

## How to Use DECOTA: A Coding Guide

DECOTA is an openly accessible methodology, which takes free-text data and outputs the themes and sub-themes underlying the data.

For each sub-theme, it also outputs:

1. How common it is (prevalence in %)
2. How common it is amongst different demographics (covariates). Here, age and gender are the default
3. Three representative quotes

You will get the following outputs:

### From R:

1. A CSV table with information about choosing the correct number of topics (k\_metrics\_graph.csv).
2. Two plots with information about choosing the correct number of topics (k\_plot\_Question1.jpg and SemCohExclusPlot.jpg).
3. A final CSV called "Ready\_for\_GPT.csv". This needs to be inputted into PART 2 of this code in Python.

### From Python:

4. A dendrogram to show the clustering of sub-themes into overall themes.
5. A final CSV with the final themes, sub-themes, prevalence information, covariate relationships and example quotes.

Users should be aware that in its current form, using DECOTA requires a basic understanding of coding principles (e.g., in R or Python). To make it as accessible to use as possible, we have set out this instruction guide of how to use the R, then Python Script.

However, it is **easier to use the Google Colab script**, which requires no downloads, and can be accessed [here](#), with full instructions within the script.

## Instructions

### 1. **Format your free-text data**, and save as a CSV (UTF-8 encoding) file.

- Your free-text data file must be formatted in the following way, for DECOTA to work without human intervention.
  - Column 1: "ID" (the ID number of your participant)
  - Column 2: "Question\_1" (your free-text data)
  - Column 3: "age" (numeric, exact age)

- Column 4: "gender" (binary, coded 1 = Male, 0 = Female and Other). Note this grouping of gender was for demonstrative purposes and can be edited in the code if desired.

Example formatting:

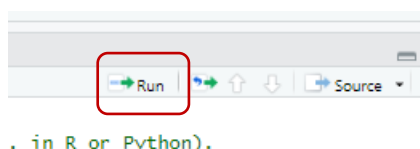
|    | A  | B           | C    | D      |
|----|----|-------------|------|--------|
| 1  | ID | Question    | age  | gender |
| 2  | 1  | I was       | 50.5 | 0      |
| 3  | 2  | I found eve | 75   | 0      |
| 4  | 3  | NOT MUCH    | 65.5 | 0      |
| 5  | 4  | The only th | 75   | 1      |
| 6  | 5  | Applicable  | 75   | 0      |
| 7  | 6  |             |      | 0      |
| 8  | 7  |             | 33   | 1      |
| 9  | 8  | Already     | 50.5 | 1      |
| 10 | 9  | Though it w | 65.5 | 0      |
| 11 | 10 | Very simpl  | 75   | 0      |
| 12 | 11 | Stuff I was | 50.5 | 1      |
| 13 | 12 | Told me thi | 65.5 | 0      |
| 14 | 13 | Clear, eas  | 50.5 | 0      |
| 15 | 14 | Not a lot   | 65.5 | 0      |
| 16 | 15 | Raising aw  | 65.5 | 0      |
| 17 | 16 | Suggested   | 65.5 | 0      |
| 18 | 17 | Very helpfu | 65.5 | 1      |
| 19 | 18 | Cleaning a  | 50.5 | 0      |

## 2. Open the code: R script ("DECOTA\_Part1\_STM.R")

- Input the path to your data where it says File\_Path "XXX". The file path will look something like "C:\\User\\Lois\\Downloads\\Data.csv"
- Input the question your participants answered in the free-text data, where it says Question "XXX". This could be something like "Why do you try and save water at home?"

```
# Main Code -----
# User inputs file path for data and question asked, in required format
File_Path <- "XXX.csv"
Question <- "XXX"
```

## 3. Run the R script.

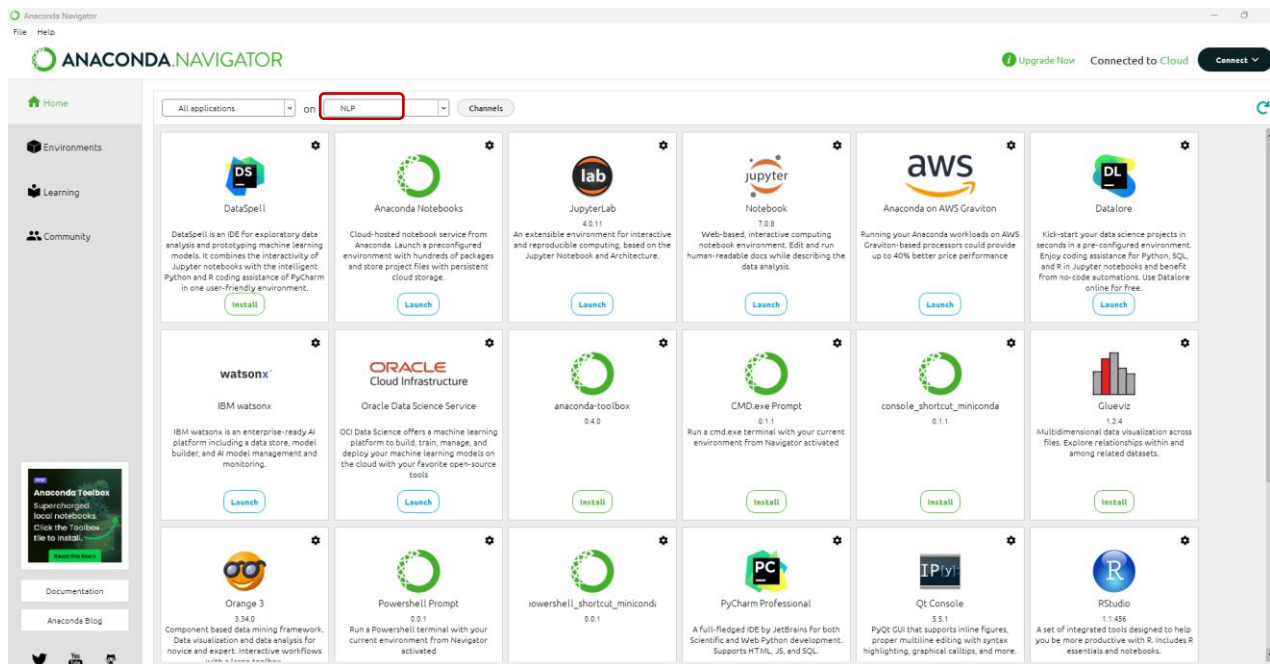


- Note that you may encounter an error on the "Make K Graph and Table" cell. Please do not worry and continue to run the following cells with the play button next to each cell.
- Throughout this R code, you will receive several outputs visible in the folder in the left hand toolbar, including:

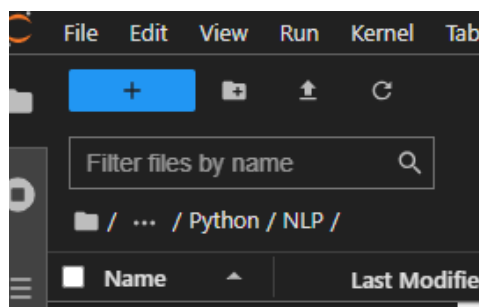
Player et al. (2024). The Use of Large Language Models for Qualitative Research (DECOTA).

- A CSV table with information about choosing the correct number of topics (k\_metrics\_graph.csv).
- Two plots with information about choosing the correct number of topics (k\_plot\_Question1.jpg and SemCohExclusPlot.jpg).
- A final CSV called “Ready\_for\_GPT.csv”. This needs to be inputted into PART 2 of this code in Python.

#### 4. Install Anaconda Navigator onto your desktop, to run the Python script.



- Create a new “environment” on the left hand side. Call this “NLP”, and select the new environment.
- You will need to add the following packages to the new environment:
  - pandas
  - openai
  - sentence\_transformers
  - scipy
  - pandas
  - numpy
  - time
  - matplotlib
- Launch “Jupyter Lab”, and navigate to the file directory where the **Ready\_for\_GPT.csv** is stored.



- Save the Python file “DECOTA\_Part2\_LLM.py” in the same directory.

## 5. Input information on Python.

- If you correctly navigated to the file directory, you should not need to edit the below section:

```
•[3]: # Import data
      file_path = 'Ready_for_GPT.csv'
      df_model1 = pd.read_csv(file_path)

      #df_model1.head()
```

- Add in your model names and api\_key in the relevant places.
  - api\_key - this is the unique licence key from your OpenAI account
  - Model1 (model="ft:gpt-3.5-turbo-0613:XXX") - the name of your fine-tuned language model 1, from OpenAI
  - Model2 (model="ft:gpt-3.5-turbo-0613:XXX") - the name of your fine-tuned language model 2, from OpenAI

## 6. Run the Python code.

- You will see the code making progress with naming the codes, clustering, then naming themes. For example, it counts through as it names the codes:

```
1 / 25
2 / 25
3 / 25
4 / 25
5 / 25
6 / 25
7 / 25
8 / 25
9 / 25
10 / 25
11 / 25
12 / 25
13 / 25
```

- The default cut-off value for the dendrogram is 0.8. This can be changed if the user desires, using the dendrogram to decide.

```
•[11]: # Define a cutoff distance for cluster formation
      cutoff_distance = 0.80
```

- The final file “Themes\_and\_Codes.csv” will be saved in the file director you set.

## 7. Visualise results with a Sankey Diagram.

- This code is provided in the “DECOTA\_Part3\_SankeyDiagram.R” code.
- Input your file\_path as before. The file you need is the “Themes\_and\_Codes.csv”.

```
# Read in the data
file_path <- "XXX.csv"
df <- read_csv(file_path)
```

- Run the code, and find the output in your file directory.
- It will look like the below example:

