Christian Adam, Xavier Fernández i Marín, Oliver James, Anita Manatschal, Carolin Rapp, Eva Thomann

Differential discrimination against EU mobile citizens

2021-04-26 10:03

Contents

1
Introduction

This report is the replication material for the article "Differential discrimination against EU mobile citizens: Experimental evidence from bureaucratic choice settings" published at the *Journal of European Public Policy*.

This document presents the analysis of the YouGov survey on bureaucratic discrimination. Starting from the original YouGov survey, it performs the data cleaning operations, and performs the analysis of the model reported in the article.

The experimental design is based on a conjoint analysis with the following characteristics:

- Single choice, without the possibility of not choosing any profile.
- 5 attributes (nationality, language proficiency, gender, profession and age)
- 2 levels per attribute, except age (with 3 levels)
- 6 sets per outcome (Welfare / Rights)
- 2 cards per set

2

Data cleaning

Import the original SPSS dataset that contains the following number of observations and variables

```
source("load_packages.R")
library(foreign)
orig ← read.spss("data/DE22775503_Discrimination-of-EU-Citizens-by-Bureaucracy_Data-Set_2020-03-23.sav",
  to.data.frame = TRUE)
dim(orig)
```

```
→ [1] 3778  298
```

```
names(orig)
```

```
→   [1] "RecordNo"                    "weight"
→   [3] "gender"                      "age_int"
→   [5] "age_quota"                   "sta"
→   [7] "educ_neu"                    "voceduc_neu"
→   [9] "emps"                        "emps_quote"
→  [11] "work_sector_2019"            "public_sector_type"
→  [13] "target"                      "sample"
→  [15] "q1"                          "q2_1"
→  [17] "q2_2"                        "q2_3"
→  [19] "q2_4"                        "q2_5"
→  [21] "q2_6"                        "q2_7"
→  [23] "q2_8"                        "q2_66"
→  [25] "q2_99"                       "Treatment"
→  [27] "first_conjoint"              "V1_1_attribute_1"
→  [29] "V1_1_attribute_2"            "V1_1_attribute_3"
→  [31] "V1_1_attribute_4"            "V1_1_attribute_5"
→  [33] "V1_2_attribute_1"            "V1_2_attribute_2"
→  [35] "V1_2_attribute_3"            "V1_2_attribute_4"
→  [37] "V1_2_attribute_5"            "V1_3_attribute_1"
→  [39] "V1_3_attribute_2"            "V1_3_attribute_3"
→  [41] "V1_3_attribute_4"            "V1_3_attribute_5"
→  [43] "V1_4_attribute_1"            "V1_4_attribute_2"
→  [45] "V1_4_attribute_3"            "V1_4_attribute_4"
→  [47] "V1_4_attribute_5"            "V1_5_attribute_1"
→  [49] "V1_5_attribute_2"            "V1_5_attribute_3"
```

```
→  [51] ”V1_5_attribute_4”              ”V1_5_attribute_5”
→  [53] ”V1_6_attribute_1”              ”V1_6_attribute_2”
→  [55] ”V1_6_attribute_3”              ”V1_6_attribute_4”
→  [57] ”V1_6_attribute_5”              ”V1_7_attribute_1”
→  [59] ”V1_7_attribute_2”              ”V1_7_attribute_3”
→  [61] ”V1_7_attribute_4”              ”V1_7_attribute_5”
→  [63] ”V1_8_attribute_1”              ”V1_8_attribute_2”
→  [65] ”V1_8_attribute_3”              ”V1_8_attribute_4”
→  [67] ”V1_8_attribute_5”              ”V1_9_attribute_1”
→  [69] ”V1_9_attribute_2”              ”V1_9_attribute_3”
→  [71] ”V1_9_attribute_4”              ”V1_9_attribute_5”
→  [73] ”V1_10_attribute_1”             ”V1_10_attribute_2”
→  [75] ”V1_10_attribute_3”             ”V1_10_attribute_4”
→  [77] ”V1_10_attribute_5”             ”V1_11_attribute_1”
→  [79] ”V1_11_attribute_2”             ”V1_11_attribute_3”
→  [81] ”V1_11_attribute_4”             ”V1_11_attribute_5”
→  [83] ”V1_12_attribute_1”             ”V1_12_attribute_2”
→  [85] ”V1_12_attribute_3”             ”V1_12_attribute_4”
→  [87] ”V1_12_attribute_5”             ”V1a_1”
→  [89] ”V1bb_1”                        ”V1b_1”
→  [91] ”V1c_1”                         ”V1a_2”
→  [93] ”V1bb_2”                        ”V1b_2”
→  [95] ”V1c_2”                         ”V1a_3”
→  [97] ”V1bb_3”                        ”V1b_3”
→  [99] ”V1c_3”                         ”V1a_4”
→ [101] ”V1bb_4”                        ”V1b_4”
→ [103] ”V1c_4”                         ”V1a_5”
→ [105] ”V1bb_5”                        ”V1b_5”
→ [107] ”V1c_5”                         ”V1a_6”
→ [109] ”V1bb_6”                        ”V1b_6”
→ [111] ”V1c_6”                         ”q3_1”
→ [113] ”q3_2”                          ”q3_99”
→ [115] ”q3t1”                          ”q3t2”
→ [117] ”q4”                            ”q5”
→ [119] ”q6_1”                          ”q6_2”
→ [121] ”q6_99”                         ”q6t1”
→ [123] ”q6t2”                          ”q7_1”
→ [125] ”q7_2”                          ”q7_3”
→ [127] ”q7_4”                          ”q7_5”
→ [129] ”q7_6”                          ”q7_99”
→ [131] ”q7t”                           ”Q7T0”
→ [133] ”Q7T1”                          ”Q7T2”
→ [135] ”V2_1_attribute_1”              ”V2_1_attribute_2”
→ [137] ”V2_1_attribute_3”              ”V2_1_attribute_4”
→ [139] ”V2_1_attribute_5”              ”V2_2_attribute_1”
→ [141] ”V2_2_attribute_2”              ”V2_2_attribute_3”
→ [143] ”V2_2_attribute_4”              ”V2_2_attribute_5”
→ [145] ”V2_3_attribute_1”              ”V2_3_attribute_2”
→ [147] ”V2_3_attribute_3”              ”V2_3_attribute_4”
```

```
→ [149] ”V2_3_attribute_5”              ”V2_4_attribute_1”
→ [151] ”V2_4_attribute_2”              ”V2_4_attribute_3”
→ [153] ”V2_4_attribute_4”              ”V2_4_attribute_5”
→ [155] ”V2_5_attribute_1”              ”V2_5_attribute_2”
→ [157] ”V2_5_attribute_3”              ”V2_5_attribute_4”
→ [159] ”V2_5_attribute_5”              ”V2_6_attribute_1”
→ [161] ”V2_6_attribute_2”              ”V2_6_attribute_3”
→ [163] ”V2_6_attribute_4”              ”V2_6_attribute_5”
→ [165] ”V2_7_attribute_1”              ”V2_7_attribute_2”
→ [167] ”V2_7_attribute_3”              ”V2_7_attribute_4”
→ [169] ”V2_7_attribute_5”              ”V2_8_attribute_1”
→ [171] ”V2_8_attribute_2”              ”V2_8_attribute_3”
→ [173] ”V2_8_attribute_4”              ”V2_8_attribute_5”
→ [175] ”V2_9_attribute_1”              ”V2_9_attribute_2”
→ [177] ”V2_9_attribute_3”              ”V2_9_attribute_4”
→ [179] ”V2_9_attribute_5”              ”V2_10_attribute_1”
→ [181] ”V2_10_attribute_2”             ”V2_10_attribute_3”
→ [183] ”V2_10_attribute_4”             ”V2_10_attribute_5”
→ [185] ”V2_11_attribute_1”             ”V2_11_attribute_2”
→ [187] ”V2_11_attribute_3”             ”V2_11_attribute_4”
→ [189] ”V2_11_attribute_5”             ”V2_12_attribute_1”
→ [191] ”V2_12_attribute_2”             ”V2_12_attribute_3”
→ [193] ”V2_12_attribute_4”             ”V2_12_attribute_5”
→ [195] ”V2a_1”                         ”V2bb_1”
→ [197] ”V2b_1”                         ”V2c_1”
→ [199] ”V2a_2”                         ”V2bb_2”
→ [201] ”V2b_2”                         ”V2c_2”
→ [203] ”V2a_3”                         ”V2bb_3”
→ [205] ”V2b_3”                         ”V2c_3”
→ [207] ”V2a_4”                         ”V2bb_4”
→ [209] ”V2b_4”                         ”V2c_4”
→ [211] ”V2a_5”                         ”V2bb_5”
→ [213] ”V2b_5”                         ”V2c_5”
→ [215] ”V2a_6”                         ”V2bb_6”
→ [217] ”V2b_6”                         ”V2c_6”
→ [219] ”q8”                            ”q8a”
→ [221] ”Q8A0”                          ”Q8A1”
→ [223] ”Q8A2”                          ”q9”
→ [225] ”leftright_neu_shift”           ”migration”
→ [227] ”migration_country_1”           ”migration_country_2”
→ [229] ”migration_country_3”           ”migration_country_4”
→ [231] ”migration_country_5”           ”migration_country_6”
→ [233] ”migration_country_7”           ”migration_country_8”
→ [235] ”migration_country_9”           ”migration_country_10”
→ [237] ”migration_country_11”          ”migration_country_955”
→ [239] ”migration_cu_recode”           ”attitude_eu”
→ [241] ”attitude_immigration”          ”big5_CP_Agreeableness”
→ [243] ”big5_CP_Agreeableness_rc”      ”big5_CP_Conscientiousness”
→ [245] ”big5_CP_Conscientiousness_rc” ”big5_CP_Extraversion”
```

```
→ [247] "big5_CP_Extraversion_rc"      "big5_CP_Neuroticism"
→ [249] "big5_CP_Neuroticism_rc"       "big5_CP_Openness"
→ [251] "big5_CP_Openness_rc"          "pinc"
→ [253] "hinc"                         "starttime"
→ [255] "START0"                       "START1"
→ [257] "START2"                       "endtime"
→ [259] "ENDTI0"                       "ENDTI1"
→ [261] "ENDTI2"                       "time_p_age_gender"
→ [263] "time_p_sta"                   "time_p_educ_neu"
→ [265] "time_p_voceduc_neu"           "time_p_emps"
→ [267] "time_p_work_sector_2019"      "time_p_public_sector_type"
→ [269] "time_p_q1"                    "time_p_q2"
→ [271] "time_p_V1_Intro"              "time_p_V1_1"
→ [273] "time_p_V1_2"                  "time_p_V1_3"
→ [275] "time_p_V1_4"                  "time_p_V1_5"
→ [277] "time_p_V1_6"                  "time_p_q3"
→ [279] "time_p_q4"                    "time_p_q5"
→ [281] "time_p_q6"                    "time_p_q7"
→ [283] "time_p_V2_Intro"              "time_p_V2_1"
→ [285] "time_p_V2_2"                  "time_p_V2_3"
→ [287] "time_p_V2_4"                  "time_p_V2_5"
→ [289] "time_p_V2_6"                  "time_p_q8"
→ [291] "time_p_q8a"                   "time_p_q9"
→ [293] "time_p_info_1"                "time_p_leftright_neu_shift"
→ [295] "time_p_migration"             "time_p_migration_country"
→ [297] "time_p_attitude_eu"           "time_p_attitude_immigration"
```

Delete the first 25 observations that correspond to individuals to whom the old questionnaire (YouGov test) was administered.

```
orig ← orig[-c(1:25),]

# Write some specific functions to better clarify data cleaning part later
clean.ideology ← function(x) {
  x ← as.character(x)
  x[x == "links"] ← "0"
  x[x == "rechts"] ← "10"
  return(as.integer(as.numeric(as.character(x))))
}
clean.education ← function(x) {
  x ← as.character(x)
  x[x == "Ohne Schulabschluss"] ← "Primary"
  x[x == "Haupt-(Volks-)schulabschluss"] ← "Primary"
  x[x == "Noch in schulischer Ausbildung"] ← "Still studying"
  x[x == "Realschul- oder gleichwertiger Abschluss (POS, Mittlere Reife)"] ← "Secondary"
  x[x == "Abitur, Fachhochschulreife"] ← "Tertiary"
  x[x == "keine Angabe"] ← NA
  return(factor(x, levels = c("Primary",
                              "Secondary",
                              "Still studying",
```

```r
                              "Tertiary")))
}
clean.attitude <- function(x) {
  x[x == "Weiß nicht / keine Angabe"] <- NA
  return(as.integer(as.numeric(x)))
}
clean.citizenship <- function(x) {
  x <- as.character(x)
  x[x == "Weiß nicht / keine Angabe"] <- NA
  x[x != "Besitze nicht die deutsche Staatsbürgerschaft"] <- "0"
  x[x == "Besitze nicht die deutsche Staatsbürgerschaft"] <- "1"
  return(as.integer(as.numeric(x)))
}
clean.eurights <- function(x) {
  x[x == "weiß nicht / keine Angabe"] <- NA
  return(as.integer(as.numeric(x)))
}
clean.reader <- function(x) {
  x <- as.character(x)
  x[x == "Ja"] <- 1
  x[x == "Nein"] <- 0
  x[x == "Weiß nicht"] <- NA
  return(as.integer(as.numeric(x)))
}
clean.timespent <- function(h, m, na, H, M) {
  time <- rep(NA, length(h))
  H[is.na(H)] <- 0
  M[is.na(M)] <- 0
  time <- (H * 60 + M) / 60
  time[na == "Yes"] <- NA
  return(time)
}
clean.socialnetworks <- function(data, vars, no.sn, na, other) {
  sn <- rep(0, length(na))
  SN <- data[,vars]
  sn.M <- apply(SN, 1, function(x) ifelse(length(which(x == "Yes")) > 0, TRUE, FALSE))
  sn[sn.M] <- 1
  sn[data[no.sn] == "Yes"] <- 0
  sn[na == "Yes"] <- NA
  return(as.integer(sn))
}
```

Process individual-level variables and generate a tidy object (`I`):

- Translate variable names and values into English.
- Calculate time taken to complete the survey using start and end times (`Time`, in minutes).
- Name `Outcomes` to welfare and rights, instead of first and second conjoint.
- Recoding missing data properly, instead of one category of the replies

("keine Angabe").

- Select only some variables, the ones that we will work with.
- Generate an individual identificator to match with all other objects (id).

```
I <- orig %>%  # Individual-level variables
  tibble::as_tibble() %>%
  # Generate a unique individual ID (id) used as a main reference
  # Gets its value from the position of the individual
  # Avoid using the one given in the sample, as it starts with 0.
  mutate(id = 1:n()) %>%
  mutate(Gender = as.factor(ifelse(gender == "männlich", "Male", "Female"))) %>%
  mutate(Age = as.integer(age_int)) %>%
  mutate(Education = clean.education(educ_neu)) %>%
  #mutate(Target = target) %>%
  # Convert times to proper time format in R
  mutate(endtime = as.POSIXct(as.character(str_sub(endtime, 1, 19)), "%Y-%m-%d %H:%M:%S")) %>%
  mutate(starttime = as.POSIXct(as.character(str_sub(starttime, 1, 19)), "%Y-%m-%d %H:%M:%S")) %>%
  mutate(Time = as.numeric(as.character(difftime(endtime, starttime)))) %>%
  mutate(TimeCompleted = endtime) %>%
  mutate(DateCompleted = as.Date(endtime)) %>%
  mutate(Period = as.integer(as.numeric(DateCompleted - min(DateCompleted) + 1))) %>%
  # Give meaningful names to population groups
  mutate(Population = as.factor(ifelse(as.character(target) == 'Nicht "Öffentliche Verwaltung"',
                                      "General population", "Public administration"))) %>%
  mutate(Treatment = as.factor(ifelse(as.character(Treatment) == "Treatment gezeigt", "Shown", "Not shown")
  mutate(Migration = as.factor(ifelse(migration == "ja", "Yes", "No"))) %>%
  # Personality traits are simply converted into integers
  mutate(Agreeableness = as.integer(as.numeric(big5_CP_Agreeableness))) %>%
  mutate(Conscientiousness = as.integer(as.numeric(big5_CP_Conscientiousness))) %>%
  mutate(Extraversion = as.integer(as.numeric(big5_CP_Extraversion))) %>%
  mutate(Neuroticism = as.integer(as.numeric(big5_CP_Neuroticism))) %>%
  mutate(Openess = as.integer(as.numeric(big5_CP_Openness))) %>%
  mutate(Income = hinc) %>%
  mutate(Ideology = clean.ideology(leftright_neu_shift)) %>%
  mutate(`Immigration, favourable` = clean.attitude(attitude_immigration)) %>%
  mutate(`EU, favourable` = clean.attitude(attitude_eu)) %>%
  mutate(`No german citizenship` = clean.citizenship(q8)) %>%
  mutate(`EU, same rights` = clean.eurights(q9)) %>%
  mutate(`Political interest` = clean.timespent(q3_1, q3_2, q3_99, q3t1, q3t2)) %>%
  mutate(`Book reader` = clean.reader(q5)) %>%
  mutate(`Press reader` = clean.reader(q4)) %>%
  mutate(`Internet use` = clean.timespent(q6_1, q6_2, q6_99, q6t1, q6t2)) %>%
  mutate(`On social networks` = clean.socialnetworks(
    data = .,
    vars = paste0("q7_", 1:5),
    no.sn = "q7_6",
    na = "q7_99",
    other = c("q7t", "Q7T0", "Q7T1", "Q7T2"))) %>%
  #
```

```
  # Manually check that C1 is for welfare ind C2 for rights, internaly in the
  # YouGov surey
  mutate(`FirstShown` = as.factor(ifelse(first_conjoint == "Conjoint 1 (ALG2) zuerst",
                                 "Welfare", "Rights"))) %>%
  mutate(Region = sta) %>%
  # Also possible to clean:
  # public_sector_type
  # work_sector_2019
  # emps
  select(id, Time, TimeCompleted, DateCompleted, Period, FirstShown,
         Population, Treatment,
         #Target,
         Region,
         Agreeableness, Conscientiousness, Extraversion,
         Neuroticism, Openess,
         Gender, Age, Education,
         Ideology, Migration,
         `Immigration, favourable`, `EU, favourable`,
         `No german citizenship`, `EU, same rights`,
         `Political interest`,
         `Book reader`, `Press reader`,
         `Internet use`, `On social networks`,
         Income)

# Recode non-responses
levels(I$Income)[levels(I$Income) == "keine Angabe"] <- NA

# Indicate types of variables
personality.variables <- c("Agreeableness", "Conscientiousness", "Extraversion",
                           "Neuroticism", "Openess")
```

Example of the structure and content of the `I` object

```
I
```

```
-> # A tibble: 3,753 x 29
->       id  Time TimeCompleted       DateCompleted Period FirstShown Population
->    <int> <dbl> <dttm>              <date>         <int> <fct>      <fct>
->  1     1  6.15 2020-02-27 09:29:27 2020-02-27         1 Rights     General popu~
->  2     2 11.4  2020-02-27 10:24:51 2020-02-27         1 Rights     General popu~
->  3     3  4.95 2020-02-27 09:24:14 2020-02-27         1 Rights     General popu~
->  4     4  8.6  2020-02-27 09:31:46 2020-02-27         1 Rights     General popu~
->  5     5 13.1  2020-02-27 10:00:44 2020-02-27         1 Rights     General popu~
->  6     6  3.48 2020-02-27 09:52:13 2020-02-27         1 Rights     General popu~
->  7     7  7.4  2020-02-27 09:52:07 2020-02-27         1 Welfare    General popu~
->  8     8  7.95 2020-02-27 09:54:56 2020-02-27         1 Rights     General popu~
->  9     9  6.82 2020-02-27 09:32:57 2020-02-27         1 Rights     General popu~
-> 10    10 13.8  2020-02-27 10:03:53 2020-02-27         1 Rights     General popu~
-> # ... with 3,743 more rows, and 22 more variables: Treatment <fct>,
-> #   Region <fct>, Agreeableness <int>, Conscientiousness <int>,
```

```
→ #   Extraversion <int>, Neuroticism <int>, Openess <int>, Gender <fct>,
→ #   Age <int>, Education <fct>, Ideology <int>, Migration <fct>,
→ #   Immigration, favourable <int>, EU, favourable <int>,
→ #   No german citizenship <int>, EU, same rights <int>,
→ #   Political interest <dbl>, Book reader <int>, Press reader <int>,
→ #   Internet use <dbl>, On social networks <int>, Income <fct>
```

Descriptive statistics of the object containing individual data.

```
summary(I)
```

```
→        id             Time        TimeCompleted
→   Min.   :   1   Min.   :    3   Min.   :2020-02-27 09:21:41
→   1st Qu.: 939   1st Qu.:    7   1st Qu.:2020-02-27 16:08:52
→   Median :1877   Median :    9   Median :2020-02-27 20:46:01
→   Mean   :1877   Mean   :  115   Mean   :2020-03-02 01:03:45
→   3rd Qu.:2815   3rd Qu.:   13   3rd Qu.:2020-03-04 18:34:06
→   Max.   :3753   Max.   :24024   Max.   :2020-03-16 13:49:50
→
→   DateCompleted            Period        FirstShown
→   Min.   :2020-02-27   Min.   : 1.00   Rights :1890
→   1st Qu.:2020-02-27   1st Qu.: 1.00   Welfare:1863
→   Median :2020-02-27   Median : 1.00
→   Mean   :2020-03-01   Mean   : 4.44
→   3rd Qu.:2020-03-04   3rd Qu.: 7.00
→   Max.   :2020-03-16   Max.   :19.00
→
→                   Population       Treatment                   Region
→   General population    :2974   Not shown:1878   Nordrhein-Westfalen: 887
→   Public administration: 779   Shown    :1875   Bayern             : 501
→                                                  Baden-Württemberg  : 389
→                                                  Niedersachsen      : 343
→                                                  Hessen             : 273
→                                                  Sachsen            : 226
→                                                  (Other)            :1134
→   Agreeableness   Conscientiousness  Extraversion    Neuroticism      Openess
→   Min.   : 4.0   Min.   : 7.0      Min.   : 4.0   Min.   : 4.0   Min.   : 4.0
→   1st Qu.:16.0   1st Qu.:19.0      1st Qu.:10.0   1st Qu.:10.0   1st Qu.:16.0
→   Median :19.0   Median :22.0      Median :16.0   Median :13.0   Median :19.0
→   Mean   :19.5   Mean   :21.3      Mean   :14.6   Mean   :13.3   Mean   :18.6
→   3rd Qu.:22.0   3rd Qu.:25.0      3rd Qu.:16.0   3rd Qu.:16.0   3rd Qu.:22.0
→   Max.   :28.0   Max.   :28.0      Max.   :28.0   Max.   :28.0   Max.   :28.0
→
→     Gender          Age                 Education        Ideology      Migration
→   Female:1942   Min.   :18.0   Primary        : 754   Min.   : 0.00   No  :3355
→   Male  :1811   1st Qu.:36.0   Secondary      :1357   1st Qu.: 3.00   Yes : 386
→                 Median :51.0   Still studying:  53   Median : 5.00   NA's:  12
→                 Mean   :48.7   Tertiary       :1589   Mean   : 4.65
→                 3rd Qu.:61.0                          3rd Qu.: 5.00
→                 Max.   :91.0                          Max.   :10.00
```

```
→                                                       NA's    :3
→  Immigration, favourable EU, favourable No german citizenship EU, same rights
→  Min.   :1.0             Min.   :1.0     Min.   :0.00          Min.   :1.0
→  1st Qu.:2.0             1st Qu.:2.0     1st Qu.:0.00          1st Qu.:3.0
→  Median :2.0             Median :3.0     Median :0.00          Median :3.0
→  Mean   :2.4             Mean   :3.1     Mean   :0.02          Mean   :3.2
→  3rd Qu.:3.0             3rd Qu.:4.0     3rd Qu.:0.00          3rd Qu.:4.0
→  Max.   :5.0             Max.   :4.0     Max.   :1.00          Max.   :4.0
→  NA's   :454             NA's   :400     NA's   :38            NA's   :273
→  Political interest  Book reader    Press reader    Internet use
→  Min.   : 0.0       Min.   :0.00   Min.   :0.00   Min.   : 0.0
→  1st Qu.: 0.5       1st Qu.:0.00   1st Qu.:0.00   1st Qu.: 2.0
→  Median : 1.0       Median :1.00   Median :1.00   Median : 3.0
→  Mean   : 1.3       Mean   :0.67   Mean   :0.56   Mean   : 4.1
→  3rd Qu.: 1.8       3rd Qu.:1.00   3rd Qu.:1.00   3rd Qu.: 5.0
→  Max.   :24.0       Max.   :1.00   Max.   :1.00   Max.   :24.0
→  NA's   :724        NA's   :38     NA's   :74     NA's   :433
→  On social networks                          Income
→  Min.   :0.0        EUR 2.000 bis unter EUR 2.500: 401
→  1st Qu.:0.0        EUR 1.500 bis unter EUR 2.000: 366
→  Median :1.0        EUR 2.500 bis unter EUR 3.000: 366
→  Mean   :0.7        EUR 1.000 bis unter EUR 1.500: 359
→  3rd Qu.:1.0        EUR 3.000 bis unter EUR 3.500: 322
→  Max.   :1.0        (Other)                      :1208
→  NA's   :64         NA's                         : 731
```

Process individual-level variables associated with time taken to complete each of the questions of the survey and store them into a tidy object (`I.time`).

- Calculate time taken to complete several relevant parts (`time`, in seconds).
- Assign the concrete `Outcome` and `Exercise`.

```
I.time <- orig %>%  # Individual-level variables
  tibble::as_tibble() %>%
  mutate(id = 1:n()) %>%
  select(id, starts_with("time_p_V")) %>%
  # Move from a wide format where there are many variables into a long format
  # where previous variable names get a variable (Variable), to help processing
  # and cleaning them.
  pivot_longer(-id, names_to = "Variable", values_to = "time") %>%
  # The original names of the columns are separated with "_", and therefore we
  # can take advantage of this by giving each part a different variable name.
  separate(Variable, sep = "_", into = c("Nothing1", "Nothing2", "Outcome", "Exercise")) %>%
  filter(Exercise != "neu") %>%
  mutate(Exercise = as.factor(ifelse(Exercise == "Intro", "Reading", Exercise))) %>%
  # As previously, manually confirm that V1 belongs to Welfare and V2 to Rights
  mutate(Outcome = as.factor(ifelse(Outcome == "V1", "Welfare", "Rights"))) %>%
  select(id, Outcome, Exercise, time)
```

```
# There are also variables START1 .. ENDTI0
# that probably allow us to know whether someone resumed the survey
# but are all empty
I.time
```

```
→ # A tibble: 52,542 x 4
→        id Outcome Exercise  time
→     <int> <fct>   <fct>    <dbl>
→  1      1 Welfare Reading   4.35
→  2      1 Welfare 1         7.98
→  3      1 Welfare 2         7.79
→  4      1 Welfare 3        11.4
→  5      1 Welfare 4         8.98
→  6      1 Welfare 5         8.78
→  7      1 Welfare 6        11.0
→  8      1 Rights  Reading   1.64
→  9      1 Rights  1        14.9
→ 10      1 Rights  2         7.88
→ # … with 52,532 more rows
```

```
summary(I.time)
```

```
→        id          Outcome          Exercise          time
→  Min.   :   1   Rights :26271   1      :7506   Min.   :    0
→  1st Qu.: 939   Welfare:26271   2      :7506   1st Qu.:    7
→  Median :1877                   3      :7506   Median :   11
→  Mean   :1877                   4      :7506   Mean   :   20
→  3rd Qu.:2815                   5      :7506   3rd Qu.:   18
→  Max.   :3753                   6      :7506   Max.   :21941
→                                 Reading:7506
```

Process the experimental data and variables.

- Each observation is a response (`Decision`, 0 or 1, `Score` for the continuous rate) to one of the six `Exercise`(s) (1:6) that exist in each of the two `Outcome` (Welfare, Rights).
- Merge with the `Population` (General population or Public administration), the `Treatment` (Shown or not shown)
- Store the order in which each profile has been shown (`ProfileOrder`).
- Clean the features of the profiles (`Nationality`, `Gender`, `Language`, `Profession`, `Age`).

```
# This object stores all the experimental data
E.full ← orig %>% # Experiment-level variables
  tibble::as_tibble() %>%
  mutate(id = 1:n()) %>%
  # Select onli variables starting with "V" (except voceduc_neu), that
  # correspond to the experimental data
  select(id, starts_with("V")) %>%
  select(-voceduc_neu) %>%
```

```
  # Instead of a long list of variables, move them to the long format where a
  # new variable (Variable) gets their previous variable name.
  pivot_longer(-id, names_to = "Variable", values_to = "value")

# With this object E.full we now differentiate the cleaning of the attributes of
# the profiles shown (E.profiles) and the cleaning of the responses (E.rates).

# Clean the profiles of the options presented
E.profiles <- E.full %>%
  # These are the variables that are named 'attribute'
  filter(str_detect(Variable, "attribute")) %>%
  separate(Variable, sep = "_", into = c("Outcome", "Profile", "Nothing", "Feature")) %>%
  select(-Nothing) %>%
  mutate(Outcome = as.factor(ifelse(Outcome == "V1", "Welfare", "Rights"))) %>%
  mutate(Profile = as.integer(as.numeric(Profile))) %>%
  # check if Profile is odd or even, for assigning its order
  mutate(ProfileOrder = as.integer(ifelse(Profile %% 2 != 0, 1, 2))) %>%
  # Manually assign Profiles 1 and 2 to the first Exercise, 3 and 4 to the
  # second exercise, etc...
  # A bit manual and rustic, but we need it in this format.
  mutate(Exercise = case_when(
           Profile %in% 1:2 ~ 1L,
           Profile %in% 3:4 ~ 2L,
           Profile %in% 5:6 ~ 3L,
           Profile %in% 7:8 ~ 4L,
           Profile %in% 9:10 ~ 5L,
           Profile %in% 11:12 ~ 6L)) %>%
  # Manually map the features into our variable names.
  mutate(Variable = as.factor(recode(Feature,
                        `1` = "Nationality",
                        `2` = "Gender",
                        `3` = "Language",
                        `4` = "Profession",
                        `5` = "Age"))) %>%
  select(-Feature) %>%
  pivot_wider(names_from = Variable, values_from = value) %>%
  mutate(Nationality = as.factor(ifelse(Nationality == "Niederlande", "Netherlands", "Romania"))) %>%
  mutate(Gender = as.factor(ifelse(Gender == "Männlich", "Male", "Female"))) %>%
  mutate(Language = as.factor(ifelse(Language == "Fließend deutsch", "Full", "No"))) %>%
  mutate(Profession = as.factor(ifelse(Profession %in% c("Arzt", "Ärztin"), "Doctor", "Nurse"))) %>%
  mutate(Age = factor(Age)) %>%
  select(id, Outcome, Exercise, Profile, ProfileOrder, Nationality, Gender, Language, Profession, Age)

# Responses to the survey
E.rates <- E.full %>%
  # Now we want anything but the attributes of the profiles, namely the
  # responses
  filter(!str_detect(Variable, "attribute")) %>%
  separate(Variable, sep = "_", into = c("OutcomePlus", "Exercise")) %>%
```

```r
  mutate(Exercise = as.integer(as.numeric(Exercise))) %>%
  separate(OutcomePlus, sep = 2, into = c("Outcome", "Rating")) %>%
  # Again, manually check and assign V1 to welfare and V2 to rights.
  mutate(Outcome = as.factor(ifelse(str_detect(Outcome, "V1"), "Welfare", "Rights"))) %>%
  # Manually check that the letters used for ratings are indeed used for the
  # three responses.
  mutate(Rating = ifelse(Rating == "a", "Prioritization", Rating)) %>%
  mutate(Rating = ifelse(Rating == "b", "Profile 1", Rating)) %>%
  mutate(Rating = ifelse(Rating == "c", "Profile 2", Rating)) %>%
  mutate(Rating = ifelse(Rating == "bb", "Certainty", Rating)) %>%
  pivot_wider(names_from = Rating, values_from = value) %>%
  # Manually recode to whom the respondent has given the priority.
  mutate(Prioritization = as.integer(ifelse(Prioritization %in% c("Anfrage von Person 1",
                                                 "Antragsteller 1"), 1, 2))) %>%
  mutate(Certainty = as.character(Certainty)) %>%
  mutate(Certainty = ifelse(Certainty == "1 - Überhaupt nicht sicher", "1", Certainty)) %>%
  mutate(Certainty = ifelse(Certainty == "10 - Äußerst sicher", "10", Certainty)) %>%
  mutate(Certainty = as.integer(as.numeric(Certainty))) #%>%
  #select(-c(`Profile 1`, `Profile 2`))
  #   # For the numerical ratings, it is a bit trickier and has to be done for the
  #   # two responses.
  #   mutate(`Profile 1` = as.character(`Profile 1`)) %>%
  #   mutate(`Profile 1` = ifelse(`Profile 1` == "1 - Überhaupt nicht wahrscheinlich", "1", `Profile 1`)) %
  #   mutate(`Profile 1` = ifelse(`Profile 1` == "7 - Äußerst wahrscheinlich", "7", `Profile 1`)) %>%
  #   mutate(`Profile 1` = as.integer(as.numeric(`Profile 1`))) %>%
  #   mutate(`Profile 2` = as.character(`Profile 2`)) %>%
  #   mutate(`Profile 2` = ifelse(`Profile 2` == "1 - Überhaupt nicht wahrscheinlich", "1", `Profile 2`)) %
  #   mutate(`Profile 2` = ifelse(`Profile 2` == "7 - Äußerst wahrscheinlich", "7", `Profile 2`)) %>%
  #   mutate(`Profile 2` = as.integer(as.numeric(`Profile 2`)))


# Now match rates with profiles.
# Very dirty, but there is no way to automatize, because there is no association
# between variable names and experiment/profile taken
# So first we must join profiles with binary ratings and assign the decision
# and then join this with the numerical rates to assign the ratings.
E <- E.profiles %>%
  # Start with profiles and join them with rates for the binary prioritization.
  left_join(select(E.rates, id, Outcome, Exercise, Prioritization, Certainty)) %>%
  # The binary prioritization is assessed here where the order in which the
  # profile has been shown matches the prioritized one.
  mutate(Decision = as.integer(ifelse(ProfileOrder == Prioritization, 1, 0))) %>%
  # Now join also the rates of the numerical scores.
  # Although they are deleted, because are not asked in the survey.
  left_join(E.rates %>%
            select(-Prioritization, -Certainty) %>%
            pivot_longer(-c(id, Outcome, Exercise), names_to = "Profile", values_to = "Score") %>%
            mutate(ProfileOrder = as.integer(ifelse(Profile == "Profile 1", 1, 2))) %>%
            select(-Profile, -Score))
```

```
# Also include the Population and the Treatment in the experiment data, for
# convenience, although in terms of cleanliness, these variables belong to the
# individual-level data.
E ← right_join(select(I, id, Population, Treatment), E)

feature.variables ← c("Nationality", "Gender", "Language",
                       "Profession", "Age")

E

→ # A tibble: 90,072 x 15
→       id Population  Treatment Outcome Exercise Profile ProfileOrder Nationality
→    <int> <fct>       <fct>     <fct>      <int>   <int>        <int> <fct>
→  1     1 General po~ Not shown Welfare        1       1            1 Romania
→  2     1 General po~ Not shown Welfare        1       2            2 Romania
→  3     1 General po~ Not shown Welfare        2       3            1 Romania
→  4     1 General po~ Not shown Welfare        2       4            2 Netherlands
→  5     1 General po~ Not shown Welfare        3       5            1 Romania
→  6     1 General po~ Not shown Welfare        3       6            2 Romania
→  7     1 General po~ Not shown Welfare        4       7            1 Romania
→  8     1 General po~ Not shown Welfare        4       8            2 Netherlands
→  9     1 General po~ Not shown Welfare        5       9            1 Netherlands
→ 10     1 General po~ Not shown Welfare        5      10            2 Netherlands
→ # ... with 90,062 more rows, and 7 more variables: Gender <fct>,
→ #   Language <fct>, Profession <fct>, Age <fct>, Prioritization <int>,
→ #   Certainty <int>, Decision <int>

summary(E)

→        id                       Population         Treatment        Outcome
→  Min.   :   1   General population    :71376   Not shown:45072   Rights :45036
→  1st Qu.: 939   Public administration:18696   Shown    :45000   Welfare:45036
→  Median :1877
→  Mean   :1877
→  3rd Qu.:2815
→  Max.   :3753
→     Exercise       Profile        ProfileOrder      Nationality       Gender
→  Min.   :1.0   Min.   : 1.00   Min.   :1.0   Netherlands:44858   Female:45155
→  1st Qu.:2.0   1st Qu.: 3.75   1st Qu.:1.0   Romania    :45214   Male  :44917
→  Median :3.5   Median : 6.50   Median :1.5
→  Mean   :3.5   Mean   : 6.50   Mean   :1.5
→  3rd Qu.:5.0   3rd Qu.: 9.25   3rd Qu.:2.0
→  Max.   :6.0   Max.   :12.00   Max.   :2.0
→  Language       Profession     Age          Prioritization   Certainty
→  Full:44875   Doctor:44989   25:30025   Min.   :1.00   Min.   : 1.00
→  No  :45197   Nurse :45083   40:29954   1st Qu.:1.00   1st Qu.: 5.00
→                              55:30093   Median :1.00   Median : 7.00
→                                         Mean   :1.46   Mean   : 6.88
→                                         3rd Qu.:2.00   3rd Qu.: 9.00
```

```
→                                         Max.   :2.00   Max.   :10.00
→      Decision
→    Min.   :0.0
→    1st Qu.:0.0
→    Median :0.5
→    Mean   :0.5
→    3rd Qu.:1.0
→    Max.   :1.0
```

```
orig %>%
  select(V1a_1, endtime, V1b_1, V1c_1, V1bb_1) %>%
  slice(1:30)

orig %>%
  select(V1a_1, V1b_1, V1c_1, V1bb_1) %>%
  slice((n()-5):n())

save(orig,
     I, I.time, personality.variables,
     E, feature.variables,
     file = "yougov.RData")

source("load_packages.R")
load("yougov.RData")
run.pcp ← TRUE
or ← function(x, significant = 2) {
  or ← as.character(signif((x - 1) * 100, significant))
  #or[or < 0] ← paste0("▯ ", str_replace(or[or < 0], "-", ""), "%")
  or[or < 0] ← paste0("\U25Bd ", str_replace(or[or < 0], "-", ""), "%")
  #or[or > 0] ← paste0("▯ ", or[or > 0], "%")
  or[or > 0] ← paste0("\U25B3 ", or[or > 0], "%")
  or[or == 0] ← "="
  return(or)
}
inv.logit ← function(x) return(1 / (1 + (exp(-(x)))))
```

3
Model: decision, baseline

The reference profiled individual has the following characteristics:

- Romanian
- Female
- No German proficiency
- Doctor
- 40 years old

Therefore, the values of the Intercept correspond to this profile.

```
# Use a reference dataset different from the original,
# in case we want to apply filters or perform transformations.
d ← E %>%
  mutate(Observation = 1:n())

# Define the outcome variable and calculate its size.
Y ← d$Decision
n0 ← length(Y)

# The profile factors are cleaned, establishing reference categories and
# selecting the relevant variables that will be passed onto the final matrix.
# Store their names to be used later in the figures.
X ← d %>%
  mutate(`Nationality Netherlands` = ifelse(Nationality == "Netherlands", 1, 0)) %>%
  mutate(`Gender Male` = ifelse(Gender == "Male", 1, 0)) %>%
  mutate(`Profession Nurse` = ifelse(Profession == "Nurse", 1, 0)) %>%
  mutate(`Language Full` = ifelse(Language == "Full", 1, 0)) %>%
  mutate(`Age 25` = ifelse(Age == "25", 1, 0)) %>%
  mutate(`Age 55` = ifelse(Age == "55", 1, 0)) %>%
  mutate(`First shown` = ifelse(ProfileOrder == 1, 1, 0)) %>%
  select(`Nationality Netherlands`,
         `Gender Male`,
         `Profession Nurse`,
         `Language Full`,
         `Age 25`,
         `Age 55`,
         `First shown`) %>%
  as.matrix()
```

```
X ← cbind("(Intercept)" = 1, X)
nF ← dim(X)[[2]]
feature.label ← dimnames(X)[[2]]
feature.label.order ← c("Nationality Netherlands",
                         "Language Full",
                         "Profession Nurse",
                         "Age 25",
                         "Age 55",
                         "Gender Male")

# Define a matrix for later use in multivariate normal priors.
b0 ← rep(0, nF)
B0 ← diag(nF)
diag(B0) ← 2.5^-2

# Identify outcome, exercise, population and treatment with numbers,
# count their size and keep labels for later.
id.outcome ← as.numeric(d$Outcome)
outcome.label ← levels(d$Outcome)
nOutcome ← length(outcome.label)

id.exercise ← d$Exercise
exercise.label ← sort(unique(d$Exercise))
nE ← length(exercise.label)

id.population ← as.numeric(d$Population)
population.label ← levels(d$Population)
nP ← length(population.label)

id.treatment ← as.numeric(d$Treatment)
treatment.label ← levels(d$Treatment)
nT ← length(treatment.label)

first.shown.profile ← ifelse(d$ProfileOrder == 1, 1, 0)

# Store all the variables needed by JAGS in a single list
D ← list(
  nO = nO,
  id_outcome = id.outcome, nOutcome = nOutcome,
  id_exercise = id.exercise , nE = nE,
  id_population = id.population, nP = nP,
  id_treatment = id.treatment, nT = nT,
  first_shown_profile = first.shown.profile,
  X = unname(X), nF = nF, b0 = b0, B0 = B0,
  Y = Y)
```

Model description:

$$
\begin{aligned}
Y_o &\sim \mathcal{B}(\pi_o) && \text{Main data component} \\
\pi_o &= \text{logit}(\theta_{O,p,t,f} F_{o,f}) && \text{Linear relationship} \\
\theta_{O,p,t,f} &\sim \mathcal{N}(\Theta_{O,f}, \sigma_{\theta_{O,f}}) && \text{Priors for explanatory variables} \\
\Theta_{O,f} &\sim \mathcal{N}(\Omega_f, \sigma_{\Theta_f}) && \text{Prior for the effects shared by outcome} \\
\sigma_{\theta_{O,f}} &\sim \mathcal{IG}(1,1) && \text{Prior for SD, pooling of population and treatment} \\
\sigma_{\Theta_f} &\sim \mathcal{IG}(1,1) && \text{Prior for SD, pooling of outcome} \\
\Omega_f &\sim \mathcal{N}(0,1) && \text{Prior for the effects of every feature}
\end{aligned}
$$

Where:

- Y: Outcome variable capturing whether a profile has been prioritized (1) or not (0).
- o: Observation
- o: Outcomes (Welfare/ Rights)
- p: Population (General population / Public administration)
- t: Treatment (Shown / Not shown)
- F: Matrix with the observations of Features (the discrimination sources, plus intercept and first shown profile), for each experimental data point.
- f: Feature
- $\theta_{O,p,t,f}$: Main parameters of interest capturing the discrimination effects by outcome, population, treatment and profiles' feature.
- $\Theta_{O,f}$: Hyper-parameters capturing the shared effect of outcomes and features over population and treatment.
- Ω_f: Hyper-parameters capturing the shared effect of features over outcomes, population and treatment.
- $\sigma_{\theta_{O,f}}$: Between population/treatment and within outcome/feature standard deviations.
- σ_{Θ_f}: Between outcome and within feature standard deviations.

The $\sigma_{\theta_{O,f}}$ can also be seen as a pooling factor. The higher, the more freely are the effects of population and treatment to vary from the overall $\Theta_{O,f}$ shared by outcome and feature. If it is restricted to be very close to 0 then it assumes that there should be a lot of variation from the two populations and two treatments to make them different. Otherwise, they "borrow strength" from the overall means.

So far we have used a very relaxed pooling (almost none). A model with a restriction very close to zero is the "Cauchy" model.

This model estimates approximately the following number of parameters:

```
# theta
(nOutcome * nP * nT * nF) +
# Theta
(nOutcome * nF) +
# Omega
(nF) +
# sigma_theta
(nOutcome * nF) +
```

```
# sigma_Theta
(nF)
```

```
→ [1] 112
```

```
M ← "Decision, Baseline"
M.lab ← "decision-baseline"
# Define the JAGS model (Bugs language) in a character vector
# and write it in a file which name is taken from the labels that we just
# defined the model will be called.
m ← "
model {
  for (o in 1:nO) {
    Y[o] ~ dbern(p[o])
    logit(p[o]) ← inprod(theta[id_outcome[o],id_population[o],id_treatment[o],1:nF], X[o,1:nF])
  }
  # Priors for effects
  for (f in 1:nF) {
    for (ocm in 1:nOutcome) {
      for (p in 1:nP) {
        for (t in 1:nT) {
          #theta[ocm,p,t,f] ~ dnorm(Theta[ocm,f], sigma_theta[ocm,f]^-2)
          theta[ocm,p,t,f] ~ dnorm(Theta[ocm,f], tau_theta[ocm,f])
        }
      }
      #sigma_theta[ocm,f] ~ dt(0, 0.5^-2, 1)T(0,)
      tau_theta[ocm,f] ~ dgamma(1, 1)
      sigma_theta[ocm,f] ← 1 / sqrt(tau_theta[ocm,f])
      #Theta[ocm,f] ~ dnorm(Omega[f], sigma_Theta[f]^-2)
      Theta[ocm,f] ~ dnorm(Omega[f], tau_Theta[f])
    }
    #sigma_Theta[f] ~ dt(0, 0.5^-2, 1)T(0,)
    tau_Theta[f] ~ dgamma(1, 1)
    sigma_Theta[f] ← 1 / sqrt(tau_Theta[f])
  }
  Omega ~ dmnorm(b0, B0)
}
"
write(m, file = paste("models/model-", M.lab, ".bug", sep = ""))
# Define which parameters we want to get from the model.
par ← NULL
par ← c(par, "theta")
par ← c(par, "Theta", "sigma_theta")
par ← c(par, "Omega", "sigma_Theta")
#if (run.pcp) par ← c(par, "p")
par.pcp ← c("p")
# Set the sampling characteristics
adapt ← 5e2
burnin ← 1e3#1e4
run ← 2e3
```

```
n.pcp ← 50
chains ← 3
method ← "parallel"
thin ← 5

# Run the sampler and save its results
# This chunk only runs when required (eval = FALSE), so that we do not have to
# run the model every time
t0 ← proc.time()
#rj ← run.jags(model = paste("models/model-", M.lab, ".bug", sep = ""),
#                data = dump.format(D, checkvalid = FALSE),
#                inits = list(list(.RNG.name = "base::Super-Duper", .RNG.seed = 1),
#                             list(.RNG.name = "base::Super-Duper", .RNG.seed = 2),
#                             list(.RNG.name = "base::Wichmann-Hill", .RNG.seed = 3)),
#                modules = "glm",
#                n.chains = chains,
#                adapt = adapt,
#                burnin = burnin, sample = run,
#                thin = thin,
#                monitor = par, method = method, summarise = FALSE)

my.jags ← function(seed, model, data, inits = NULL, n.adapt, n.burnin, n.samples, thin) {
  require(rjags)
  load.module('glm')
  load.module('lecuyer')
  load.module('dic')
  jm ← jags.model(model, n.chains = 1, n.adapt = n.adapt,
    inits = list(list(.RNG.name = "base::Super-Duper", .RNG.seed = seed)),
    data = data)
  update(jm, n.iter = burnin)
  s ← coda.samples(jm, variable.names = par, n.iter = run * thin, thin = thin)
  s.pcp ← coda.samples(jm, variable.names = par.pcp, n.iter = n.pcp, thin = 1)
  return(list(s, s.pcp))
}


s.both ← mclapply(
  X = seq_len(chains),
  FUN = my.jags,
  model = paste("models/model-", M.lab, ".bug", sep = ""),
  data = D,
  n.adapt = adapt, n.burnin = burnin, n.samples = run, thin = thin,
  mc.preschedule = FALSE,
  mc.cores = chains)

# Not very elegant, but works now. Must be improved
s ← as.mcmc.list(list(s.both[[1]][[1]][[1]],
                       s.both[[2]][[1]][[1]],
                       s.both[[3]][[1]][[1]]))
```

```
s.p ← as.mcmc.list(list(s.both[[1]][[2]][[1]],
                        s.both[[2]][[2]][[1]],
                        s.both[[3]][[2]][[1]]))

save(s, file = paste("samples-", M.lab, ".RData", sep = ""))
save(s.p, file = paste("samples-p-", M.lab, ".RData", sep = ""))
proc.time() - t0

load(file = paste("samples-", M.lab, ".RData", sep = ""))
load(file = paste("samples-p-", M.lab, ".RData", sep = ""))

ggmcmc(ggs(s, family = "^theta|^Theta|^Omega"),
       file = paste0("ggmcmc-", M.lab, ".pdf"),
       param_page = 8,
       plot = c("traceplot", "geweke", "Rhat"))
```

3.1 *Part-worth contributions*

Figures 3.1 and 3.4 show the discrimination on all the characteristics. The parameters are the result of a logistic transformation, and therefore must be interpreted in terms of the odds (by exponentiating them) or predicted probabilities. They represent the *relative importances* of each feature (except for age, that the relative importances must be roughly divided by 2).

The caterpillar plot shows the distribution of the values for the parameters, where the dot represents the median expected effect, and the band covers 95 percent of the credible interval (also called the highest posterior density, HPD). In Bayesian statistics, however, there is not such thing as "statistically significant" for a parameter, as a clear cut indication, but results are presented in terms of probabilities of interest (such as the probability that the parameter is positive). See a Bayesian interpretation below, in the "Differential discrimination" output.

Comparing general population against public administration allows us to see the effect of having a smaller sample in the later group, by the sizes of the uncertainty bands. At this stage, I would avoid making any sort of interpretation of the public administration results (except, maybe, the strong deviation of the parameter regarding language effects in rights when treatment is not shown, that is remarkably far away from the general population).

For both outcomes, language is the most important feature to discriminate. In terms of magnitude, though, it is more important in rights than in welfare, but specially when the treatment is shown. For welfare, there are also signs of profession being relevant for discrimination, which is not at all suggested in rights.

```
# Use ggmcmc's features to get all the parameters in a tidy format
# Save some of them for later, to make model comparisons.
L.theta ← plab("theta", list(Outcome = outcome.label,
                             Population = population.label,
                             Treatment = treatment.label,
```

```
                    Feature = feature.label))
S.theta ← ggs(s, family = "^theta\\[", par_labels = L.theta) %>%
  mutate(Model = M)
save(S.theta, file = paste("samples-", M.lab, "-theta", ".RData", sep = ""))
ci.theta ← ci(S.theta)
```

Report baseline probabilities of picking the reference individual.

```
tb ← ci.theta %>%
  filter(Feature == "(Intercept)") %>%
  arrange(desc(abs(median))) %>%
  mutate(`Odds Ratio` = exp(median)) %>%
  mutate(`Expected odds ratio` = or(`Odds Ratio`)) %>%
  mutate(`Expected probability` = inv.logit(median)) %>%
  select(Outcome, Treatment, Population, `Expected odds ratio`, `Expected probability`)

tc ← "Expected odds ratio and probability of prioritization for the reference profile."
if (knitr::is_latex_output()) {
  kable(tb, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 10)
} else {
  kable(tb, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10, position = "center", bootstrap_options = "striped", full_width = T)
}
```

Table 3.1: Expected odds ratio and probability of prioritization for the reference profile.

Outcome	Treatment	Population	Expected odds ratio	Expected probability
Rights	Not shown	Public administration	▽ 51%	0.3288
Welfare	Not shown	Public administration	▽ 49%	0.3364
Rights	Shown	Public administration	▽ 48%	0.3441
Welfare	Shown	Public administration	▽ 46%	0.3515
Rights	Not shown	General population	▽ 46%	0.3519
Rights	Shown	General population	▽ 44%	0.3589
Welfare	Not shown	General population	▽ 42%	0.3670
Welfare	Shown	General population	▽ 41%	0.3698

```
tb ← ci.theta %>%
  filter(Feature == "(Intercept)") %>%
  arrange(rev(Population), Outcome, Treatment) %>%
  mutate(`Odds Ratio` = exp(median)) %>%
  mutate(`Expected odds ratio` = or(`Odds Ratio`)) %>%
  mutate(`Expected probability` = inv.logit(median)) %>%
  select(Population, Outcome, Treatment, `Expected odds ratio`, `Expected probability`)

tc ← "Expected odds ratio and probability of prioritization for the reference profile."
if (knitr::is_latex_output()) {
  kable(tb, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
```

```
    kable_styling(font_size = 10) %>%
    pack_rows("Public administration", 5, 8) %>%
    pack_rows("General population", 5, 8)
} else {
  kable(tb, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10, position = "center", bootstrap_options = "striped", full_width = T) %>%
    pack_rows("Public administration", 1, 4) %>%
    pack_rows("General population", 5, 8)
}
```

Table 3.2: Expected odds ratio and probability of prioritization for the reference profile.

Population	Outcome	Treatment	Expected odds ratio	Expected probability
Public administration	Rights	Not shown	▽ 51%	0.3288
Public administration	Rights	Shown	▽ 48%	0.3441
Public administration	Welfare	Not shown	▽ 49%	0.3364
Public administration	Welfare	Shown	▽ 46%	0.3515
Public administration				
General population				
General population	Rights	Not shown	▽ 46%	0.3519
General population	Rights	Shown	▽ 44%	0.3589
General population	Welfare	Not shown	▽ 42%	0.3670
General population	Welfare	Shown	▽ 41%	0.3698

```
ggplot(ci.theta, aes(ymin = low, ymax = high,
                              y = median, x = Feature,
                              color = Outcome)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4)) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1, position = position_dodge(width = 0.4)) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  facet_grid(Treatment ~ Population) +
  scale_color_discrete_qualitative(palette = "Dark 2")

ggplot(ci.theta, aes(ymin = low, ymax = high,
                              y = median, x = Feature,
                              color = Treatment)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4)) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1, position = position_dodge(width = 0.4)) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  facet_grid(Outcome ~ Population) +
  scale_color_discrete_qualitative(palette = "Harmonic")
```

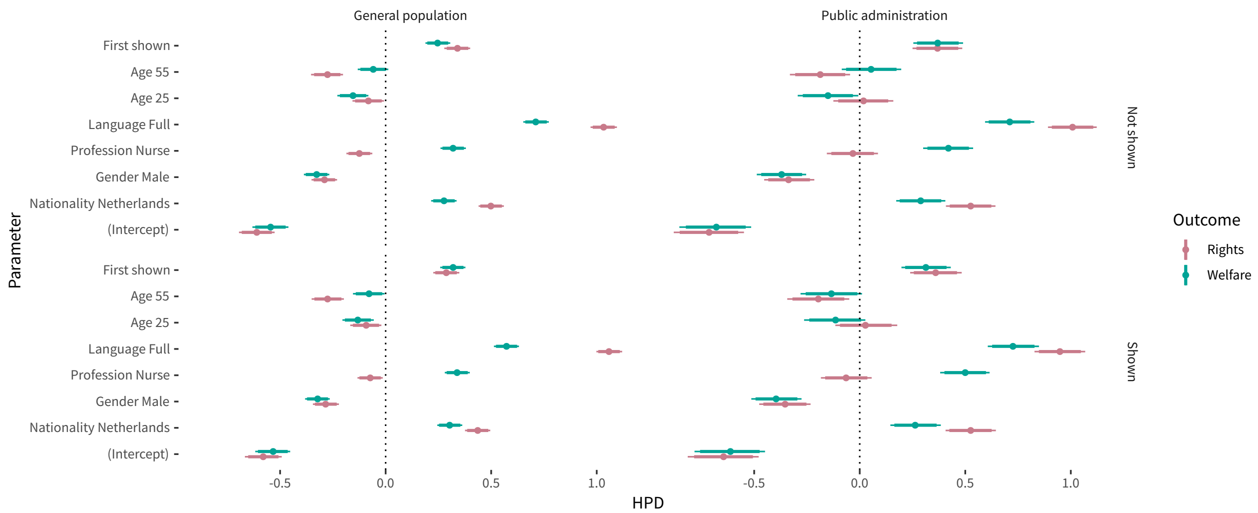

Figure 3.1: Prioritization of features, by population and treatment.

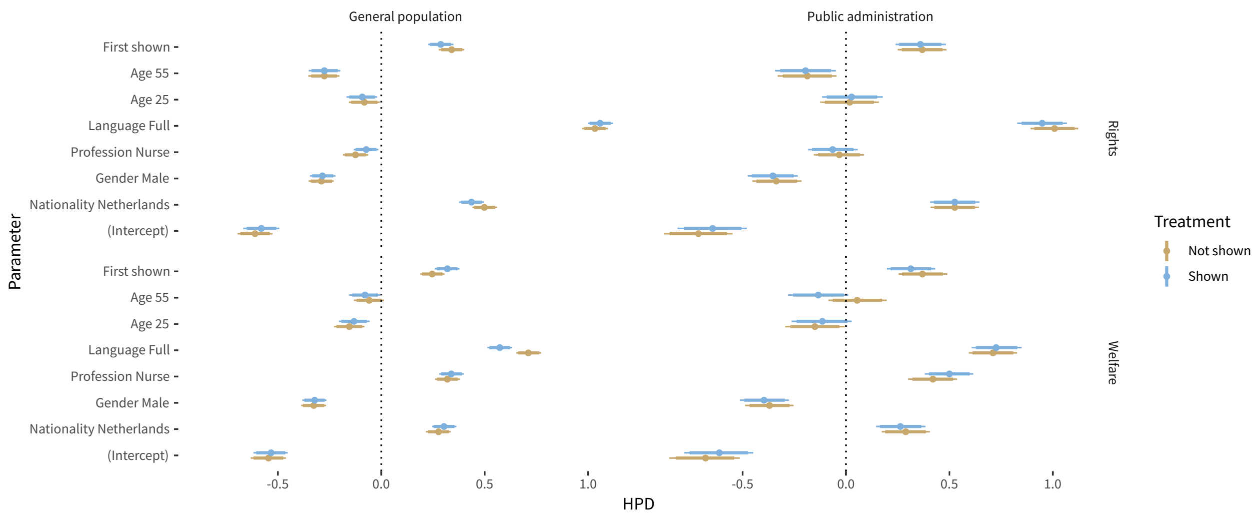

Figure 3.2: Prioritization of features, by population and outcome.

```
ci.theta %>%
  filter(!(Feature %in% c("(Intercept)", "First shown"))) %>%
  droplevels() %>%
  mutate(Feature = factor(Feature, rev(feature.label.order))) %>%
  mutate(Outcome = factor(Outcome, rev(levels(Outcome)))) %>%
  mutate(low = exp(low), high = exp(high),
         median = exp(median),
         Low = exp(Low), High = exp(High)) %>%
ggplot(aes(ymin = low, ymax = high,
           y = median, x = Feature,
           color = Treatment)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4)) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1, position = position_dodge(width = 0.4)) +
  geom_hline(aes(yintercept = 1), lty = 3) +
  xlab("Parameter") +
  ylab("Odds Ratio (HPD)") +
#  ylab("HPD") +
  facet_grid(Outcome ~ Population) +
  scale_color_grey(start = 0.6, end = 0.2) +
  theme(strip.text.x = element_text(face = "bold"),
        strip.text.y = element_text(face = "bold")) +
  scale_y_continuous(trans = log_trans(), breaks = c(0.7, 1, 1.5, 2)) +
  theme(panel.spacing = unit(2, "lines"))
```

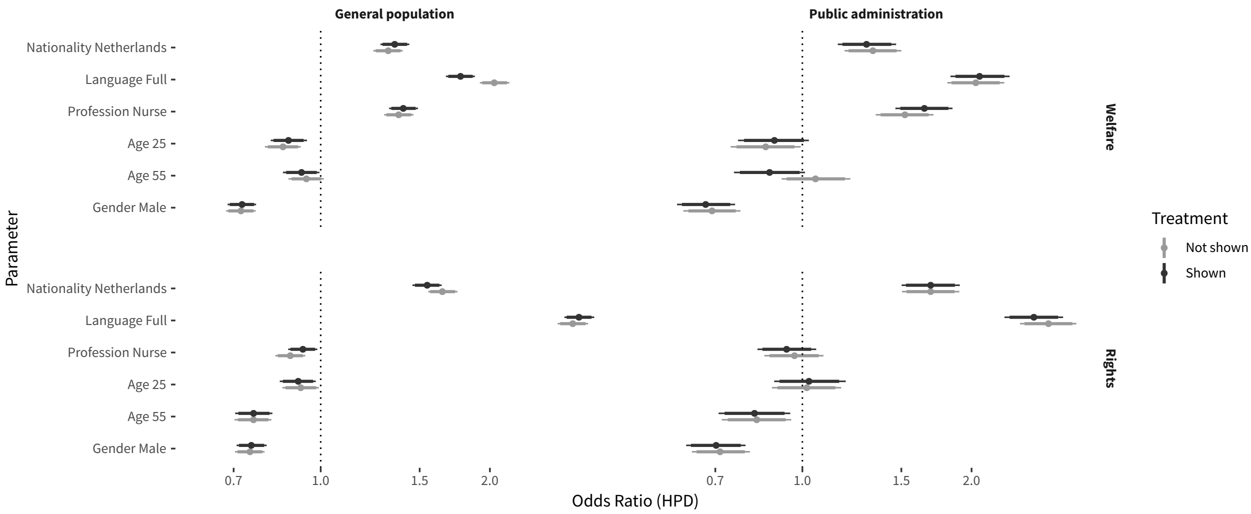

Figure 3.3: Prioritization of features, by population and outcome.

```
ggplot(ci.theta, aes(ymin = low, ymax = high,
                     y = median, x = Feature,
                     color = Population)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4)) +
  geom_linerange(position = position_dodge(width = 0.4)) +
```

```
  geom_linerange(aes(ymin = Low, ymax = High), size = 1, position = position_dodge(width = 0.4)) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  facet_grid(Treatment ~ Outcome) +
  scale_color_discrete_qualitative(palette = "Dynamic")
```

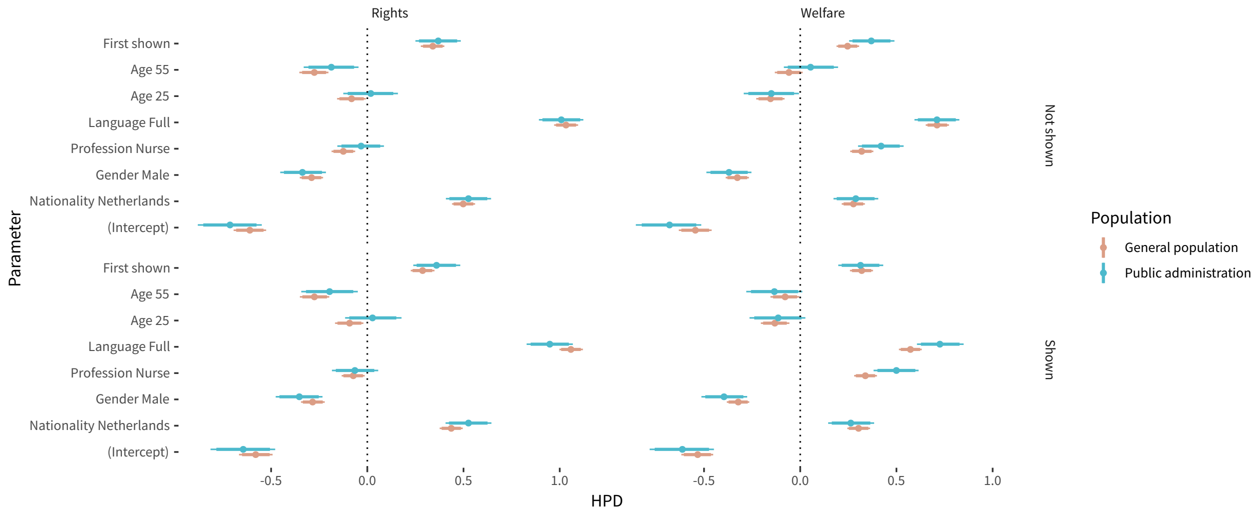

Figure 3.4: Prioritization of features, by treatment and outcome.

```
ci.theta %>%
  filter(!(Feature %in% c("(Intercept)", "First shown"))) %>%
  droplevels() %>%
  mutate(Feature = factor(Feature, rev(feature.label.order))) %>%
  mutate(Outcome = factor(Outcome, rev(levels(Outcome)))) %>%
  mutate(low = exp(low), high = exp(high),
         median = exp(median),
         Low = exp(Low), High = exp(High)) %>%
ggplot(aes(ymin = low, ymax = high,
           y = median, x = Feature,
           color = Population)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4)) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1, position = position_dodge(width = 0.4)) +
  geom_hline(aes(yintercept = 1), lty = 3) +
  xlab("Parameter") +
  ylab("Odds Ratio (HPD)") +
#  ylab("HPD") +
  facet_grid(Treatment ~ Outcome) +
  scale_color_grey(start = 0.8, end = 0.2) +
  scale_y_continuous(trans = log_trans(), breaks = c(0.7, 1, 1.5, 2)) +
  theme(strip.text.x = element_text(face = "bold"),
        strip.text.y = element_text(face = "bold")) +
  theme(panel.spacing = unit(2, "lines"))
```

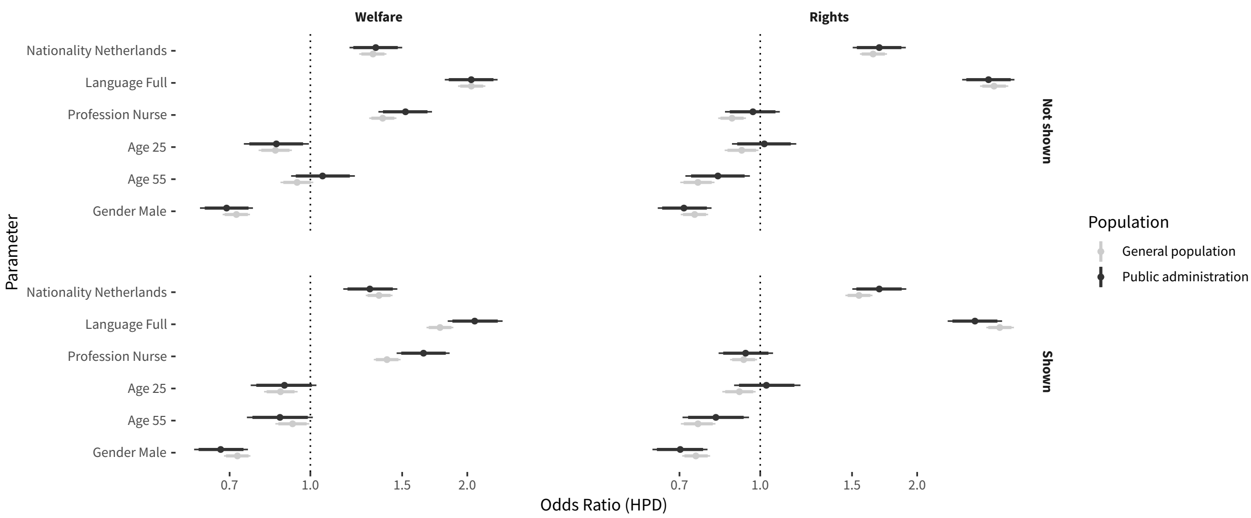

Figure 3.5: Prioritization of features, by treatment and outcome.

```
L.Theta ← plab("Theta", list(Outcome = outcome.label,
                             Feature = feature.label))
S.Theta ← ggs(s, family = "^Theta\\[", par_labels = L.Theta)
ggplot(ci(S.Theta), aes(ymin = low, ymax = high,
                             y = median, x = Feature,
                             color = Outcome)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.3)) +
  geom_linerange(position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1, position = position_dodge(width = 0.3)) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  scale_color_discrete_qualitative(palette = "Dark 2")
```

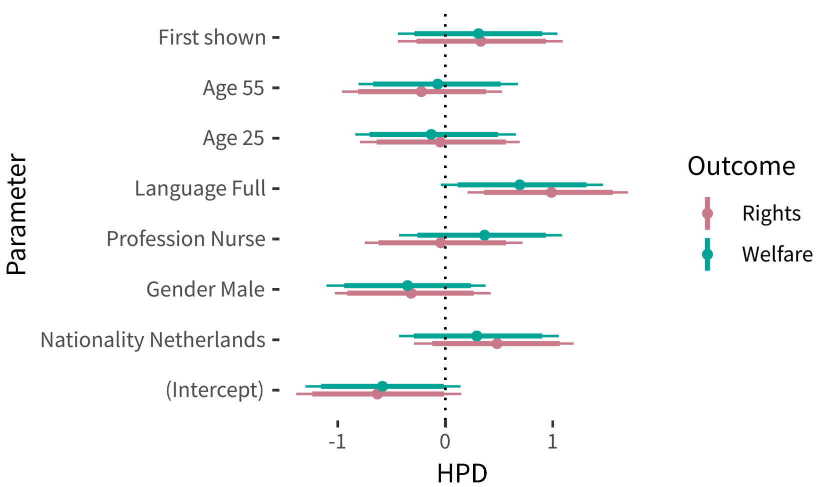

Figure 3.6: Prioritization of features (grand mean).

```
L.Omega ← plab("Omega", list(Feature = feature.label))
S.Omega ← ggs(s, family = "^Omega\\[", par_labels = L.Omega)
```

```
ggs_caterpillar(S.Omega) +
  geom_vline(xintercept = 0, lty = 3)
```

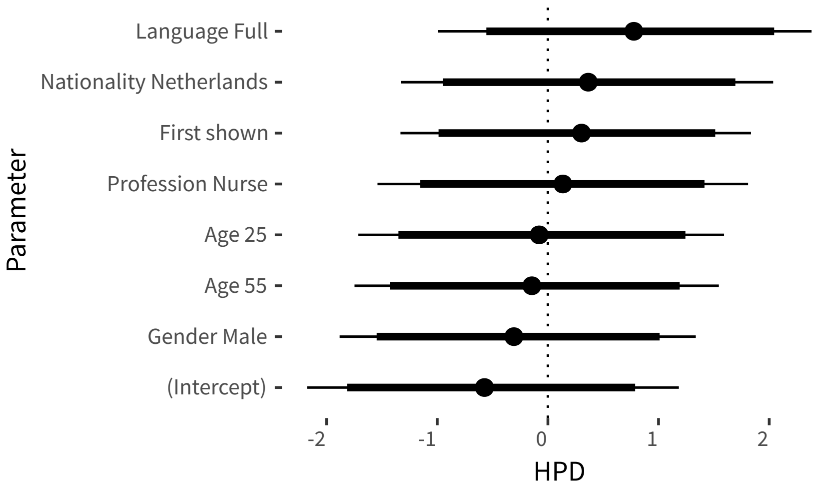

Figure 3.7: Prioritization of features (grand-grand mean).

List median expected effects.

```
tb ← S.Omega %>%
  filter(Feature ≠ "(Intercept)") %>%
  ci() %>%
  arrange(desc(abs(median))) %>%
  mutate(`Odds Ratio` = exp(median)) %>%
  mutate(`Expected effect` = or(`Odds Ratio`)) %>%
  select(Feature, `Odds Ratio`, `Expected effect`)

tc ← "Odds ratio of expected grand-effects, sorted by magnitude."
if (knitr::is_latex_output()) {
  kable(tb, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 10)
} else {
  kable(tb, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10, position = "center", bootstrap_options = "striped", full_width = T)
}
```

Table 3.3: Odds ratio of expected grand-effects, sorted by magnitude.

Feature	Odds Ratio	Expected effect
Language Full	2.1741	△ 120%
Nationality Netherlands	1.4403	△ 44%
Gender Male	0.7355	▽ 26%
First shown	1.3551	△ 36%
Age 55	0.8653	▽ 13%
Profession Nurse	1.1448	△ 14%
Age 25	0.9244	▽ 7.6%

On average, overall, individuals with full German are 2.17 times more likely to be prioritized. Dutch citizens are 44 percent more likely to be prioritized. Nurses are 14 percent more likely. Compared to individuals aged 40, younger and older ones are 7.6 and 14 percent less likely. Men are 26 percent less likely to be prioritized.

3.2 *Attribute importance*

Figures 3.8 and 3.9 show the relative attribute importance. Language accounts for approximately half the importance that respondents give to the prioritization (more in rights than in welfare). In public administration population the relative importances are more distributed across attributes. But at the same time, the relative importances between profession and nationality change more radically: they base their decision more on nationality when rights are involved, and profession when welfare is involved.

```
importance <- function(x) {
  if (length(x) == 1) {
    return(abs(x))
  } else {
    return(max(x) - min(x))
  }
}
S.importance <- S.theta %>%
  filter(!Feature %in% c("First shown", "(Intercept)")) %>%
  ci() %>%
  select(Outcome, Population, Treatment, Feature, value = median) %>%
  separate(Feature, " ", into = c("Attribute", "Level")) %>%
  group_by(Outcome, Population, Treatment, Attribute) %>%
  summarize(Importance = importance(value))

S.relative.importance <- S.importance %>%
  group_by(Outcome, Population, Treatment) %>%
  mutate(`Relative importance` = Importance / sum(Importance)) %>%
  mutate(Attribute = factor(Attribute, rev(c("Nationality",
                                             "Language",
                                             "Profession",
                                             "Age",
                                             "Gender"))))
save(S.relative.importance, file = paste("relative-importance-", M.lab, "-theta", ".RData", sep = ""))

ggplot(S.relative.importance,
       aes(x = Outcome, y = `Relative importance`, fill = Attribute)) +
  geom_bar(position = "fill", stat = "identity") +
  facet_grid(Population ~ Treatment) +
  scale_fill_brewer(type = "qual", palette = "Set1")

ggplot(S.relative.importance,
       aes(x = Treatment, y = `Relative importance`, fill = Attribute)) +
  geom_bar(position = "fill", stat = "identity") +
```

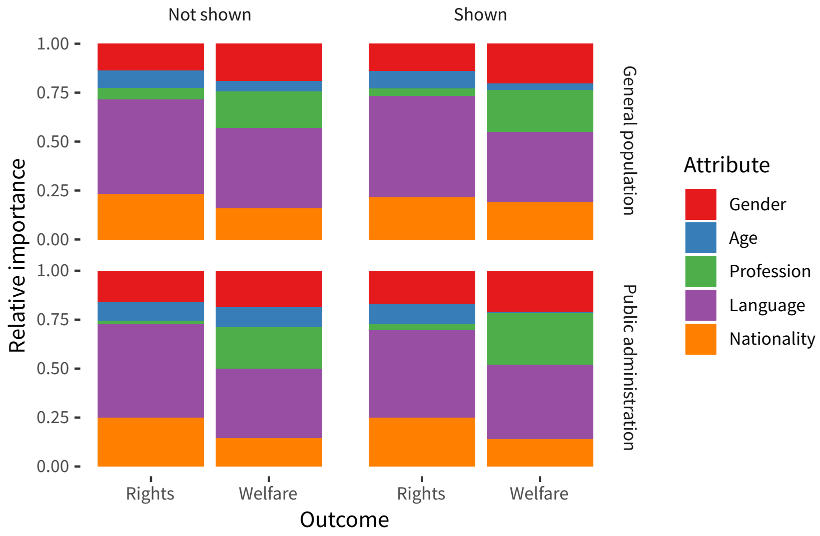

Figure 3.8: Relative importances, by outcome, population and treatment. Focus on comparing outcomes.

```
  facet_grid(Population ~ Outcome) +
  scale_fill_brewer(type = "qual", palette = "Set1")
```

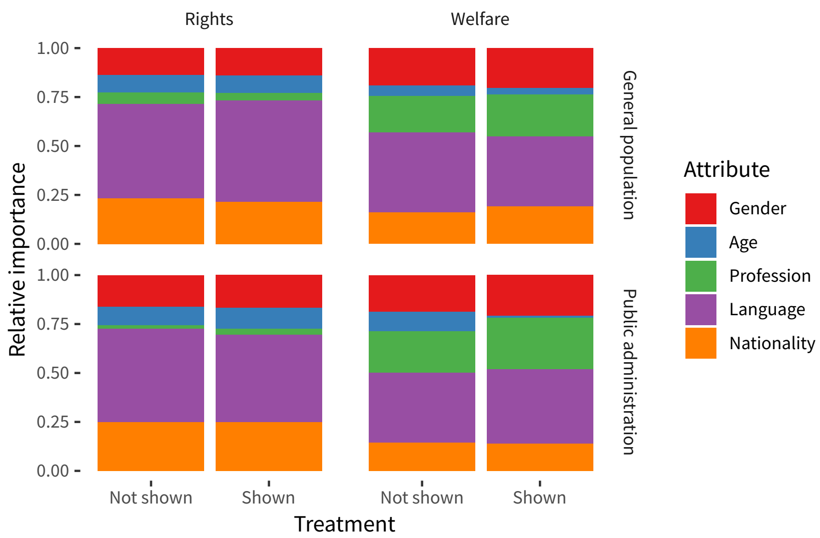

Figure 3.9: Relative importances, by outcome, population and treatment. Focus on comparing treatment.

```
S.relative.importance %>%
  ungroup() %>%
  mutate(Outcome = factor(Outcome, rev(levels(Outcome)))) %>%
ggplot(aes(x = Treatment, y = `Relative importance`, fill = Attribute)) +
  geom_bar(position = "fill", stat = "identity") +
  facet_grid(Population ~ Outcome) +
  theme(strip.text.x = element_text(face = "bold"),
        strip.text.y = element_text(face = "bold")) +
  scale_fill_grey(start = 0.8, end = 0.2)

ggplot(S.relative.importance,
       aes(x = Population, y = `Relative importance`, fill = Attribute)) +
```

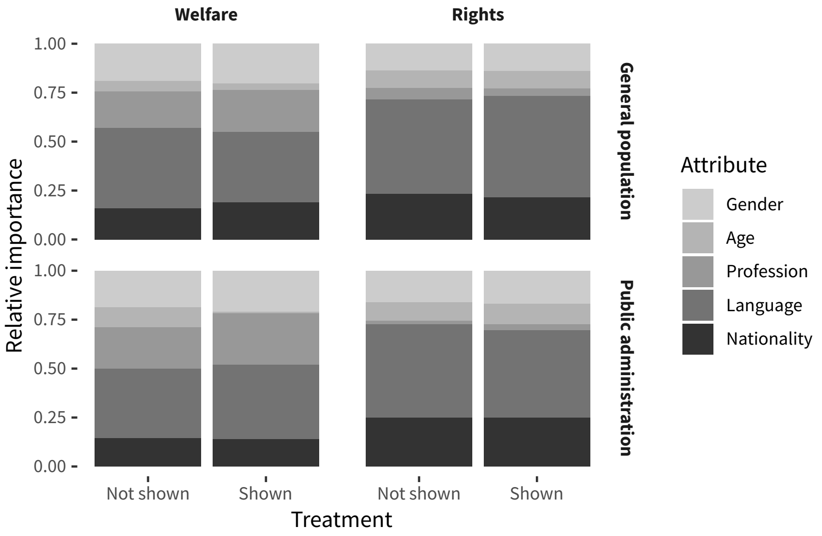

Figure 3.10: Relative importances, by outcome, population and treatment. Focus on comparing treatment.

```
  geom_bar(position = "fill", stat = "identity") +
  facet_grid(Outcome ~ Treatment) +
  scale_fill_brewer(type = "qual", palette = "Set1")
```

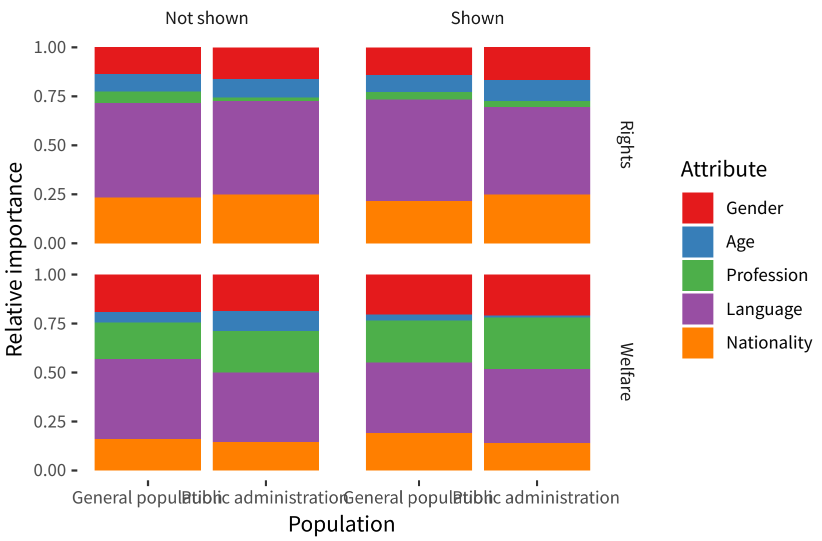

Figure 3.11: Relative importances, by population, treatment and outcome. Focus on comparing population.

```
S.relative.importance %>%
  ungroup() %>%
  mutate(Outcome = factor(Outcome, rev(levels(Outcome)))) %>%
  mutate(Population = str_replace(Population, " ", "\n")) %>%
ggplot(aes(x = Population, y = `Relative importance`, fill = Attribute)) +
  geom_bar(position = "fill", stat = "identity") +
  facet_grid(Outcome ~ Treatment) +
  theme(strip.text.x = element_text(face = "bold"),
        strip.text.y = element_text(face = "bold")) +
  scale_fill_grey(start = 0.8, end = 0.2)
```

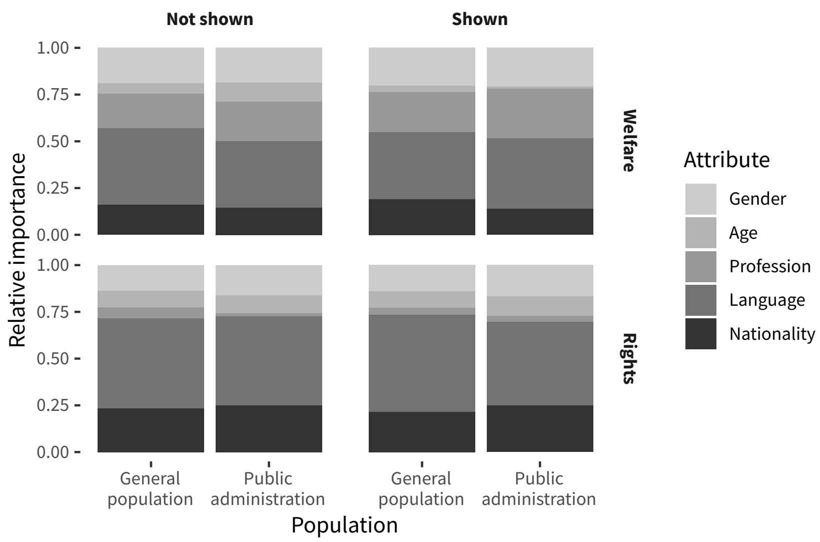

Figure 3.12: Relative importances, by population, treatment and outcome. Focus on comparing population.

3.3 *Differential discrimination*

Figure 3.13 shows the differential discrimination that welfare outcomes have over rights. Histograms show the distribution of the values of the differential discrimination in each simulation. Therefore, a histogram that clearly overlaps 0.5 indicates that there is no trace of differential discrimination. When the histogram barely overlaps zero it indicates evidence of a differential discrimination, either welfare effects being greater than rights (right to the zero) or rights effects being greater than welfare (left to the zero).

For the differential effect on discrimination based on Nationality, it can be read as follows: evidence that there is a differential effect of higher discrimination in welfare than in rights when the treatment is shown is 100 percent (also when the treatment is not shown). More specifically, the evidence is > 0.99983 (one in 6,000).

```
# With every iteration and every chaing a value of interest has been
# generated, a sample from the target distribution of interest is retrieved.
# Now simply pivot the cleaned version so that two columns with the different
# values of Outcome are generated, and can be compared to assess the
# differential discrimination.
dd.outcome ← S.theta %>%
  select(Iteration, Chain, Outcome, Population, Treatment, Feature, value) %>%
  pivot_wider(names_from = Outcome, values_from = value) %>%
  mutate(`Differential discrimination on welfare` = Rights - Welfare)
# Prepare the table
tb ← dd.outcome %>%
  group_by(Population, Treatment, Feature) %>%
  summarize(`Prob Welfare > Rights` = length(which(`Differential discrimination on welfare` > 0)) / n()) %>
  arrange(desc(`Prob Welfare > Rights`))
# Improve the readability of the numbers that will appear in the figures
fig.labels ← tb %>%
  mutate(P = paste0(100 * signif(`Prob Welfare > Rights`, 2), "%")) %>%
```

```
  mutate(P = str_replace(P, "^0\\.", "\\."))

ggplot(filter(dd.outcome, Population == "General population"),
       aes(x = `Differential discrimination on welfare`)) +
  geom_histogram(binwidth = 0.01) +
  facet_grid(Treatment ~ Feature) +
  geom_text(data = filter(fig.labels, Population == "General population"),
            aes(x = -Inf, y = Inf, label = P), size = rel(0.8),
            hjust = -0.5, vjust = +1.5) +
  geom_vline(xintercept = 0, lty = 3) +
  xlim(c(-0.75, 0.75)) +
  ggtitle("Differential discrimination on welfare (General population)")
```

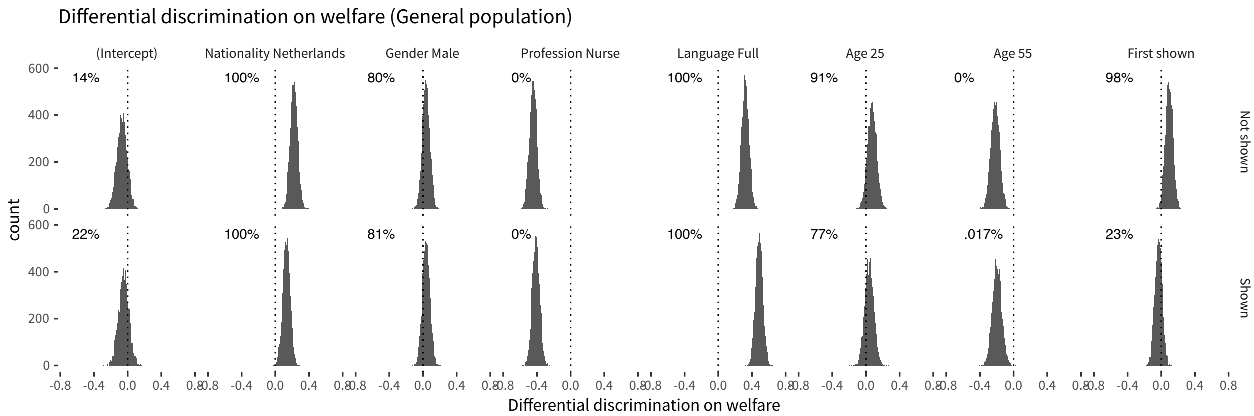

Figure 3.13: Differential discrimination on welfare over rights. Only general population is shown. Values indicate the probability of having found a differential discrimination favouring welfare.

```
tb <- dd.outcome %>%
  group_by(Population, Treatment, Feature) %>%
  summarize(`Prob Welfare > Rights` = length(which(`Differential discrimination on welfare` > 0)) / n()) %>
  arrange(desc(`Prob Welfare > Rights`))

tc <- "Probability of finding a differential discrimination on welfare over rights."
if (knitr::is_latex_output()) {
  kable(tb, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 8)
} else {
  kable(tb, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 8, position = "center", bootstrap_options = "striped", full_width = T)
}
```

Table 3.4: Probability of finding a differential discrimination on welfare over rights.

Population	Treatment	Feature	Prob Welfare > Rights
General population	Not shown	Nationality Netherlands	1.0000
General population	Not shown	Language Full	1.0000
General population	Shown	Language Full	1.0000
Public administration	Not shown	Language Full	1.0000
Public administration	Shown	Nationality Netherlands	0.9995
General population	Shown	Nationality Netherlands	0.9987

Public administration	Not shown	Nationality Netherlands	0.9982
Public administration	Shown	Language Full	0.9942
General population	Not shown	First shown	0.9828
Public administration	Not shown	Age 25	0.9482
Public administration	Shown	Age 25	0.9108
General population	Not shown	Age 25	0.9097
General population	Shown	Gender Male	0.8113
General population	Not shown	Gender Male	0.8012
General population	Shown	Age 25	0.7692
Public administration	Shown	First shown	0.6980
Public administration	Shown	Gender Male	0.6838
Public administration	Not shown	Gender Male	0.6595
Public administration	Not shown	First shown	0.4970
Public administration	Shown	(Intercept)	0.3920
Public administration	Not shown	(Intercept)	0.3902
Public administration	Shown	Age 55	0.2735
General population	Shown	First shown	0.2265
General population	Shown	(Intercept)	0.2183
General population	Not shown	(Intercept)	0.1432
Public administration	Not shown	Age 55	0.0068
General population	Shown	Age 55	0.0002
General population	Not shown	Profession Nurse	0.0000
General population	Not shown	Age 55	0.0000
General population	Shown	Profession Nurse	0.0000
Public administration	Not shown	Profession Nurse	0.0000
Public administration	Shown	Profession Nurse	0.0000

Figure 3.14 shows the differential discrimination that treated individuals (shown) have over not treated (not shown).

```
dd.treatment <- S.theta %>%
  select(Iteration, Chain, Outcome, Population, Treatment, Feature, value) %>%
  pivot_wider(names_from = Treatment, values_from = value) %>%
  mutate(`Differential discrimination on treatment shown` = `Not shown` - Shown)
tb <- dd.treatment %>%
  group_by(Population, Outcome, Feature) %>%
  summarize(`Prob Shown > Not shown` = length(which(`Differential discrimination on treatment shown` > 0))
  arrange(desc(`Prob Shown > Not shown`))
fig.labels <- tb %>%
  mutate(P = paste0(100 * signif(`Prob Shown > Not shown`, 2), "%")) %>%
  mutate(P = str_replace(P, "^0\\.", "\\."))

ggplot(filter(dd.treatment, Population == "General population"),
      aes(x = `Differential discrimination on treatment shown`)) +
  geom_histogram(binwidth = 0.01) +
  facet_grid(Outcome ~ Feature) +
  geom_text(data = filter(fig.labels, Population == "General population"),
            aes(x = -Inf, y = Inf, label = P), size = rel(0.8),
            hjust = -0.5, vjust = +1.5) +
  geom_vline(xintercept = 0, lty = 3) +
  xlim(c(-0.75, 0.75)) +
  ggtitle("Differential discrimination on treatment shown")


tb <- dd.treatment %>%
```

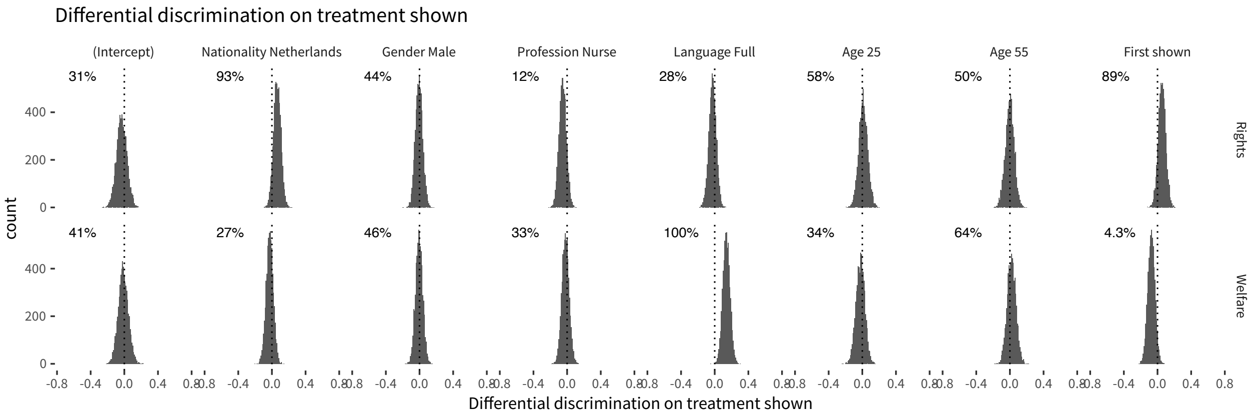

Figure 3.14: Differential discrimination on treated individuals over non treated. Only general population is shown. Values indicate the probability of having found a differential discrimination favouring treated individuals.

```
  group_by(Population, Outcome, Feature) %>%
  summarize(`Prob Shown > Not shown` = length(which(`Differential discrimination on treatment shown` > 0))
  arrange(desc(`Prob Shown > Not shown`))

tc <- "Probability of finding a differential discrimination on treatment shown over non shown."
if (knitr::is_latex_output()) {
  kable(tb, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 8)
} else {
  kable(tb, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 8, position = "center", bootstrap_options = "striped", full_width = T)
}
```

Table 3.5: Probability of finding a differential discrimination on treatment shown over non shown.

Population	Outcome	Feature	Prob Shown > Not shown
General population	Welfare	Language Full	0.9995
Public administration	Welfare	Age 55	0.9640
General population	Rights	Nationality Netherlands	0.9253
General population	Rights	First shown	0.8888
Public administration	Rights	Language Full	0.7623
Public administration	Welfare	First shown	0.7493
General population	Welfare	Age 55	0.6408
Public administration	Rights	Profession Nurse	0.6375
Public administration	Welfare	Nationality Netherlands	0.6228
Public administration	Welfare	Gender Male	0.6215
Public administration	Rights	Gender Male	0.5838
General population	Rights	Age 25	0.5788
Public administration	Rights	Age 55	0.5408
Public administration	Rights	First shown	0.5362
Public administration	Rights	Nationality Netherlands	0.5060
General population	Rights	Age 55	0.5030
Public administration	Rights	Age 25	0.4575
General population	Welfare	Gender Male	0.4557
General population	Rights	Gender Male	0.4418
Public administration	Welfare	Language Full	0.4300
General population	Welfare	(Intercept)	0.4128
Public administration	Welfare	Age 25	0.3642
General population	Welfare	Age 25	0.3405
General population	Welfare	Profession Nurse	0.3265

General population	Rights	(Intercept)	0.3133
Public administration	Welfare	(Intercept)	0.2925
Public administration	Rights	(Intercept)	0.2878
General population	Rights	Language Full	0.2827
General population	Welfare	Nationality Netherlands	0.2688
Public administration	Welfare	Profession Nurse	0.1748
General population	Rights	Profession Nurse	0.1167
General population	Welfare	First shown	0.0430

Figure 3.15 shows the differential discrimination that public administration have over general population.

```
dd.population ← S.theta %>%
  select(Iteration, Chain, Outcome, Population, Treatment, Feature, value) %>%
  pivot_wider(names_from = Population, values_from = value) %>%
  mutate(`Differential discrimination on public administration` = `Public administration` - `General popula
tb ← dd.population %>%
  group_by(Outcome, Treatment, Feature) %>%
  summarize(`Prob PA > General` = length(which(`Differential discrimination on public administration` > 0))
  arrange(desc(`Prob PA > General`))
fig.labels ← tb %>%
  mutate(P = paste0(100 * signif(`Prob PA > General`, 2), "%")) %>%
  mutate(P = str_replace(P, "^0\\.", "\\."))

ggplot(filter(dd.population, Treatment == "Shown"),
       aes(x = `Differential discrimination on public administration`)) +
  geom_histogram(binwidth = 0.01) +
  facet_grid(Outcome ~ Feature) +
  geom_text(data = filter(fig.labels, Treatment == "Shown"),
            aes(x = -Inf, y = Inf, label = P), size = rel(0.8),
            hjust = -0.5, vjust = +1.5) +
  geom_vline(xintercept = 0, lty = 3) +
  xlim(c(-0.75, 0.75)) +
  ggtitle("Differential discrimination on public administration")
```

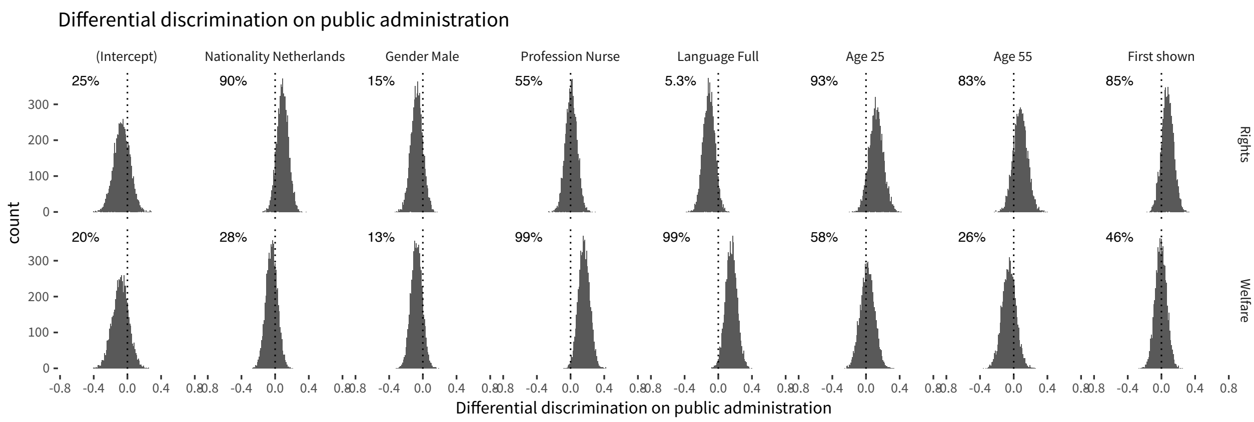

Figure 3.15: Differential discrimination on public administration over general population. Only treatment is shown. Values indicate the probability of having found a differential discrimination favouring public administration.

```
tb <- dd.population %>%
  group_by(Outcome, Treatment, Feature) %>%
  summarize(`Prob PA > General` = length(which(`Differential discrimination on public administration` > 0))
  arrange(desc(`Prob PA > General`))

tc <- "Probability of finding a differential discrimination on public administration over general populatio
if (knitr::is_latex_output()) {
  kable(tb, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 8)
} else {
  kable(tb, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 8, position = "center", bootstrap_options = "striped", full_width = T)
}
```

Table 3.6: Probability of finding a differential discrimination on public administration over general population.

Outcome	Treatment	Feature	Prob PA > General
Welfare	Shown	Profession Nurse	0.9920
Welfare	Shown	Language Full	0.9867
Welfare	Not shown	First shown	0.9655
Welfare	Not shown	Profession Nurse	0.9295
Rights	Shown	Age 25	0.9263
Welfare	Not shown	Age 55	0.9140
Rights	Not shown	Profession Nurse	0.9110
Rights	Shown	Nationality Netherlands	0.9040
Rights	Not shown	Age 25	0.8860
Rights	Not shown	Age 55	0.8645
Rights	Shown	First shown	0.8513
Rights	Shown	Age 55	0.8262
Rights	Not shown	First shown	0.6592
Rights	Not shown	Nationality Netherlands	0.6588
Welfare	Shown	Age 25	0.5808
Welfare	Not shown	Nationality Netherlands	0.5700
Rights	Shown	Profession Nurse	0.5510
Welfare	Not shown	Age 25	0.5157
Welfare	Not shown	Language Full	0.4967
Welfare	Shown	First shown	0.4640
Rights	Not shown	Language Full	0.3567
Welfare	Shown	Nationality Netherlands	0.2820
Welfare	Not shown	Gender Male	0.2618
Welfare	Shown	Age 55	0.2558
Rights	Shown	(Intercept)	0.2548
Rights	Not shown	Gender Male	0.2493
Welfare	Shown	(Intercept)	0.1993
Rights	Shown	Gender Male	0.1545
Rights	Not shown	(Intercept)	0.1428
Welfare	Shown	Gender Male	0.1347
Welfare	Not shown	(Intercept)	0.0790
Rights	Shown	Language Full	0.0530

3.4 *Model fit*

Model fit is assessed using the percent of cases correctly predicted (PCP). On average, we would find that simply by chance we can predict 0.5 of the prioritizations. The PCP is the percentage of cases correctly predicted (those predicted as being prioritized indeed being prioritized plus those predicted not being prioritized indeed not being prioritized, divided by all cases).

Figure 3.16 shows the posterior distribution of PCP, and the following table its median expected value.

```
threshold <- d %>%
  group_by(Outcome, Population, Treatment) %>%
  summarize(Threshold = length(which(Decision == 1)) / n())

L.p <- plab("p", list(Observation = 1:nO)) %>%
  mutate(Observation = as.numeric(as.character(Observation)))

S.full <- ggs(s.p, family = "^p\\[", par_labels = L.p, sort = FALSE) %>%
  select(Iteration, Chain, Observation, value) %>%
  left_join(select(d, Observation, Population, Treatment, Outcome, Decision)) %>%
  left_join(threshold) %>%
  mutate(Correct = if_else( (value < Threshold & Decision == 0) |
                            (value > Threshold & Decision == 1),
                          TRUE, FALSE))

S <- S.full %>%
  group_by(Outcome, Population, Treatment, Iteration, Chain) %>%
  summarize(PCP = length(which(Correct)) / n())


ggplot(S, aes(x = PCP, color = Population)) +
  geom_density() +
  facet_grid(Treatment ~ Outcome) +
  expand_limits(x = c(0, 1)) +
  geom_vline(xintercept = 0, lty = 3) +
  scale_color_discrete_qualitative(palette = "Dynamic")
```

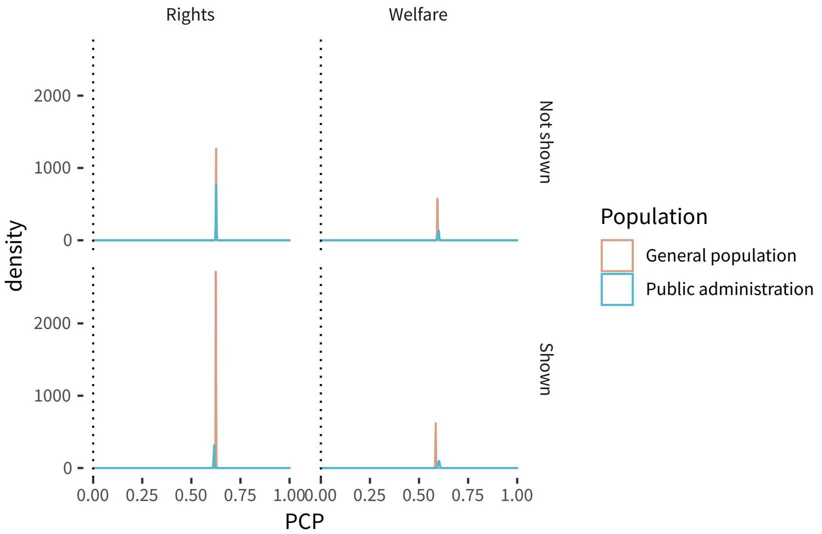

Figure 3.16: Model fit: Percent correctly predicted.

```
t.pcp ← S %>%
  group_by(Population, Treatment, Outcome) %>%
  summarize(`Median PCP` = mean(PCP))
t.pcp.m ← mutate(t.pcp, Model = M)
save(t.pcp.m, file = paste0("table-pcp-", M.lab, ".RData"))

tc ← "Posterior median percent correctly predicted, by outcome, treatment and
population."
if (knitr::is_latex_output()) {
  kable(t.pcp, format = "latex", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10)
} else {
  kable(t.pcp, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10, position = "center", bootstrap_options = "striped", full_width = F)
}
```

Table 3.7: Posterior median percent correctly predicted, by outcome, treatment and population.

Population	Treatment	Outcome	Median PCP
General population	Not shown	Rights	0.6258
General population	Not shown	Welfare	0.5938
General population	Shown	Rights	0.6248
General population	Shown	Welfare	0.5849
Public administration	Not shown	Rights	0.6264
Public administration	Not shown	Welfare	0.5989
Public administration	Shown	Rights	0.6155
Public administration	Shown	Welfare	0.6010

List of the 10 worstly predicted cases, on average.

```
wpo ← S.full %>%
  group_by(Observation) %>%
  summarize(Distance = mean(Decision - value)) %>%
  arrange(desc(abs(Distance))) %>%
  slice(1:10) %>%
  select(Observation) %>%
  unlist(use.names = FALSE)
rm(S.full)
invisible(gc()) # Save memory

E[wpo,] %>%
  select(id, Population, Treatment, Outcome,
         Nationality, Gender, Language, Profession, Age,
         Decision) %>%
  arrange(id) %>%
  data.frame()
```

```
→       id         Population Treatment Outcome Nationality Gender Language
→ 1     10 General population Not shown  Rights     Romania   Male       No
→ 2    477 General population Not shown  Rights     Romania   Male       No
```

```
→ 3   572 General population Not shown  Rights     Romania   Male       No
→ 4   582 General population Not shown  Rights     Romania   Male       No
→ 5   658 General population Not shown  Rights     Romania   Male       No
→ 6   850 General population Not shown  Rights     Romania   Male       No
→ 7   862 General population Not shown  Rights     Romania   Male       No
→ 8  1144 General population Not shown  Rights     Romania   Male       No
→ 9  1182 General population Not shown  Rights     Romania   Male       No
→ 10 1248 General population Not shown  Rights     Romania   Male       No
→    Profession Age Decision
→ 1       Nurse  55        1
→ 2       Nurse  55        1
→ 3       Nurse  55        1
→ 4       Nurse  55        1
→ 5       Nurse  55        1
→ 6       Nurse  55        1
→ 7       Nurse  55        1
→ 8       Nurse  55        1
→ 9       Nurse  55        1
→ 10      Nurse  55        1
```

3.5 *Model fit contrasting respondent quality*

```
load("poor_quality_individual.RData")

threshold <- d %>%
  group_by(Outcome, Population, Treatment) %>%
  summarize(Threshold = length(which(Decision == 1)) / n())

L.p <- plab("p", list(Observation = 1:nO)) %>%
  mutate(Observation = as.numeric(as.character(Observation)))

S.full <- ggs(s.p, family = "^p\\[", par_labels = L.p, sort = FALSE) %>%
  select(Iteration, Chain, Parameter, Observation, value) %>%
  left_join(select(d, id, Observation, Population, Treatment, Outcome, Decision)) %>%
  left_join(threshold) %>%
  mutate(Correct = if_else( (value < Threshold & Decision == 0) |
                            (value > Threshold & Decision == 1),
                          TRUE, FALSE))

S <- S.full %>%
  mutate(Respondent = ifelse(id %in% id.poor, "Low quality", "Regular")) %>%
  group_by(Outcome, Population, Treatment, Respondent, Iteration, Chain) %>%
  summarize(PCP = length(which(Correct)) / n())

ggplot(S, aes(x = PCP, y = Treatment,
              color = Respondent)) +
  geom_boxplot(position = position_dodge(width = 0.3)) +
  facet_grid(Population ~ Outcome) +
  scale_color_discrete_qualitative(palette = "Dynamic")
```

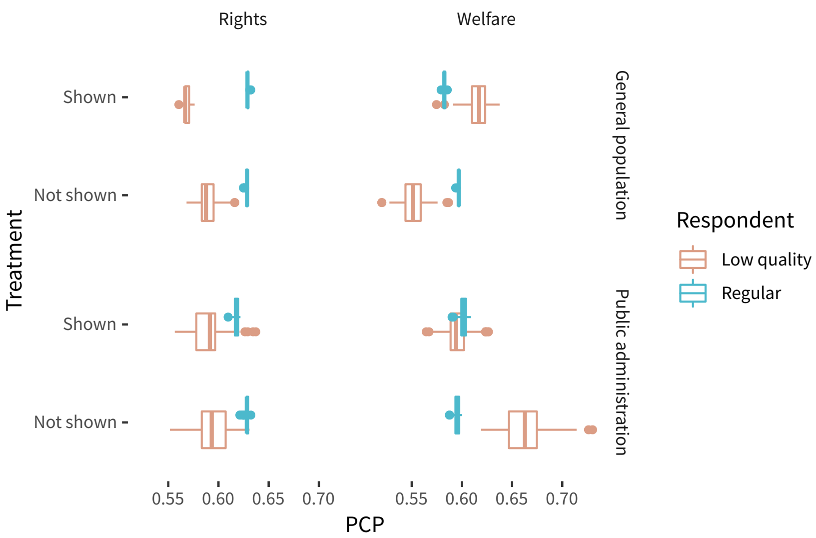

Figure 3.17: Model fit: Percent correctly predicted.

```
t.pcp <- S %>%
  group_by(Population, Treatment, Outcome, Respondent) %>%
  summarize(`Median PCP` = mean(PCP))
t.pcp.m.respondent <- mutate(t.pcp, Model = M)
save(t.pcp.m.respondent, file = paste0("table-pcp-respondent-", M.lab, ".RData"))

tc <- "Posterior median percent correctly predicted, by outcome, treatment, population and respondent quali
if (knitr::is_latex_output()) {
  kable(t.pcp, format = "latex", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10)
} else {
  kable(t.pcp, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10, position = "center", bootstrap_options = "striped", full_width = F)
}
```

3.6 *Hypothesis 1: Dutch are treated more favourably*

```
tb <- S.theta %>%
  filter(Feature == "Nationality Netherlands") %>%
  group_by(Outcome, Population, Treatment) %>%
  summarize(`Prob (H1)` = length(which(value > 0)) / n()) %>%
  arrange(desc(`Prob (H1)`))

tc <- "Evidence for Hypothesis 1."
if (knitr::is_latex_output()) {
  kable(tb, format = "latex", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10)
} else {
  kable(tb, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10, position = "center", bootstrap_options = "striped", full_width = F)
}
```

Table 3.8: Posterior median percent correctly predicted, by outcome, treatment, population and respondent quality.

Population	Treatment	Outcome	Respondent	Median PCP
General population	Not shown	Rights	Low quality	0.5886
General population	Not shown	Rights	Regular	0.6284
General population	Not shown	Welfare	Low quality	0.5515
General population	Not shown	Welfare	Regular	0.5969
General population	Shown	Rights	Low quality	0.5683
General population	Shown	Rights	Regular	0.6292
General population	Shown	Welfare	Low quality	0.6159
General population	Shown	Welfare	Regular	0.5826
Public administration	Not shown	Rights	Low quality	0.5938
Public administration	Not shown	Rights	Regular	0.6282
Public administration	Not shown	Welfare	Low quality	0.6623
Public administration	Not shown	Welfare	Regular	0.5954
Public administration	Shown	Rights	Low quality	0.5893
Public administration	Shown	Rights	Regular	0.6179
Public administration	Shown	Welfare	Low quality	0.5956
Public administration	Shown	Welfare	Regular	0.6015

Table 3.9: Evidence for Hypothesis 1.

Outcome	Population	Treatment	Prob (H1)
Rights	General population	Not shown	1
Rights	General population	Shown	1
Rights	Public administration	Not shown	1
Rights	Public administration	Shown	1
Welfare	General population	Not shown	1
Welfare	General population	Shown	1
Welfare	Public administration	Not shown	1
Welfare	Public administration	Shown	1

3.7 *Hypothesis 2: Fluent german speakers are treated more favourably*

```
tb <- S.theta %>%
  filter(Feature == "Language Full") %>%
  group_by(Outcome, Population, Treatment) %>%
  summarize(`Prob (H2)` = length(which(value > 0)) / n()) %>%
  arrange(desc(`Prob (H2)`))

tc <- "Evidence for Hypothesis 2."
if (knitr::is_latex_output()) {
  kable(tb, format = "latex", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10)
} else {
  kable(tb, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10, position = "center", bootstrap_options = "striped", full_width = F)
}
```

Table 3.10: Evidence for Hypothesis 2.

Outcome	Population	Treatment	Prob (H2)
Rights	General population	Not shown	1
Rights	General population	Shown	1
Rights	Public administration	Not shown	1
Rights	Public administration	Shown	1
Welfare	General population	Not shown	1
Welfare	General population	Shown	1
Welfare	Public administration	Not shown	1
Welfare	Public administration	Shown	1

3.8 *Hypothesis 3: Bureaucrats discriminate less*

This hypothesis only considers absolute discrimination (absolute values), not the direction).

```
tb <- S.theta %>%
  pivot_wider(c(Iteration, Chain, Outcome, Treatment, Feature),
              names_from = Population, values_from = value) %>%
  group_by(Outcome, Treatment, Feature) %>%
  summarize(`Prob (H3)` = length(which(abs(`Public administration`) < abs(`General population`))) / n()) %>
  arrange(desc(`Prob (H3)`))

tc <- "Evidence for Hypothesis 3."
if (knitr::is_latex_output()) {
  kable(tb, format = "latex", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10)
} else {
  kable(tb, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10, position = "center", bootstrap_options = "striped", full_width = F)
}
```

Table 3.11: Evidence for Hypothesis 3.

Outcome	Treatment	Feature	Prob (H3)
Rights	Shown	Language Full	0.9470
Rights	Not shown	Profession Nurse	0.9015
Rights	Not shown	Age 55	0.8645
Rights	Shown	Age 55	0.8262
Rights	Shown	Age 25	0.7183
Welfare	Shown	Nationality Netherlands	0.7180
Rights	Not shown	Age 25	0.6742
Rights	Not shown	Language Full	0.6433
Welfare	Shown	Age 25	0.5793
Welfare	Shown	First shown	0.5360
Rights	Shown	Profession Nurse	0.5277
Welfare	Not shown	Age 25	0.5157
Welfare	Not shown	Language Full	0.5033
Welfare	Not shown	Age 55	0.4592
Welfare	Not shown	Nationality Netherlands	0.4300
Rights	Not shown	Nationality Netherlands	0.3412
Rights	Not shown	First shown	0.3408
Welfare	Not shown	Gender Male	0.2618
Rights	Shown	(Intercept)	0.2548
Rights	Not shown	Gender Male	0.2493
Welfare	Shown	Age 55	0.2487
Welfare	Shown	(Intercept)	0.1993
Rights	Shown	Gender Male	0.1545
Rights	Shown	First shown	0.1487
Rights	Not shown	(Intercept)	0.1428
Welfare	Shown	Gender Male	0.1347
Rights	Shown	Nationality Netherlands	0.0960
Welfare	Not shown	(Intercept)	0.0790
Welfare	Not shown	Profession Nurse	0.0705
Welfare	Not shown	First shown	0.0345
Welfare	Shown	Language Full	0.0133
Welfare	Shown	Profession Nurse	0.0080

3.9 *Hypothesis 4: respondents treat more favourably applicants that are similar to them*

Not possible to assess at the aggregate level.

3.10 *Hypothesis 5: Managerial accountability reduces discrimination*

In other words, the existence of the treatment (shown) reduces discrimination.

This hypothesis only considers absolute discrimination (absolute values), not the direction).

```
tb <- S.theta %>%
  pivot_wider(c(Iteration, Chain, Outcome, Population, Feature),
              names_from = Treatment, values_from = value) %>%
  group_by(Outcome, Population, Feature) %>%
  summarize(`Prob (H5)` = length(which(
    (`Not shown` < 0 & `Shown` > `Not shown`) |
    (`Not shown` > 0 & `Shown` < `Not shown`))) / n()) %>%
  arrange(desc(`Prob (H5)`))

tc <- "Evidence for Hypothesis 5."
if (knitr::is_latex_output()) {
  kable(tb, format = "latex", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10)
} else {
  kable(tb, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10, position = "center", bootstrap_options = "striped", full_width = F)
}
```

Table 3.12: Evidence for Hypothesis 5.

Outcome	Population	Feature	Prob (H5)
Welfare	General population	Language Full	0.9995
Rights	General population	Nationality Netherlands	0.9253
Rights	General population	First shown	0.8888
Rights	General population	Profession Nurse	0.8833
Welfare	Public administration	Age 55	0.7962
Rights	Public administration	Language Full	0.7623
Welfare	Public administration	First shown	0.7493
Rights	Public administration	Age 25	0.7280
Rights	Public administration	(Intercept)	0.7122
Welfare	Public administration	(Intercept)	0.7075
Rights	General population	(Intercept)	0.6867
Welfare	General population	Age 25	0.6595
Welfare	Public administration	Age 25	0.6550
Welfare	Public administration	Nationality Netherlands	0.6228
Rights	Public administration	Profession Nurse	0.6207
Welfare	General population	(Intercept)	0.5872
Rights	General population	Gender Male	0.5582
Welfare	General population	Gender Male	0.5443
Rights	Public administration	First shown	0.5362
Rights	Public administration	Nationality Netherlands	0.5060
Rights	General population	Age 55	0.4970
Rights	Public administration	Age 55	0.4652
Rights	General population	Age 25	0.4370
Welfare	Public administration	Language Full	0.4300
Rights	Public administration	Gender Male	0.4162
Welfare	General population	Age 55	0.4122
Welfare	Public administration	Gender Male	0.3785
Welfare	General population	Profession Nurse	0.3265
Rights	General population	Language Full	0.2827
Welfare	General population	Nationality Netherlands	0.2688
Welfare	Public administration	Profession Nurse	0.1748
Welfare	General population	First shown	0.0430

Programming environment

```
sessionInfo()
```

```
→ R version 4.0.5 (2021-03-31)
→ Platform: x86_64-pc-linux-gnu (64-bit)
→ Running under: Gentoo/Linux
→
→ Matrix products: default
→ BLAS:   /usr/lib64/blas/blis/libblas.so.3
→ LAPACK: /usr/lib64/libopenblas_haswellp-r0.3.14.so
→
→ locale:
→  [1] LC_CTYPE=en_US             LC_NUMERIC=C
→  [3] LC_TIME=en_US              LC_COLLATE=en_US
→  [5] LC_MONETARY=en_US          LC_MESSAGES=ca_AD.UTF-8
→  [7] LC_PAPER=ca_AD.UTF-8       LC_NAME=C
→  [9] LC_ADDRESS=C               LC_TELEPHONE=C
→ [11] LC_MEASUREMENT=ca_AD.UTF-8 LC_IDENTIFICATION=C
→
→ attached base packages:
→ [1] parallel  grid      stats     graphics  grDevices utils     datasets
→ [8] methods   base
→
→ other attached packages:
→  [1] foreign_0.8-81      scales_1.1.1        stringr_1.4.0
→  [4] GGally_2.1.1        gridExtra_2.3       ggthemes_4.2.4
→  [7] extrafont_0.17      colorspace_2.0-0    forcats_0.5.1
→ [10] kableExtra_1.3.4    ggmcmc_1.5.1.1      ggplot2_3.3.3
→ [13] tidyr_1.1.3         dplyr_1.0.5         runjags_2.2.0-2
→ [16] rjags_4-10          coda_0.19-4         tufte_0.9
→ [19] tikzDevice_0.12.3.1 rmarkdown_2.7       knitr_1.32
→ [22] colorout_1.2-2
→
→ loaded via a namespace (and not attached):
→  [1] Rcpp_1.0.6         svglite_2.0.0      lattice_0.20-41    ps_1.6.0
→  [5] assertthat_0.2.1   digest_0.6.27      utf8_1.2.1         R6_2.5.0
→  [9] plyr_1.8.6         evaluate_0.14      httr_1.4.2         pillar_1.6.0
→ [13] rlang_0.4.10       rstudioapi_0.13    extrafontdb_1.0    labeling_0.4.2
→ [17] webshot_0.5.2      munsell_0.5.0      compiler_4.0.5     xfun_0.22
→ [21] pkgconfig_2.0.3    systemfonts_1.0.1  htmltools_0.5.1.1  tidyselect_1.1.0
→ [25] tibble_3.1.1       bookdown_0.21      reshape_0.8.8      fansi_0.4.2
```

```
→ [29] viridisLite_0.4.0  crayon_1.4.1      withr_2.4.2       Rttf2pt1_1.3.8
→ [33] gtable_0.3.0       lifecycle_1.0.0   DBI_1.1.1         magrittr_2.0.1
→ [37] cli_2.4.0          stringi_1.5.3     farver_2.1.0      xml2_1.3.2
→ [41] ellipsis_0.3.1     generics_0.1.0    vctrs_0.3.7       RColorBrewer_1.1-2
→ [45] tools_4.0.5        glue_1.4.2        purrr_0.3.4       yaml_2.2.1
→ [49] filehash_2.4-2     rvest_1.0.0
```