Reproduction of Analyses in Lohr (1999) using the `survey` package

Tobias Verbeke

2008-09-25

Contents

1 Introduction

The Introduction chapter does not contain any numerical examples demonstrating survey methodology. Before reproducing the analyses of the following chapters, we load the `SDA` package

```
> library(SDA)
```

The `survey` package is loaded as well as it was specified as a dependency of the `SDA` package.

2 Simple Probability Samples

3 Ratio and Regression Estimation

3.1 Ratio Estimation

```
> agsrsDesign <- svydesign(ids = ~1, weights = ~1,
+     data = agsrs)
```

```
> svyratio(numerator = ~acres92, denominator = ~acres87,
+     design = agsrsDesign)

Ratio estimator: svyratio.survey.design2(numerator = ~acres92, denominator = ~acres87,
    design = agsrsDesign)
Ratios=
          acres87
acres92 0.9865652
SEs=
            acres87
acres92 0.006053015
```

```
> seedlings <- data.frame(tree = 1:10, x = c(1, 0,
+     8, 2, 76, 60, 25, 2, 1, 31), y = c(0, 0, 1, 2,
+     10, 15, 3, 2, 1, 27))
> names(seedlings) <- c("tree", "x", "y")
```

3.2 Regression Estimation

```
> pf <- data.frame(photo = c(10, 12, 7, 13, 13, 6,
+     17, 16, 15, 10, 14, 12, 10, 5, 12, 10, 10, 9,
+     6, 11, 7, 9, 11, 10, 10), field = c(15, 14, 9,
+     14, 8, 5, 18, 15, 13, 15, 11, 15, 12, 8, 13,
+     9, 11, 12, 9, 12, 13, 11, 10, 9, 8))
```

3.3 Estimation in Domains

3.4 Models for Ratio and Regression Estimation

```
> recacr87 <- agsrs$acres87
> recacr87[recacr87 > 0] <- 1/recacr87[recacr87 > 0]
> model1 <- lm(acres92 ~ 0 + acres87, weights = recacr87,
+     data = agsrs)
> summary(model1)

Call:
lm(formula = acres92 ~ 0 + acres87, data = agsrs, weights = recacr87)
```

```
> plot(I(acres92/10^6) ~ I(acres87/10^6), xlab = "Millions of Acres Devoted to Farms (
+     ylab = "Millions of Acres Devoted to Farms (1992)",
+     data = agsrs)
> abline(lm(I(acres92/10^6) ~ 0 + I(acres87/10^6),
+     data = agsrs), col = "red", lwd = 2)
```

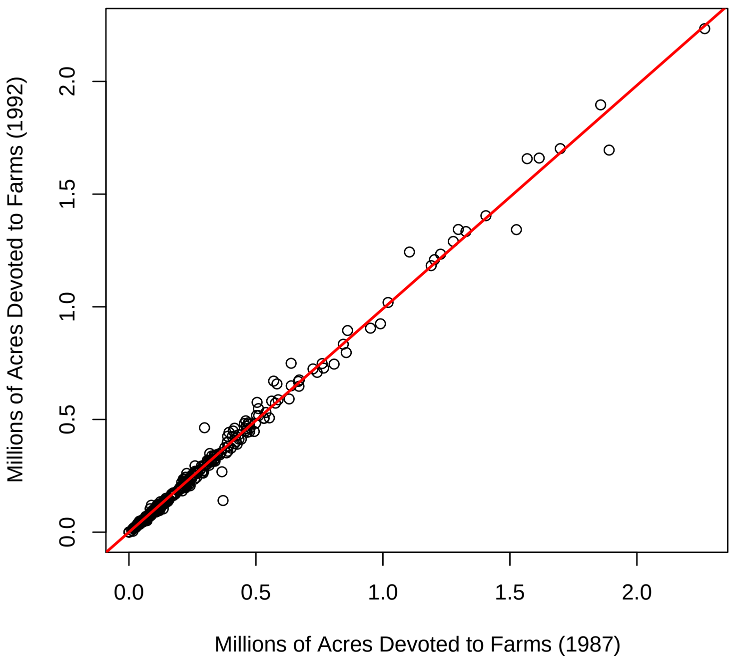


Figure 1: Figure 3.1, p. 64

```
> plot(y ~ x, data = seedlings, xlab = "Seedlings Alive (March 1992)",
+     ylab = "Seedlings That Survived (February 1994)")
```

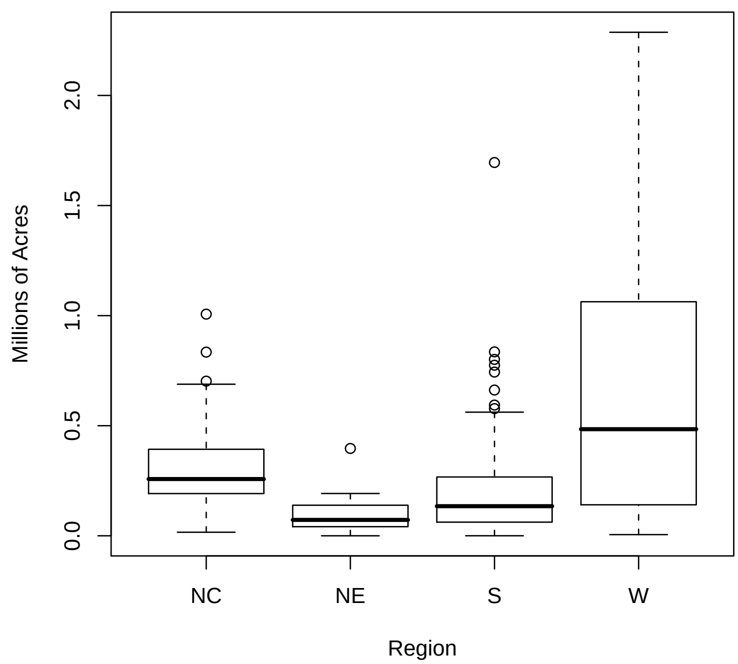


Figure 2: Figure 3.4, p. 73

```
Residuals:
     Min       1Q   Median       3Q      Max
-369.878  -22.090   -5.736   10.764  311.713

Coefficients:
        Estimate Std. Error t value Pr(>|t|)
acres87 0.986565   0.004844   203.7   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 46.1 on 298 degrees of freedom
```

```
> wtresid <- resid(model1)/sqrt(agsrs$acres87)
> plot(wtresid ~ I(agsrs$acres87/10^6), xlab = "Millions of Acres Devoted to Farms (19
+     ylab = "Weighted Residuals")
```

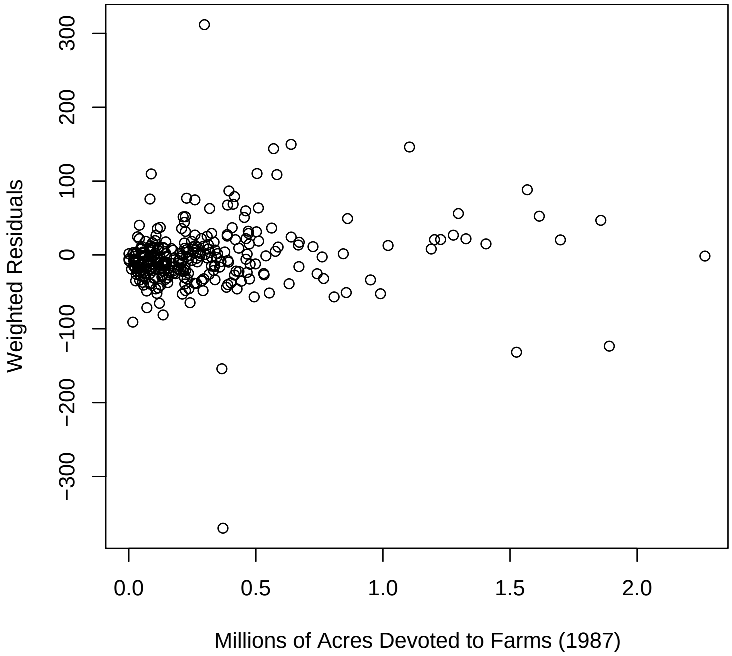


Figure 3: Figure 3.6, p. 85

```
Multiple R-squared: 0.9929,       Adjusted R-squared: 0.9928
F-statistic: 4.149e+04 on 1 and 298 DF,  p-value: < 2.2e-16
```

4 Stratified Sampling

```
> boxplot(acres92/10^6 ~ region, xlab = "Region", ylab = "Millions of Acres",
+     data = agstrat)
```

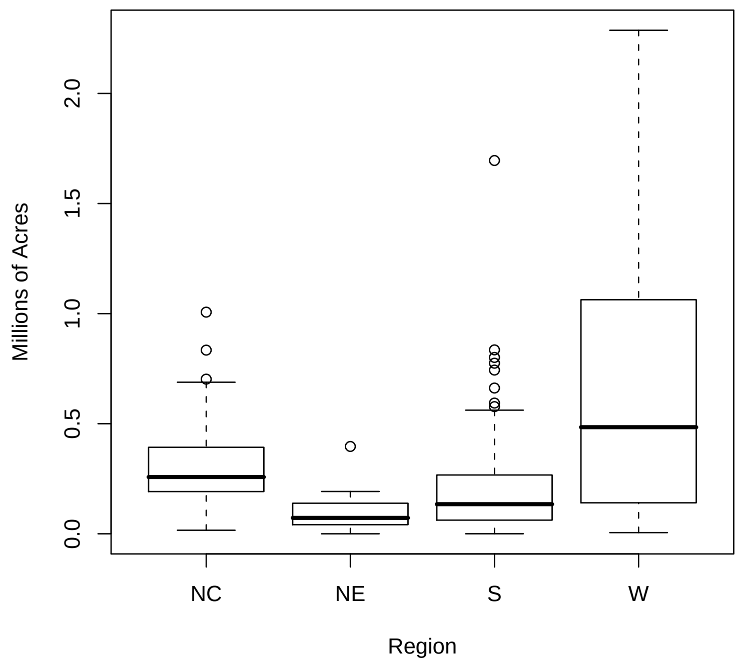


Figure 4: Figure 4.1, p. 97

5 Cluster Sampling with Equal Probabilities

5.1 Notation for Cluster Sampling

No analyses contained in this section.

5.2 One-Stage Cluster Sampling

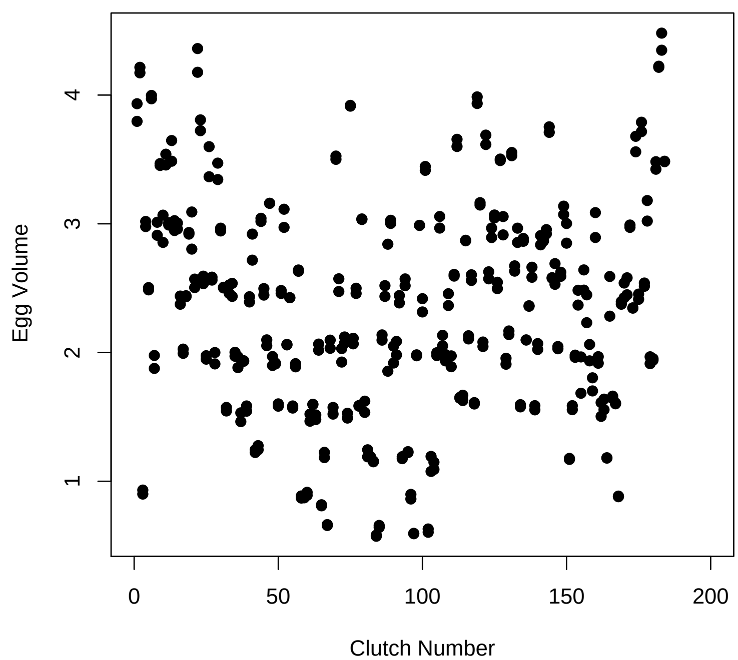
```
> GPA <- cbind(expand.grid(1:4, 1:5), gpa = c(3.08,
+     2.6, 3.44, 3.04, 2.36, 3.04, 3.28, 2.68, 2, 2.56,
+     2.52, 1.88, 3, 2.88, 3.44, 3.64, 2.68, 1.92,
```

```
+     3.28, 3.2))
> names(GPA)[1:2] <- c("person_num", "cluster")
> GPA$pwt <- 100/5
> clusterDesign <- svydesign(ids = ~cluster, weights = ~pwt,
+     data = GPA)
> svytotal(~gpa, design = clusterDesign)
```

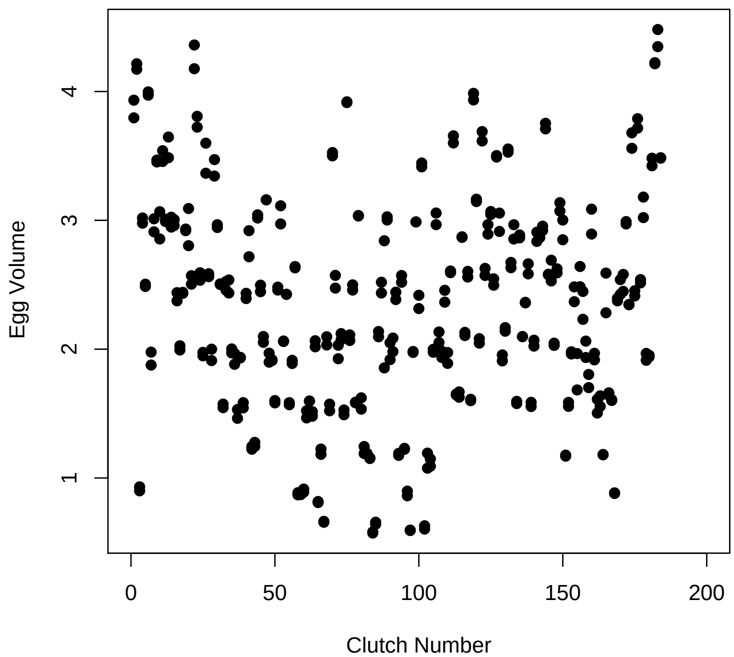
```
     total     SE
gpa 1130.4 67.167
```

5.3 Two-Stage Cluster Sampling

```
> plot(volume ~ clutch, xlim = c(0, 200), pch = 19,
+     data = coots, xlab = "Clutch Number", ylab = "Egg Volume")
```

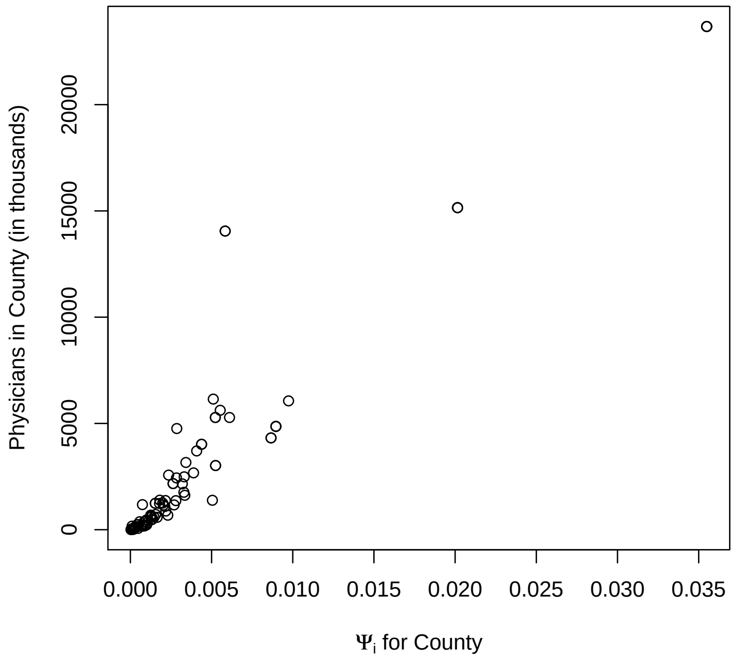



```
> plot(volume ~ clutch, xlim = c(0, 200), pch = 19,
+     data = coots, xlab = "Clutch Number", ylab = "Egg Volume")
```



6 Sampling with Unequal Probabilities

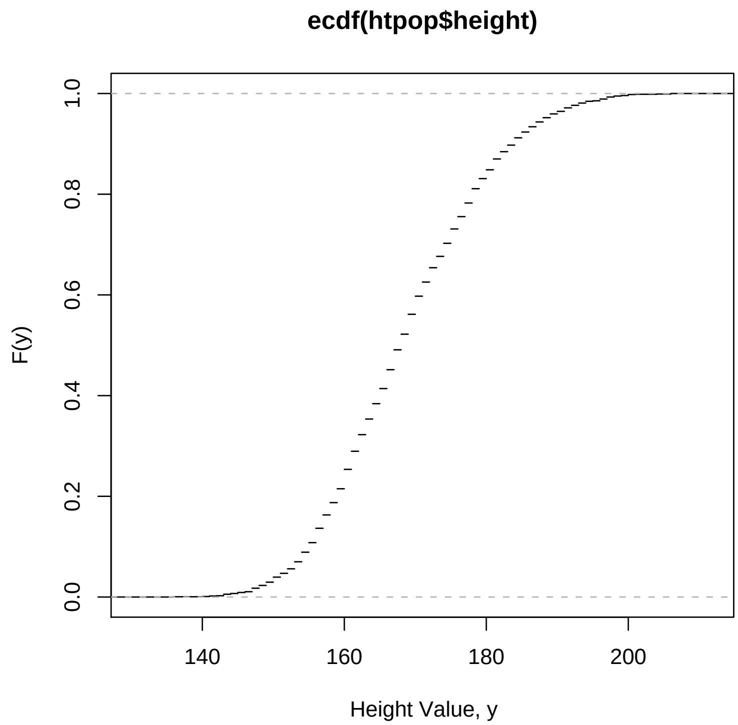
```
> data(statepop)
> statepop$psi <- statepop$popn/255077536

> plot(phys ~ psi, data = statepop, xlab = expression(paste(Psi[i],
+     " for County")), ylab = "Physicians in County (in thousands)")
```



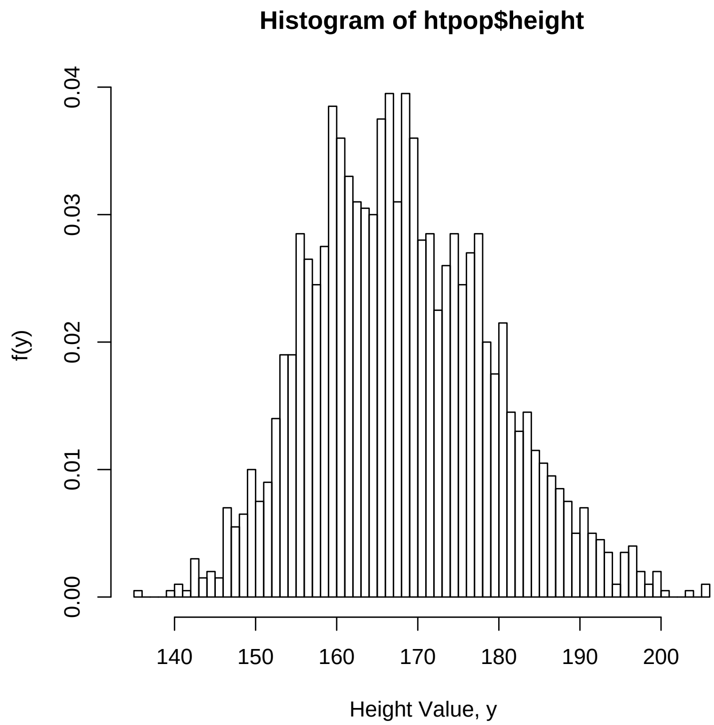
7 Complex Surveys

7.1 Estimating a Distribution Function

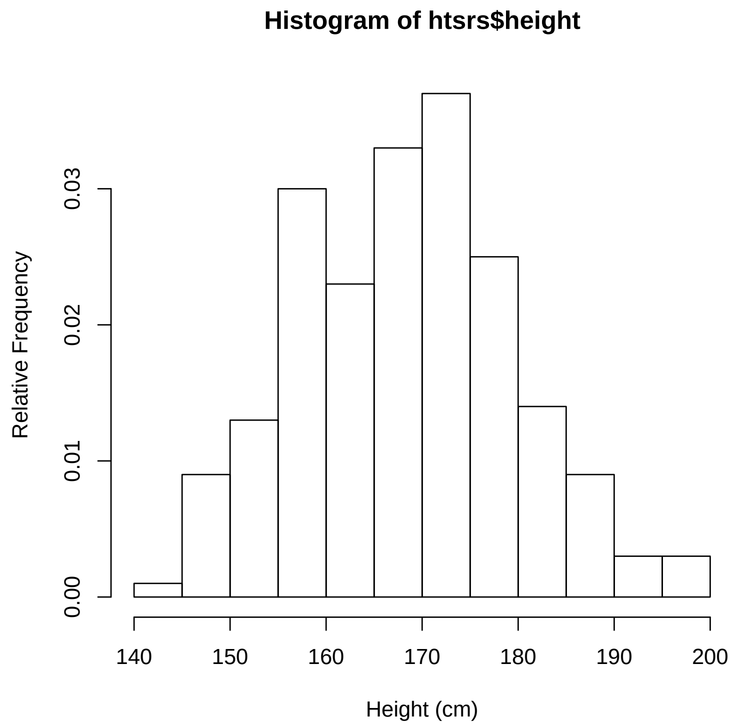
```
> data(htpop)
> popecdf <- ecdf(htpop$height)
> plot(popecdf, do.points = FALSE, ylab = "F(y)", xlab = "Height Value, y")
```



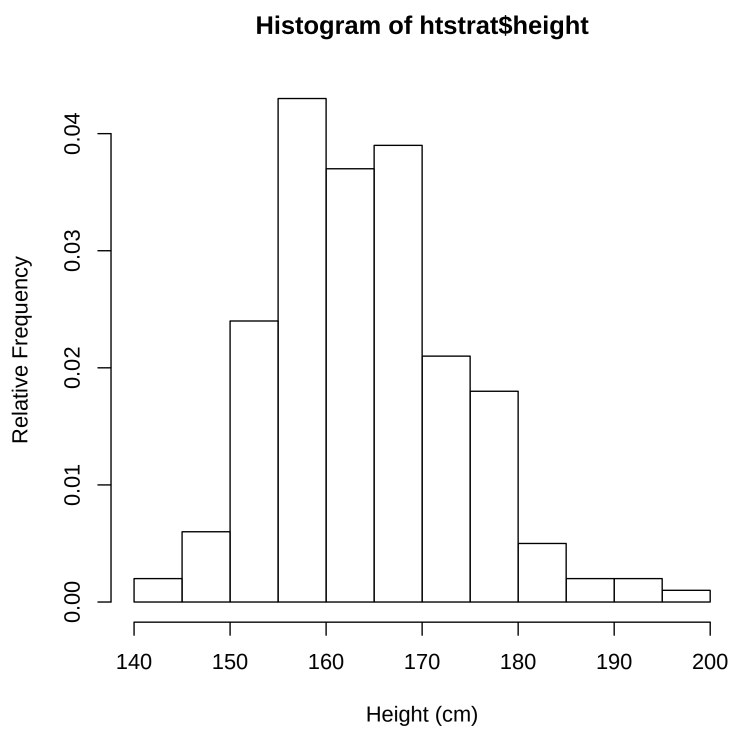

```
> minht <- min(htpop$height)
> breaks <- c(minht - 1, seq(from = minht, to = max(htpop$height),
+     by = 1))
> hist(htpop$height, ylab = "f(y)", breaks = breaks,
+     xlab = "Height Value, y", freq = FALSE)
```



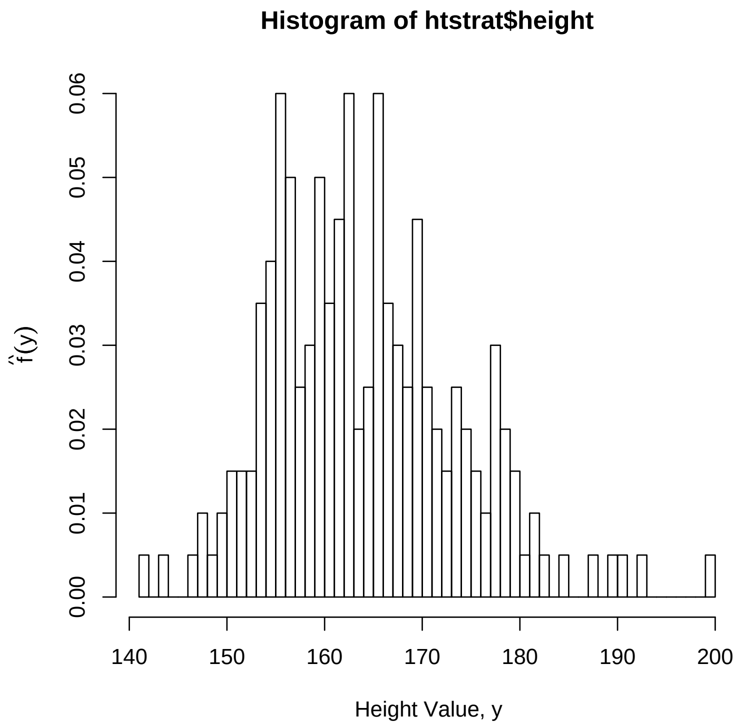

```
> data(htsrs)
> hist(htsrs$height, ylab = "Relative Frequency", xlab = "Height (cm)",
+     freq = FALSE)
```



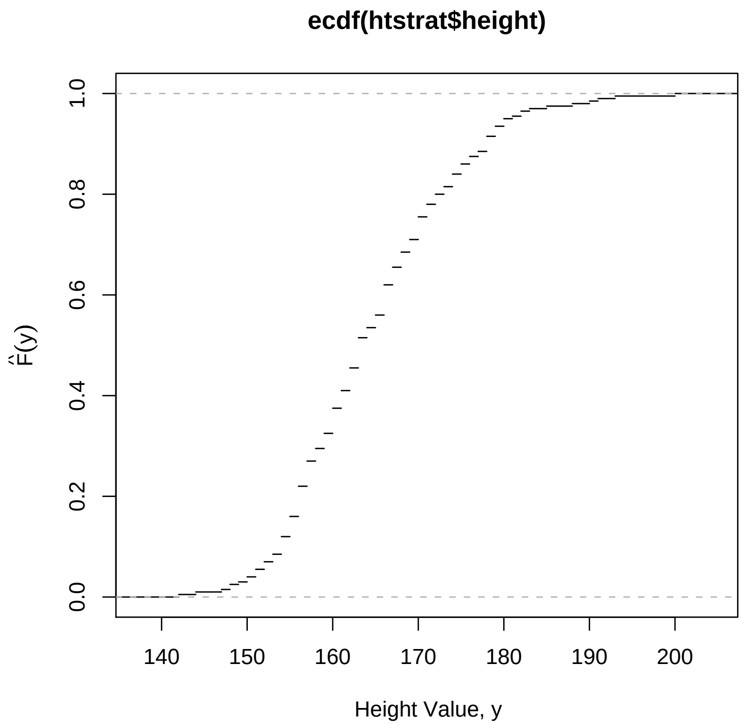

```
> data(htstrat)
> hist(htstrat$height, ylab = "Relative Frequency",
+      xlab = "Height (cm)", freq = FALSE)
```




```
> minht <- min(htstrat$height)
> breaks <- c(minht - 1, seq(from = minht, to = max(htstrat$height),
+     by = 1))
> hist(htstrat$height, ylab = expression(hat(f)(y)),
+     breaks = breaks, xlab = "Height Value, y", freq = FALSE)
```

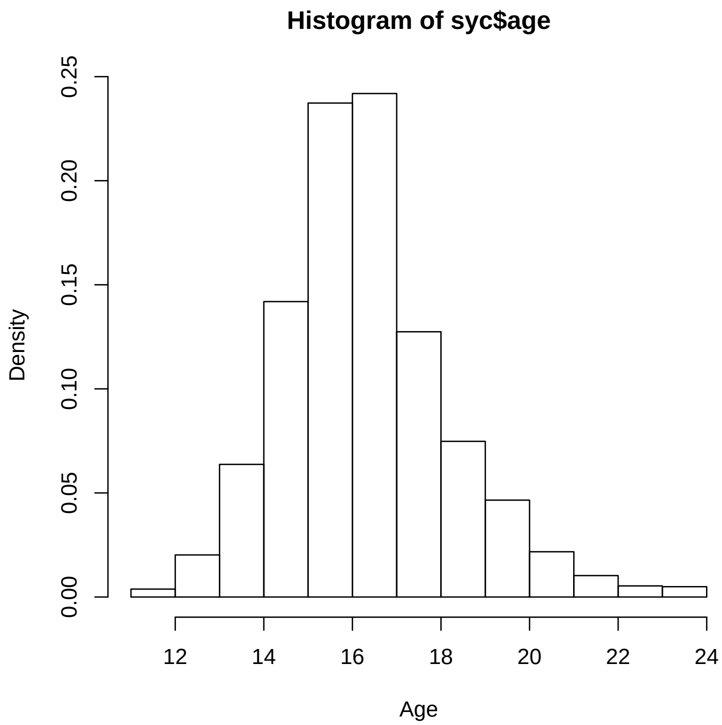



```
> stratecdf <- ecdf(htstrat$height)
> plot(stratecdf, do.points = FALSE, ylab = expression(hat(F)(y)),
+     xlab = "Height Value, y")
```



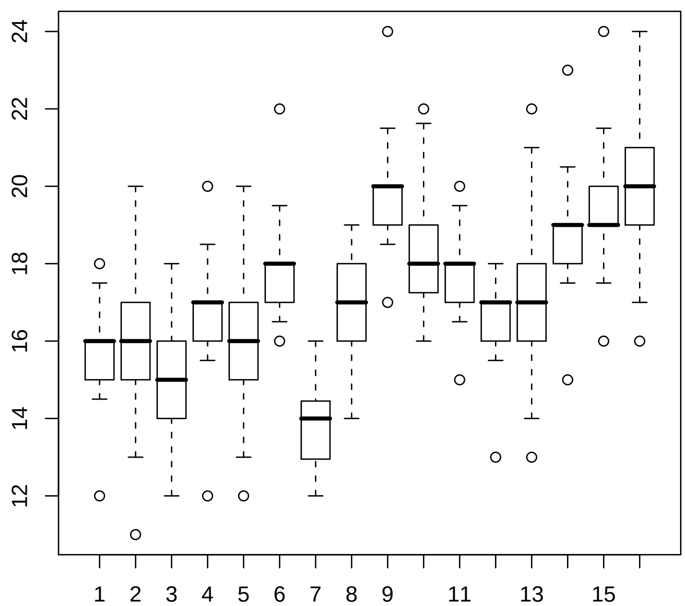
7.2 Plotting Data from a Complex Survey

```
> data(syc)
> hist(syc$age, freq = FALSE, xlab = "Age")
```



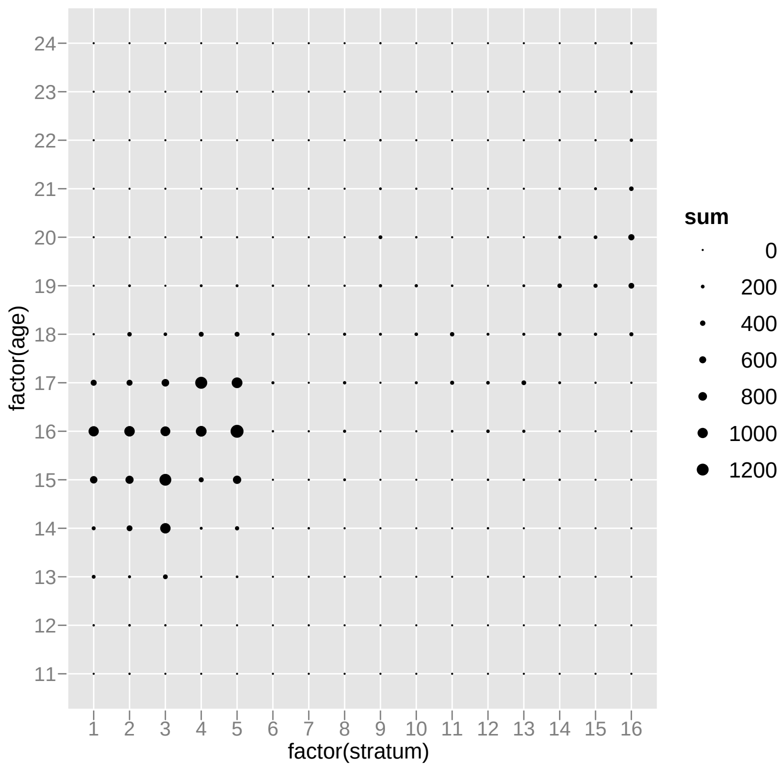
Note that in its current implementation, `svyboxplot` will only plot minimum and maximum as outliers if they are situated outside the whiskers. Other outliers are not plotted (see `?svyboxplot`). This explains the minor difference with Figure 7.8 on p. 237 of Lohr (1999).

```
> sycdesign <- svydesign(ids = ~psu, strata = ~stratum,
+     data = syc, weights = ~finalwt)
> oo <- options(survey.lonely.psu = "certainty")
> svyboxplot(age ~ factor(stratum), design = sycdesign)
> options(oo)
```



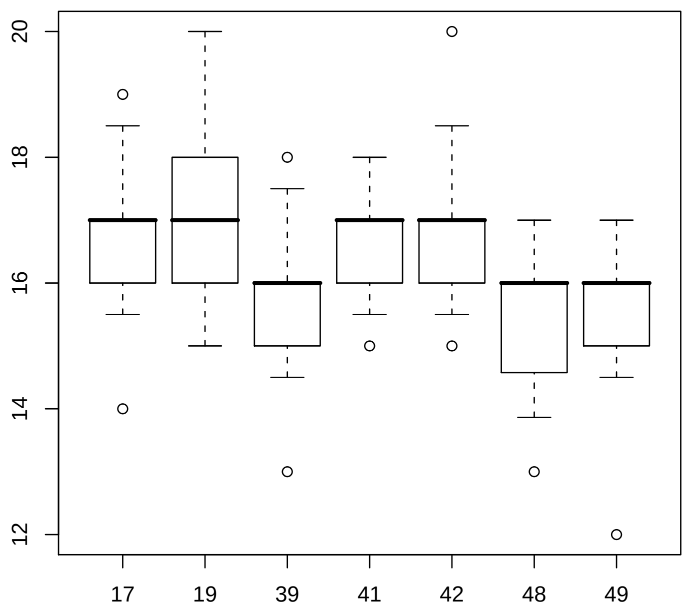
This kind of plot is particularly easy to formulate in the grammar of graphics, i.e. using the `ggplot2` package :

```
> p <- ggplot(syc, aes(x = factor(stratum), y = factor(age)))
> g <- p + stat_sum(aes(group = 1, weight = finalwt,
+     size = ..sum..))
> print(g)
```

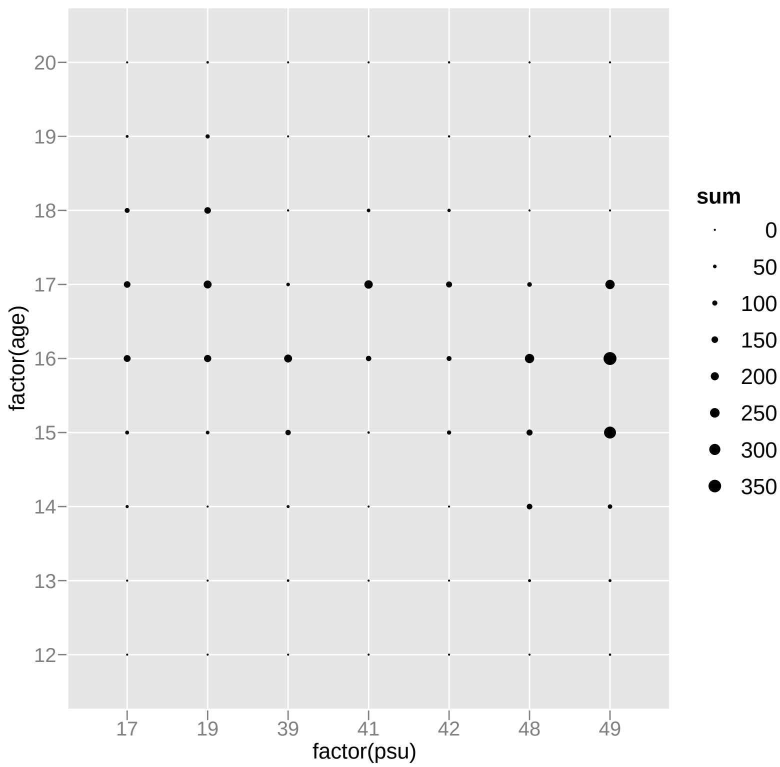


Note that in its current implementation, `svyboxplot` will only plot minimum and maximum as outliers if they are situated outside the whiskers. Other outliers are not plotted (see `?svyboxplot`). This explains the minor difference with Figure 7.10 on p. 238 of Lohr (1999).

```
> oo <- options(survey.lonely.psu = "certainty")
> sycstrat5 <- subset(sycdesign, stratum == 5)
> svyboxplot(age ~ factor(psu), design = sycstrat5)
> options(oo)
```




```
> sycstrat5df <- subset(syc, stratum == 5)
> p <- ggplot(sycstrat5df, aes(x = factor(psu), y = factor(age)))
> g <- p + stat_sum(aes(group = 1, weight = finalwt,
+     size = ..sum..))
> print(g)
```


20
19
18
17
16
15
14
13
12
factor(age)
17
19
39
41
42
48
49
factor(psu)
sum
0
50
100
150
200
250
300
350

8 Nonresponse

9 Variance Estimation in Complex Surveys

9.1 Linearization (Taylor Series) Methods

9.2 Random Group Methods

9.3 Resampling and Replication Methods

9.4 Generalized Variance Functions

9.5 Confidence Intervals

10 Categorical Data Analysis in Complex Surveys

10.1 Chi-Square Tests with Multinomial Sampling

```
> hh <- rbind(c(119, 188), c(88, 105))
> rownames(hh) <- c("cableYes", "cableNo")
> colnames(hh) <- c("computerYes", "computerNo")
> addmargins(hh)

         computerYes computerNo Sum
cableYes         119        188 307
cableNo           88        105 193
Sum              207        293 500

> chisq.test(hh, correct = FALSE)

        Pearson's Chi-squared test

data:  hh
X-squared = 2.281, df = 1, p-value = 0.1310

> nst <- rbind(c(46, 222), c(41, 109), c(17, 40), c(8,
+     26))
```

```
> colnames(nst) <- c("NR", "R")
> rownames(nst) <- c("generalStudent", "generalTutor",
+     "psychiatricStudent", "psychiatricTutor")
> addmargins(nst)

                    NR   R Sum
generalStudent      46 222 268
generalTutor        41 109 150
psychiatricStudent  17  40  57
psychiatricTutor     8  26  34
Sum                112 397 509

> chisq.test(nst, correct = FALSE)

        Pearson's Chi-squared test

data:  nst
X-squared = 8.2176, df = 3, p-value = 0.04172

> afp <- data.frame(nAccidents = 0:7, nPilots = c(12475,
+     4117, 1016, 269, 53, 14, 6, 2))
> lambdahat <- sum(afp$nAccidents * afp$nPilots/sum(afp$nPilots))
> observed <- afp$nPilots
> expected <- dpois(0:7, lambda = lambdahat) * sum(afp$nPilots)
> sum((observed - expected)^2/expected)

[1] 1935.127
```

10.2 Effects of Survey Design on Chi-Square Tests

```
> hh2 <- rbind(c(238, 376), c(176, 210))
> rownames(hh2) <- c("cableYes", "cableNo")
> colnames(hh2) <- c("computerYes", "computerNo")
> addmargins(hh2)

         computerYes computerNo  Sum
cableYes         238        376  614
cableNo          176        210  386
Sum              414        586 1000

> chisq.test(hh2, correct = FALSE)
```

```
        Pearson's Chi-squared test

data:  hh2
X-squared = 4.5621, df = 1, p-value = 0.03269
```

10.3 Corrections to Chi-Square Tests

11 Regression with Complex Survey Data

11.1 Model-Based Regression in Simple Random Samples

11.2 Regression in Complex Surveys

12 Other Topics in Sampling