

Mehrabi et al. 2022. Research priorities for global food security
under extreme events. Supplementary data and code.

Zia Mehrabi

04 July 2022

Contents

| | | |
|----------|-------------------------------------|----------|
| 1 | Aims | 2 |
| 2 | Reproducibility | 2 |
| 3 | Threats | 2 |
| 3.1 | Read data | 2 |
| 3.2 | Compute the ranks | 3 |
| 3.3 | Save | 4 |
| 4 | Research questions | 5 |
| 4.1 | Read data | 5 |
| 4.2 | Compute the ranks | 5 |
| 4.3 | Low and high effort split | 6 |
| 4.4 | Save | 6 |
| | References | 7 |

1 Aims

The aim of this document is to provide the scripts for reproducing the final results shown in Mehrabi et al., Research priorities for global food security under extreme events, One Earth (2022), <https://doi.org/10.1016/j.oneear.2022.06.008>. It provides a simple ranking exercise of expert elicited threats and research questions based on perceived priority responses collected through online surveys. Please see the paper for detailed background for this study. You can contact Zia Mehrabi at the above email address if you have any questions, queries or corrections for this work.

2 Reproducibility

We use the **R** packages `knitr` (Xie 2021) and `checkpoint` (Ooi, de Vries, and Microsoft 2021). The package `knitr` facilitates producing a dynamic document that contains all the steps required to analyze the data. `checkpoint()` will install all packages versions that we used in our analysis to avoid result discrepancies that may arise from software differences. Thus the reader is provided with all the code to fully reproduce the analysis, and adapt it for other analyses.

```
require("checkpoint")
checkpoint(snapshotDate = "2022-07-01")
```

For the analysis in this document we will be using the `tidyverse` (Wickham 2021) and `ordinal` (Christensen 2019) packages.

```
packages <- c("ordinal", "tidyverse")
invisible(lapply(packages, require,
  character.only = TRUE))
```

3 Threats

3.1 Read data

First we read in the data. We note that the data has been anonymized to remove personal identifying information of individual survey respondents (e.g. emails, names, IP addresses etc). We do not provide all upstream processing steps (which includes reading in raw data imported from the online survey instrument administered in Qualtrics) for this reason.

The input data dimensions characteristics and additional meta-data e.g. variable descriptors are shown below.

```
threats <- read_rds("threatsin.rds")
str(threats)

## tibble [1,648 x 7] (S3: tbl_df/tbl/data.frame)
## $ id      : Factor w/ 32 levels "1","2","3","4",...: 1 1 1 1 1 1 1 1 1 1 ...
##   .. attr(*, "comment")= chr "unique identifier for threats"
## $ title    : chr [1:1648] "Adaptive tipping points" "Adaptive tipping points" "Adaptive tipping po
##   .. attr(*, "comment")= chr "short title for each threat"
## $ threat   : chr [1:1648] "An increase in extreme events frequency and severity leading to continu
##   .. attr(*, "comment")= chr "text description of each threat"
## $ score     : Ord.factor w/ 5 levels "1"<"2"<"3"<"4"<...: 5 5 5 4 5 5 4 5 4 4 ...
##   .. attr(*, "comment")= chr "likert score, from 1-5, low-high, unanchored intervals"
## $ respondent: Factor w/ 30 levels "1","2","3","4",...: 11 24 25 5 3 4 27 23 18 2 ...
##   .. attr(*, "comment")= chr "unique identifier for the respondent"
## $ type      : Factor w/ 2 levels "Impact","Probability": 1 1 1 1 1 1 1 1 1 1 ...
##   .. attr(*, "comment")= chr "likert score response variable"
## $ id.type   : Factor w/ 64 levels "1 Impact","1 Probability",...: 1 1 1 1 1 1 1 1 1 1 ...
##   .. attr(*, "comment")= chr "concatenation of id and type"
```

```
## - attr(*, "na.action")= 'omit' Named int [1:150] 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658
## ..- attr(*, "names")= chr [1:150] "1649" "1650" "1651" "1652" ...
```

3.2 Compute the ranks

Next we compute both the modes and the probabilities of those modes for each question and response variable using hierarchical cumulative link models, conditioning on respondents which we treat as random effects. We don't explicitly model correlations between impact and probability scores.

```
tmp_mod <- clmm2(score ~ id.type,
  random = respondent, data = threats,
  Hess = T, nAGQ = 1)
```

```
## Warning: clmm2 may not have converged:
## optimizer 'ucminf' terminated with max|gradient|: 5.9419741689604e-05
```

We save the results in a table where we extract the population level predictions, identify the mode, and concatenate both the mode and the probabilities.

```
threats$fitted <- predict(tmp_mod,
  newdata = threats) #get pop effects

ranks <- threats %>%
  mutate(score = factor(score,
    ordered = F)) %>%
  group_by(id, type, score) %>%
  summarize(fit = unique(fitted)) %>%
  ungroup() %>%
  group_by(id, type) %>%
  filter(fit == max(fit)) %>%
  mutate(score.prob = as.numeric(paste(score,
    fit, sep = ""))) %>%
  ungroup()
```

'summarise()' has grouped output by 'id', 'type'. You can override using the '.groups' argument.

We then compute the rankings for each response, make the data frame wide and compute the mean rank for each threat using ranks for each response.

```
ranks <- ranks %>%
  group_by(type) %>%
  mutate(rank = (min_rank(desc(score.prob))))

ranks.w <- ranks %>%
  pivot_wider(., names_from = type,
    names_sep = ".", values_from = c(score,
    fit, score.prob, rank)) %>%
  unnest(cols = everything())

ranks.w$mean.rank <- ((ranks.w$rank.Impact +
  ranks.w$rank.Probability)/2)
```

We add the text descriptions back in.

```
descript <- threats %>%
  select(id, title, threat) %>%
  distinct()
```

```
out <- right_join(descript, ranks.w,  
  by = "id")
```

3.3 Save

We then save the output as a csv. Note, some of the text in this file may differ from that in the final manuscript due to iterative in-text editing of the paper.

```
write.csv(out, "threatsout.csv")
```

4 Research questions

4.1 Read data

We first read in the data. As for the threats file, we see the characteristics of the data, and variable descriptions.

```
questions <- read_rds("questionsin.rds")
questions$expertise <- factor(questions$expertise,
  ordered = F)
str(questions)

## tibble [6,890 x 7] (S3: tbl_df/tbl/data.frame)
## $ id      : Factor w/ 147 levels "10","100","101",...: 137 137 137 137 137 137 137 137 137 137 ...
## ..- attr(*, "comment")= chr "unique identifier for the questions"
## $ text     : chr [1:6890] "How context dependent are on-farm resilience practices across the world
## ..- attr(*, "comment")= chr "text description of each question"
## $ type     : Factor w/ 2 levels "Difficulty","Impact": 2 2 2 2 2 2 2 2 2 2 ...
## ..- attr(*, "comment")= chr "likert score response variable"
## $ respondent: Factor w/ 29 levels "1","2","3","4",...: 14 4 5 14 19 11 7 16 12 23 ...
## ..- attr(*, "comment")= chr "unique identifier for the respondent"
## $ score    : Ord.factor w/ 5 levels "1"<"2"<"3"<"4"<...: 5 5 4 4 4 3 2 2 3 3 ...
## ..- attr(*, "comment")= chr "likert score, from 1-5, low-high, unanchored intervals"
## $ expertise : Factor w/ 5 levels "1","2","3","4",...: 5 5 2 4 4 4 1 1 3 2 ...
## $ id.type   : Factor w/ 294 levels "10 Difficulty",...: 274 274 274 274 274 274 274 274 274 274 ...
## ..- attr(*, "comment")= chr "concatenation of id and type"
```

4.2 Compute the ranks

We then compute the ranks, here as we have self declared expertise levels for each expert, we condition the population estimates on those expertise levels, so results are comparable, regardless of expertise.

```
tmp_mod.q <- clmm2(score ~ id.type +
  expertise, random = respondent,
  data = questions, Hess = T,
  nAGQ = 1)

## Warning: clmm2 may not have converged:
## optimizer 'ucminf' terminated with max|gradient|: 0.00204197970431015
```

We then save the results to a table. We use the model above to predict the modes and probabilities of those models given maximum expertise for each question and response. And then find the ranks of questions along each response.

```
questions.exp0 <- questions #create new data-frame.
questions$expertise <- as.factor(5) #set expertise to be high for predictions
questions$fitted <- predict(tmp_mod.q,
  newdata = questions.exp0) #get pop effects at high expertise

ranks.q <- questions %>%
  group_by(id, text, type, score) %>%
  summarize(fit = unique(fitted)) %>%
  ungroup %>%
  group_by(id, text, type) %>%
  filter(fit == max(fit)) %>%
  mutate(score.prob = paste(score,
    fit, sep = ""))
```

```
## 'summarise()' has grouped output by 'id', 'text', 'type', 'score'. You can override using the '.group'
ranks.q <- ranks.q %>%
  group_by(type) %>%
  mutate(rank = (min_rank(desc(score.prob))))
```

4.3 Low and high effort split

Next we identify the top 50 questions in terms of perceived impact. We also split the questions into low vs high effort groups based on percentile split of the difficulty within the high impact questions.

```
i <- subset(ranks.q, type == "Impact" &
  rank <= 50)
d <- subset(ranks.q, id %in% i$id &
  type == "Difficulty")
d$score.prob <- as.numeric(d$score.prob)
d$high.low <- ifelse(d$score.prob >
  quantile(d$score.prob, probs = 0.5),
  "high", "low")
d$diff.score <- paste(as.numeric(d$score),
  as.numeric(d$score.prob))
d$impact.score <- i$score.prob
d$impact.rank <- i$rank
d$diff.rank <- d$rank
```

4.4 Save

Finally we save the output. Again, as for threats, some of the text in this file may differ from that in the final manuscript due to iterative in-text editing of the paper.

```
out <- d %>%
  ungroup() %>%
  select(id, text, high.low,
    impact.score, impact.rank,
    diff.score, diff.rank)

write.csv(out, "questionsout.csv")
```

References

- Christensen, Rune Haubo Bojesen. 2019. *Ordinal: Regression Models for Ordinal Data*. <https://github.com/runehaubo/ordinal>.
- Ooi, Hong, Andrie de Vries, and Microsoft. 2021. *Checkpoint: Install Packages from Snapshots on the Checkpoint Server for Reproducibility*. <https://github.com/RevolutionAnalytics/checkpoint>.
- Wickham, Hadley. 2021. *Tidyverse: Easily Install and Load the Tidyverse*. <https://CRAN.R-project.org/package=tidyverse>.
- Xie, Yihui. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. <https://yihui.org/knitr/>.