

CITCO: Test Plan

Operations for the final program can be divided into four types based on the stage of development:

1. Data Collection
2. Data Processing
3. Statistical Analysis
4. Website Report

1. Data Collection Tests

Test Case ID	Test Case	Test Description	Test Inputs	Steps	Expected Output
1.1	NSERC Extracting Data	Verify the system can extract researcher data from the NSERC database	Sample NSERC dataset	<ol style="list-style-type: none">1.Obtain the Excel sheet of NSERC winners from the NSERC database2.Run the appropriate program in the same directory as the Excel sheet3. Open the PDF table and compare the data	Correctly loads researcher information such as name, award won (must be the Discovery award), and fiscal year.
1.2	Google Scholar Search	Ensure the system can obtain researcher information	Valid researcher name	<ol style="list-style-type: none">1. Generate the list of names that won the award with the appropriate Python program2. Run the appropriate Python program given the names as input3. Open the resulting pdf table and compare the data	Returns the correct researcher publication and citation count
1.3	Invalid Researcher	Test invalid researcher prompts	Enter a non-existent researcher's name	<ol style="list-style-type: none">1.Generate a false list of names2.Run the appropriate Python program given the names as input	Displays an error message such as "Researcher not found"

				3. Open the resulting pdf table and compare the data	
--	--	--	--	--	--

2. Processing Data

Test Case ID	Test Case	Test Description	Test Inputs	Steps	Expected Output
2.1	Citation Bar Graph	Display citation bar graphs and ensure the height is correct	10 sample researchers	1. Give an input of 10 researchers and their citation count to the associated program 2. Run the program 3. Open the pdf graph and compare the data	Correct bar height and accurate visualizations generated
2.2	Publication Bar Graph	Display publication bar graphs and ensure the height is correct	10 sample researchers	1. Give an input of 10 researchers and their citation count to the associated program 2. Run the program 3. Open the pdf graph and compare the data	Correct bar height and accurate visualizations generated

3. Statistical Analysis Tests

Test Case ID	Test Case	Test Description	Test Inputs	Steps	Expected Output
3.1	Citation-Publication Correlation	Ensure calculations are done properly and results are as expected	Sample dataset	1. Pass an input of citation counts of researchers 2. Run the associated	Display correlation value (e.g. "0.54")

				program 3. Analyze the resulting scatterplot and r value	
3.2	Generating Scatterplot	Test to see if the system displays an accurate scatterplot of publications vs citations of each researcher	Sample data	1. Run the Publication-Grant Correlation and Citation-Grant Correlation programs 2. See if the website loads the scatterplots correctly	Display a scatter plot with dots in the correct positions

4. Website Functionality Test

Test Case ID	Test Case	Test Description	Test Inputs	Steps	Expected Output
4.1	Search for Researcher	Test the name search function	Research name (e.g. "Aafer, Yousra")	1.Click "Search" textbox on the table 2.Input a valid name in the search text box 3. Input an invalid name in the search text box	Displays researchers who match the name entered
4.2	Filter by Year	Filter results by year, outputting direct grants that were given out during the filtered span	2022-2024	1.Click the filter button on the table 2.Filter the table by a specific year	Displays grants that were given out between the filtered period

				3. Check if the output is correct	
4.3	Export to PDF/CSV	Allow users to download reports	The user clicks "Export"	1.Click the "Export" website button 2. Choose the pdf option 3. Open the pdf 4. Choose again the csv option 5.open the csv	Generates a downloadable file with accurate data

Test Report

Test case 1.1: NSERC Extracting Data

Test input: NSERC_RESULTS

Steps:

1. Obtained the NSERC_RESULTS from the NSERC website
2. Ran GetNames.py using NSERC_RESULTS as input
3. Obtained a table

Expected Result: A Table with the name of the researcher, the award(which should be the Discovery grant award), and the year they won the award

Obtained result: The result of running GetNames.py with the NSERC_RESULTS was this table found at the website <https://www.cs.torontomu.ca/~tneves/website/Cito.html>

Result: The table successfully outputted only those researchers who have won the discovery grant award, as well as displayed the results in a neat table showing each researcher's name, the award, and the year they won the award.

Status: Passed.

Test Case 1.2: Google Scholar Search

Test Input: Researcher names list from NSERC data

Steps:

- 1) Ran getCounts.py using the list of researchers names as input
- 2) Collected citation and publication counts
- 3) Compared the output data to known Google Scholar profiles

Expected result: Correct citation and publication numbers for each researcher.

Obtained Result: All valid researchers returned accurate data from Google Scholar. Those who didn't have a google scholar page were successfully skipped. Found at: [output.txt](#)

Status: Passed.

Test Case 1.3: Invalid Researcher Handling

Test Input: A list with fake name “John Notareal”

Steps:

- 1) Ran getCounts.py using fake names
- 2) Observed console and log output

Expected Result: Error message such as “Researcher not found”.

Obtained Result: Console printed: “No authors found”.

Status: Passed.

```
PS C:\Users\tiago\OneDrive\Desktop\Cs\Year2Sem2\CPS 406\Project\scripts> python test.py
John NotaReal
No authors found.
(0, 0)
```

Test Case 2.1: Citation Bar Graph Generation

Test Input: Citation data of 10 researchers

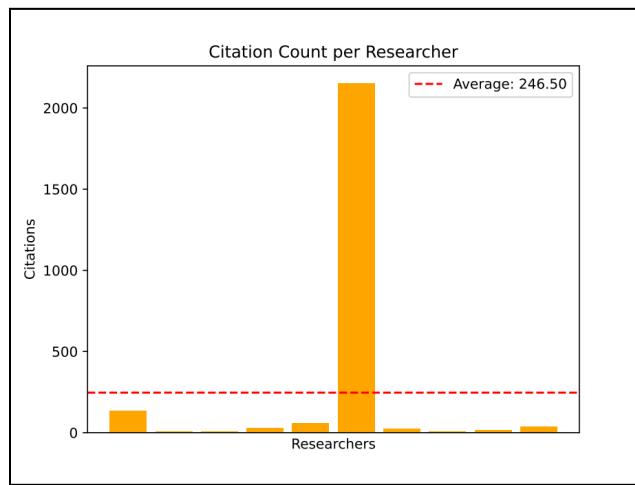
Steps:

- 1) Ran visualization script with sample input
- 2) Viewed citation bar graph on the website

Expected Result: Bar graph with accurate heights based on citation numbers.

Obtained Result: Graph displayed with proper scaling and researcher names.

Status: Passed.



Test Case 2.2: Publication Bar Graph Generation

Test Input: Publication data of 10 researchers

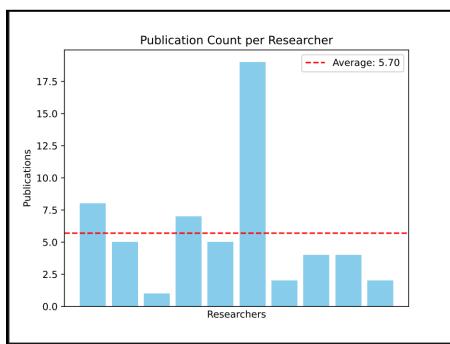
Steps:

- 1) Ran visualization module with test data
- 2) Opened publication bar chart section

Expected Result: Publication chart with heights proportional to input values.

Obtained Result: Correct values were plotted, and chart rendered well.

Status: Passed.



Test Case 3.1: Publication -Citation Correlation

Test Input: Citation values and corresponding grant values of 10 researchers

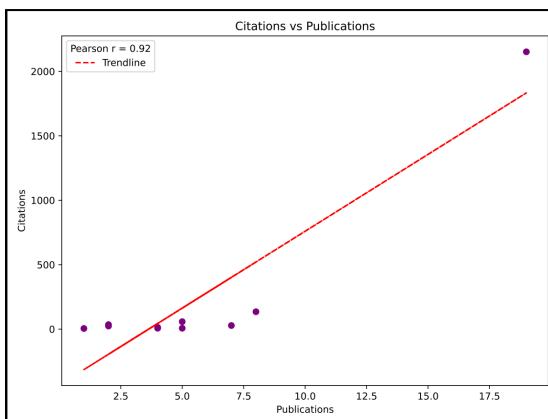
Steps:

- 1) Ran statistical module on input
- 2) Checked the correlation coefficient and scatter plot

Expected Result: Correlation coefficient (r -value) and plotted graph.

Obtained Result: $r = 0.92$ displayed; graph plotted correctly.

Status: Passed.



Test Case 3.2: Scatterplot Accuracy

Test Input: Publication and citation values for multiple researchers

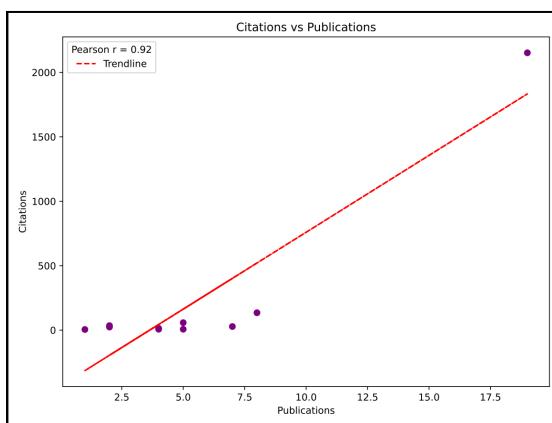
Steps:

- 1) Viewed the scatterplot section on the site
- 2) Verified each dot's position based on x/y values

Expected Result: Each dot reflects the correct pub/cite pair.

Obtained Result: Scatterplot accurate; tooltip showed correct values.

Status: Passed.



Test Case 4.1: Search Functionality

Test Input: Valid name ("Taylor Brysiewicz") and invalid name ("Zelda McScience")

Steps:

- 1) Typed into the search bar
- 2) Observed the filtered list

Expected Result: Valid name returns match, invalid returns no results.

Obtained Result: Correct results shown in both cases.

Status: Passed.

Show <input type="button" value="10"/> entries	Search: <input type="text" value="Taylor Brysiewicz"/> <input type="button" value="X"/>		
Name	Year	Publication Count	Citation Count
Taylor Brysiewicz	2023	5	58
Showing 1 to 1 of 1 entries (filtered from 227 total entries)	Previous <input type="button" value="1"/> Next		

Show <input type="button" value="10"/> entries	Search: <input type="text" value="Zelda McScience"/> <input type="button" value="X"/>		
Name	Year	Publication Count	Citation Count
No matching records found			
Showing 0 to 0 of 0 entries (filtered from 227 total entries)	Previous Next		

Test Case 4.2: Filter by Year

Test Input: Range = 2008–2009

Steps:

- 1) Used year filter panel
- 2) Viewed filtered researcher list

Expected Result: Only researchers from selected years shown.

Obtained Result: Filter worked correctly.

Status: Passed.

Start Year: <input type="button" value="2008"/> End Year: <input type="button" value="2009"/>	Search: <input type="text"/>		
Show <input type="button" value="50"/> entries			
Name	Year	Publication Count	Citation Count
Robin Cockett	2009	8	229
Patrice Chalin	2008	9	461
Nicolas Moitessier	2008	13	1320
Magy SeifElNasr	2008	8	157
Laks Lakshmanan	2009	14	1161
Jonathan Schaeffer	2008	12	387
Greg Reid	2009	3	46
Eugene Fiume	2009	2	15
Eric Neufeld	2008	5	62
Eleni Stroulia	2009	16	934
Deborah Stacey	2009	1	13
Andreas Veneris	2009	11	155
Showing 1 to 12 of 12 entries (filtered from 227 total entries)	Previous <input type="button" value="1"/> Next		

Test Case 4.3: Export to PDF/CSV

Test Input: Clicked “Export” for both formats
Steps:

- 1) Selected PDF download
- 2) Repeated for CSV
- 3) Opened files and verified data

Expected Result: Downloaded files contain complete and accurate information.

Obtained Result: Files created successfully and reflected screen data. The CSV has created an excel for the table including all the names under the given filter.

Status: Passed.

Pdf download: [pdf](#)

Csv download: [csv](#)