Overview of the experiment:

This experiment is a Stroop Task. The participant sees the names of colours presented in various colours, sometimes the colour matches the name and sometimes it doesn't. They have to respond with what the colour of the word is, ignoring the text of the word.

For the experimenter only: The purpose is to compare responses when the colour and text match (congruent) with when the colour and text don't match (incongruent).

Running the experiment:

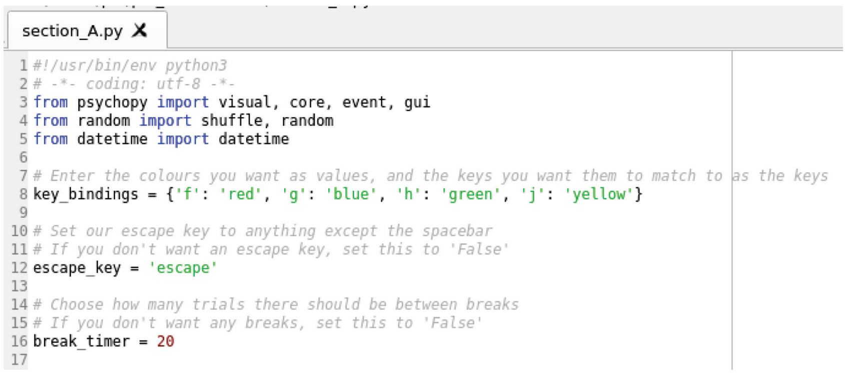
These instructions are based on running the experiment on the Virtual Machine (VM). The instructions will generalise to other systems and devices, but slight changes might have to be made, and the screenshots might look slightly different.

Loading the script

- Open Spyder3. In the VM found on the top bar or under Applications > Development > Spyder3.
- Click File > Open to open a file.
- Navigate through the file system to the pin_assessment2 folder. Then open the section_A.py file.

Checking the script is correct

- When you open the file, the start of it should look like Figure 1 to the right.
- Check that the keys and colours (line 8), escape key (line 12), and the break timer (line 16) are set to what your supervisor has told you.
- IMPORTANT: Do not change any of these unless you have been explicitly told to.
- If you change the setting, ensure you follow the instructions in the code. IMPORTANT: letters and word should be in inverted commas. 'False' and numbers will work with or without inverted commas. IMPORTANT: number should be written as digits, NOT words, e.g. 10 not ten.

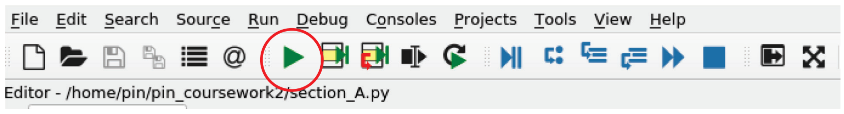


```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  from psychopy import visual, core, event, gui
4  from random import shuffle, random
5  from datetime import datetime
6
7  # Enter the colours you want as values, and the keys you want them to match to as the keys
8  key_bindings = {'f': 'red', 'g': 'blue', 'h': 'green', 'j': 'yellow'}
9
10 # Set our escape key to anything except the spacebar
11 # If you don't want an escape key, set this to 'False'
12 escape_key = 'escape'
13
14 # Choose how many trials there should be between breaks
15 # If you don't want any breaks, set this to 'False'
16 break_timer = 20
17
```

Figure 1. The start of the section_A program

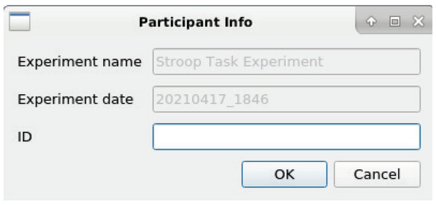
Running the experiment

- The experiment can be started while the participant is there or beforehand.
- Click Start (outlined in red in Figure 2), found in the top left corner or Run > Run to begin the experiment.
- A dialog will open with the experiment details, as seen in Figure 3. Enter the participant ID and click OK.
- Then the actual experiment window will open, this may take a few seconds. It will display the introduction screen for the participant, as seen in Figure 4. IMPORTANT: If you have done this before the participant arrives, stop at this point. The participant should continue to the experiment when they're ready.



Editor - /home/pin/pin_coursework2/section_A.py

Figure 2. The toolbar at the top of Spyder, with the start button circled in red



Experiment name: Stroop Task Experiment

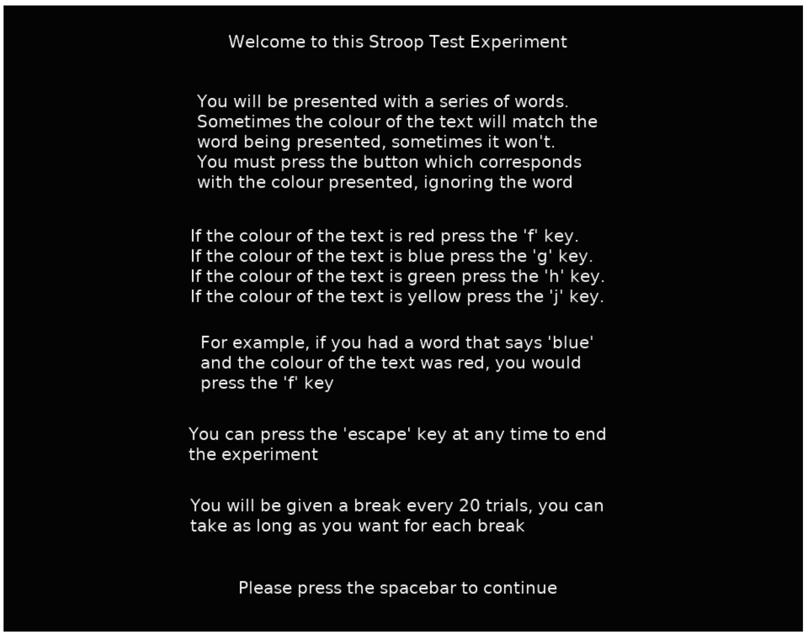
Experiment date: 20210417_1846

ID:

Figure 3. The inital dialog to enter the participant ID

For the participant

- Greet the participant when they arrive.
- The introduction screen has all they need to complete the experiment. Check they understand and answer any questions.



Welcome to this Stroop Test Experiment

You will be presented with a series of words.
Sometimes the colour of the text will match the word being presented, sometimes it won't.
You must press the button which corresponds with the colour presented, ignoring the word

If the colour of the text is red press the 'f' key.
If the colour of the text is blue press the 'g' key.
If the colour of the text is green press the 'h' key.
If the colour of the text is yellow press the 'j' key.

For example, if you had a word that says 'blue' and the colour of the text was red, you would press the 'f' key

You can press the 'escape' key at any time to end the experiment

You will be given a break every 20 trials, you can take as long as you want for each break

Please press the spacebar to continue

Figure 4. Introduction screen

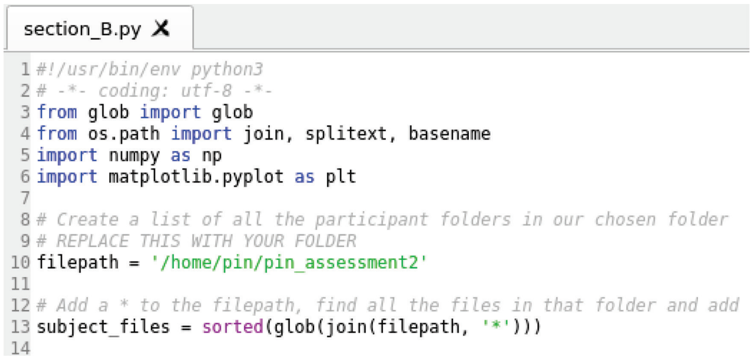
After the experiment

- Thank the participant.
- Close Spyder3. The output file will have saved automatically.

Running the analysis:

Opening the script

- Open the file section_B.py in the same way you opened section_A.py to run the experiment above.



```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
from glob import glob
from os.path import join, splitext, basename
import numpy as np
import matplotlib.pyplot as plt

# Create a list of all the participant folders in our chosen folder
# REPLACE THIS WITH YOUR FOLDER
filepath = '/home/pin/pin_assessment2'

# Add a * to the filepath, find all the files in that folder and add
subject_files = sorted(glob(join(filepath, '*')))

```

Figure 5. The start of the section_B program

Adding the correct folder

- The start of the file should look like the screenshot in Figure 5.
- Replace the text within the inverted commas with the filepath to the folder which contains the output files. If the files were created using the program above, and haven't been moved, this will be '/home/pin/pin_assessment2' as shown in the screenshot. IMPORTANT: file path must have inverted commas around it. Also, check the initial forward-slash is included if it's supposed to be.
- If it's a different folder, you'll be given it or have to find it yourself. The easiest way to get the filepath yourself is to navigate to the folder in your file manager and copy the filepath from the top bar. Shown in the VM in Figure 6.

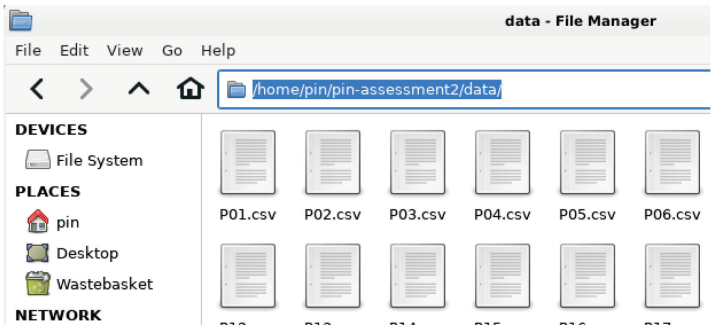


Figure 6. The filepath of the data folder can be copied from the top bar in the File Explorer

Running the analysis

- Same as above, press the Start button highlighted in Figure 2 or Run > Run to begin the program.

Checking the output

- Looking at the output of the program in the IPython console window, which should be found to the bottom right of Spyder.
- At the top there will be a line telling you how many participants were included in this analysis. Check this is the same as the number of participants you intended to analyse.
- In this output there will also be a table with the descriptive statistics for each participant and then the group as a whole. This text can be selected and copied if needed.
- A figure will open with graphs comparing the two conditions. This is saved in the same folder as the Section_A and Section_B files under the name group.png.

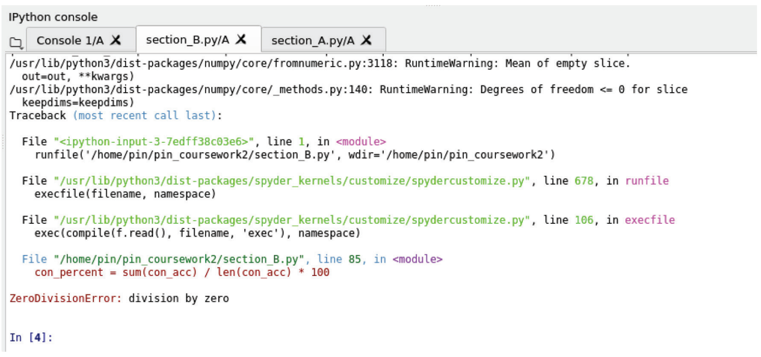
Thank you for your help!

Handling Issues:

If something breaks and either of the programs crash, these are common issues that you should check:

- Go back to the IMPORTANT points in the document and check you're following them correctly.
- For running the analysis, check that the folder with the data in is correct, and has been entered correctly.
- Check that the beginnings of your programs look like Figure 1 and Figure 5 for sections A and B respectively. Accidentally adding a letter somewhere can cause issues.

If you can't solve an issue, please contact your supervisor. If the issue is related to the program crashing, it might be useful to include a screenshot of the IPython window to give them a better idea of what's gone wrong. An example of this can be seen in Figure 7.



```
/usr/lib/python3/dist-packages/numpy/core/fromnumeric.py:3118: RuntimeWarning: Mean of empty slice.
  out=out, **kwargs)
/usr/lib/python3/dist-packages/numpy/core/_methods.py:140: RuntimeWarning: Degrees of freedom <= 0 for slice
  keepdims=keepdims)
Traceback (most recent call last):

  File "<ipython-input-3-7edff38c03e6>", line 1, in <module>
    runfile('/home/pin/pin_coursework2/section_B.py', wdir='/home/pin/pin_coursework2')

  File "/usr/lib/python3/dist-packages/spyder_kernels/customize/spydercustomize.py", line 678, in runfile
    execfile(filename, namespace)

  File "/usr/lib/python3/dist-packages/spyder_kernels/customize/spydercustomize.py", line 106, in execfile
    exec(compile(f.read(), filename, 'exec'), namespace)

  File "/home/pin/pin_coursework2/section_B.py", line 85, in <module>
    con_percent = sum(con_acc) / len(con_acc) * 100

ZeroDivisionError: division by zero


In [4]:
```

Figure 7. Screenshot of an error message