Data Analysis Readme

Alkis Antoniades
alkis.antoniades56@gmail.com

March 30, 2022

This document aims to provide information as to how one may reproduce the results presented in Chapter 4 of my MSc thesis project report, titled: "Everyday locations as cues to smoke" [1]. This and the data analysis files have been peer reviewed using a relevant checklist that can be viewed in the "Data Repository Review.pdf" document located in the "Accompanying Documents" folder. This folder also contains knitted pdf files outputted after running the analysis code, and you may generate your own by following the instructions in this document.

For the ipq_data.sav the following notices apply:

> "You can download the IPQ Data set provided that you agree with the following conditions:
>
> 1. You may not change these data in any way.
> 2. You may make and give away verbatim copies of the data without restriction, provided that you duplicate all of the original copyright notices.
> 3. Publications of analyses of these data must reference the copyright holders.
> 4. You may charge a reasonable copying fee for any distribution of these data. You may not charge a fee for this data itself."

If you do not agree with the license information and the specific notices for the ipq_data.sav, please do not download the files, and if you are reading this after having downloaded them, please delete them.

Contents

1 Data Files Description **3**
1.1 Raw Data Folder 3

1.1.1 VR System Data − > [N]_responses.csv 3
1.1.2 Qualtrics Data − > qualtrics_responses.csv 3
1.1.3 IPQ Database Data − > ipq_data.sav 3
1.2 Data Folder 3
1.2.1 auq_data.csv 4
1.2.2 auq_scores_data.csv 4
1.2.3 cbuq_data.csv 4
1.2.4 experience_familiarity_data.csv 4
1.2.5 sop_data.csv 4
1.2.6 ipq_DB_means.csv 5

2 Script Files Description **5**
2.1 Data_Pre_Processing.ipynb **and** Data_Pre_Processing.py 5
2.2 Experience_Familiarity_Analysis.Rmd 5
2.3 Sense_of_Presence_and_Interface_Usability.Rmd 6
2.4 Prior_Sensitivity_Analysis.Rmd 6
2.5 Other Files 6
2.5.1 results_output.txt 6

3 How to Run the Code **7**
3.1 Python for Pre-Processing 7
3.1.1 Using Only Python 7
3.1.2 Using Python Through Jupyter Lab 7
3.2 R in Docker for Analysis 9
3.2.1 Running R Code in Docker 9

1 Data Files Description

The sections below provide a high level description of the data contained in each file mentioned.

1.1 Raw Data Folder

Folder containing un-processed data files described below. This is to be used in the (optional) pre-processing step using the *Data_Pre_Processing.ipynb* script described in Section 2.1 to output data files for use in the data analysis scripts described in Sections 2.2 and 2.3. The folders here also contain files with the data file column descriptions mentioned below.

1.1.1 VR System Data $->$ [N]_responses.csv

Data obtained via questionnaires administered by our VR system. Each file contained in this folder corresponds to responses of each user, with their given identifier as part of the file name.

Columns are described in the *VR System Data Column Descriptions.xlsx* file in the same folder.

1.1.2 Qualtrics Data $->$ qualtrics_responses.csv

Data obtained via a Qualtrics questionnaire administered at the end of the VR experiment.

Row 2 presents the question participants were asked to respond to.

Columns are described in *qualtrics_responses - Column Descriptions.xlsx* file in the same folder.

1.1.3 IPQ Database Data $->$ ipq_data.sav

File provided by the IGROUP [2], containing Sense of Presence data of 542 research cases. A description of this dataset may be viewed on their website at `http://www.igroup.org/pq/ipq/data.php`. This is used by the *Data_Pre_Processing.ipynb* script to output mean scores for comparison against the scores we obtained. You may open it in SPSS or view its contents using the Pandas package in Python, as shown in the Data_Pre_Processing.ipynb script.

1.2 Data Folder

This folder contains the data files outputted by the *Data_Pre_Processing.ipynb* script using files in the Raw Data folder.

1.2.1 auq_data.csv

File containing processed data obtained by the Additional Usability Questionnaire, including the Ease of Material Collection question. This file also includes the free-text responses provided as the reasoning for each given score.

Columns correspond to the question numbers, and the PID corresponds to the participant identifier used in referencing their free-text responses in Section 4.3.2.1 of the thesis report.

1.2.2 auq_scores_data.csv

File containing processed data obtained by the Additional Usability Questionnaire, including the Ease of Material Collection question. This file only contains the numerical score.

Columns correspond to the question numbers.

1.2.3 cbuq_data.csv

File containing processed data obtained by the Component-Based Usability Questionnaire (BRINKMAN, W.-P., HAAKMA, R., AND BOUWHUIS, D. G. The theoretical foundation and validity of a component-based usability questionnaire. Behaviour & Information Technology 28, 2 (2009), 121–137.). The column corresponds to the overall mean score obtained from each participant.

1.2.4 experience_familiarity_data.csv

File containing processed data obtained via questionnaires administered by our VR system.

Columns correspond to:

1. subject: Participant identifier
2. env_pers: Whether the environment presented was personalized (1) or not (0).
3. env_name: The name of the environment in which the given score was reported.
4. experience_familiarity: The experience familiarity score given.
5. sop_[x]: The overall sense of presence reported by the given participant.

1.2.5 sop_data.csv

File containing the obtained mean scores for each subscale of the presence questionnaire.

Columns correspond to:

1. PRES_Score: General presence subscale score.
2. SP_Score: Spatial presence subscale score.
3. INV_Score: Involvement subscale score.
4. REAL_Score: Experienced realism subscale score.

1.2.6 ipq_DB_means.csv

File containing the mean sense of presence subscale scores of 542 research cases provided by the IGROUP [2].

2 Script Files Description

The sections below provide a high level description of the operations performed by each of the three code files mentioned.

2.1 Data_Pre_Processing.ipynb and Data_Pre_Processing.py

Python scripts that load files from the Raw Data folder, processes the data contained as mentioned in Section 4.1.6.1 of the thesis report, and outputs data files to be used by the two analysis code files described in the following two sections. Additionally, these scripts calculates Chronbach's alpha for relevant data, as well as norm means for the IGROUP sense of presence questionnaire [2], both of which are outputted in the "results_output.txt" file. The aforementioned norm means are used for the purpose of comparing our sense of presence analysis results to compare how our system performs against results contained in the IGROUP database provided.

The pre-processing step has already been performed and the resulting data files can be accessed in the Data folder, however, you may re-run this code yourself to output the files, if you so prefer using the instructions in Section 3. The two files contain the same code, with the difference being that you may find viewing the .ipynb file more comfortable, and it also offers knitting a pdf file of the entire code run.

2.2 Experience_Familiarity_Analysis.Rmd

R script that loads data files from the Data folder and performs the analyses described in Sections 4.1.6.2 and 4.1.6.5, to produce the results reported in Section 4.2.1 and Table 4.4 in Section 4.2.3.2 of the thesis report.

Succinctly, performs analysis and obtains results as to:

1. Whether personalization affects experience familiarity elicited.
2. Whether the different sense of presence questionnaire subscales affect experience familiarity elicited.

2.3 Sense_of_Presence_and_Interface_Usability.Rmd

R script that loads data files from the Data folder and performs the analyses described in Sections 4.1.6.3 and 4.1.6.4, to produce the results reported in Sections 4.2.2, and Table 4.3 in Section 4.2.3.2 of the thesis report.

Succinctly, performs analysis and obtains results as to examine the:

1. Sense of presence elicited by our system.
2. Component-based usability questionnaire scores.
3. Additional usability question scores.

2.4 Prior_Sensitivity_Analysis.Rmd

R script that loads data files from the Data folder and performs sensitivity analysis on the priors of models used in the above two R scripts.

2.5 Other Files

2.5.1 results_output.txt

File containing results of Chronbach's alpha for the sense of presence and component-based usability questionnaires, as well as the mean scores of the IGROUP database.

3 How to Run the Code

Important Note: There is very high probability that errors may occur when using the commands provided below, due to incorrect symbols being copied from the file, or commands being copied with additional characters. Try copying the commands provided to a text editor first to perhaps have the symbols adapted automatically. Alternatively, please look at what the symbol is, for instance quotation marks, underscores, or tilde, and type those manually if you get errors when using a command provided.

3.1 Python for Pre-Processing

Make sure that you have Python 3 installed on your computer. To see whether you have it installed and which version you have, enter the command "python –version" in cmd. If you do not have it installed, visit https://www.python.org/ to download the latest version.

We have two alternatives here. You may run either the "Data_Pre_Processing.ipynb" file through Jupyter Lab, or the Data_Pre_Processing.py file through Python. The main difference is that Jupyter Lab makes viewing the code more comfortable, and you may knit a pdf output of the code run in the .ipynb file, however, it requires you to install Jupyter Lab. Alternatively, you may directly run the .py file code in cmd, which will produce the same result apart from a pdf file.

3.1.1 Using Only Python

Navigate in the folder that the analysis files are located. Type "cmd" in the address bar of this folder to open the command prompt there. Type the following command without the quotation marks or brackets:

"python Data_Pre_Processing.py"

This will output files in the Data folder, and produce a results_output.txt file.

3.1.2 Using Python Through Jupyter Lab

Make sure that you have Jupyter Lab installed, or have another solution to run .ipynb files.

To install Jupyter Lab, first make sure you have Python 3 and pip installed on your computer. To see whether you have them installed and which versions you have, enter the commands "python –version" and "pip –version" in cmd. If you do not have either installed, visit https://www.python.org/ to download the latest version of Python. Python version later than 3.4 includes pip by default.

To install Jupyter Lab, open cmd and use the command "pip install jupyterlab", without the quotation marks.

To open the relevant code file with Jupyter Lab, navigate to the folder containing it, type "jupyter lab" in the address bar of that folder and press Enter to open Jupyter Lab in your default internet browser.

To run the code contained in the entire script, click "Run" on the top bar, and click "Run All Cells". This should create the relevant data files in the Data folder, as well as the "results_output.txt" file in the main folder.

Alternatively, after opening cmd you can enter the following command without the quotes for the same result.
"jupyter nbconvert –to pdf –execute Data_Pre_Processing.ipynb
–output=Data_Pre_Processing_Run_Output.pdf"

3.2 R in Docker for Analysis

To run the analysis code in a Docker container you first need to have a Docker image. You can either build this image yourself or download it from Docker Hub.

Building Docker Image. Navigate in the folder that the analysis Rmd files are located. In Windows, click on the address bar and copy the folder path address. Type "cmd" in the address bar of this folder to open the command prompt there. Type the following command without the quotation marks or brackets:

"docker build -t [Image Name] ."

If you opened cmd from somewhere else, either navigate to the data analysis directory containing the Dockerfile, or modify the above command to be:

"docker build -f [Folder Path] -t [Image Name] ."

Replace [Image Name] with a suitable name for the image. Make sure that you include a period at the end of the command, as shown above. For example:

"docker build -f [Folder Path] -t expfam-in-vr ."

Replace [Folder Path] with the folder path that you copied from the address bar, of the folder containing the Dockerfile and analysis files, or otherwise enter the path to this folder manually.

Downloading Docker Image. You may alternatively download the image from Docker Hub using the following command:

docker pull aantoniades/expfam-in-vr

3.2.1 Running R Code in Docker

Navigate in the folder that the analysis Rmd files are located. In Windows, click on the address bar and copy the address. Type "cmd" in the address bar to open the command prompt. Type the following command without the surrounding quotation marks:

"docker run -d -p 8787:8787 -v "[Folder Path]":/home/rstudio/analysis -e PASSWORD=[password] [Image Name]"

Replace [Folder Path] with the folder path that you copied from the address bar, of the folder containing the Dockerfile and analysis files, or otherwise enter the path to this folder manually. Replace [password] with the desired value. Replace [Image Name] with the name of the image you created or if you downloaded it from Docker Hub, this would be "aantoniades/expfam-in-vr". Remove brackets.

Open an internet browser and type "http://localhost:8787/" without the quotation marks to open RStudio on the local host. For username enter "rstudio" and for password the value entered above.

Running Code and Manually Outputting to File. On the bottom right window section you should see the file structure. Click on the "analysis" folder to open it. You will see the files contained within. Open the .Rmd files and run the code segments as desired. You can run the entire file and export the output by clicking on "Knit" and then selecting the desired export format (HTML, PDF, Word). This will execute the code and render the Markdown file and its outputs in a file of the selected format.

Running Code and Outputting to File via a Console Command. Instead of running the code as mentioned above, access the Console in RStudio and enter the following commands to have it run the code in the specified files and render the output to a pdf file.

"rmarkdown::render("~/analysis/[File Name]", "pdf_document")"

Replace [File Name] with the name of the file you want to run and output, including the file extension, in this case, .Rmd, without the brackets. You can alternatively replace "pdf_document" with "html_document" if you would like to output in that format instead. Do this for each .Rmd file you want to run and output.

For the files provided, these are the following commands:

rmarkdown::render("~/analysis/Experience_Familiarity_Analysis.Rmd", "pdf_document")

rmarkdown::render("~/analysis/Sense_of_Presence_and_Interface_Usability.Rmd",
"pdf_document")

rmarkdown::render("~/analysis/Prior_Sensitivity_Analysis.Rmd", "pdf_document")

References

[1] Antoniades, A. Everyday environments as cues to smoke: Personalized environments in virtual reality to elicit smoking cravings. Msc thesis, Delft University of Technology, Delft, 2022 [Online]. http://resolver.tudelft.nl/uuid:4786dd32-eba2-41da-9079-f539cf62e72c.

[2] IGroup. Presence questionnaire. Retrieved from http://www.igroup.org/pq/ipq/index.php.