Analyzing HINTS 6 data using R

Jacob Rohde

2024

Loading required libraries and data into R

The code chunk below loads the required R packages and data to the working directory. For this example, I load the HINTS 6 SAS data set using the "*haven*" package.

```
library(haven) # For loading data from SAS, SPSS, or STATA into R
library(dplyr) # For data manipulation
library(survey) # For analyzing complex survey data
library(srvyr) # For manipulating survey objects with dplyr
library(broom) # For presenting tidy data tables
library(rstudioapi) # For setting a working directory

# Setting the working directory to file location
setwd(dirname(getActiveDocumentContext()$path))

# Load data
df = haven::read_sas("hints6_public.sas7bdat")
```

Recoding survey variables

The code chunk below shows an example of how to use 'dplyr' to create new variables or recode existing ones.

```
df = df |>
  dplyr::mutate(gender = case_match(factor(BirthGender),
                                    '1' ~ 'Male',
                                    '2' ~ 'Female')) |>

  dplyr::mutate(edu = case_match(factor(Education),
                                 c('1', '2') ~ 'Less than high school',
                                 '3' ~ '12 years or completed high school',
                                 c('4', '5') ~ 'Some college',
                                 c('6', '7') ~ 'College graduate or higher')) |>

  dplyr::mutate(SeekCancerInfo = case_match(SeekCancerInfo,
                                            1 ~ 1,
                                            2 ~ 0))

# Setting the reference level for categorical variables
df$gender = relevel(factor(df$gender, ordered = F),
                    ref = 'Male')
```

```
df$edu = relevel(factor(df$edu, ordered = F),
                 ref = 'Less than high school')
```

Example analytic procedures using a replicate weights approach

The code chunk below creates a survey design object to account for replicate weights when running statistical analyses.

```
svy_obj_rep = as_survey_rep(.data = df,
                            weights = PERSON_FINWT0,
                            repweights = num_range(prefix = "PERSON_FINWT",
                                                   range = 1:50),
                            type = "JKn",
                            scale = 0.98,
                            rscales = rep(1, times = 50))
```

Computing a crosstab and chi-square test:

```
# Crosstab
svy_obj_rep |>
  dplyr::filter(is.na(edu) == F,
                is.na(gender) == F) |>
  dplyr::group_by(edu, gender) |>
  dplyr::summarize(n = n(),
                   total = survey_total(),
                   pct = survey_prop())

## When `proportion` is unspecified, `survey_prop()` now defaults to `proportion = TR
UE`.
## ℹ This should improve confidence interval coverage.
## This message is displayed once per session.

## # A tibble: 8 × 7
## # Groups:   edu [4]
##   edu                                gender     n    total total_se   pct  pct_se
##   <fct>                              <fct>  <int>    <dbl>    <dbl> <dbl>   <dbl>
## 1 Less than high school              Male     155   9.67e6 1466802. 0.596 0.0438
## 2 Less than high school              Female   228   6.57e6  622501. 0.404 0.0438
## 3 12 years or completed high school  Male     375   2.61e7 1805283. 0.507 0.0186
## 4 12 years or completed high school  Female   686   2.54e7 1122464. 0.493 0.0186
## 5 College graduate or higher         Male    1127   3.71e7  300649. 0.476 0.00321
## 6 College graduate or higher         Female  1582   4.08e7  383616. 0.524 0.00321
## 7 Some college                       Male     642   4.39e7 1237523. 0.472 0.00761
## 8 Some college                       Female  1023   4.92e7  821645. 0.528 0.00761

# Chi-square test
svy_obj_rep |>
  svychisq(formula = ~ gender + edu,
           statistic = "F")

##
##  Pearson's X^2: Rao & Scott adjustment
##
```

```
## data:  NextMethod()
## F = 4.6411, ndf = 1.7392, ddf = 85.2215, p-value = 0.01586
```

Computing a logistic regression:

```
logistic_model = svy_obj_rep |>
  svyglm(formula = SeekCancerInfo ~ edu + gender,
         family = quasibinomial())

# For displaying general summary statistics
summary(logistic_model)

##
## Call:
## svyglm(svy_obj_rep, formula = SeekCancerInfo ~ edu + gender,
##     family = quasibinomial())
##
## Survey design:
## Called via srvyr
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        -1.54231    0.22823  -6.758 2.33e-08 ***
## edu12 years or completed high school  0.20986    0.25088   0.836  0.40730
## eduCollege graduate or higher         1.51143    0.22256   6.791 2.08e-08 ***
## eduSome college                       0.94539    0.24552   3.851  0.00037 ***
## genderFemale                          0.71424    0.08912   8.014 3.29e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 5778.992)
##
## Number of Fisher Scoring iterations: 4

# For displaying odds ratios and 95% confidence intervals
tidy(logistic_model,
     conf.int = T,
     conf.level = 0.95,
     exponentiate = T)

## # A tibble: 5 × 7
##   term                  estimate std.error statistic  p.value conf.low conf.high
##   <chr>                    <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)              0.214    0.228     -6.76  2.33e- 8    0.135     0.339
## 2 edu12 years or compl…    1.23     0.251      0.836 4.07e- 1    0.744     2.04
## 3 eduCollege graduate …    4.53     0.223      6.79  2.08e- 8    2.90      7.10
## 4 eduSome college          2.57     0.246      3.85  3.70e- 4    1.57      4.22
## 5 genderFemale             2.04     0.0891     8.01  3.29e-10    1.71      2.44
```

Computing a linear regression:

```
linear_model = svy_obj_rep |>
  svyglm(formula = GeneralHealth ~ edu + gender,
         family = gaussian())

summary(linear_model)
```

```
##
## Call:
## svyglm(svy_obj_rep, formula = GeneralHealth ~ edu + gender, family = gaussian())
##
## Survey design:
## Called via srvyr
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                           2.96358    0.11274  26.287  < 2e-16 ***
## edu12 years or completed high school -0.27363    0.12832  -2.132   0.0385 *
## eduCollege graduate or higher        -0.67084    0.12127  -5.532 1.54e-06 ***
## eduSome college                      -0.34157    0.12737  -2.682   0.0102 *
## genderFemale                          0.05092    0.03589   1.418   0.1629
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 7112.641)
##
## Number of Fisher Scoring iterations: 2
```

Example analytic procedures using a Taylor Series linearization approach

The code chunk below creates a survey design object to account for Taylor Series linearization sample weights when running statistical analyses.

```
svy_obj_linear = as_survey_design(.data = df,
                                  ids = VAR_CLUSTER,
                                  strata = VAR_STRATUM,
                                  weights = PERSON_FINWT0,
                                  nest = T)
```

Computing a crosstab and chi-square test:

```
# Crosstab
svy_obj_linear |>
  dplyr::filter(is.na(edu) == F,
                is.na(gender) == F) |>
  dplyr::group_by(edu, gender) |>
  dplyr::summarize(n = n(),
                   total = survey_total(),
                   pct = survey_prop())
```

```
## # A tibble: 8 × 7
## # Groups:   edu [4]
##   edu                                gender     n     total total_se   pct pct_se
##   <fct>                              <fct>  <int>     <dbl>    <dbl> <dbl>  <dbl>
## 1 Less than high school              Male     155  9673127. 1416389. 0.596 0.0424
## 2 Less than high school              Female   228  6566822.  651865. 0.404 0.0424
## 3 12 years or completed high school  Male     375 26089157. 2031080. 0.507 0.0246
## 4 12 years or completed high school  Female   686 25412934. 1382808. 0.493 0.0246
## 5 College graduate or higher         Male    1127 37075982. 1630403. 0.476 0.0149
```

```
## 6 College graduate or higher        Female  1582 40801112. 1558491. 0.524 0.0149
## 7 Some college                      Male     642 43928337. 3284813. 0.472 0.0216
## 8 Some college                      Female  1023 49212485. 2385658. 0.528 0.0216
```

```
# Chi-square test
svy_obj_linear |>
  svychisq(formula = ~ gender + edu,
           statistic = "F")
```

```
##
##  Pearson's X^2: Rao & Scott adjustment
##
## data:  NextMethod()
## F = 2.6956, ndf = 2.8714, ddf = 562.7859, p-value = 0.04772
```

Computing a logistic regression:

```
logistic_model = svy_obj_linear |>
  svyglm(formula = SeekCancerInfo ~ edu + gender,
         family = quasibinomial())

# For displaying general summary statistics
summary(logistic_model)
```

```
##
## Call:
## svyglm(formula = SeekCancerInfo ~ edu + gender, design = svy_obj_linear,
##     family = quasibinomial())
##
## Survey design:
## Called via srvyr
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        -1.54231    0.22078  -6.986 4.52e-11 ***
## edu12 years or completed high school  0.20986    0.23763   0.883    0.378
## eduCollege graduate or higher         1.51143    0.21311   7.092 2.47e-11 ***
## eduSome college                       0.94539    0.22781   4.150 5.00e-05 ***
## genderFemale                          0.71424    0.09063   7.881 2.37e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 0.996206)
##
## Number of Fisher Scoring iterations: 4
```

```
# For displaying odds ratios and 95% confidence intervals
tidy(logistic_model,
     conf.int = T,
     conf.level = 0.95,
     exponentiate = T)
```

```
## # A tibble: 5 × 7
##   term                  estimate std.error statistic  p.value conf.low conf.high
##   <chr>                    <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)              0.214    0.221     -6.99  4.52e-11    0.138     0.331
```

```
## 2 edu12 years or compl…    1.23     0.238      0.883 3.78e- 1    0.772     1.97
## 3 eduCollege graduate …    4.53     0.213      7.09  2.47e-11    2.98      6.90
## 4 eduSome college          2.57     0.228      4.15  5.00e- 5    1.64      4.03
## 5 genderFemale             2.04     0.0906     7.88  2.37e-13    1.71      2.44
```

Computing a linear regression:

```
linear_model = svy_obj_linear |>
  svyglm(formula = GeneralHealth ~ edu + gender,
         family = gaussian())

summary(linear_model)

##
## Call:
## svyglm(formula = GeneralHealth ~ edu + gender, design = svy_obj_linear,
##     family = gaussian())
##
## Survey design:
## Called via srvyr
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                           2.96358    0.11884  24.937  < 2e-16 ***
## edu12 years or completed high school -0.27363    0.13079  -2.092  0.03775 *
## eduCollege graduate or higher        -0.67084    0.12749  -5.262 3.79e-07 ***
## eduSome college                      -0.34157    0.12686  -2.692  0.00772 **
## genderFemale                          0.05092    0.04364   1.167  0.24476
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.222734)
##
## Number of Fisher Scoring iterations: 2
```

Combining HINTS 5 Cycle 4 data with HINTS 6

The code chunk below loads HINTS 6 and HINTS 5 Cycle 4 SAS files into R as separate data objects (make sure both files are in the same working directory).

```
# HINTS 6 file
df_H6 = haven::read_sas("hints6_public.sas7bdat")

# HINTS 5 Cycle 4 file
df_H5C4 = haven::read_sas("hints5_cycle4_public.sas7bdat")
```

Create new sample weights and merge the two data sets:

```
# Create variable names
nwgt_var_names = c(paste0('nwgt', 1:100))
var_names = c(paste0('PERSON_FINWT', 1:50))

# Create Hints 5 Cycle 4 group weights
df_H5C4 = df_H5C4 |>
```

```
  dplyr::mutate(hints_edition = 'Hints 5 Cycle 4') |>
  dplyr::mutate(nwgt0 = PERSON_FINWT0)

for(i in 1:100){
  if(i <= 50){
    df_H5C4[nwgt_var_names[i]] = df_H5C4[var_names[i]]
  }

  if(i > 50){
    df_H5C4[nwgt_var_names[i]] = df_H5C4$PERSON_FINWT0
  }
}

# Create Hints 6 group weights
df_H6 = df_H6 |>
  dplyr::mutate(hints_edition = 'HINTS 6') |>
  dplyr::mutate(nwgt0 = PERSON_FINWT0)

for(i in 1:100){
  if(i <= 50){
    df_H6[nwgt_var_names[i]] = df_H6$PERSON_FINWT0
  }

  if(i > 50){
    df_H6[nwgt_var_names[i]] = df_H6[var_names[i-50]]
  }
}

# Merge the data sets
df_multi = plyr::rbind.fill(df_H5C4, df_H6)

# Display number of respondents from both survey editions
table(df_multi$hints_edition)

##
## Hints 5 Cycle 4         HINTS 6
##            3865            6252
```

The example code below can be used to run simple frequencies on two common variables ("SeekCancerInfo" and "ChanceAskQuestions") in the HINTS 6 and HINTS 5 Cycle 4 merged data set using a replicate weights approach:

```
# Create the replicate weights survey design object
svy_obj_rep_merged = as_survey_rep(.data = df_multi,
                                   weights = nwgt0,
                                   repweights = num_range(prefix = "nwgt",
                                                          range = 1:100),
                                   type = "JKn",
                                   scale = 0.98,
                                   rscales = rep(1, times = 100))

# Crosstab
svy_obj_rep_merged |>
  dplyr::filter(ChanceAskQuestions > 0,
                SeekCancerInfo > 0) |>
```

```
  dplyr::group_by(ChanceAskQuestions, SeekCancerInfo) |>
  dplyr::summarize(n = n(),
                   total = survey_total(),
                   pct = survey_prop())
```

The example code below can be used to run simple frequencies on two common variables ("SeekCancerInfo" and "ChanceAskQuestions") in the HINTS 6 and HINTS 5 Cycle 4 merged data set using a Taylor Series linearization approach:

```
# Create the Taylor Series linearization survey design object
svy_obj_linear_merged = as_survey_design(.data = df_multi,
                                         ids = VAR_CLUSTER,
                                         strata = VAR_STRATUM,
                                         weights = PERSON_FINWT0,
                                         nest = T)

# Crosstab
svy_obj_linear_merged |>
  dplyr::filter(ChanceAskQuestions > 0,
                SeekCancerInfo > 0) |>
  dplyr::group_by(ChanceAskQuestions, SeekCancerInfo) |>
  dplyr::summarize(n = n(),
                   total = survey_total(),
                   pct = survey_prop())
```