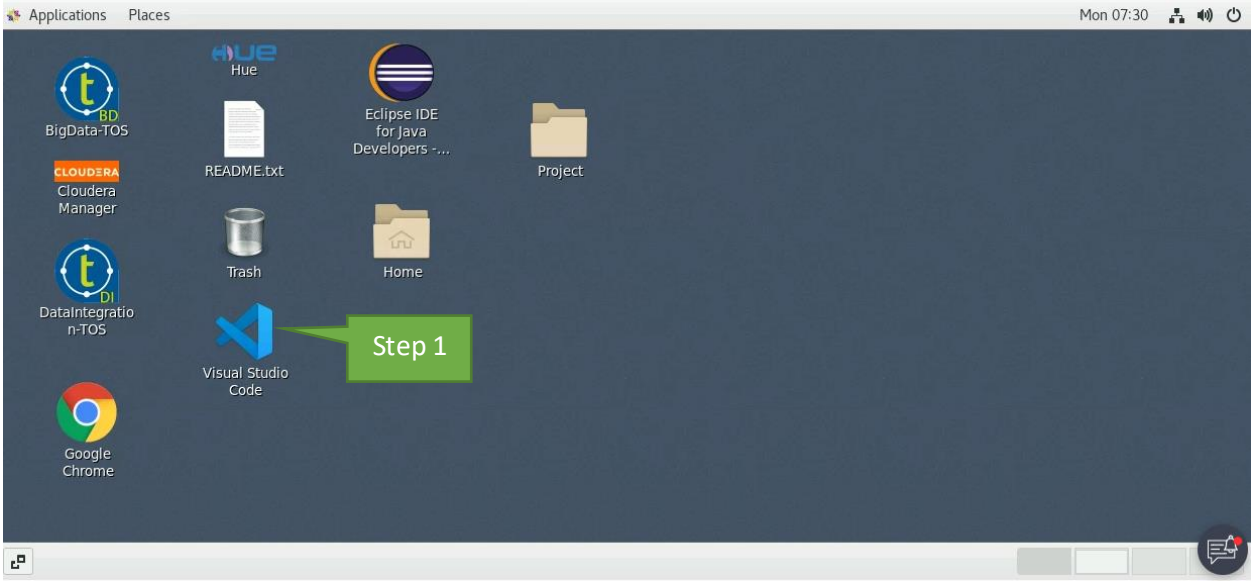
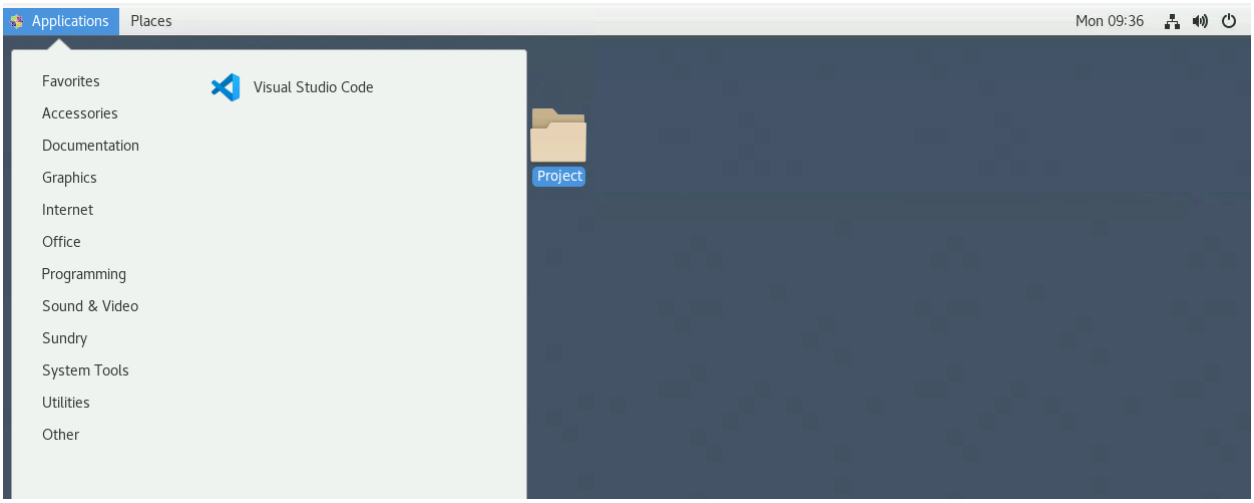
Guide to use VSCODE for PySpark development, sample test execution and making final submission of solution.

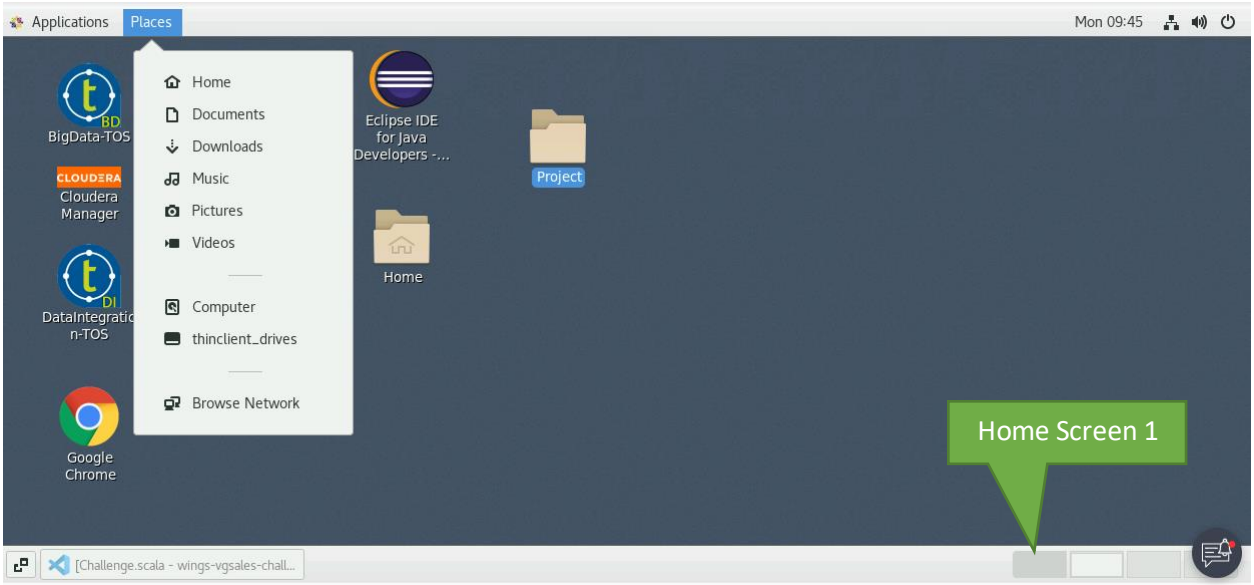
Step 1 – Launch Visual Studio Code from Desktop. Please give couple of minutes to launch to complete.

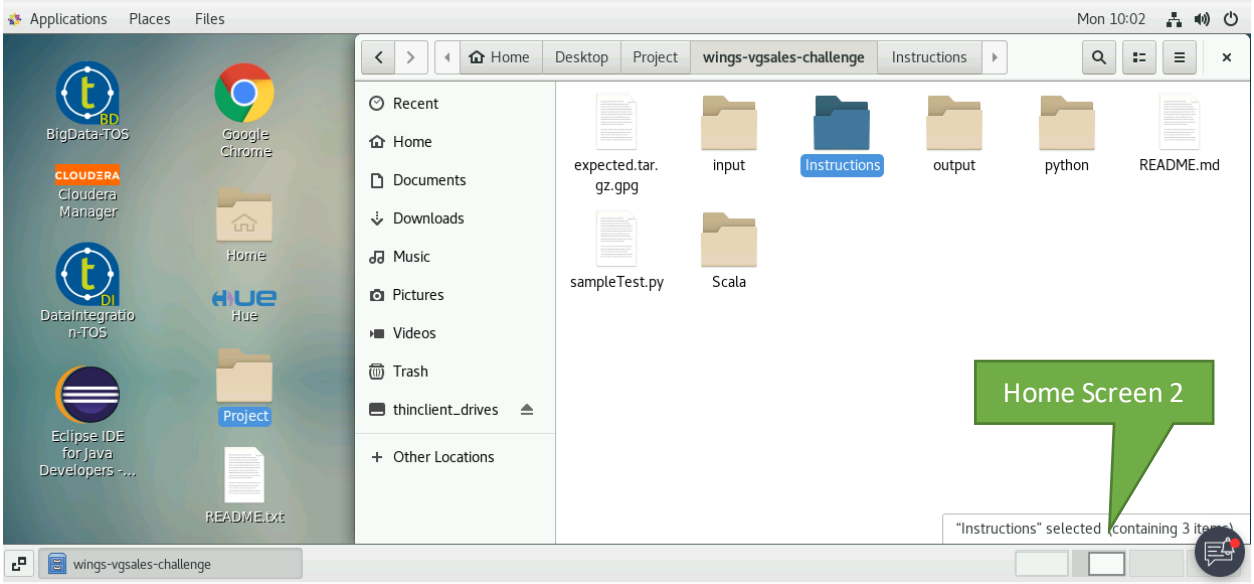


Alternatively, you can launch from “Application -> Programming” menu as shown below.

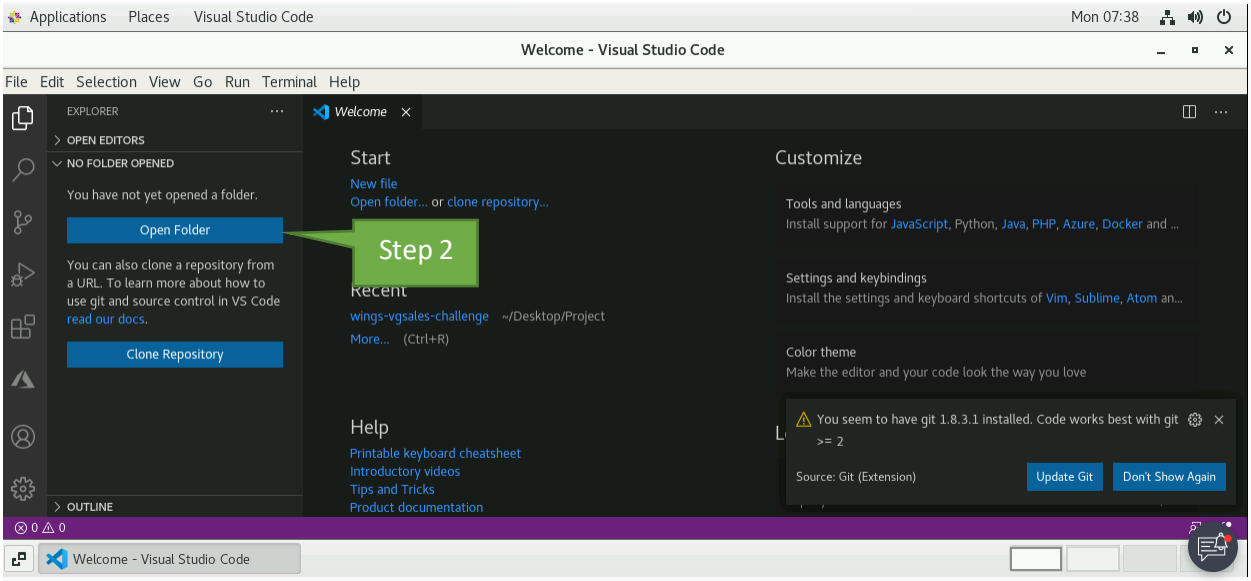


The “Places” menu provides navigational links to other directories. Usually, you have 4 home screens that you can choose as shown in bottom right corner. It may be useful to open VSCODE in one screen and project instructions in another

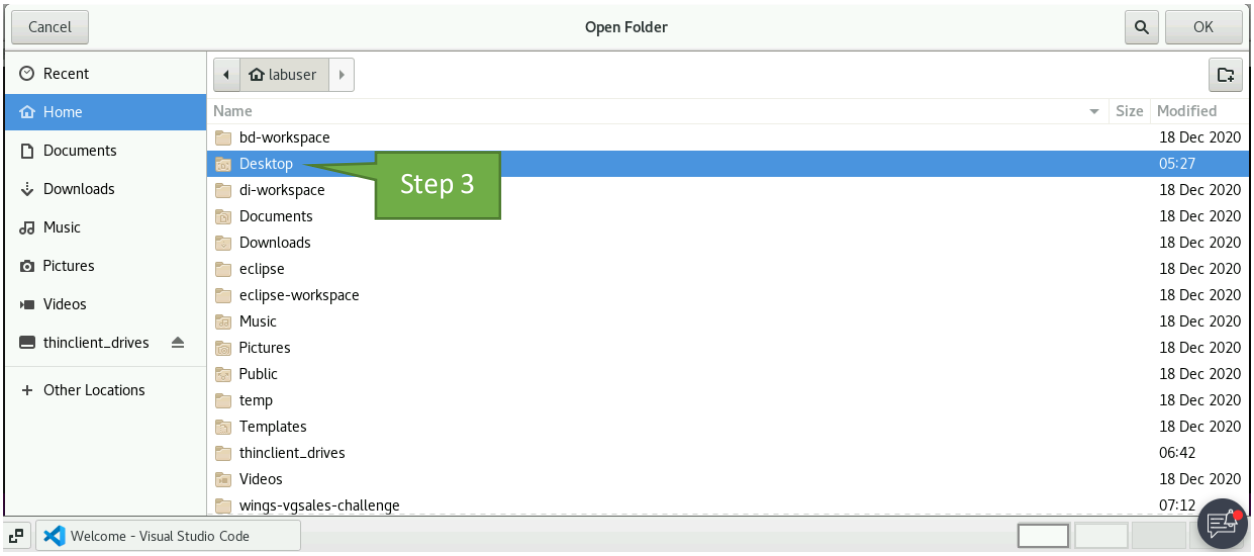




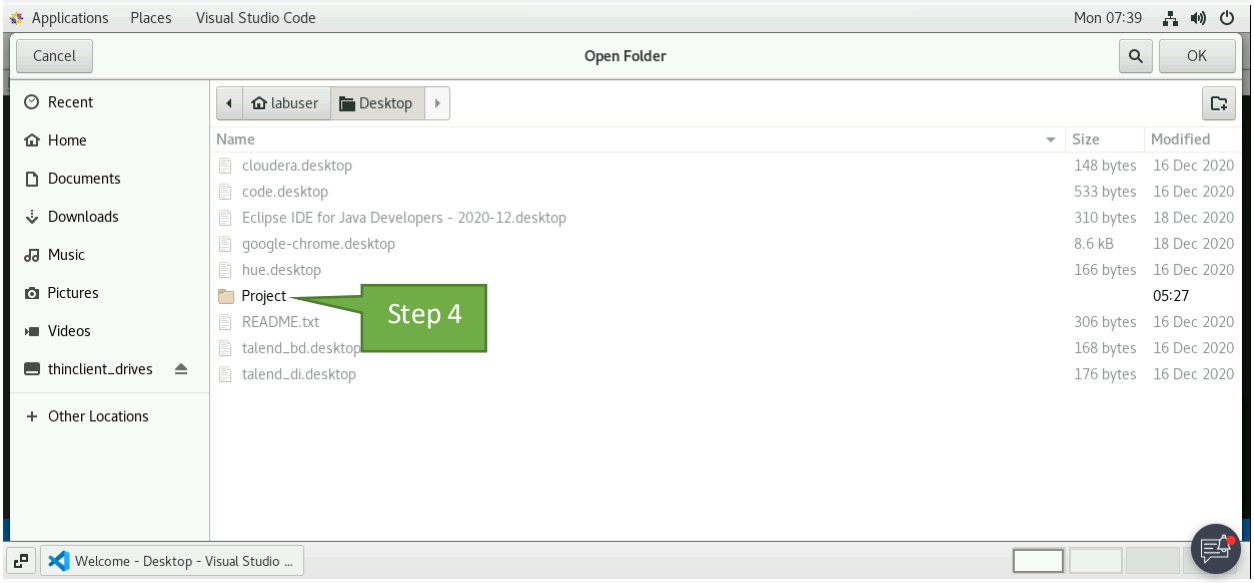
Step 2 – Click Open folder from VSCode home screen



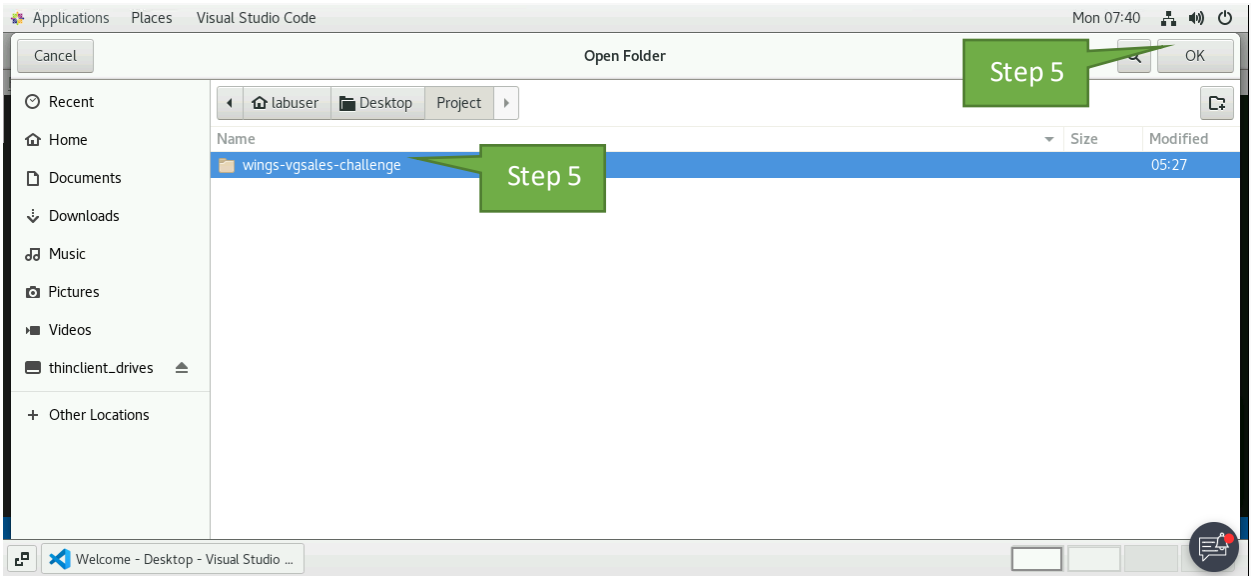
Step 3 – Traverse and double click Desktop folder



Step 4 – Traverse and double click Project folder

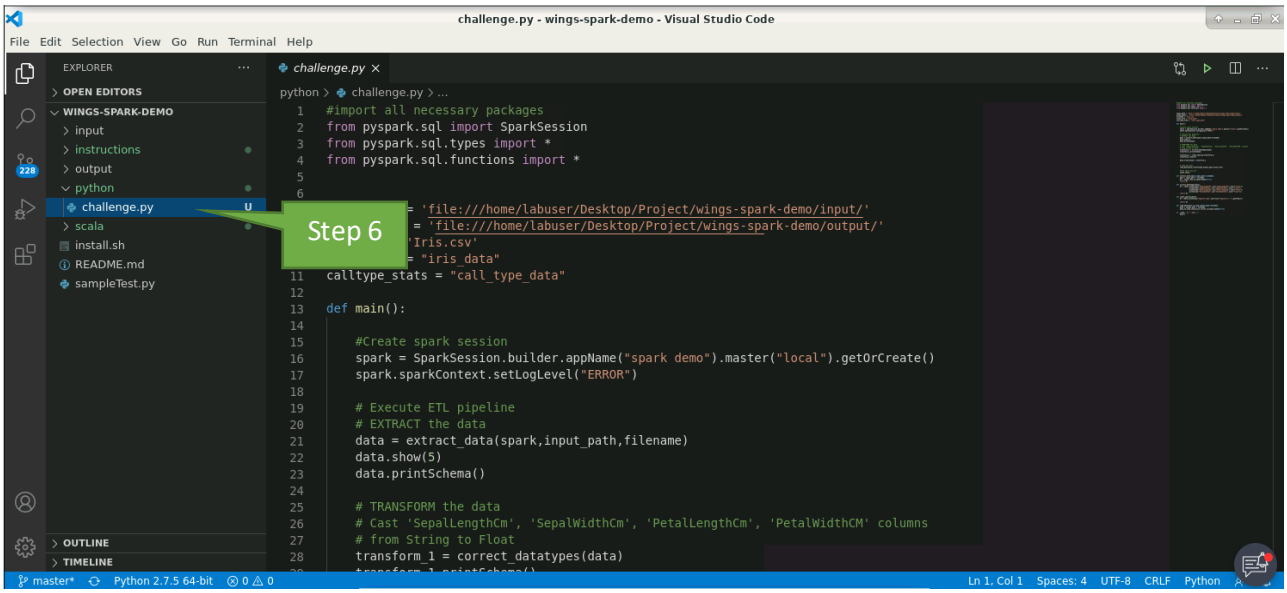


Step 5 – Click on the wings-vgsales-challenge folder and select "OK". **Note – This is just for instructional purpose. The folder name may vary during the exam.**

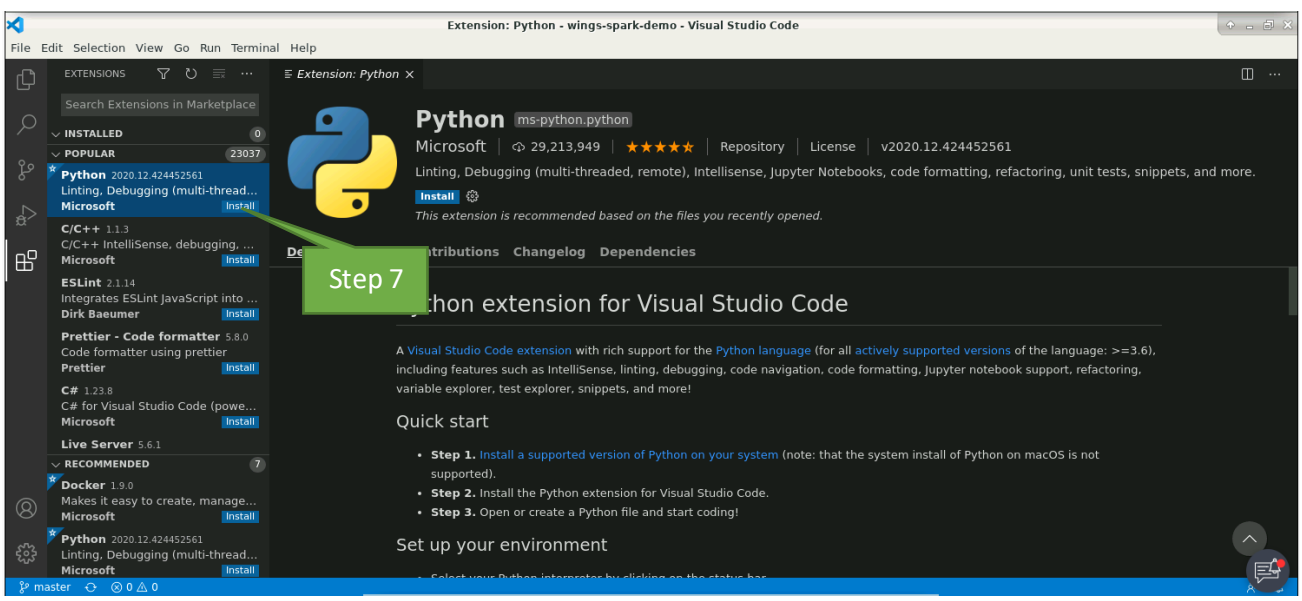


Step 6 – Open the “challenge.py” file within “python” folder

Note: This file contains code blocks which is left intentionally blank. But most of the program flow is already built for you. You need to fill in the code in the blank section where you are specifically asked to as per the problem statement.



Step 7 – Python extension may already be installed for you. If not, please install the extension as shown below.



Step 8 – As shown below (right) you are provided with a file with incomplete code block. You are required to read the problem statement from “Challenge.html” and find the function where you need to add your code in “challenge.py”. Complete the code as per the instructions. Example for Problem 1 is shown below.

Problem Statement from Challenge.html

Code implementation in challenge.py

Problem Statements:

PROBLEM 1:

Extract the input csv data from local storage into a dataframe.

Instructions:

Step 8

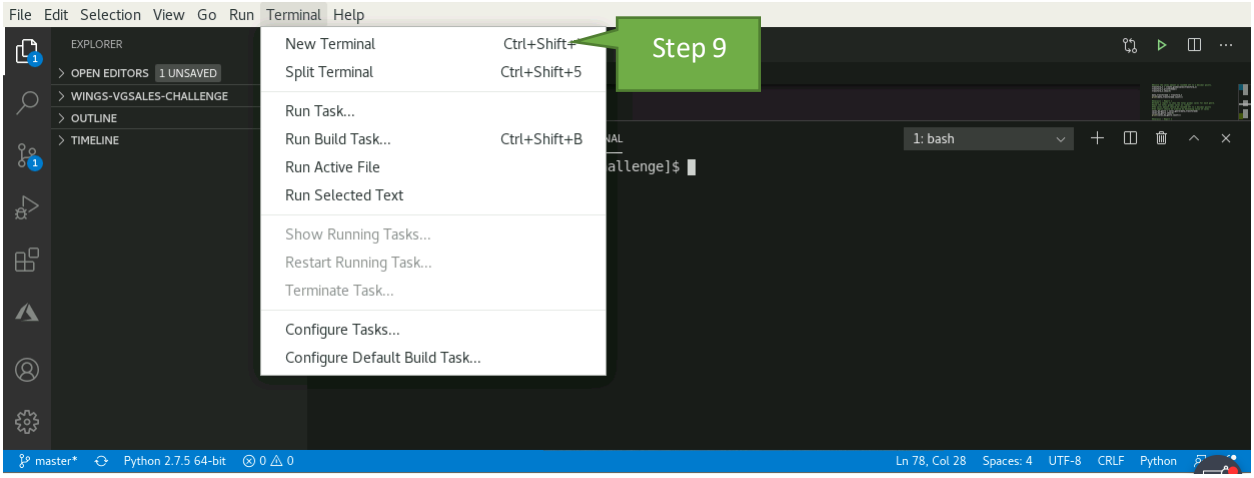
- Complete the function “extract_data” to solve this problem.
- Following details have already been defined for you in the program
  - Input File name: “vgsales.csv”
  - Input File path: “Project/wings-vgsales-challenge/input/”
  - Schema: It is already defined and stored as “schema”

python > challenge.py > ...

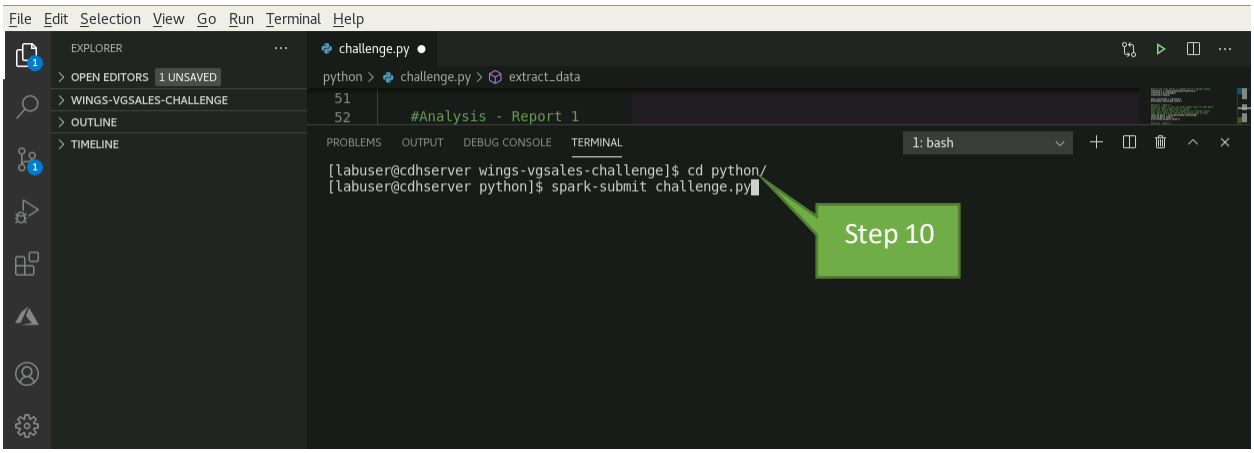
```
67
68
69      # LOAD the data
70      load_data(data_transformed,output_path,final_file)
71      load_data(sales_by_genre,output_path,report1)
72      load_data(sport_max_sales,output_path,report2)
73
74      #Stop spark session
75      spark.stop()
76
77  def extract_data(spark,input_path,filename):
78      #WRITE YOUR CODE BELOW THIS LINE
79
80      #WRITE YOUR CODE ABOVE THIS LINE
81      return df
82
```

Step 8

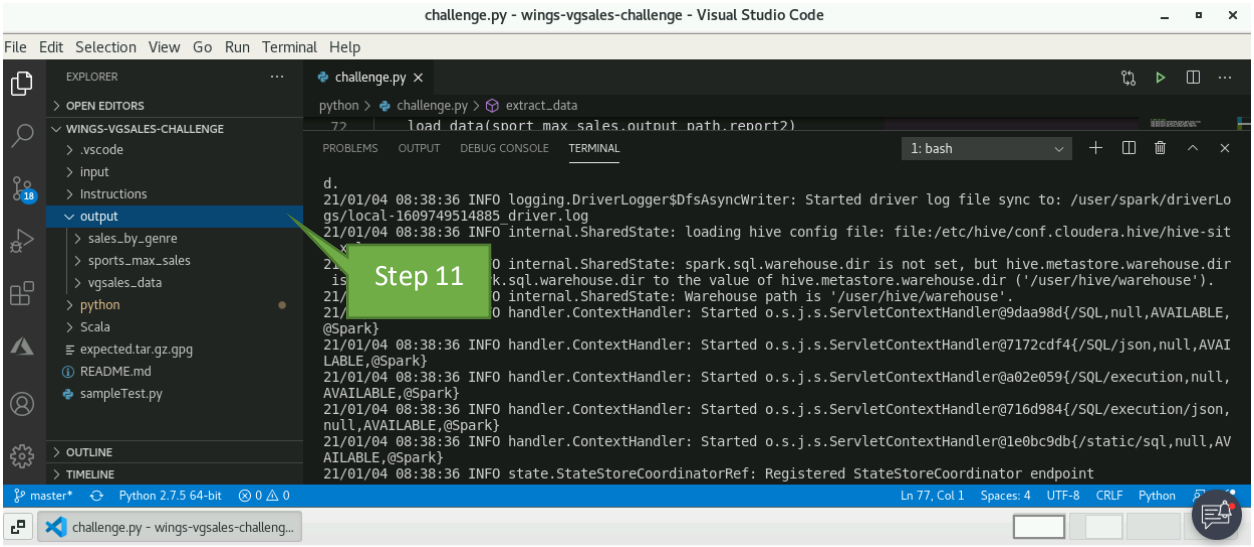
Step 9 – Once you complete your code and when ready to execute it, open a new terminal by clicking “Terminal -> New Terminal” as shown below.



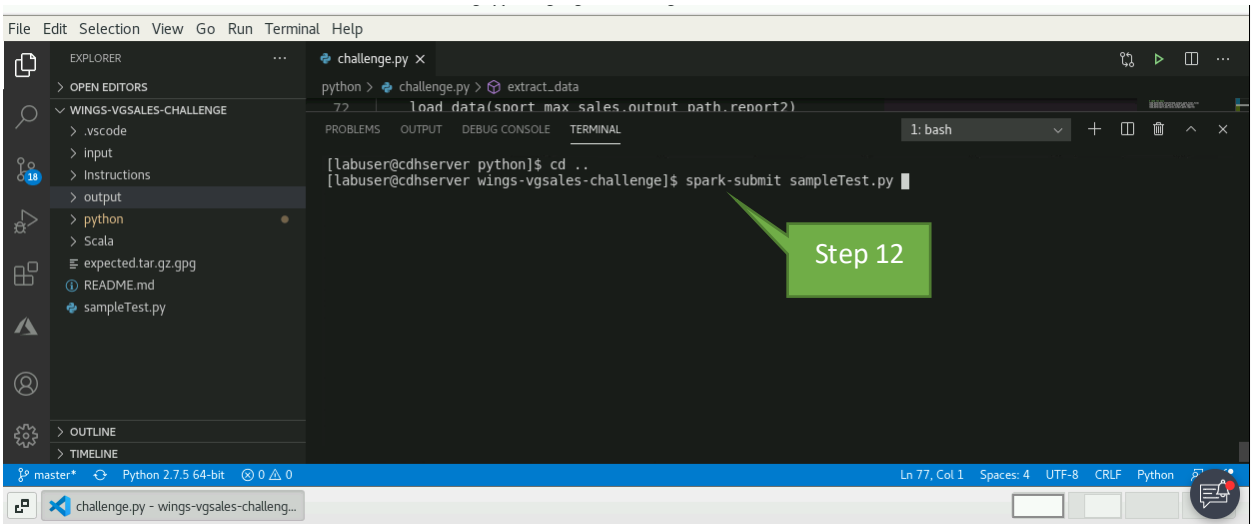
Step 10 – Make sure you are inside the “python” folder. Else navigate into “python” folder by executing command “cd python” in terminal. Once within the folder, submit the “challenge.py” file to cluster by executing the command “spark-submit challenge.py” in terminal.



Step 11 – On successful execution of the job, the required output files should be created under the “output” folder as shown below.

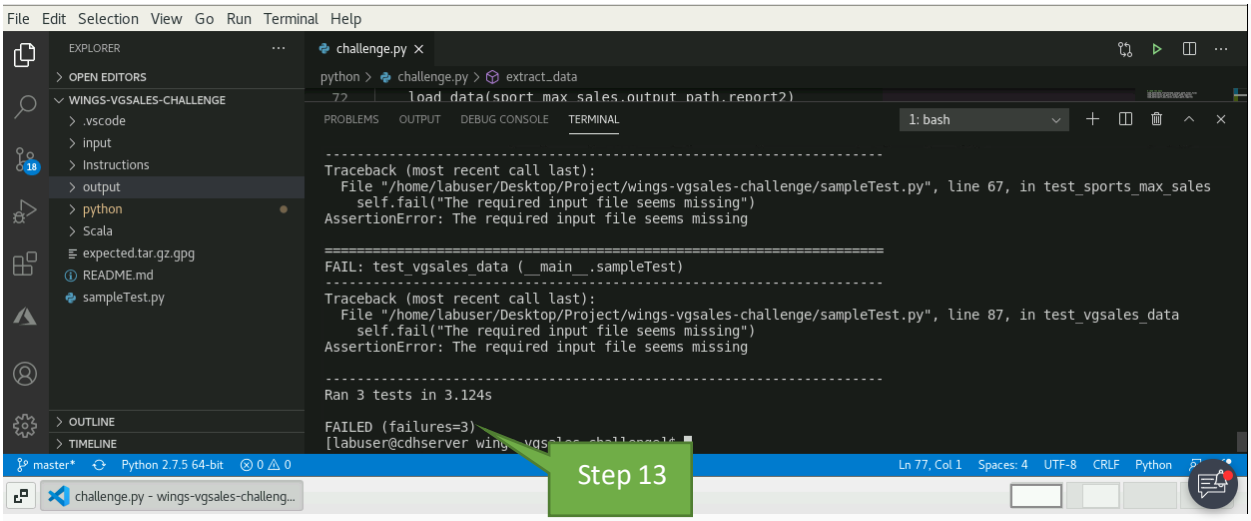


Step 12 – You have an option to execute sample test and confirm if you are on right track. **Note:** as name indicates these are just sample tests. Actual scoring will validate various aspects of your code and data. Make sure you are inside the challenge folder. Else navigate into that folder by executing command "cd .." in terminal (assuming you come here after Step 10, 11). Once within the folder, submit the sampleTest.py file to cluster by executing the command "spark-submit sampleTest.py" in terminal.

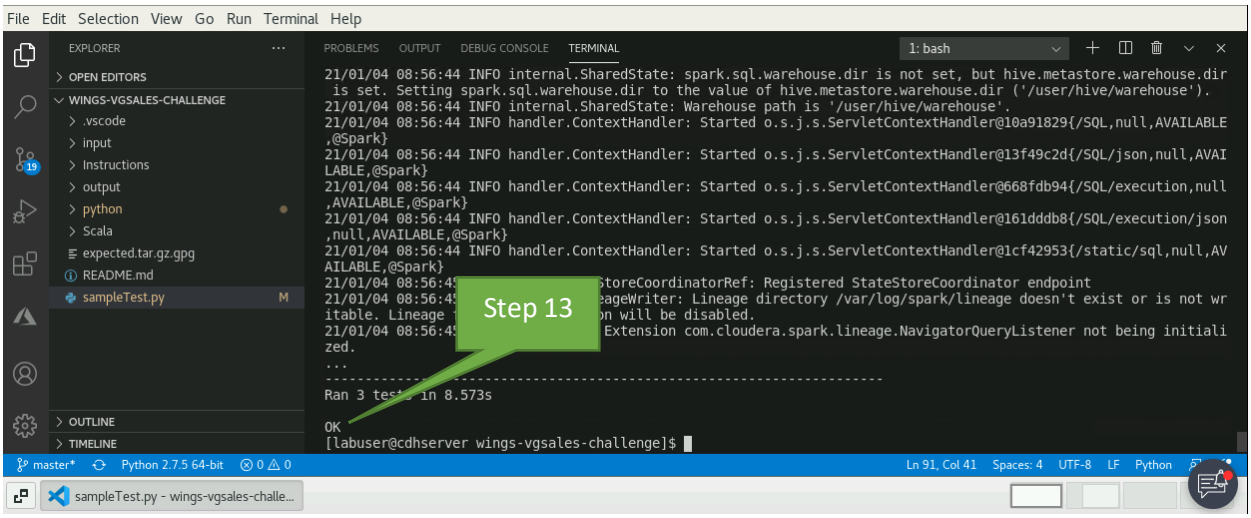


Step 13 – The results of execution is displayed in terminal. If any error found, make sure you revisit the code.

Example of tests failed



Example of tests passed



Step 14 – Once you are all set, you can submit your code by clicking "Submit" button on top right corner and then clicking "CONFIRM" as shown below. Once you submit, you may not be able to take the test again. Please note that at the end of the timer, the solution will be auto submitted for scoring if not manually submitted.