

ch07output

2022-04-19

```
# Code from Chapter 7 of R Companion for Sampling: Design and Analysis by  
# Yan Lu and Sharon L. Lohr  
# All code is presented for educational purposes only and without warranty.
```

```
##### Install the R packages needed for the chapter
```

```
library(survey)
```

```
## Loading required package: grid
```

```
## Loading required package: Matrix
```

```
## Loading required package: survival
```

```
##
```

```
## Attaching package: 'survey'
```

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

```
##
```

```
## dotchart
```

```
library(sampling)
```

```
##
```

```
## Attaching package: 'sampling'
```

```
## The following objects are masked from 'package:survival':
```

```
##
```

```
## cluster, strata
```

```
library(SDAResources)
```

```
##### Selecting a Stratified Two-Stage Sample #####
```

```
# create data frame classeslong (from ch06.R)
```

```
data(classes)
```

```
classeslong<-classes[rep(1:nrow(classes),times=classes$class_size),]
```

```
classeslong$studentid <- sequence(classes$class_size)
```

```
nrow(classeslong)
```

```
## [1] 647
```

```
table(classeslong$class) # check class sizes
```

```
##
```

```
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

```
## 44 33 26 22 76 63 20 44 54 34 46 24 46 100 15
```

```
head(classeslong)
```

```
## class class_size studentid
```

```

## 1      1      44      1
## 1.1    1      44      2
## 1.2    1      44      3
## 1.3    1      44      4
## 1.4    1      44      5
## 1.5    1      44      6

# define strata
classeslong$strat<-rep(3,nrow(classeslong))
classeslong$strat[classeslong$class_size > 40]<-2
classeslong$strat[classeslong$class_size > 70]<-1
# table(classeslong$class,classeslong$strat)
# order data by stratum
classeslong2<-classeslong[order(classeslong$strat),]
# check the stratum construction
table(classeslong2$strat,classeslong2$class_size)

##
##      15  20  22  24  26  33  34  44  46  54  63  76 100
##  1    0   0   0   0   0   0   0   0   0   0   0  76 100
##  2    0   0   0   0   0   0   0  88  92  54  63   0   0
##  3   15  20  22  24  26  33  34   0   0   0   0   0   0

nrow(classeslong2) # number of students in population

## [1] 647

# select a stratified two stage cluster sample,
# stratum: strat
# psu: class
# ssu: studentid
# number of psus selected n =2, size=rep(n=2,3 strata) (srswor)
# number of students selected m_i =3 size=rep(m_i= 3,6 classes) (srswor)
numbersselect<-list(table(classeslong2$strat),rep(2,3),rep(3,6))
numbersselect

## [[1]]
##
##  1  2  3
## 176 297 174
##
## [[2]]
## [1] 2 2 2
##
## [[3]]
## [1] 3 3 3 3 3 3

# select a stratified two-stage cluster sample
set.seed(75745)
tempid<-mstage(classeslong2,stage=list("stratified","cluster","stratified"),
               varnames=list("strat","class","studentid"),
               size=numbersselect, method=list("","srswor","srswor"))

# get data
sample3<-getdata(classeslong2,tempid)[[3]] #3rd stage
sample3$finalweight<-1/sample3$Prob
# check sum of weights, should be close to number of students in population
# (but not exactly equal, since psus not selected with prob proportional to M_i)

```

```
sum(sample3$finalweight)
```

```
## [1] 624
```

```
sample3 # print the sample
```

```
##      class class_size studentid strat ID_unit Prob_ 3 _stage      Prob
## 5.31      5          76         32     1      32      0.03947368 0.03947368
## 5.42      5          76         43     1      43      0.03947368 0.03947368
## 5.61      5          76         62     1      62      0.03947368 0.03947368
## 14.9     14         100         10     1      86      0.03000000 0.03000000
## 14.37    14         100         38     1     114      0.03000000 0.03000000
## 14.79    14         100         80     1     156      0.03000000 0.03000000
## 8.34      8          44         35     2     318      0.06818182 0.02272727
## 8.39      8          44         40     2     323      0.06818182 0.02272727
## 8.28      8          44         29     2     312      0.06818182 0.02272727
## 9.26      9          54         27     2     354      0.05555556 0.01851852
## 9.28      9          54         29     2     356      0.05555556 0.01851852
## 9.38      9          54         39     2     366      0.05555556 0.01851852
## 7.17      7          20         18     3     572      0.15000000 0.04285714
## 7.5       7          20          6     3     560      0.15000000 0.04285714
## 7.10      7          20         11     3     565      0.15000000 0.04285714
## 12.6     12          24          7     3     615      0.12500000 0.03571429
## 12.9     12          24         10     3     618      0.12500000 0.03571429
## 12.16    12          24         17     3     625      0.12500000 0.03571429
```

```
##      finalweight
```

```
## 5.31      25.33333
## 5.42      25.33333
## 5.61      25.33333
## 14.9     33.33333
## 14.37    33.33333
## 14.79    33.33333
## 8.34     44.00000
## 8.39     44.00000
## 8.28     44.00000
## 9.26     54.00000
## 9.28     54.00000
## 9.38     54.00000
## 7.17     23.33333
## 7.5      23.33333
## 7.10     23.33333
## 12.6     28.00000
## 12.9     28.00000
## 12.16    28.00000
```

```
sample1<-getdata(classeslong2,tempid)[[1]] #1st stage
```

```
sample2<-getdata(classeslong2,tempid)[[2]] #2nd stage
```

```
names(sample1)
```

```
## [1] "class"          "class_size"      "studentid"      "strat"
## [5] "ID_unit"        "Prob_ 1 _stage" "Stratum"
```

```
table(sample1$`Prob_ 1 _stage`)
```

```
##
```

```
## 1
```

```
## 647
table(sample2$strat,sample2$`Prob_ 2 _stage`) # Selection probs for psus in strata
```

```
##
##      0.285714285714286 0.333333333333333 1
## 1      0                0 176
## 2      0                98 0
## 3      44               0 0
```

```
table(sample3$class,sample3$`Prob_ 3 _stage`) # Selection probs for ssus in psus
```

```
##
##      0.03 0.0394736842105263 0.0555555555555556 0.0681818181818182 0.125 0.15
## 5      0                3                0                0                0 0
## 7      0                0                0                0                0 3
## 8      0                0                0                3                0 0
## 9      0                0                3                0                0 0
## 12     0                0                0                0                3 0
## 14     3                0                0                0                0 0
```

```
##### Estimating Quantiles #####
```

```
##### Example 7.5
```

```
data(htsrs)
dhtsrs<-svydesign(id = ~1,weights=rep(2000/200,200),fpc=rep(2000,200), data=htsrs)
# cdf treated as step function, gives values in Table 7.1 of SDA
svyquantile(~height, dhtsrs, quantiles=c(0.25,0.5,0.75,0.9), ties = "discrete")
```

```
##      0.25 0.5 0.75 0.9
## height 160 169 176 184
```

```
# interpolated quantiles (usually preferred method)
svyquantile(~height, dhtsrs, quantiles=c(0.25,0.5,0.75,0.9), ties = "rounded")
```

```
##      0.25    0.5 0.75    0.9
## height 159.7 168.75 176 183.4
```

```
##### Examples 7.6 and 9.12
```

```
data(htstrat)
popsize_recode <- c('F' = 1000, 'M' = 1000)
# create a new variable popsize for population size
htstrat$popsize<-popsize_recode[htstrat$gender]
head(as.data.frame(htstrat))
```

```
##      rn height gender popsize
## 1 201    166     F    1000
## 2 965    163     F    1000
## 3 490    166     F    1000
## 4 249    155     F    1000
## 5 260    154     F    1000
## 6 324    160     F    1000
```

```
# design object
# svydesign calculates the weights here from the fpc argument
dhtstrat<-svydesign(id = ~1, strata = ~gender, fpc = ~popsize,
```

```

      data = htstrat)
# ties = "discrete" gives values in Table 7.1 of SDA
svyquantile(~height, dhtstrat, c(0.25,0.5,0.75,0.9), ties = "discrete")

##          0.25 0.5 0.75 0.9
## height  161 168  177 182

# ties = "rounded" gives values in Example 9.12 of SDA
svyquantile(~height, dhtstrat, c(0.25,0.5,0.75,0.9), ties = "rounded",
            ci=TRUE, interval.type = "Wald")

## $quantiles
##          0.25          0.5          0.75          0.9
## height 160.7143 167.5556 176.625 181.5
##
## $CIs
## , , height
##
##          0.25          0.5          0.75          0.9
## (lower 159.3556 165.8078 173.3572 178.7176
## upper) 162.0247 170.0942 178.5439 190.1679

##### Computing Estimates from Stratified Multistage Samples #####

#### Example 7.9

data(nhanes)
nrow(nhanes) #9971

## [1] 9971

names(nhanes)

## [1] "sdmvstra" "sdmvpsu" "wtint2yr" "wtmec2yr" "ridstatr" "ridageyr"
## [7] "ridagemn" "riagendr" "ridreth3" "dmdeduc2" "dmdfmsiz" "indfmpir"
## [13] "bmxwt" "bmxht" "bmxbmi" "bmxwaist" "bmxleg" "bmxarml"
## [19] "bmxarmc" "bmdavsad" "lbxtc" "bpxpls" "sbp" "dbp"
## [25] "bpread"

# count number of observations with missing value for ridageyr, bmx bmi
sum(is.na(nhanes$ridageyr)) # ridageyr gives age in years

## [1] 0

sum(is.na(nhanes$bmx bmi)) # bmx bmi gives BMI

## [1] 1215

# define age20d and bmi30
nhanes$age20d<-rep(0,nrow(nhanes))
nhanes$age20d[nhanes$ridageyr >=20 & !is.na(nhanes$bmx bmi)]<-1
nhanes$bmi30<-nhanes$bmx bmi
nhanes$bmi30[nhanes$bmx bmi>30]<-1
nhanes$bmi30[nhanes$bmx bmi<=30]<-0
nhanes$bmi30<-factor(nhanes$bmi30) # set bmi30 as a categorical variable
# check missing value counts for new variables
sum(is.na(nhanes$age20d))

## [1] 0

```

```
sum(is.na(nhanes$bmi30))
```

```
## [1] 1215
```

```
sum(nhanes$age20d) # how many records in domain?
```

```
## [1] 5406
```

```
head(nhanes)
```

```
##   sdmvstra sdmvpsu  wtint2yr  wtmec2yr ridstatr ridageyr ridagemn riagendr
## 1      125      1 134671.37 135629.51      2      62      NA      1
## 2      125      1 24328.56 25282.43      2      53      NA      1
## 3      131      1 12400.01 12575.84      2      78      NA      1
## 4      131      1 102718.00 102078.63      2      56      NA      2
## 5      126      2 17627.67 18234.74      2      42      NA      2
## 6      128      1 11252.31 10878.68      2      72      NA      2
##   ridreth3 dmdeduc2 dmdfmsiz indfmpir bmxwt bmxht bmxbmi bmxwaist bmxleg
## 1         3         5         2     4.39  94.8 184.5  27.8  101.1  43.3
## 2         3         3         1     1.32  90.4 171.4  30.8  107.9  38.0
## 3         3         3         2     1.51  83.4 170.1  28.8  116.5  35.6
## 4         3         5         1     5.00 109.8 160.9  42.4  110.1  38.5
## 5         4         4         5     1.23  55.2 164.9  20.3   80.4  37.4
## 6         1         2         5     2.82  64.4 150.0  28.6   92.9  34.4
##   bmxarml bmxarmc bmdavsad lbxtc bpxpls sbp dbp bpread age20d bmi30
## 1    43.6   35.9   22.8   173    76 120  63     2     1     0
## 2    40.0   33.2   27.3   265    72 137  85     2     1     1
## 3    37.0   31.0   26.6   229    56 134  45     2     1     0
## 4    37.7   38.3   25.1   174    78 135  69     2     1     1
## 5    36.0   27.2    NA   204    76 106  55     2     1     0
## 6    33.5   31.4   23.1   190    64 121  59     2     1     0
```

```
# stratified cluster design
```

```
d0709 <- svydesign(id = ~sdmvpsu, strata=~sdmvstra, weights=~wtmec2yr,
                 nest=TRUE, data = nhanes)
```

```
# domain estimation, age20+
```

```
d0709sub<-subset(d0709, age20d ==1)
```

```
d0709sub
```

```
## Stratified 1 - level Cluster Sampling design (with replacement)
```

```
## With (30) clusters.
```

```
## subset(d0709, age20d == 1)
```

```
# Request means and design effects
```

```
nhmeans<-svymeans(~bmxbmi+bmi30, d0709sub, deff=TRUE)
```

```
degf(d0709sub)
```

```
## [1] 15
```

```
nhmeans
```

```
##           mean           SE  DEff
## bmxbmi 29.389101 0.253197 7.1248
## bmi300 0.607775 0.015856 5.7003
## bmi301 0.392225 0.015856 5.7003
```

```
confint(nhmeans,df=degf(d0709sub))
```

```
##           2.5 %      97.5 %
```

```

## bmxbmi 28.8494243 29.9287768
## bmi300 0.5739798 0.6415707
## bmi301 0.3584293 0.4260202

# Find quantiles
svyquantile(~bmxbmi, d0709sub, quantiles=c(0.05,0.25,0.5,0.75,0.95),
            ties = "rounded",ci=TRUE, interval.type="Wald")

## $quantiles
##           0.05      0.25      0.5      0.75      0.95
## bmxbmi 20.29893 24.35349 28.2349 33.06615 42.64092
##
## $CIs
## , , bmxbmi
##
##           0.05      0.25      0.5      0.75      0.95
## (lower 19.83403 23.92667 27.55465 32.35400 41.91584
## upper) 20.70609 24.84391 28.91359 33.64129 43.47766

##### Univariate Plots from Complex Surveys #####

#### Examples 7.10, 7.11, and 7.12

# histogram and smoothed density function
data(htstrat)
# set graphics parameters, 1*2 plots, axis labels horizontal
par(mfrow=c(1,2),las=1,mar=c(2.1,4.1,2.1,0.3))
# Histogram overlaid with kernel density curve (without weight information)
# Displays the sample values, but does not estimate population histogram
# freq=FALSE changes the vertical axis to density
# breaks tell how many breakpoints to use
hist(htstrat$height,main="Without weights", xlab = "Height (cm)",
     breaks = 10, col="gray90", freq=FALSE, xlim=c(140,200), ylim=c(0,0.045))
# overlaid with kernel density curve
lines(density(htstrat$height),lty=1,lwd=2)

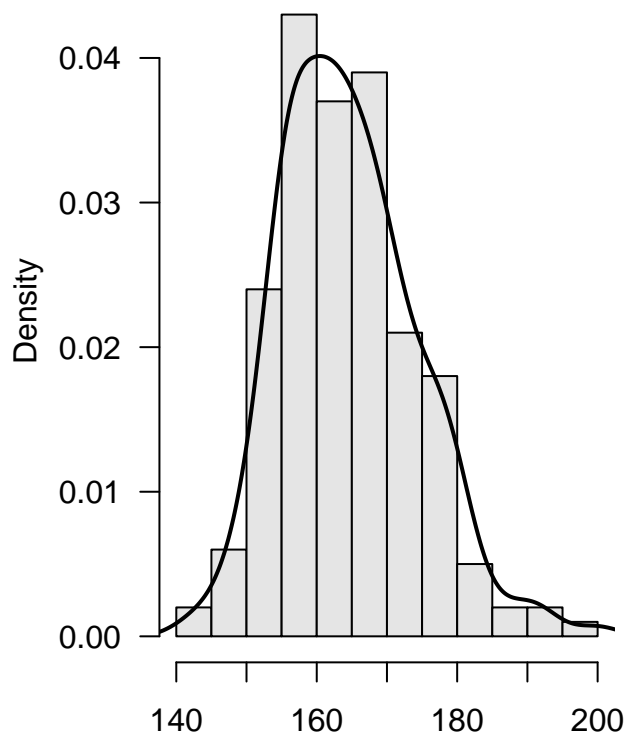
# Histogram (with weight information)
# create survey design object, weights calculated from fpc here
d0710 <- svydesign(id = ~1, strata = ~gender, fpc = c(rep(1000,160),rep(1000,40)),
                 data = htstrat)
d0710

## Stratified Independent Sampling design
## svydesign(id = ~1, strata = ~gender, fpc = c(rep(1000, 160),
##      rep(1000, 40)), data = htstrat)

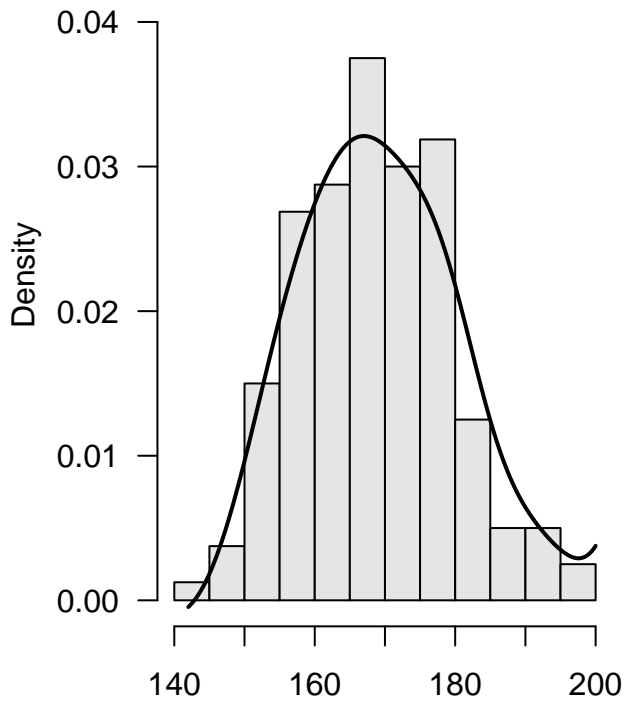
svyhist(~height,d0710, main="With weights",xlab = "Height (cm)",
        breaks = 10, col="gray90", freq=FALSE,xlim=c(140,200), ylim=c(0,0.045))
dens1<-svsmooth(~height,d0710,bandwidth=5)
lines(dens1,lwd=2) # draw the density line

```

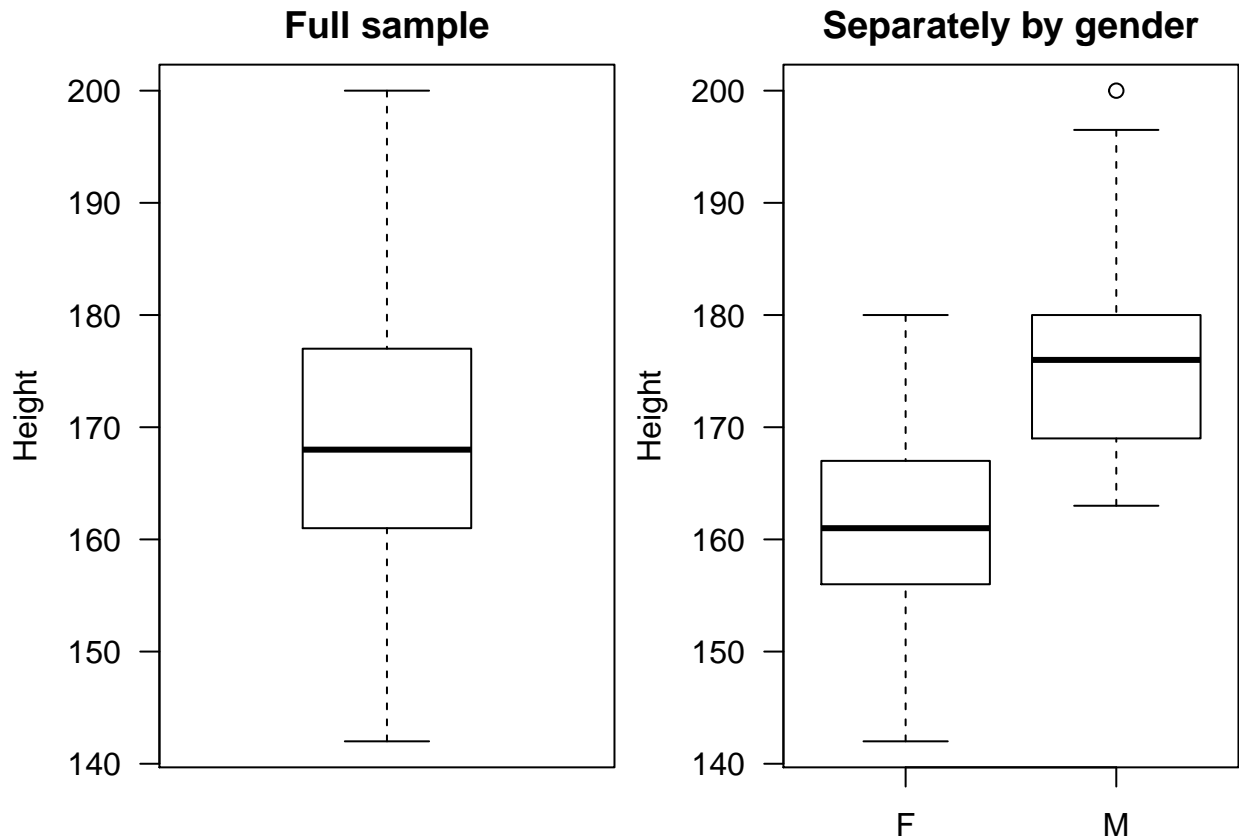
Without weights



With weights



```
# boxplot
par(mfrow=c(1,2),las=1,mar=c(2.1,4.1,2.1,0.3))
# boxplot (with weight information)
svyboxplot(height~1,d0710,ylab="Height",xlab=" ", main="Full sample")
svyboxplot(height~gender,d0710,ylab="Height",xlab="Gender",
           main="Separately by gender")
```

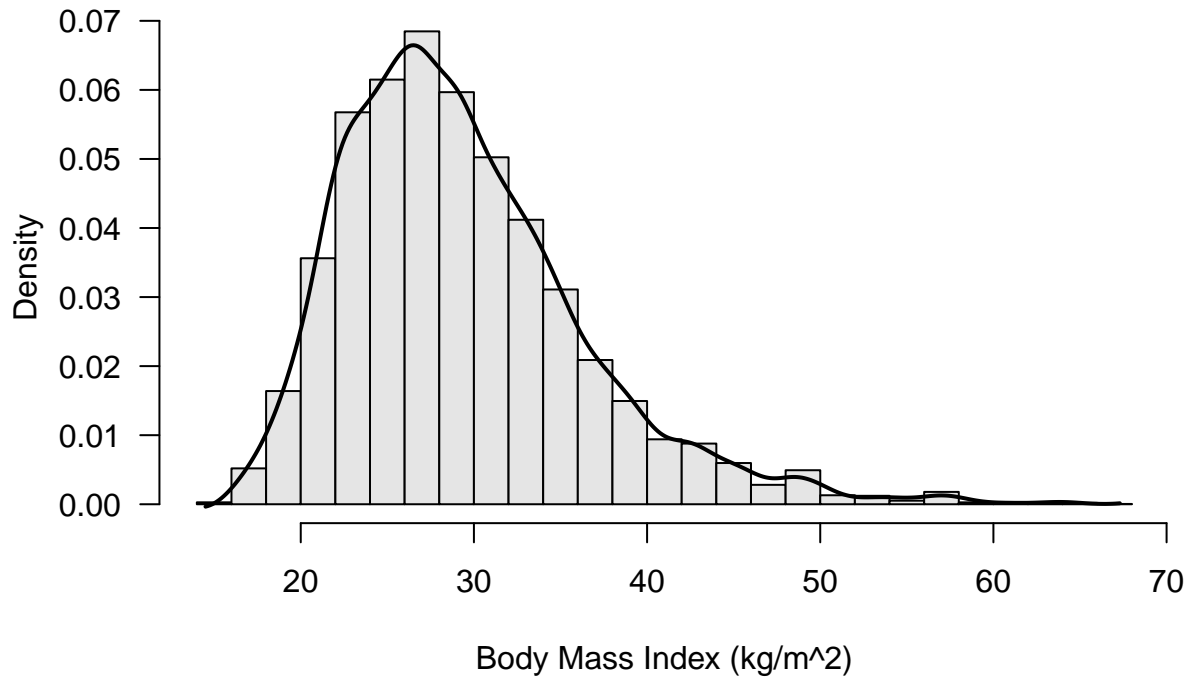



```

# histograms for domains
# Restore graphics settings
par(mfrow=c(1,1),las=1,mar=c(5.1, 4.1, 4.1, 2.1))
svyhist(~bmx bmi,d0709sub, main="Histogram of body mass index for adults",
        breaks = 30, col="gray90",xlab = "Body Mass Index (kg/m^2)")
dens2<-svsmooth(~bmx bmi,d0709sub)
lines(dens2,lwd=2)

```

Histogram of body mass index for adults



```
##### Scatterplots from Complex Surveys #####
```

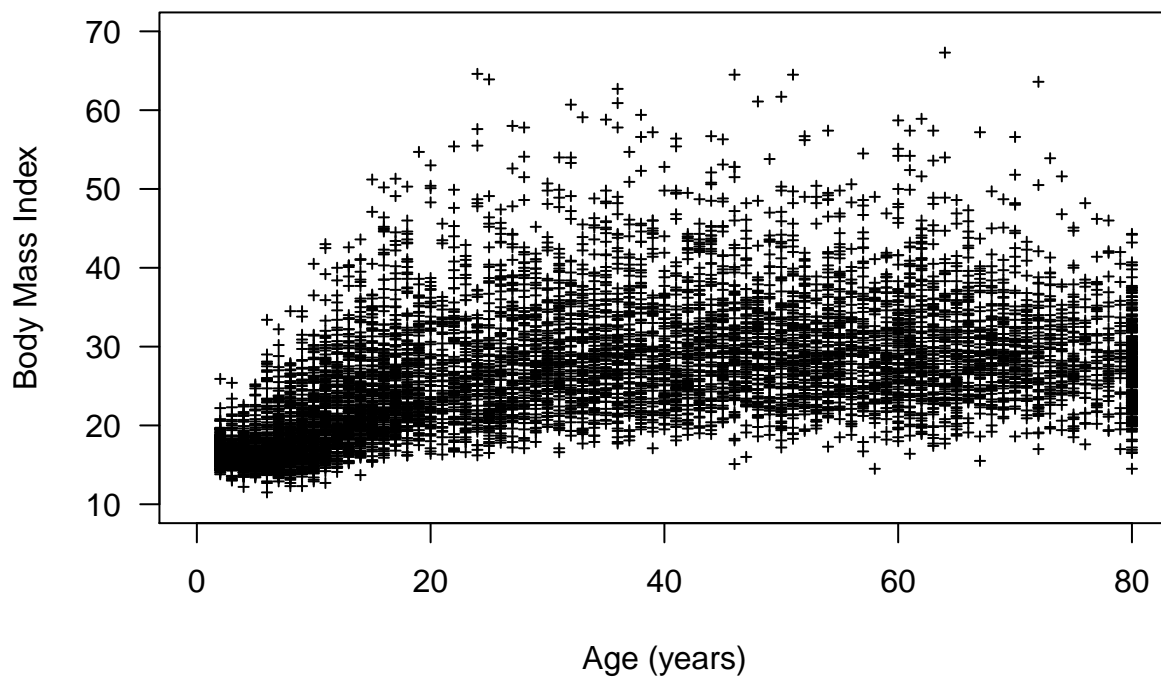
```
### Unweighted plots
```

```
# scatterplot without weights
```

```
par(las=1) # make tick mark labels horizontal
```

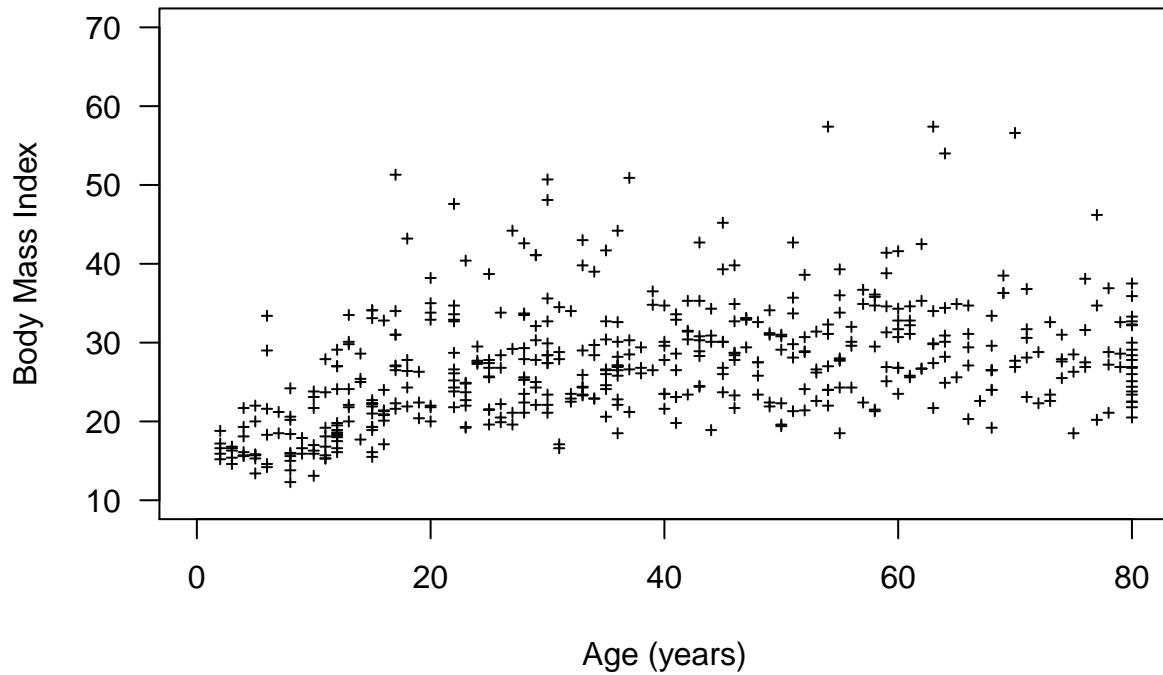
```
plot(nhanes$ridageyr, nhanes$bmx bmi, xlab="Age (years)", ylab="Body Mass Index",  
     main="Scatterplot without weights", pch=3, cex=0.5,  
     ylim=c(10, 70), xlim=c(0, 80))
```

Scatterplot without weights



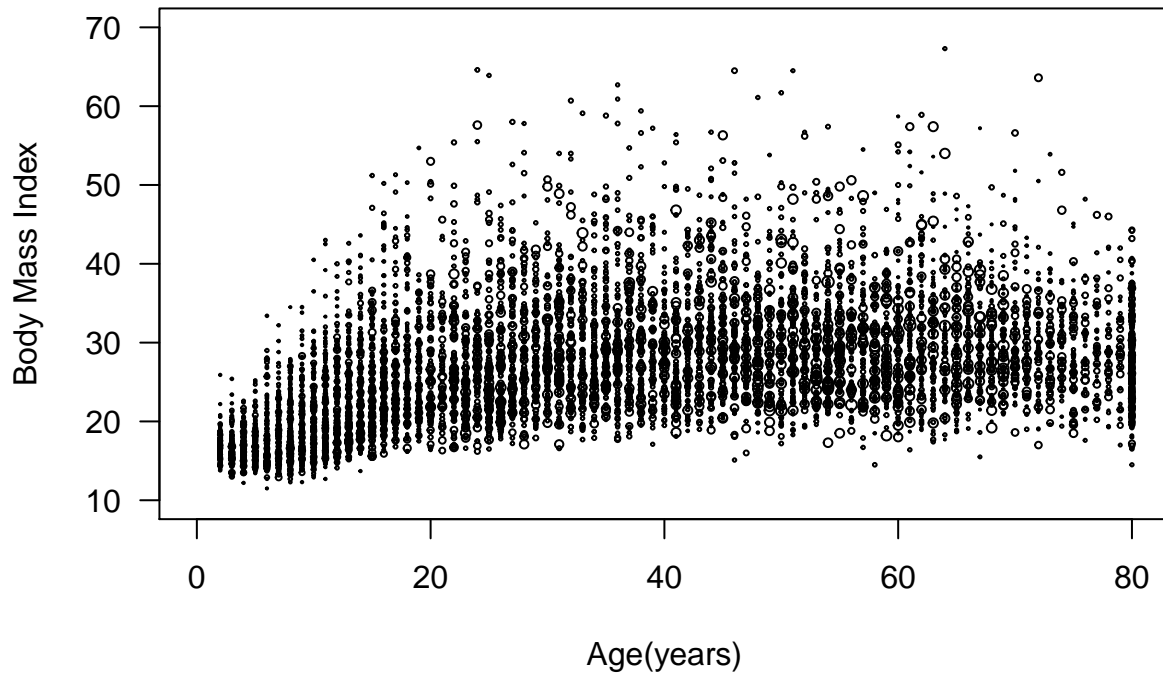
```
### Plot subsample of data
# select subsample with probability proportional to weights
set.seed(2847654)
subsamp<-sample(1:nrow(nhanes),500,replace=TRUE,prob=nhanes$wtmec2yr)
par(las=1) # make tick mark labels horizontal
plot(nhanes$ridageyr[subsamp],nhanes$bmxbmi[subsamp],
      xlab="Age (years)",ylab="Body Mass Index",
      main="Scatterplot of pps subsample",pch=3,cex=0.5,
      ylim=c(10,70),xlim=c(0,80))
```

Scatterplot of pps subsample



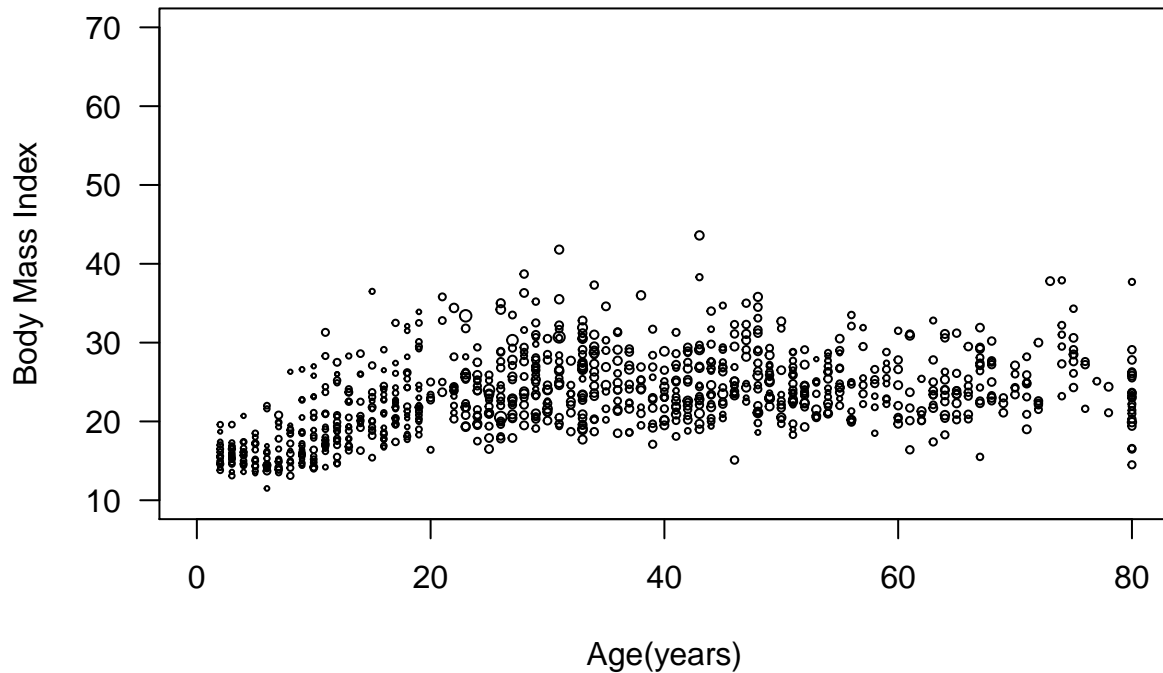
```
### Bubble plots
par(las=1) # make tick mark labels horizontal
svyplot(bmxbmi~ridageyr, design=d0709, style="bubble", inches=0.03,
        xlab="Age(years)", ylab="Body Mass Index", xlim=c(0,80), ylim=c(10,70),
        main="Weighted bubble plot of BMI versus age")
```

Weighted bubble plot of BMI versus age



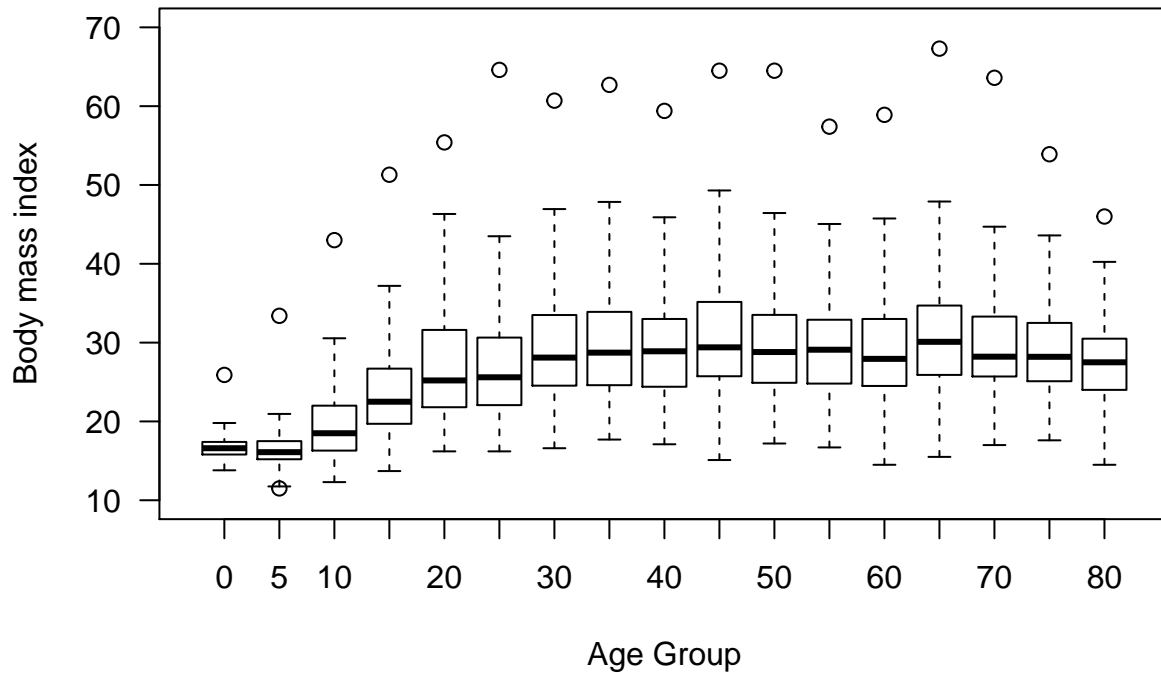
```
### Plot data for a domain
# define subset
d0709subA<-subset(d0709, ridreth3==6)
par(las=1) # make tick mark labels horizontal
svyplot(bmxbmi~ridageyr, design=d0709subA, style="bubble", inches = 0.03,
        xlab="Age(years)", ylab="Body Mass Index", xlim=c(0,80), ylim=c(10,70),
        main="Weighted bubble plot of BMI versus age for Asian Americans")
```

Weighted bubble plot of BMI versus age for Asian Americans



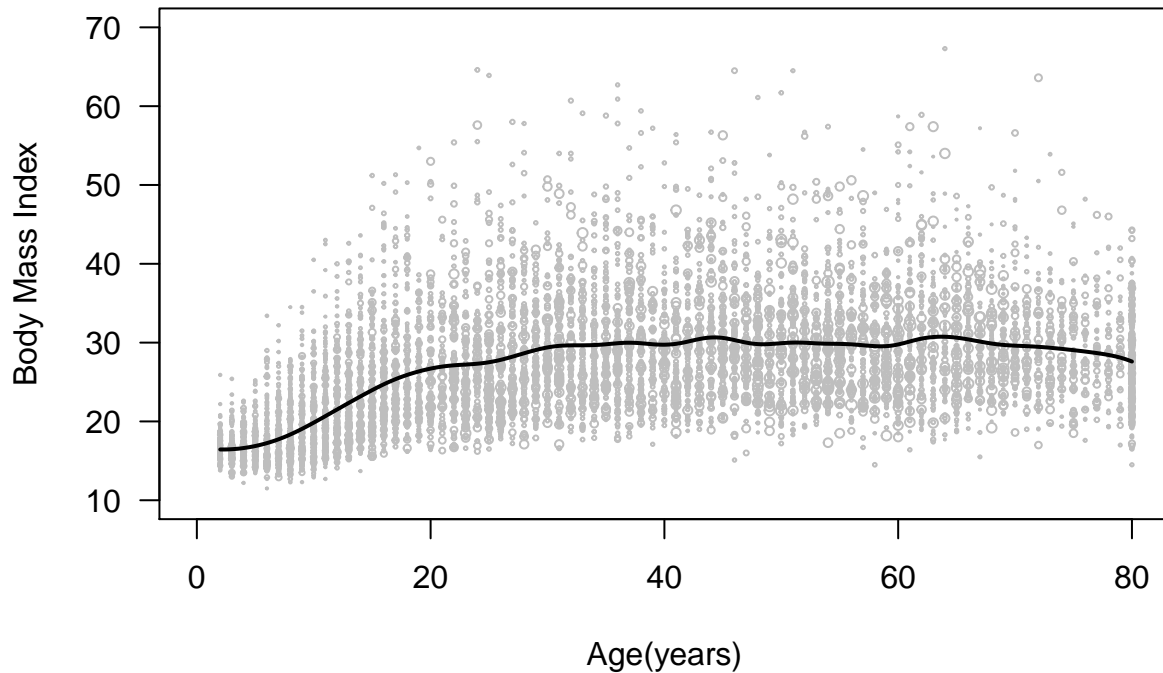
```
### Side-by-side boxplots
# include agegroup in the data frame
nhanes$agegroup<-5*round(nhanes$ridageyr/5)
d0709 <- svydesign(id = ~sdmvpsu, strata = ~ sdmvstra, nest=TRUE,
                 weights=~wtmec2yr, data = nhanes)
par(las=1) # make tick mark labels horizontal
svyboxplot(bmxbmi~factor(agegroup),d0709,ylab="Body mass index",xlab="Age Group",
           ylim=c(10,70),main="Side-by-side boxplots of BMI for age groups")
```

Side-by-side boxplots of BMI for age groups



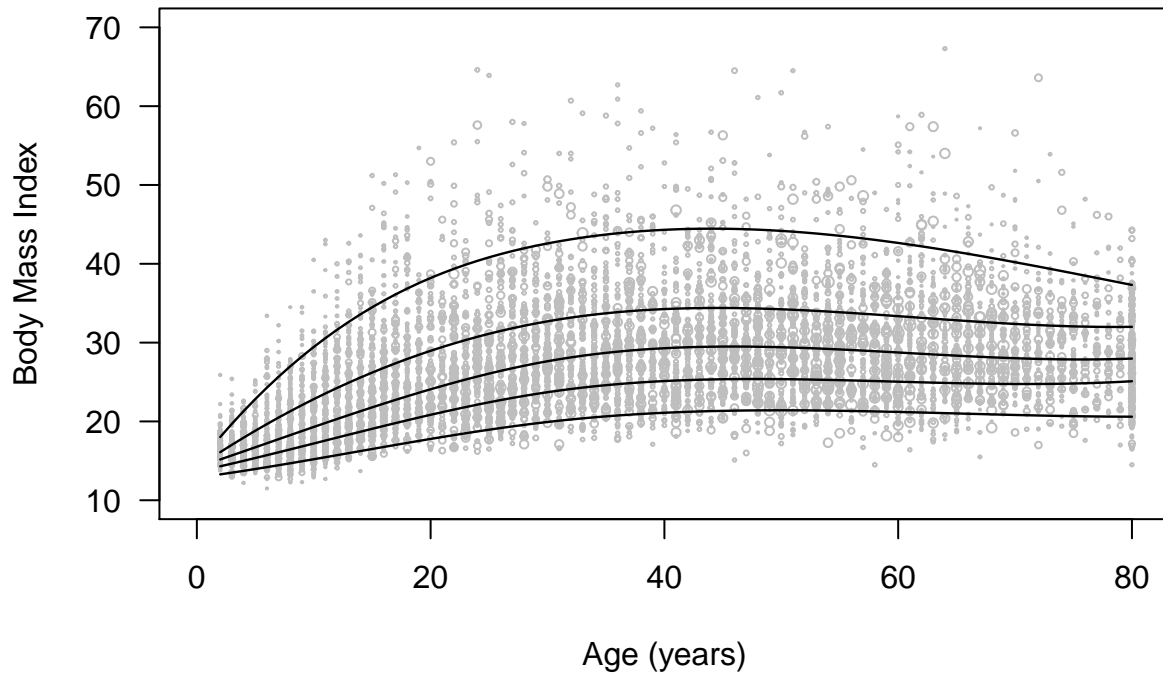
```
### Smoothed trend line for mean
# Smoothed trend line with bubble plot of BMI versus age
# plot data bmx bmi~ridageyr
par(las=1) # make tick mark labels horizontal
svyplot(bmx bmi~ridageyr, design=d0709, style="bubble", basecol="gray", inches=0.03,
        xlab="Age(years)", ylab="Body Mass Index", xlim=c(0,80), ylim=c(10,70),
        main="Smoothed trend line with bubble plot of BMI versus age")
# plot smoothing trend line
# library(KernSmooth) # install and load the package if not already done
smth<-svsmooth(bmx bmi~ridageyr,d0709)
lines(smth,lwd=2)
```

Smoothed trend line with bubble plot of BMI versus age



```
### Smoothed trend lines for quantiles
# Smoothed quantile trend lines with bubble plot of BMI versus age
# library(quantreg) # install and load the package if not already done
# plot data bmx bmi~ridageyr
par(las=1) # make tick mark labels horizontal
svyplot(bmx bmi~ridageyr, design=d0709, style="bubble", basecol="gray", inches=0.03,
        xlab="Age (years)", ylab="Body Mass Index", xlim=c(0,80), ylim=c(10,70),
        main="Smoothed quantile trend lines")
# plot smoothed trend lines for quantiles
taus<-c(.05,.25,.5,.75,.95)
for (i in 1:length(taus)) {
  qsmth<-svsmooth(bmx bmi~ridageyr,d0709, quantile=taus[i],method="quantreg")
  lines(qsmth,lwd=1.2)
}
```


Smoothed quantile trend lines

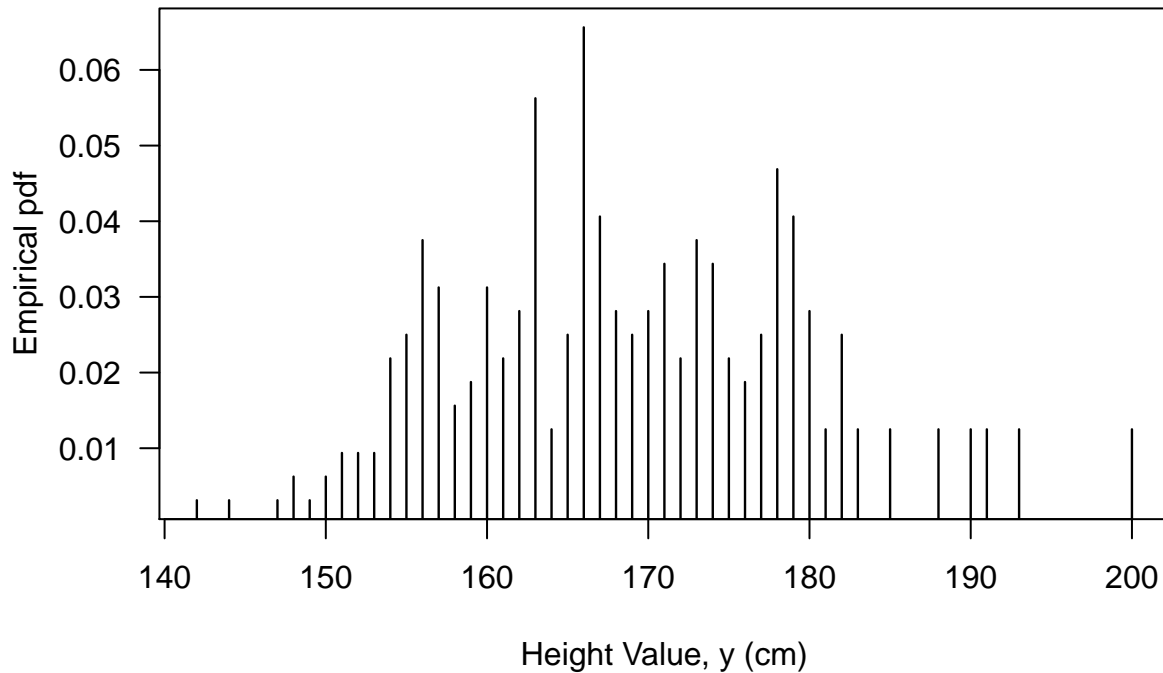


```
##### Empirical Probability Mass and Distribution Functions #####
```

```
##### Example 7.5
```

```
# Empirical pmf for stratified sample of heights  
# define sampling weight  
htstrat$sampwt <- 1000/sum(htstrat$gender=="F")  
htstrat$sampwt[htstrat$gender=="M"] <- 1000/sum(htstrat$gender=="M")  
# use function emppmf to calculate pmf  
strresult <- emppmf(htstrat$height,htstrat$sampwt)  
# plot  
par(las=1)  
plot(strresult$vals, strresult$epmf,type="h",xlab="Height Value, y (cm)",  
      ylab="Empirical pdf",lwd=1.2,  
      main="Empirical pdf for stratified sample of heights (weighted)")
```

Empirical pdf for stratified sample of heights (weighted)



```
# Empirical cdf of height for data htpop, and for data htstrat, with and without weights
# data(htstrat)
# Recall that
d0710 <- svydesign(id = ~1, strata = ~gender, fpc = c(rep(1000,160),rep(1000,40)),
                 data = htstrat)
cdf.weighted<-svycdf(~height, d0710)
cdf.weighted
```

```
## Weighted ECDFs: svycdf(~height, d0710)
```

```
## evaluate the function for height 144
```

```
cdf.weighted[[1]](144)
```

```
## [1] 0.00625
```

```
## compare to population and unweighted sample ecdfs.
```

```
cdf.pop<-ecdf(htpop$height) # ecdf for population
```

```
cdf.samp<-ecdf(htstrat$height) # unweighted ecdf of sample
```

```
par(las=1,mar=c(5.1,4.1,2.1,2.1))
```

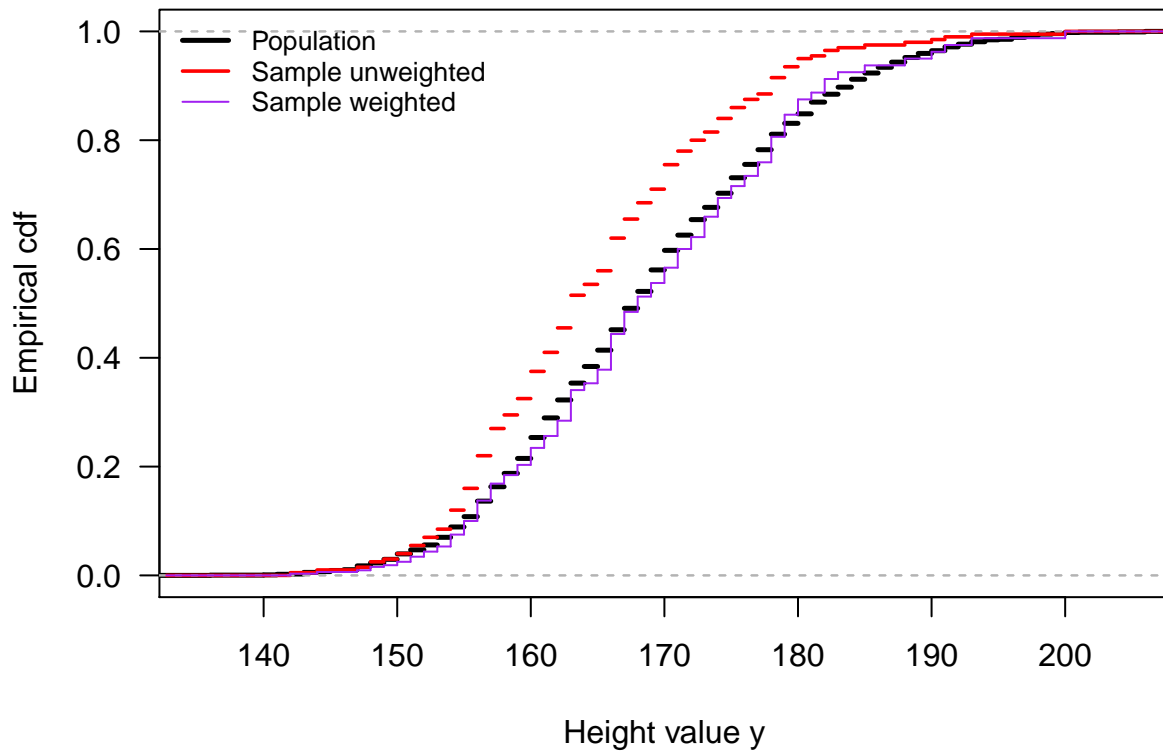
```
plot(cdf.pop, do.points = FALSE,
     xlab="Height value y",ylab="Empirical cdf",xlim=c(135,205),lwd=2.5,
     main="Empirical cdfs for population and sample")
```

```
lines(cdf.samp, col="red", do.points = FALSE, lwd=1.8)
```

```
lines(cdf.weighted[[1]], do.points = FALSE, col = "purple",lwd=1)
```

```
legend("topleft", legend=c("Population", "Sample unweighted", "Sample weighted"),
     col=c("black", "red", "purple"),lwd=c(2.5,1.8,1),cex=0.8,bty="n")
```

Empirical cdfs for population and sample



```
##### ADDITIONAL CODE NOT IN BOOK #####
```

```
##### Graphing Complex Survey Data with ggplot2 Functions #####
```

```
### Install the ggplot2 package if you have not already done so
```

```
library(ggplot2)
```

```
##
```

```
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:SDAResources':
```

```
##
```

```
## seals
```

```
data(nhanes)
```

```
# scatterplot without weights
```

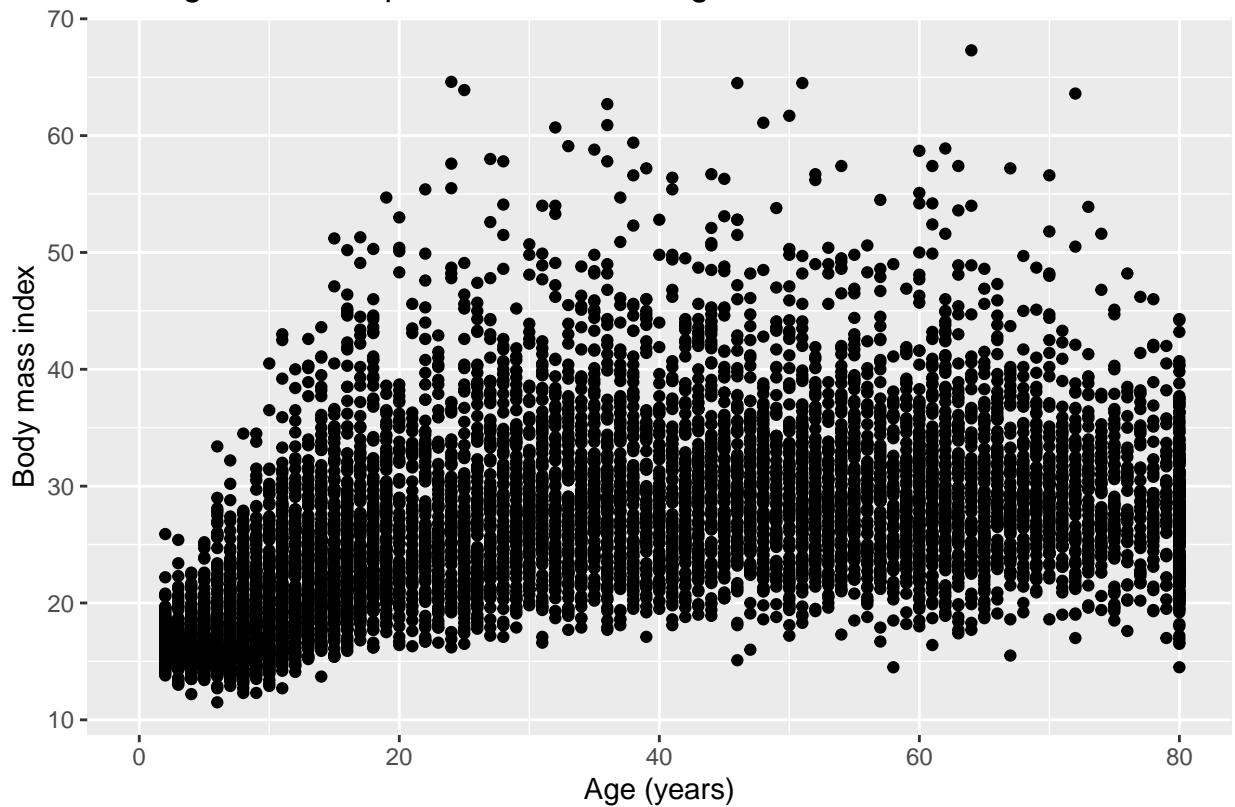
```
plot1 <- ggplot(nhanes, aes(x=ridageyr, y=bmxbmi)) + geom_point()
```

```
# print plot with title and axis labels
```

```
plot1 + labs(title = "Unweighted scatterplot of BMI versus age", x = "Age (years)", y = "Body mass index")
```

```
## Warning: Removed 1215 rows containing missing values (geom_point).
```

Unweighted scatterplot of BMI versus age

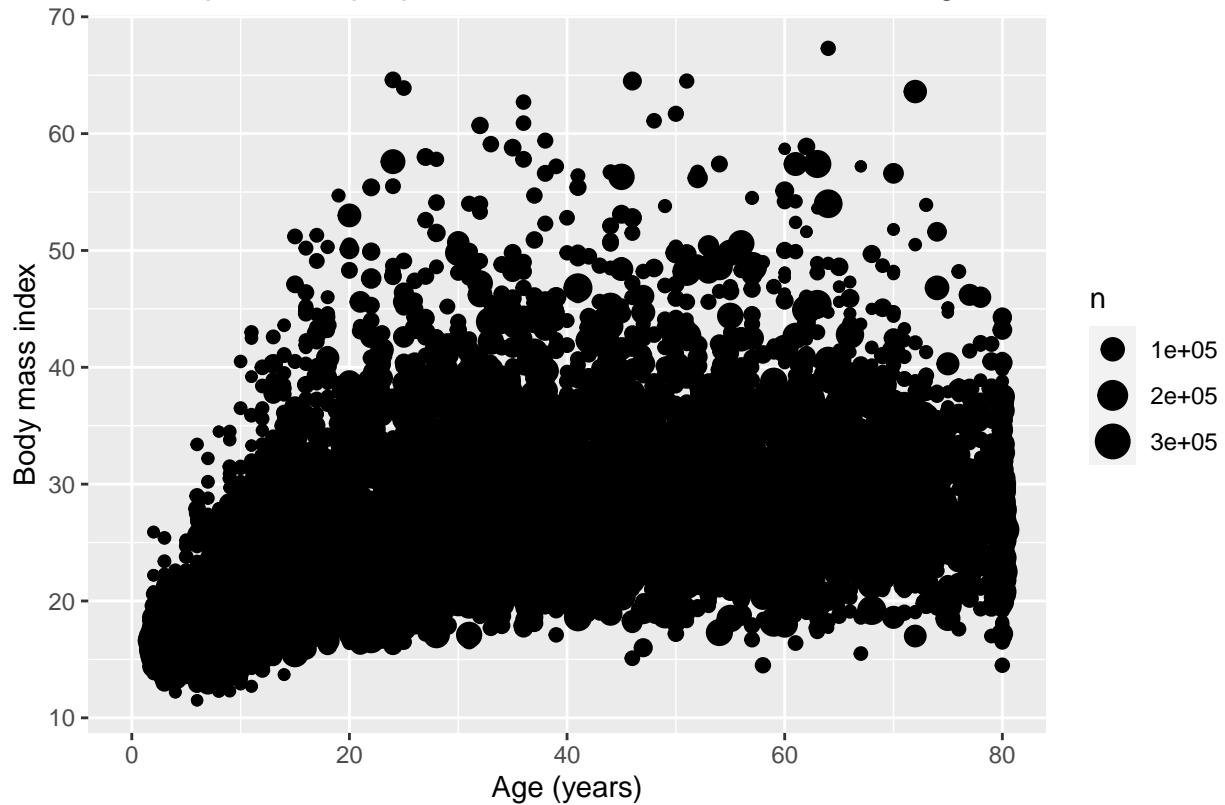


```
# ggplot scatterplot weighted  
# initial plot without weights  
plot2 <- ggplot(nhanes, aes(x = ridageyr, y = bmx bmi)) + geom_point()  
# adding title, x, and y labels  
plot2 <- plot2 + labs(title="Bubble plot, size proportional to sum of observation weight",  
  x = "Age (years)", y = "Body mass index")  
# find the sum of weights at each distinct (x,y) pair using stat_sum  
g2 <- plot2 + stat_sum(aes(weight = wt mec2yr, size = ..n..))  
print(g2)
```

```
## Warning: Removed 1215 rows containing non-finite values (stat_sum).
```

```
## Warning: Removed 1215 rows containing missing values (geom_point).
```

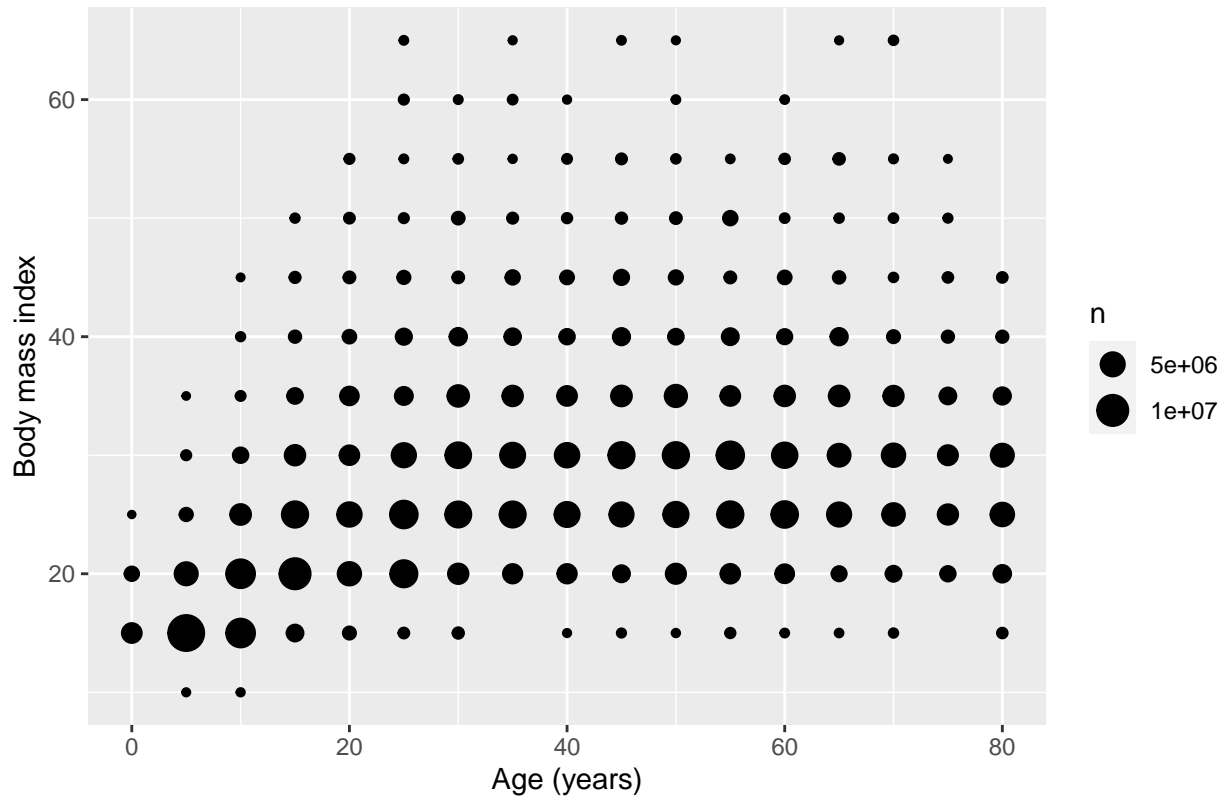
Bubble plot, size proportional to sum of observation weight



```
# binned circle plot
mround <- function(x,base){
  base*round(x/base)
}
nhanes$bmigroup<-mround(nhanes$bmxbmi,5)
nhanes$agegroup<-mround(nhanes$ridageyr,5)
circleage<-nhanes[order(nhanes$agegroup),]
plot3 <- ggplot(circleage, aes(x = agegroup, y = bmigroup)) + labs(title="Binned circle plot",
  x = "Age (years)", y = "Body mass index")
g3 <- plot3 + stat_sum(aes(weight = wtmecl2yr, size = ..n..))
print(g3)
```

```
## Warning: Removed 1215 rows containing non-finite values (stat_sum).
```

Binned circle plot



```
# side-by-side boxplots
plot4 <- ggplot(circlege, aes(x = factor(agegroup), y = bmx bmi))
plot4 <- plot4 + stat_sum(aes(weight=wtmec2yr, size = ..n..)) +
  labs(title="Boxplots", x ="Age (years)", y = "Body mass index")
g4 <- plot4 + stat_summary(fun=mean, geom="point", shape=23, size=4) +geom_boxplot()
print(g4)
```

```
## Warning: Removed 1215 rows containing non-finite values (stat_sum).
```

```
## Warning: Removed 1215 rows containing non-finite values (stat_summary).
```

```
## Warning: Removed 1215 rows containing non-finite values (stat_boxplot).
```

