Software Testing Report

Victoria State Accident

Danny Thai (s5230918)

Sean Karl Angelo Enarbia (s5228570)

Table of Contents

[1.0 Unit Tests 3](#_Toc49779837)

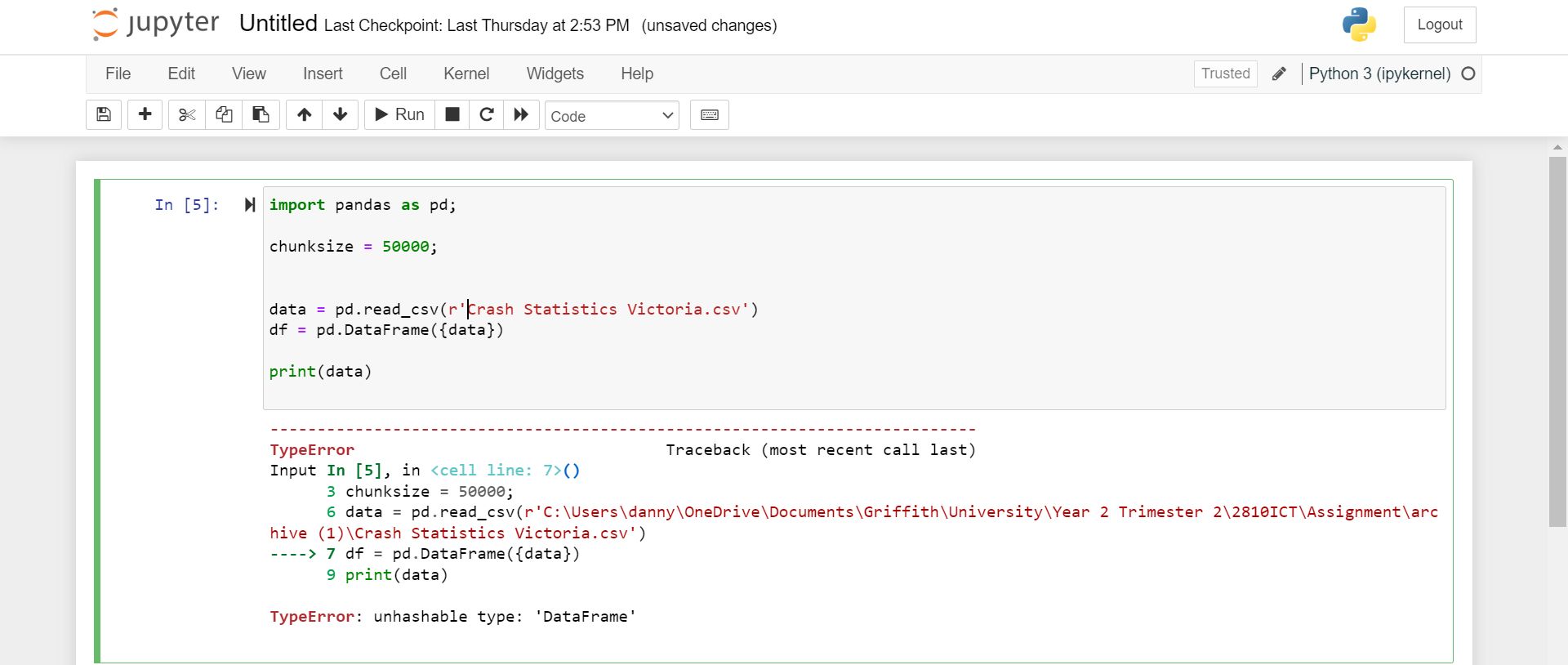
[2.0 Coverage Report 4](#_Toc49779838)

[3.0 Requirements Acceptance Testing 5](#_Toc49779839)

# Unit Tests

| **No** | **Test Case** | **Expected Results** | **Actual Results** |
| --- | --- | --- | --- |
| **1.0** | Entering prompt command to open the data. | The data of the csv should open somewhere in the internal data structure mainly in PyCharm with panda or Jupiter’s notebook. | Showing all the data of the csv in large table rows in Jupiter’s notebook or PyCharm running command prompt. |
| 1.1 | Testing to show all the data in csv in PyCharm or Jupiter’s notebook. | Allow the user to show the data in rows of tables including all the accidents, times, and types either in PyCharm running command or Jupiter’s notebook. | The csv file should be able to open finely either in PyCharm or Jupiter’s notebook showing the whole data. |
| 1.2 | Test to see if the csv can show results | Displaying all the whole data sets when running in both internal data structures. | Be able to read and display the csv file in the data structure. |
| **2.0** | Histogram Functions | Be able to display chart showing the percentages for the accident types in a chart for all days, times, and types. | Unable to show the chart due to error coding for the csv data for PyCharm and Jupiter’s Notebook. |
| 2.1 | Testing empty input dictionary | Displaying an error message prioritising to the incorrect input of coding in the data structure. | Displaying an error message and finding the incorrect codes and fix it. |
| **3.0** | Testing the code of the csv for the search bar | Able to show a search bar either at the top of the data structure for the users to type in and search. | The search bar for PyCharm would occur below when running and for Jupiter’s notebook to opening the file. |
| 3.1 | Testing to see the keywords in the filters for the search bar | The filters would be next to the search bar for the users to fill in for the expected search result. | The filters for PyCharm would occur below when running and similar for Jupiter’s notebook. |
| 3.2 | Testing the search bar for specific results | The search results should input the correct results when the user type in the search bar. | The results should show when PyCharm is running for the users to type in below and show the results. The same method works in Jupyter’s notebook. |
| **4.0** | Testing the code of the csv to show all crashes and fatal injuries | The csv should show the table rows for all crashes and fatal injuries. | The results have shown all crashes and fatal injuries for both PyCharm and Jupyter’s notebook. |
| 4.1 | Testing to see the percentages of the collision | The csv data are able show the percentages of the collision in table rows. | The results have shown all percentages collision for both PyCharm and Jupyter’s notebook when running in prompt or opening the file. |
| 4.2 | Testing to show tables rows for each accident types | The csv data are able show the percentages of the collision in table rows. | The results have shown all percentages in each table rows for both PyCharm and Jupyter’s notebook when running in prompt or opening the file. |
| **5.0** | Testing the filters for days, times and week | The filters next to the search bar should have the days, times and weeks before typing in the search bar. | Unable to implement due to additional coding for the csv in both PyCharm and Jupyter’s notebook. |
| **6.0** | Dataset Values | The csv data showing all the dataset values for the whole table is shown correctly. | The results have shown the dataset values are valid when shown in running or file opening. |
| **7.0** | Representing the dates for all crashes | The csv data are able show all the dates for all crashes next to their rows. | The results have shown all the dates for all crashes for both PyCharm and Jupyter’s notebook when running in prompt or opening the file. |
| **8.0** | Accidental Time Zones | The csv data are able the time zones when the accident has occurred. | The results have shown all the time zones when the accidental occurred for both PyCharm and Jupyter’s notebook when running in prompt or opening the file. |
| 8.1 | Testing to show alcohol times and percentages | The csv data are able show the percentages of the collision in table rows. | The results have shown the times and percentages for alcohol for both PyCharm and Jupyter’s notebook. |

# Coverage Report



For the coverage of our unit testing, we have decided to use Jupyter’s notebook along with pandas to conduct our testing of our software. For the first testing, the functions seem able to read the file finely while opening the csv file. However, there seems to be error code for opening the Data frame data for the Victorian State Accident, due to the file size largely with the table rows.

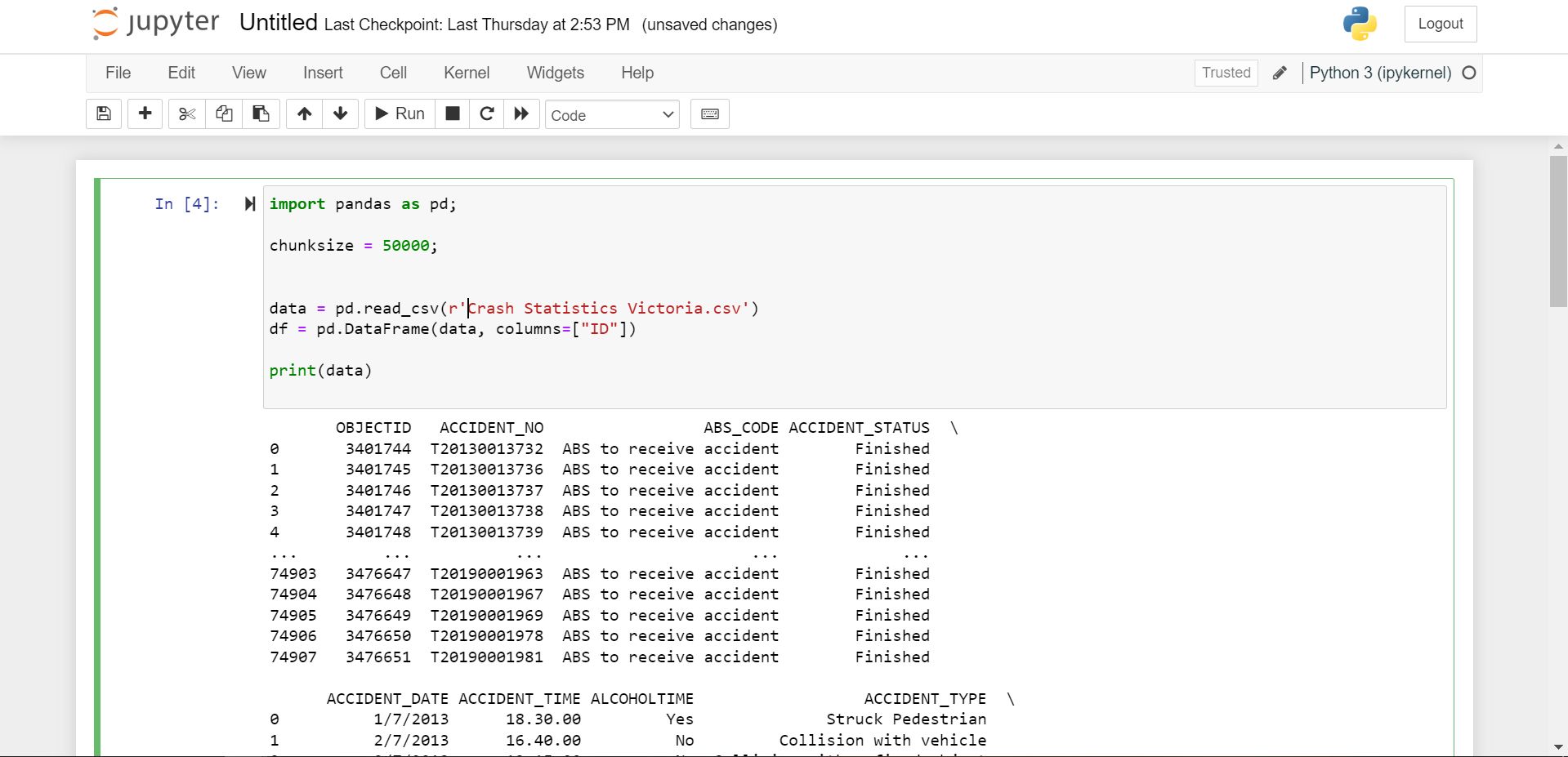
# Requirements Acceptance Testing

The second testing of the csv file seems to run very nicely. The function of the code is running well, the statement shows the user that the results of the incidents. The result shows the table rows of the incident showing Object id, Accident numbers, codes, and the status. The condition of this code section is clear and there is no error among.

Graphical user interface, text, application, email

Description automatically generated

The third testing we conduct is showing the data the data frame such as the names of each rows. The function seems to run perfectly as it represents the frames for the titles on top. The statement showing the code and printing the data frames is showing the results.



The fourth test we conduct is showing the data result for columns with ID printing for the result. The function of the code is running perfectly as the result is showing the ID for the Object. The statement is running the functions perfectly without error occurring and another would be that the results is showing the time zones such as the Dates, alcohol time and the types.

| **Software  Requirement No** | **Test** | **Implemented (Full /Partial/ None)** | **Test Results (Pass/ Fail)** | **Comments (for partial implementation or failed test results)** |
| --- | --- | --- | --- | --- |
| 1 | The program shall have the dataset and accepted as one in the command line. | Partial | Fail | Due to the size of the csv file, Jupyter’s notebook cannot handled largely size files. |
| 2 | The program shall have each name of the accident include the path of the file with the levels | Full | Pass |  |
| 3 | The program of the dataset will show the results for all crashes, fatalities, and injuries in a table row | Full | Pass |  |
| 4 | Displaying a message if the argument in the code is a directory instead of a different function | Partial | Pass | Largely size file cannot be loaded in Jupyter’s notebook |
| 5 | There will be a search bar for the users to enter the dataset for the Victorian State Accident | Full | Pass |  |
| 6 | Entering the search bar for the dataset will show the user the result of the accident | Full | Pass |  |
| 7 | For the users to search the specific results with the function to show all types of accidents | Full | Pass |  |
| 8 | The program also shows time zones for all accidents that occurred in the dataset | Full | Pass |  |
| 9 | The program should be able to accept as many levels for each search results with filters as the user wants to input. | None | Fail | We did not added the required code for this particular test when coding the data structure. |