ASDA3 Analysis Examples Replication Chapter 13

R Code

```
# Note: the svylme from the Survey package only fits LMM, not GLMM models
# Note SEM with survey data was previously available from lavaan.survey package, but this package was recently withdrawn
from CRAN, not shown here


# Section 13.3.1 : develop function for ci for alpha, this is for the SEM example in run Mplus, shown as an example when
working with Mplus
ci.alpha_in_cds = function(alpha, se)
{
   l = log(alpha/(1-alpha)) # use logit
                            # transformation first
   sel = se/(alpha*(1-alpha)) # get SE of logit of alpha
   ci_l_lo = l-1.96*sel # lower and upper endpoint
                        # of 95%-CI
   ci_l_up = l+1.96*sel # for logit of alpha
   ci_lo = 1/(1+exp(-ci_l_lo)) # transform back both ends
   ci_up = 1/(1+exp(-ci_l_up)) # of a CI for
                               # original alpha
   ci = c(ci_lo, ci_up) # print the resulting CI
   ci                   # of coefficient alpha to screen
}

ci.alpha_in_cds(0.880,0.010)
ci.alpha_in_cds(0.689,0.006)

###
# 13.4 Methods for Non-probability samples

# Analyses using NONPROBEST
# if needed: install.packages("NonProbEst")
library(NonProbEst)

# Read data from probability sample (ANES) from population
popdata <- read.csv("P:/ASDA3/Data Sets for Analysis Examples and Stata R Code/anes_wisc.csv", h=T)

# Remove cases with missing values
popdata2 <- popdata[complete.cases(popdata),]

# Read data from WI poll, with common covariates, QR 13.4.1
poll_nomiss <- read.csv("P:/ASDA3/Data Sets for Analysis Examples and Stata R Code/abc_wisc.csv", h=T)

# Declare covariates
covariates <- c("male", "age1", "age2", "age3", "age4", "age5", "somecoll", "coll", "postcoll", "black", "hisp", "other",
"demparty", "otherparty", "lib", "con")

# Set propensities
pi <- propensities(poll_nomiss, popdata2, covariates,
                   algorithm = "knn", smooth = T, tuneGrid = data.frame(k = seq(3, 11, by = 2)))

# Calculate weights for NP sample per Valliant (2020)
wiv <- valliant_weights(propensities = pi$convenience)

# calculate weights for NP sample per Valliant and Dever (2011)
wivd <- vd_weights(convenience_propensities = pi$convenience,
                   reference_propensities = pi$reference)
```

```
summary(wiv)

mean_estimation(sample = poll_nomiss, weights = wiv, estimated_vars = "trumpind")

# size of population for WI
n1 <- 3298041

# Estimation of the 95% confidence interval
estim <- mean_estimation(sample = poll_nomiss, weights = wiv, estimated_vars = "trumpind")

std_dev <- fast_jackknife_variance(sample = poll_nomiss, weights = wiv, estimated_vars = "trumpind", N = n1)
std_dev

confidence_interval(estimation = estim, std_dev = std_dev, confidence = 0.95)

# Calibration, per Valliant and Dever (2011)

totals <- data.frame(male = 0.5*3298041, demparty = 0.4*3298041, otherparty = 0.2*3298041)
totals <- colSums(totals)

aux.vars <- c("male", "demparty", "otherparty")

initial_weights <- wiv   # from initial code above

wiv2 <- calib_weights(poll_nomiss[, aux.vars], totals, initial_weights, N = 3298041, method = "raking")

# Re-compute the estimated proportion and re-apply the JRR estimation
mean_estimation(sample = poll_nomiss, weights = wiv2, estimated_vars = "trumpind", N = 3298041)

v <- fast_jackknife_variance(poll_nomiss, wiv2, estimated_vars = "trumpind", N = 3298041)

ci <- confidence_interval(mean_estimation(sample = poll_nomiss, weights = wiv2, estimated_vars = "trumpind", N =
3298041), sqrt(v))
ci # check values to match .4097016 and .3706205, .4487827

##
# DR approach in nonprobsvy
# If needed: install.packages("nonprobsvy")

library(nonprobsvy)

# survey design object for reference sample
pop.des <- svydesign(ids = ~1, data = popdata2, weights = ~ weight)

# run DR model with survey design
nonprob(
  selection = ~ male + age1 + age2 + age3 + age4 + age5 + somecoll + coll + postcoll + black + hisp + other + demparty +
otherparty + lib + con,
  outcome = trumpind ~ male + age1 + age2 + age3 + age4 + age5 + somecoll + coll + postcoll + black + hisp + other +
demparty + otherparty + lib + con,
  data = poll_nomiss,
  svydesign = pop.des,
  method_outcome = "glm",
  family_outcome = "binomial"
)
```

```
##
# MUBP (Measure of unadjusted bias for proportions)

# Install packages and then load
install.packages("tidyverse","broom")

# Load necessary libraries
library(tidyverse)
library(broom)
library(survey)

# Load necessary functions for Bayesian MUBP approach, available via github.com as follows
source("https://github.com/bradytwest/IndicesOfNISB/raw/master/proxyDrawsMUBP_sumstats.R");
source("https://github.com/bradytwest/IndicesOfNISB/raw/master/mubp_functions.R");

# size of non-selected sample for WI
n0 <- 3298041

# read data from probability sample (ANES) from population
popdata <- read_csv("P:/ASDA3/Data Sets for Analysis Examples and Stata R Code/anes_wisc.csv")

# we only need to account for the weights to get pop. estimates
pop.des <- svydesign(ids = ~1, data = popdata, weights = ~ weight)

# get means for common covariates
means.pop <- c(svymean(~male, pop.des, na.rm=T)[1],     #ref=female
               svymean(~age1, pop.des, na.rm=T)[1],     #ref=age6
               svymean(~age2, pop.des, na.rm=T)[1],
               svymean(~age3, pop.des, na.rm=T)[1],
               svymean(~age4, pop.des, na.rm=T)[1],
               svymean(~age5, pop.des, na.rm=T)[1],
               svymean(~somecoll, pop.des, na.rm=T)[1], #ref=hsless
               svymean(~coll, pop.des, na.rm=T)[1],
               svymean(~postcoll, pop.des, na.rm=T)[1],
                svymean(~black, pop.des, na.rm=T)[1],    #ref=white
               svymean(~hisp, pop.des, na.rm=T)[1],
               svymean(~other, pop.des, na.rm=T)[1],
               svymean(~demparty, pop.des, na.rm=T)[1], #ref=repparty
               svymean(~otherparty, pop.des, na.rm=T)[1],
               svymean(~lib, pop.des, na.rm=T)[1],      #ref=mod/noideo
               svymean(~con, pop.des, na.rm=T)[1])

# variance-covariance matrix
varcov.pop <- svyvar(~ male + age1 + age2 + age3 + age4 + age5 + somecoll + coll + postcoll + black + hisp + other +
demparty + otherparty + lib + con, na.rm=TRUE, design=pop.des)

# read in data and perform the MUBP calculations
poll_nomiss <- read_csv("P:/ASDA3/Data Sets for Analysis Examples and Stata R Code/abc_wisc.csv")

###################
# MUBP calculations
###################

# Selected sample Y
y1 <- poll_nomiss$trumpind
# Selected sample size
n1 <- length(y1)

# Probit model using selected sample [Y|Z,S=1]
```

```
fit <- glm(trumpind ~ male + age1 + age2 + age3 + age4 + age5 + somecoll + coll + postcoll + black + hisp + other +
demparty + otherparty + lib + con, family=binomial(link="probit"), data=poll_nomiss)

# Predictors Z
z1_withInt <- model.matrix(fit)
z1 <- z1_withInt[,-1] # remove intercept (added later)

set.seed(41279)

draws <- proxyDrawsMSB(y1, z1, n1, means.pop, varcov.pop, n0-n1, phi=0, drawphi=TRUE, scaleX=TRUE, nreps=1000)

ymean_mubpadj_p50 = mean(y1) - quantile(draws$msb,0.5)
ymean_mubpadj_p50
ymean_mubpadj_lb = mean(y1) - quantile(draws$msb,0.975)
ymean_mubpadj_lb
ymean_mubpadj_ub = mean(y1) - quantile(draws$msb,0.025)
ymean_mubpadj_ub
# Match .509, 0.47, 0.554


# RPMS example
install.packages("rpms")
library(rpms)

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

# Gather needed variables
nhanes <- nhanes1112[, c(7, 8, 9, 10, 11, 19, 24, 34, 35, 36)]

# Complete cases
nhanes <- nhanes[complete.cases(nhanes),]

dtree <- rpms(bpxsy1 ~ bmxbmi + age + white + black + other + indfmpir, data = nhanes, weights = ~wtmec2yr, strata =
~sdmvstra, clusters = ~sdmvpsu)

dtree$partition

dtree

# Ignore the NHANES design feature as a comparison
stree <- rpms(bpxsy1 ~ bmxbmi + age + white + black + other + indfmpir, data = nhanes)

stree$partition

stree
```

R Results

```
> # ASDA3, Chapter 13
>
> # Note: the svylme from the Survey package only fits LMM, not GLMM models
> # Note SEM with survey data was previously available from lavaan.survey package, but this package was recently
withdrawn from CRAN, not shown here
>
>
> # Section 13.3.1 : develop function for ci for alpha, this is for the SEM example in run Mplus, shown as an example
when working with Mplus
> ci.alpha_in_cds = function(alpha, se)
+ {
+    l = log(alpha/(1-alpha)) # use logit
+                             # transformation first
+    sel = se/(alpha*(1-alpha)) # get SE of logit of alpha
+    ci_l_lo = l-1.96*sel # lower and upper endpoint
+                         # of 95%-CI
+    ci_l_up = l+1.96*sel # for logit of alpha
+    ci_lo = 1/(1+exp(-ci_l_lo)) # transform back both ends
+    ci_up = 1/(1+exp(-ci_l_up)) # of a CI for
+                                # original alpha
+    ci = c(ci_lo, ci_up) # print the resulting CI
+    ci                   # of coefficient alpha to screen
+ }
>
> ci.alpha_in_cds(0.880,0.010)
[1] 0.859 0.898
> ci.alpha_in_cds(0.689,0.006)
[1] 0.677 0.701
>
> ###
> # 13.4 Methods for Non-probability samples
>
> # Analyses using NONPROBEST
> # if needed: install.packages("NonProbEst")
> library(NonProbEst)
>
> # Read data from probability sample (ANES) from population
> popdata <- read.csv("P:/ASDA3/Data Sets for Analysis Examples and Stata R Code/anes_wisc.csv", h=T)
>
> # Remove cases with missing values
> popdata2 <- popdata[complete.cases(popdata),]
>
> # Read data from WI poll, with common covariates, QR 13.4.1
> poll_nomiss <- read.csv("P:/ASDA3/Data Sets for Analysis Examples and Stata R Code/abc_wisc.csv", h=T)
>
> # Declare covariates
> covariates <- c("male", "age1", "age2", "age3", "age4", "age5", "somecoll", "coll", "postcoll", "black", "hisp",
"other", "demparty", "otherparty", "lib", "con")
>
> # Set propensities
> pi <- propensities(poll_nomiss, popdata2, covariates,
+                    algorithm = "knn", smooth = T, tuneGrid = data.frame(k = seq(3, 11, by = 2)))
>
> # Calculate weights for NP sample per Valliant (2020)
> wiv <- valliant_weights(propensities = pi$convenience)
```

```
> # calculate weights for NP sample per Valliant and Dever (2011)
> wivd <- vd_weights(convenience_propensities = pi$convenience,
+                    reference_propensities = pi$reference)
>
> summary(wiv)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    1.10    1.18    1.21    1.28    2.00
>
> mean_estimation(sample = poll_nomiss, weights = wiv, estimated_vars = "trumpind")
trumpind
   0.394
>
> # size of population for WI
> n1 <- 3298041
>
> # Estimation of the 95% confidence interval
> estim <- mean_estimation(sample = poll_nomiss, weights = wiv, estimated_vars = "trumpind")
>
> std_dev <- fast_jackknife_variance(sample = poll_nomiss, weights = wiv, estimated_vars = "trumpind", N = n1)
> std_dev
trumpind
0.000307
>
> confidence_interval(estimation = estim, std_dev = std_dev, confidence = 0.95)
lower.trumpind upper.trumpind
         0.394          0.395
>
> # Calibration, per Valliant and Dever (2011)
>
> totals <- data.frame(male = 0.5*3298041, demparty = 0.4*3298041, otherparty = 0.2*3298041)
> totals <- colSums(totals)
>
> aux.vars <- c("male", "demparty", "otherparty")
>
> initial_weights <- wiv   # from initial code above
>
> wiv2 <- calib_weights(poll_nomiss[, aux.vars], totals, initial_weights, N = 3298041, method = "raking")
>
> # Re-compute the estimated proportion and re-apply the JRR estimation
> mean_estimation(sample = poll_nomiss, weights = wiv2, estimated_vars = "trumpind", N = 3298041)
trumpind
    0.41
>
> v <- fast_jackknife_variance(poll_nomiss, wiv2, estimated_vars = "trumpind", N = 3298041)
>
> ci <- confidence_interval(mean_estimation(sample = poll_nomiss, weights = wiv2, estimated_vars = "trumpind", N =
3298041), sqrt(v))
> ci # check values to match .4097016 and .3706205, .4487827
lower.trumpind upper.trumpind
         0.371          0.449

> # check values to match .4097016 and .3706205, .4487827
```

```
> ##
> # DR approach in nonprobsvy
> # If needed: install.packages("nonprobsvy")
>
> library(nonprobsvy)
>
> # survey design object for reference sample
> pop.des <- svydesign(ids = ~1, data = popdata2, weights = ~ weight)
>
> # run DR model with survey design
> nonprob(
+   selection = ~ male + age1 + age2 + age3 + age4 + age5 + somecoll + coll + postcoll + black + hisp + other + demparty
+ otherparty + lib + con,
+   outcome = trumpind ~ male + age1 + age2 + age3 + age4 + age5 + somecoll + coll + postcoll + black + hisp + other +
demparty + otherparty + lib + con,
+   data = poll_nomiss,
+   svydesign = pop.des,
+   method_outcome = "glm",
+   family_outcome = "binomial"
+ )

Call:
nonprob(data = poll_nomiss, selection = ~male + age1 + age2 +
    age3 + age4 + age5 + somecoll + coll + postcoll + black +
    hisp + other + demparty + otherparty + lib + con, outcome = trumpind ~
    male + age1 + age2 + age3 + age4 + age5 + somecoll + coll +
        postcoll + black + hisp + other + demparty + otherparty +
        lib + con, svydesign = pop.des, method_outcome = "glm",
    family_outcome = "binomial")

Estimated population mean with overall std.err and confidence interval:

          mean     SE lower_bound upper_bound
trumpind 0.521 0.0397       0.443       0.598
>

> ##
> # MUBP (Measure of unadjusted bias for proportions)
>
> # Install packages and then load
> install.packages("tidyverse","broom")
Warning: package ‘tidyverse’ is in use and will not be installed
>
> # Load necessary libraries
> library(tidyverse)
> library(broom)
> library(survey)
>
> # Load necessary functions for Bayesian MUBP approach, available via github.com as follows
> source("https://github.com/bradytwest/IndicesOfNISB/raw/master/proxyDrawsMUBP_sumstats.R");
> source("https://github.com/bradytwest/IndicesOfNISB/raw/master/mubp_functions.R");
Loading required package: msm
Warning message:
In library(package, lib.loc = lib.loc, character.only = TRUE, logical.return = TRUE,  :
  there is no package called ‘msm’
>
```

```
> # size of non-selected sample for WI
> n0 <- 3298041

> # read data from probability sample (ANES) from population
> popdata <- read_csv("P:/ASDA3/Data Sets for Analysis Examples and Stata R Code/anes_wisc.csv")

[1mindexing[0m [34manes_wisc.csv[0m [==========================================================] [32m776.85MB/s[0m, eta: [36m
0s[0m

Rows: 205 Columns: 32
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr  (2): state, state_registered
dbl (30): weight, stratum, cluster, registered, likely, male, age1, age2, age3, age4, age5, age6...

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
>
> # we only need to account for the weights to get pop. estimates
> pop.des <- svydesign(ids = ~1, data = popdata, weights = ~ weight)
>
> # get means for common covariates
> means.pop <- c(svymean(~male, pop.des, na.rm=T)[1],     #ref=female
+                svymean(~age1, pop.des, na.rm=T)[1],     #ref=age6
+                svymean(~age2, pop.des, na.rm=T)[1],
+                svymean(~age3, pop.des, na.rm=T)[1],
+                svymean(~age4, pop.des, na.rm=T)[1],
+                svymean(~age5, pop.des, na.rm=T)[1],
+                svymean(~somecoll, pop.des, na.rm=T)[1], #ref=hsless
+                svymean(~coll, pop.des, na.rm=T)[1],
+                svymean(~postcoll, pop.des, na.rm=T)[1],
+               svymean(~black, pop.des, na.rm=T)[1],    #ref=white
+                svymean(~hisp, pop.des, na.rm=T)[1],
+                svymean(~other, pop.des, na.rm=T)[1],
+                svymean(~demparty, pop.des, na.rm=T)[1], #ref=repparty
+                svymean(~otherparty, pop.des, na.rm=T)[1],
+                svymean(~lib, pop.des, na.rm=T)[1],      #ref=mod/noideo
+                svymean(~con, pop.des, na.rm=T)[1])
>
> # variance-covariance matrix
> varcov.pop <- svyvar(~ male + age1 + age2 + age3 + age4 + age5 + somecoll + coll + postcoll + black + hisp + other +
demparty + otherparty + lib + con, na.rm=TRUE, design=pop.des)
>
> # read in data and perform the MUBP calculations
> poll_nomiss <- read_csv("P:/ASDA3/Data Sets for Analysis Examples and Stata R Code/abc_wisc.csv")

[1mindexing[0m [34mabc_wisc.csv[0m [==========================================================] [32m2.15GB/s[0m, eta: [36m
0s[0m

Rows: 803 Columns: 29
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr  (1): state
dbl (28): weight, trumpind, male, age1, age2, age3, age4, age5, age6, hsless, somecoll, coll, po...

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
> ###################
> # MUBP calculations
> ###################

> # Selected sample Y
> y1 <- poll_nomiss$trumpind
> # Selected sample size
> n1 <- length(y1)
>
> # Probit model using selected sample [Y|Z,S=1]
> fit <- glm(trumpind ~ male + age1 + age2 + age3 + age4 + age5 + somecoll + coll + postcoll + black + hisp + other +
demparty + otherparty + lib + con, family=binomial(link="probit"), data=poll_nomiss)
>
> # Predictors Z
> z1_withInt <- model.matrix(fit)
> z1 <- z1_withInt[,-1] # remove intercept (added later)
>
> set.seed(41279)
>
> draws <- proxyDrawsMSB(y1, z1, n1, means.pop, varcov.pop, n0-n1, phi=0, drawphi=TRUE, scaleX=TRUE, nreps=1000)
>
> ymean_mubpadj_p50 = mean(y1) - quantile(draws$msb,0.5)
> ymean_mubpadj_p50
  50%
0.509
> ymean_mubpadj_lb = mean(y1) - quantile(draws$msb,0.975)
> ymean_mubpadj_lb
97.5%
 0.47
> ymean_mubpadj_ub = mean(y1) - quantile(draws$msb,0.025)
> ymean_mubpadj_ub
 2.5%
0.554
> # Match .509, 0.47, 0.554
```

```
> # RPMS example
> install.packages("rpms")
Warning: package ‘rpms’ is in use and will not be installed
> library(rpms)
>
> load("P:/ASDA3/Data Sets for Analysis Examples and Stata R Code/nhanes1112.rdata")
>
> # Gather needed variables
> nhanes <- nhanes1112[, c(7, 8, 9, 10, 11, 19, 24, 34, 35, 36)]

> # Complete cases
> nhanes <- nhanes[complete.cases(nhanes),]

> dtree <- rpms(bpxsy1 ~ bmxbmi + age + white + black + other + indfmpir, data = nhanes, weights = ~wtmec2yr, strata =
~sdmvstra, clusters = ~sdmvpsu)


> dtree$partition
   node    n
1    16  175
2    17  341
3     9  364
4    80  156
5    81  146
6   164  306
7   165  195
8    83  204
9    21  696
10   11 1562
11    6 1434
12   14  294
13   15  286

                                                                                                  splits
1                                                 age <= 50.5 & bmxbmi <= 19.95 & age <= 12.5 & bmxbmi <= 16.15
2                                                  age <= 50.5 & bmxbmi <= 19.95 & age <= 12.5 & bmxbmi > 16.15
3                                                                    age <= 50.5 & bmxbmi <= 19.95 & age > 12.5
4              age <= 50.5 & bmxbmi > 19.95 & bmxbmi <= 26.85 & age <= 27.5 & age <= 13.5 & bmxbmi <= 22.55
5               age <= 50.5 & bmxbmi > 19.95 & bmxbmi <= 26.85 & age <= 27.5 & age <= 13.5 & bmxbmi > 22.55
6  age <= 50.5 & bmxbmi > 19.95 & bmxbmi <= 26.85 & age <= 27.5 & age > 13.5 & black <= 0.5 & white <= 0.5
7   age <= 50.5 & bmxbmi > 19.95 & bmxbmi <= 26.85 & age <= 27.5 & age > 13.5 & black <= 0.5 & white > 0.5
8                  age <= 50.5 & bmxbmi > 19.95 & bmxbmi <= 26.85 & age <= 27.5 & age > 13.5 & black > 0.5
9                                               age <= 50.5 & bmxbmi > 19.95 & bmxbmi <= 26.85 & age > 27.5
10                                                           age <= 50.5 & bmxbmi > 19.95 & bmxbmi > 26.85
11                                                                                 age > 50.5 & age <= 70.5
12                                                                        age > 50.5 & age > 70.5 & age <= 78.5
13                                                                         age > 50.5 & age > 70.5 & age > 78.5
```

	value	cvar	mean	N_hat	p_se
1	97.9	1.56e-05	97.9	3650921	7.92
2	102	4.12e-05	101.6	7665499	9.20
3	107	1.18e-05	106.9	11310735	10.63
4	103	2.05e-05	102.7	3663154	8.52
5	108	Inf	107.9	3032237	Inf
6	109	Inf	109.2	6874978	Inf
7	112	Inf	112.1	15819741	Inf
8	114	Inf	114.2	3473886	Inf
9	115	3.57e-06	114.8	32421548	11.90
10	118	2.82e-06	118.3	65652655	13.05
11	127	4.87e-06	127.4	62580901	17.59
12	133	3.47e-05	133.5	10116992	17.42
13	144	8.44e-05	144.0	9378667	22.03

```
> # Ignore the NHANES design feature as a comparison
> stree <- rpms(bpxsy1 ~ bmxbmi + age + white + black + other + indfmpir, data = nhanes)

> stree$partition
   node   n
1    16 256
2    34 203
3    35 139
4    36 130
5    37 156
6    38 125
7    39 246
8   160 158
9   322 157
10  323 160
11   81 329
12   82 149
13   83 105
14   42 232
15   43 281
16   44 487
17   45 219
18   92 205
19   93 126
20   47 199
21   96 164
22   97 130
23   49 123
24   25 255
25   26 176
26   27 133
27   28 482
28   29 321
29   30 124
30   31 189
```

	splits
1	age <= 49.5 & age <= 15.5 & bmxbmi <= 19.35 & bmxbmi <= 16.65
2	age <= 49.5 & age <= 15.5 & bmxbmi <= 19.35 & bmxbmi > 16.65 & age <= 11.5
3	age <= 49.5 & age <= 15.5 & bmxbmi <= 19.35 & bmxbmi > 16.65 & age > 11.5
4	age <= 49.5 & age <= 15.5 & bmxbmi > 19.35 & age <= 11.5 & bmxbmi <= 22.45
5	age <= 49.5 & age <= 15.5 & bmxbmi > 19.35 & age <= 11.5 & bmxbmi > 22.45
6	age <= 49.5 & age <= 15.5 & bmxbmi > 19.35 & age > 11.5 & bmxbmi <= 21.95
7	age <= 49.5 & age <= 15.5 & bmxbmi > 19.35 & age > 11.5 & bmxbmi > 21.95
8	age <= 49.5 & age > 15.5 & bmxbmi <= 27.45 & age <= 34.5 & black <= 0.5 & white <= 0.5 & bmxbmi <= 21.45
9	age <= 49.5 & age > 15.5 & bmxbmi <= 27.45 & age <= 34.5 & black <= 0.5 & white <= 0.5 & bmxbmi > 21.45 & other <= 0.5
10	age <= 49.5 & age > 15.5 & bmxbmi <= 27.45 & age <= 34.5 & black <= 0.5 & white <= 0.5 & bmxbmi > 21.45 & other > 0.5
11	age <= 49.5 & age > 15.5 & bmxbmi <= 27.45 & age <= 34.5 & black <= 0.5 & white > 0.5
12	age <= 49.5 & age > 15.5 & bmxbmi <= 27.45 & age <= 34.5 & black > 0.5 & age <= 21.5
13	age <= 49.5 & age > 15.5 & bmxbmi <= 27.45 & age <= 34.5 & black > 0.5 & age > 21.5
14	age <= 49.5 & age > 15.5 & bmxbmi <= 27.45 & age > 34.5 & bmxbmi <= 23.55
15	age <= 49.5 & age > 15.5 & bmxbmi <= 27.45 & age > 34.5 & bmxbmi > 23.55
16	age <= 49.5 & age > 15.5 & bmxbmi > 27.45 & age <= 36.5 & bmxbmi <= 35.45
17	age <= 49.5 & age > 15.5 & bmxbmi > 27.45 & age <= 36.5 & bmxbmi > 35.45
18	age <= 49.5 & age > 15.5 & bmxbmi > 27.45 & age > 36.5 & white <= 0.5 & bmxbmi <= 33.55
19	age <= 49.5 & age > 15.5 & bmxbmi > 27.45 & age > 36.5 & white <= 0.5 & bmxbmi > 33.55
20	age <= 49.5 & age > 15.5 & bmxbmi > 27.45 & age > 36.5 & white > 0.5
21	age > 49.5 & age <= 62.5 & black <= 0.5 & age <= 57.5 & indfmpir <= 4.135 & age <= 53.5
22	age > 49.5 & age <= 62.5 & black <= 0.5 & age <= 57.5 & indfmpir <= 4.135 & age > 53.5
23	age > 49.5 & age <= 62.5 & black <= 0.5 & age <= 57.5 & indfmpir > 4.135
24	age > 49.5 & age <= 62.5 & black <= 0.5 & age > 57.5
25	age > 49.5 & age <= 62.5 & black > 0.5 & age <= 57.5
26	age > 49.5 & age <= 62.5 & black > 0.5 & age > 57.5
27	age > 49.5 & age > 62.5 & age <= 77.5 & white <= 0.5
28	age > 49.5 & age > 62.5 & age <= 77.5 & white > 0.5
29	age > 49.5 & age > 62.5 & age > 77.5 & indfmpir <= 1.375
30	age > 49.5 & age > 62.5 & age > 77.5 & indfmpir > 1.375

	value	cvar	mean	N_hat	p_se
1	98.1	0.295	98.1	256	8.71
2	101	0.405	100.8	203	9.09
3	105	0.525	104.5	139	8.57
4	104	0.557	103.7	130	8.54
5	107	0.472	106.9	156	8.61
6	107	0.646	106.8	125	9.02
7	110	0.319	110.5	246	8.87
8	107	0.779	107.1	158	11.13
9	109	0.65	109.4	157	10.14
10	112	0.624	111.5	160	10.02
11	112	0.317	112.5	329	10.23
12	113	0.775	112.7	149	10.78
13	117	1.67	116.8	105	13.31
14	113	0.74	113.3	232	13.13
15	118	0.589	117.8	281	12.88
16	117	0.272	116.7	487	11.53
17	121	0.897	120.7	219	14.05
18	121	1.16	121.0	205	15.45
19	126	2.36	126.0	126	17.32
20	119	0.761	119.2	199	12.33
21	125	1.68	124.5	164	16.64
22	127	2.12	126.8	130	16.66
23	119	1.56	118.9	123	13.91
24	127	1.12	127.5	255	16.96
25	129	2.21	129.1	176	19.76
26	136	3.94	136.5	133	22.98
27	135	0.748	135.3	482	19.01
28	131	0.896	130.6	321	16.99
29	148	5.93	147.8	124	27.22
30	140	2.36	140.4	189	21.18