ASDA3 Analysis Examples Replication Chapter 5

R Code

```
# data production prior to analysis examples
# need ncsr part1 part2 and population weight data
# nhanes subset of >=18 years of age
# hrs hh weight and financial respondents
# hrs respondent weight for individual level analyses
# can create subsets at the design data set level

setwd("P:/ASDA3/Replication R/Chapter 5")
getwd()

# data production
# load packages
library (survey)
library(haven)

#nhanes first
# example of how to read in from SAS, not used here
#nhanesdata <- read_sas("P:\\ASDA3\\Replication SAS\\nhanes1112_f.sas7bdat")

# load nhanes1112.rdata
load ("P:/ASDA3/Data Sets for Analysis Examples and Stata R Code/nhanes1112.rdata")
nhanesdata <- nhanes1112
summary(nhanesdata)

#create factor variables
nhanesdata$racec <- factor(nhanesdata$ridreth1, levels = 1: 5 , labels =c("Mexican", "Other Hispanic", "White",
"Black", "Other"))
nhanesdata$marcatc <- factor(nhanesdata$marcat, levels = 1: 3, labels =c("Married", "Previously Married", "Never
Married"))
nhanesdata$edcatc <- factor(nhanesdata$edcat, levels = 1: 4, labels =c("0-11", "12", "13-15","16+"))
nhanesdata$bp_catc <- factor(nhanesdata$bp_cat, levels = 1: 4, labels =c("Normal", "Pre-HBP", "Stage 1
HBP","Stage 2 HBP"))
#nhanesdata$agecsq <- (nhanesdata$age * nhanesdata$age)

names(nhanesdata)

nhanessvy2 <- svydesign(strata=~sdmvstra, id=~sdmvpsu, weights=~wtmec2yr, data=nhanesdata, nest=T)
subnhanes <- subset(nhanessvy2 , age >= 18)
names (nhanessvy2)

##
#ncs-r next

load ("P:/ASDA3/Data Sets for Analysis Examples and Stata R Code/ncsr.rdata")
names(ncsr)
summary(ncsr)

#create factor versions with labels
ncsr$racec <- factor(ncsr$racecat, levels = 1: 4, labels =c("Other", "Hispanic", "Black", "White"))
ncsr$mar3catc <- factor(ncsr$mar3cat, levels = 1: 3, labels =c("Married", "Previously Married", "Never
Married"))
ncsr$ed4catc <- factor(ncsr$ed4cat, levels = 1: 4, labels =c("0-11", "12", "13-15","16+"))
ncsr$sexc <- factor(ncsr$sex, levels = 1:2, labels=c("Male","Female"))
ncsr$ag4catc <- factor(ncsr$ag4cat, levels = 1:4, labels=c("18-29", "30-44", "45-59", "60+"))
ncsr$mdec <- factor(ncsr$mde, level = 1:2, labels=c("No","Yes"))
```

```
# part 1 data n=9282
ncsrsvyp1 <- svydesign(strata=~sestrat, id=~seclustr, weights=~ncsrwtsh, data=ncsr, nest=T)
names (ncsrsvyp1)

# part 2 data n=5692
ncsrp2 <- subset(ncsr, !is.na(ncsrwtlg))
ncsrsvyp2 <- svydesign(strata=~sestrat, id=~seclustr, weights=~ncsrwtlg, data=ncsrp2, nest=T)
names (ncsrsvyp2)

# rescale weight to sum to population in 2001
ncsr$popweight <- (ncsr$ncsrwtsh*(209128094/9282))

ncsrsvypop <- svydesign(strata=~sestrat, id=~seclustr, weights=~popweight, data=ncsr, nest=T)
summary(ncsrsvypop)

##
#hrs 2012 final data from SAS
#both hh and r weights are needed plus financial respondent for hh level analysis

#library(haven)
load ("P:/ASDA3/Data Sets for Analysis Examples and Stata R Code/hrs12.rdata")
names(hrs12)

hrssvyhh <- svydesign(strata=~stratum, id=~secu, weights=~nwgthh , data=hrs12, nest=T)
summary(hrssvyhh)

hrssvysub <-subset(hrssvyhh, nfinr==1)
summary(hrssvysub)

hrssvyr <- svydesign(strata=~stratum, id=~secu, weights=~nwgtr , data=hrs12, nest=T)
summary(hrssvyr)

# Figures 5.1 and 5.2
svyhist(~lbxtc, subset (nhanessvy2, age >=18), main="", col="grey80", xlab ="Histogram of Total Cholesterol")

# create a variable called gender for boxplot
nhanessvy2<-update(nhanessvy2, gender=cut(riagendr, c(1, 2, Inf), right=F))

svyboxplot(lbxtc~gender , subset (nhanessvy2, age >=18), col="grey80", ylab="Total Cholesterol", xlab ="1=Male
2=Female")

# Example 5.3
svytotal (~mde, ncsrsvypop, deff=T)
confint(svytotal(~mde, ncsrsvypop))

# Example 5.3 MDE over marital status
ex53 <- svyby (~mde, ~mar3catc, ncsrsvypop, svytotal)
ex53 <- svyby (~mde, ~mar3catc, ncsrsvypop, svytotal, deff=T)
ex53
confint(ex53)

# Example 5.4 HRS HH Level Wealth/Total Assets
svyby (~h11atota, ~I(nfinr==1), hrssvyhh, na.rm=T, svytotal)
confint(svyby (~h11atota, ~I(nfinr==1), hrssvyhh, na.rm=T, ci=T, svytotal))

# Example 5.5 HRS HH Income
svyby (~h11itot, ~I(nfinr==1), hrssvyhh, na.rm=T, svymean)
```

```
confint(svyby (~h11itot, ~I(nfinr==1), hrssvyhh, na.rm=T, ci=T, svymean))

# Example 5.5 using alternative Bayesian approach
# load all relevant libraries for csSampling

install.packages("devtools") # if you have not installed "devtools" package

# install_github("RyanHornby/csSampling") # lots of packages needed, will take a second, already loaded here

library(csSampling)
library(rstan)
library(brms)
library(survey)
rstan_options(auto_write = TRUE)

# read in the full HRS data set for the third edition
load("P:\\ASDA3\\Data Sets for Analysis Examples and Stata R Code\\hrs12.rdata")

# create complete data set with variables of interest
hrs.red <- hrs12[hrs12$nfinr == 1, c("h11itot", "secu", "stratum", "nwgthh")]
hrs.red <- hrs.red[complete.cases(hrs.red),]

# need to normalize HRS weights to match what is done for Stan modeling
hrs.red$wtsc <- hrs.red$nwgthh / mean(hrs.red$nwgthh)

# survey design object
hrs.des <- svydesign(id = ~secu, strata = ~stratum, weights = ~wtsc, nest = T, data = hrs.red)

# Bayesian approach, flat prior (non-informative), which is default
set.seed(41279)
model_formula <- formula("h11itot|weights(wtsc) ~ 1")
mod.brms <- cs_sampling_brms(svydes = hrs.des, brmsmod = brmsformula(model_formula), data = hrs.red, family =
gaussian())

# Print results
mod.brms$stan_fit

# With Normal Prior, mean = 70000, SD (SE) = 1000
mod.brms2 <- cs_sampling_brms(svydes = hrs.des, brmsmod = brmsformula(model_formula), prior =
c(set_prior("normal(70000,1000)", class = "Intercept")), data = hrs.red, family = gaussian())

# Example 5.6 Mean Systolic Blood Pressure, NHANES data
a <- svymean(~bpxsy1, subset (nhanessvy2, age >=18), na.rm=TRUE)
coef(a)
SE(a)
confint(a)

# Example 5.6 alternative Bayesian approach
# create complete data set with variables of interest
nhanes.red <- nhanes1112[nhanes1112$age >= 18 & !is.na(nhanes1112$age), c("bpxsy1", "sdmvpsu", "sdmvstra",
"wtmec2yr")]
nhanes.red <- nhanes.red[complete.cases(nhanes.red),]

# need to normalize NHANES weights to match what is done for Stan modeling
nhanes.red$wtsc <- nhanes.red$wtmec2yr / mean(nhanes.red$wtmec2yr)

# survey design object
nhanes.des <- svydesign(id = ~sdmvpsu, strata = ~sdmvstra, weights = ~wtsc, nest = T, data = nhanes.red)
```

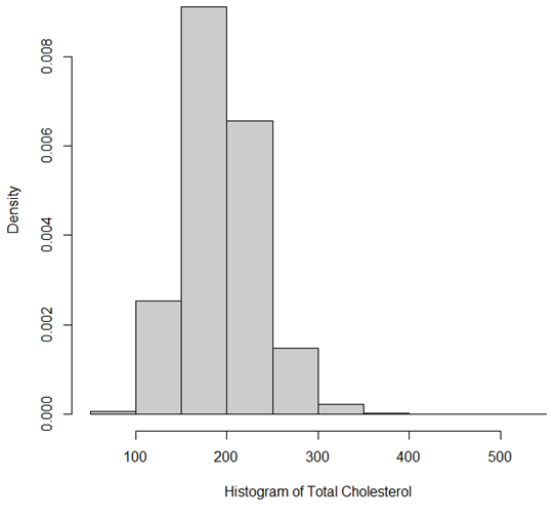
```
# Bayesian approach, flat prior
set.seed(41279)
model_formula <- formula("bpxsy1|weights(wtsc) ~ 1")
mod.brms <- cs_sampling_brms(svydes = nhanes.des, brmsmod = brmsformula(model_formula), data = nhanes.red,
family = gaussian())

# Print results
mod.brms$stan_fit

# Example 5.7
svyby (~h11atota, ~I(nfinr==1), hrssvyhh, na.rm=T, ci=T, svymean)
confint(svyby(~h11atota, ~I(nfinr==1), hrssvyhh, na.rm=T, ci=T, svymean))

# Example 5.7 Alternative Bayesian Approach
# create complete data set with variables of interest
hrs.red <- hrs12[hrs12$nfinr == 1, c("h11atota", "secu", "stratum", "nwgthh")]
hrs.red <- hrs.red[complete.cases(hrs.red),]

# need to normalize HRS weights to match what is done for Stan modeling
hrs.red$wtsc <- hrs.red$nwgthh / mean(hrs.red$nwgthh)

# survey design object
hrs.des <- svydesign(id = ~secu, strata = ~stratum, weights = ~wtsc, nest = T, data = hrs.red)

# With Normal Prior, mean = 400000, SD (SE) = 20000
set.seed(41279)
model_formula <- formula("h11atota|weights(wtsc) ~ 1")
mod.brms <- cs_sampling_brms(svydes = hrs.des, brmsmod = brmsformula(model_formula), prior =
c(set_prior("normal(400000,20000)", class = "Intercept")), data = hrs.red, family = gaussian())

# Print results
mod.brms$stan_fit

# Plot posteriors, figure 5.4
plot(mod.brms, varnames = "Intercept")


# Example 5.8 Standard Deviation of Total HH Wealth HRS data
# create a data object with weights only but no design variables
hrswgt <- svydesign(id=~1, weights=~nwgthh, data=hrs12)
summary(hrswgt)

# obtain mean
a <- svymean(~h11atota, design=hrswgt, na.rm=T, deff="replace")
a

# use sqrt of variance to obtain standard deviation
sd <- sqrt(svyvar(~h11atota, design = hrswgt, na.rm=T))
sd

# refer back to Example 5.7 for Bayesian Results from intercept only regression, not shown again here

# Example 5.9 Population Percentiles for total HH Wealth HRS data, in subset of nfinr=1
q <- svyquantile(~h11atota, hrssvysub, c(.25,.5,.75), na.rm=T, ci=T)
q
# Obtain SE from confidence intervals, see R documentation for details
SE(q)
```

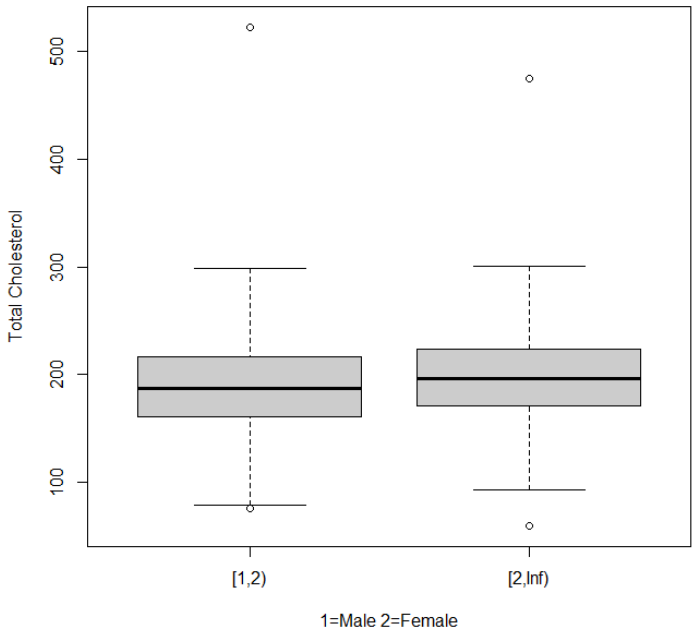
```
# Example 5.10 Lorenz Curve and GINI coefficient not available in R Survey Package

# Example 5.10 Available in "convey" package, requires vardpoor too

library(vardpoor)
library(convey)

# linearized design, use hrssvysub created previously
hrssvyhh_c <- convey_prep(hrssvyhh)

# now can subset to financial respondents (after convey_prep)
sub_hrssvyhh_c <- subset( hrssvyhh_c , nfinr==1)

# run svygini and svylorenz using subset, note that R does not require negative set to 0 as Stata does
svygini( ~h11atota, design = sub_hrssvyhh_c)

# note that the results will differ from Stata due to negative values set to 0 in Stata but not R
svylorenz( ~h11atota, sub_hrssvyhh_c, seq(0,1,.1), alpha = .01 )

# Fig 5.5 example 5.11 Relationship between 2 continuous variables, note this is weighted and design based
svyplot(lbdhdd ~ lbxtc, subset(subnhanes, age >= 18), style="bubble", ylab="HDL", xlab="Total Cholesterol")

# Fig 5.5 with survey-weighted LOWESS smoother
smoother <- svysmooth(lbdhdd ~ lbxtc, subset(subnhanes, age >= 18), bandwidth=10)
# add smooth curve to plot
lines(smoother, col="black", lwd=4, lty=1)


# Example 5.11 Correlation between Total and High Cholesterol, NHANES DATA
# create standardized versions of variables first, then use in regression
nhanesdata$stdlbxtc <- (nhanesdata$lbxtc - 194.4355)/41.05184
summary(nhanesdata$lbxtc + nhanesdata$stdlbxtc)

nhanesdata$stdlbdhdd <- (nhanesdata$lbdhdd -52.83826)/14.93157
summary(nhanesdata$stdlbxtc)

#reset survey design and subset
nhanessvy2 <- svydesign(strata=~sdmvstra, id=~sdmvpsu, weights=~wtmec2yr, data=nhanesdata, nest=T)
subnhanes <- subset(nhanessvy2 , age >= 18)

#Design based linear regression to obtain correlation and correct SE
summary(Ex5_11_svyglm <- svyglm(stdlbxtc ~ stdlbdhdd, design=subnhanes))
confint(Ex5_11_svyglm)


# Example 5.12 Ratio Estimator for HDD to Total Cholesterol
ex5_12 <- svyby (~lbdhdd, denominator=~lbxtc, by=~I(age >= 18), nhanessvy2, na.rm=T, ci=T, svyratio)
ex5_12
confint(ex5_12)


# Example 5.13 Proportions of Diabetes by Gender in Subpopulation of Age >=70
subhrs70 <- subset(hrssvyr, nage >= 70)
ex5_13 <- svyby(~diabetes, ~gender, subhrs70, svymean, keep.names=T, na.rm=T)
print(ex5_13)
confint(ex5_13)
```

```
# Example 5.14 Mean Systolic Blood Pressure by Gender, Age 46+ NHANES
subnhanes46 <-subset(nhanessvy2, age >= 46)
# riagendr 1=Male 2=Female
ex5_14 <- svyby(~bpxsy1, ~riagendr, subnhanes46, svymean, keep.names=T, DEFF=T, na.rm=T)
print(ex5_14)
confint(ex5_14)

# Example 5.15 Differences in Mean HH Wealth by Educational Attainment, HRS data
# Codes for edcat :  1=0-11 2=12 3=13-15 4=16+ years of education
summary(hrssvysub)

ex5_15 <- svyby(~h11atota, ~edcat, hrssvysub, svymean, na.rm=T)
print(ex5_15)
confint(ex5_15)
svycontrast(ex5_15, list(avg=c(.5,0,0,.5), diff=c(1,0,0,-1)))

# Example 5.16 Differences in Total Wealth over Time 2010 to 2012, HRS data
# Use 2010 and 2012 data set prepared in SAS

library(haven)
# use sas data input
hrs_2010_2012 <- read_sas("P:/ASDA3/Replication SAS/hrs_2010_2012_both_f.sas7bdat")
summary(hrs_2010_2012)
names(hrs_2010_2012)

hrs2010_2012 <- svydesign(strata=~stratum, id=~secu, weights=~hhweight, data=hrs_2010_2012, nest=T)
subhrs2010 <- subset(hrs2010_2012, finr2010_2012==1)

ex5_16 <- svyby (~totwealth, ~year, design=subhrs2010, keep.vars=T, svymean)
# get coefficients
coef(ex5_16)
# get standard errors
se(ex5_16)

# use svycontrast
contrast <- svycontrast(ex5_16, list(avg=c(.5,.5), diff=c(1,-1)))
print(contrast)
```

R Results

```
# figures 5.1 and 5.2
svyhist(~lbxtc, subset (nhanessvy2, age >=18), main="", col="grey80", xlab ="Histogram of Total Cholesterol")

#Create a variables called gender for boxplot
nhanessvy2<-update(nhanessvy2, gender=cut(riagendr, c(1, 2, Inf), right=F))

svyboxplot(lbxtc~gender , subset (nhanessvy2, age >=18), col="grey80", ylab="Total Cholesterol", xlab ="1=Male
2=Female")
```





```
> #Example 5.3
> svytotal (~mde, ncsrsvypop, deff=T)
       total       SE  DEff
mde 40092207  2567488 9.028
> confint(svytotal(~mde, ncsrsvypop))
       2.5 %   97.5 %
mde 35060023 45124390
>
> #Example 5.3 MDE over marital status
> ex53 <- svyby (~mde, ~mar3catc, ncsrsvypop, svytotal)
> ex53 <- svyby (~mde, ~mar3catc, ncsrsvypop, svytotal, deff=T)
> ex53
                             mar3catc      mde        se DEff.mde
Married                       Married 20304191 1584108.6 6.817920
Previously Married Previously Married 10360671  702621.5 2.966192
Never Married           Never Married  9427345  773137.6 3.063915
> confint(ex53)
                      2.5 %   97.5 %
Married            17199395 23408986
Previously Married  8983558 11737783
Never Married       7912024 10942667



> #Example 5.4 HRS HH Level Wealth/Total Assets
> svyby (~H11ATOTA, ~I(nfinr==1), hrssvyhh, na.rm=T, svytotal)
      I(nfinr == 1)     H11ATOTA           se
FALSE         FALSE 1.701334e+13 1.031603e+12
TRUE           TRUE 2.526686e+13 1.353710e+12
> confint(svyby (~H11ATOTA, ~I(nfinr==1), hrssvyhh, na.rm=T, ci=T, svytotal))
             2.5 %       97.5 %
FALSE 1.499144e+13 1.903525e+13
TRUE  2.261364e+13 2.792009e+13

> #Example 5.5 HRS HH Income
> svyby (~H11ITOT, ~I(nfinr==1), hrssvyhh, na.rm=T, svymean)
      I(nfinr == 1)  H11ITOT       se
FALSE         FALSE 98737.91 3007.883
TRUE           TRUE 71382.40 1937.229
> confint(svyby (~H11ITOT, ~I(nfinr==1), hrssvyhh, na.rm=T, ci=T, svymean))
         2.5 %   97.5 %
FALSE 92842.57 104633.3
TRUE  67585.50  75179.3
```

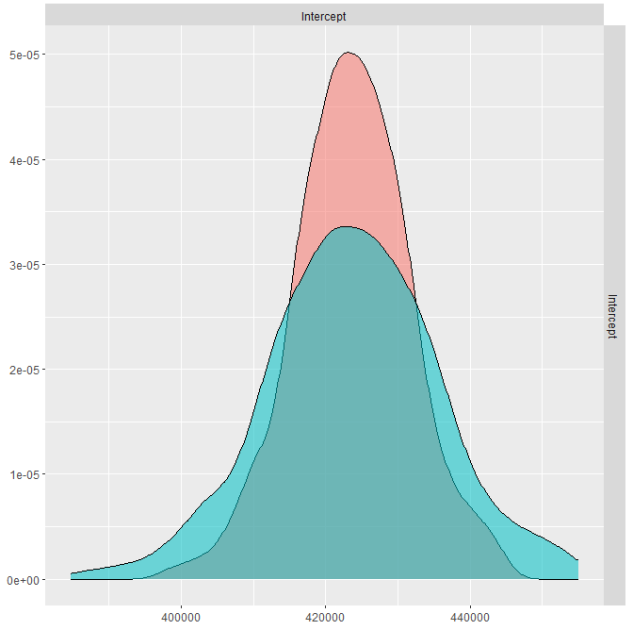
```
> Example 5.5 Bayesian Approach
# Print results
> mod.brms$stan_fit
Inference for Stan model: anon_model.
1 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=1000.

                  mean se_mean      sd       2.5%        25%        50%
Intercept     71338.32   50.82 1071.99   69280.69   70581.08   71356.31
sigma        122158.21   23.03  733.41  120675.49  121675.94  122159.07
lprior          -27.03    0.00    0.05     -27.13     -27.07     -27.03
b_Intercept   71338.32   50.82 1071.99   69280.69   70581.08   71356.31
lp__        -186370.73    0.06    1.06 -186373.56 -186371.16 -186370.40
                  75%      97.5% n_eff Rhat
Intercept     72051.2   73460.27   445    1
sigma        122674.3  123567.19  1014    1
lprior          -27.0     -26.93   499    1
b_Intercept   72051.2   73460.27   445    1
lp__        -186370.0 -186369.70   365    1

Samples were drawn using NUTS(diag_e) at Tue Mar 11 09:02:41 2025.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).

> # With Normal Prior, mean = 70000, SD (SE) = 1000
> mod.brms2 <- cs_sampling_brms(svydes = hrs.des, brmsmod = brmsformula(model_formula), prior =
c(set_prior("normal(70000,1000)", class = "Intercept")), data = hrs.red, family = gaussian())
[1] "compiling stan model"
recompiling to avoid crashing R session
[1] "stan fitting"

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0.00427 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 42.7 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 105.602 seconds (Warm-up)
Chain 1:                14.096 seconds (Sampling)
Chain 1:                119.698 seconds (Total)
Chain 1:
[1] "gradient evaluation"
Warning message:
```

```
In mrbweights(design$cluster, design$strata, design$fpc, ...) :
  Design is sampled with replacement: only first stage used
> mod.brms2$stan_fit
Inference for Stan model: anon_model.
1 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=1000.

                  mean se_mean     sd       2.5%        25%        50%
Intercept     70639.61   27.83 714.29   69248.30   70175.67   70623.06
sigma        122149.39   27.22 691.39  120831.13  121682.20  122145.02
lprior          -22.93    0.02   0.58     -24.55     -23.12     -22.71
b_Intercept   70639.61   27.83 714.29   69248.30   70175.67   70623.06
lp__        -186366.52    0.05   0.97 -186369.12 -186366.87 -186366.22
                  75%      97.5% n_eff Rhat
Intercept     71118.65   72034.69   659 1.00
sigma        122574.12  123551.22   645 1.00
lprior          -22.52     -22.46   605 1.00
b_Intercept   71118.65   72034.69   659 1.00
lp__        -186365.83 -186365.60   394 1.01

Samples were drawn using NUTS(diag_e) at Tue Mar 11 09:06:19 2025.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

```
> #Example 5.6 Mean Systolic Blood Pressure, NHANES data
> a <- svymean(~BPXSY1 , subset (nhanessvy2, age >=18), na.rm=TRUE)
> coef(a)
  BPXSY1
122.0292
> SE(a)
          BPXSY1
BPXSY1 0.6163389
> confint(a)
          2.5 %   97.5 %
BPXSY1 120.8212 123.2372
```

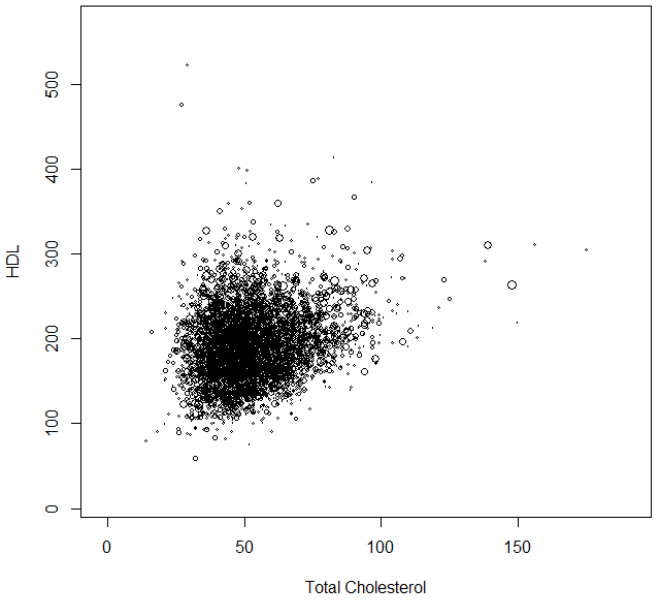
```
> Example 5.6
> # Bayesian approach, flat prior
> set.seed(41279)
> model_formula <- formula("bpxsy1|weights(wtsc) ~ 1")
> mod.brms <- cs_sampling_brms(svydes = nhanes.des, brmsmod = brmsformula(model_formula), data = nhanes.red,
family = gaussian())
[1] "compiling stan model"
recompiling to avoid crashing R session
[1] "stan fitting"

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0.001451 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 14.51 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 5.301 seconds (Warm-up)
Chain 1:                4.441 seconds (Sampling)
Chain 1:                9.742 seconds (Total)
Chain 1:
[1] "gradient evaluation"
Warning message:
In mrbweights(design$cluster, design$strata, design$fpc, ...) :
  Design is sampled with replacement: only first stage used
>
> # Print results
> mod.brms$stan_fit
Inference for Stan model: anon_model.
1 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=1000.

                mean se_mean   sd      2.5%       25%       50%
Intercept     122.02    0.01 0.24    121.54    121.86    122.02
sigma          17.37    0.01 0.17     17.05     17.26     17.38
lprior         -7.47    0.00 0.01     -7.49     -7.48     -7.47
b_Intercept   122.02    0.01 0.24    121.54    121.86    122.02
lp__        -21937.91    0.05 1.02 -21940.46 -21938.30 -21937.61
                75%     97.5% n_eff Rhat
Intercept     122.18    122.46  1001    1
sigma          17.49     17.72   957    1
lprior         -7.46     -7.44   974    1
b_Intercept   122.18    122.46  1001    1
lp__        -21937.19 -21936.95   412    1
> #Example 5.7
```

```
> svyby (~H11ATOTA, ~I(nfinr==1), hrssvyhh, na.rm=T, ci=T, svymean)
      I(nfinr == 1) H11ATOTA       se
FALSE         FALSE 563269.1 26670.34
TRUE           TRUE 428470.8 17353.77
> confint(svyby(~H11ATOTA, ~I(nfinr==1), hrssvyhh, na.rm=T, ci=T, svymean))
         2.5 %   97.5 %
FALSE 510996.2 615542.0
TRUE  394458.0 462483.5

#Example 5.7 Using Alternative Bayesian Approach
> # With Normal Prior, mean = 400000, SD (SE) = 20000
> # Print results
> mod.brms$stan_fit
Inference for Stan model: anon_model.
1 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=1000.

                  mean se_mean      sd       2.5%        25%        50%
Intercept    423606.38  353.97 7955.77  407199.96  418380.45  423627.66
sigma       1113687.40  234.70 6466.77 1101343.40 1108952.54 1113625.82
lprior          -29.93    0.02    0.48     -31.14     -30.19     -29.85
b_Intercept  423606.38  353.97 7955.77  407199.96  418380.45  423627.66
lp__        -217737.15    0.04    0.86 -217739.44 -217737.53 -217736.89
                   75%      97.5% n_eff Rhat
Intercept    428870.98  439998.28   505 1.00
sigma       1118325.22 1125996.74   759 1.00
lprior          -29.58     -29.22   514 1.00
b_Intercept  428870.98  439998.28   505 1.00
lp__        -217736.53 -217736.26   482 1.01
Samples were drawn using NUTS(diag_e) at Tue Mar 11 09:25:04 2025.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).

# Plot posteriors from previous model, intercept only
> plot(mod.brms, varnames = "Intercept")
```

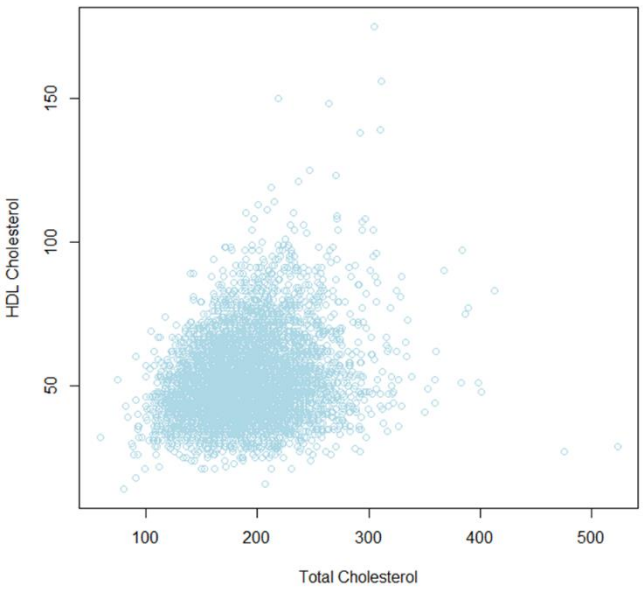


```
#Example 5.8 Standard Deviation of Total HH Wealth HRS data
#Create a data object with weights only but no design variables
> hrswgt <- svydesign(id=~1, weights=~nwgthh, data=hrs)
> #obtain mean
> a <- svymean(~H11ATOTA, design=hrswgt, na.rm=T, deff="replace")
> a
           mean     SE   DEff
H11ATOTA 474129  11592 2.0536
> # use sqrt of variance to obtain standard deviation
> sd <- sqrt(svyvar(~H11ATOTA, design = hrswgt, na.rm=T))
> sd
         variance        SE
H11ATOTA  1143716 1.657e+11



#Example 5.9 Population Percentiles for total HH Wealth HRS data, in subset of nfinr=1
> q <- svyquantile(~H11ATOTA, hrssvysub, c(.25,.5,.75), na.rm=T, ci=T)
> q
$H11ATOTA
     quantile ci.2.5 ci.97.5        se
0.25    22000  18000   26900  2221.401
0.5    142000 127000  158000  7737.463
0.75   440000 404500  480000 18844.465

attr(,"hasci")
[1] TRUE
attr(,"class")
[1] "newsvyquantile"
> # Obtain SE from confidence intervals, see R documentation for details
> SE(q)
H11ATOTA.0.25  H11ATOTA.0.5 H11ATOTA.0.75
     2221.401      7737.463     18844.465



> #Example 5.10 Lorenz Curve and GINI coefficient not available in R Survey Package


> #Example 5.10 Available in "convey" package
> library(vardpoor)
> library(convey)
>  # linearized design, use hrssvysub created previously
> hrssvyhh_c <- convey_prep(hrssvyhh)
> # now can subset to financial respondents (after convey_prep)
> sub_hrssvyhh_c <- subset( hrssvyhh_c , nfinr==1)
> # run svygini and svylorenz using subset, note that R does not require negative set to 0 as Stata
> svygini( ~H11ATOTA, design = sub_hrssvyhh_c)
            gini     SE
H11ATOTA 0.73897 0.0094
> svylorenz( ~H11ATOTA, sub_hrssvyhh_c, seq(0,1,.1), alpha = .01 )
           lorenz     SE
L(0)    0.0000000 0.0000
L(0.1) -0.0071723 0.0008
L(0.2) -0.0065605 0.0009
L(0.3) -0.0012259 0.0011
L(0.4)  0.0131167 0.0018
```

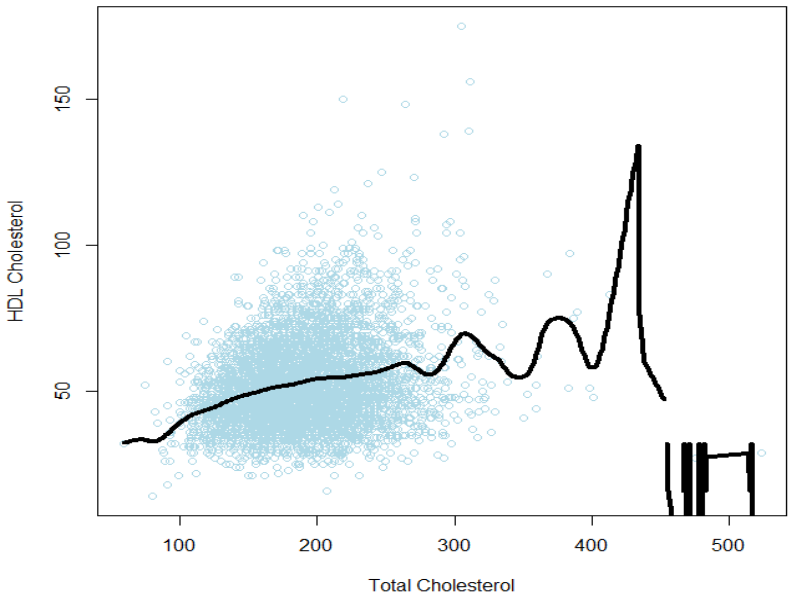
```
L(0.5)  0.0388491 0.0029
L(0.6)  0.0811900 0.0046
L(0.7)  0.1464493 0.0068
L(0.8)  0.2498023 0.0099
L(0.9)  0.4234323 0.0140
#Example 5.11, Figure 5.5 Plot of Relationship between 2 continuous variables, note this is weighted and design based
> svyplot(lbxtc~lbdhdd, subset(subnhanes, age>=18), style="bubble", ylab="HDL", xlab="Total Cholesterol")
```




```
> # unweighted scatter plot
> plot(lbdhdd ~ lbxtc, subset(nhanes1112, age >= 18), xlab="Total Cholesterol", ylab="HDL Cholesterol", col="lightblue")
```



```
> # survey-weighted LOWESS smoother
> smoother <- svysmooth(lbdhdd ~ lbxtc, subset(nhanes.des, age >= 18), bandwidth=10)
> # add smooth curve to plot
> lines(smoother, col="black", lwd=4, lty=1)
```




```
#EXAMPLE 5.11 Correlation between Total and High Cholesterol, NHANES Data
> summary(nhanesdata$lbxtc + nhanesdata$stdlbxtc)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   55.7   153.0   178.6   182.9   209.4   531.0    2768

> #nhanesdata$stdlbdhdd <- (nhanesdata$LBDHDD-52.83826)/14.93157
> summary(nhanesdata$stdlbxtc)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
-3.2991 -0.9850 -0.3760 -0.2737  0.3548  8.0036    2768

> #reset survey design and subset
> nhanessvy2 <- svydesign(strata=~sdmvstra, id=~sdmvpsu, weights=~WTMEC2YR, data=nhanesdata, nest=T)
> subnhanes <- subset(nhanessvy2 , age >= 18)

> #Design based linear regression to obtain correlation and correct SE
> summary(Ex5_11_svyglm <- svyglm(stdlbxtc ~ stdlbdhdd, design=subnhanes))
Call:
svyglm(formula = stdlbxtc ~ stdlbdhdd, design = subnhanes)

Survey design:
subset(nhanessvy2, age >= 18)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -8.916e-07  2.568e-02    0.00        1
stdlbdhdd    2.414e-01  1.104e-02   21.87  2.4e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Dispersion parameter for gaussian family taken to be 1.003306)
Number of Fisher Scoring iterations: 2
```

```
#Example 5.12 Ratio Estimator for HDD to Total Cholesterol
> ex5_12 <- svyby (~lbdhdd, denominator=~lbxtc, by=~I(age >= 18), nhanessvy2, na.rm=T, ci=T, svyratio)
> confint(ex5_12)
          2.5 %    97.5 %
FALSE 0.3265713 0.3374500
TRUE  0.2660798 0.2774246



#Example 5.13 Proportions of Diabetes by Gender in Subpopulation of Age >=70
> subhrs70 <- subset(hrssvyr, nage >= 70)
> ex5_13 <- svyby(~diabetes, ~gender, subhrs70, svymean, keep.names=T, na.rm=T)
> print(ex5_13)
  gender  diabetes          se
1      1 0.2736113 0.007468441
2      2 0.2269743 0.008554564
> confint(ex5_13)
      2.5 %    97.5 %
1 0.2589734 0.2882491
2 0.2102077 0.2437409
>
> #Example 5.14 Mean Systolic Blood Pressure by Gender, Age 46+ NHANES
> subnhanes46 <-subset(nhanessvy2, age >= 46)
>
> #riagendr 1=M 2=F
> ex5_14 <- svyby(~BPXSY1, ~riagendr, subnhanes46, svymean, keep.names=T, na.rm=T)
> print(ex5_14)
  riagendr   BPXSY1        se
1        1 128.3005 0.8687054
2        2 128.1820 0.9460163
> confint(ex5_14)
     2.5 %   97.5 %
1 126.5979 130.0032
2 126.3278 130.0361



#Example 5.14 Mean Systolic Blood Pressure by Gender, Age 46+ NHANES
> subnhanes46 <-subset(nhanessvy2, age >= 46)
>
> #riagendr 1=M 2=F
> ex5_14 <- svyby(~BPXSY1, ~riagendr, subnhanes46, svymean, keep.names=T, na.rm=T)
> print(ex5_14)
  riagendr   BPXSY1        se
1        1 128.3005 0.8687054
2        2 128.1820 0.9460163
> confint(ex5_14)
     2.5 %   97.5 %
1 126.5979 130.0032
2 126.3278 130.0361
```

```
> #Example 5.15 Differences in Mean HH Wealth by Educational Attainment, HRS data
> #CODES FOR EDCAT: 1=0-11 2=12 3=13-15 4=16+ YEARS OF EDUCATION
> summary(hrssvysub)
Stratified 1 - level Cluster Sampling design (with replacement)
With (112) clusters.
subset(hrssvyhh, nfinr == 1)
Probabilities:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
5.594e-05 1.959e-04 3.110e-04       Inf 5.961e-04       Inf
Stratum Sizes:
             1   2   3   4   5   6   7   8   9  10  11  12  13  14  15 16
obs        257 409 247 228 194 245 281 275 253 228 185 253 134 105 125 96
design.PSU   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2  2
actual.PSU   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2  2
            17  18  19  20  21  22 23 24 25  26  27  28  29  30  31  32
obs        192 123 204 273 233 129 77 52 61 217 424 308 350 364 364 314
design.PSU   2   2   2   2   2   2  2  2  2   2   2   2   2   2   2   2
actual.PSU   2   2   2   2   2   2  2  2  2   2   2   2   2   2   2   2
            33  34  35  36  37  38  39  40  41  42  43  44  45  46  47
obs        405 339 217 358 136 258 242 454 345 349 498 445 406 467 308
design.PSU   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
actual.PSU   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
            48  49  50  51  52 53 54  55 56
obs        203 244 463 387 111 96 40 124 96
design.PSU   2   2   2   2   2  2  2   2  2
actual.PSU   2   2   2   2   2  2  2   2  2
Data variables:
 [1] "hhid"       "pn"         "R11BMI"     "nage"       "nfinr"
 [6] "gender"     "secu"       "stratum"    "nwgthh"     "nwgtr"
[11] "H11ATOTA"   "H11ITOT"    "marcat"     "edcat"      "racecat"
[16] "diabetes"   "numfalls24" "age65p"     "arthritis"  "finr"
[21] "female"     "age70"

> ex5_15 <- svyby(~H11ATOTA, ~edcat, hrssvysub, svymean, na.rm=T)
> print(ex5_15)
  edcat H11ATOTA       se
1     1 122088.6 10595.60
2     2 259027.2  9802.47
3     3 336308.6 17201.79
4     4 834141.0 46477.79
> confint(ex5_15)
     2.5 %   97.5 %
1 101321.6 142855.6
2 239814.7 278239.6
3 302593.7 370023.5
4 743046.2 925235.8
> svycontrast(ex5_15, list(avg=c(.5,0,0,.5), diff=c(1,0,0,-1)))
     contrast    SE
avg    478115 23835
diff  -712052 47670
```

```
# Example 5.15 Alternative Bayesian approach
> # Bayesian approach, flat prior
> set.seed(41279)
> model_formula <- formula("h11atota|weights(wtsc) ~ edcat1")
> mod.brms <- cs_sampling_brms(svydes = hrs.des, brmsmod = brmsformula(model_formula), data = hrs.red, family =
gaussian())
[1] "compiling stan model"
[1] "stan fitting"

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0.002207 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 22.07 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 112.688 seconds (Warm-up)
Chain 1:                7.492 seconds (Sampling)
Chain 1:                120.18 seconds (Total)
Chain 1:
[1] "gradient evaluation"
Warning message:
In mrbweights(design$cluster, design$strata, design$fpc, ...) :
  Design is sampled with replacement: only first stage used
>
> # Print results
> mod.brms$stan_fit
Inference for Stan model: anon_model.
1 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=1000.

                  mean se_mean       sd       2.5%        25%        50%
b[1]        -714328.45 1382.52 39318.31 -787398.12 -742169.73 -713721.08
Intercept    489282.21  643.77 18720.77  455206.21  476119.83  488701.43
sigma       1475791.70  385.55 12459.57 1451900.58 1467600.99 1475073.37
lprior          -34.73    0.00     0.14     -35.00     -34.82     -34.72
b_Intercept  831857.96  605.38 22082.74  788225.76  817499.64  831468.02
lp__         -99090.33    0.05     1.16  -99093.46  -99090.83  -99090.01
                  75%      97.5% n_eff Rhat
b[1]        -687158.32 -638514.87   809    1
Intercept    501496.04  527016.88   846    1
sigma       1484257.31 1499886.78  1044    1
lprior          -34.63     -34.46   815    1
b_Intercept  847382.64  874227.85  1331    1
lp__         -99089.50  -99089.02   506    1
```

```
Samples were drawn using NUTS(diag_e) at Tue Mar 11 09:55:42 2025.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).


> # With informative prior for coefficient
> mod.brms <- cs_sampling_brms(svydes = hrs.des, brmsmod = brmsformula(model_formula), prior =
c(set_prior("normal(-700000,40000)", class = "b", coef="edcat1")), data = hrs.red, family = gaussian())
[1] "compiling stan model"
[1] "stan fitting"

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0.002221 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 22.21 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 107.866 seconds (Warm-up)
Chain 1:                6.471 seconds (Sampling)
Chain 1:                114.337 seconds (Total)
Chain 1:
[1] "gradient evaluation"
Warning message:
In mrbweights(design$cluster, design$strata, design$fpc, ...) :
  Design is sampled with replacement: only first stage used
> mod.brms$stan_fit
Inference for Stan model: anon_model.
1 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=1000.

                  mean se_mean       sd       2.5%        25%        50%
b[1]        -706037.28  938.40 28602.24 -767783.07 -725412.31 -706001.45
Intercept    491640.72  547.15 17964.01  455484.21  479337.62  492298.13
sigma       1476180.16  355.25 12712.81 1451441.85 1467936.16 1475660.19
lprior          -46.53    0.02     0.40     -47.73     -46.65     -46.43
b_Intercept  830240.21  541.82 20544.48  793277.62  814591.42  829873.99
lp__         -99101.91    0.05     1.18  -99104.88  -99102.42  -99101.63
                   75%      97.5% n_eff Rhat
b[1]        -686439.35 -651279.80   929    1
Intercept    503305.84  525090.92  1078    1
sigma       1484201.16 1503216.29  1281    1
lprior          -46.30     -46.06   451    1
```

```
b_Intercept  845749.59  866541.10  1438    1
lp__         -99101.05  -99100.58   511    1

Samples were drawn using NUTS(diag_e) at Tue Mar 11 10:00:35 2025.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat=1).



#Example 5.16 Differences in Total Wealth over Time 2010 to 2012, HRS data
> ex5_16 <- svyby (~totwealth, ~year, design=subhrs2010, keep.vars=T, svymean)
> coef(ex5_16)
    2010     2012
432829.6 437807.6
> SE(ex5_16)
[1] 16010.53 17016.29
> contrast <- svycontrast(ex5_16, list(avg=c(.5,.5), diff=c(1,-1)))
Warning message:
In vcov.svyby(stat) : Only diagonal elements of vcov() available
> print(contrast)
     contrast    SE
avg  435318.6 11682
diff  -4978.1 23364
> confint(contrast)
         2.5 %    97.5 %
avg  412421.98 458215.21
diff -50771.29  40815.16
```