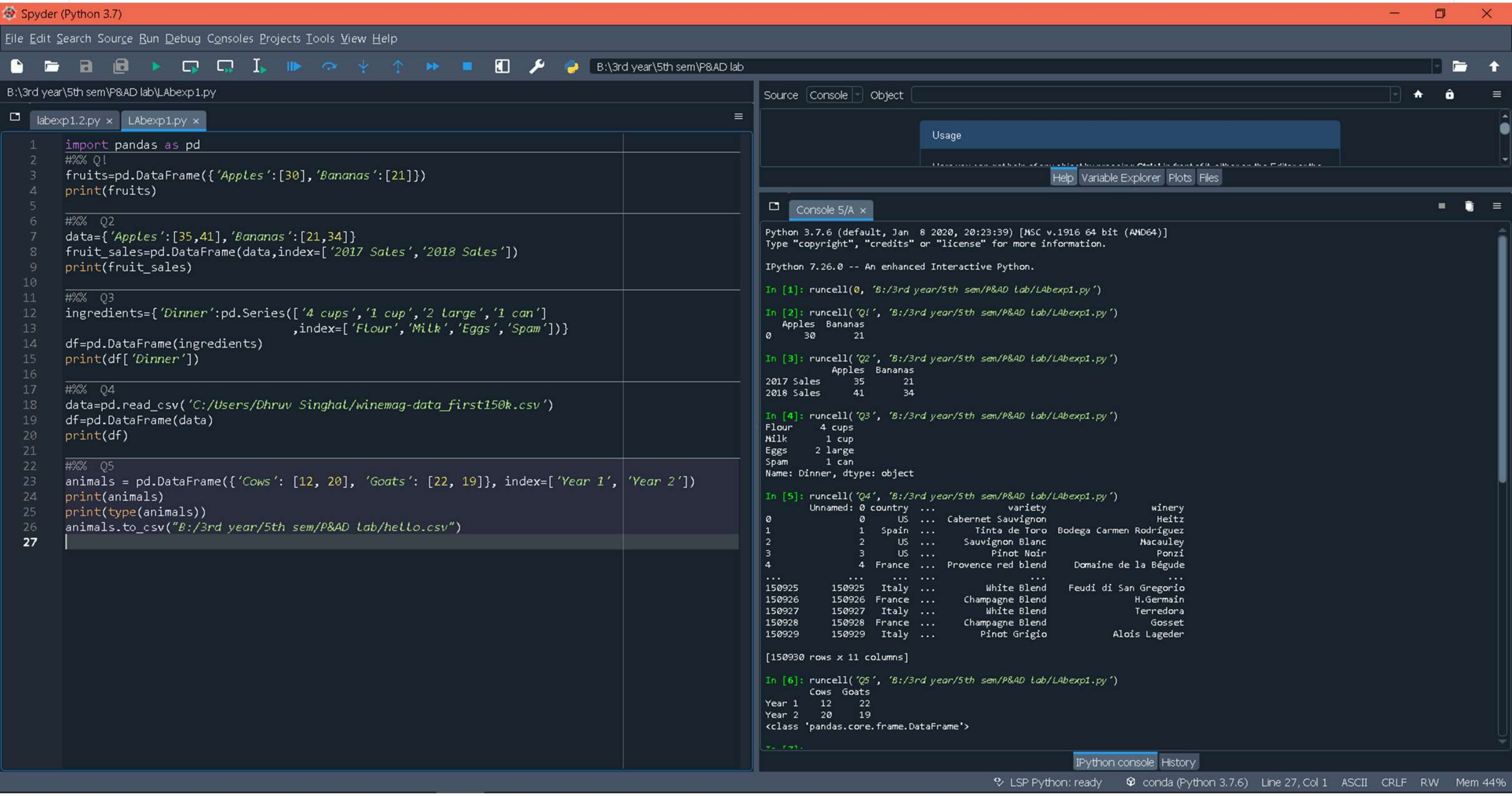
Pattern and Anomaly Detection Lab
Experiment 2
Using NumPy and Pandas

Submitted By:
Dhruv Singhal
500075346
R177219074
AIML B3(SEM 5)

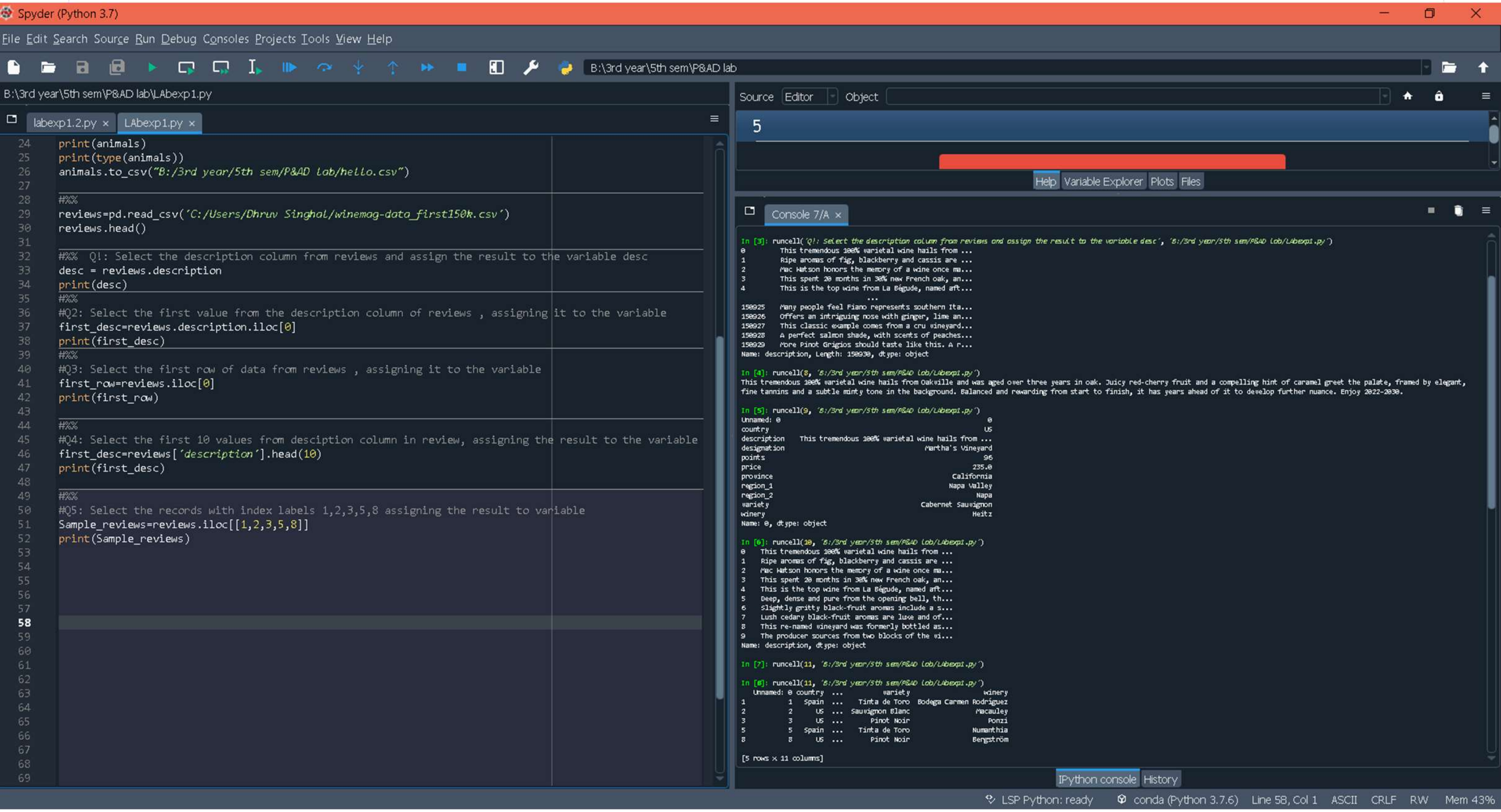
Submitted To:
Dr. Gopal Phartiyal
Professor
SOCS
UPES

Pandas



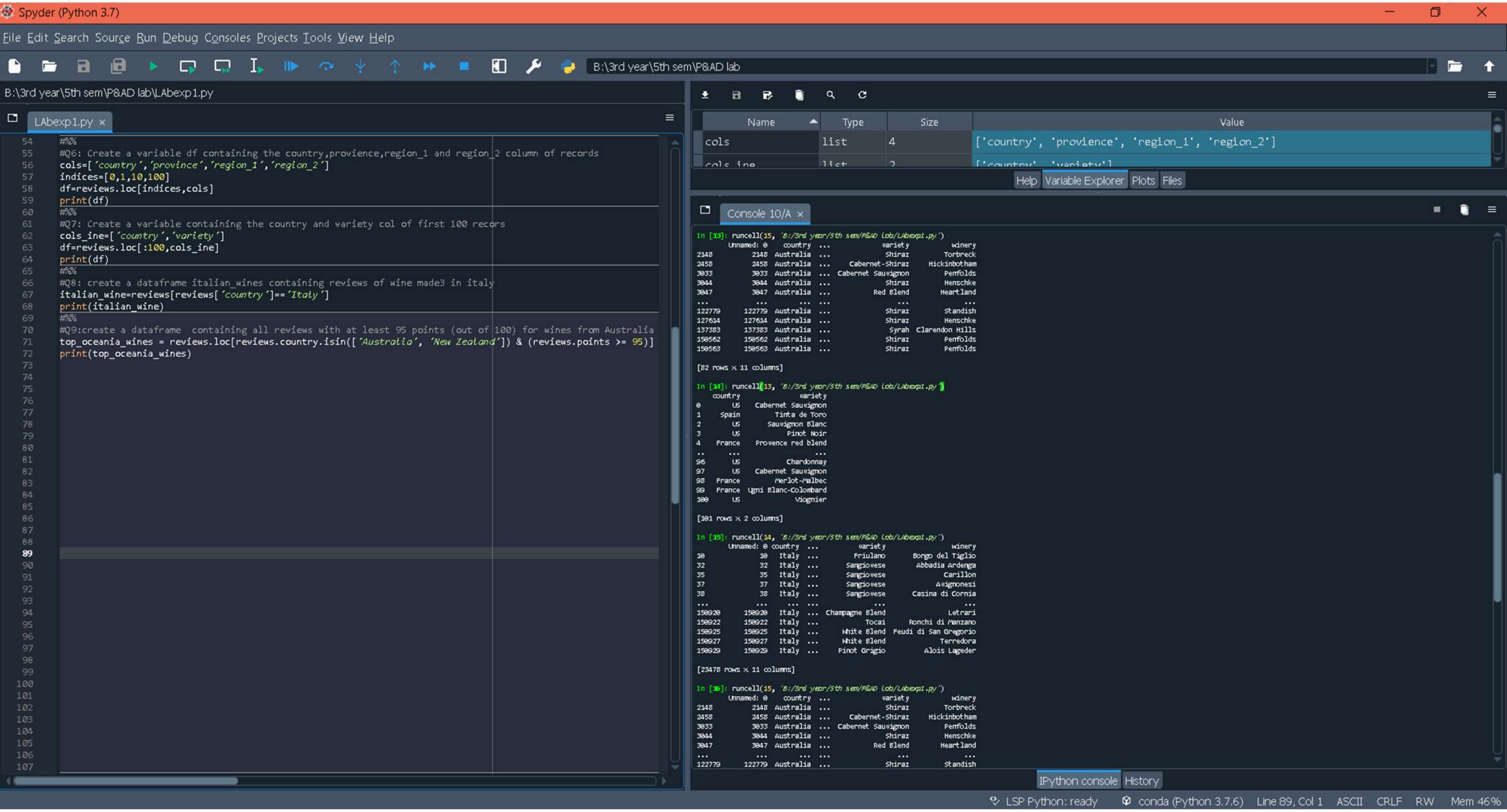
```python
import pandas as pd
#%% Q!
fruits=pd.DataFrame({'Apples':[30],'Bananas':[21]})
print(fruits)

#%%  Q2
data={'Apples':[35,41],'Bananas':[21,34]}
fruit_sales=pd.DataFrame(data,index=['2017 Sales','2018 Sales'])
print(fruit_sales)

#%%  Q3
ingredients={'Dinner':pd.Series(['4 cups','1 cup','2 large','1 can']
                                 ,index=['Flour','Milk','Eggs','Spam'])}
df=pd.DataFrame(ingredients)
print(df['Dinner'])

#%%  Q4
data=pd.read_csv('C:/Users/Dhruv Singhal/winemag-data_first150k.csv')
df=pd.DataFrame(data)
print(df)

#%%  Q5
animals = pd.DataFrame({'Cows': [12, 20], 'Goats': [22, 19]}, index=['Year 1', 'Year 2'])
print(animals)
print(type(animals))
animals.to_csv("B:/3rd year/5th sem/P&AD lab/hello.csv")
```

```
Python 3.7.6 (default, Jan  8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.26.0 -- An enhanced Interactive Python.

In [1]: runcell(0, 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')

In [2]: runcell('Q!', 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
   Apples  Bananas
0      30       21

In [3]: runcell('Q2', 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
            Apples  Bananas
2017 Sales      35       21
2018 Sales      41       34

In [4]: runcell('Q3', 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
Flour     4 cups
Milk       1 cup
Eggs     2 large
Spam       1 can
Name: Dinner, dtype: object

In [5]: runcell('Q4', 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
        Unnamed: 0 country  ...                variety                 winery
0                0      US  ...     Cabernet Sauvignon                  Heitz
1                1   Spain  ...         Tinta de Toro  Bodega Carmen Rodríguez
2                2      US  ...        Sauvignon Blanc               Macauley
3                3      US  ...             Pinot Noir                  Ponzi
4                4  France  ...     Provence red blend     Domaine de la Bégude
...            ...     ...  ...                    ...                    ...
150925      150925   Italy  ...            White Blend    Feudi di San Gregorio
150926      150926  France  ...        Champagne Blend              H.Germain
150927      150927   Italy  ...            White Blend              Terredora
150928      150928  France  ...        Champagne Blend                 Gosset
150929      150929   Italy  ...           Pinot Grigio          Alois Lageder

[150930 rows x 11 columns]

In [6]: runcell('Q5', 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
        Cows  Goats
Year 1    12     22
Year 2    20     19
<class 'pandas.core.frame.DataFrame'>
```


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
B:\3rd year\5th sem\P&AD lab
B:\3rd year\5th sem\P&AD lab\LAbexp1.py
labexp1.2.py
LAbexp1.py
Source Editor Object
5
Help Variable Explorer Plots Files
Console 7/A
IPython console History
LSP Python: ready
conda (Python 3.7.6)
Line 58, Col 1
ASCII
CRLF
RW
Mem 43%

```python
print(animals)
print(type(animals))
animals.to_csv("B:/3rd year/5th sem/P&AD lab/hello.csv")

#%%
reviews=pd.read_csv('C:/Users/Dhruv Singhal/winemag-data_first150k.csv')
reviews.head()

#%%  Q1: Select the description column from reviews and assign the result to the variable desc
desc = reviews.description
print(desc)
#%%
#Q2: Select the first value from the description column of reviews , assigning it to the variable
first_desc=reviews.description.iloc[0]
print(first_desc)
#%%
#Q3: Select the first row of data from reviews , assigning it to the variable
first_row=reviews.iloc[0]
print(first_row)

#%%
#Q4: Select the first 10 values from desciption column in review, assigning the result to the variable
first_desc=reviews['description'].head(10)
print(first_desc)

#%%
#Q5: Select the records with index labels 1,2,3,5,8 assigning the result to variable
Sample_reviews=reviews.iloc[[1,2,3,5,8]]
print(Sample_reviews)
```

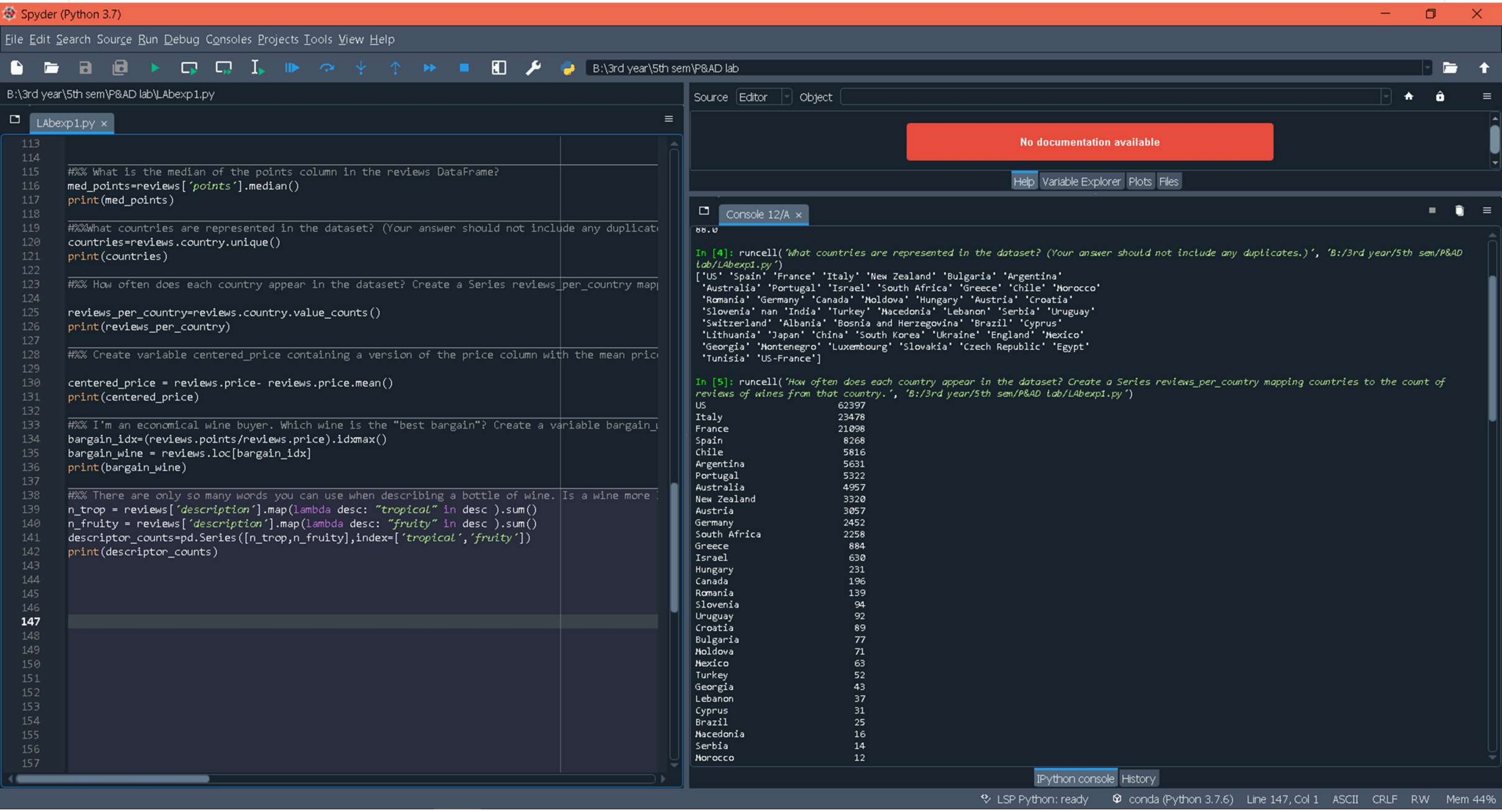
```
In [3]: runcell('Q1: Select the description column from reviews and assign the result to the variable desc', 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
0         This tremendous 100% varietal wine hails from ...
1         Ripe aromas of fig, blackberry and cassis are ...
2         Mac Watson honors the memory of a wine once ma...
3         This spent 20 months in 30% new French oak, an...
4         This is the top wine from La Bégude, named aft...
                                ...
150925    Many people feel Fiano represents southern Ita...
150926    Offers an intriguing nose with ginger, lime an...
150927    This classic example comes from a cru vineyard...
150928    A perfect salmon shade, with scents of peaches...
150929    More Pinot Grigios should taste like this. A r...
Name: description, Length: 150930, dtype: object

In [4]: runcell(8, 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
This tremendous 100% varietal wine hails from Oakville and was aged over three years in oak. Juicy red-cherry fruit and a compelling hint of caramel greet the palate, framed by elegant,
fine tannins and a subtle minty tone in the background. Balanced and rewarding from start to finish, it has years ahead of it to develop further nuance. Enjoy 2022-2030.

In [5]: runcell(9, 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
Unnamed: 0                                                     0
country                                                       US
description    This tremendous 100% varietal wine hails from ...
designation                                    Martha's Vineyard
points                                                        96
price                                                      235.0
province                                              California
region_1                                             Napa Valley
region_2                                                    Napa
variety                                       Cabernet Sauvignon
winery                                                     Heitz
Name: 0, dtype: object

In [6]: runcell(10, 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
0    This tremendous 100% varietal wine hails from ...
1    Ripe aromas of fig, blackberry and cassis are ...
2    Mac Watson honors the memory of a wine once ma...
3    This spent 20 months in 30% new French oak, an...
4    This is the top wine from La Bégude, named aft...
5    Deep, dense and pure from the opening bell, th...
6    Slightly gritty black-fruit aromas include a s...
7    Lush cedary black-fruit aromas are luxe and of...
8    This re-named vineyard was formerly bottled as...
9    The producer sources from two blocks of the vi...
Name: description, dtype: object

In [7]: runcell(11, 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')

In [8]: runcell(11, 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
   Unnamed: 0 country  ...             variety                    winery
1           1   Spain  ...       Tinta de Toro  Bodega Carmen Rodríguez
2           2      US  ...   Sauvignon Blanc                   Macauley
3           3      US  ...         Pinot Noir                      Ponzi
5           5   Spain  ...       Tinta de Toro                 Numanthia
8           8      US  ...         Pinot Noir                  Bergström

[5 rows x 11 columns]
```


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
B:\3rd year\5th sem\P&AD lab
B:\3rd year\5th sem\P&AD lab\LAbexp1.py
LAbexp1.py
Help Variable Explorer Plots Files
Console 10/A
IPython console History
LSP Python: ready
conda (Python 3.7.6)
Line 89, Col 1
ASCII
CRLF
RW
Mem 46%

```
#%%
#Q6: Create a variable df containing the country,provience,region_1 and region_2 column of records
cols=['country','province','region_1','region_2']
indices=[0,1,10,100]
df=reviews.loc[indices,cols]
print(df)
#%%
#Q7: Create a variable containing the country and variety col of first 100 recors
cols_ine=['country','variety']
df=reviews.loc[:100,cols_ine]
print(df)
#%%
#Q8: create a dataframe italian_wines containing reviews of wine made3 in italy
italian_wine=reviews[reviews['country']=='Italy']
print(italian_wine)
#%%
#Q9:create a dataframe  containing all reviews with at least 95 points (out of 100) for wines from Australia
top_oceania_wines = reviews.loc[reviews.country.isin(['Australia', 'New Zealand']) & (reviews.points >= 95)]
print(top_oceania_wines)
```

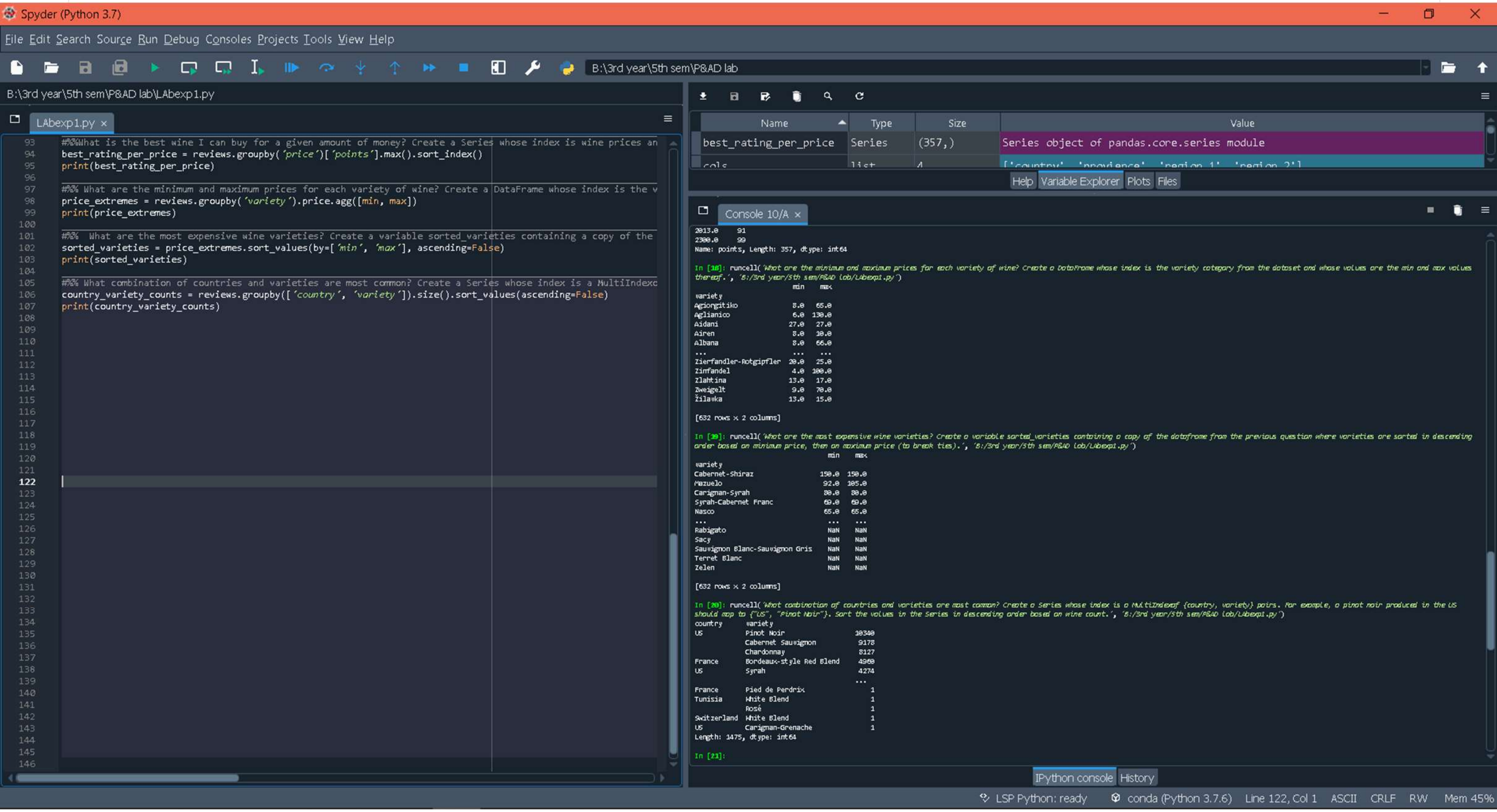
Name	Type	Size	Value
cols	list	4	['country', 'provience', 'region_1', 'region_2']
cols_ine	list	2	['country', 'variety']

```
In [13]: runcell(15, 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
        Unnamed: 0    country  ...                variety             winery
2148          2148  Australia  ...                 Shiraz           Torbreck
2458          2458  Australia  ...         Cabernet-Shiraz       Hickinbotham
3033          3033  Australia  ...  Cabernet Sauvignon           Penfolds
3044          3044  Australia  ...                 Shiraz           Henschke
3047          3047  Australia  ...              Red Blend          Heartland
...            ...        ...  ...                    ...                ...
122779      122779  Australia  ...                 Shiraz           Standish
127614      127614  Australia  ...                 Shiraz           Henschke
137383      137383  Australia  ...                  Syrah  Clarendon Hills
150562      150562  Australia  ...                 Shiraz           Penfolds
150563      150563  Australia  ...                 Shiraz           Penfolds

[82 rows x 11 columns]

In [14]: runcell(13, 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
    country                variety
0        US     Cabernet Sauvignon
1     Spain          Tinta de Toro
2        US        Sauvignon Blanc
3        US             Pinot Noir
4    France     Provence red blend
..      ...                    ...
96       US             Chardonnay
97       US     Cabernet Sauvignon
98   France           Merlot-Malbec
99   France  Ugni Blanc-Colombard
100      US               Viognier

[101 rows x 2 columns]

In [15]: runcell(14, 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
        Unnamed: 0 country  ...            variety                   winery
10              10   Italy  ...           Friulano        Borgo del Tiglio
32              32   Italy  ...         Sangiovese         Abbadia Ardenga
35              35   Italy  ...         Sangiovese                Carillon
37              37   Italy  ...         Sangiovese              Avignonesi
38              38   Italy  ...         Sangiovese       Casina di Cornia
...            ...     ...  ...                ...                     ...
150920      150920   Italy  ...  Champagne Blend                  Letrari
150922      150922   Italy  ...              Tocai       Ronchi di Manzano
150925      150925   Italy  ...        White Blend  Feudi di San Gregorio
150927      150927   Italy  ...        White Blend               Terredora
150929      150929   Italy  ...       Pinot Grigio           Alois Lageder

[23478 rows x 11 columns]

In [16]: runcell(15, 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
        Unnamed: 0    country  ...                variety             winery
2148          2148  Australia  ...                 Shiraz           Torbreck
2458          2458  Australia  ...         Cabernet-Shiraz       Hickinbotham
3033          3033  Australia  ...  Cabernet Sauvignon           Penfolds
3044          3044  Australia  ...                 Shiraz           Henschke
3047          3047  Australia  ...              Red Blend          Heartland
...            ...        ...  ...                    ...                ...
122779      122779  Australia  ...                 Shiraz           Standish
```


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
B:\3rd year\5th sem\P&AD lab
B:\3rd year\5th sem\P&AD lab\LAbexp1.py
LAbexp1.py
Source Editor Object
No documentation available
Help Variable Explorer Plots Files
Console 12/A
IPython console History
LSP Python: ready conda (Python 3.7.6) Line 147, Col 1 ASCII CRLF RW Mem 44%

```
#%% What is the median of the points column in the reviews DataFrame?
med_points=reviews['points'].median()
print(med_points)

#%%What countries are represented in the dataset? (Your answer should not include any duplicat
countries=reviews.country.unique()
print(countries)

#%% How often does each country appear in the dataset? Create a Series reviews_per_country map

reviews_per_country=reviews.country.value_counts()
print(reviews_per_country)

#%% Create variable centered_price containing a version of the price column with the mean pric

centered_price = reviews.price- reviews.price.mean()
print(centered_price)

#%% I'm an economical wine buyer. Which wine is the "best bargain"? Create a variable bargain_
bargain_idx=(reviews.points/reviews.price).idxmax()
bargain_wine = reviews.loc[bargain_idx]
print(bargain_wine)

#%% There are only so many words you can use when describing a bottle of wine. Is a wine more
n_trop = reviews['description'].map(lambda desc: "tropical" in desc ).sum()
n_fruity = reviews['description'].map(lambda desc: "fruity" in desc ).sum()
descriptor_counts=pd.Series([n_trop,n_fruity],index=['tropical','fruity'])
print(descriptor_counts)
```

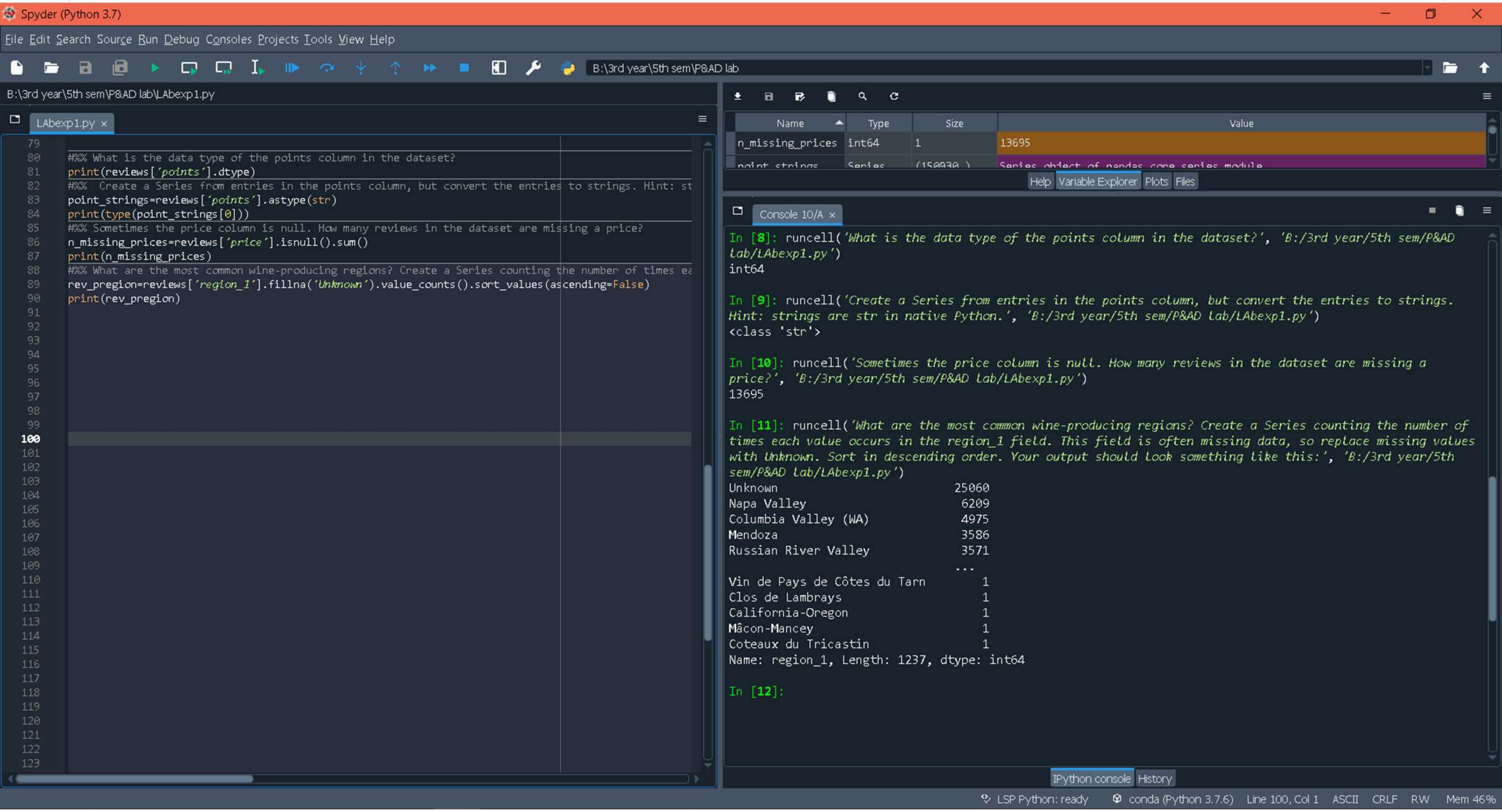
```
88.0

In [4]: runcell('What countries are represented in the dataset? (Your answer should not include any duplicates.)', 'B:/3rd year/5th sem/P&AD
lab/LAbexp1.py')
['US' 'Spain' 'France' 'Italy' 'New Zealand' 'Bulgaria' 'Argentina'
 'Australia' 'Portugal' 'Israel' 'South Africa' 'Greece' 'Chile' 'Morocco'
 'Romania' 'Germany' 'Canada' 'Moldova' 'Hungary' 'Austria' 'Croatia'
 'Slovenia' nan 'India' 'Turkey' 'Macedonia' 'Lebanon' 'Serbia' 'Uruguay'
 'Switzerland' 'Albania' 'Bosnia and Herzegovina' 'Brazil' 'Cyprus'
 'Lithuania' 'Japan' 'China' 'South Korea' 'Ukraine' 'England' 'Mexico'
 'Georgia' 'Montenegro' 'Luxembourg' 'Slovakia' 'Czech Republic' 'Egypt'
 'Tunisia' 'US-France']

In [5]: runcell('How often does each country appear in the dataset? Create a Series reviews_per_country mapping countries to the count of
reviews of wines from that country.', 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
US                        62397
Italy                     23478
France                    21098
Spain                      8268
Chile                      5816
Argentina                  5631
Portugal                   5322
Australia                  4957
New Zealand                3320
Austria                    3057
Germany                    2452
South Africa               2258
Greece                      884
Israel                      630
Hungary                     231
Canada                      196
Romania                     139
Slovenia                     94
Uruguay                      92
Croatia                      89
Bulgaria                     77
Moldova                      71
Mexico                       63
Turkey                       52
Georgia                      43
Lebanon                      37
Cyprus                       31
Brazil                       25
Macedonia                    16
Serbia                       14
Morocco                      12
```



Name	Type	Size	Value
best_rating_per_price	Series	(357,)	Series object of pandas.core.series module

```python
#%%What is the best wine I can buy for a given amount of money? Create a Series whose index is wine prices an
best_rating_per_price = reviews.groupby('price')['points'].max().sort_index()
print(best_rating_per_price)

#%% What are the minimum and maximum prices for each variety of wine? Create a DataFrame whose index is the v
price_extremes = reviews.groupby('variety').price.agg([min, max])
print(price_extremes)

#%%  What are the most expensive wine varieties? Create a variable sorted_varieties containing a copy of the
sorted_varieties = price_extremes.sort_values(by=['min', 'max'], ascending=False)
print(sorted_varieties)

#%% What combination of countries and varieties are most common? Create a Series whose index is a MultiIndexc
country_variety_counts = reviews.groupby(['country', 'variety']).size().sort_values(ascending=False)
print(country_variety_counts)
```

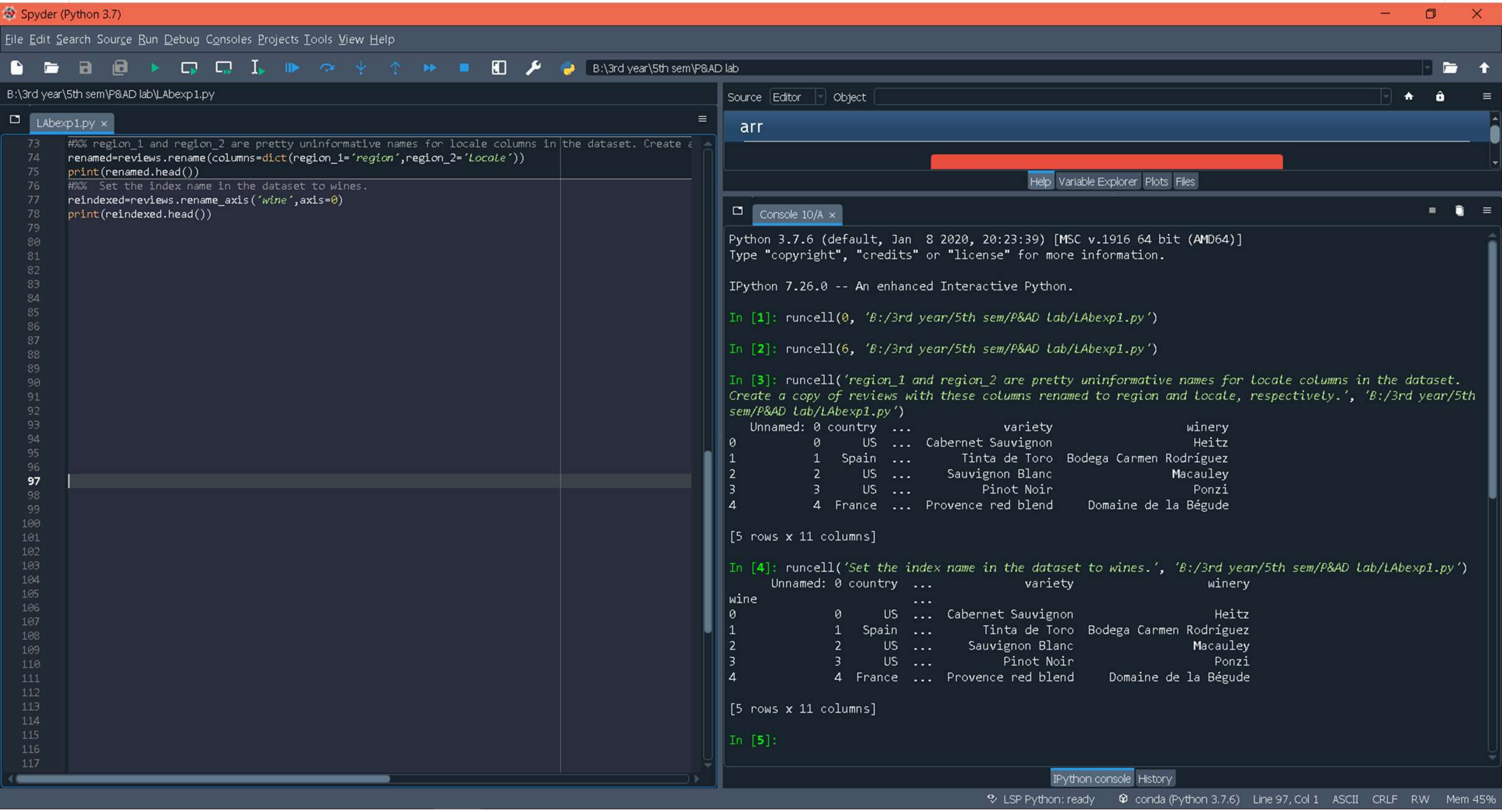
```
2013.0    91
2300.0    99
Name: points, Length: 357, dtype: int64

In [18]: runcell('What are the minimum and maximum prices for each variety of wine? Create a DataFrame whose index is the variety category from the dataset and whose values are the min and max values thereof.', 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
                        min    max
variety
Agiorgitiko             8.0   65.0
Aglianico               6.0  130.0
Aidani                 27.0   27.0
Airen                   8.0   10.0
Albana                  8.0   66.0
...                     ...    ...
Zierfandler-Rotgipfler 20.0   25.0
Zinfandel               4.0  100.0
Zlahtina               13.0   17.0
Zweigelt                9.0   70.0
Žilavka                13.0   15.0

[632 rows x 2 columns]

In [19]: runcell('What are the most expensive wine varieties? Create a variable sorted_varieties containing a copy of the dataframe from the previous question where varieties are sorted in descending order based on minimum price, then on maximum price (to break ties).', 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
                                  min    max
variety
Cabernet-Shiraz                 150.0  150.0
Mazuelo                          92.0  105.0
Carignan-Syrah                   80.0   80.0
Syrah-Cabernet Franc             69.0   69.0
Nasco                            65.0   65.0
...                               ...    ...
Rabigato                          NaN    NaN
Sacy                              NaN    NaN
Sauvignon Blanc-Sauvignon Gris    NaN    NaN
Terret Blanc                      NaN    NaN
Zelen                             NaN    NaN

[632 rows x 2 columns]

In [20]: runcell('What combination of countries and varieties are most common? Create a Series whose index is a MultiIndexof {country, variety} pairs. For example, a pinot noir produced in the US should map to {"US", "Pinot Noir"}. Sort the values in the Series in descending order based on wine count.', 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
country      variety
US           Pinot Noir                  10340
             Cabernet Sauvignon           9178
             Chardonnay                   8127
France       Bordeaux-style Red Blend     4960
US           Syrah                        4274
                                          ...
France       Pied de Perdrix                 1
Tunisia      White Blend                     1
             Rosé                            1
Switzerland  White Blend                     1
US           Carignan-Grenache               1
Length: 1475, dtype: int64

In [21]:
```



```python
#%% What is the data type of the points column in the dataset?
print(reviews['points'].dtype)
#%%  Create a Series from entries in the points column, but convert the entries to strings. Hint: st
point_strings=reviews['points'].astype(str)
print(type(point_strings[0]))
#%% Sometimes the price column is null. How many reviews in the dataset are missing a price?
n_missing_prices=reviews['price'].isnull().sum()
print(n_missing_prices)
#%% What are the most common wine-producing regions? Create a Series counting the number of times ea
rev_pregion=reviews['region_1'].fillna('Unknown').value_counts().sort_values(ascending=False)
print(rev_pregion)
```

Name	Type	Size	Value
n_missing_prices	int64	1	13695
point_strings	Series	(150930,)	Series object of pandas.core.series module

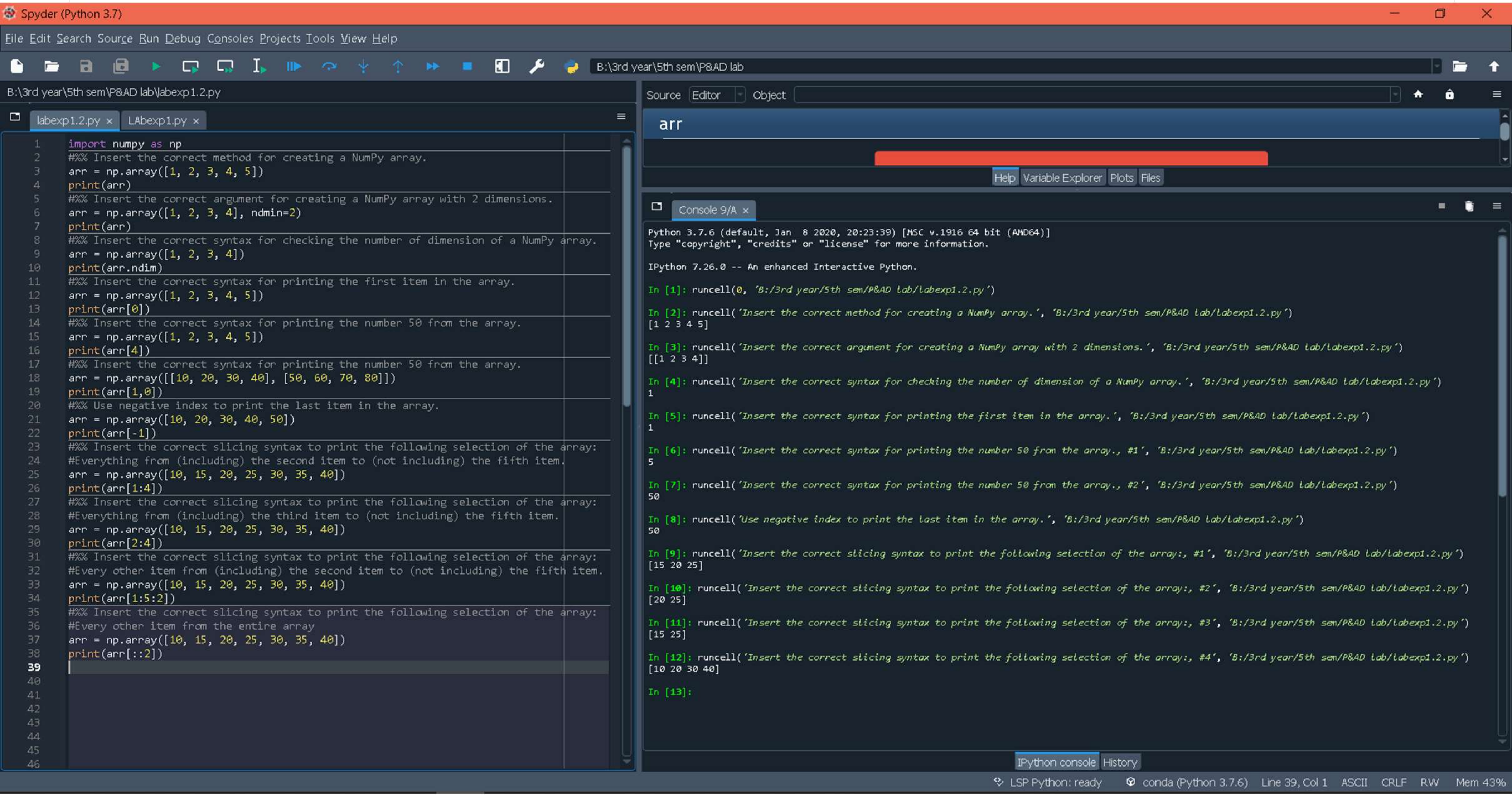
```
In [8]: runcell('What is the data type of the points column in the dataset?', 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
int64

In [9]: runcell('Create a Series from entries in the points column, but convert the entries to strings. Hint: strings are str in native Python.', 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
<class 'str'>

In [10]: runcell('Sometimes the price column is null. How many reviews in the dataset are missing a price?', 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
13695

In [11]: runcell('What are the most common wine-producing regions? Create a Series counting the number of times each value occurs in the region_1 field. This field is often missing data, so replace missing values with Unknown. Sort in descending order. Your output should look something like this:', 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
Unknown                         25060
Napa Valley                      6209
Columbia Valley (WA)             4975
Mendoza                          3586
Russian River Valley             3571
                                 ...
Vin de Pays de Côtes du Tarn        1
Clos de Lambrays                    1
California-Oregon                   1
Mâcon-Mancey                        1
Coteaux du Tricastin                1
Name: region_1, Length: 1237, dtype: int64

In [12]:
```


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
B:\3rd year\5th sem\P&AD lab
B:\3rd year\5th sem\P&AD lab\LAbexp1.py
LAbexp1.py
Source Editor Object
arr
Help Variable Explorer Plots Files
Console 10/A
IPython console History
LSP Python: ready conda (Python 3.7.6) Line 97, Col 1 ASCII CRLF RW Mem 45%

```
#%% region_1 and region_2 are pretty uninformative names for locale columns in the dataset. Create a
renamed=reviews.rename(columns=dict(region_1='region',region_2='Locale'))
print(renamed.head())
#%%  Set the index name in the dataset to wines.
reindexed=reviews.rename_axis('wine',axis=0)
print(reindexed.head())
```

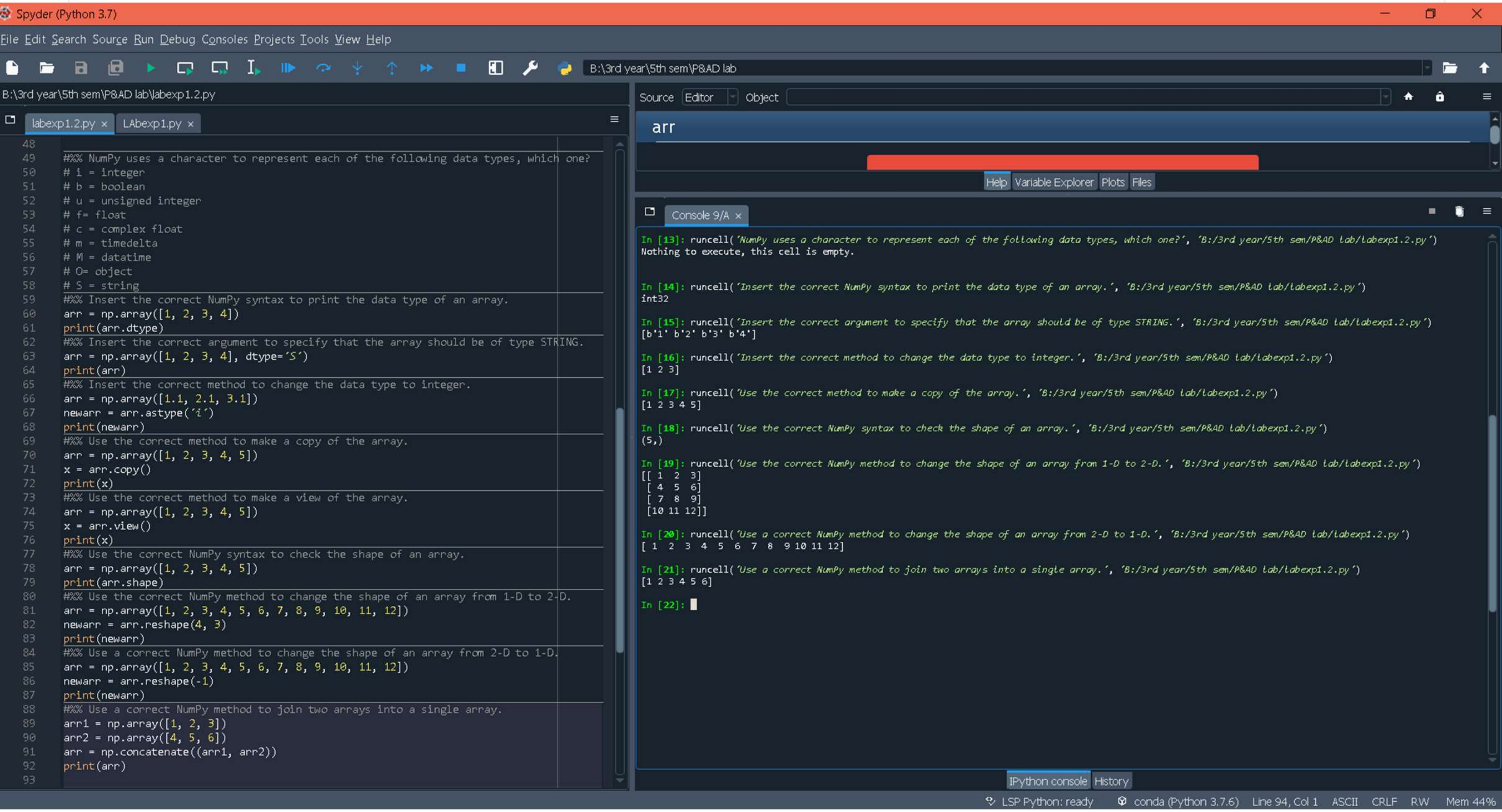
```
Python 3.7.6 (default, Jan  8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.26.0 -- An enhanced Interactive Python.

In [1]: runcell(0, 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')

In [2]: runcell(6, 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')

In [3]: runcell('region_1 and region_2 are pretty uninformative names for locale columns in the dataset.
Create a copy of reviews with these columns renamed to region and locale, respectively.', 'B:/3rd year/5th
sem/P&AD lab/LAbexp1.py')
   Unnamed: 0 country  ...              variety                   winery
0           0      US  ...   Cabernet Sauvignon                    Heitz
1           1   Spain  ...        Tinta de Toro  Bodega Carmen Rodríguez
2           2      US  ...      Sauvignon Blanc                 Macauley
3           3      US  ...           Pinot Noir                    Ponzi
4           4  France  ...   Provence red blend      Domaine de la Bégude

[5 rows x 11 columns]

In [4]: runcell('Set the index name in the dataset to wines.', 'B:/3rd year/5th sem/P&AD lab/LAbexp1.py')
      Unnamed: 0 country  ...              variety                   winery
wine                      ...
0              0      US  ...   Cabernet Sauvignon                    Heitz
1              1   Spain  ...        Tinta de Toro  Bodega Carmen Rodríguez
2              2      US  ...      Sauvignon Blanc                 Macauley
3              3      US  ...           Pinot Noir                    Ponzi
4              4  France  ...   Provence red blend      Domaine de la Bégude

[5 rows x 11 columns]

In [5]:
```

NumPy


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
B:\3rd year\5th sem\P&AD lab
B:\3rd year\5th sem\P&AD lab\labexp1.2.py
labexp1.2.py
LAbexp1.py
Source Editor Object
arr
Help Variable Explorer Plots Files
Console 9/A
IPython console History
LSP Python: ready
conda (Python 3.7.6)
Line 39, Col 1
ASCII
CRLF
RW
Mem 43%

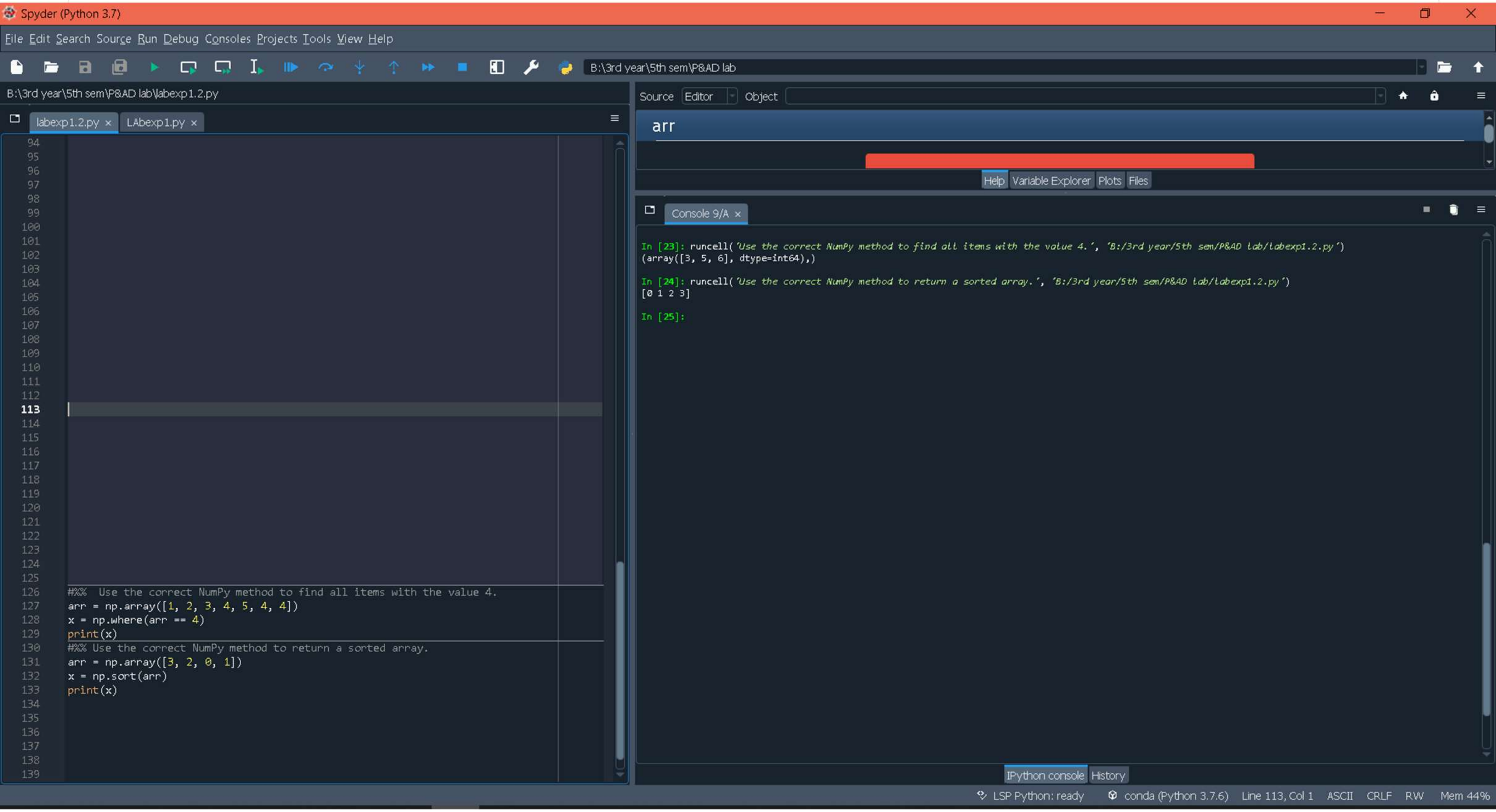
```python
import numpy as np
#%% Insert the correct method for creating a NumPy array.
arr = np.array([1, 2, 3, 4, 5])
print(arr)
#%% Insert the correct argument for creating a NumPy array with 2 dimensions.
arr = np.array([1, 2, 3, 4], ndmin=2)
print(arr)
#%% Insert the correct syntax for checking the number of dimension of a NumPy array.
arr = np.array([1, 2, 3, 4])
print(arr.ndim)
#%% Insert the correct syntax for printing the first item in the array.
arr = np.array([1, 2, 3, 4, 5])
print(arr[0])
#%% Insert the correct syntax for printing the number 50 from the array.
arr = np.array([1, 2, 3, 4, 5])
print(arr[4])
#%% Insert the correct syntax for printing the number 50 from the array.
arr = np.array([[10, 20, 30, 40], [50, 60, 70, 80]])
print(arr[1,0])
#%% Use negative index to print the last item in the array.
arr = np.array([10, 20, 30, 40, 50])
print(arr[-1])
#%% Insert the correct slicing syntax to print the following selection of the array:
#Everything from (including) the second item to (not including) the fifth item.
arr = np.array([10, 15, 20, 25, 30, 35, 40])
print(arr[1:4])
#%% Insert the correct slicing syntax to print the following selection of the array:
#Everything from (including) the third item to (not including) the fifth item.
arr = np.array([10, 15, 20, 25, 30, 35, 40])
print(arr[2:4])
#%% Insert the correct slicing syntax to print the following selection of the array:
#Every other item from (including) the second item to (not including) the fifth item.
arr = np.array([10, 15, 20, 25, 30, 35, 40])
print(arr[1:5:2])
#%% Insert the correct slicing syntax to print the following selection of the array:
#Every other item from the entire array
arr = np.array([10, 15, 20, 25, 30, 35, 40])
print(arr[::2])
```

```
Python 3.7.6 (default, Jan  8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.26.0 -- An enhanced Interactive Python.

In [1]: runcell(0, 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')

In [2]: runcell('Insert the correct method for creating a NumPy array.', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
[1 2 3 4 5]

In [3]: runcell('Insert the correct argument for creating a NumPy array with 2 dimensions.', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
[[1 2 3 4]]

In [4]: runcell('Insert the correct syntax for checking the number of dimension of a NumPy array.', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
1

In [5]: runcell('Insert the correct syntax for printing the first item in the array.', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
1

In [6]: runcell('Insert the correct syntax for printing the number 50 from the array., #1', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
5

In [7]: runcell('Insert the correct syntax for printing the number 50 from the array., #2', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
50

In [8]: runcell('Use negative index to print the last item in the array.', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
50

In [9]: runcell('Insert the correct slicing syntax to print the following selection of the array:, #1', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
[15 20 25]

In [10]: runcell('Insert the correct slicing syntax to print the following selection of the array:, #2', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
[20 25]

In [11]: runcell('Insert the correct slicing syntax to print the following selection of the array:, #3', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
[15 25]

In [12]: runcell('Insert the correct slicing syntax to print the following selection of the array:, #4', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
[10 20 30 40]

In [13]:
```



```python

#%% NumPy uses a character to represent each of the following data types, which one?
# i = integer
# b = boolean
# u = unsigned integer
# f= float
# c = complex float
# m = timedelta
# M = datatime
# O= object
# S = string
#%% Insert the correct NumPy syntax to print the data type of an array.
arr = np.array([1, 2, 3, 4])
print(arr.dtype)
#%% Insert the correct argument to specify that the array should be of type STRING.
arr = np.array([1, 2, 3, 4], dtype='S')
print(arr)
#%% Insert the correct method to change the data type to integer.
arr = np.array([1.1, 2.1, 3.1])
newarr = arr.astype('i')
print(newarr)
#%% Use the correct method to make a copy of the array.
arr = np.array([1, 2, 3, 4, 5])
x = arr.copy()
print(x)
#%% Use the correct method to make a view of the array.
arr = np.array([1, 2, 3, 4, 5])
x = arr.view()
print(x)
#%% Use the correct NumPy syntax to check the shape of an array.
arr = np.array([1, 2, 3, 4, 5])
print(arr.shape)
#%% Use the correct NumPy method to change the shape of an array from 1-D to 2-D.
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
newarr = arr.reshape(4, 3)
print(newarr)
#%% Use a correct NumPy method to change the shape of an array from 2-D to 1-D.
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
newarr = arr.reshape(-1)
print(newarr)
#%% Use a correct NumPy method to join two arrays into a single array.
arr1 = np.array([1, 2, 3])
arr2 = np.array([4, 5, 6])
arr = np.concatenate((arr1, arr2))
print(arr)

```

```
In [13]: runcell('NumPy uses a character to represent each of the following data types, which one?', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
Nothing to execute, this cell is empty.


In [14]: runcell('Insert the correct NumPy syntax to print the data type of an array.', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
int32

In [15]: runcell('Insert the correct argument to specify that the array should be of type STRING.', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
[b'1' b'2' b'3' b'4']

In [16]: runcell('Insert the correct method to change the data type to integer.', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
[1 2 3]

In [17]: runcell('Use the correct method to make a copy of the array.', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
[1 2 3 4 5]

In [18]: runcell('Use the correct NumPy syntax to check the shape of an array.', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
(5,)

In [19]: runcell('Use the correct NumPy method to change the shape of an array from 1-D to 2-D.', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]

In [20]: runcell('Use a correct NumPy method to change the shape of an array from 2-D to 1-D.', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
[ 1  2  3  4  5  6  7  8  9 10 11 12]

In [21]: runcell('Use a correct NumPy method to join two arrays into a single array.', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
[1 2 3 4 5 6]

In [22]: 
```



```python
#%%  Use the correct NumPy method to find all items with the value 4.
arr = np.array([1, 2, 3, 4, 5, 4, 4])
x = np.where(arr == 4)
print(x)
#%% Use the correct NumPy method to return a sorted array.
arr = np.array([3, 2, 0, 1])
x = np.sort(arr)
print(x)
```

```
In [23]: runcell('Use the correct NumPy method to find all items with the value 4.', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
(array([3, 5, 6], dtype=int64),)

In [24]: runcell('Use the correct NumPy method to return a sorted array.', 'B:/3rd year/5th sem/P&AD lab/labexp1.2.py')
[0 1 2 3]

In [25]:
```