Assignment 1 Data analytics & communication

Sebastian Pusch (S5488079), Ivan Hegeman (s4789725)

1 Critically evaluating bachelor projects

1.1 Research summary

The study by Kaveh Rasouli investigates the effects of mindful practices and positive fantasizing interventions on perseverative cognition (PC) and risk of depressive relapse. To examine these effects, participants completed the app-based Sustained Attention to Response Task (SART), which provided both self-reported measures of PC and objective cognitive data, such as response time. The experiment involved two distinct participant groups: individuals with remitted Major Depressive Disorder (rMDD) and Never-Depressed (ND). The results show that both generally led to faster response times across groups, but reduced task focus among rMDD participants. Positive fantasizing had a stronger influence on task performance in ND participants. However, both interventions were associated with impaired inhibitory control.

1.2 Strong point of the research

We found that this research has a very thorough participant selection, which seems necessary as they are working with people with remitted Major Depressive Disorder (rMDD). First, the rMDD participants are required to have had at least 2 depressive episodes (following the criteria in the Diagnostic Statistical Manual version 5; DSM-5), but it is also important for the researchers that the participants are not currently in a depressive episodes, which is why every rMDD participant has to score 21 or lower on the IDS-SR30 (a score of 21 or lower indicate no clinically relevant depressive symptoms). Second, participants should not be taking any psychotropic or neuroactive medications (such as antidepressants, benzodiapenes, etc.), to prevent having medication influence the outcomes of the study. Third, participants should not have engaged in any mindfulness practices within the last two years, to make sure that they do not have any previous practice which once again could influence the outcomes. Finally, the Never Depressed (ND) participants were also tested for depression (using the IDS-SR30 test similarly), and any candidate with a score higher than 13 would not be able to participate to the study. Overall, we found that the selection of the participants was very considerate of the many factors that could influence the study (and acted accordingly to prevent as much influence on the outcome as possible), while also taking into consideration the challenges and implications of working with participants with major depressive episodes.

1.3 Weakness of the research

In our view, the main weakness of this research lies in the decision to have participants undergo both the mindfulness and positive fantasizing interventions, which complicates reproducibility. As the author points out, this setup increases participants' familiarity with the task, leading to greater task fluency. This could skew results in the second intervention phase, as participants might complete the task more easily simply due to practice, which could in turn leave more room for PC-like thought patterns to emerge. This raises concerns about the reliability of objective measurements, especially if improved performance is driven by task repetition rather than the intervention itself. Additionally, while a one-month wash-out period was introduced to minimize carry-over effects, this might not be enough to account for psychological changes

in rMDD individuals, who are particularly prone to shifts in mood and thinking over time. This makes it harder to draw clear conclusions from within-subject comparisons, as observed differences may reflect these natural changes rather than the specific effects of the interventions.

1.4 Potential Improvements

In our opinion, the use of a between-subjects design would be better suited for this experiment, as it would eliminate the effects of task repetition and practice. With a between-subjects design, participants would only undergo one of the interventions (mindfulness or positive fantasizing), allowing for a clearer comparison between the two groups without the potential influence of increased task familiarity. This would reduce the risk of task fluency affecting the results and provide a more accurate assessment of the interventions' true effects. Also, a between-subjects design would remove the effects that the fluctuating cognitive and emotional states of rMDD participants could have on the experiment.

If a between-subjects design is not possible, having an extended wash-out period would be beneficial, with participants performing the second task only if they are in the same cognitive and emotional state as for the first experiment (this could be measured using some tests).

2 2. Making sure a planned Bachelor project is replicable and reproducible

2.1 Potential BSc project

A potential bachelor project we would be interested in is researching the creation of false memories through a cognitive model. More specifically, we aim to model how, when participants are presented with a list of semantically related words (such as 'bed,' 'cushion,' 'moon,' etc) they may later falsely remember seeing a related but non-presented word like 'sleep.' We would develop a cognitive model that performs the Deese-Roediger-McDermott (DRM) task, replicating the experimental setup originally designed by Deese and later improved by Roediger and McDermott, in order to investigate how false memories are formed. The goal of this project would be to develop a cognitive model that replicates the results of Roediger and McDermott's research, aiming to mimic human behavior as closely as possible in order to understand through our model how false memories are created.

2.2 Reproducibility and replicability

Reproducibility is the ability to perform an experiment in the same way as it was designed and performed in existing research. To make research reproducible, a clear and complete experiment design is required, including how participants are selected, what variables are collected, what information researchers and participants are aware of during the experiment, and more. Furthermore, reproducibility also requires a clear description of how the collected data was cleaned and processed, the criteria by which outliers were removed, and what tests (including inconclusive ones) were performed. Ideally, all collected data, scripts, and other tools should be made public. These steps allow future research to reproduce the experiment. Reproducibility is necessary but not sufficient for replicability. Replicability is the process of performing the same experiment and reaching the same conclusion through statistically similar results. To increase the likelihood of replicable findings, a clear hypothesis and analysis plan should be registered before the data is collected. This reduces the bias induced by the natural human tendency to observe patterns in noise. However, removing bias does not guarantee reproducibility, as findings from previous research might not generalize to other circumstances. This is especially true in the field of cognitive psychology, as experiments on human participants are intrinsically less likely to generalize well compared to other fields.

2.3 How are we making our thesis reproducible?

First, to ensure reproducibility, the experimental setup must be clearly and fully defined. This includes specifying the exact lists of semantically related words to be used, the number of words per list, the presentation order (randomized or fixed), and the timing for each word's display. Furthermore, the internal structure and functioning of the cognitive model should be thoroughly described. This means specifying the model's architecture, how the words are represented internally, how associations between words are modeled, how activation spreads, and how memory creation/retrieval is simulated. All parameter settings and assumptions should be reported clearly. Finally, the full source code of the model should be shared openly. The code should be well-documented with clear comments explaining key parts, and a README file should be included with instructions on how to run the model.

2.4 How are we making our thesis replicable?

Even though we are using a model for our study, there still are a few key points to consider that could have an influence on the replicability of our study. First, hardware differences could affect replicability, especially if the model's performance depends on factors such as processing speed, memory capacity, or other system characteristics. Therefore, it is important to document the hardware used during the experiment to ensure that others can achieve comparable results. Also, any random processes that are involved in the model (such as randomized presentation orders, stochastic memory retrieval, or noise in activation levels) should be explicitly documented. Random number generators should be seeded and the seed value reported. This way, anyone rerunning the model can replicate the exact sequence of random choices and thus achieve identical outputs.

3 Getting familiar with the tidyverse

3.1 Describe your dataset

We chose a dataset that includes climate related disaster frequencies. Each observation is grouped by year and country and contains various country information (ISO codes, name, etc.), a description of the indicator, the unit, the source of the data and CTS related info. The data is in wide format, meaning that the years, within the range 1980-2024, are columns instead of values.

```
library(tidyverse)
```

```
data = read.csv('./data/phys_risks_climate_related_distaster_frequency.csv')
```

3.2 Mutate

```
data_mutated <- mutate(data, difference_80_24 = X1980 - X2024)

head(data_mutated$difference_80_24)
```

```
## [1] NA NA -4 NA NA -7
```

The mutate() command adds a new column that is a function of existing variables. In our case, we create a new column that computes the difference in climate disasters between 1980 and 2024 for each type.

3.3 Count

```
num_rows = 10
```

The `count` function allows us to group by and count all distinct entries of a specified column.

For example, we can first group by country and then apply `head` which will output the first 10 rows:

```
head(count(data, Country), n = num_rows)
```

```
##                          Country  n
## 1  Afghanistan, Islamic Rep. of 14
## 2                        Albania 14
## 3                        Algeria 14
## 4                 American Samoa  6
## 5                         Angola  8
## 6                       Anguilla  4
## 7            Antigua and Barbuda  6
## 8                      Argentina 14
## 9                Armenia, Rep. of 12
## 10                     Australia 14
```

Otherwise, if we first apply `head` and subsequently `count`, since the first 10 entries contain the same country, we will obtain a single row (with n = 10):

```
count(head(data, n = num_rows), Country)
```

```
##                         Country  n
## 1 Afghanistan, Islamic Rep. of 10
```

3.4 Filter

```
head(subset(data, select = c("Country", "X1980", "X2024")))
```

```
##                         Country X1980 X2024
## 1 Afghanistan, Islamic Rep. of    NA    NA
## 2 Afghanistan, Islamic Rep. of    NA     1
## 3 Afghanistan, Islamic Rep. of     1     5
## 4 Afghanistan, Islamic Rep. of    NA     2
## 5 Afghanistan, Islamic Rep. of    NA    NA
## 6 Afghanistan, Islamic Rep. of     1     8
```

```
filtered_data <- filter(data, Indicator == "Climate related disasters frequency, Number of Disasters: T

head(subset(filtered_data, select = c("Country", "X1980", "X2024")))
```

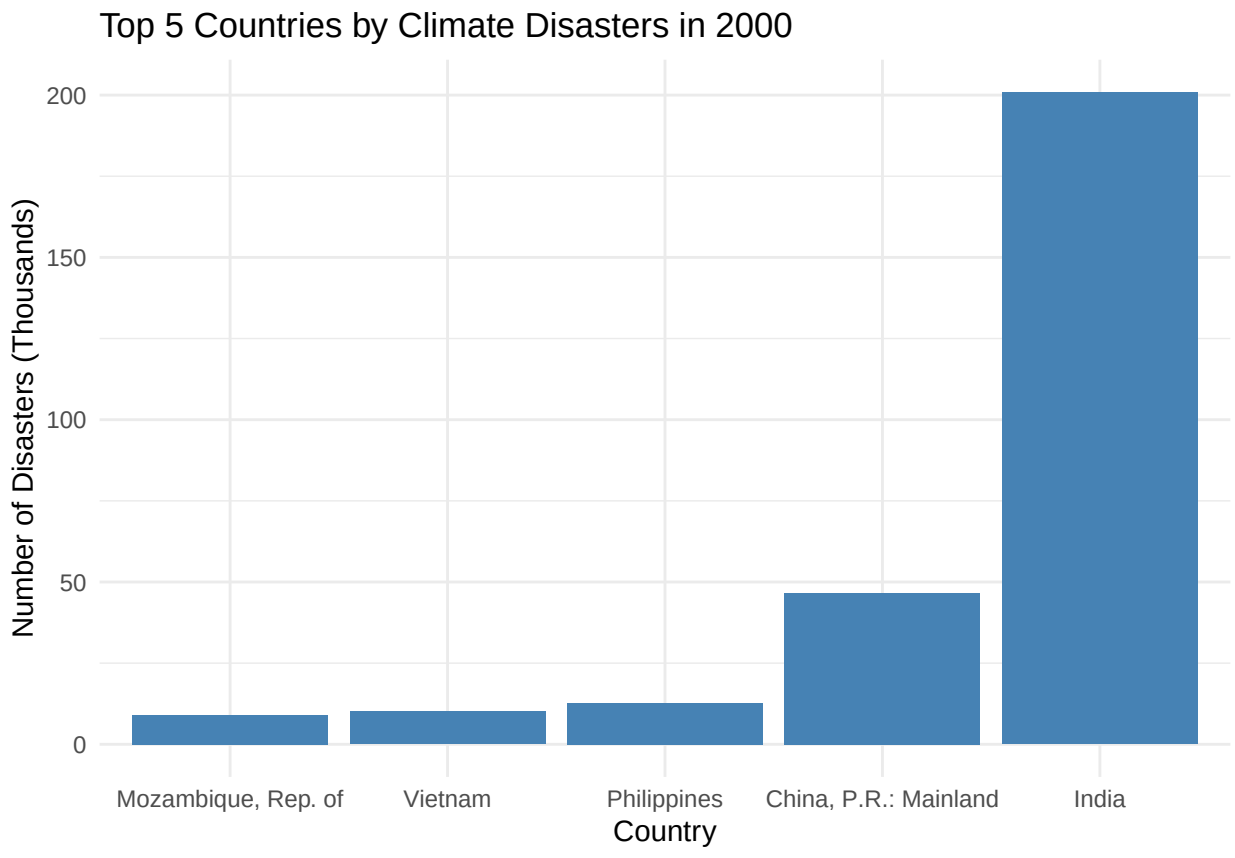
```
##                         Country X1980 X2024
## 1 Afghanistan, Islamic Rep. of     1     8
```

```
## 2                    Albania    NA    NA
## 3                    Algeria    NA     2
## 4             American Samoa    NA    NA
## 5                     Angola    NA     1
## 6                   Anguilla    NA    NA
```

The filter() command is used to keep only the rows that satisfy certain conditions (similarly to how subset() only keeps certain columns). In our case, we use it to only keep the total number of climate disasters, and we also subset the result to only keep the years 1980 and 2024 because otherwise our table would be too big.

3.5 Make a plot with ggplot

```
data %>%
  group_by(Country) %>%                                  # multiple entries per country
  summarise(X2000 = sum(X2000, na.rm = TRUE)) %>%   # sum frequencies
  arrange(desc(X2000)) %>%                               # sort by frequency
  slice_head(n = 5) %>%                                  # get top 5
  ggplot(aes(x = reorder(Country, X2000), y = X2000 / 1000)) +
  geom_col(fill = "steelblue") +
  scale_y_continuous(labels = function(x) format(x / 1000, scientific = FALSE)) +
  labs(title = "Top 5 Countries by Climate Disasters in 2000",
       x = "Country",
       y = "Number of Disasters (Thousands)") +
  theme_minimal()
```



3.6 Summarizing data

```
head(subset(data, select = c("Country", "X2024")))
```

```
##                         Country X2024
## 1 Afghanistan, Islamic Rep. of    NA
## 2 Afghanistan, Islamic Rep. of     1
## 3 Afghanistan, Islamic Rep. of     5
## 4 Afghanistan, Islamic Rep. of     2
## 5 Afghanistan, Islamic Rep. of    NA
## 6 Afghanistan, Islamic Rep. of     8
```

```
summarized_data <- data %>%
  group_by(Country) %>%
  summarize(total = sum(X2024, na.rm=TRUE))

head(summarized_data)
```

```
## # A tibble: 6 x 2
##   Country                        total
##   <chr>                          <int>
## 1 Afghanistan, Islamic Rep. of 1280184
## 2 Albania                            0
## 3 Algeria                        24004
## 4 American Samoa                     0
## 5 Angola                       5600002
## 6 Anguilla                           0
```

Summarize is used to summarize each group to one row. In our case, we used it to get the sum of all the values in the 2024 column, grouped by Country.

3.7 Spreading and gathering data

```
years = paste0('X', as.character(1980:2024)) # create a range 1980-2024 containing all col names

long_data = pivot_longer(data, years, names_to = 'Year', values_to = 'Frequency')

head(arrange(long_data, desc(Frequency)))
```

```
## # A tibble: 6 x 12
##   ObjectId Country ISO2  ISO3  Indicator          Unit  Source CTS.Code CTS.Name
##      <int> <chr>   <chr> <chr> <chr>              <chr> <chr>  <chr>    <chr>
## 1      811 India   IN    IND   Climate related d~ Numb~ The E~ ""       ""
## 2      811 India   IN    IND   Climate related d~ Numb~ The E~ ""       ""
## 3      806 India   IN    IND   Climate related d~ Numb~ The E~ ""       ""
## 4      811 India   IN    IND   Climate related d~ Numb~ The E~ ""       ""
## 5      806 India   IN    IND   Climate related d~ Numb~ The E~ ""       ""
## 6      806 India   IN    IND   Climate related d~ Numb~ The E~ ""       ""
## # i 3 more variables: CTS.Full.Descriptor <chr>, Year <chr>, Frequency <int>
```

3.8 Separating and uniting

```
head(subset(data, select=c("Country", "X2023", "X2024")))
```

```
##                         Country X2023 X2024
## 1 Afghanistan, Islamic Rep. of    NA    NA
## 2 Afghanistan, Islamic Rep. of     1     1
## 3 Afghanistan, Islamic Rep. of     2     5
## 4 Afghanistan, Islamic Rep. of    NA     2
## 5 Afghanistan, Islamic Rep. of    NA    NA
## 6 Afghanistan, Islamic Rep. of     3     8
```

```
united_data <- unite(data, "2023-2024", X2023:X2024, sep="-", na.rm=TRUE)
head(subset(united_data, select=c("Country", "2023-2024")))
```

```
##                         Country 2023-2024
## 1 Afghanistan, Islamic Rep. of
## 2 Afghanistan, Islamic Rep. of       1-1
## 3 Afghanistan, Islamic Rep. of       2-5
## 4 Afghanistan, Islamic Rep. of         2
## 5 Afghanistan, Islamic Rep. of
## 6 Afghanistan, Islamic Rep. of       3-8
```

The unite() command unites multiple columns into one, and enables us to choose the separator. In our case, we chose to unite the columns 2023 and 2024.

```
separated_data <- separate(united_data, "2023-2024", c("new_2023", "new_2024"), sep = "-", fill="right")
head(subset(separated_data, select=c("Country", "new_2023", "new_2024")))
```

```
##                         Country new_2023 new_2024
## 1 Afghanistan, Islamic Rep. of              <NA>
## 2 Afghanistan, Islamic Rep. of        1        1
## 3 Afghanistan, Islamic Rep. of        2        5
## 4 Afghanistan, Islamic Rep. of        2     <NA>
## 5 Afghanistan, Islamic Rep. of              <NA>
## 6 Afghanistan, Islamic Rep. of        3        8
```

The separate() command separates a column into multiple ones using a regex filter. In our case, we once again separated the columns we united. Something interesting to note is that the new_2023 column contains empty values, while the new_2024 column contains NA values, which is because we decided to use fill = "right", which means that the right column will be filled with missing values.

4 Contributions

Ivan 50% Sebastian 50%