教育代际传递

王敏杰

2019-07-20

本文系重复南京大学的《教育人力资本的代际传递研究》

```
library(tidyverse)
library(here)
library(fs)
library(purrr)
library(haven)
library(broom)
```

1 摘要

改革开放以来, 我国经济社会得到飞速发展, 但是也存在着贫富差距加大和社会阶层固化等问题。代际传递是影响社会流动的一个重要方面。教育代际传递性越强说明教育代际流动性越差, 社会阶层固化现象越严重。教育人力资本作为人力资本的重要部分, 受教育水平的高低往往代表一个人的人力资本存量水平。本文采用 2016 年的中国家庭追踪调查数据 (CFPS), 来对教育人力资本的代际传递问题进行研究, 在相关理论基础上, 主要运用描述性统计、教育代际转换矩阵和有序 logit 回归进行实证分析。本文不仅对整体样本进行回归分析, 还对样本进行分类分析。研究结果表明, 父母的受教育程度对子女的受教育水平有显著的正向影响, 母亲相对父亲影响作用更大, 母亲的受教育水平提高时, 子女处于低水平学历的概率减少的更多, 处于高水平的学历的概率增加的更大。对家庭内部而言, 当父母的受教育水平相匹配时, 更有利于子女的受教育水平的提高。从子女的性别差异来看, 当父母的整体受教育程度提高时, 女性的受教育水平向上提高一个级别的概率更大, 而且女性接受高等教育的可能性也更大。从教育代际流动性来看, 农村的教育代际流动性低于城市, 西部地区的教育代际流动性低于中部和东部地区,70 后群体的教育代际流动性低于 80 后群体。总体来说, 我国教育代际流动性增强, 教育不公平性降低, 但是我国教育在城乡和区域之间还存在较大差异。

2 思路

- 成人表，限定：出生日期在 1970 年到 1989 年这段时间的被调查者
- 家庭关系表，找到被调查者父母的 pid
- 最后成人表中根据父母的 pid，找到父母的教育情况
- 形成数据框 pid, edu, mother_pid, father_pid, mother_edu, father_edu
- lmm 模型

高等教育代际传递：

- 人力资本
- 社会资本
- 70 后、80 后年龄段
- 子女性别
- 父母教育程度
- 父母特征（离异，单亲，收入）
- 地区
- 城乡
- 少数民族
- 户籍

3 数据

3.1 成人表

```
cfps2016adult <- read_dta("../data/2016AllData/cfps2016adult_201808.dta",
  encoding = "GB2312"
)
```

筛选**成人表**中相关的变量

```
pre_adult <- cfps2016adult %>%
  dplyr::select(
    pid,                      # 个人 ID
    # fid16,                  # 2016 年家庭样本编码
    # provcd16,               # 2016 年省国标码
    # countyid16,             # 2016 年区县顺序码
    # urban16,                # 基于国家统计局资料的城乡分类
    cfps_birthy,              # 出生年份
    cfps_gender,              # 性别
    cfps_age,                 # 年龄
    qp201,                    # 健康状况
    qea0,                     # 当前婚姻状态
    qn4001,                   # 是否是党员
    pa301                       # 现在的户口状况
  ) %>%
  filter(between(cfps_birthy, 1970, 1989))

pre_adult
#> # A tibble: 12,763 x 8
```

```
#>         pid cfps_birthy cfps_gender  cfps_age   qp201    qea0 qn4001   pa301
#>     <dbl+l>    <dbl+lbl>    <dbl+lbl> <dbl+lbl> <dbl+l> <dbl+l> <dbl+> <dbl+l>
#>   1  1.01e8        1987      0 [女]        29 4 [一般]~  2 [在婚 (~ 0 [否] 1 [农业户 ~
#>   2  1.02e8        1989      1 [男]        28 3 [比较健 ~ 2 [在婚 (~ 0 [否] 1 [农业户 ~
#>   3  1.02e8        1986      0 [女]        30 2 [很健康 ~ 2 [在婚 (~ 0 [否] 1 [农业户 ~
#>   4  1.03e8        1986      0 [女]        31 3 [比较健 ~ 2 [在婚 (~ 0 [否] 3 [非农业 ~
#>   5  1.04e8        1987      0 [女]        29 3 [比较健 ~ 2 [在婚 (~ 1 [是] 1 [农业户 ~
#>   6  1.04e8        1982      1 [男]        34 2 [很健康 ~ 2 [在婚 (~ 1 [是] 3 [非农业 ~
#>   7  1.07e8        1987      1 [男]        29 1 [非常健 ~ 2 [在婚 (~ 0 [否] 1 [农业户 ~
#>   8  1.07e8        1987      0 [女]        29 3 [比较健 ~ 2 [在婚 (~ 0 [否] 1 [农业户 ~
#>   9  1.08e8        1989      1 [男]        28 1 [非常健 ~ 2 [在婚 (~ 0 [否] 1 [农业户 ~
#> 10  1.08e8        1986      1 [男]        31 2 [很健康 ~ 2 [在婚 (~ 1 [是] 3 [非农业 ~
#> # ... with 12,753 more rows
```

3.2 家庭关系表

```
cfps2016famconf <- read_dta("../data/2016AllData/cfps2016famconf_201804.dta",
  encoding = "GB2312"
)
```

筛选**家庭关系表**中相关的变量

```
pre_family <- cfps2016famconf %>%
  dplyr::select(
    pid,                 # 个人样本编码
    # fid16,             # 2016 年家庭样本编码
    fid_provcd16,        # 2016 年省国标码
    # fid_countyid16,    # 2016 年区县顺序码
    fid_urban16,         # 基于国家统计局资料的城乡分类
    tb4_a16_p,           # 个人最高学历
    tb4_a16_f,           # 父亲最高学历
    tb4_a16_m            # 母亲最高学历
  )

pre_family
#> # A tibble: 58,179 x 6
#>         pid fid_provcd16 fid_urban16      tb4_a16_p tb4_a16_f     tb4_a16_m
#>     <dbl+l>     <dbl+lbl>    <dbl+lbl>      <dbl+lbl>  <dbl+lbl>     <dbl+lbl>
#>   1  1.00e8   11 [北京市]     1 [城镇] 4 [高中/中专/技校/职 ~  2 [小学]       1 [文盲/半文盲]~
#>   2  1.00e8   11 [北京市]     1 [城镇] 4 [高中/中专/技校/职 ~ -8 [不适用]~   -8 [不适用]
```

```
#>   3  1.00e8  13 [河北省]     1 [城镇] 3 [初中]                 -8 [不适用]~   -8 [不适用]
#>   4  1.00e8  13 [河北省]     0 [乡村] 1 [文盲/半文盲]~          3 [初中]        4 [高中/中专/技校 ~
#>   5  1.00e8  43 [湖南省]     1 [城镇] 1 [文盲/半文盲]~          6 [大学本科]~  6 [大学本科]
#>   6  1.00e8  43 [湖南省]     1 [城镇] 6 [大学本科]             -8 [不适用]~   -8 [不适用]
#>   7  1.00e8  43 [湖南省]     1 [城镇] 1 [文盲/半文盲]~          6 [大学本科]~  6 [大学本科]
#>   8  1.01e8  13 [河北省]     1 [城镇] 6 [大学本科]             -8 [不适用]~   -8 [不适用]
#>   9  1.01e8  13 [河北省]     1 [城镇] 1 [文盲/半文盲]~          5 [大专]        6 [大学本科]
#> 10  1.01e8  13 [河北省]     1 [城镇] 3 [初中]                 -8 [不适用]~   -8 [不适用]
#> # ... with 58,169 more rows
```

3.3 合并

```
df_set <- pre_adult %>% left_join(pre_family, by = "pid")
df_set
#> # A tibble: 12,763 x 13
#>         pid cfps_birthy cfps_gender cfps_age   qp201    qea0 qn4001   pa301
#>     <dbl+>   <dbl+lbl>   <dbl+lbl> <dbl+lb> <dbl+l> <dbl+l> <dbl+> <dbl+l>
#>   1 1.01e8        1987       0 [女]       29 4 [一般]~  2 [在婚 (~ 0 [否] 1 [农业户 ~
#>   2 1.02e8        1989       1 [男]       28 3 [比较健 ~ 2 [在婚 (~ 0 [否] 1 [农业户 ~
#>   3 1.02e8        1986       0 [女]       30 2 [很健康 ~ 2 [在婚 (~ 0 [否] 1 [农业户 ~
#>   4 1.03e8        1986       0 [女]       31 3 [比较健 ~ 2 [在婚 (~ 0 [否] 3 [非农业 ~
#>   5 1.04e8        1987       0 [女]       29 3 [比较健 ~ 2 [在婚 (~ 1 [是] 1 [农业户 ~
#>   6 1.04e8        1982       1 [男]       34 2 [很健康 ~ 2 [在婚 (~ 1 [是] 3 [非农业 ~
#>   7 1.07e8        1987       1 [男]       29 1 [非常健 ~ 2 [在婚 (~ 0 [否] 1 [农业户 ~
#>   8 1.07e8        1987       0 [女]       29 3 [比较健 ~ 2 [在婚 (~ 0 [否] 1 [农业户 ~
#>   9 1.08e8        1989       1 [男]       28 1 [非常健 ~ 2 [在婚 (~ 0 [否] 1 [农业户 ~
#> 10 1.08e8        1986       1 [男]       31 2 [很健康 ~ 2 [在婚 (~ 1 [是] 3 [非农业 ~
#> # ... with 12,753 more rows, and 5 more variables: fid_provcd16 <dbl+lbl>,
#> #   fid_urban16 <dbl+lbl>, tb4_a16_p <dbl+lbl>, tb4_a16_f <dbl+lbl>,
#> #   tb4_a16_m <dbl+lbl>
```

```
df_set %>%
  map(~ count(data.frame(x = .x), x))
```

```
df_set %>% colnames()
#>  [1] "pid"          "cfps_birthy"  "cfps_gender"  "cfps_age"
#>  [5] "qp201"        "qea0"         "qn4001"       "pa301"
#>  [9] "fid_provcd16" "fid_urban16"  "tb4_a16_p"    "tb4_a16_f"
#> [13] "tb4_a16_m"
```

```r
df_set %>%
  count(pa301)
#> # A tibble: 6 x 2
#>                      pa301     n
#>                  <dbl+lbl> <int>
#> 1 -1 [不知道]                  2
#> 2  1 [农业户口]             8468
#> 3  3 [非农业户口]           3151
#> 4  5 [没有户口]               12
#> 5 79 [不适用 (非中国国籍)]     3
#> 6 NA                        1127
```

```r
a <- df_set %>%
  count(fid_provcd16) %>%
  surveytoolbox::extract_vallab("fid_provcd16")

b <- df_set %>%
  count(fid_provcd16)

w <- b %>% left_join(a, by = c("fid_provcd16" = "id") )
w
#> # A tibble: 31 x 3
#>          fid_provcd16     n fid_provcd16.y
#>             <dbl+lbl> <int> <chr>
#>  1 11 [北京市]         109 北京市
#>  2 12 [天津市]          87 天津市
#>  3 13 [河北省]         752 河北省
#>  4 14 [山西省]         521 山西省
#>  5 15 [内蒙古自治区]     4 内蒙古自治区
#>  6 21 [辽宁省]         986 辽宁省
#>  7 22 [吉林省]         230 吉林省
#>  8 23 [黑龙江省]       332 黑龙江省
#>  9 31 [上海市]         594 上海市
#> 10 32 [江苏省]         249 江苏省
#> # ... with 21 more rows
```

```r
w %>% filter(n < 100) %>%
      mutate(sum = sum(n))
#> # A tibble: 7 x 4
#>             fid_provcd16     n fid_provcd16.y     sum
```

```
#>                <dbl+lbl> <int> <chr>            <int>
#> 1 12 [天津市]               87 天津市             128
#> 2 15 [内蒙古自治区]          4 内蒙古自治区       128
#> 3 46 [海南省]                8 海南省             128
#> 4 54 [西藏自治区]            1 西藏自治区         128
#> 5 63 [青海省]                3 青海省             128
#> 6 64 [宁夏回族自治区]        4 宁夏回族自治区     128
#> 7 65 [新疆维吾尔自治区]     21 新疆维吾尔自治区   128
```

中国 34 个省级行政区：

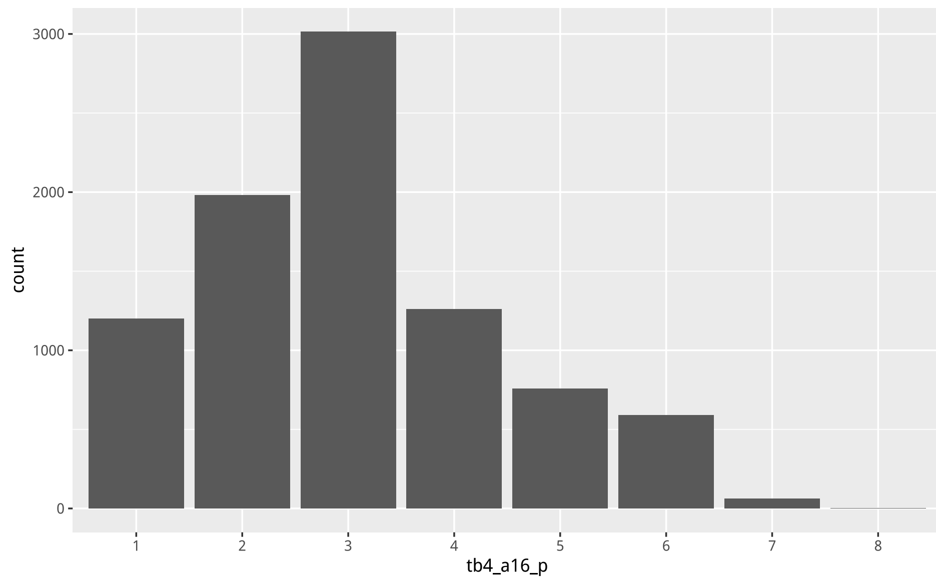
- 中部地区，包括湖北 42、湖南 43、河南 41、安徽 34、江西 36、山西 14 六个相邻省份
- 西部地区，包括西藏 54、新疆 65、青海 63、甘肃 62、宁夏 64、云南 53、贵州 52、四川 51、陕西 61、重庆 50、广西 45、内蒙古 15
- 东部地区，包括广东 44、福建 35、浙江 33、江苏 32、山东 37、上海 31、北京 11、天津 12、河北 13
- 其他地区，辽宁省 21、吉林省 22、黑龙江省 23、海南省 46

```
tb <- df_set %>%
  # 区域
  # mutate(region = case_when(
  #   fid_provcd16 %in% c(-1, -2) ~ eastern,
  #   fid_provcd16 %in% c(-8) ~ central,
  #   fid_provcd16 %in% c(-8) ~ western,
  #   TRUE ~ other
  # )) %>%

  # 城乡分类
  filter(fid_urban16 %in% c(0, 1)) %>%

  # 现在的户口状况
  filter(pa301 %in% c(1, 3)) %>%

  # 健康状态
  filter(qp201 %in% c(1, 2, 3, 4, 5)) %>%

  # 当前婚姻状态
  filter(qea0 %in% c(1, 2, 3, 4, 5)) %>%

  # 是否是党员
  filter(qn4001 %in% c(1, 0)) %>%
```

```r
  # 个人最高学历
  #filter(tb4_a16_p %in% 1:8) %>%

  # # 父亲最高学历
   #filter(!tb4_a16_f %in% c(-9, -8, -1, 0)) %>%

  # # 母亲最高学历
   #filter(!tb4_a16_m %in% c(-9, -8, -1, 0)) %>%

  # 学历
  filter_at(vars(tb4_a16_p:tb4_a16_m), all_vars(. %in% 1:8) ) %>%



  identity()
```

```r
tb
#> # A tibble: 8,868 x 13
#>        pid cfps_birthy cfps_gender cfps_age   qp201    qea0 qn4001   pa301
#>      <dbl+>   <dbl+lbl>   <dbl+lbl> <dbl+lb> <dbl+l> <dbl+l> <dbl+> <dbl+l>
#>   1 1.02e8        1989       1 [男]       28 3 [比较健 ~ 2 [在婚（~ 0 [否] 1 [农业户 ~
#>   2 1.04e8        1982       1 [男]       34 2 [很健康 ~ 2 [在婚（~ 1 [是] 3 [非农业 ~
#>   3 1.09e8        1974       1 [男]       43 3 [比较健 ~ 2 [在婚（~ 0 [否] 1 [农业户 ~
#>   4 1.09e8        1985       1 [男]       31 3 [比较健 ~ 2 [在婚（~ 0 [否] 3 [非农业 ~
#>   5 1.10e8        1971       1 [男]       45 3 [比较健 ~ 2 [在婚（~ 0 [否] 3 [非农业 ~
#>   6 1.10e8        1989       1 [男]       28 3 [比较健 ~ 1 [未婚]~  0 [否] 3 [非农业 ~
#>   7 1.10e8        1973       0 [女]       43 3 [比较健 ~ 2 [在婚（~ 0 [否] 1 [农业户 ~
#>   8 1.10e8        1974       1 [男]       42 3 [比较健 ~ 2 [在婚（~ 0 [否] 3 [非农业 ~
#>   9 1.10e8        1978       0 [女]       38 3 [比较健 ~ 2 [在婚（~ 0 [否] 3 [非农业 ~
#> 10 1.10e8        1976       1 [男]       40 2 [很健康 ~ 2 [在婚（~ 0 [否] 1 [农业户 ~
#> # ... with 8,858 more rows, and 5 more variables: fid_provcd16 <dbl+lbl>,
#> #   fid_urban16 <dbl+lbl>, tb4_a16_p <dbl+lbl>, tb4_a16_f <dbl+lbl>,
#> #   tb4_a16_m <dbl+lbl>
```

```r
tb %>%
  mutate_at(vars(tb4_a16_p), as.factor) %>%
  ggplot(aes(x = tb4_a16_p)) +
  geom_bar(scale = 4)
```




```r
library(summarytools)

view(dfSummary(tb))
```

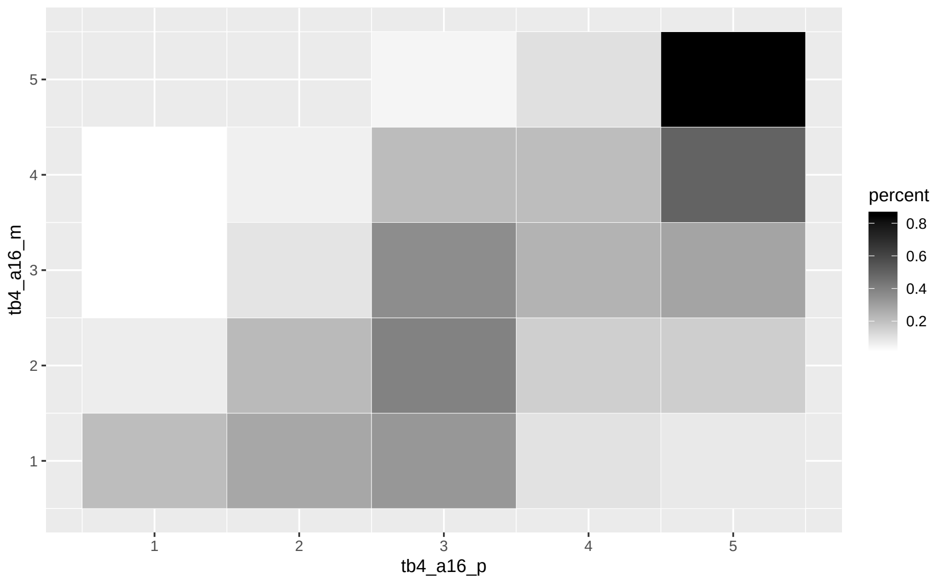
4 代际转换矩阵分析

```r
tb1.1 <- tb %>%
  haven::zap_labels() %>%
  mutate_at(
    vars(tb4_a16_p:tb4_a16_m),
    list(~ case_when(
      . %in% c(5, 6, 7, 8) ~ 5,
      TRUE ~ .
    ))
  )

tb1.1
#> # A tibble: 8,868 x 13
#>        pid cfps_birthy cfps_gender cfps_age qp201  qea0 qn4001 pa301
#>      <dbl>       <dbl>       <dbl>    <dbl> <dbl> <dbl>  <dbl> <dbl>
#>   1 1.02e8        1989           1       28     3     2      0     1
#>   2 1.04e8        1982           1       34     2     2      1     3
#>   3 1.09e8        1974           1       43     3     2      0     1
#>   4 1.09e8        1985           1       31     3     2      0     3
#>   5 1.10e8        1971           1       45     3     2      0     3
```

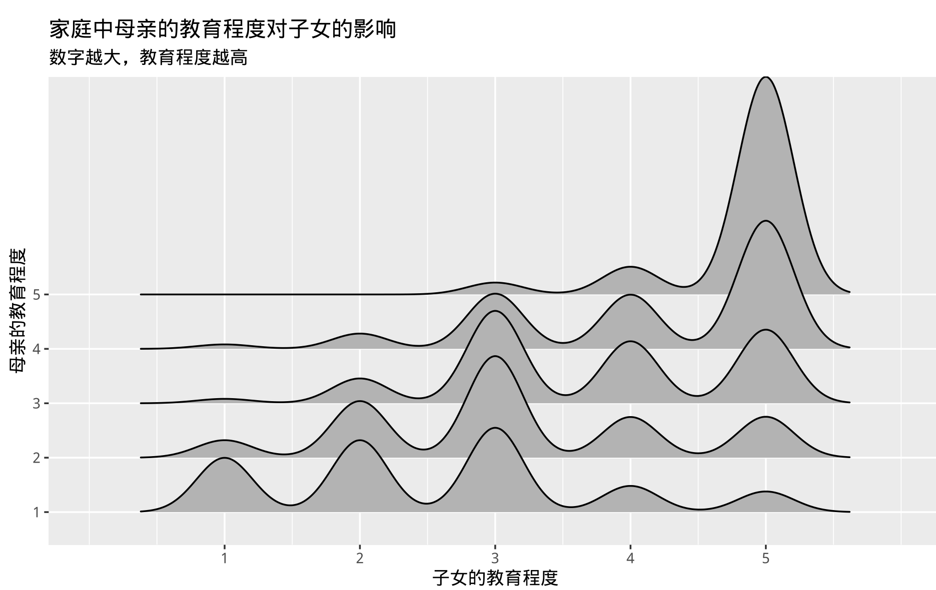
```
#>   6 1.10e8        1989           1        28     3     1      0     3
#>   7 1.10e8        1973           0        43     3     2      0     1
#>   8 1.10e8        1974           1        42     3     2      0     3
#>   9 1.10e8        1978           0        38     3     2      0     3
#> 10 1.10e8        1976           1        40     2     2      0     1
#> # ... with 8,858 more rows, and 5 more variables: fid_provcd16 <dbl>,
#> #   fid_urban16 <dbl>, tb4_a16_p <dbl>, tb4_a16_f <dbl>, tb4_a16_m <dbl>
```

```
tb1.1 %>%
  count(tb4_a16_m, tb4_a16_f) %>%
  group_by(tb4_a16_m) %>%
  mutate(percent = n/sum(n) ) %>%
  dplyr::select(-n) %>%
  pivot_wider(names_from = tb4_a16_f,
              values_from = percent)
#> # A tibble: 5 x 6
#>   tb4_a16_m    `1`    `2`   `3`    `4`     `5`
#>       <dbl>  <dbl>  <dbl> <dbl>  <dbl>   <dbl>
#> 1         1 0.469   0.279 0.170 0.0741 0.00722
#> 2         2 0.129   0.469 0.271 0.115  0.0153
#> 3         3 0.0809  0.198 0.457 0.221  0.0428
#> 4         4 0.0785  0.153 0.285 0.393  0.0900
#> 5         5 0.0308 NA     0.2   0.277  0.492
```

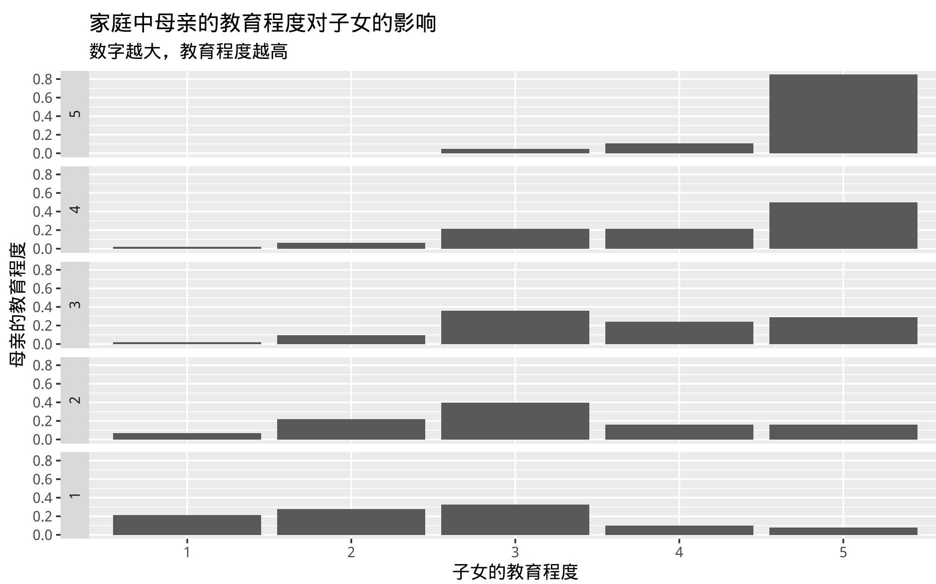
```
tb1.1 %>%
  count(tb4_a16_m, tb4_a16_p) %>%
  group_by(tb4_a16_m) %>%
  mutate(percent = n/sum(n) ) %>%
  ggplot(aes(x = tb4_a16_p, y = tb4_a16_m, fill = percent)) +
  geom_tile(color = "white") +
  scale_fill_gradient(low = "white", high = "black")
```




```r
library(ggridges)
tb1.1 %>%
  mutate_at(vars(tb4_a16_m), as.factor) %>%
  ggplot(aes(x = tb4_a16_p, y = tb4_a16_m)) +
  geom_density_ridges(scale = 4) +
  scale_x_continuous(limits = c(0, 6), breaks = c(1:5)) +
  labs(title = "家庭中母亲的教育程度对子女的影响",
       subtitle = "数字越大，教育程度越高",
       x = "子女的教育程度",
       y = "母亲的教育程度")
```



```r
tb1.1 %>%
  count(tb4_a16_m, tb4_a16_p) %>%
  group_by(tb4_a16_m) %>%
  mutate(percent = n/sum(n) ) %>%
  ungroup() %>%
  mutate_at(vars(tb4_a16_m:tb4_a16_p), as.factor) %>%
  mutate_at(vars(tb4_a16_m), ~forcats::fct_rev(.)) %>%
  ggplot(aes(x = tb4_a16_p, y = percent)) +
  geom_col() +
  facet_grid(vars(tb4_a16_m), switch = "y") +
  labs(title = " 家庭中母亲的教育程度对子女的影响",
       subtitle = " 数字越大，教育程度越高",
       x = " 子女的教育程度",
       y = " 母亲的教育程度")
```



5 有序 logistic 回归分析

Ordinal logistic regression model

```r
# xb = c(1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
# lf = c(1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0)
# lx = c(1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3)
# ps = c(16, 5, 6, 6, 7, 19, 5, 2, 7, 1, 0, 10)
#
# table <- data.frame(xb, lf, lx, ps)
#
```

```r
# library(MASS)
# fit <- polr( as.ordered(lx) ~ xb + lf, weight = ps, Hess = T, data = table)
# summary(fit)
```

```r
tb1.2 <- tb1.1 %>%
  dplyr::select(
    edu = tb4_a16_p,
    f_edu = tb4_a16_f,
    m_edu = tb4_a16_m,
    sex =  cfps_gender ) %>%
  mutate_at(vars(edu, f_edu, m_edu, sex), as.factor) %>%
  #mutate_at(vars(edu), ~fct_inorder(., ordered = TRUE))
  mutate_at(vars(edu), ~fct_inseq(., ordered = TRUE))

tb1.2
#> # A tibble: 8,868 x 4
#>    edu   f_edu m_edu sex
#>    <ord> <fct> <fct> <fct>
#>  1 5     3     2     1
#>  2 5     4     4     1
#>  3 3     1     1     1
#>  4 4     4     4     1
#>  5 4     3     2     1
#>  6 5     1     1     1
#>  7 2     2     1     0
#>  8 4     2     2     1
#>  9 5     4     2     0
#> 10 3     3     2     1
#> # ... with 8,858 more rows
```

```r
tb1.2 %>% write_rds("tb1.2.rds")
tb1.2 <- read_rds("tb1.2.rds")
```

```r
tb1.2 %>% pull(edu) %>% levels()
#> [1] "1" "2" "3" "4" "5"
```

5.1 MASS 包 polr

```r
library(MASS)
# https://stats.idre.ucla.edu/r/dae/ordinal-logistic-regression/
```

```r
# https://towardsdatascience.com/implementing-and-interpreting-ordinal-logistic-regression-1ee699274c
mod_mass <- polr(edu ~ f_edu + m_edu + sex,
                 data = tb1.2,
                 Hess = TRUE)

summary(mod_mass)
#> Call:
#> polr(formula = edu ~ f_edu + m_edu + sex, data = tb1.2, Hess = TRUE)
#>
#> Coefficients:
#>          Value Std. Error t value
#> f_edu2 0.6839    0.05218  13.105
#> f_edu3 1.1287    0.05745  19.647
#> f_edu4 1.4422    0.07187  20.066
#> f_edu5 2.4877    0.16115  15.437
#> m_edu2 0.6552    0.04987  13.140
#> m_edu3 1.2782    0.06196  20.631
#> m_edu4 1.9674    0.09362  21.014
#> m_edu5 3.3211    0.35888   9.254
#> sex1   0.4057    0.03935  10.309
#>
#> Intercepts:
#>     Value    Std. Error t value
#> 1|2  -0.7427   0.0460   -16.1287
#> 2|3   0.7154   0.0451    15.8499
#> 3|4   2.4386   0.0516    47.2225
#> 4|5   3.4299   0.0571    60.0252
#>
#> Residual Deviance: 24873.97
#> AIC: 24899.97
```

```r
library(broom)
broom::tidy(mod_mass)
#> # A tibble: 13 x 5
#>    term   estimate std.error statistic coefficient_type
#>    <chr>     <dbl>     <dbl>     <dbl> <chr>
#>  1 f_edu2    0.684    0.0522      13.1  coefficient
#>  2 f_edu3    1.13     0.0574      19.6  coefficient
#>  3 f_edu4    1.44     0.0719      20.1  coefficient
#>  4 f_edu5    2.49     0.161       15.4  coefficient
```

```
#>   5 m_edu2     0.655    0.0499     13.1  coefficient
#>   6 m_edu3     1.28     0.0620     20.6  coefficient
#>   7 m_edu4     1.97     0.0936     21.0  coefficient
#>   8 m_edu5     3.32     0.359       9.25 coefficient
#>   9 sex1       0.406    0.0394     10.3  coefficient
#> 10 1|2       -0.743    0.0460    -16.1  zeta
#> 11 2|3        0.715    0.0451     15.8  zeta
#> 12 3|4        2.44     0.0516     47.2  zeta
#> 13 4|5        3.43     0.0571     60.0  zeta
```

5.2 ordinal 包

```
library(ordinal)
mod_ordinal <- clm(edu ~ f_edu + m_edu + sex,
                   data = tb1.2,
                   link = "logit",
                   thresholds = "flexible"
                  )

broom::tidy(mod_ordinal)
#> # A tibble: 13 x 6
#>    term   estimate std.error statistic  p.value coefficient_type
#>    <chr>     <dbl>     <dbl>     <dbl>    <dbl> <chr>
#>  1 1|2      -0.743    0.0460    -16.1  1.60e-58 alpha
#>  2 2|3       0.715    0.0451     15.8  1.41e-56 alpha
#>  3 3|4       2.44     0.0516     47.2  0.       alpha
#>  4 4|5       3.43     0.0571     60.0  0.       alpha
#>  5 f_edu2    0.684    0.0522     13.1  3.08e-39 beta
#>  6 f_edu3    1.13     0.0574     19.6  6.08e-86 beta
#>  7 f_edu4    1.44     0.0719     20.1  1.45e-89 beta
#>  8 f_edu5    2.49     0.161      15.4  9.26e-54 beta
#>  9 m_edu2    0.655    0.0499     13.1  1.95e-39 beta
#> 10 m_edu3    1.28     0.0620     20.6  1.45e-94 beta
#> 11 m_edu4    1.97     0.0936     21.0  4.91e-98 beta
#> 12 m_edu5    3.32     0.359       9.25 2.16e-20 beta
#> 13 sex1      0.406    0.0394     10.3  6.43e-25 beta
```

6 Bayesian framework

https://kevinstadler.github.io/blog/bayesian-ordinal-regression-with-random-effects-using-brms/

```
tb1.3 <- tb1.2 #%>% mutate(edu = fct_inorder(edu, ordered = TRUE))
#tb1.3
```

```
mod_brms2 <- brm(edu ~ f_edu + m_edu + sex,
                data = tb1.3,
                family = cumulative(link = "logit"),
                prior = c(prior(normal(0, 10), class = Intercept),
                          prior(normal(0, 10), class = b)),
                iter = 2000, warmup = 1000, cores = 2, chains = 2,
                inits = 0
               )
```

```
bform <- bf(
  Score ~ Species + Region + (1|ID),
  disc ~ <your predictors>
)
M.SR <- brm(bform, data = IUULong, family = cumulative)
```

这个 disc 什么意思

```
loo_compare(mod_brms,mod_brms2)
```

```
summary(mod_brms)
#>  Family: cumulative
#>   Links: mu = logit; disc = identity
#> Formula: edu ~ f_edu + m_edu + sex
#>    Data: tb1.3 (Number of observations: 8868)
#> Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
#>          total post-warmup samples = 4000
#>
#> Population-Level Effects:
#>              Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
#> Intercept[1]    -0.74      0.05    -0.83    -0.65       4451 1.00
#> Intercept[2]     0.72      0.05     0.63     0.81       4077 1.00
#> Intercept[3]     2.44      0.05     2.34     2.54       3680 1.00
#> Intercept[4]     3.43      0.06     3.32     3.54       3581 1.00
#> f_edu2           0.68      0.05     0.58     0.79       3619 1.00
#> f_edu3           1.13      0.06     1.02     1.24       3144 1.00
#> f_edu4           1.44      0.07     1.30     1.58       3692 1.00
```

```
#> f_edu5           2.49      0.16     2.19     2.81        4630 1.00
#> m_edu2           0.66      0.05     0.55     0.75        3493 1.00
#> m_edu3           1.28      0.06     1.16     1.40        3092 1.00
#> m_edu4           1.97      0.09     1.79     2.16        3322 1.00
#> m_edu5           3.36      0.37     2.66     4.14        4739 1.00
#> sex1             0.41      0.04     0.33     0.48        4358 1.00
#>
#> Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
#> is a crude measure of effective sample size, and Rhat is the potential
#> scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
mod_brms %>%
  fixef() %>%
  inv_logit_scaled()
#>               Estimate Est.Error      Q2.5     Q97.5
#> Intercept[1] 0.3222728 0.5115304 0.3027821 0.3418688
#> Intercept[2] 0.6718145 0.5114026 0.6521577 0.6916798
#> Intercept[3] 0.9198417 0.5129786 0.9122747 0.9271664
#> Intercept[4] 0.9687132 0.5142414 0.9651538 0.9719302
#> f_edu2       0.6646727 0.5128096 0.6419024 0.6870248
#> f_edu3       0.7557355 0.5141635 0.7348004 0.7759580
#> f_edu4       0.8089485 0.5177114 0.7864676 0.8298084
#> f_edu5       0.9234457 0.5396801 0.8990206 0.9432275
#> m_edu2       0.6583263 0.5126977 0.6352876 0.6799780
#> m_edu3       0.7822547 0.5152219 0.7615582 0.8023547
#> m_edu4       0.8775007 0.5235837 0.8569375 0.8962772
#> m_edu5       0.9664032 0.5926308 0.9345263 0.9842600
#> sex1         0.6003698 0.5098171 0.5814625 0.6185164
```

```
library(tidybayes)
mod_brms %>% get_variables()
#>  [1] "b_Intercept[1]" "b_Intercept[2]" "b_Intercept[3]" "b_Intercept[4]"
#>  [5] "b_f_edu2"       "b_f_edu3"       "b_f_edu4"       "b_f_edu5"
#>  [9] "b_m_edu2"       "b_m_edu3"       "b_m_edu4"       "b_m_edu5"
#> [13] "b_sex1"         "lp__"           "accept_stat__"  "stepsize__"
#> [17] "treedepth__"    "n_leapfrog__"   "divergent__"    "energy__"
```

```
mod_brms %>% posterior_samples()
```

```
plot(mod_brms)
```

```r
p1 <-
  posterior_samples(mod_brms) %>%
  dplyr::select(starts_with("b_")) %>%
  mutate_all(inv_logit_scaled) %>%
  gather() %>%
  group_by(key) %>%
  summarise(mean = mean(value),
            sd   = sd(value),
            ll   = quantile(value, probs = .025),
            ul   = quantile(value, probs = .975))
p1
#> # A tibble: 13 x 5
#>    key              mean      sd    ll    ul
#>    <chr>           <dbl>   <dbl> <dbl> <dbl>
#>  1 b_f_edu2        0.665 0.0114  0.642 0.687
#>  2 b_f_edu3        0.756 0.0105  0.735 0.776
#>  3 b_f_edu4        0.809 0.0110  0.786 0.830
#>  4 b_f_edu5        0.923 0.0113  0.899 0.943
#>  5 b_Intercept[1] 0.322 0.0101  0.303 0.342
#>  6 b_Intercept[2] 0.672 0.0101  0.652 0.692
#>  7 b_Intercept[3] 0.920 0.00383 0.912 0.927
#>  8 b_Intercept[4] 0.969 0.00173 0.965 0.972
#>  9 b_m_edu2        0.658 0.0114  0.635 0.680
#> 10 b_m_edu3        0.782 0.0104  0.762 0.802
#> 11 b_m_edu4        0.877 0.0102  0.857 0.896
#> 12 b_m_edu5        0.964 0.0128  0.935 0.984
#> 13 b_sex1          0.600 0.00942 0.581 0.619
```

```r
p <-
  posterior_samples(mod_brms) %>%
  dplyr::select(starts_with("b_")) %>%
  gather() %>%
  group_by(key) %>%
  summarise(mean = mean(value),
            sd   = sd(value),
            ll   = quantile(value, probs = .025),
            ul   = quantile(value, probs = .975))


p
#> # A tibble: 13 x 5
```

```
#>    key               mean     sd      ll      ul
#>    <chr>            <dbl>  <dbl>   <dbl>   <dbl>
#>  1 b_f_edu2         0.684 0.0512  0.584   0.786
#>  2 b_f_edu3         1.13  0.0567  1.02    1.24
#>  3 b_f_edu4         1.44  0.0709  1.30    1.58
#>  4 b_f_edu5         2.49  0.159   2.19    2.81
#>  5 b_Intercept[1] -0.743 0.0461 -0.834  -0.655
#>  6 b_Intercept[2]  0.716 0.0456  0.629   0.808
#>  7 b_Intercept[3]  2.44  0.0519  2.34    2.54
#>  8 b_Intercept[4]  3.43  0.0570  3.32    3.54
#>  9 b_m_edu2         0.656 0.0508  0.555   0.754
#> 10 b_m_edu3         1.28  0.0609  1.16    1.40
#> 11 b_m_edu4         1.97  0.0944  1.79    2.16
#> 12 b_m_edu5         3.36  0.375   2.66    4.14
#> 13 b_sex1           0.407 0.0393  0.329   0.483
```

7 对比

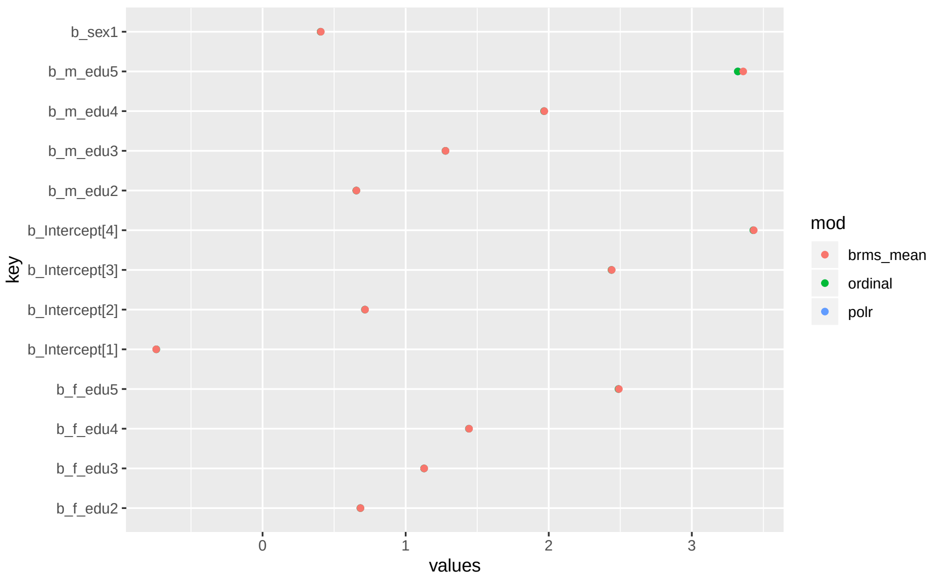
```
a <- broom::tidy(mod_mass) %>%
  dplyr::select(term, polr = estimate) %>%
  mutate(key =  stringr::str_c("b_", term)) %>%
  mutate(key = case_when(
    key == "b_1|2" ~ "b_Intercept[1]",
    key == "b_2|3" ~ "b_Intercept[2]",
    key == "b_3|4" ~ "b_Intercept[3]",
    key == "b_4|5" ~ "b_Intercept[4]",
    TRUE ~ key
    )) %>%
  dplyr::select(-term)
a
#> # A tibble: 13 x 2
#>      polr key
#>     <dbl> <chr>
#>  1  0.684 b_f_edu2
#>  2  1.13  b_f_edu3
#>  3  1.44  b_f_edu4
#>  4  2.49  b_f_edu5
#>  5  0.655 b_m_edu2
```

```
#>   6  1.28   b_m_edu3
#>   7  1.97   b_m_edu4
#>   8  3.32   b_m_edu5
#>   9  0.406 b_sex1
#> 10 -0.743 b_Intercept[1]
#> 11  0.715 b_Intercept[2]
#> 12  2.44   b_Intercept[3]
#> 13  3.43   b_Intercept[4]
```

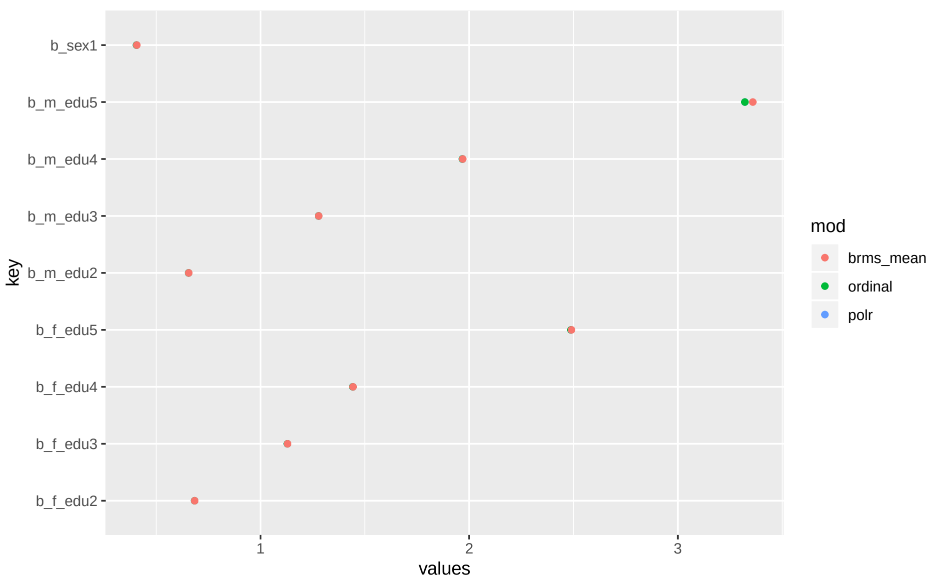
```
b <-
  broom::tidy(mod_ordinal) %>%
  dplyr::select(term, ordinal = estimate) %>%
  mutate(key =  stringr::str_c("b_", term)) %>%
  mutate(key = case_when(
    key == "b_1|2" ~ "b_Intercept[1]",
    key == "b_2|3" ~ "b_Intercept[2]",
    key == "b_3|4" ~ "b_Intercept[3]",
    key == "b_4|5" ~ "b_Intercept[4]",
    TRUE ~ key
    )) %>%
  dplyr::select(-term)
b
#> # A tibble: 13 x 2
#>    ordinal key
#>      <dbl> <chr>
#>  1  -0.743 b_Intercept[1]
#>  2   0.715 b_Intercept[2]
#>  3   2.44   b_Intercept[3]
#>  4   3.43   b_Intercept[4]
#>  5   0.684 b_f_edu2
#>  6   1.13   b_f_edu3
#>  7   1.44   b_f_edu4
#>  8   2.49   b_f_edu5
#>  9   0.655 b_m_edu2
#> 10   1.28   b_m_edu3
#> 11   1.97   b_m_edu4
#> 12   3.32   b_m_edu5
#> 13   0.406 b_sex1
```

```r
c <- p %>%
  dplyr::select(key, brms_mean = mean)

t <- a %>%
  left_join(b, by = "key") %>%
  left_join(c, by = "key")
t
#> # A tibble: 13 x 4
#>      polr key            ordinal brms_mean
#>     <dbl> <chr>            <dbl>     <dbl>
#>  1  0.684 b_f_edu2         0.684     0.684
#>  2  1.13  b_f_edu3         1.13      1.13
#>  3  1.44  b_f_edu4         1.44      1.44
#>  4  2.49  b_f_edu5         2.49      2.49
#>  5  0.655 b_m_edu2         0.655     0.656
#>  6  1.28  b_m_edu3         1.28      1.28
#>  7  1.97  b_m_edu4         1.97      1.97
#>  8  3.32  b_m_edu5         3.32      3.36
#>  9  0.406 b_sex1           0.406     0.407
#> 10 -0.743 b_Intercept[1]  -0.743    -0.743
#> 11  0.715 b_Intercept[2]   0.715     0.716
#> 12  2.44  b_Intercept[3]   2.44      2.44
#> 13  3.43  b_Intercept[4]   3.43      3.43
```

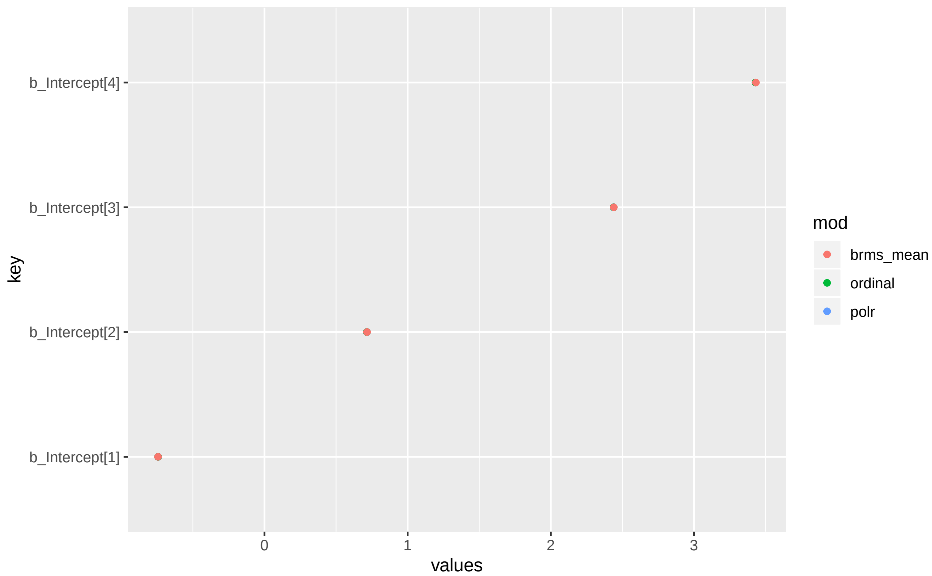
```r
t %>%
  tidyr::pivot_longer(-starts_with("key"), names_to = "mod",  values_to = "values" ) %>%
  ggplot(aes(x = values, y = key, color = mod)) +
  geom_point()
```




```
t %>%
  filter(stringr::str_detect(key, "^b_Intercept", negate = TRUE)) %>%
  tidyr::pivot_longer(-starts_with("key"), names_to = "mod",  values_to = "values" ) %>%
  ggplot(aes(x = values, y = key, color = mod)) +
  geom_point()
```




```
t %>%
  filter(stringr::str_detect(key, "^b_Intercept", negate = FALSE)) %>%
  tidyr::pivot_longer(-starts_with("key"), names_to = "mod",  values_to = "values" ) %>%
  ggplot(aes(x = values, y = key, color = mod)) +
  geom_point()
```



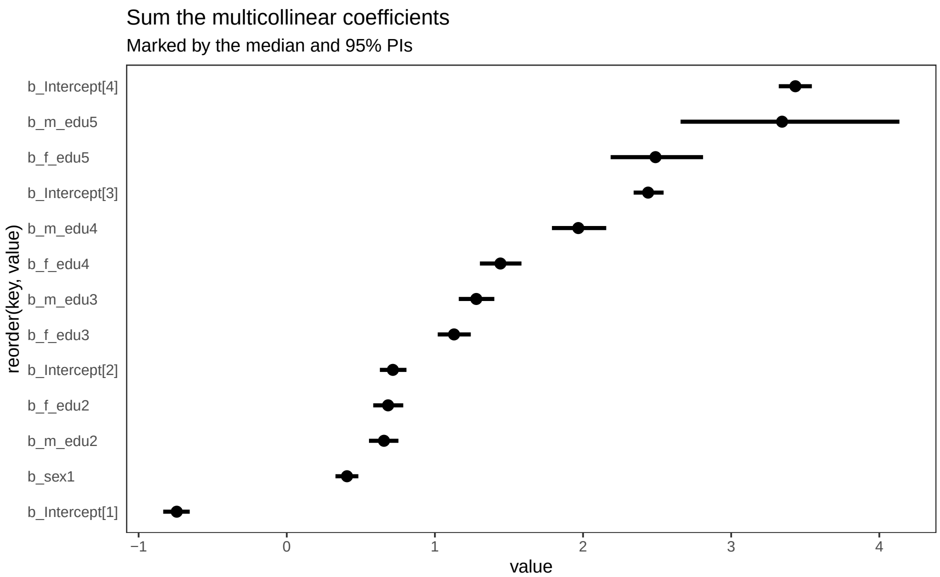

```
t %>%
  filter(stringr::str_detect(key, "^b_Intercept", negate = TRUE)) %>%
  mutate(diff = ordinal - brms_mean)
#> # A tibble: 9 x 5
#>    polr key      ordinal brms_mean      diff
#>   <dbl> <chr>      <dbl>     <dbl>     <dbl>
#> 1 0.684 b_f_edu2   0.684     0.684 -0.000333
#> 2 1.13  b_f_edu3   1.13      1.13  -0.000719
#> 3 1.44  b_f_edu4   1.44      1.44  -0.000956
#> 4 2.49  b_f_edu5   2.49      2.49  -0.00245
#> 5 0.655 b_m_edu2   0.655     0.656 -0.000625
#> 6 1.28  b_m_edu3   1.28      1.28  -0.000662
#> 7 1.97  b_m_edu4   1.97      1.97  -0.00160
#> 8 3.32  b_m_edu5   3.32      3.36  -0.0381
#> 9 0.406 b_sex1     0.406     0.407 -0.00130
```

8 tidybayes

```
# https://mjskay.github.io/tidybayes/articles/tidy-brms.html
posterior_samples(mod_brms) %>%
  dplyr::select(starts_with("b_")) %>%
  gather()
```

```
posterior_samples(mod_brms) %>%
  dplyr::select(starts_with("b_")) %>%
```

```r
  gather() %>%

  ggplot(aes(x = value, y = reorder(key, value))) +
  geom_halfeyeh(fill = "firebrick",
                point_interval = median_qi, .width = .95) +
  labs(title    = "Sum the multicollinear coefficients",
       subtitle = "Marked by the median and 95% PIs") +
  theme_bw() +
  theme(panel.grid   = element_blank(),
        axis.ticks.y = element_blank(),
        axis.text.y  = element_text(hjust = 0))
```




```r
mod_brms %>%
  spread_draws(b_Intercept[condition])
```

```r
mod_brms %>%
  spread_draws(b_Intercept[condition]) %>%
  group_by(condition) %>%   # this line not necessary (done by spread_draws)
  median_qi(b_Intercept)
#> # A tibble: 4 x 7
#>   condition b_Intercept .lower .upper .width .point .interval
#>       <int>       <dbl>  <dbl>  <dbl>  <dbl> <chr>  <chr>
#> 1         1      -0.743 -0.834 -0.655   0.95 median qi
#> 2         2       0.716  0.629  0.808   0.95 median qi
#> 3         3       2.44   2.34   2.54    0.95 median qi
#> 4         4       3.43   3.32   3.54    0.95 median qi
```

```r
#ggplot(aes(y = condition, x = b_Intercept)) +
#geom_halfeyeh()
```

8.1 理论依据

- 理论公式
- brms 代码
- 解释 (不推翻 freq 的解释，增强版)

输出结果得到有序分类 logistic 回归模型中截距和回归系数的最大似然估计值，确定出回归方程为：

$$\begin{aligned}\text{logit}\left(p_1\right) &= \ln\left(\frac{p_1}{p_2+p_3}\right) = -2.667 + 1.319x_1 + 1.797x_2 \\ \text{logit}\left(p_1+p_2\right) &= \ln\left(\frac{p_1+p_2}{p_3}\right) = -1.813 + 1.319x_1 + 1.797x_2\end{aligned}$$

然后 `inv_logit_scaled`