MRP_analysis.R

Nathan

2024-08-01

```
# multilevel regression and post-stratification analysis
# with skills for life survey data and
# Newham resident survey data

library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(glue)
library(gtsummary)
```

```
## Warning: package 'gtsummary' was built under R version 4.3.3
```

```
# # load regression data
# {
#   data <-
#     haven::read_dta(
#       file = "C:/Users/Nathan/Documents/Newham Fellowship/data/Skills for Life Survey 2011/UKDA-7240-
#
#   save(data, file = "data/skills_for_life_data.RData")
# }


load(here::here("data/skills_for_life_data.RData"))


#################
# data cleaning

# select variables
```

```r
data <-
  data |>
  select(
    WORKINGSTATUS2,
    GROSS_ANNUAL_INCOME_OLDBANDS,
    BUK,
    QxTenu1,
    Sex1,
    AGE1NET,
    Sesol,              # is English first language
    ETHNICSIMPLE,
    HIQUAL,
    CLITSPEAK,          # ENFL everyday English skills (literacy and speaking)
    IMDSCOREB4,         # Index of Multiple Deprivation banded into deciles
    NSSEC7,
    # outcomes
    SUMMARYCOMP,        # self-assessed computer skills (summary)
    TSKILLA,            # self-assessed computer skills (summary 2)
    COMBLIT,            # self-assessed reading a writing (summary)
    LiteracyScoreA_1,                   # literacy level
    starts_with("LiteracyThreshold"),   # literacy threshold
    NumeracyScoreA_1,                   # numeracy level
    starts_with("NumeracyThreshold"),   # numeracy threshold,
    MultipleChoiceLevelA_1,             # ICT level
    MultipleChoiceLevelA_1Thres,        # ICT threshold
    # weights
    rimweight2003,
    rimweightLIT2003,
    rimweightNUM2003,
    rimweightICT2003,
    rimweightNUMICT2003,
    rimweightLITICT2003,
    rimweightLITNUM2003
  )


# these are the S4L variables used in
# Rowlands (2015) British Journal of General Practice
#
# job status: National Statistics Socioeconomic Classification 3 bands
# (Managerial/professional, Intermediate, Routine/manual/ students/unemployed)
# employment status: employed, not employed
# gross income: >=10000, <10000
# place of birth: UK, non UK
# home ownership: Owns or part-owns home, does not own home
# sex: male, female
# age: 16-44, >=45
# first language: English, other
# ethnicity: white, black and minority ethnic
# qualification level: NQF >= level at age 16 (level 2), below level 2
# area deprivation: IMD quintiles

# matching with original survey
```

```r
# NSSEC7: 1 Higher managerial and professional
#         2 Lower managerial and professional
#         3 Intermediate
#         4 Small employers and own account workers
#         5 Lower supervisory and technical
#         6 Semi-routine occupations
#         7 Routine occupations
#         8 Never worked/ long term unemployed
#         9 Full-time student
#         10 Not classifiable
# WORKINGSTATUS2: 0-No, 1-Yes
# GROSS_ANNUAL_INCOME_OLDBANDS: {<£5,000, £5,000 - £9,999},
# {£10,000 - £14,999, £15,000 - £19,999, £20,000 - £29,999}
# BUK: 1-Yes, 2-No
# QxTenu1: 1-Own home outright or with a mortgage or loan
# Sex1: 1-Male, 2-Female
# AGE1NET: {16-24, 25-44}, 45-65
# Sesol: 1-Yes, 2-No
# ETHNICSIMPLE: 1-White, 2-BME
# HIQUAL: {1-4), {5-Level 1 qualification or below}
# IMDSCOREB4: 1,...,9

model_dat <-
  data |>
  # remove class
  mutate(
    WORKINGSTATUS2 = unclass(WORKINGSTATUS2),
    GROSS_ANNUAL_INCOME_OLDBANDS = unclass(GROSS_ANNUAL_INCOME_OLDBANDS),
    BUK = unclass(BUK),
    QxTenu1 = unclass(QxTenu1),
    Sex1 = unclass(Sex1),
    AGE1NET = unclass(AGE1NET),
    Sesol = unclass(Sesol),
    ETHNICSIMPLE = unclass(ETHNICSIMPLE),
    HIQUAL = unclass(HIQUAL),
    IMDSCOREB4 = unclass(IMDSCOREB4),
    NSSEC7 = unclass(NSSEC7),
    LiteracyThresholdA_1 = unclass(LiteracyThresholdA_1),
    NumeracyThresholdA_1 = unclass(NumeracyThresholdA_1),
    MultipleChoiceLevelA_1Thres = unclass(MultipleChoiceLevelA_1Thres),
    LiteracyScoreA_1 = unclass(LiteracyScoreA_1),
    NumeracyScoreA_1 = unclass(NumeracyScoreA_1)) |>
  # relabel and order levels
  transmute(
    workingstatus = factor(WORKINGSTATUS2, levels = 1:0, labels = c("Yes","No")),
    gross_income =
      ifelse(GROSS_ANNUAL_INCOME_OLDBANDS %in% 1:2,
             "<10000",
             ifelse(GROSS_ANNUAL_INCOME_OLDBANDS %in% 3:6,
                    ">=10000", "other")) |>
      factor(levels = c(">=10000", "<10000", "other")),
    uk_born = factor(BUK, levels = 1:2, labels = c("Yes", "No")),
    sex = factor(Sex1, levels = c(2,1), c("Female", "Male")),
```

```
    own_home = ifelse(QxTenu1 == 1, "Yes", "No") |>
      factor(levels = c("Yes", "No")),
    age = ifelse(AGE1NET %in% 1:2, "16-44",
                 ifelse(AGE1NET == 3, ">=45", "other")) |>
      factor(levels = c("16-44", ">=45")),
    english_lang = factor(Sesol, levels = 1:2, labels = c("Yes", "No")),
    ethnicity = factor(ETHNICSIMPLE, levels = 1:2, labels = c("White", "BME")),
    qualification = ifelse(HIQUAL %in% 1:4, ">=level 2", "<=Level 1") |>
      factor(levels = c(">=level 2", "<=Level 1")),
    imd = factor(IMDSCOREB4),
    job_status = ifelse(NSSEC7 %in% 1:2, "higher",
                        ifelse(NSSEC7 == 3, "intermediate",
                               ifelse(NSSEC7 %in% 4:10, "lower", "other"))) |>
      factor(levels = c("intermediate", "lower", "higher")),
    lit_thresholdL1 =
      ifelse(LiteracyThresholdA_1 == 1, "below",
             ifelse(LiteracyThresholdA_1 == 2, "above", "other")),
    lit_thresholdL2 = ifelse(LiteracyScoreA_1 == 5, "above",
                             ifelse(LiteracyScoreA_1 %in% 1:4, "below", "other")),   # >= L2
    num_thresholdEL3 =
      ifelse(NumeracyThresholdA_1 == 1, "below",
             ifelse(NumeracyThresholdA_1 == 2, "above", "other")),
    num_thresholdL1 = ifelse(NumeracyScoreA_1 == 4:5, "above",
                             ifelse(NumeracyScoreA_1 %in% 1:3, "below", "other")),  # >= L1
    ict_thresholdEL3 =
      ifelse(MultipleChoiceLevelA_1Thres == 1, "below",
             ifelse(MultipleChoiceLevelA_1Thres == 2, "above", "other")),
    weights = unclass(rimweight2003),
    lit_weightsL1 = unclass(rimweightLIT2003),
    num_weightsEL3 = unclass(rimweightNUM2003),
    ict_weightsEL3 = unclass(rimweightICT2003)
  ) |>
  filter(!is.na(age),
         !is.na(ethnicity))

# test specific data sets
# where have answered question

lit_dat <- model_dat |>
  filter(lit_thresholdL2 %in% c("above", "below")) |>
  mutate(lit_thresholdL2 = as.factor(lit_thresholdL2),
         lit_thresholdL2_bin = as.integer(lit_thresholdL2) - 1L) |>
  select(-lit_weightsL1, -num_weightsEL3, -ict_weightsEL3, -num_thresholdEL3, -num_thresholdL1, -ict_th

num_dat <- model_dat |>
  filter(num_thresholdL1 %in% c("above", "below")) |>
  mutate(num_thresholdL1 = as.factor(num_thresholdL1),
         num_thresholdL1_bin = as.integer(num_thresholdL1) - 1L) |>
  select(-lit_weightsL1, -num_weightsEL3, -ict_weightsEL3, -num_thresholdEL3, -lit_thresholdL2, -ict_th

ict_dat <- model_dat |>
  filter(ict_thresholdEL3 %in% c("above", "below")) |>
  mutate(ict_thresholdEL3 = as.factor(ict_thresholdEL3),
```

```
        ict_thresholdEL3_bin = as.integer(ict_thresholdEL3) - 1L) |>
  select(-lit_weightsL1, -num_weightsEL3, -ict_weightsEL3, -num_thresholdEL3, -num_thresholdL1, -lit_th

#################
# summary stats

lit_dat$lit_thresholdL2 |> table() |> prop.table()
```

```
##
##     above     below
## 0.5694397 0.4305603
```

```
#######################
# logistic regressions

rhs <- "1 + sex + age + ethnicity + uk_born + english_lang + qualification + workingstatus + job_status

# unweighted
lit_glm <- glm(glue("lit_thresholdL2_bin ~ {rhs}"), data = lit_dat, family = binomial(), weights = weig
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
# lit_glm
suppressWarnings({
  tbl_regression(lit_glm, exponentiate = TRUE)
})
```

Characteristic	OR[1]	95% CI[1]	p-value
sex			
Female	—	—	
Male	1.21	1.07, 1.36	0.002
age			
16-44	—	—	
>=45	1.51	1.33, 1.72	<0.001
ethnicity			
White	—	—	
BME	1.41	1.15, 1.73	0.001
uk_born			
Yes	—	—	
No	1.08	0.84, 1.38	0.5
english_lang			
Yes	—	—	
No	2.34	1.75, 3.12	<0.001
qualification			
>=level 2	—	—	
<=Level 1	2.80	2.44, 3.20	<0.001
workingstatus			

Yes	—	—	
No	0.82	0.69, 0.98	0.027
job_status			
intermediate	—	—	
lower	1.75	1.43, 2.15	<0.001
higher	0.86	0.70, 1.07	0.2
gross_income			
>=10000	—	—	
<10000	1.12	0.93, 1.34	0.2
other	1.55	1.31, 1.82	<0.001
own_home			
Yes	—	—	
No	1.43	1.25, 1.62	<0.001
imd			
1	—	—	
2	1.27	1.08, 1.49	0.005
3	1.48	1.22, 1.79	<0.001
4	1.78	1.44, 2.20	<0.001
5	2.23	1.75, 2.85	<0.001
6	2.68	1.90, 3.81	<0.001
7	1.75	1.15, 2.67	0.009
8	1.52	0.78, 3.01	0.2
9	0.74	0.09, 6.43	0.8

[1]OR = Odds Ratio, CI = Confidence Interval

```
# see Table 3 in Rowlands (2015)

num_glm <- glm(glue("num_thresholdL1_bin ~ {rhs}"), data = num_dat, family = binomial(), weights = weigh
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
# num_glm
suppressWarnings({
  tbl_regression(num_glm, exponentiate = TRUE)
})
```

Characteristic	**OR**[1]	**95% CI**[1]	**p-value**
sex			
Female	—	—	
Male	0.52	0.45, 0.60	<0.001
age			
16-44	—	—	
>=45	1.20	1.03, 1.41	0.018
ethnicity			

White	—	—	
BME	2.34	1.80, 3.07	<0.001
uk_born			
Yes	—	—	
No	1.02	0.75, 1.38	>0.9
english_lang			
Yes	—	—	
No	0.78	0.55, 1.11	0.2
qualification			
>=level 2	—	—	
<=Level 1	2.96	2.48, 3.55	<0.001
workingstatus			
Yes	—	—	
No	0.99	0.79, 1.23	>0.9
job_status			
intermediate	—	—	
lower	1.71	1.34, 2.17	<0.001
higher	0.76	0.59, 0.97	0.026
gross_income			
>=10000	—	—	
<10000	1.24	0.99, 1.54	0.058
other	1.61	1.32, 1.97	<0.001
own_home			
Yes	—	—	
No	1.65	1.41, 1.93	<0.001
imd			
1	—	—	
2	1.09	0.90, 1.32	0.4
3	1.61	1.29, 2.02	<0.001
4	1.96	1.51, 2.54	<0.001
5	2.24	1.66, 3.04	<0.001
6	1.84	1.21, 2.87	0.005
7	2.84	1.59, 5.41	<0.001
8	2.44	1.01, 7.05	0.067
9	129,020	0.00,	>0.9

[1]OR = Odds Ratio, CI = Confidence Interval

```
ict_glm <- glm(glue("ict_thresholdEL3_bin ~ {rhs}"), data = ict_dat, family = binomial(), weights = wei
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
# ict_glm
suppressWarnings({
  tbl_regression(num_glm, exponentiate = TRUE)
})
```

Characteristic	OR[1]	95% CI[1]	p-value
sex			
Female	—	—	
Male	0.52	0.45, 0.60	<0.001
age			
16-44	—	—	
>=45	1.20	1.03, 1.41	0.018
ethnicity			
White	—	—	
BME	2.34	1.80, 3.07	<0.001
uk_born			
Yes	—	—	
No	1.02	0.75, 1.38	>0.9
english_lang			
Yes	—	—	
No	0.78	0.55, 1.11	0.2
qualification			
>=level 2	—	—	
<=Level 1	2.96	2.48, 3.55	<0.001
workingstatus			
Yes	—	—	
No	0.99	0.79, 1.23	>0.9
job_status			
intermediate	—	—	
lower	1.71	1.34, 2.17	<0.001
higher	0.76	0.59, 0.97	0.026
gross_income			
>=10000	—	—	
<10000	1.24	0.99, 1.54	0.058
other	1.61	1.32, 1.97	<0.001
own_home			
Yes	—	—	
No	1.65	1.41, 1.93	<0.001
imd			
1	—	—	
2	1.09	0.90, 1.32	0.4
3	1.61	1.29, 2.02	<0.001
4	1.96	1.51, 2.54	<0.001
5	2.24	1.66, 3.04	<0.001
6	1.84	1.21, 2.87	0.005
7	2.84	1.59, 5.41	<0.001
8	2.44	1.01, 7.05	0.067
9	129,020	0.00,	>0.9

[1] OR = Odds Ratio, CI = Confidence Interval

```r
# partial pooling?


######################
# post-stratification

# CRAN package, including newer methods from ML:
#   https://cran.r-project.org/web/packages/autoMrP/vignettes/autoMrP_vignette.pdf
#
# This is the Stan vignette:
#   https://mc-stan.org/rstanarm/articles/mrp.html
#
# Some interesting extensions:
#   https://bookdown.org/jl5522/MRP-case-studies/
#
# Stacked Regression and Poststratification (SRP)?
# (Breiman, 1996)


## prediction

unique_workingstatus <- unique(lit_dat$workingstatus)
unique_gross_income <- unique(lit_dat$gross_income)
unique_uk_born <- unique(lit_dat$uk_born)
unique_sex <- unique(lit_dat$sex)
unique_own_home <- unique(lit_dat$own_home)
unique_age <- unique(lit_dat$age)
unique_english_lang <- unique(lit_dat$english_lang)
unique_ethnicity <- unique(lit_dat$ethnicity)
unique_qualification <- unique(lit_dat$qualification)
unique_imd <- unique(lit_dat$imd)
unique_job_status <- unique(lit_dat$job_status)

# generate all combinations
combs_df <- expand.grid(
  workingstatus = unique_workingstatus,
  gross_income = unique_gross_income,
  uk_born = unique_uk_born,
  sex = unique_sex,
  own_home = unique_own_home,
  age = unique_age,
  english_lang = unique_english_lang,
  ethnicity = unique_ethnicity,
  qualification = unique_qualification,
  imd = unique_imd,
  job_status = unique_job_status
) |> as_tibble()

combs_df$predicted_prob <- predict(lit_glm, combs_df, type = 'response')
```

```r
###################################
# join with target population data

# IMD is at LSOA level
ward_lookup <- read.csv(here::here("raw_data/Ward - neighbourhood - quadrant 2024.csv"))

imd_dat <- read.csv(here::here("raw_data/localincomedeprivationdata_Newham.csv")) |>
  rename(LSOA11CD = "LSOA.code..2011.",
         imd = "Index.of.Multiple.Deprivation..IMD..Decile..where.1.is.most.deprived.10..of.LSOAs.",
         pop = "Total.population..mid.2015..excluding.prisoners.") |>
  select(LSOA11CD, imd, pop)

LSOA_lookup <-
  read.csv(here::here("raw_data/Lower_Layer_Super_Output_Area_(2011)_to_Ward_(2015)_Lookup_in_England_a
  filter(LAD15NM == "Newham")

imd_lookup <-
  LSOA_lookup |>
  merge(ward_lookup, by.x = "WD15NM", by.y = "Ward") |>
  merge(imd_dat) |>
  group_by(imd) |>
  summarize(pop = sum(pop)) |>
  mutate(p_imd = pop / sum(pop)) |>
  select(-pop)

# from resident survey report summary tables
# unless otherwise indicated

total_dat <-
  combs_df |>
  merge(
    tribble(~age, ~p_age,
            "16-44", 0.15 + 0.27 + 0.22,
            ">=45", 0.15 + 0.11 + 0.1)) |>
  merge(
    tribble(~sex, ~p_sex,
            "Male", 0.54,
            "Female", 0.46)) |>
  merge(
    tribble(~ethnicity, ~p_ethn,
          "White", 0.30,
          "BME", 0.70)) |>
  merge(
    tribble(~workingstatus, ~p_workstatus,
        "Yes", 0.65,
        "No", 0.35)) |>
  merge(
    tribble(~own_home, ~p_own_home,
            "Yes", 0.35,
            "No", 0.65)) |>
  # ONS census 2021 Highest level of qualification
  merge(
    tribble(~qualification, ~p_qual,
```

```
          ">=level 2", 0.57,
          "<=Level 1", 0.43)) |>
  # Q61: household gross income before tax
  # Q70: Are you the main or joint householder? e.g.responsible for bills such as rent, mortgage and ut
  # this would be good but its mostly 'not answered'!
  # Q54 What is your average monthly pay?
  #
  ##TODO: break down by LSOA and map to CNA
  ##  read from Newham tab in saiefy1920finalqaddownload280923.xlsx
  merge(
    tribble(~gross_income, ~p_income,
          ">=10000", 0.9,
          "<10000", 0.1)) |>
  # census 2021 usual resident population
  merge(
    tribble(~uk_born, ~p_uk,
          "Yes", 0.455 + 0.001 + 0.004 + 0.003,
          "No", 0.553)) |>
  # Q77 How well can you speak English?
  # 1   Very well
  # 2   Well
  # 3   Not well
  #
  # ONS census 2021 English as main language
  merge(
    tribble(~english_lang, ~p_english,
          "Yes", 0.6537,
          "No", 0.3463)) |>
  # AB: higher and intermediate managerial, administrative and professional occupations
  # C1: supervisory, clerical and junior managerial, administrative and professional occupations
  # C2: skilled manual occupations
  # DE: semi-skilled and unskilled manual and lowest grade occupations
  #
  # tribble(~job_status_ASG, ~job_status, ~prop,
  #         "AB", "higher", 0.167,
  #         "C1", "intermediate", 0.276,
  #         "C2", "lower", 0.234,
  #         "DE", "lower", 0.323)
  merge(
    tribble(~job_status, ~p_job,
          "higher", 0.167,
          "intermediate", 0.276,
          "lower", 0.234 + 0.323)) |>
  merge(imd_lookup) |>
  # calculate product of probabilities, assuming independence
  rowwise() |>
  mutate(product_p = prod(c_across(starts_with("p_")))) |>
  ungroup()

write.csv(total_dat, here::here("data/total_dat.csv"))

##TODO:
# merge(
```

```
#   tribble(~area_cna, ~p_cna,
#           "beckton", 0.05,
#           "custom_house_and_canning_town", 0.14,
#           "east_ham", 0.1,
#           "forest_gate", 0.12,
#           "green_street", 0.13,
#           "manor_park", 0.12,
#           "plaistow", 0.14,
#           "royal_docks", 0.07,
#           "stratford_and_west_ham", 0.13))


# stratification

poststratified_estimates <-
  total_dat |>
  # group_by(area) |>
  summarize(estimate = weighted.mean(predicted_prob, product_p))

poststratified_estimates
```

```
## # A tibble: 1 x 1
##   estimate
##      <dbl>
## 1    0.557
```

```
# compare estimates
```