Survey

12/3/2020

First, we can load the data and tidy it:

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
load("gss/gss_orig.rda")
gss_survey <- gss_orig %>%
  filter(!stringr::str_detect(sample, "blk oversamp")) %>% # this is for weighting
  select(vpsu, vstrat, year, age, sex, college = degree, partyid, hompop, hours = hrs1, income, class, 
  mutate_if(is.factor, ~ fct_collapse(., NULL = c("IAP", "NA", "iap", "na"))) %>%
  mutate(
    age = age %>%
      fct_recode("89" = "89 or older",
                 NULL = "DK") %>%
      as.character() %>%
      as.numeric(),
    hompop = hompop %>%
      fct_collapse(NULL = c("DK")) %>%
      as.character() %>%
      as.numeric(),
    hours = hours %>%
      fct_recode("89" = "89+ hrs",
                 NULL = "DK") %>%
      as.character() %>%
      as.numeric(),
    weight = weight %>%
      as.character() %>%
      as.numeric(),
    partyid = fct_collapse(
      partyid,
      dem = c("strong democrat", "not str democrat"),
      rep = c("strong republican", "not str republican"),
      ind = c("ind,near dem", "independent", "ind,near rep"),
```

```
      other = "other party"
    ),
    income = factor(income, ordered = TRUE),
    college = fct_collapse(
      college,
      degree = c("junior college", "bachelor", "graduate"),
      "no degree"  = c("lt high school", "high school"),
      NULL = "dk"
    )
  ) %>%
  filter(year >= 2000) %>%
  filter(partyid %in% c("dem", "rep")) %>%
  drop_na()
gss_survey$partyid <- factor(gss_survey$partyid)
```

Now, we construct a complex sample survey design.

```
library(survey)
```

```
## Loading required package: grid

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loading required package: survival

##
## Attaching package: 'survey'

## The following object is masked from 'package:graphics':
##
##     dotchart
```

```
gss_design <-
    svydesign(
        ~ vpsu ,
        strata = ~ vstrat ,
        data = gss_survey ,
        weights = ~ weight ,
        nest = TRUE
    )
```

Now, we fit the logistic-regression model with weights using the svyglm() function from the survey package. A slight wrinkle is that we must use the quasibinomial rather than the binomial family to avoid a warning about noninteger counts produced by the use of differential sampling weights.

```r
 options(survey.lonely.psu="certainty")
glm_result <-
    svyglm(
        partyid ~ age + sex + college + hompop + hours +
          income + class + finrela + wrkgovt + marital +
        educ + race + incom16 + weight, design=gss_design, family=quasibinomial)

summary(glm_result)
```

```
##
## Call:
## svyglm(formula = partyid ~ age + sex + college + hompop + hours +
##     income + class + finrela + wrkgovt + marital + educ + race +
##     incom16 + weight, design = gss_design, family = quasibinomial)
##
## Survey design:
## svydesign(~vpsu, strata = ~vstrat, data = gss_survey, weights = ~weight,
##     nest = TRUE)
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.035e+00  1.742e+00   0.594  0.55262
## age                      -2.101e-03  2.925e-03  -0.718  0.47279
## sexfemale                -3.702e-01  6.605e-02  -5.606 3.18e-08 ***
## collegedegree             9.463e-02  1.297e-01   0.729  0.46602
## hompop                   -1.219e-02  2.813e-02  -0.433  0.66497
## hours                     8.473e-04  2.443e-03   0.347  0.72883
## income.L                 -6.907e-01  4.547e-01  -1.519  0.12931
## income.Q                  8.539e-01  4.921e-01   1.735  0.08323 .
## income.C                 -1.856e-01  4.465e-01  -0.416  0.67774
## income^4                 -2.612e-01  4.612e-01  -0.566  0.57130
## income^5                 -3.621e-01  4.636e-01  -0.781  0.43502
## income^6                  4.748e-01  4.919e-01   0.965  0.33482
## income^7                 -5.626e-01  4.733e-01  -1.189  0.23507
## income^8                 -1.759e-02  4.786e-01  -0.037  0.97069
## income^9                 -5.981e-01  4.895e-01  -1.222  0.22223
## income^10                 9.038e-01  4.862e-01   1.859  0.06354 .
## income^11                 1.018e+00  5.725e-01   1.779  0.07581 .
## classworking class       -1.799e-01  2.060e-01  -0.873  0.38299
## classmiddle class         1.712e-01  2.079e-01   0.824  0.41037
## classupper class         -1.891e-01  2.883e-01  -0.656  0.51211
## classDK                   1.480e-01  6.643e-01   0.223  0.82373
## finrelabelow average     -7.540e-02  2.005e-01  -0.376  0.70698
## finrelaaverage           -6.255e-02  2.059e-01  -0.304  0.76141
## finrelaabove average      1.707e-01  2.103e-01   0.812  0.41732
## finrelafar above average  4.044e-01  3.009e-01   1.344  0.17938
## finrelaDK                -1.177e-01  6.076e-01  -0.194  0.84649
## wrkgovtprivate           -3.669e-02  8.547e-02  -0.429  0.66790
## wrkgovtDK                -2.877e-01  3.322e-01  -0.866  0.38681
## maritalwidowed           -2.673e-01  1.898e-01  -1.408  0.15954
## maritaldivorced          -2.852e-01  1.036e-01  -2.752  0.00611 **
## maritalseparated         -5.502e-01  2.445e-01  -2.250  0.02480 *
## maritalnever married     -7.328e-01  9.311e-02  -7.870 1.68e-14 ***
```

```
## educ1                      -2.338e+00  2.102e+00  -1.112  0.26662
## educ2                      -7.691e-01  1.791e+00  -0.429  0.66782
## educ3                      -1.278e+00  1.859e+00  -0.687  0.49205
## educ4                      -1.241e+01  1.707e+00  -7.274 1.11e-12 ***
## educ5                      -2.814e-01  1.756e+00  -0.160  0.87270
## educ6                      -1.664e+00  1.768e+00  -0.941  0.34686
## educ7                      -9.246e-01  1.928e+00  -0.480  0.63168
## educ8                      -5.689e-01  1.732e+00  -0.329  0.74262
## educ9                      -1.143e+00  1.746e+00  -0.655  0.51298
## educ10                     -6.100e-01  1.714e+00  -0.356  0.72199
## educ11                     -4.604e-01  1.710e+00  -0.269  0.78781
## educ12                     -5.378e-01  1.700e+00  -0.316  0.75183
## educ13                     -4.291e-01  1.701e+00  -0.252  0.80096
## educ14                     -6.224e-01  1.699e+00  -0.366  0.71429
## educ15                     -7.389e-01  1.702e+00  -0.434  0.66431
## educ16                     -8.392e-01  1.702e+00  -0.493  0.62220
## educ17                     -1.158e+00  1.707e+00  -0.678  0.49784
## educ18                     -1.222e+00  1.700e+00  -0.719  0.47253
## educ19                     -1.518e+00  1.737e+00  -0.874  0.38235
## educ20                     -1.430e+00  1.711e+00  -0.836  0.40364
## raceblack                  -2.787e+00  1.543e-01 -18.065  < 2e-16 ***
## raceother                  -1.158e+00  1.197e-01  -9.680  < 2e-16 ***
## incom16below average        2.964e-01  1.504e-01   1.971  0.04921 *
## incom16average              3.502e-01  1.382e-01   2.534  0.01155 *
## incom16above average        2.764e-01  1.508e-01   1.833  0.06732 .
## incom16far above average    2.627e-01  2.239e-01   1.173  0.24116
## weight                      1.989e-01  6.331e-02   3.142  0.00176 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 1.024276)
##
## Number of Fisher Scoring iterations: 10
```

Note: The global option, options(survey.lonely.psu="fail"), makes it an error to have a stratum with a single, non-certainty PSU. Changing it to options(survey.lonely.psu="certainty"), single-PSU stratum makes no contribution to the variance (for multistage sampling it makes no contribution at that level of sampling). This is an alternative to specifying fpc, and is useful to run the regression without error.

```
probs_survey<-predict(glm_result, gss_survey, type = "response")
preds_survey<-ifelse(probs_survey >=.5, 1, 0)
conf_log_survey <- table(preds_survey, gss_survey$partyid)
conf_log_survey
```

```
##
## preds_survey  dem  rep
##            0 2266  850
##            1 1050 1634
```

```
n <- length(gss_survey$partyid)
false_pos_survey <- conf_log_survey[1,2]
false_neg_survey <- conf_log_survey[2,1]
error_survey <- 1/n *(false_pos_survey + false_neg_survey)
error_survey
```

```
## [1] 0.3275862
```

```
1 - error_survey
```

```
## [1] 0.6724138
```

We see that the training error rate is 0.3275862 for the logistic regression with weights.

training and testing

However, to compare it to the tidymodels approach, we must also perform the same analysis with a training and testing set. We do so with the same initial split used in the tidymodels approach: Now, we construct a complex sample survey design.

```
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 0.1.2 --
```

```
## v broom     0.7.2      v recipes   0.1.15
## v dials     0.0.9      v rsample   0.0.8
## v infer     0.5.3      v tune      0.1.2
## v modeldata 0.1.0      v workflows 0.2.1
## v parsnip   0.1.4      v yardstick 0.0.7
```

```
## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x Matrix::expand()  masks tidyr::expand()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x Matrix::pack()    masks tidyr::pack()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## x Matrix::unpack()  masks tidyr::unpack()
## x recipes::update() masks Matrix::update(), stats::update()
```

```
set.seed(1)
split <- initial_split(data = gss_survey, prop = 3/4)
gss_train <- training(split)
gss_test <- testing(split)

gss_design_train <-
    svydesign(
        ~ vpsu ,
        strata = ~ vstrat ,
        data = gss_train ,
        weights = ~ weight ,
        nest = TRUE
    )
```

Now, we fit the logistic-regression model with weights using the svyglm() function from the survey package. A slight wrinkle is that we must use the quasibinomial rather than the binomial family to avoid a warning about noninteger counts produced by the use of differential sampling weights.

```
 options(survey.lonely.psu="certainty")
glm_result_train <-
    svyglm(
        partyid ~ age + sex + college + hompop + hours +
          income + class + finrela + wrkgovt + marital +
        educ + race + incom16 + weight, design=gss_design, family=quasibinomial)

summary(glm_result_train)
```

```
##
## Call:
## svyglm(formula = partyid ~ age + sex + college + hompop + hours +
##     income + class + finrela + wrkgovt + marital + educ + race +
##     incom16 + weight, design = gss_design, family = quasibinomial)
##
## Survey design:
## svydesign(~vpsu, strata = ~vstrat, data = gss_survey, weights = ~weight,
##     nest = TRUE)
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.035e+00  1.742e+00   0.594  0.55262
## age                      -2.101e-03  2.925e-03  -0.718  0.47279
## sexfemale                -3.702e-01  6.605e-02  -5.606 3.18e-08 ***
## collegedegree             9.463e-02  1.297e-01   0.729  0.46602
## hompop                   -1.219e-02  2.813e-02  -0.433  0.66497
## hours                     8.473e-04  2.443e-03   0.347  0.72883
## income.L                 -6.907e-01  4.547e-01  -1.519  0.12931
## income.Q                  8.539e-01  4.921e-01   1.735  0.08323 .
## income.C                 -1.856e-01  4.465e-01  -0.416  0.67774
## income^4                 -2.612e-01  4.612e-01  -0.566  0.57130
## income^5                 -3.621e-01  4.636e-01  -0.781  0.43502
## income^6                  4.748e-01  4.919e-01   0.965  0.33482
## income^7                 -5.626e-01  4.733e-01  -1.189  0.23507
## income^8                 -1.759e-02  4.786e-01  -0.037  0.97069
## income^9                 -5.981e-01  4.895e-01  -1.222  0.22223
## income^10                 9.038e-01  4.862e-01   1.859  0.06354 .
## income^11                 1.018e+00  5.725e-01   1.779  0.07581 .
## classworking class       -1.799e-01  2.060e-01  -0.873  0.38299
## classmiddle class         1.712e-01  2.079e-01   0.824  0.41037
## classupper class         -1.891e-01  2.883e-01  -0.656  0.51211
## classDK                   1.480e-01  6.643e-01   0.223  0.82373
## finrelabelow average     -7.540e-02  2.005e-01  -0.376  0.70698
## finrelaaverage           -6.255e-02  2.059e-01  -0.304  0.76141
## finrelaabove average      1.707e-01  2.103e-01   0.812  0.41732
## finrelafar above average  4.044e-01  3.009e-01   1.344  0.17938
## finrelaDK                -1.177e-01  6.076e-01  -0.194  0.84649
## wrkgovtprivate           -3.669e-02  8.547e-02  -0.429  0.66790
## wrkgovtDK                -2.877e-01  3.322e-01  -0.866  0.38681
## maritalwidowed           -2.673e-01  1.898e-01  -1.408  0.15954
```

```
## maritaldivorced          -2.852e-01  1.036e-01  -2.752  0.00611 **
## maritalseparated         -5.502e-01  2.445e-01  -2.250  0.02480 *
## maritalnever married     -7.328e-01  9.311e-02  -7.870 1.68e-14 ***
## educ1                    -2.338e+00  2.102e+00  -1.112  0.26662
## educ2                    -7.691e-01  1.791e+00  -0.429  0.66782
## educ3                    -1.278e+00  1.859e+00  -0.687  0.49205
## educ4                    -1.241e+01  1.707e+00  -7.274 1.11e-12 ***
## educ5                    -2.814e-01  1.756e+00  -0.160  0.87270
## educ6                    -1.664e+00  1.768e+00  -0.941  0.34686
## educ7                    -9.246e-01  1.928e+00  -0.480  0.63168
## educ8                    -5.689e-01  1.732e+00  -0.329  0.74262
## educ9                    -1.143e+00  1.746e+00  -0.655  0.51298
## educ10                   -6.100e-01  1.714e+00  -0.356  0.72199
## educ11                   -4.604e-01  1.710e+00  -0.269  0.78781
## educ12                   -5.378e-01  1.700e+00  -0.316  0.75183
## educ13                   -4.291e-01  1.701e+00  -0.252  0.80096
## educ14                   -6.224e-01  1.699e+00  -0.366  0.71429
## educ15                   -7.389e-01  1.702e+00  -0.434  0.66431
## educ16                   -8.392e-01  1.702e+00  -0.493  0.62220
## educ17                   -1.158e+00  1.707e+00  -0.678  0.49784
## educ18                   -1.222e+00  1.700e+00  -0.719  0.47253
## educ19                   -1.518e+00  1.737e+00  -0.874  0.38235
## educ20                   -1.430e+00  1.711e+00  -0.836  0.40364
## raceblack                -2.787e+00  1.543e-01 -18.065  < 2e-16 ***
## raceother                -1.158e+00  1.197e-01  -9.680  < 2e-16 ***
## incom16below average      2.964e-01  1.504e-01   1.971  0.04921 *
## incom16average            3.502e-01  1.382e-01   2.534  0.01155 *
## incom16above average      2.764e-01  1.508e-01   1.833  0.06732 .
## incom16far above average  2.627e-01  2.239e-01   1.173  0.24116
## weight                    1.989e-01  6.331e-02   3.142  0.00176 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 1.024276)
##
## Number of Fisher Scoring iterations: 10
```

Note: The global option, options(survey.lonely.psu="fail"), makes it an error to have a stratum with a single, non-certainty PSU. Changing it to options(survey.lonely.psu="certainty"), single-PSU stratum makes no contribution to the variance (for multistage sampling it makes no contribution at that level of sampling). This is an alternative to specifying fpc, and is useful to run the regression without error.

```r
probs_survey_test <- predict(glm_result_train, gss_test, type = "response")
preds_survey_test <- ifelse(probs_survey_test >=.5, 1, 0)
conf_log_survey_test <- table(preds_survey_test, gss_test$partyid)
conf_log_survey_test
```

```
##
## preds_survey_test dem rep
##                 0 527 217
##                 1 300 406
```

```r
n_test <- length(gss_test$partyid)
false_pos_survey_test <- conf_log_survey_test[1,2]
false_neg_survey_test <- conf_log_survey_test[2,1]
error_survey_test <- 1/n_test *(false_pos_survey_test + false_neg_survey_test)
error_survey_test
```

```
## [1] 0.3565517
```

```r
1 - error_survey_test
```

```
## [1] 0.6434483
```

We see that the testing error rate is 0.3565517 for the logistic regression with weights and the amount correctly predicted is 0.6434483.