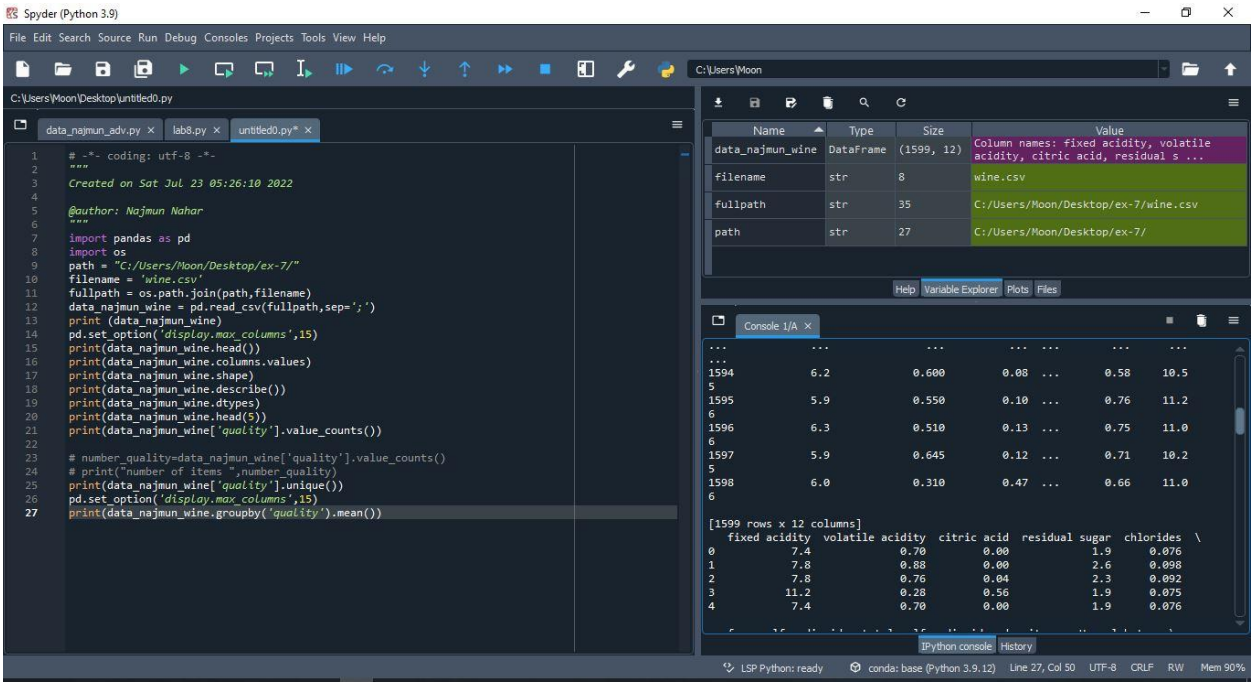
Name: Najmun Nahar

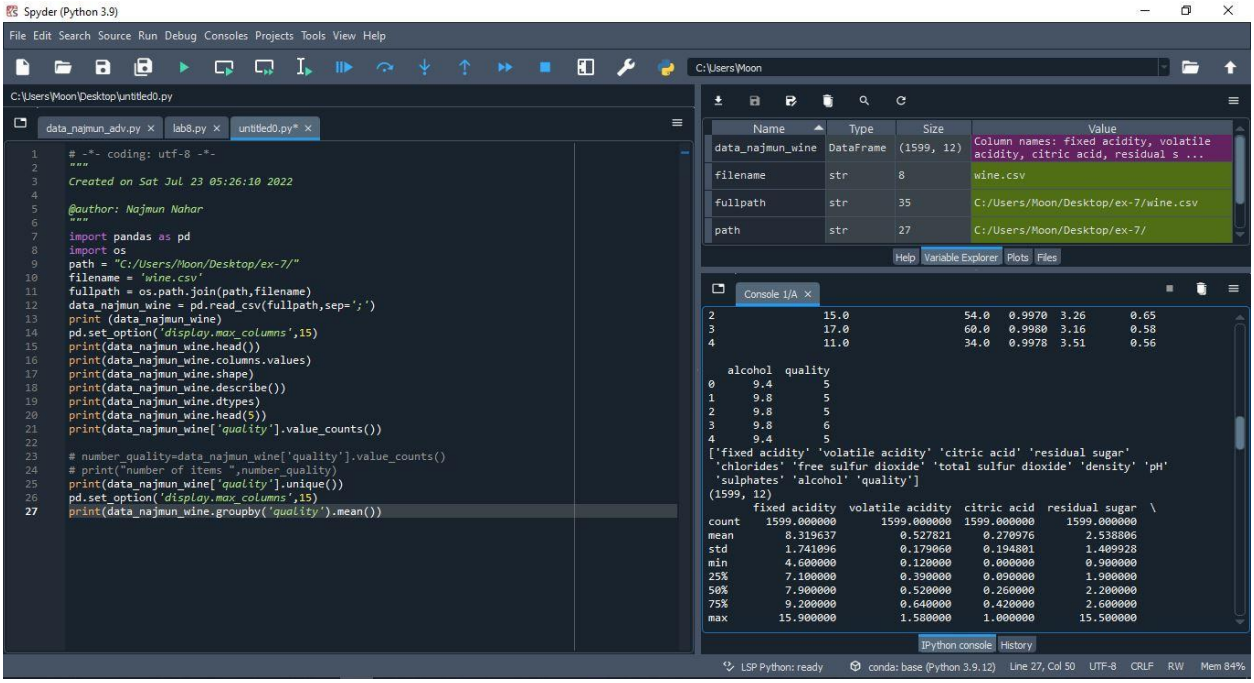
ID: 301160081

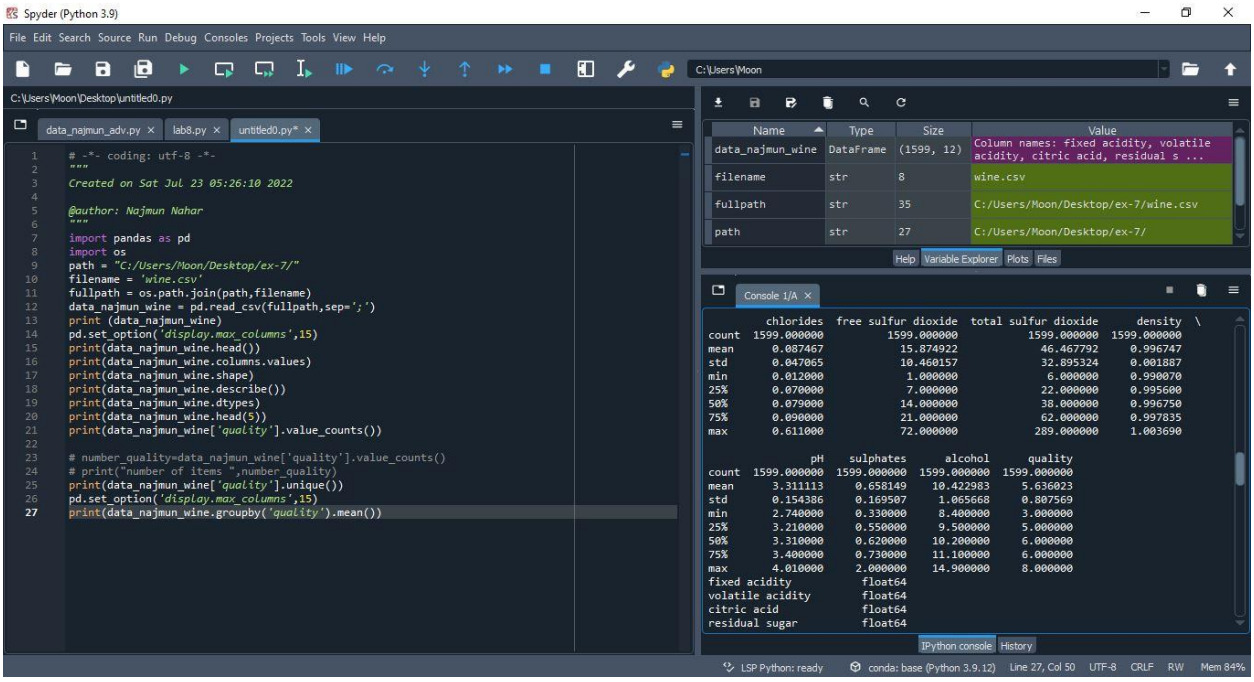
Lab-9

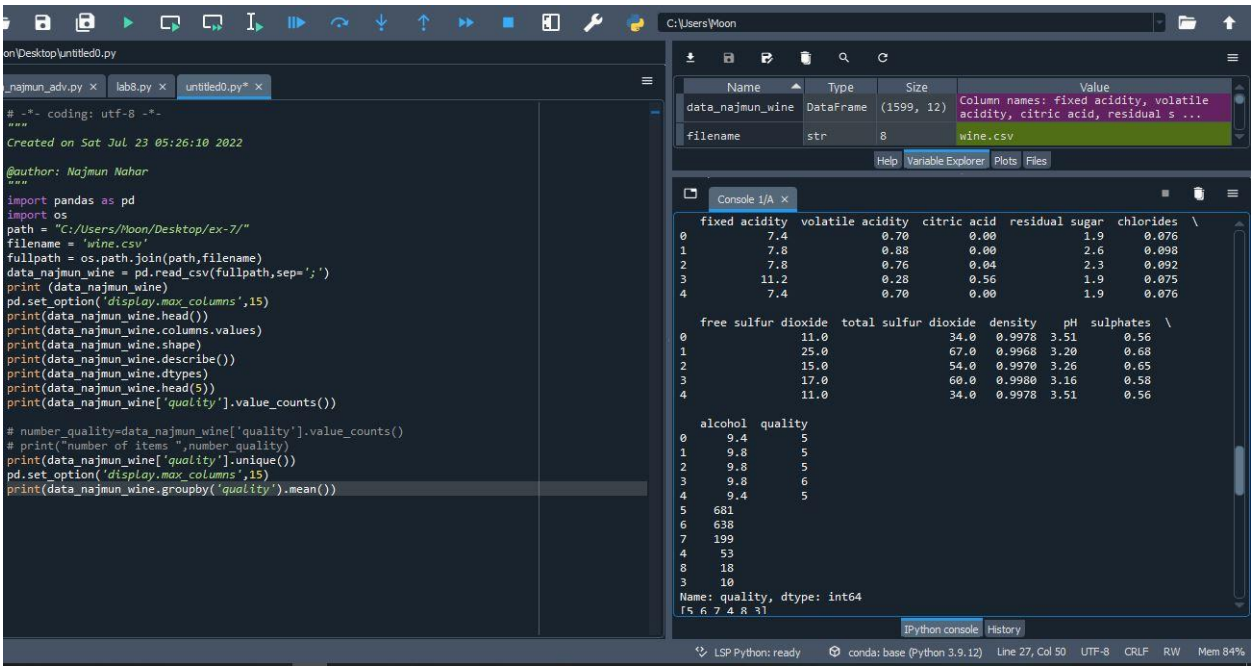
1.





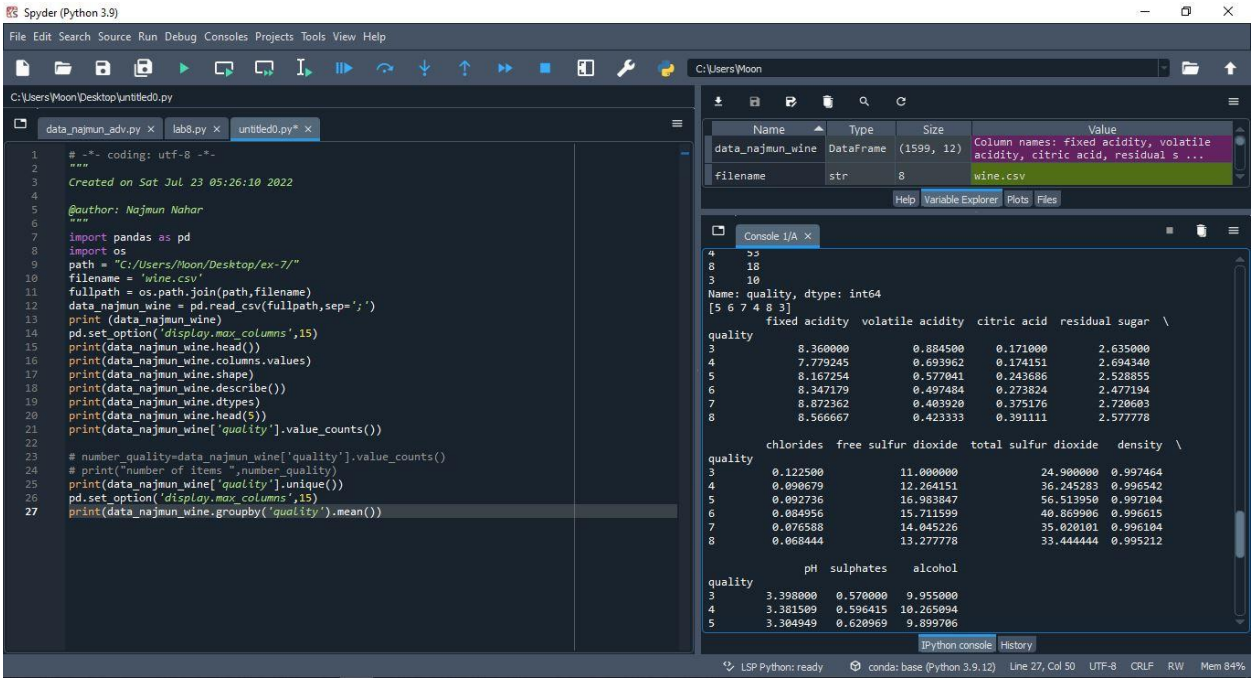

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon
C:\Users\Moon\Desktop\untitled0.py
data_najmun_adv.py
lab8.py
untitled0.py*
-*- coding: utf-8 -*-
"""
Created on Sat Jul 23 05:26:10 2022
@author: Najmun Nahar
"""
import pandas as pd
import os
path = "C:/Users/Moon/Desktop/ex-7/"
filename = 'wine.csv'
fullpath = os.path.join(path,filename)
data_najmun_wine = pd.read_csv(fullpath,sep=';')
print (data_najmun_wine)
pd.set_option('display.max_columns',15)
print(data_najmun_wine.head())
print(data_najmun_wine.columns.values)
print(data_najmun_wine.shape)
print(data_najmun_wine.describe())
print(data_najmun_wine.dtypes)
print(data_najmun_wine.head(5))
print(data_najmun_wine['quality'].value_counts())
# number_quality=data_najmun_wine['quality'].value_counts()
# print("number of items ",number_quality)
print(data_najmun_wine['quality'].unique())
pd.set_option('display.max_columns',15)
print(data_najmun_wine.groupby('quality').mean())
Name Type Size Value
data_najmun_wine DataFrame (1599, 12) Column names: fixed acidity, volatile acidity, citric acid, residual s ...
filename str 8 wine.csv
fullpath str 35 C:/Users/Moon/Desktop/ex-7/wine.csv
path str 27 C:/Users/Moon/Desktop/ex-7/
Help Variable Explorer Plots Files
Console 1/A
2 15.0 54.0 0.9970 3.26 0.65
3 17.0 60.0 0.9980 3.16 0.58
4 11.0 34.0 0.9978 3.51 0.56
alcohol quality
0 9.4 5
1 9.8 5
2 9.8 5
3 9.8 6
4 9.4 5
['fixed acidity' 'volatile acidity' 'citric acid' 'residual sugar'
'chlorides' 'free sulfur dioxide' 'total sulfur dioxide' 'density' 'pH'
'sulphates' 'alcohol' 'quality']
(1599, 12)
fixed acidity volatile acidity citric acid residual sugar \
count 1599.000000 1599.000000 1599.000000 1599.000000
mean 8.319637 0.527821 0.270976 2.538806
std 1.741096 0.179060 0.194801 1.409928
min 4.600000 0.120000 0.000000 0.900000
25% 7.100000 0.390000 0.090000 1.900000
50% 7.900000 0.520000 0.260000 2.200000
75% 9.200000 0.640000 0.420000 2.600000
max 15.900000 1.580000 1.000000 15.500000
IPython console History
LSP Python: ready
conda: base (Python 3.9.12)
Line 27, Col 50
UTF-8
CRLF
RW
Mem 84%


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon
C:\Users\Moon\Desktop\untitled0.py
data_najmun_adv.py
lab8.py
untitled0.py*
-*- coding: utf-8 -*-
"""
Created on Sat Jul 23 05:26:10 2022
@author: Najmun Nahar
"""
import pandas as pd
import os
path = "C:/Users/Moon/Desktop/ex-7/"
filename = 'wine.csv'
fullpath = os.path.join(path,filename)
data_najmun_wine = pd.read_csv(fullpath,sep=';')
print (data_najmun_wine)
pd.set_option('display.max_columns',15)
print(data_najmun_wine.head())
print(data_najmun_wine.columns.values)
print(data_najmun_wine.shape)
print(data_najmun_wine.describe())
print(data_najmun_wine.dtypes)
print(data_najmun_wine.head(5))
print(data_najmun_wine['quality'].value_counts())
# number_quality=data_najmun_wine['quality'].value_counts()
# print("number of items ",number_quality)
print(data_najmun_wine['quality'].unique())
pd.set_option('display.max_columns',15)
print(data_najmun_wine.groupby('quality').mean())
Name Type Size Value
data_najmun_wine DataFrame (1599, 12) Column names: fixed acidity, volatile acidity, citric acid, residual s ...
filename str 8 wine.csv
fullpath str 35 C:/Users/Moon/Desktop/ex-7/wine.csv
path str 27 C:/Users/Moon/Desktop/ex-7/
Help Variable Explorer Plots Files
Console 1/A
chlorides free sulfur dioxide total sulfur dioxide density \
count 1599.000000 1599.000000 1599.000000 1599.000000
mean 0.087467 15.874922 46.467792 0.996747
std 0.047065 10.460157 32.895324 0.001887
min 0.012000 1.000000 6.000000 0.990070
25% 0.070000 7.000000 22.000000 0.995600
50% 0.079000 14.000000 38.000000 0.996750
75% 0.090000 21.000000 62.000000 0.997835
max 0.611000 72.000000 289.000000 1.003690
pH sulphates alcohol quality
count 1599.000000 1599.000000 1599.000000 1599.000000
mean 3.311113 0.658149 10.422983 5.636023
std 0.154386 0.169507 1.065668 0.807569
min 2.740000 0.330000 8.400000 3.000000
25% 3.210000 0.550000 9.500000 5.000000
50% 3.310000 0.620000 10.200000 6.000000
75% 3.400000 0.730000 11.100000 6.000000
max 4.010000 2.000000 14.900000 8.000000
fixed acidity float64
volatile acidity float64
citric acid float64
residual sugar float64
IPython console History
LSP Python: ready
conda: base (Python 3.9.12)
Line 27, Col 50
UTF-8
CRLF
RW
Mem 84%



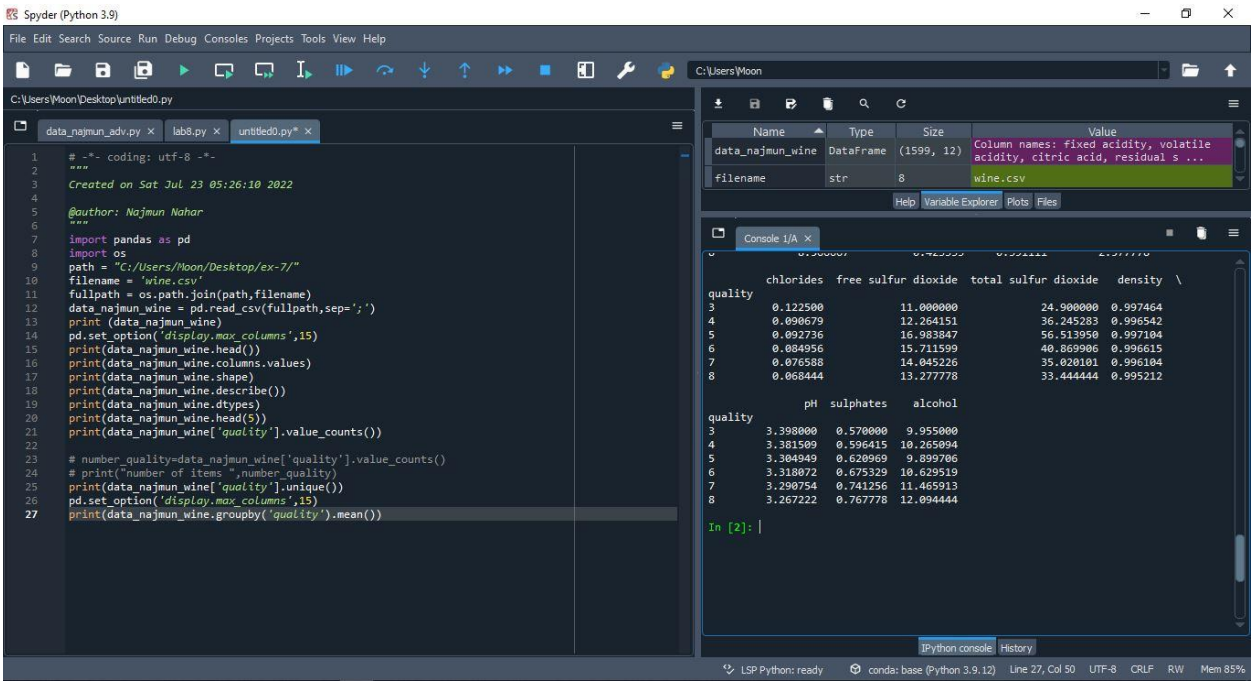
```
# -*- coding: utf-8 -*-
"""
Created on Sat Jul 23 05:26:10 2022

@author: Najmun Nahar
"""
import pandas as pd
import os
path = "C:/Users/Moon/Desktop/ex-7/"
filename = 'wine.csv'
fullpath = os.path.join(path,filename)
data_najmun_wine = pd.read_csv(fullpath,sep=';')
print (data_najmun_wine)
pd.set_option('display.max_columns',15)
print(data_najmun_wine.head())
print(data_najmun_wine.columns.values)
print(data_najmun_wine.shape)
print(data_najmun_wine.describe())
print(data_najmun_wine.dtypes)
print(data_najmun_wine.head(5))
print(data_najmun_wine['quality'].value_counts())

# number_quality=data_najmun_wine['quality'].value_counts()
# print("number of items ",number_quality)
print(data_najmun_wine['quality'].unique())
pd.set_option('display.max_columns',15)
print(data_najmun_wine.groupby('quality').mean())
```

```
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            7.4              0.70         0.00             1.9      0.076
1            7.8              0.88         0.00             2.6      0.098
2            7.8              0.76         0.04             2.3      0.092
3           11.2              0.28         0.56             1.9      0.075
4            7.4              0.70         0.00             1.9      0.076

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 11.0                  34.0   0.9978  3.51       0.56
1                 25.0                  67.0   0.9968  3.20       0.68
2                 15.0                  54.0   0.9970  3.26       0.65
3                 17.0                  60.0   0.9980  3.16       0.58
4                 11.0                  34.0   0.9978  3.51       0.56

   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5
5    681
6    638
7    199
4     53
8     18
3     10
Name: quality, dtype: int64
[5 6 7 4 8 3]
```

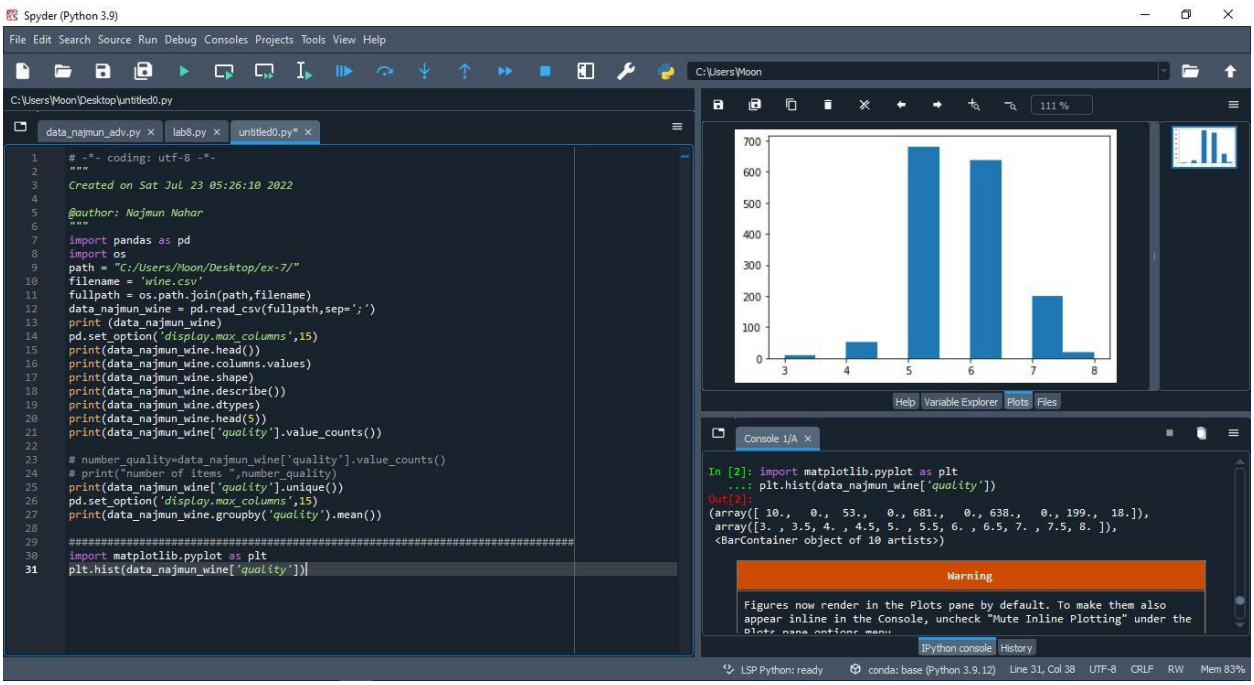


```
# -*- coding: utf-8 -*-
"""
Created on Sat Jul 23 05:26:10 2022

@author: Najmun Nahar
"""
import pandas as pd
import os
path = "C:/Users/Moon/Desktop/ex-7/"
filename = 'wine.csv'
fullpath = os.path.join(path,filename)
data_najmun_wine = pd.read_csv(fullpath,sep=';')
print (data_najmun_wine)
pd.set_option('display.max_columns',15)
print(data_najmun_wine.head())
print(data_najmun_wine.columns.values)
print(data_najmun_wine.shape)
print(data_najmun_wine.describe())
print(data_najmun_wine.dtypes)
print(data_najmun_wine.head(5))
print(data_najmun_wine['quality'].value_counts())

# number_quality=data_najmun_wine['quality'].value_counts()
# print("number of items ",number_quality)
print(data_najmun_wine['quality'].unique())
pd.set_option('display.max_columns',15)
print(data_najmun_wine.groupby('quality').mean())
```

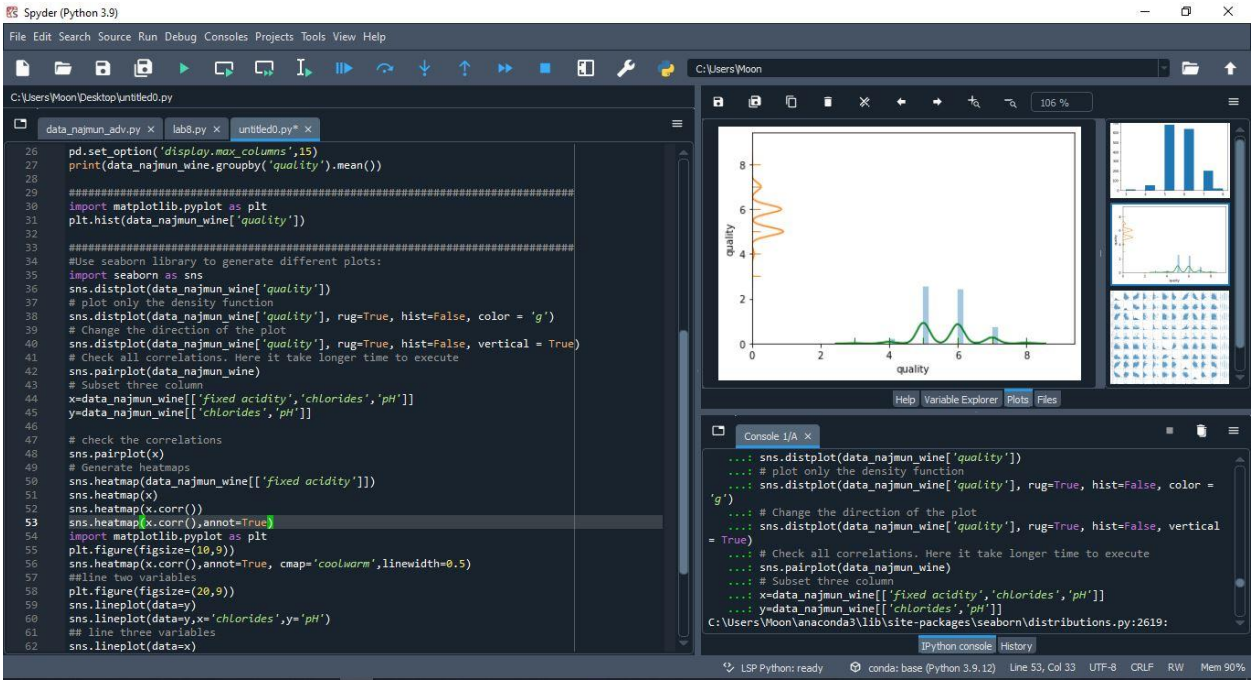
```
4     53
8     18
3     10
Name: quality, dtype: int64
[5 6 7 4 8 3]
         fixed acidity  volatile acidity  citric acid  residual sugar  \
quality
3             8.360000          0.884500     0.171000        2.635000
4             7.779245          0.693962     0.174151        2.694340
5             8.167254          0.577041     0.243686        2.528855
6             8.347179          0.497484     0.273824        2.477194
7             8.872362          0.403920     0.375176        2.720603
8             8.566667          0.423333     0.391111        2.577778

         chlorides  free sulfur dioxide  total sulfur dioxide   density  \
quality
3         0.122500            11.000000             24.900000  0.997464
4         0.090679            12.264151             36.245283  0.996542
5         0.092736            16.983847             56.513950  0.997104
6         0.084956            15.711599             40.869906  0.996615
7         0.076588            14.045226             35.020101  0.996104
8         0.068444            13.277778             33.444444  0.995212

                pH  sulphates    alcohol
quality
3        3.398000   0.570000   9.955000
4        3.381509   0.596415  10.265094
5        3.304949   0.620969   9.899706
```


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon\Desktop\untitled0.py
data_najmun_adv.py
lab8.py
untitled0.py*
-*- coding: utf-8 -*-
"""
Created on Sat Jul 23 05:26:10 2022
@author: Najmun Nahar
"""
import pandas as pd
import os
path = "C:/Users/Moon/Desktop/ex-7/"
filename = 'wine.csv'
fullpath = os.path.join(path,filename)
data_najmun_wine = pd.read_csv(fullpath,sep=';')
print (data_najmun_wine)
pd.set_option('display.max_columns',15)
print(data_najmun_wine.head())
print(data_najmun_wine.columns.values)
print(data_najmun_wine.shape)
print(data_najmun_wine.describe())
print(data_najmun_wine.dtypes)
print(data_najmun_wine.head(5))
print(data_najmun_wine['quality'].value_counts())
# number_quality=data_najmun_wine['quality'].value_counts()
# print("number of items ",number_quality)
print(data_najmun_wine['quality'].unique())
pd.set_option('display.max_columns',15)
print(data_najmun_wine.groupby('quality').mean())
Name
Type
Size
Value
data_najmun_wine
DataFrame
(1599, 12)
Column names: fixed acidity, volatile acidity, citric acid, residual s ...
filename
str
8
wine.csv
Help
Variable Explorer
Plots
Files
Console 1/A
chlorides free sulfur dioxide total sulfur dioxide density \
quality
3 0.122500 11.000000 24.900000 0.997464
4 0.090679 12.264151 36.245283 0.996542
5 0.092736 16.983847 56.513950 0.997104
6 0.084956 15.711599 40.869906 0.996615
7 0.076588 14.045226 35.020101 0.996104
8 0.068444 13.277778 33.444444 0.995212
pH sulphates alcohol
quality
3 3.398000 0.570000 9.955000
4 3.381509 0.596415 10.265094
5 3.304949 0.620969 9.899706
6 3.318072 0.675329 10.629519
7 3.290754 0.741256 11.465913
8 3.267222 0.767778 12.094444
In [2]:
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 27, Col 50
UTF-8
CRLF
RW
Mem 85%

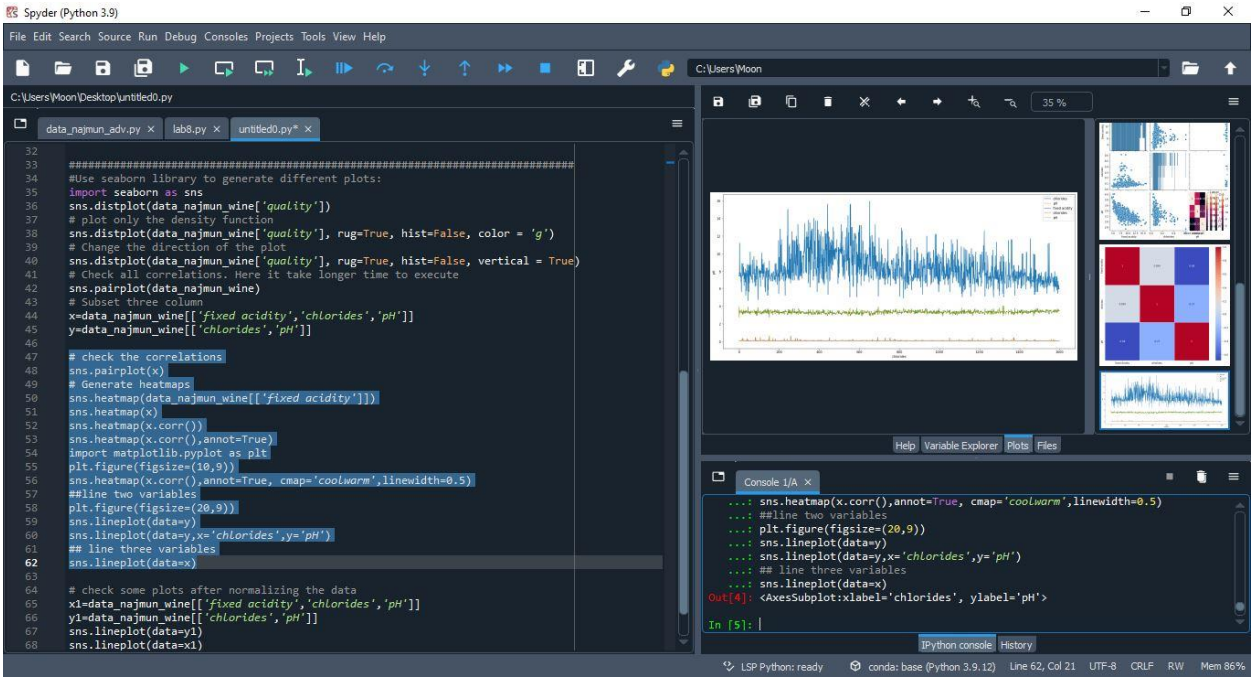
2.

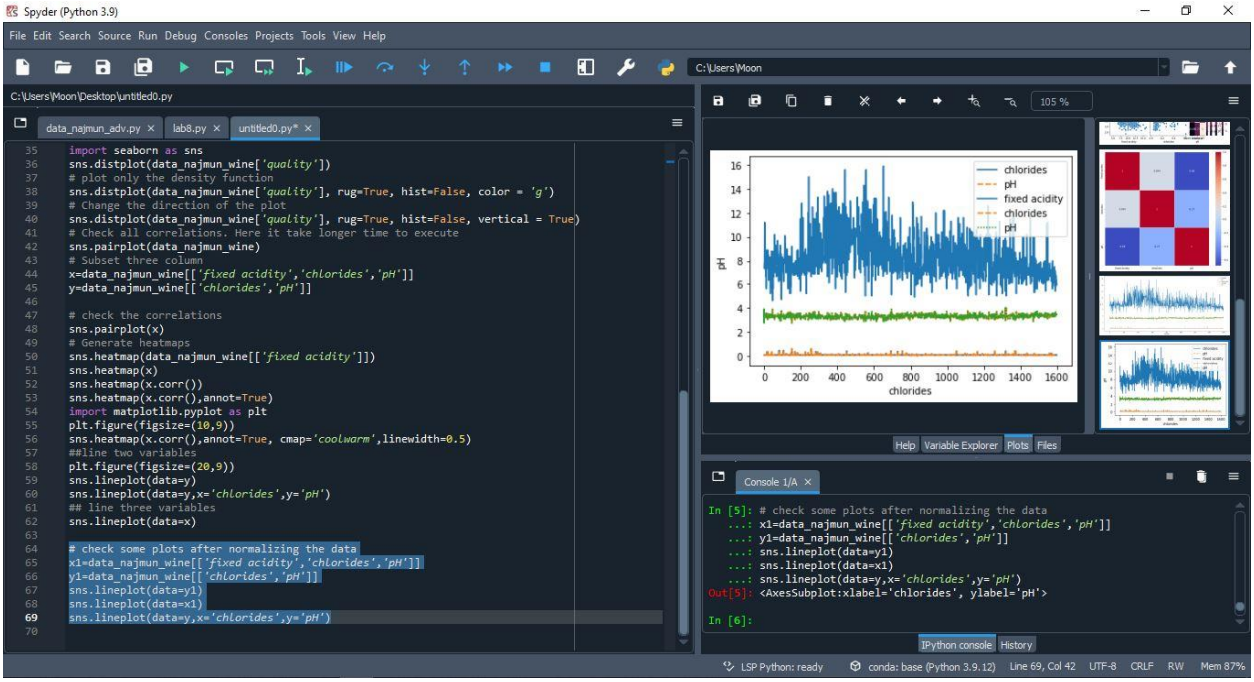

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon\Desktop\untitled0.py
data_najmun_adv.py
lab8.py
untitled0.py*
-*- coding: utf-8 -*-
"""
Created on Sat Jul 23 05:26:10 2022
@author: Najmun Nahar
"""
import pandas as pd
import os
path = "C:/Users/Moon/Desktop/ex-7/"
filename = 'wine.csv'
fullpath = os.path.join(path,filename)
data_najmun_wine = pd.read_csv(fullpath,sep=';')
print (data_najmun_wine)
pd.set_option('display.max_columns',15)
print(data_najmun_wine.head())
print(data_najmun_wine.columns.values)
print(data_najmun_wine.shape)
print(data_najmun_wine.describe())
print(data_najmun_wine.dtypes)
print(data_najmun_wine.head(5))
print(data_najmun_wine['quality'].value_counts())
# number_quality=data_najmun_wine['quality'].value_counts()
# print("number of items ",number_quality)
print(data_najmun_wine['quality'].unique())
pd.set_option('display.max_columns',15)
print(data_najmun_wine.groupby('quality').mean())
import matplotlib.pyplot as plt
plt.hist(data_najmun_wine['quality'])
700
600
500
400
300
200
100
0
3
4
5
6
7
8
Help
Variable Explorer
Plots
Files
Console 1/A
In [2]: import matplotlib.pyplot as plt
...: plt.hist(data_najmun_wine['quality'])
Out[2]:
(array([10., 0., 53., 0., 681., 0., 638., 0., 199., 18.]),
array([3. , 3.5, 4. , 4.5, 5. , 5.5, 6. , 6.5, 7. , 7.5, 8.]),
<BarContainer object of 10 artists>)
Warning
Figures now render in the Plots pane by default. To make them also appear inline in the Console, uncheck "Mute Inline Plotting" under the
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 31, Col 38
UTF-8
CRLF
RW
Mem 83%

3.

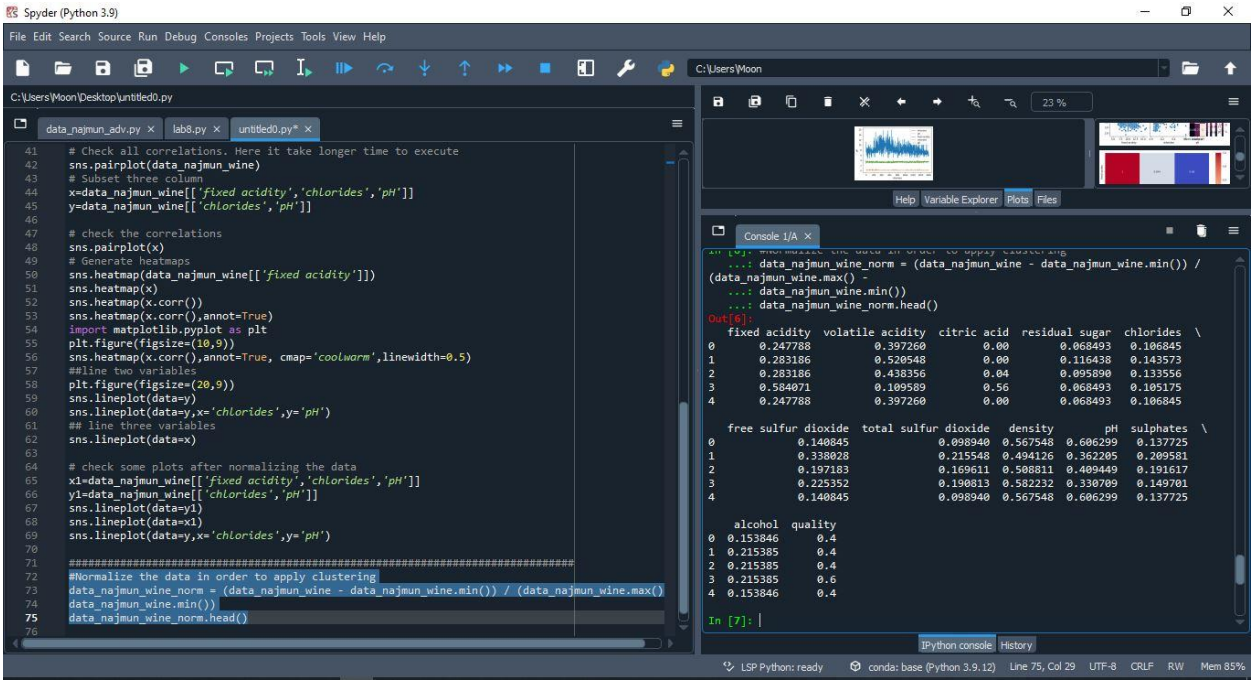

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon
C:\Users\Moon\Desktop\untitled0.py
data_najmun_adv.py
lab8.py
untitled0.py*
pd.set_option('display.max_columns',15)
print(data_najmun_wine.groupby('quality').mean())
import matplotlib.pyplot as plt
plt.hist(data_najmun_wine['quality'])
#Use seaborn library to generate different plots:
import seaborn as sns
sns.distplot(data_najmun_wine['quality'])
# plot only the density function
sns.distplot(data_najmun_wine['quality'], rug=True, hist=False, color = 'g')
# Change the direction of the plot
sns.distplot(data_najmun_wine['quality'], rug=True, hist=False, vertical = True)
# Check all correlations. Here it take longer time to execute
sns.pairplot(data_najmun_wine)
# Subset three column
x=data_najmun_wine[['fixed acidity','chlorides','pH']]
y=data_najmun_wine[['chlorides','pH']]
# check the correlations
sns.pairplot(x)
# Generate heatmaps
sns.heatmap(data_najmun_wine[['fixed acidity']])
sns.heatmap(x)
sns.heatmap(x.corr())
sns.heatmap(x.corr(),annot=True)
import matplotlib.pyplot as plt
plt.figure(figsize=(10,9))
sns.heatmap(x.corr(),annot=True, cmap='coolwarm',linewidth=0.5)
##line two variables
plt.figure(figsize=(20,9))
sns.lineplot(data=y)
sns.lineplot(data=y,x='chlorides',y='pH')
line three variables
sns.lineplot(data=x)
Console 1/A
...: sns.distplot(data_najmun_wine['quality'])
...: # plot only the density function
...: sns.distplot(data_najmun_wine['quality'], rug=True, hist=False, color = 'g')
...: # Change the direction of the plot
...: sns.distplot(data_najmun_wine['quality'], rug=True, hist=False, vertical = True)
...: # Check all correlations. Here it take longer time to execute
...: sns.pairplot(data_najmun_wine)
...: # Subset three column
...: x=data_najmun_wine[['fixed acidity','chlorides','pH']]
...: y=data_najmun_wine[['chlorides','pH']]
C:\Users\Moon\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
Help
Variable Explorer
Plots
Files
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 53, Col 33
UTF-8
CRLF
RW
Mem 90%

4.

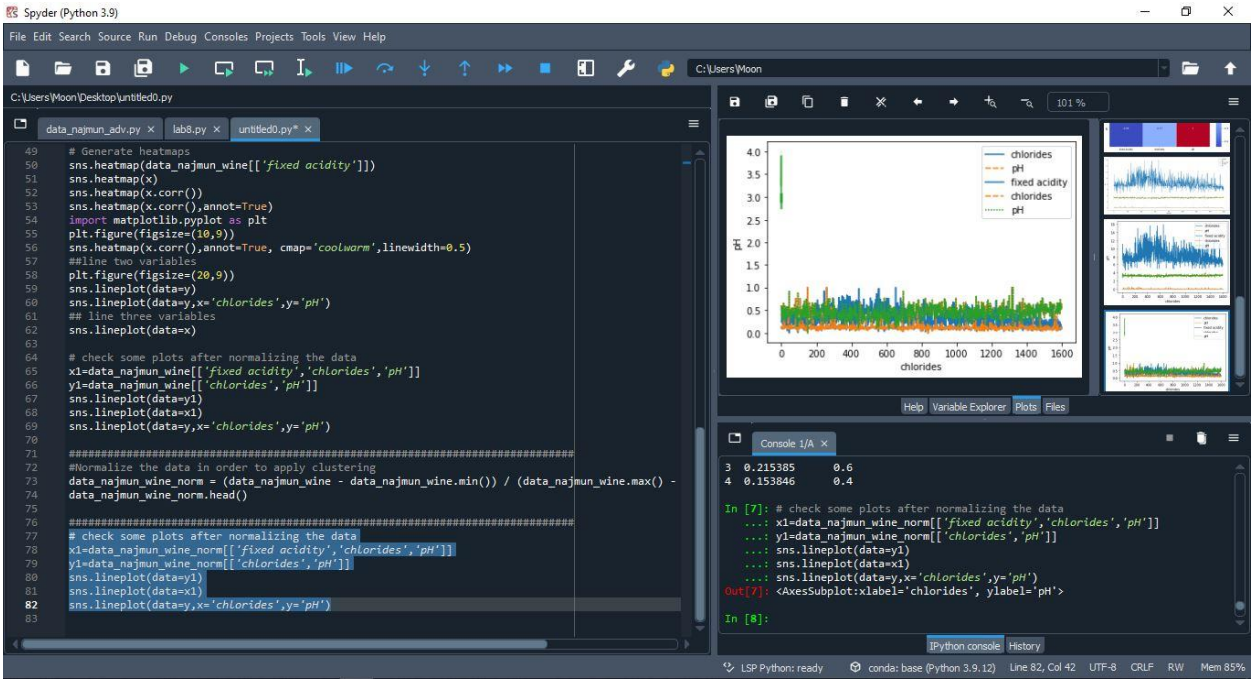

Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon
C:\Users\Moon\Desktop\untitled0.py
data_najmun_adv.py
lab8.py
untitled0.py*
#Use seaborn library to generate different plots:
import seaborn as sns
sns.distplot(data_najmun_wine['quality'])
# plot only the density function
sns.distplot(data_najmun_wine['quality'], rug=True, hist=False, color = 'g')
# Change the direction of the plot
sns.distplot(data_najmun_wine['quality'], rug=True, hist=False, vertical = True)
# Check all correlations. Here it take longer time to execute
sns.pairplot(data_najmun_wine)
# Subset three column
x=data_najmun_wine[['fixed acidity','chlorides','pH']]
y=data_najmun_wine[['chlorides','pH']]
# check the correlations
sns.pairplot(x)
# Generate heatmaps
sns.heatmap(data_najmun_wine[['fixed acidity']])
sns.heatmap(x)
sns.heatmap(x.corr())
sns.heatmap(x.corr(),annot=True)
import matplotlib.pyplot as plt
plt.figure(figsize=(10,9))
sns.heatmap(x.corr(),annot=True, cmap='coolwarm',linewidth=0.5)
##line two variables
plt.figure(figsize=(20,9))
sns.lineplot(data=y)
sns.lineplot(data=y,x='chlorides',y='pH')
line three variables
sns.lineplot(data=x)
# check some plots after normalizing the data
x1=data_najmun_wine[['fixed acidity','chlorides','pH']]
y1=data_najmun_wine[['chlorides','pH']]
sns.lineplot(data=y1)
sns.lineplot(data=x1)
Console 1/A
...: sns.heatmap(x.corr(),annot=True, cmap='coolwarm',linewidth=0.5)
...: ##line two variables
...: plt.figure(figsize=(20,9))
...: sns.lineplot(data=y)
...: sns.lineplot(data=y,x='chlorides',y='pH')
...: ## line three variables
...: sns.lineplot(data=x)
Out[4]: <AxesSubplot:xlabel='chlorides', ylabel='pH'>
In [5]:
Help
Variable Explorer
Plots
Files
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 62, Col 21
UTF-8
CRLF
RW
Mem 86%


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon\Desktop\untitled0.py
data_najmun_adv.py
lab8.py
untitled0.py*
import seaborn as sns
sns.distplot(data_najmun_wine['quality'])
# plot only the density function
sns.distplot(data_najmun_wine['quality'], rug=True, hist=False, color = 'g')
# Change the direction of the plot
sns.distplot(data_najmun_wine['quality'], rug=True, hist=False, vertical = True)
# Check all correlations. Here it take longer time to execute
sns.pairplot(data_najmun_wine)
# Subset three column
x=data_najmun_wine[['fixed acidity','chlorides','pH']]
y=data_najmun_wine[['chlorides','pH']]
# check the correlations
sns.pairplot(x)
# Generate heatmaps
sns.heatmap(data_najmun_wine[['fixed acidity']])
sns.heatmap(x)
sns.heatmap(x.corr())
sns.heatmap(x.corr(),annot=True)
import matplotlib.pyplot as plt
plt.figure(figsize=(10,9))
sns.heatmap(x.corr(),annot=True, cmap='coolwarm',linewidth=0.5)
##line two variables
plt.figure(figsize=(20,9))
sns.lineplot(data=y)
sns.lineplot(data=y,x='chlorides',y='pH')
line three variables
sns.lineplot(data=x)
# check some plots after normalizing the data
x1=data_najmun_wine[['fixed acidity','chlorides','pH']]
y1=data_najmun_wine[['chlorides','pH']]
sns.lineplot(data=y1)
sns.lineplot(data=x1)
sns.lineplot(data=y,x='chlorides',y='pH')
chlorides
pH
fixed acidity
Console 1/A
In [5]: # check some plots after normalizing the data
...: x1=data_najmun_wine[['fixed acidity','chlorides','pH']]
...: y1=data_najmun_wine[['chlorides','pH']]
...: sns.lineplot(data=y1)
...: sns.lineplot(data=x1)
...: sns.lineplot(data=y,x='chlorides',y='pH')
Out[5]: <AxesSubplot:xlabel='chlorides', ylabel='pH'>
In [6]:
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 69, Col 42
UTF-8
CRLF
RW
Mem 87%

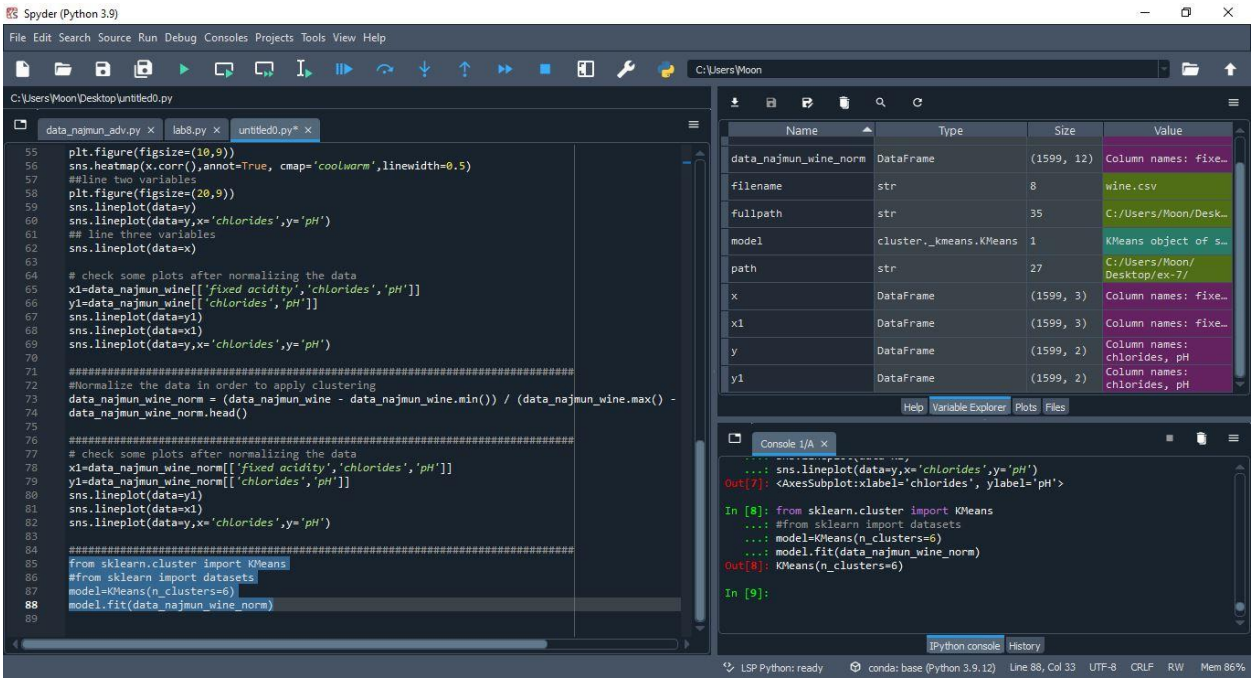
5.


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon\Desktop\untitled0.py
data_najmun_adv.py
lab8.py
untitled0.py*
# Check all correlations. Here it take longer time to execute
sns.pairplot(data_najmun_wine)
# Subset three column
x=data_najmun_wine[['fixed acidity','chlorides','pH']]
y=data_najmun_wine[['chlorides','pH']]
# check the correlations
sns.pairplot(x)
# Generate heatmaps
sns.heatmap(data_najmun_wine[['fixed acidity']])
sns.heatmap(x)
sns.heatmap(x.corr())
sns.heatmap(x.corr(),annot=True)
import matplotlib.pyplot as plt
plt.figure(figsize=(10,9))
sns.heatmap(x.corr(),annot=True, cmap='coolwarm',linewidth=0.5)
##line two variables
plt.figure(figsize=(20,9))
sns.lineplot(data=y)
sns.lineplot(data=y,x='chlorides',y='pH')
line three variables
sns.lineplot(data=x)
# check some plots after normalizing the data
x1=data_najmun_wine[['fixed acidity','chlorides','pH']]
y1=data_najmun_wine[['chlorides','pH']]
sns.lineplot(data=y1)
sns.lineplot(data=x1)
sns.lineplot(data=y,x='chlorides',y='pH')
#Normalize the data in order to apply clustering
data_najmun_wine_norm = (data_najmun_wine - data_najmun_wine.min()) / (data_najmun_wine.max()
data_najmun_wine.min())
data_najmun_wine_norm.head()
Console 1/A
...: data_najmun_wine_norm = (data_najmun_wine - data_najmun_wine.min()) /
(data_najmun_wine.max() -
...: data_najmun_wine.min())
...: data_najmun_wine_norm.head()
Out[6]:
fixed acidity volatile acidity citric acid residual sugar chlorides \
0 0.247788 0.397260 0.00 0.068493 0.106845
1 0.283186 0.520548 0.00 0.116438 0.143573
2 0.283186 0.438356 0.04 0.095890 0.133556
3 0.584071 0.109589 0.56 0.068493 0.105175
4 0.247788 0.397260 0.00 0.068493 0.106845
free sulfur dioxide total sulfur dioxide density pH sulphates \
0 0.140845 0.098940 0.567548 0.606299 0.137725
1 0.338028 0.215548 0.494126 0.362205 0.209581
2 0.197183 0.169611 0.508811 0.409449 0.191617
3 0.225352 0.190813 0.582232 0.330709 0.149701
4 0.140845 0.098940 0.567548 0.606299 0.137725
alcohol quality
0 0.153846 0.4
1 0.215385 0.4
2 0.215385 0.4
3 0.215385 0.6
4 0.153846 0.4
In [7]:
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 75, Col 29
UTF-8
CRLF
RW
Mem 85%

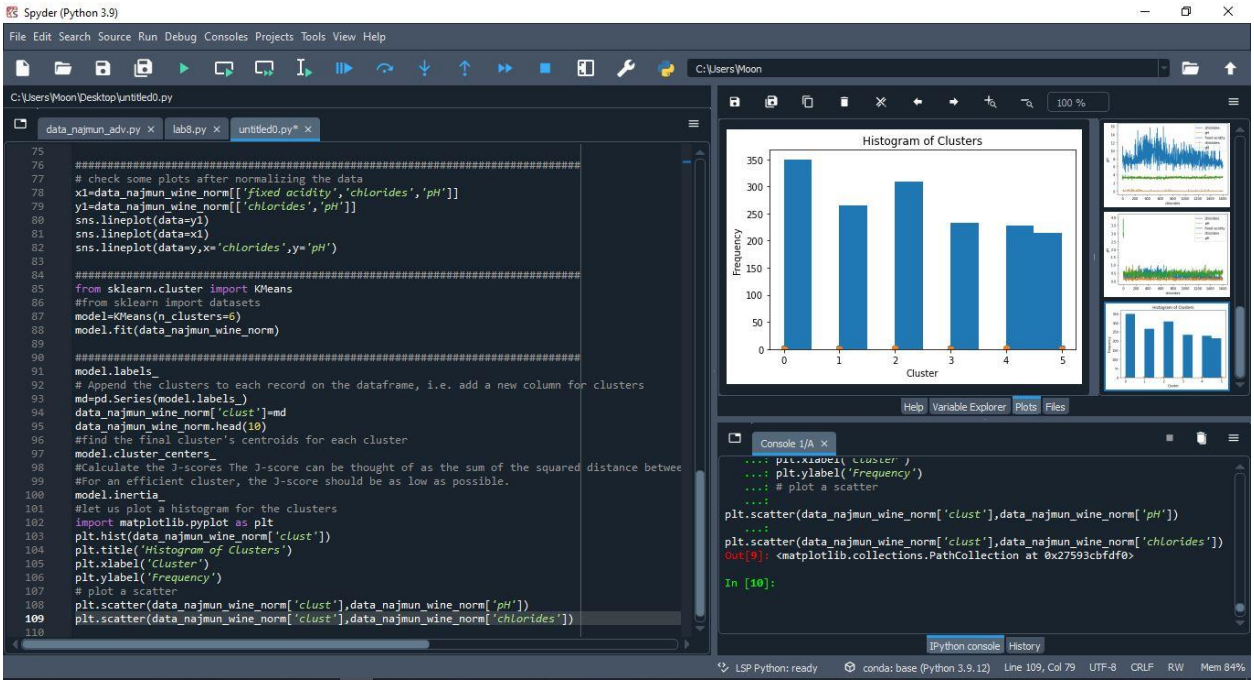
6.


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon
C:\Users\Moon\Desktop\untitled0.py
data_najmun_adv.py
lab8.py
untitled0.py*
# Generate heatmaps
sns.heatmap(data_najmun_wine[['fixed acidity']])
sns.heatmap(x)
sns.heatmap(x.corr())
sns.heatmap(x.corr(),annot=True)
import matplotlib.pyplot as plt
plt.figure(figsize=(10,9))
sns.heatmap(x.corr(),annot=True, cmap='coolwarm',linewidth=0.5)
##line two variables
plt.figure(figsize=(20,9))
sns.lineplot(data=y)
sns.lineplot(data=y,x='chlorides',y='pH')
line three variables
sns.lineplot(data=x)
# check some plots after normalizing the data
x1=data_najmun_wine[['fixed acidity','chlorides','pH']]
y1=data_najmun_wine[['chlorides','pH']]
sns.lineplot(data=y1)
sns.lineplot(data=x1)
sns.lineplot(data=y,x='chlorides',y='pH')
#Normalize the data in order to apply clustering
data_najmun_wine_norm = (data_najmun_wine - data_najmun_wine.min()) / (data_najmun_wine.max() -
data_najmun_wine_norm.head()
# check some plots after normalizing the data
x1=data_najmun_wine_norm[['fixed acidity','chlorides','pH']]
y1=data_najmun_wine_norm[['chlorides','pH']]
sns.lineplot(data=y1)
sns.lineplot(data=x1)
sns.lineplot(data=y,x='chlorides',y='pH')
chlorides
pH
fixed acidity
Help
Variable Explorer
Plots
Files
Console 1/A
3 0.215385 0.6
4 0.153846 0.4
In [7]: # check some plots after normalizing the data
...: x1=data_najmun_wine_norm[['fixed acidity','chlorides','pH']]
...: y1=data_najmun_wine_norm[['chlorides','pH']]
...: sns.lineplot(data=y1)
...: sns.lineplot(data=x1)
...: sns.lineplot(data=y,x='chlorides',y='pH')
Out[7]: <AxesSubplot:xlabel='chlorides', ylabel='pH'>
In [8]:
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 82, Col 42
UTF-8
CRLF
RW
Mem 85%

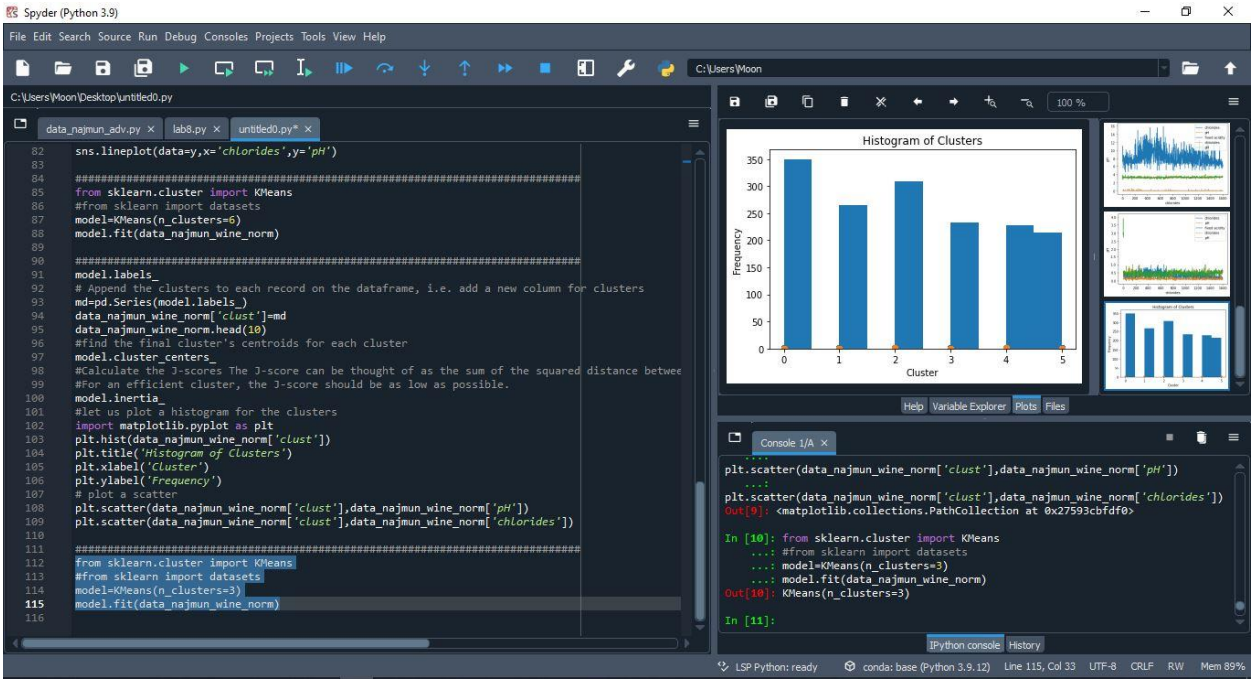
7.


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon
C:\Users\Moon\Desktop\untitled0.py
data_najmun_adv.py
lab8.py
untitled0.py*
plt.figure(figsize=(10,9))
sns.heatmap(x.corr(),annot=True, cmap='coolwarm',linewidth=0.5)
##line two variables
plt.figure(figsize=(20,9))
sns.lineplot(data=y)
sns.lineplot(data=y,x='chlorides',y='pH')
line three variables
sns.lineplot(data=x)
# check some plots after normalizing the data
x1=data_najmun_wine[['fixed acidity','chlorides','pH']]
y1=data_najmun_wine[['chlorides','pH']]
sns.lineplot(data=y1)
sns.lineplot(data=x1)
sns.lineplot(data=y,x='chlorides',y='pH')
#Normalize the data in order to apply clustering
data_najmun_wine_norm = (data_najmun_wine - data_najmun_wine.min()) / (data_najmun_wine.max() -
data_najmun_wine_norm.head()
# check some plots after normalizing the data
x1=data_najmun_wine_norm[['fixed acidity','chlorides','pH']]
y1=data_najmun_wine_norm[['chlorides','pH']]
sns.lineplot(data=y1)
sns.lineplot(data=x1)
sns.lineplot(data=y,x='chlorides',y='pH')
from sklearn.cluster import KMeans
#from sklearn import datasets
model=KMeans(n_clusters=6)
model.fit(data_najmun_wine_norm)
Name
Type
Size
Value
data_najmun_wine_norm
DataFrame
(1599, 12)
Column names: fixe…
filename
str
8
wine.csv
fullpath
str
35
C:/Users/Moon/Desk…
model
cluster._kmeans.KMeans
1
KMeans object of s…
path
str
27
C:/Users/Moon/ Desktop/ex-7/
x
DataFrame
(1599, 3)
Column names: fixe…
x1
DataFrame
(1599, 3)
Column names: fixe…
y
DataFrame
(1599, 2)
Column names: chlorides, pH
y1
DataFrame
(1599, 2)
Column names: chlorides, pH
Help
Variable Explorer
Plots
Files
Console 1/A
...: sns.lineplot(data=y,x='chlorides',y='pH')
Out[7]: <AxesSubplot:xlabel='chlorides', ylabel='pH'>
In [8]: from sklearn.cluster import KMeans
...: #from sklearn import datasets
...: model=KMeans(n_clusters=6)
...: model.fit(data_najmun_wine_norm)
Out[8]: KMeans(n_clusters=6)
In [9]:
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 88, Col 33
UTF-8
CRLF
RW
Mem 86%

9.


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon
C:\Users\Moon\Desktop\untitled0.py
data_najmun_adv.py
lab8.py
untitled0.py*
# check some plots after normalizing the data
x1=data_najmun_wine_norm[['fixed acidity','chlorides','pH']]
y1=data_najmun_wine_norm[['chlorides','pH']]
sns.lineplot(data=y1)
sns.lineplot(data=x1)
sns.lineplot(data=y,x='chlorides',y='pH')
from sklearn.cluster import KMeans
#from sklearn import datasets
model=KMeans(n_clusters=6)
model.fit(data_najmun_wine_norm)
model.labels_
# Append the clusters to each record on the dataframe, i.e. add a new column for clusters
md=pd.Series(model.labels_)
data_najmun_wine_norm['clust']=md
data_najmun_wine_norm.head(10)
#find the final cluster's centroids for each cluster
model.cluster_centers_
#Calculate the J-scores The J-score can be thought of as the sum of the squared distance betwee
#For an efficient cluster, the J-score should be as low as possible.
model.inertia_
#let us plot a histogram for the clusters
import matplotlib.pyplot as plt
plt.hist(data_najmun_wine_norm['clust'])
plt.title('Histogram of Clusters')
plt.xlabel('Cluster')
plt.ylabel('Frequency')
# plot a scatter
plt.scatter(data_najmun_wine_norm['clust'],data_najmun_wine_norm['pH'])
plt.scatter(data_najmun_wine_norm['clust'],data_najmun_wine_norm['chlorides'])
Histogram of Clusters
Frequency
Cluster
Help
Variable Explorer
Plots
Files
Console 1/A
plt.scatter(data_najmun_wine_norm['clust'],data_najmun_wine_norm['pH'])
plt.scatter(data_najmun_wine_norm['clust'],data_najmun_wine_norm['chlorides'])
Out[9]: <matplotlib.collections.PathCollection at 0x27593cbfdf0>
In [10]:
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 109, Col 79
UTF-8
CRLF
RW
Mem 84%

10.


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Moon
C:\Users\Moon\Desktop\untitled0.py
data_najmun_adv.py
lab8.py
untitled0.py*
sns.lineplot(data=y,x='chlorides',y='pH')
from sklearn.cluster import KMeans
#from sklearn import datasets
model=KMeans(n_clusters=6)
model.fit(data_najmun_wine_norm)
model.labels_
# Append the clusters to each record on the dataframe, i.e. add a new column for clusters
md=pd.Series(model.labels_)
data_najmun_wine_norm['clust']=md
data_najmun_wine_norm.head(10)
#find the final cluster's centroids for each cluster
model.cluster_centers_
#Calculate the J-scores The J-score can be thought of as the sum of the squared distance betwee
#For an efficient cluster, the J-score should be as low as possible.
model.inertia_
#let us plot a histogram for the clusters
import matplotlib.pyplot as plt
plt.hist(data_najmun_wine_norm['clust'])
plt.title('Histogram of Clusters')
plt.xlabel('Cluster')
plt.ylabel('Frequency')
# plot a scatter
plt.scatter(data_najmun_wine_norm['clust'],data_najmun_wine_norm['pH'])
plt.scatter(data_najmun_wine_norm['clust'],data_najmun_wine_norm['chlorides'])
from sklearn.cluster import KMeans
#from sklearn import datasets
model=KMeans(n_clusters=3)
model.fit(data_najmun_wine_norm)
Histogram of Clusters
Frequency
Cluster
Help
Variable Explorer
Plots
Files
Console 1/A
plt.scatter(data_najmun_wine_norm['clust'],data_najmun_wine_norm['pH'])
plt.scatter(data_najmun_wine_norm['clust'],data_najmun_wine_norm['chlorides'])
Out[9]: <matplotlib.collections.PathCollection at 0x27593cbfdf0>
In [10]: from sklearn.cluster import KMeans
#from sklearn import datasets
model=KMeans(n_clusters=3)
model.fit(data_najmun_wine_norm)
Out[10]: KMeans(n_clusters=3)
In [11]:
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 115, Col 33
UTF-8
CRLF
RW
Mem 89%