

# Lab 1: Carlpedia Project

## Objectives

- Get to know your group members
- Get to know Carlpedia
- Plan user tests to determine the usability of Carlpedia
- Develop a program in Python to statistically analyze and present the results of your user tests

## Introduction and background

### Introducing Carlpedia:

Doug Foxgrover and others have been designing a website, a wiki, to compile different kinds of technical support information, as well as other information important to different constituencies on campus. For example, a department may use the wiki to keep track of departmental tasks and deadlines; post information for students about the major, courses, and prerequisite structure; and to maintain various “howto” documents (how to apply for a job in the department, how to use specialized software, how to turn in your homework, etc.). The wiki is called [Carlpedia](#).

Currently the wiki structure consists of a “dashboard”, a page that links to the other areas of content. Each entity has its own content area; the home page for each content area is called a “landing page”. [Here](#) is the Carlpedia landing page, and [here](#) is the landing page for ITS. For this project, we will be focused on the landing pages, since they are the links to the true content of the site.

Doug has already explained a bit about Carlpedia’s structure and interface. For the rest of the lab, you will be familiarizing yourself with the abilities and limitations of the site, and will come up with a way to evaluate its usability.

## Your tasks

### Searching Carlpedia:

With your group members, spend some time exploring Carlpedia, starting at the landing page and/or the Dashboard. What elements of the website do you like or dislike? Why? You might want to try out the following searches that are typical of different user groups of the website:

1. Find out how to set up a printer on a Mac/Windows machine.
2. Find hours and email of the ITS help desk.
3. Where (physically) can you find help from ITS?
4. Where can you find the password-changing page?
5. Get help designing/printing a poster.
6. Find information about music software on campus.
7. What is CreativeSuite? What version does Carleton use?
8. Get help on diagramming text (Wordle).
9. Find information on virus protection available for computers at Carleton.
10. What software does Carleton have for analyzing data?
11. Get help making a chart or graph.
12. How do I connect to HOME?

You might also want to try browsing the tutorials and the search tips, too, as you explore the site.

If you are logged in to Carlpedia, you also have the ability to create and edit pages. Try creating your own page right now in your own space with some biographical information about yourself and your group members.

## Discussion Questions:

1. What kinds of questions could we ask about this website design?
2. Was there any information that you were not able to find?
3. What are some of the critical subcomponents of the tasks you performed on this website?
4. How could the readings from this week, especially today's reading, inform your analysis of this website?
5. Pick 2 questions of interest. What kinds of methods could be used to investigate these questions?

## Study Design:

For the next week, you and your groups will be designing a research study and collecting some data to be analyzed next week in lab.

1. Pick a research question. This question should be relatively narrow in scope. Consider the task you want to evaluate, the user group you are interested in, and the kind of evaluation you want to do (emotional, efficiency). Your study design should involve at least one comparison (e.g., between two user groups, between two user tasks, between two different survey questions measured on the same scale).
2. Pick a research method. What data are you going to collect? (e.g., preference surveys, emotional surveys, task completion, time to complete a task, think aloud protocols, structured interviews)
3. What tasks will you have your participants do? Design your user tasks and any survey instruments that you want to use.
4. How will you collect your data? (e.g., a tape recorder, a Google survey, a stop watch)
5. How many participants will you evaluate?
6. How will you analyze your results (quantitatively and/or qualitatively)?

## Pros and Cons of different methods

- *Think aloud protocols and structured interviews.* You need to select your tasks and comparison groups carefully. You will need to have a way to code and quantify the feedback that you get from participant. The information you get may be fuzzy and hard to generalize to a variety of different people and situations.
- *Survey data.* You will need to select your tasks carefully. You will need to write a survey that will give you information that will help you answer your question, but be sure to watch out for different kinds of biases, either in how you ask your questions or in how your participants answer your questions. The information you get may seem restricted, it may imply future questions to answer, surveys are often subject to different types of bias.
- *Task completion times and accuracy.* You need to select your tasks carefully, and you may need a relatively large number of tasks. You don't get feedback on the participants' subjective experience.

## Homework

### Writing

Once you have your study design, you and your group need to write an American Psychological Association (APA) style introduction and method section. These papers have an explicit structure, which is outlined below, and they use a specific citation and reference style, which we will discuss a bit on Monday. See the sample APA style paper on Moodle for a sample, with further notes on APA style.

- The **introduction** lays out the background necessary for the reader to understand why the study is being conducted. This may include background research done by other researchers, theoretical principles that might direct the research approach, and general information that informs the reader about the research's importance. Remember that this section of the paper is a rhetorical, thesis-driven argument, and as such, it should have a thesis and solid paragraph structure.
- The **method** section tells the reader how the study is conducted. It gives explicit detail about who your participants are (in a section labelled Participants) and what your participants are doing (in sections labeled Materials and Procedure).
- The **results** section gives the reader an analysis of your findings, including relevant qualitative and quantitative data.
- The **discussion** section reviews the data in the framework of the background research presented in the introduction, including a discussion of any conclusions that can be drawn and any important implications for the research field or society at large.

Your group should create a GoogleDoc for this paper. (Be sure that it is shared with all group members!)

***Your group must submit a rough draft of your method section to Mija for approval before you can collect data!*** (Monday or Tuesday might be a good day to shoot for.)

The finished introduction and method should be submitted on Moodle by next Friday's class.

## **Data Collection**

In the next week, after you have written your method section, emailed it to Mija, and received approval to go ahead, you need to collect data from participants, as specified in your method. Enter your numerical data into a spreadsheet in GoogleDocs that you share with your partner. You need to bring your data to class next Friday.

## Carlpeia Paper Evaluation

*Title and Abstract* \_\_\_\_\_/10 pts

- Title establishes reader interest in the paper.
- Abstract summarizes the main findings of the paper.

*Introduction* \_\_\_\_\_/20 pts

- The section establishes the research question,
- Grounds the motivation for investigating the research question, and
- Outlines the background literature that pertains to the research question.

*Method* \_\_\_\_\_/10 pts

- The research design fits the research question.
- The research design is explained clearly and concisely,
- Making use of APA style subsections.

*Results* \_\_\_\_\_/10 pts

- Relevant analyses are selected,
- Conducted correctly,
- Reported thoroughly, and
- Interpreted reasonably.

*Discussion* \_\_\_\_\_/20 pts

- Study results are summarized clearly.
- Limitations of the study results are discussed.
- Implications, in the form of design recommendations, are suggested and supported.

*Organization* \_\_\_\_\_/20 pts

- The paper has a clear thesis that addresses the research question.
- Paragraphs have topic sentences that support the thesis,
- With effective transitions between ideas within paragraphs and between paragraphs.
- Published evidence (from readings and study) is used effectively to support the thesis.
- Paper makes use of APA style sections.

*Editing* \_\_\_\_\_

- No more than one or two errors in grammar, punctuation or spelling

Overall grade: \_\_\_\_\_/100 pts

## Analysis Program

For next Friday, write a program to analyze the results from your user tests and present the results clearly and concisely to the user. You will perform basic statistical tests on your data.

Your program should do the following:

- Calculate the mean of your population data
- Calculate the standard deviation of your population data
- Calculate what percentage of your data points fall within one standard deviation of the mean

Each data point should be stored as a separate variable. In addition, your program should print out the calculated values, clearly labeled, to the screen.

### Hints:

- Don't worry too much about number formatting at this point. Your results will likely be printed out to more decimal points than necessary. When we discuss strings in a few weeks, we'll learn better ways to present numerical data. At this point in the term, it's more important that your calculations are correct.
- You will be running your user tests this week as well, and may not have all of your results until later in the week. This does not mean that you have to wait until you have all of your responses before writing your program! One trick that programmers use is to use "dummy data" to test that a program is working, when access to the actual data is not possible. You can then easily hand-check your calculations as well.
- The best, most efficient, most clever program in the world is useless if the user cannot fathom what it's doing. A big part of helping the user understand your program is to think carefully about the program's input and output. Are you soliciting data from the user in a way that's clear what you expect from the user? Are there specific ranges your program expects for these values? Similarly, are you presenting the results in a way that's clear for the user to understand? Are any numerical values clearly labeled, with units where appropriate? Do you clearly indicate the purpose of the program at the outset?
- This is also a good time to think about the structure and readability of your program. Just like you divide your essays into paragraphs and sections, so too should you use whitespace and comments to logically divide the sections of your program. Comments are like section headers; variable names are like descriptive nouns.

Turn in your program on Moodle on the CS page.

## Programming Rubric

Your program will be evaluated on the following criteria:

- **Execution:**
  - Program runs to completion with no errors
- **Correctness:**
  - Program performs the expected/required analyses
  - Program utilizes all data appropriately
  - Program performs the correct calculations on the data
  - Results generated by the program are correct (no math or logic errors)
- **Python elements:**
  - Variables: appropriate number, appropriate use of assignment
  - Calculations (math library)
  - Input/output (input, print)
  - Loops
    - Correct number of iterations
    - Correct syntax for setting up the loop and loop counter
    - Appropriate statements inside the loop
    - Correct termination condition(s)
  - Conditionals
    - Appropriate condition (boolean) statement(s)
    - Appropriate number of branches, including default condition where appropriate
    - Correct syntax
- **Style:**
  - Variable names clearly specify the data stored
  - Comments clearly indicate what the program is doing
  - Comments use the correct format/style
  - Whitespace logically separates the different tasks of the program, to aid readability
  - (if functions used) Functions are clearly separate from “main” code
- **User-centric design:**
  - Results are clearly labeled (e.g., “average response time (s): “)
  - Results are presented in appropriate format (integer vs. float)
  - Program states what it is doing clearly and concisely up front
  - Program prompts for input by presenting clear instructions to the user
  - Displayed text is spaced appropriately and is formatted in a way that is easy to read and follow