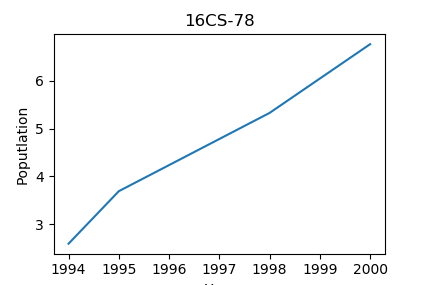
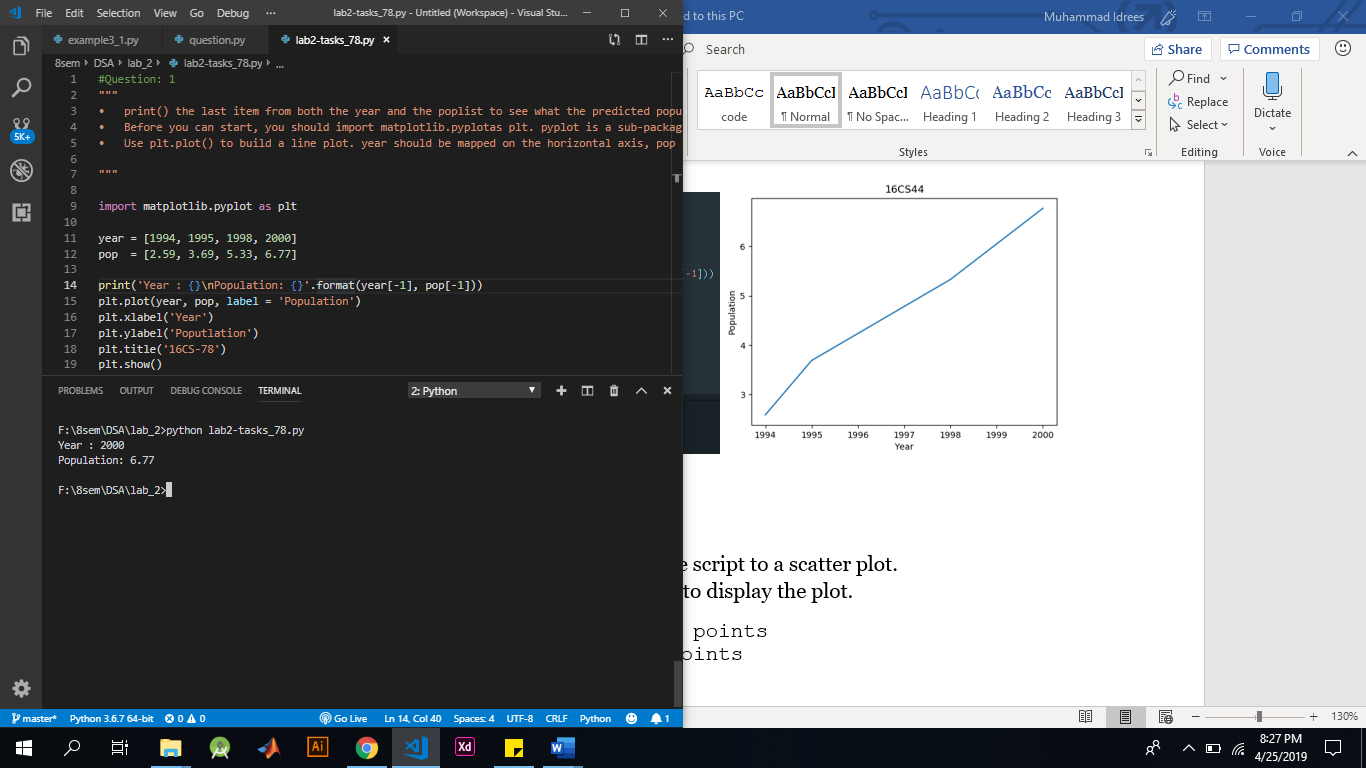
Practical# 2

# Introduction to Python for Data Sciences

### Question: 1

* print() the last item from both the year and the poplist to see what the predicted population for the year 2100 is.
* Before you can start, you should import matplotlib.pyplotas plt. pyplot is a sub-package of matplotlib, hence the dot.
* Use plt.plot() to build a line plot. year should be mapped on the horizontal axis, pop on the vertical axis. Don't forget to finish off with the show() function to actually display the plot.

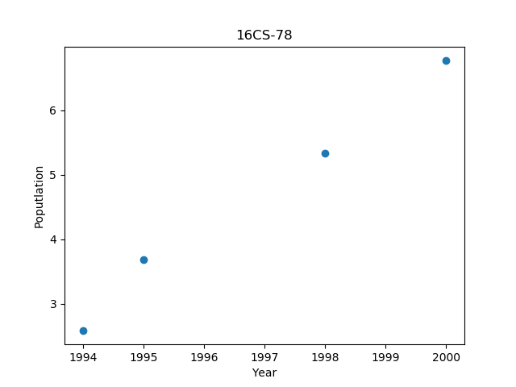
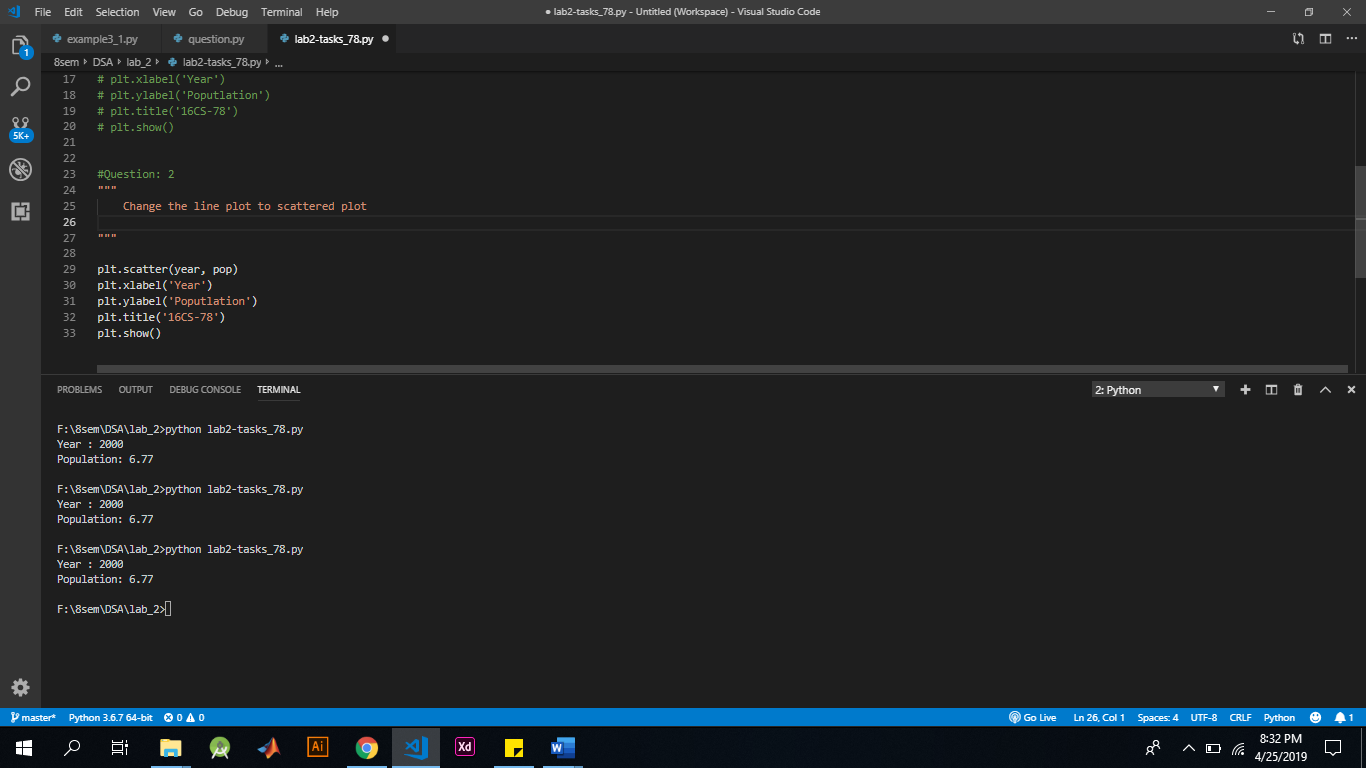
Answer:



### Question: 2

* Change the line plot that's coded in the script to a scatter plot.
* Finish off your script with plt.show() to display the plot.

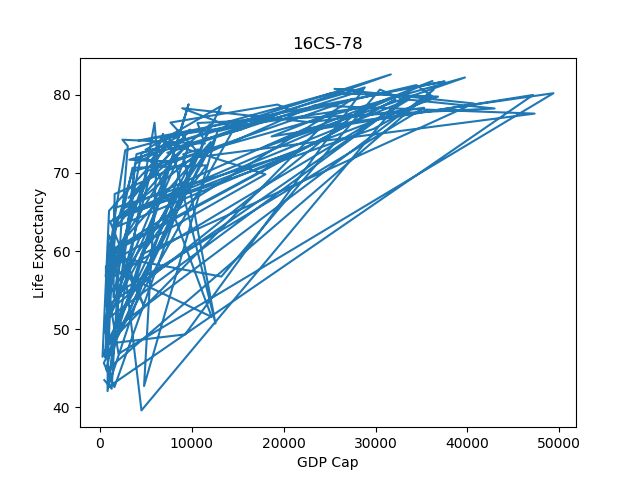
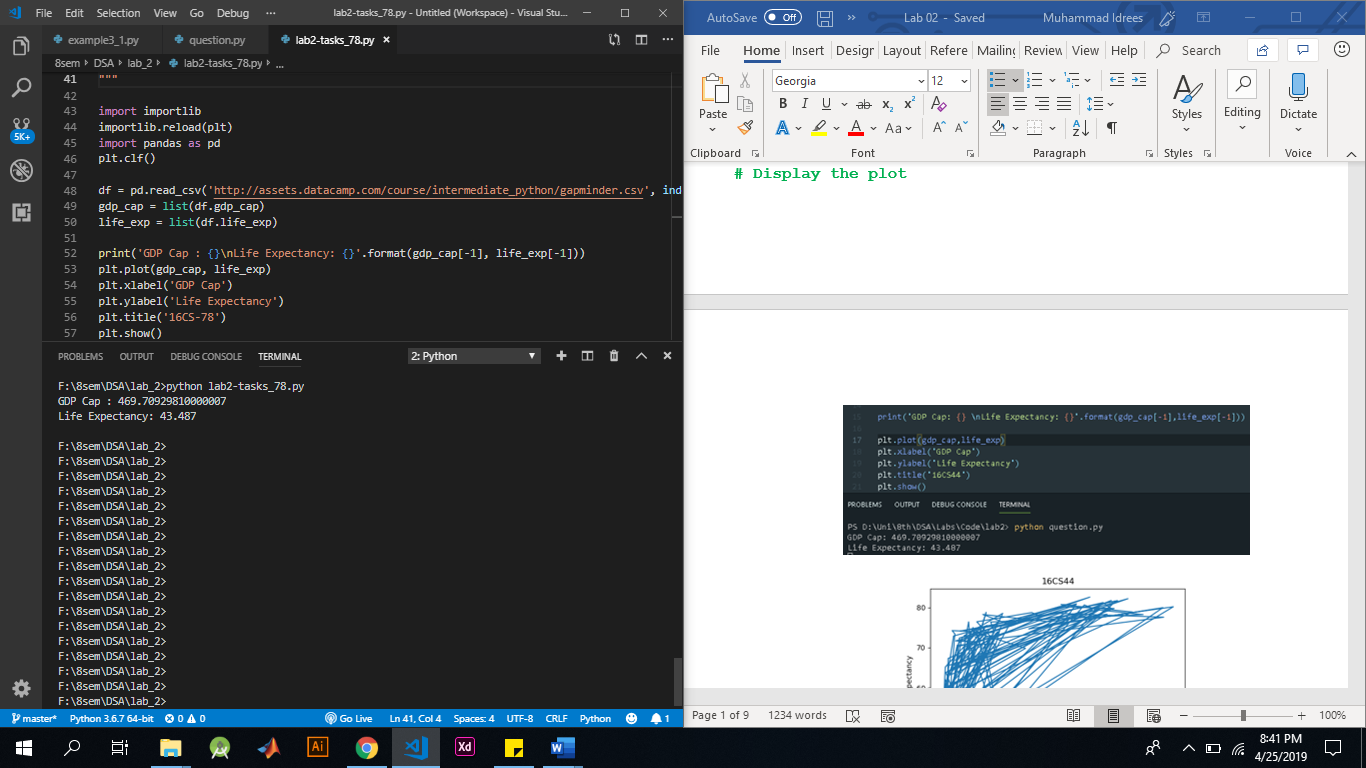
Answer:



### Question: 3

* Print the last item from both the list gdp\_cap, and the list life\_exp; it is information about Zimbabwe.
* Build a line chart, with gdp\_cap on the x-axis, and life\_exp on the y-axis. Does it make sense to plot this data on a line plot?
* Don't forget to finish off with a plt.show() command, to actually display the plot.

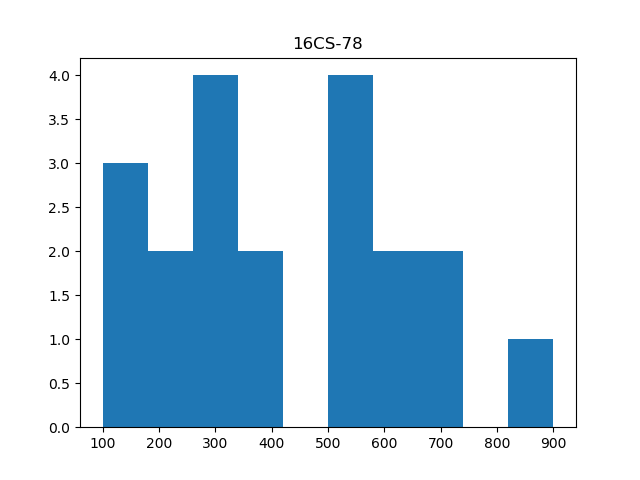
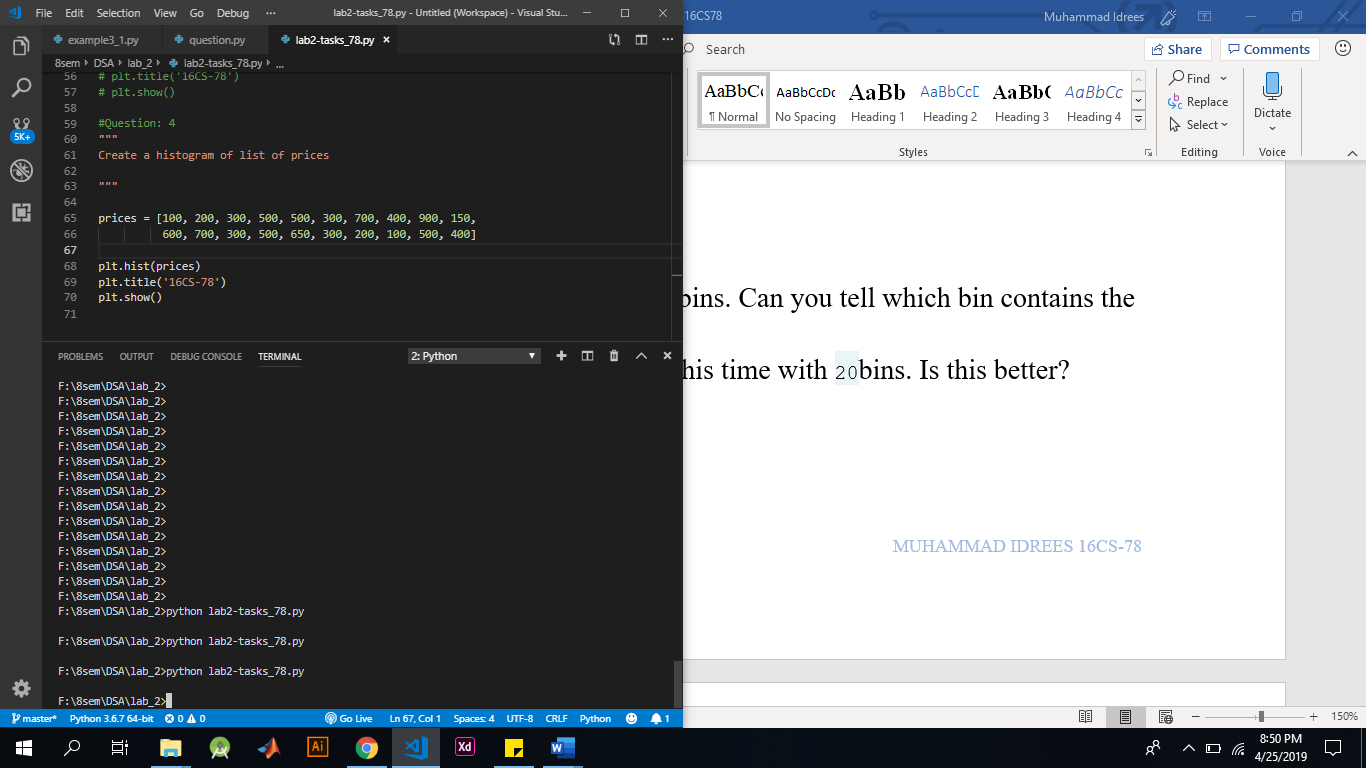
Answer:



### Question: 4

* Use plt.hist() to create a histogram of the values in prices(a list containing atleast 20 different values). Do not specify the number of bins; Python will set the number of bins to 10 by default for you.
* Add plt.show() to actually display the histogram. Can you tell which bin contains the most observations?

Answer:



### Question: 5

* Build a histogram of prices, with 5 bins. Can you tell which bin contains the most observations?
* Build another histogram of prices, this time with 20bins. Is this better?

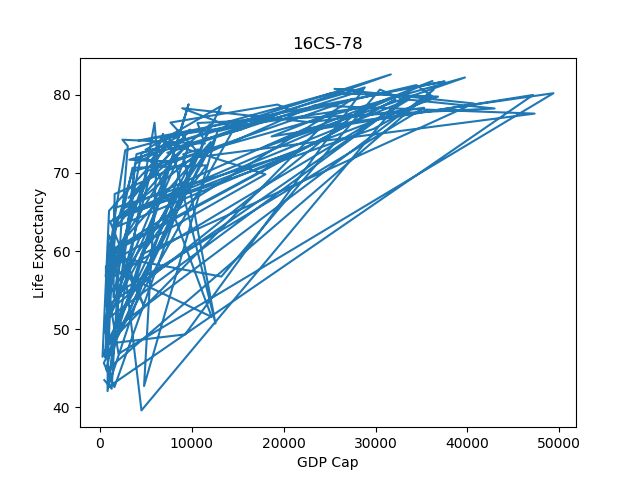
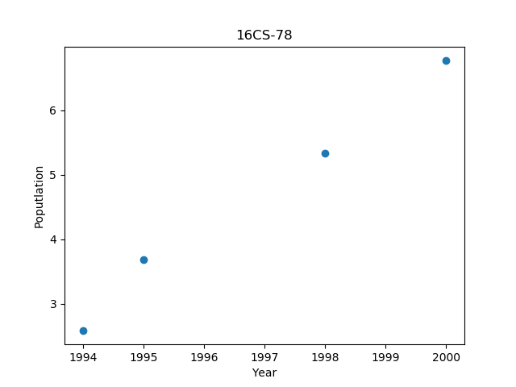
Answer:

### 

### Question: 6

Repeat question 2 and 3 and add valid xlabel, ylabel and title which should be your roll number.

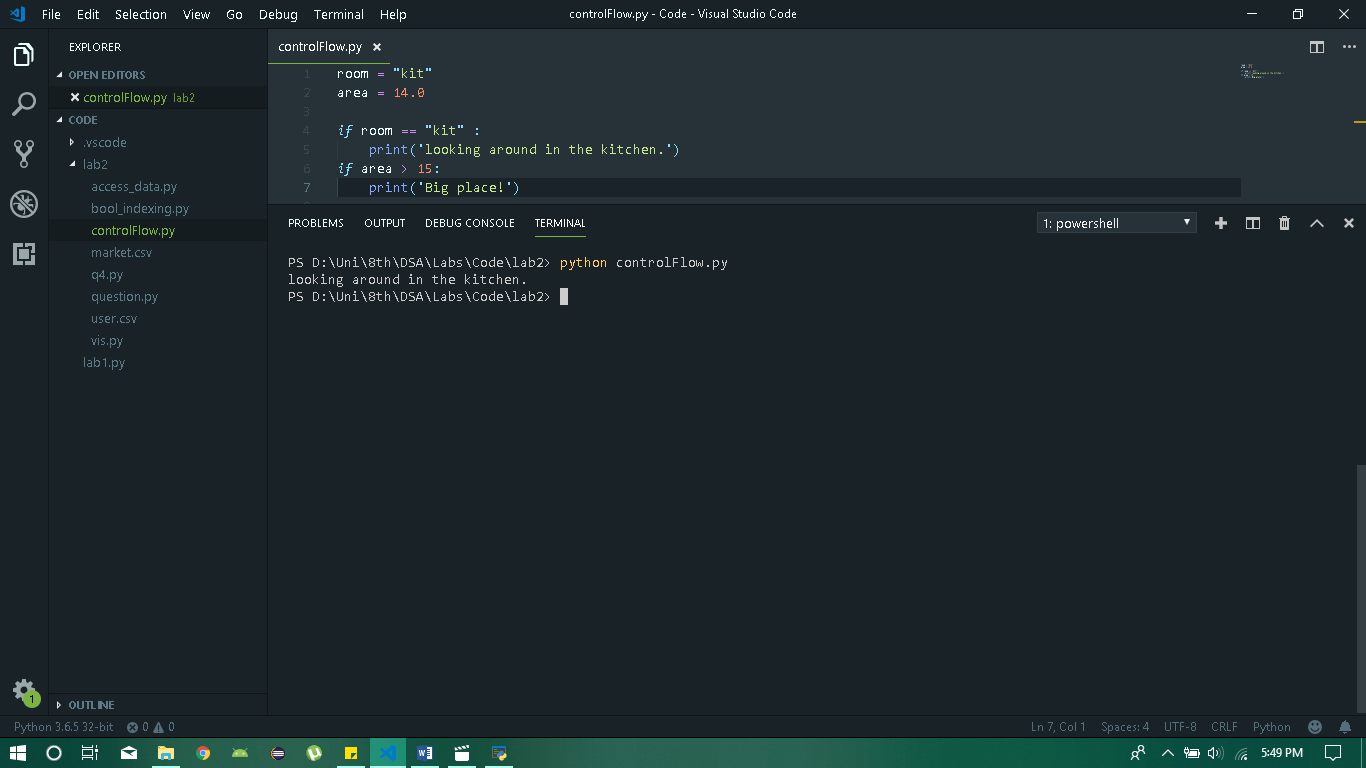
Answer:



### Question: 7

* Examine the if statement that prints out "Looking around in the kitchen." if room equals "kit".
* Write another if statement that prints out "big place!" if area is greater than 15.

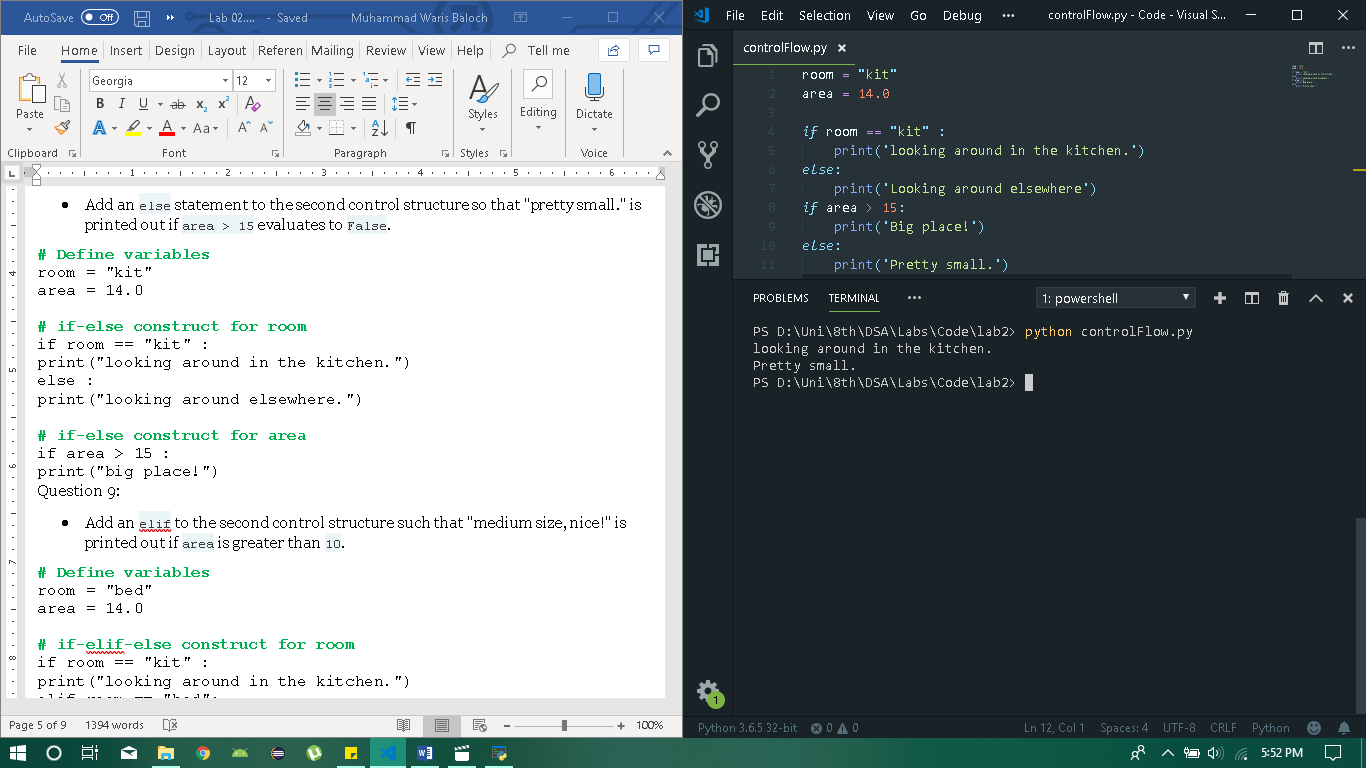
Answer:



### Question: 8

Add an else statement to the second control structure so that "pretty small." is printed out if area > 15 evaluates to False.

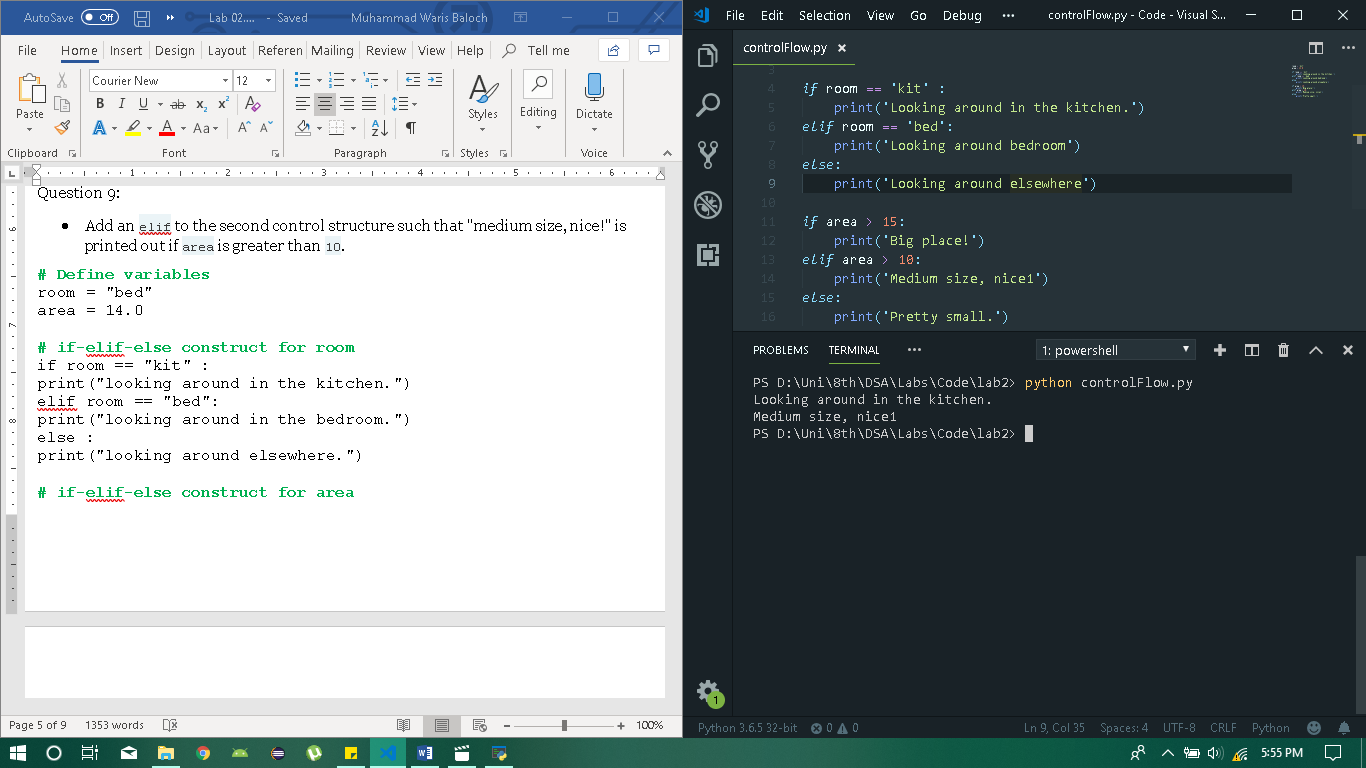
Answer:



### Question: 9

Add an elif to the second control structure such that "medium size, nice!" is printed out if area is greater than 10.

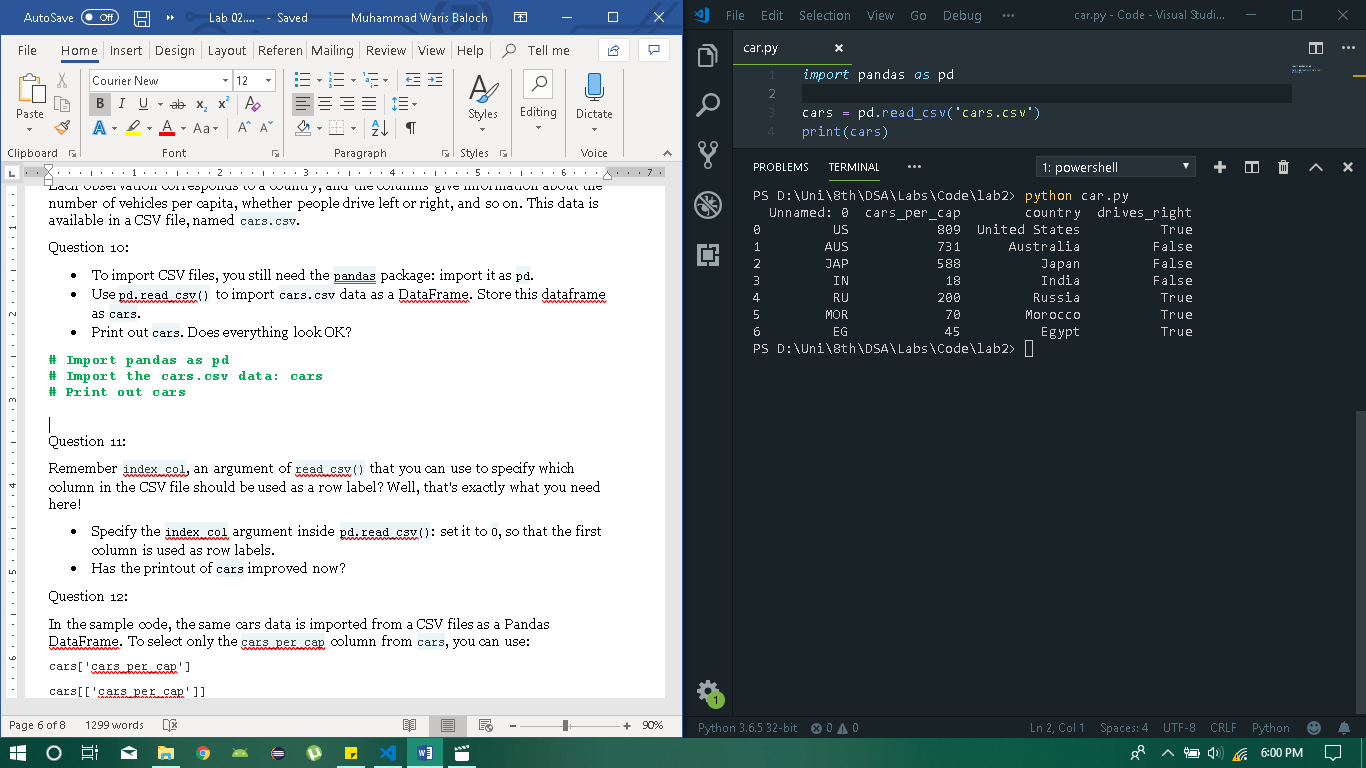
Answer:



### Question: 10

* To import CSV files, you still need the pandas package: import it as pd.
* Use pd.read\_csv() to import cars.csv data as a DataFrame. Store this dataframe as cars.
* Print out cars. Does everything look OK?

Answer:

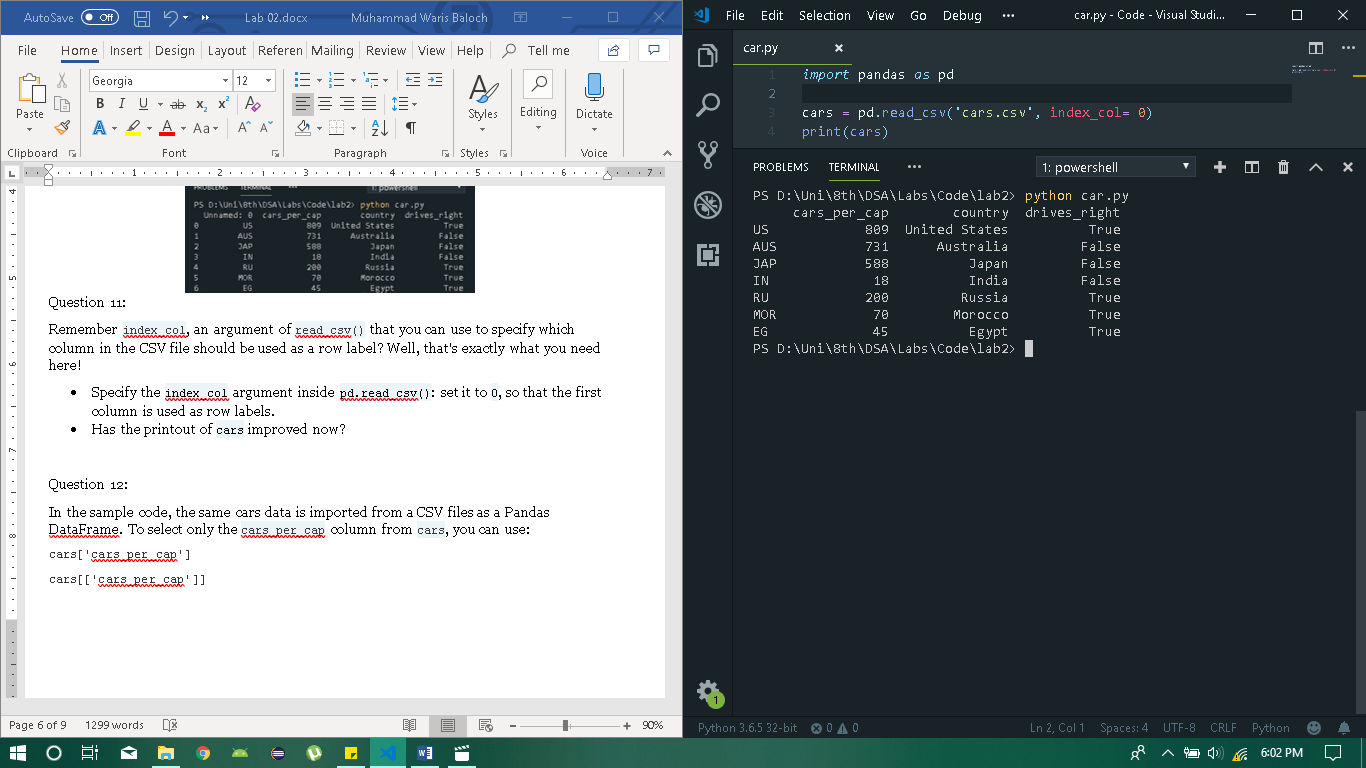


### Question: 11

Remember index\_col, an argument of read\_csv() that you can use to specify which column in the CSV file should be used as a row label? Well, that's exactly what you need here!

* Specify the index\_col argument inside pd.read\_csv(): set it to 0, so that the first column is used as row labels.
* Has the printout of cars improved now?

Answer:



### Question: 12

n the sample code, the same cars data is imported from a CSV files as a Pandas DataFrame. To select only the cars\_per\_cap column from cars, you can use:

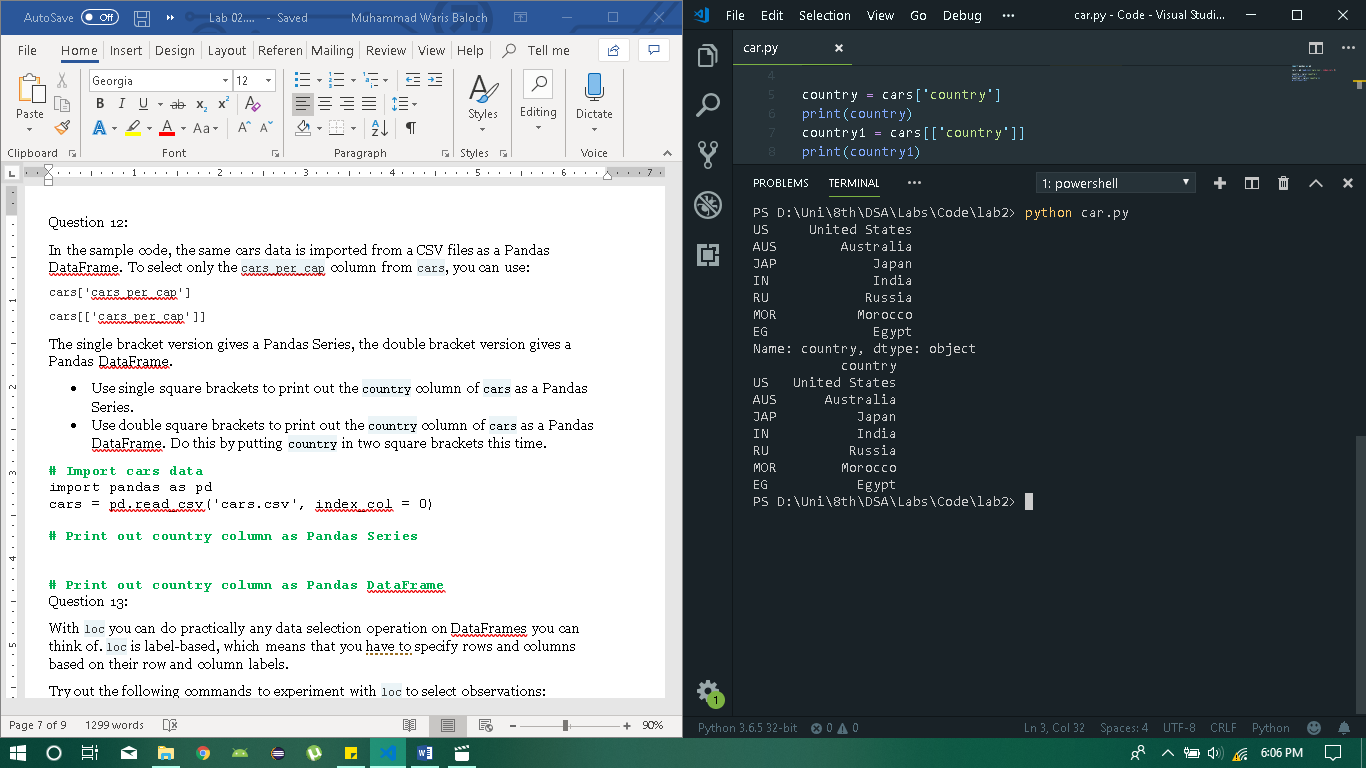
cars['cars\_per\_cap']

cars[['cars\_per\_cap']]

The single bracket version gives a Pandas Series, the double bracket version gives a Pandas DataFrame.

* Use single square brackets to print out the country column of cars as a Pandas Series.
* Use double square brackets to print out the country column of cars as a Pandas DataFrame. Do this by putting country in two square brackets this time.

Answer:



### Question: 13

With loc you can do practically any data selection operation on DataFrames you can think of. loc is label-based, which means that you have to specify rows and columns based on their row and column labels.

Try out the following commands to experiment with loc to select observations:

cars.loc['RU']

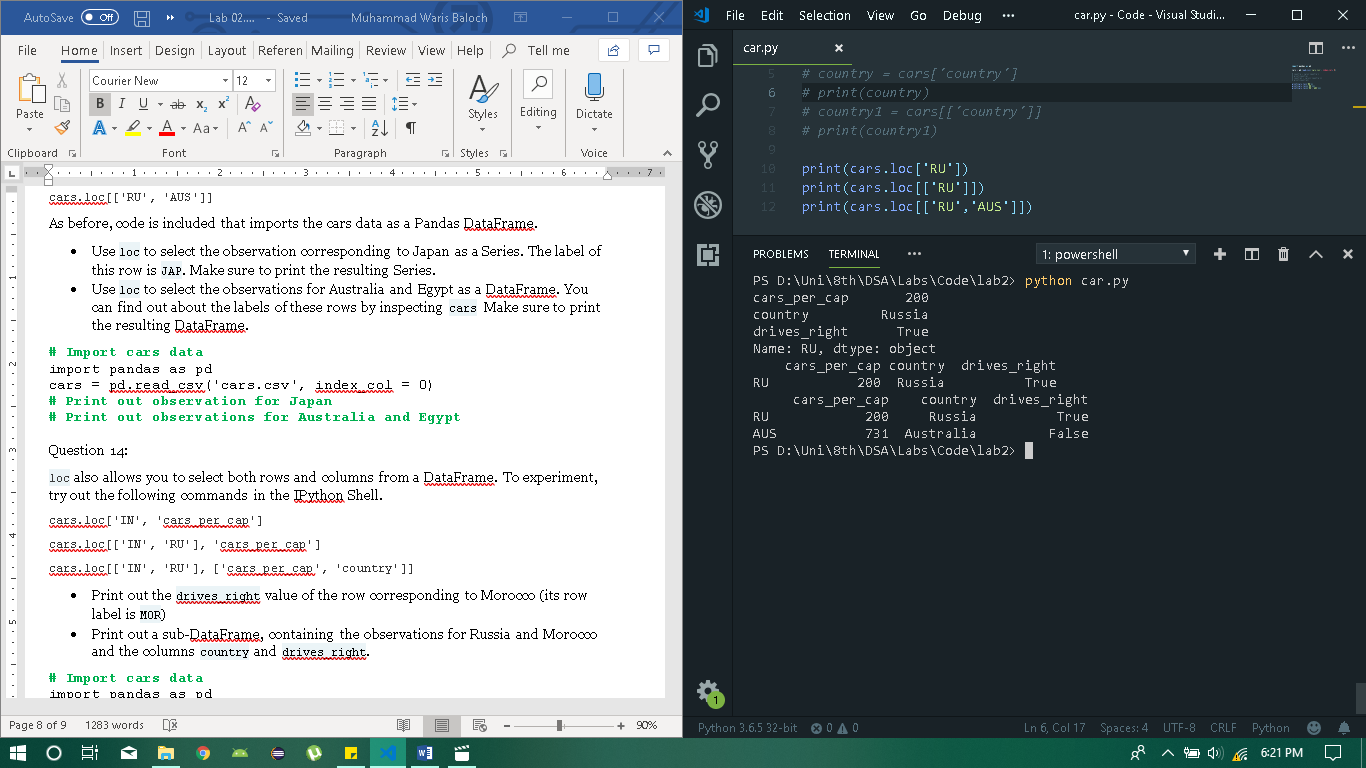
cars.loc[['RU']]

cars.loc[['RU', 'AUS']]

As before, code is included that imports the cars data as a Pandas DataFrame.

* Use loc to select the observation corresponding to Japan as a Series. The label of this row is JAP. Make sure to print the resulting Series.
* Use loc to select the observations for Australia and Egypt as a DataFrame. You can find out about the labels of these rows by inspecting cars  Make sure to print the resulting DataFrame.

Answer:



### Question: 14

loc also allows you to select both rows and columns from a DataFrame. To experiment, try out the following commands in the IPython Shell.

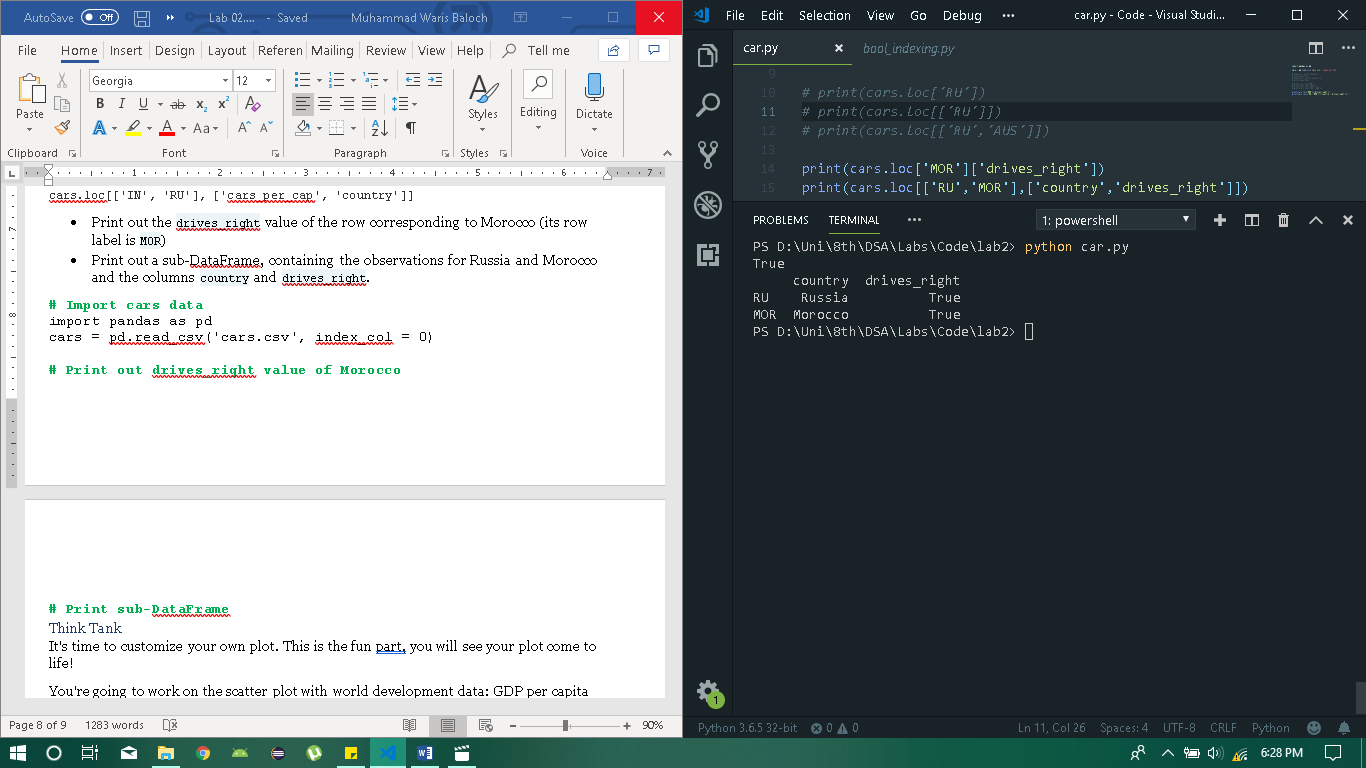
cars.loc['IN', 'cars\_per\_cap']

cars.loc[['IN', 'RU'], 'cars\_per\_cap']

cars.loc[['IN', 'RU'], ['cars\_per\_cap', 'country']]

* Print out the drives\_right value of the row corresponding to Morocco (its row label is MOR)
* Print out a sub-DataFrame, containing the observations for Russia and Morocco and the columns country and drives\_right.

Answer:



### Question: 15

* The strings xlab and ylab are already set for you. Use these variables to set the label of the x- and y-axis.
* The string title is also coded for you. Use it to add a title to the plot.
* After these customizations, finish the script with plt.show()to actually display the plot.

Answer:

