

Causal Inference: What If. R and Stata code for Exercises

Book by M. A. Hernán and J. M. Robins R code by Joy Shi and Sean McGrath
Stata code by Eleanor Murray and Roger Logan
R Markdown code by Tom Palmer

16 June 2023

Contents

Preface	v
Downloading the code	v
Installing dependency packages	vi
Downloading the datasets	vi
 R code	 3
11. Why model?	3
Program 11.1	3
Program 11.2	4
Program 11.3	6
12. IP Weighting and Marginal Structural Models	7
Program 12.1	7
Program 12.2	9
Program 12.3	12
Program 12.4	15
Program 12.5	16
Program 12.6	17
Program 12.7	20
13. Standardization and the parametric G-formula	25
Program 13.1	25
Program 13.2	27
Program 13.3	28
Program 13.4	30
14. G-estimation of Structural Nested Models	33
Program 14.1	33
Program 14.2	34
Program 14.3	38
15. Outcome regression and propensity scores	41
Program 15.1	41
Program 15.2	45
Program 15.3	48
Program 15.4	54
16. Instrumental variables estimation	59
Program 16.1	59
Program 16.2	60
Program 16.3	60
Program 16.4	61
Program 16.5	62

17. Causal survival analysis	65
Program 17.1	65
Program 17.2	66
Program 17.3	68
Program 17.4	70
Program 17.5	73
Session information: R	77
 Stata code	 81
11. Why model: Stata	81
Program 11.1	81
Program 11.2	86
Program 11.3	88
12. IP Weighting and Marginal Structural Models: Stata	91
Program 12.1	91
Program 12.2	93
Program 12.3	95
Program 12.4	100
Program 12.5	102
Program 12.6	105
Program 12.7	107
13. Standardization and the parametric G-formula: Stata	113
Program 13.1	113
Program 13.2	115
Program 13.3	119
Program 13.4	124
14. G-estimation of Structural Nested Models: Stata	127
Program 14.1	127
Program 14.2	129
Program 14.3	134
15. Outcome regression and propensity scores: Stata	137
Program 15.1	137
Program 15.2	140
Program 15.3	143
Program 15.4	149
16. Instrumental variables estimation: Stata	157
Program 16.1	157
Program 16.2	160
Program 16.3	161
Program 16.4	161
Program 16.5	163
17. Causal survival analysis: Stata	165
Program 17.1	165
Program 17.2	166
Program 17.3	169
Program 17.4	176
Session information: Stata	183

Preface

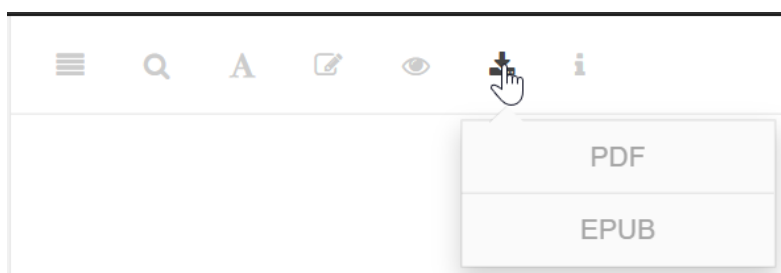
This book presents code examples from [Hernán and Robins \(2020\)](#), which is available in draft form from the following webpage.

<https://www.hsph.harvard.edu/miguel-hernan/causal-inference-book/>

The R code is based on the code by Joy Shi and Sean McGrath given [here](#).

The Stata code is based on the code by Eleanor Murray and Roger Logan given [here](#).

This repo is rendered at <https://remlapmot.github.io/cibookex-r/>. Click the download button above for the pdf and eBook versions.

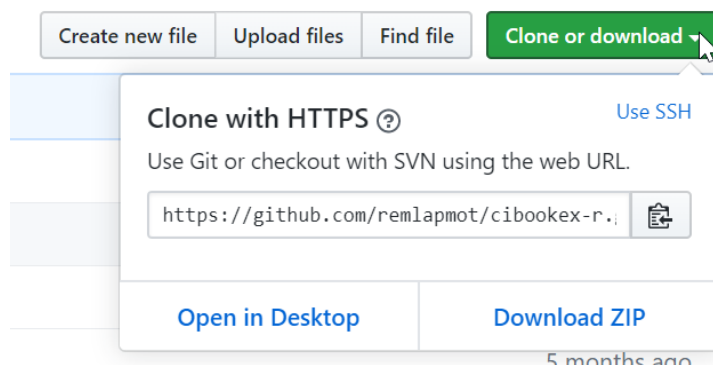


Downloading the code

The repo is available on GitHub [here](#). There are a number of ways to download the code.

Either,

- click the green *Clone or download* button then choose to *Open in Desktop* or *Download ZIP*.



The *Desktop* option means open in the [GitHub Desktop](#) app (if you have that installed on your machine). The *ZIP* option will give you a zip archive of the repo, which you then unzip.

- or fork the repo into your own GitHub account and then clone or download your forked repo to your machine.



Installing dependency packages

It is easiest to open the repo in RStudio, as an RStudio project, by doubling click the `.Rproj` file. This makes sure that R's working directory is at the top level of the repo. If you don't want to open the repo as a project set the working directory to the top level of the repo directories using `setwd()`. Then run:

```
# install.packages("devtools") # uncomment if devtools not installed
devtools::install_dev_deps()
```

Downloading the datasets

We assume that you have downloaded the data from the Causal Inference Book website and saved it to a `data` subdirectory. You can do this manually or with the following code (nb. we use the [here](#) package to reference the data subdirectory).

```
library(here)

dataurls <- list()
stub <- "https://cdn1.sph.harvard.edu/wp-content/uploads/sites/1268/"
dataurls[[1]] <- paste0(stub, "2012/10/nhefs_sas.zip")
dataurls[[2]] <- paste0(stub, "2012/10/nhefs_stata.zip")
dataurls[[3]] <- paste0(stub, "2017/01/nhefs_excel.zip")
dataurls[[4]] <- paste0(stub, "1268/20/nhefs.csv")

temp <- tempfile()
for (i in 1:3) {
  download.file(dataurls[[i]], temp)
  unzip(temp, exdir = "data")
}

download.file(dataurls[[4]], here("data", "nhefs.csv"))
```

R code

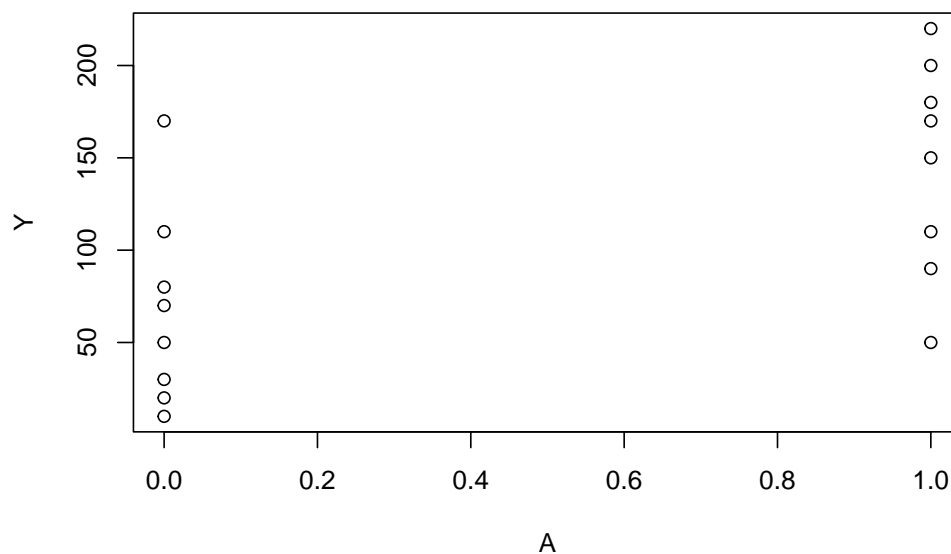
11. Why model?

Program 11.1

- Sample averages by treatment level
- Data from Figures 11.1 and 11.2

```
A <- c(1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0)
Y <- c(200, 150, 220, 110, 50, 180, 90, 170, 170, 30,
      70, 110, 80, 50, 10, 20)

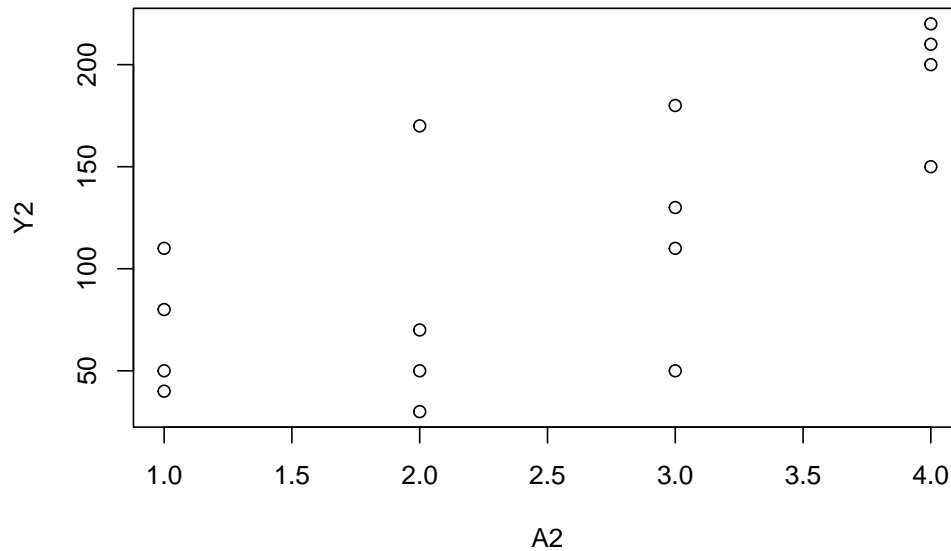
plot(A, Y)
```



```
summary(Y[A == 0])
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    10.0   27.5   60.0   67.5   87.5   170.0
summary(Y[A == 1])
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    50.0   105.0   160.0   146.2   185.0   220.0

A2 <- c(1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4)
Y2 <- c(110, 80, 50, 40, 170, 30, 70, 50, 110, 50, 180,
      130, 200, 150, 220, 210)
```

```
plot(A2, Y2)
```



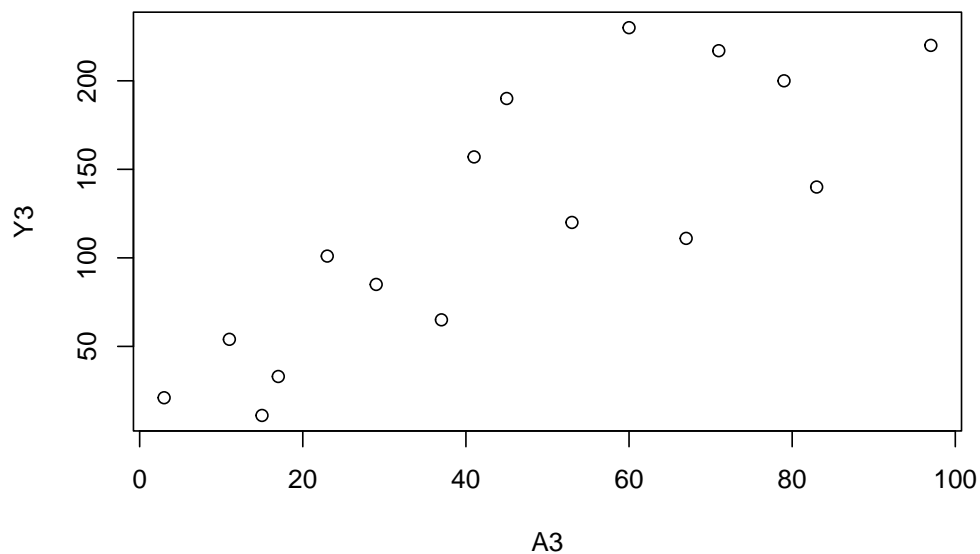
```
summary(Y2[A2 == 1])
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      40.0   47.5   65.0   70.0   87.5   110.0
summary(Y2[A2 == 2])
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       30    45    60    80    95    170
summary(Y2[A2 == 3])
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      50.0   95.0  120.0  117.5  142.5  180.0
summary(Y2[A2 == 4])
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     150.0  187.5  205.0  195.0  212.5  220.0
```

Program 11.2

- 2-parameter linear model
- Data from Figures 11.3 and 11.1

```
A3 <-
  c(3, 11, 17, 23, 29, 37, 41, 53, 67, 79, 83, 97, 60, 71, 15, 45)
Y3 <-
  c(21, 54, 33, 101, 85, 65, 157, 120, 111, 200, 140, 220, 230, 217,
    11, 190)

plot(Y3 ~ A3)
```



```
summary(glm(Y3 ~ A3))
##
## Call:
## glm(formula = Y3 ~ A3)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  24.5464    21.3300   1.151 0.269094
## A3           2.1372     0.3997   5.347 0.000103 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1944.109)
##
##    Null deviance: 82800  on 15  degrees of freedom
## Residual deviance: 27218  on 14  degrees of freedom
## AIC: 170.43
##
## Number of Fisher Scoring iterations: 2
predict(glm(Y3 ~ A3), data.frame(A3 = 90))
##      1
## 216.89

summary(glm(Y ~ A))
##
## Call:
## glm(formula = Y ~ A)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   67.50     19.72   3.424  0.00412 **
```

```
## A          78.75      27.88   2.824  0.01352 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 3109.821)
##
##      Null deviance: 68344  on 15  degrees of freedom
## Residual deviance: 43538  on 14  degrees of freedom
## AIC: 177.95
##
## Number of Fisher Scoring iterations: 2
```

Program 11.3

- 3-parameter linear model
- Data from Figure 11.3

```
Asq <- A3 * A3

mod3 <- glm(Y3 ~ A3 + Asq)
summary(mod3)
##
## Call:
## glm(formula = Y3 ~ A3 + Asq)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.40688    31.74777  -0.233   0.8192
## A3           4.10723     1.53088   2.683   0.0188 *
## Asq          -0.02038     0.01532  -1.331   0.2062
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1842.697)
##
##      Null deviance: 82800  on 15  degrees of freedom
## Residual deviance: 23955  on 13  degrees of freedom
## AIC: 170.39
##
## Number of Fisher Scoring iterations: 2
predict(mod3, data.frame(cbind(A3 = 90, Asq = 8100)))
##      1
## 197.1269
```

12. IP Weighting and Marginal Structural Models

Program 12.1

- Descriptive statistics from NHEFS data (Table 12.1)

```
library(here)

# install.packages("readxl") # install package if required
library("readxl")

nhefs <- read_excel(here("data", "NHEFS.xls"))
nhefs$cens <- ifelse(is.na(nhefs$wt82), 1, 0)

# provisionally ignore subjects with missing values for weight in 1982
nhefs.nmv <-
  nhefs[which(!is.na(nhefs$wt82)),]

lm(wt82_71 ~ qsmk, data = nhefs.nmv)
##
## Call:
## lm(formula = wt82_71 ~ qsmk, data = nhefs.nmv)
##
## Coefficients:
## (Intercept)          qsmk
##      1.984         2.541
# Smoking cessation
predict(lm(wt82_71 ~ qsmk, data = nhefs.nmv), data.frame(qsmk = 1))
##      1
## 4.525079
# No smoking cessation
predict(lm(wt82_71 ~ qsmk, data = nhefs.nmv), data.frame(qsmk = 0))
##      1
## 1.984498

# Table
summary(nhefs.nmv[which(nhefs.nmv$qsmk == 0),]$age)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  25.00  33.00   42.00  42.79  51.00   72.00
summary(nhefs.nmv[which(nhefs.nmv$qsmk == 0),]$wt71)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   40.82   59.19   68.49   70.30   79.38  151.73
```

```

summary(nhefs.nmv[which(nhefs.nmv$qsmk == 0),]$smokeintensity)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.00  15.00   20.00   21.19  30.00   60.00
summary(nhefs.nmv[which(nhefs.nmv$qsmk == 0),]$smokeyrs)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.00  15.00   23.00   24.09  32.00   64.00

summary(nhefs.nmv[which(nhefs.nmv$qsmk == 1),]$age)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     25.00  35.00   46.00   46.17  56.00   74.00
summary(nhefs.nmv[which(nhefs.nmv$qsmk == 1),]$wt71)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     39.58  60.67   71.21   72.35  81.08  136.98
summary(nhefs.nmv[which(nhefs.nmv$qsmk == 1),]$smokeintensity)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1.0   10.0   20.0   18.6   25.0   80.0
summary(nhefs.nmv[which(nhefs.nmv$qsmk == 1),]$smokeyrs)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.00  15.00   26.00   26.03  35.00   60.00

table(nhefs.nmv$qsmk, nhefs.nmv$sex)
##
##          0    1
##    0 542 621
##    1 220 183
prop.table(table(nhefs.nmv$qsmk, nhefs.nmv$sex), 1)
##
##          0          1
##    0 0.4660361 0.5339639
##    1 0.5459057 0.4540943

table(nhefs.nmv$qsmk, nhefs.nmv$race)
##
##          0    1
##    0 993 170
##    1 367  36
prop.table(table(nhefs.nmv$qsmk, nhefs.nmv$race), 1)
##
##          0          1
##    0 0.85382631 0.14617369
##    1 0.91066998 0.08933002

table(nhefs.nmv$qsmk, nhefs.nmv$education)
##
##          1    2    3    4    5
##    0 210 266 480  92 115
##    1  81  74 157  29  62
prop.table(table(nhefs.nmv$qsmk, nhefs.nmv$education), 1)
##
##          1          2          3          4          5
##    0 0.18056750 0.22871883 0.41272571 0.07910576 0.09888220
##    1 0.20099256 0.18362283 0.38957816 0.07196030 0.15384615

```

```

table(nhefs.nmv$qsmk, nehs.nmv$exercise)
##
##      0   1   2
##  0 237 485 441
##  1  63 176 164
prop.table(table(nhefs.nmv$qsmk, nehs.nmv$exercise), 1)
##
##           0           1           2
##  0 0.2037833 0.4170249 0.3791917
##  1 0.1563275 0.4367246 0.4069479

table(nhefs.nmv$qsmk, nehs.nmv$active)
##
##      0   1   2
##  0 532 527 104
##  1 170 188  45
prop.table(table(nhefs.nmv$qsmk, nehs.nmv$active), 1)
##
##           0           1           2
##  0 0.4574377 0.4531384 0.0894239
##  1 0.4218362 0.4665012 0.1116625

```

Program 12.2

- Estimating IP weights
- Data from NHEFS

```

# Estimation of ip weights via a logistic model
fit <- glm(
  qsmk ~ sex + race + age + I(age ^ 2) +
    as.factor(education) + smokeintensity +
    I(smokeintensity ^ 2) + smokeyrs + I(smokeyrs ^ 2) +
    as.factor(exercise) + as.factor(active) + wt71 + I(wt71 ^ 2),
  family = binomial(),
  data = nehs.nmv
)
summary(fit)
##
## Call:
## glm(formula = qsmk ~ sex + race + age + I(age^2) + as.factor(education) +
##      smokeintensity + I(smokeintensity^2) + smokeyrs + I(smokeyrs^2) +
##      as.factor(exercise) + as.factor(active) + wt71 + I(wt71^2),
##      family = binomial(), data = nehs.nmv)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.2425191   1.3808360  -1.624  0.104369
## sex            -0.5274782   0.1540496  -3.424  0.000617 ***
## race           -0.8392636   0.2100665  -3.995  6.46e-05 ***
## age              0.1212052   0.0512663   2.364  0.018068 *
## I(age^2)       -0.0008246   0.0005361  -1.538  0.124039
## as.factor(education)2 -0.0287755  0.1983506  -0.145  0.884653

```

```

## as.factor(education)3 0.0864318 0.1780850 0.485 0.627435
## as.factor(education)4 0.0636010 0.2732108 0.233 0.815924
## as.factor(education)5 0.4759606 0.2262237 2.104 0.035384 *
## smokeintensity -0.0772704 0.0152499 -5.067 4.04e-07 ***
## I(smokeintensity^2) 0.0010451 0.0002866 3.647 0.000265 ***
## smokeyrs -0.0735966 0.0277775 -2.650 0.008061 **
## I(smokeyrs^2) 0.0008441 0.0004632 1.822 0.068398 .
## as.factor(exercise)1 0.3548405 0.1801351 1.970 0.048855 *
## as.factor(exercise)2 0.3957040 0.1872400 2.113 0.034571 *
## as.factor(active)1 0.0319445 0.1329372 0.240 0.810100
## as.factor(active)2 0.1767840 0.2149720 0.822 0.410873
## wt71 -0.0152357 0.0263161 -0.579 0.562625
## I(wt71^2) 0.0001352 0.0001632 0.829 0.407370
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 1786.1 on 1565 degrees of freedom
## Residual deviance: 1676.9 on 1547 degrees of freedom
## AIC: 1714.9
##
## Number of Fisher Scoring iterations: 4

p.qsmk.obs <-
  ifelse(nhefs.nmv$qsmk == 0,
    1 - predict(fit, type = "response"),
    predict(fit, type = "response"))

nhefs.nmv$w <- 1 / p.qsmk.obs
summary(nhefs.nmv$w)
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 1.054 1.230 1.373 1.996 1.990 16.700
sd(nhefs.nmv$w)
## [1] 1.474787

# install.packages("geepack") # install package if required
library("geepack")
msm.w <- geeglm(
  wt82_71 ~ qsmk,
  data =uhefs.nmv,
  weights = w,
  id = seqn,
  corstr = "independence"
)
summary(msm.w)
##
## Call:
## geeglm(formula = wt82_71 ~ qsmk, data =uhefs.nmv, weights = w,
## id = seqn, corstr = "independence")
##
## Coefficients:

```



```

##              Estimate Std.err   Wald Pr(>|W|)
## (Intercept)   1.7800  0.2247 62.73 2.33e-15 ***
## qsmk          3.4405  0.5255 42.87 5.86e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = independence
## Estimated Scale Parameters:
##
##              Estimate Std.err
## (Intercept)    65.06    4.221
## Number of clusters: 1566 Maximum cluster size: 1

beta <- coef(msm.w)
SE <- coef(summary(msm.w))[, 2]
lcl <- beta - qnorm(0.975) * SE
ucl <- beta + qnorm(0.975) * SE
cbind(beta, lcl, ucl)
##              beta    lcl    ucl
## (Intercept) 1.780 1.340 2.22
## qsmk        3.441 2.411 4.47

# no association between sex and qsmk in pseudo-population
xtabs(nhefs.nmv$w ~ dhefs.nmv$sex + dhefs.nmv$qsmk)
##              dhefs.nmv$qsmk
## dhefs.nmv$sex      0      1
##              0 763.6 763.6
##              1 801.7 797.2

# "check" for positivity (White women)
table(nhefs.nmv$age[nhefs.nmv$race == 0 & dhefs.nmv$sex == 1],
      dhefs.nmv$qsmk[nhefs.nmv$race == 0 & dhefs.nmv$sex == 1])
##
##      0  1
## 25 24  3
## 26 14  5
## 27 18  2
## 28 20  5
## 29 15  4
## 30 14  5
## 31 11  5
## 32 14  7
## 33 12  3
## 34 22  5
## 35 16  5
## 36 13  3
## 37 14  1
## 38  6  2
## 39 19  4
## 40 10  4
## 41 13  3
## 42 16  3

```

```
## 43 14 3
## 44 9 4
## 45 12 5
## 46 19 4
## 47 19 4
## 48 19 4
## 49 11 3
## 50 18 4
## 51 9 3
## 52 11 3
## 53 11 4
## 54 17 9
## 55 9 4
## 56 8 7
## 57 9 2
## 58 8 4
## 59 5 4
## 60 5 4
## 61 5 2
## 62 6 5
## 63 3 3
## 64 7 1
## 65 3 2
## 66 4 0
## 67 2 0
## 69 6 2
## 70 2 1
## 71 0 1
## 72 2 2
## 74 0 1
```

Program 12.3

- Estimating stabilized IP weights
- Data from NHEFS

```
# estimation of denominator of ip weights
denom.fit <-
  glm(
    qsmk ~ as.factor(sex) + as.factor(race) + age + I(age ^ 2) +
      as.factor(education) + smokeintensity +
      I(smokeintensity ^ 2) + smokeyrs + I(smokeyrs ^ 2) +
      as.factor(exercise) + as.factor(active) + wt71 + I(wt71 ^ 2),
    family = binomial(),
    data = nhefs.nmv
  )
summary(denom.fit)
##
## Call:
## glm(formula = qsmk ~ as.factor(sex) + as.factor(race) + age +
##      I(age^2) + as.factor(education) + smokeintensity + I(smokeintensity^2) +
##      smokeyrs + I(smokeyrs^2) + as.factor(exercise) + as.factor(active) +
```

```
##      wt71 + I(wt71^2), family = binomial(), data = nhefs.nmv)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.242519    1.380836   -1.62  0.10437
## as.factor(sex)1    -0.527478    0.154050   -3.42  0.00062 ***
## as.factor(race)1    -0.839264    0.210067   -4.00  6.5e-05 ***
## age              0.121205    0.051266    2.36  0.01807 *
## I(age^2)         -0.000825    0.000536   -1.54  0.12404
## as.factor(education)2 -0.028776    0.198351   -0.15  0.88465
## as.factor(education)3  0.086432    0.178085    0.49  0.62744
## as.factor(education)4  0.063601    0.273211    0.23  0.81592
## as.factor(education)5  0.475961    0.226224    2.10  0.03538 *
## smokeintensity     -0.077270    0.015250   -5.07  4.0e-07 ***
## I(smokeintensity^2)    0.001045    0.000287    3.65  0.00027 ***
## smokeyrs          -0.073597    0.027777   -2.65  0.00806 **
## I(smokeyrs^2)        0.000844    0.000463    1.82  0.06840 .
## as.factor(exercise)1  0.354841    0.180135    1.97  0.04885 *
## as.factor(exercise)2  0.395704    0.187240    2.11  0.03457 *
## as.factor(active)1    0.031944    0.132937    0.24  0.81010
## as.factor(active)2    0.176784    0.214972    0.82  0.41087
## wt71              -0.015236    0.026316   -0.58  0.56262
## I(wt71^2)           0.000135    0.000163    0.83  0.40737
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1786.1  on 1565  degrees of freedom
## Residual deviance: 1676.9  on 1547  degrees of freedom
## AIC: 1715
##
## Number of Fisher Scoring iterations: 4

pd.qsmk <- predict(denom.fit, type = "response")

# estimation of numerator of ip weights
numer.fit <- glm(qsmk ~ 1, family = binomial(), data = nhefs.nmv)
summary(numer.fit)
##
## Call:
## glm(formula = qsmk ~ 1, family = binomial(), data = nhefs.nmv)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.0598     0.0578   -18.3  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1786.1  on 1565  degrees of freedom
```

```

## Residual deviance: 1786.1 on 1565 degrees of freedom
## AIC: 1788
##
## Number of Fisher Scoring iterations: 4

pn.qsmk <- predict(numer.fit, type = "response")

nhefs.nmv$sw <-
  ifelse(nhefs.nmv$qsmk == 0, ((1 - pn.qsmk) / (1 - pd.qsmk)),
        (pn.qsmk / pd.qsmk))

summary(nhefs.nmv$sw)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.331  0.867   0.950   0.999   1.079   4.298

msm.sw <- geeglm(
  wt82_71 ~ qsmk,
  data = nhefs.nmv,
  weights = sw,
  id = seqn,
  corstr = "independence"
)
summary(msm.sw)
##
## Call:
## geeglm(formula = wt82_71 ~ qsmk, data = nhefs.nmv, weights = sw,
##        id = seqn, corstr = "independence")
##
## Coefficients:
##              Estimate Std. err Wald Pr(>|W|)
## (Intercept)    1.780    0.225 62.7 2.3e-15 ***
## qsmk           3.441    0.525 42.9 5.9e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = independence
## Estimated Scale Parameters:
##
##              Estimate Std. err
## (Intercept)    60.7    3.71
## Number of clusters: 1566 Maximum cluster size: 1

beta <- coef(msm.sw)
SE <- coef(summary(msm.sw))[, 2]
lcl <- beta - qnorm(0.975) * SE
ucl <- beta + qnorm(0.975) * SE
cbind(beta, lcl, ucl)
##              beta  lcl  ucl
## (Intercept)  1.78 1.34 2.22
## qsmk         3.44 2.41 4.47

```

```
# no association between sex and qsmk in pseudo-population
xtabs(nhefs.nmv$sw ~ dhefs.nmv$sex + dhefs.nmv$qsmk)
##               dhefs.nmv$qsmk
## dhefs.nmv$sex    0    1
##                0 567 197
##                1 595 205
```

Program 12.4

- Estimating the parameters of a marginal structural mean model with a continuous treatment Data from NHEFS

```
# Analysis restricted to subjects reporting <=25 cig/day at baseline
nhefs.nmv.s <- subset(nhefs.nmv, smokeintensity <= 25)

# estimation of denominator of ip weights
den.fit.obj <- lm(
  smkintensity82_71 ~ as.factor(sex) +
    as.factor(race) + age + I(age ^ 2) +
    as.factor(education) + smokeintensity + I(smokeintensity ^ 2) +
    smokeyrs + I(smokeyrs ^ 2) + as.factor(exercise) + as.factor(active) + wt71 +
    I(wt71 ^ 2),
  data = dhefs.nmv.s
)
p.den <- predict(den.fit.obj, type = "response")
dens.den <-
  dnorm(nhefs.nmv.s$smkintensity82_71,
    p.den,
    summary(den.fit.obj)$sigma)

# estimation of numerator of ip weights
num.fit.obj <- lm(smkintensity82_71 ~ 1, data = dhefs.nmv.s)
p.num <- predict(num.fit.obj, type = "response")
dens.num <-
  dnorm(nhefs.nmv.s$smkintensity82_71,
    p.num,
    summary(num.fit.obj)$sigma)

nhefs.nmv.s$sw.a <- dens.num / dens.den
summary(nhefs.nmv.s$sw.a)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.19   0.89   0.97   1.00   1.05   5.10

msm.sw.cont <-
  geeglm(
    wt82_71 ~ smkintensity82_71 + I(smkintensity82_71 * smkintensity82_71),
    data = dhefs.nmv.s,
    weights = sw.a,
    id = seqn,
    corstr = "independence"
  )
summary(msm.sw.cont)
```

```
##
## Call:
## geeglm(formula = wt82_71 ~ smkintensity82_71 + I(smkintensity82_71 *
##      smkintensity82_71), data = nhefs.nmv.s, weights = sw.a, id = seqn,
##      corstr = "independence")
##
## Coefficients:
##
##              Estimate Std.err Wald Pr(>|W|)
## (Intercept)      2.00452  0.29512 46.13  1.1e-11 ***
## smkintensity82_71      -0.10899  0.03154 11.94  0.00055 ***
## I(smkintensity82_71 * smkintensity82_71)  0.00269  0.00242  1.24  0.26489
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = independence
## Estimated Scale Parameters:
##
##              Estimate Std.err
## (Intercept)      60.5      4.5
## Number of clusters: 1162 Maximum cluster size: 1

beta <- coef(msm.sw.cont)
SE <- coef(summary(msm.sw.cont))[, 2]
lcl <- beta - qnorm(0.975) * SE
ucl <- beta + qnorm(0.975) * SE
cbind(beta, lcl, ucl)
##
##              beta      lcl      ucl
## (Intercept)      2.00452  1.42610  2.58295
## smkintensity82_71      -0.10899 -0.17080 -0.04718
## I(smkintensity82_71 * smkintensity82_71)  0.00269 -0.00204  0.00743
```

Program 12.5

- Estimating the parameters of a marginal structural logistic model
- Data from NHEFS

```
table(nhefs.nmv$qsmk, nhefs.nmv$death)
##
##      0      1
## 0 963 200
## 1 312  91

# First, estimation of stabilized weights sw (same as in Program 12.3)
# Second, fit logistic model below
msm.logistic <- geeglm(
  death ~ qsmk,
  data = nhefs.nmv,
  weights = sw,
  id = seqn,
  family = binomial(),
  corstr = "independence"
)
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
summary(msm.logistic)
##
## Call:
## geeglm(formula = death ~ qsmk, family = binomial(), data = nhefs.nmv,
##        weights = sw, id = seqn, corstr = "independence")
##
## Coefficients:
##             Estimate Std.err   Wald Pr(>|W|)
## (Intercept)  -1.4905   0.0789 356.50  <2e-16 ***
## qsmk          0.0301   0.1573   0.04    0.85
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = independence
## Estimated Scale Parameters:
##
##             Estimate Std.err
## (Intercept)         1  0.0678
## Number of clusters: 1566 Maximum cluster size: 1

beta <- coef(msm.logistic)
SE <- coef(summary(msm.logistic))[, 2]
lcl <- beta - qnorm(0.975) * SE
ucl <- beta + qnorm(0.975) * SE
cbind(beta, lcl, ucl)
##             beta    lcl    ucl
## (Intercept) -1.4905 -1.645 -1.336
## qsmk         0.0301 -0.278  0.338
```

Program 12.6

- Assessing effect modification by sex using a marginal structural mean model
- Data from NHEFS

```
table(nhefs.nmv$sex)
##
##    0    1
## 762 804

# estimation of denominator of ip weights
denom.fit <-
  glm(
    qsmk ~ as.factor(sex) + as.factor(race) + age + I(age ^ 2) +
      as.factor(education) + smokeintensity +
      I(smokeintensity ^ 2) + smokeyrs + I(smokeyrs ^ 2) +
      as.factor(exercise) + as.factor(active) + wt71 + I(wt71 ^ 2),
    family = binomial(),
    data = nhefs.nmv
  )
summary(denom.fit)
##
```

```
## Call:
## glm(formula = qsmk ~ as.factor(sex) + as.factor(race) + age +
##      I(age^2) + as.factor(education) + smokeintensity + I(smokeintensity^2) +
##      smokeyrs + I(smokeyrs^2) + as.factor(exercise) + as.factor(active) +
##      wt71 + I(wt71^2), family = binomial(), data = nhefs.nmv)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.242519   1.380836  -1.62  0.10437
## as.factor(sex)1    -0.527478   0.154050  -3.42  0.00062 ***
## as.factor(race)1   -0.839264   0.210067  -4.00  6.5e-05 ***
## age              0.121205   0.051266   2.36  0.01807 *
## I(age^2)         -0.000825   0.000536  -1.54  0.12404
## as.factor(education)2 -0.028776   0.198351  -0.15  0.88465
## as.factor(education)3  0.086432   0.178085   0.49  0.62744
## as.factor(education)4  0.063601   0.273211   0.23  0.81592
## as.factor(education)5  0.475961   0.226224   2.10  0.03538 *
## smokeintensity     -0.077270   0.015250  -5.07  4.0e-07 ***
## I(smokeintensity^2)    0.001045   0.000287   3.65  0.00027 ***
## smokeyrs          -0.073597   0.027777  -2.65  0.00806 **
## I(smokeyrs^2)         0.000844   0.000463   1.82  0.06840 .
## as.factor(exercise)1  0.354841   0.180135   1.97  0.04885 *
## as.factor(exercise)2  0.395704   0.187240   2.11  0.03457 *
## as.factor(active)1    0.031944   0.132937   0.24  0.81010
## as.factor(active)2    0.176784   0.214972   0.82  0.41087
## wt71              -0.015236   0.026316  -0.58  0.56262
## I(wt71^2)           0.000135   0.000163   0.83  0.40737
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1786.1  on 1565  degrees of freedom
## Residual deviance: 1676.9  on 1547  degrees of freedom
## AIC: 1715
##
## Number of Fisher Scoring iterations: 4

pd.qsmk <- predict(denom.fit, type = "response")

# estimation of numerator of ip weights
numer.fit <-
  glm(qsmk ~ as.factor(sex), family = binomial(), data = nhefs.nmv)
summary(numer.fit)
##
## Call:
## glm(formula = qsmk ~ as.factor(sex), family = binomial(), data = nhefs.nmv)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -0.9016   0.0799  -11.28 <2e-16 ***
## as.factor(sex)1  -0.3202   0.1160   -2.76  0.0058 **
```



```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1786.1  on 1565  degrees of freedom
## Residual deviance: 1778.4  on 1564  degrees of freedom
## AIC: 1782
##
## Number of Fisher Scoring iterations: 4
pn.qsmk <- predict(numer.fit, type = "response")

nhefs.nmv$sw.a <-
  ifelse(nhefs.nmv$qsmk == 0, ((1 - pn.qsmk) / (1 - pd.qsmk)),
        (pn.qsmk / pd.qsmk))

summary(nhefs.nmv$sw.a)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.29   0.88   0.96   1.00   1.08   3.80
sd(nhefs.nmv$sw.a)
## [1] 0.271

# Estimating parameters of a marginal structural mean model
msm.emm <- geeglm(
  wt82_71 ~ as.factor(qsmk) + as.factor(sex)
  + as.factor(qsmk):as.factor(sex),
  data = nhefs.nmv,
  weights = sw.a,
  id = seqn,
  corstr = "independence"
)
summary(msm.emm)
##
## Call:
## geeglm(formula = wt82_71 ~ as.factor(qsmk) + as.factor(sex) +
##       as.factor(qsmk):as.factor(sex), data = nhefs.nmv, weights = sw.a,
##       id = seqn, corstr = "independence")
##
## Coefficients:
##
##              Estimate Std.err Wald Pr(>|W|)
## (Intercept)      1.78445  0.30984 33.17  8.5e-09 ***
## as.factor(qsmk)1      3.52198  0.65707 28.73  8.3e-08 ***
## as.factor(sex)1     -0.00872  0.44882  0.00   0.98
## as.factor(qsmk)1:as.factor(sex)1 -0.15948  1.04608  0.02   0.88
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = independence
## Estimated Scale Parameters:
##
##              Estimate Std.err
## (Intercept)      60.8    3.71
```

```
## Number of clusters: 1566 Maximum cluster size: 1

beta <- coef(msm.emm)
SE <- coef(summary(msm.emm))[, 2]
lcl <- beta - qnorm(0.975) * SE
ucl <- beta + qnorm(0.975) * SE
cbind(beta, lcl, ucl)
##
##          beta      lcl      ucl
## (Intercept)    1.78445  1.177 2.392
## as.factor(qsmk)1    3.52198  2.234 4.810
## as.factor(sex)1   -0.00872 -0.888 0.871
## as.factor(qsmk)1:as.factor(sex)1 -0.15948 -2.210 1.891
```

Program 12.7

- Estimating IP weights to adjust for selection bias due to censoring
- Data from NHEFS

```
table(nhefs$qsmk, neufs$cens)
##
##      0      1
## 0 1163    38
## 1  403    25

summary(nhefs[which(nhefs$cens == 0),]$wt71)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    39.6    59.5    69.2    70.8    79.8   151.7
summary(nhefs[which(nhefs$cens == 1),]$wt71)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    36.2    63.1    72.1    76.6    87.9   169.2

# estimation of denominator of ip weights for A
denom.fit <-
  glm(
    qsmk ~ as.factor(sex) + as.factor(race) + age + I(age ^ 2) +
      as.factor(education) + smokeintensity +
      I(smokeintensity ^ 2) + smokeyrs + I(smokeyrs ^ 2) +
      as.factor(exercise) + as.factor(active) + wt71 + I(wt71 ^ 2),
    family = binomial(),
    data = neufs
  )
summary(denom.fit)
##
## Call:
## glm(formula = qsmk ~ as.factor(sex) + as.factor(race) + age +
##      I(age^2) + as.factor(education) + smokeintensity + I(smokeintensity^2) +
##      smokeyrs + I(smokeyrs^2) + as.factor(exercise) + as.factor(active) +
##      wt71 + I(wt71^2), family = binomial(), data = neufs)
##
## Coefficients:
##
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.988902   1.241279   -1.60  0.10909
```

```

## as.factor(sex)1      -0.507522    0.148232   -3.42  0.00062 ***
## as.factor(race)1     -0.850231    0.205872   -4.13  3.6e-05 ***
## age                  0.103013    0.048900    2.11  0.03515 *
## I(age^2)             -0.000605    0.000507   -1.19  0.23297
## as.factor(education)2 -0.098320    0.190655   -0.52  0.60607
## as.factor(education)3 0.015699    0.170714    0.09  0.92673
## as.factor(education)4 -0.042526    0.264276   -0.16  0.87216
## as.factor(education)5 0.379663    0.220395    1.72  0.08495 .
## smokeintensity       -0.065156    0.014759   -4.41  1.0e-05 ***
## I(smokeintensity^2)   0.000846    0.000276    3.07  0.00216 **
## smokeyrs             -0.073371    0.026996   -2.72  0.00657 **
## I(smokeyrs^2)         0.000838    0.000443    1.89  0.05867 .
## as.factor(exercise)1  0.291412    0.173554    1.68  0.09314 .
## as.factor(exercise)2  0.355052    0.179929    1.97  0.04846 *
## as.factor(active)1    0.010875    0.129832    0.08  0.93324
## as.factor(active)2    0.068312    0.208727    0.33  0.74346
## wt71                 -0.012848    0.022283   -0.58  0.56423
## I(wt71^2)            0.000121    0.000135    0.89  0.37096
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1876.3  on 1628  degrees of freedom
## Residual deviance: 1766.7  on 1610  degrees of freedom
## AIC: 1805
##
## Number of Fisher Scoring iterations: 4

pd.qsmk <- predict(denom.fit, type = "response")

# estimation of numerator of ip weights for A
numer.fit <- glm(qsmk ~ 1, family = binomial(), data = nhefs)
summary(numer.fit)
##
## Call:
## glm(formula = qsmk ~ 1, family = binomial(), data = nhefs)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.0318      0.0563   -18.3  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1876.3  on 1628  degrees of freedom
## Residual deviance: 1876.3  on 1628  degrees of freedom
## AIC: 1878
##
## Number of Fisher Scoring iterations: 4
pn.qsmk <- predict(numer.fit, type = "response")

```

```

# estimation of denominator of ip weights for C
denom.cens <- glm(
  cens ~ as.factor(qsmk) + as.factor(sex) +
    as.factor(race) + age + I(age ^ 2) +
    as.factor(education) + smokeintensity +
    I(smokeintensity ^ 2) + smokeyrs + I(smokeyrs ^ 2) +
    as.factor(exercise) + as.factor(active) + wt71 + I(wt71 ^ 2),
  family = binomial(),
  data = nhfs
)
summary(denom.cens)
##
## Call:
## glm(formula = cens ~ as.factor(qsmk) + as.factor(sex) + as.factor(race) +
##      age + I(age^2) + as.factor(education) + smokeintensity +
##      I(smokeintensity^2) + smokeyrs + I(smokeyrs^2) + as.factor(exercise) +
##      as.factor(active) + wt71 + I(wt71^2), family = binomial(),
##      data = nhfs)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      4.014466   2.576106    1.56  0.1192
## as.factor(qsmk)1    0.516867   0.287716    1.80  0.0724 .
## as.factor(sex)1     0.057313   0.330278    0.17  0.8622
## as.factor(race)1   -0.012271   0.452489   -0.03  0.9784
## age               -0.269729   0.117465   -2.30  0.0217 *
## I(age^2)           0.002884   0.001114    2.59  0.0096 **
## as.factor(education)2 -0.440788   0.419399   -1.05  0.2933
## as.factor(education)3 -0.164688   0.370547   -0.44  0.6567
## as.factor(education)4  0.138447   0.569797    0.24  0.8080
## as.factor(education)5 -0.382382   0.560181   -0.68  0.4949
## smokeintensity      0.015712   0.034732    0.45  0.6510
## I(smokeintensity^2) -0.000113   0.000606   -0.19  0.8517
## smokeyrs            0.078597   0.074958    1.05  0.2944
## I(smokeyrs^2)       -0.000557   0.001032   -0.54  0.5894
## as.factor(exercise)1 -0.971471   0.387810   -2.51  0.0122 *
## as.factor(exercise)2 -0.583989   0.372313   -1.57  0.1168
## as.factor(active)1  -0.247479   0.325455   -0.76  0.4470
## as.factor(active)2   0.706583   0.396458    1.78  0.0747 .
## wt71               -0.087887   0.040012   -2.20  0.0281 *
## I(wt71^2)           0.000635   0.000226    2.81  0.0049 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 533.36  on 1628  degrees of freedom
## Residual deviance: 465.36  on 1609  degrees of freedom
## AIC: 505.4
##
## Number of Fisher Scoring iterations: 7

```

```

pd.cens <- 1 - predict(denom.cens, type = "response")

# estimation of numerator of ip weights for C
numer.cens <-
  glm(cens ~ as.factor(qsmk), family = binomial(), data = nhefs)
summary(numer.cens)
##
## Call:
## glm(formula = cens ~ as.factor(qsmk), family = binomial(), data = nhefs)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.421      0.165  -20.75  <2e-16 ***
## as.factor(qsmk)1  0.641      0.264   2.43   0.015 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 533.36  on 1628  degrees of freedom
## Residual deviance: 527.76  on 1627  degrees of freedom
## AIC: 531.8
##
## Number of Fisher Scoring iterations: 6
pn.cens <- 1 - predict(numer.cens, type = "response")

nhefs$sw.a <-
  ifelse(nhefs$qsmk == 0, ((1 - pn.qsmk) / (1 - pd.qsmk)),
    (pn.qsmk / pd.qsmk))
nhefs$sw.c <- pn.cens / pd.cens
nhefs$sw <- nhefs$sw.c * nhefs$sw.a

summary(nhefs$sw.a)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.33   0.86   0.95   1.00   1.08   4.21
sd(nhefs$sw.a)
## [1] 0.284
summary(nhefs$sw.c)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.94   0.98   0.99   1.01   1.01   7.58
sd(nhefs$sw.c)
## [1] 0.178
summary(nhefs$sw)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.35   0.86   0.94   1.01   1.08  12.86
sd(nhefs$sw)
## [1] 0.411

msm.sw <- geeglm(
  wt82_71 ~ qsmk,
  data = nhefs,
  weights = sw,

```

```

    id = seqn,
    corstr = "independence"
)
summary(msm.sw)
##
## Call:
## geeglm(formula = wt82_71 ~ qsmk, data = nhefs, weights = sw,
##        id = seqn, corstr = "independence")
##
## Coefficients:
##              Estimate Std.err Wald Pr(>|W|)
## (Intercept)    1.662    0.233  51.0  9.3e-13 ***
## qsmk           3.496    0.526  44.2  2.9e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = independence
## Estimated Scale Parameters:
##
##              Estimate Std.err
## (Intercept)    61.8    3.83
## Number of clusters: 1566 Maximum cluster size: 1

beta <- coef(msm.sw)
SE <- coef(summary(msm.sw))[, 2]
lcl <- beta - qnorm(0.975) * SE
ucl <- beta + qnorm(0.975) * SE
cbind(beta, lcl, ucl)
##              beta  lcl  ucl
## (Intercept)  1.66 1.21 2.12
## qsmk         3.50 2.47 4.53

```

13. Standardization and the parametric G-formula

Program 13.1

- Estimating the mean outcome within levels of treatment and confounders
- Data from NHEFS

```
library(here)

#install.packages("readxl") # install package if required
library("readxl")
nhefs <- read_excel(here("data", "NHEFS.xls"))

# some preprocessing of the data
nhefs$cens <- ifelse(is.na(nhefs$wt82), 1, 0)

fit <-
  glm(
    wt82_71 ~ qsmk + sex + race + age + I(age * age) + as.factor(education)
    + smokeintensity + I(smokeintensity * smokeintensity) + smokeyrs
    + I(smokeyrs * smokeyrs) + as.factor(exercise) + as.factor(active)
    + wt71 + I(wt71 * wt71) + qsmk * smokeintensity,
    data = nhefs
  )
summary(fit)
##
## Call:
## glm(formula = wt82_71 ~ qsmk + sex + race + age + I(age * age) +
##      as.factor(education) + smokeintensity + I(smokeintensity *
##      smokeintensity) + smokeyrs + I(smokeyrs * smokeyrs) + as.factor(exercise) +
##      as.factor(active) + wt71 + I(wt71 * wt71) + qsmk * smokeintensity,
##      data = nhefs)
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -1.5881657    4.3130359   -0.368  0.712756
## qsmk                           2.5595941    0.8091486    3.163  0.001590 **
## sex                           -1.4302717    0.4689576   -3.050  0.002328 **
## race                           0.5601096    0.5818888    0.963  0.335913
## age                            0.3596353    0.1633188    2.202  0.027809 *
## I(age * age)                   -0.0061010    0.0017261   -3.534  0.000421 ***
## as.factor(education)2          0.7904440    0.6070005    1.302  0.193038
```

```
## as.factor(education)3      0.5563124  0.5561016   1.000 0.317284
## as.factor(education)4      1.4915695  0.8322704   1.792 0.073301 .
## as.factor(education)5     -0.1949770  0.7413692  -0.263 0.792589
## smokeintensity             0.0491365  0.0517254   0.950 0.342287
## I(smokeintensity * smokeintensity) -0.0009907  0.0009380  -1.056 0.291097
## smokeyrs                   0.1343686  0.0917122   1.465 0.143094
## I(smokeyrs * smokeyrs)     -0.0018664  0.0015437  -1.209 0.226830
## as.factor(exercise)1       0.2959754  0.5351533   0.553 0.580298
## as.factor(exercise)2       0.3539128  0.5588587   0.633 0.526646
## as.factor(active)1         -0.9475695  0.4099344  -2.312 0.020935 *
## as.factor(active)2         -0.2613779  0.6845577  -0.382 0.702647
## wt71                       0.0455018  0.0833709   0.546 0.585299
## I(wt71 * wt71)             -0.0009653  0.0005247  -1.840 0.066001 .
## qsmk:smokeintensity        0.0466628  0.0351448   1.328 0.184463
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 53.5683)
##
## Null deviance: 97176  on 1565  degrees of freedom
## Residual deviance: 82763  on 1545  degrees of freedom
## (63 observations deleted due to missingness)
## AIC: 10701
##
## Number of Fisher Scoring iterations: 2
nhefs$predicted.meanY <- predict(fit, nhefs)

nhefs[which(nhefs$seqn == 24770), c(
  "predicted.meanY",
  "qsmk",
  "sex",
  "race",
  "age",
  "education",
  "smokeintensity",
  "smokeyrs",
  "exercise",
  "active",
  "wt71"
)]
## # A tibble: 1 x 11
##   predicted.meanY qsmk sex race age education smokeintensity smokeyrs
##           <dbl> <dbl> <dbl> <dbl> <dbl>         <dbl>         <dbl>
## 1           0.342    0    0    0    26           4           15           12
## # i 3 more variables: exercise <dbl>, active <dbl>, wt71 <dbl>

summary(nhefs$predicted.meanY[nhefs$cens == 0])
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -10.876  1.116  3.042  2.638  4.511  9.876
summary(nhefs$wt82_71[nhefs$cens == 0])
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -41.280 -1.478  2.604  2.638  6.690 48.538
```


Program 13.2

- Standardizing the mean outcome to the baseline confounders
- Data from Table 2.2

```
id <- c(
  "Rheia",
  "Kronos",
  "Demeter",
  "Hades",
  "Hestia",
  "Poseidon",
  "Hera",
  "Zeus",
  "Artemis",
  "Apollo",
  "Leto",
  "Ares",
  "Athena",
  "Hephaestus",
  "Aphrodite",
  "Cyclope",
  "Persephone",
  "Hermes",
  "Hebe",
  "Dionysus"
)
N <- length(id)
L <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
A <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1)
Y <- c(0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0)
interv <- rep(-1, N)
observed <- cbind(L, A, Y, interv)
untreated <- cbind(L, rep(0, N), rep(NA, N), rep(0, N))
treated <- cbind(L, rep(1, N), rep(NA, N), rep(1, N))
table22 <- as.data.frame(rbind(observed, untreated, treated))
table22$id <- rep(id, 3)

glm.obj <- glm(Y ~ A * L, data = table22)
summary(glm.obj)
##
## Call:
## glm(formula = Y ~ A * L, data = table22)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.500e-01  2.552e-01   0.980   0.342
## A            2.602e-16  3.608e-01   0.000   1.000
## L            4.167e-01  3.898e-01   1.069   0.301
## A:L          -3.237e-16  4.959e-01   0.000   1.000
##
## (Dispersion parameter for gaussian family taken to be 0.2604167)
##
##      Null deviance: 5.0000  on 19  degrees of freedom
```

```
## Residual deviance: 4.1667 on 16 degrees of freedom
## (40 observations deleted due to missingness)
## AIC: 35.385
##
## Number of Fisher Scoring iterations: 2
table22$predicted.meanY <- predict(glm.obj, table22)

mean(table22$predicted.meanY[table22$interv == -1])
## [1] 0.5
mean(table22$predicted.meanY[table22$interv == 0])
## [1] 0.5
mean(table22$predicted.meanY[table22$interv == 1])
## [1] 0.5
```

Program 13.3

- Standardizing the mean outcome to the baseline confounders:
- Data from NHEFS

```
# create a dataset with 3 copies of each subject
nhefs$interv <- -1 # 1st copy: equal to original one

interv0 <- nhefs # 2nd copy: treatment set to 0, outcome to missing
interv0$interv <- 0
interv0$qsmk <- 0
interv0$wt82_71 <- NA

interv1 <- nhefs # 3rd copy: treatment set to 1, outcome to missing
interv1$interv <- 1
interv1$qsmk <- 1
interv1$wt82_71 <- NA

onesample <- rbind(nhefs, interv0, interv1) # combining datasets

# linear model to estimate mean outcome conditional on treatment and confounders
# parameters are estimated using original observations only (nhefs)
# parameter estimates are used to predict mean outcome for observations with
# treatment set to 0 (interv=0) and to 1 (interv=1)

std <- glm(
  wt82_71 ~ qsmk + sex + race + age + I(age * age)
  + as.factor(education) + smokeintensity
  + I(smokeintensity * smokeintensity) + smokeyrs
  + I(smokeyrs * smokeyrs) + as.factor(exercise)
  + as.factor(active) + wt71 + I(wt71 * wt71) + I(qsmk * smokeintensity),
  data = onesample
)
summary(std)
##
## Call:
## glm(formula = wt82_71 ~ qsmk + sex + race + age + I(age * age) +
##      as.factor(education) + smokeintensity + I(smokeintensity *
```

```
##      smokeintensity) + smokeyrs + I(smokeyrs * smokeyrs) + as.factor(exercise) +
##      as.factor(active) + wt71 + I(wt71 * wt71) + I(qsmk * smokeintensity),
##      data = onesample)
##
## Coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1.5881657    4.3130359   -0.368  0.712756
## qsmk              2.5595941    0.8091486    3.163  0.001590 **
## sex             -1.4302717    0.4689576   -3.050  0.002328 **
## race              0.5601096    0.5818888    0.963  0.335913
## age               0.3596353    0.1633188    2.202  0.027809 *
## I(age * age)     -0.0061010    0.0017261   -3.534  0.000421 ***
## as.factor(education)2    0.7904440    0.6070005    1.302  0.193038
## as.factor(education)3    0.5563124    0.5561016    1.000  0.317284
## as.factor(education)4    1.4915695    0.8322704    1.792  0.073301 .
## as.factor(education)5   -0.1949770    0.7413692   -0.263  0.792589
## smokeintensity      0.0491365    0.0517254    0.950  0.342287
## I(smokeintensity * smokeintensity) -0.0009907    0.0009380   -1.056  0.291097
## smokeyrs          0.1343686    0.0917122    1.465  0.143094
## I(smokeyrs * smokeyrs) -0.0018664    0.0015437   -1.209  0.226830
## as.factor(exercise)1     0.2959754    0.5351533    0.553  0.580298
## as.factor(exercise)2     0.3539128    0.5588587    0.633  0.526646
## as.factor(active)1      -0.9475695    0.4099344   -2.312  0.020935 *
## as.factor(active)2     -0.2613779    0.6845577   -0.382  0.702647
## wt71               0.0455018    0.0833709    0.546  0.585299
## I(wt71 * wt71)        -0.0009653    0.0005247   -1.840  0.066001 .
## I(qsmk * smokeintensity)  0.0466628    0.0351448    1.328  0.184463
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 53.5683)
##
##      Null deviance: 97176  on 1565  degrees of freedom
## Residual deviance: 82763  on 1545  degrees of freedom
##      (3321 observations deleted due to missingness)
## AIC: 10701
##
## Number of Fisher Scoring iterations: 2
onesample$predicted_meanY <- predict(std, onesample)

# estimate mean outcome in each of the groups interv=0, and interv=1
# this mean outcome is a weighted average of the mean outcomes in each combination
# of values of treatment and confounders, that is, the standardized outcome
mean(onesample[which(onesample$interv == -1), ]$predicted_meanY)
## [1] 2.56319
mean(onesample[which(onesample$interv == 0), ]$predicted_meanY)
## [1] 1.660267
mean(onesample[which(onesample$interv == 1), ]$predicted_meanY)
## [1] 5.178841
```

Program 13.4

- Computing the 95% confidence interval of the standardized means and their difference
- Data from NHEFS

```
#install.packages("boot") # install package if required
library(boot)

# function to calculate difference in means
standardization <- function(data, indices) {
  # create a dataset with 3 copies of each subject
  d <- data[indices, ] # 1st copy: equal to original one`
  d$interv <- -1
  d0 <- d # 2nd copy: treatment set to 0, outcome to missing
  d0$interv <- 0
  d0$qsmk <- 0
  d0$wt82_71 <- NA
  d1 <- d # 3rd copy: treatment set to 1, outcome to missing
  d1$interv <- 1
  d1$qsmk <- 1
  d1$wt82_71 <- NA
  d.onesample <- rbind(d, d0, d1) # combining datasets

  # linear model to estimate mean outcome conditional on treatment and confounders
  # parameters are estimated using original observations only (interv= -1)
  # parameter estimates are used to predict mean outcome for observations with set
  # treatment (interv=0 and interv=1)
  fit <- glm(
    wt82_71 ~ qsmk + sex + race + age + I(age * age) +
      as.factor(education) + smokeintensity +
      I(smokeintensity * smokeintensity) + smokeyrs + I(smokeyrs *
        smokeyrs) +
      as.factor(exercise) + as.factor(active) + wt71 + I(wt71 *
        wt71),
    data = d.onesample
  )

  d.onesample$predicted_meanY <- predict(fit, d.onesample)

  # estimate mean outcome in each of the groups interv=-1, interv=0, and interv=1
  return(c(
    mean(d.onesample$predicted_meanY[d.onesample$interv == -1]),
    mean(d.onesample$predicted_meanY[d.onesample$interv == 0]),
    mean(d.onesample$predicted_meanY[d.onesample$interv == 1]),
    mean(d.onesample$predicted_meanY[d.onesample$interv == 1]) -
      mean(d.onesample$predicted_meanY[d.onesample$interv == 0])
  ))
}

# bootstrap
results <- boot(data = nhefs,
  statistic = standardization,
  R = 5)
```

```

# generating confidence intervals
se <- c(sd(results$t[, 1]),
        sd(results$t[, 2]),
        sd(results$t[, 3]),
        sd(results$t[, 4]))
mean <- results$t0
ll <- mean - qnorm(0.975) * se
ul <- mean + qnorm(0.975) * se

bootstrap <-
  data.frame(cbind(
    c(
      "Observed",
      "No Treatment",
      "Treatment",
      "Treatment - No Treatment"
    ),
    mean,
    se,
    ll,
    ul
  ))
bootstrap

```

	V1	mean	se	ll
## 1	Observed	2.56188497106103	0.167193715782231	2.23419130968643
## 2	No Treatment	1.65212306626748	0.124944840854993	1.40723567813761
## 3	Treatment	5.1147448954934	0.422665219600556	4.2863362875586
## 4	Treatment - No Treatment	3.46262182922592	0.370544446195721	2.73636806001097
##	ul			
## 1		2.88957863243563		
## 2		1.89701045439736		
## 3		5.9431535034282		
## 4		4.18887559844087		

14. G-estimation of Structural Nested Models

Program 14.1

- Preprocessing, ranks of extreme observations, IP weights for censoring
- Data from NHEFS

```
library(here)

#install.packages("readxl") # install package if required
library("readxl")
nhefs <- read_excel(here("data", "NHEFS.xls"))

# some processing of the data
nhefs$cens <- ifelse(is.na(nhefs$wt82), 1, 0)

# ranking of extreme observations
#install.packages("Hmisc")
library(Hmisc)
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:base':
##
##      format.pval, units
describe(nhefs$wt82_71)
## nhefs$wt82_71
##           n missing distinct      Info      Mean      Gmd       .05       .10
##      1566      63      1510         1     2.638     8.337    -9.752    -6.292
##           .25      .50      .75      .90      .95
##     -1.478     2.604     6.690    11.117    14.739
##
## lowest : -41.2805 -30.5019 -30.0501 -29.0258 -25.9706
## highest: 34.0178  36.9693  37.6505  47.5113  48.5384

# estimation of denominator of ip weights for C
cw.denom <- glm(cens==0 ~ qsmk + sex + race + age + I(age^2)
                + as.factor(education) + smokeintensity + I(smokeintensity^2)
                + smokeyrs + I(smokeyrs^2) + as.factor(exercise)
                + as.factor(active) + wt71 + I(wt71^2),
                data = nhefs, family = binomial("logit"))
summary(cw.denom)
##
```

```
## Call:
## glm(formula = cens == 0 ~ qsmk + sex + race + age + I(age^2) +
##      as.factor(education) + smokeintensity + I(smokeintensity^2) +
##      smokeyrs + I(smokeyrs^2) + as.factor(exercise) + as.factor(active) +
##      wt71 + I(wt71^2), family = binomial("logit"), data = nhefs)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -4.0144661   2.5761058  -1.558  0.11915
## qsmk           -0.5168674   0.2877162  -1.796  0.07242 .
## sex            -0.0573131   0.3302775  -0.174  0.86223
## race            0.0122715   0.4524887   0.027  0.97836
## age             0.2697293   0.1174647   2.296  0.02166 *
## I(age^2)       -0.0028837   0.0011135  -2.590  0.00961 **
## as.factor(education)2  0.4407884  0.4193993   1.051  0.29326
## as.factor(education)3  0.1646881  0.3705471   0.444  0.65672
## as.factor(education)4 -0.1384470  0.5697969  -0.243  0.80802
## as.factor(education)5  0.3823818  0.5601808   0.683  0.49486
## smokeintensity    -0.0157119  0.0347319  -0.452  0.65100
## I(smokeintensity^2)    0.0001133  0.0006058   0.187  0.85171
## smokeyrs         -0.0785973  0.0749576  -1.049  0.29438
## I(smokeyrs^2)        0.0005569  0.0010318   0.540  0.58938
## as.factor(exercise)1   0.9714714  0.3878101   2.505  0.01224 *
## as.factor(exercise)2   0.5839890  0.3723133   1.569  0.11675
## as.factor(active)1     0.2474785  0.3254548   0.760  0.44701
## as.factor(active)2    -0.7065829  0.3964577  -1.782  0.07471 .
## wt71              0.0878871  0.0400115   2.197  0.02805 *
## I(wt71^2)          -0.0006351  0.0002257  -2.813  0.00490 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 533.36  on 1628  degrees of freedom
## Residual deviance: 465.36  on 1609  degrees of freedom
## AIC: 505.36
##
## Number of Fisher Scoring iterations: 7
nhefs$pd.c <- predict(cw.denom, nhefs, type="response")
nhefs$wc <- ifelse(nhefs$cens==0, 1/nhefs$pd.c, NA)
# observations with cens=1 only contribute to censoring models
```

Program 14.2

- G-estimation of a 1-parameter structural nested mean model
- Brute force search
- Data from NHEFS

G-estimation: Checking one possible value of ψ

```
#install.packages("geepack")
library("geepack")
```



```

nhefs$psi <- 3.446
nhefs$Hpsi <- nhefs$wt82_71 - nhefs$psi*nhefs$qsmk

fit <- geeglm(qsmk ~ sex + race + age + I(age*age) + as.factor(education)
+ smokeintensity + I(smokeintensity*smokeintensity) + smokeyrs
+ I(smokeyrs*smokeyrs) + as.factor(exercise) + as.factor(active)
+ wt71 + I(wt71*wt71) + Hpsi, family=binomial, data=nhefs,
weights=wc, id=seqn, corstr="independence")
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
summary(fit)
##
## Call:
## geeglm(formula = qsmk ~ sex + race + age + I(age * age) + as.factor(education) +
## smokeintensity + I(smokeintensity * smokeintensity) + smokeyrs +
## I(smokeyrs * smokeyrs) + as.factor(exercise) + as.factor(active) +
## wt71 + I(wt71 * wt71) + Hpsi, family = binomial, data = nhefs,
## weights = wc, id = seqn, corstr = "independence")
##
## Coefficients:
##


|                                    | Estimate   | Std.err   | Wald   | Pr(> W )     |
|------------------------------------|------------|-----------|--------|--------------|
| (Intercept)                        | -2.403e+00 | 1.329e+00 | 3.269  | 0.070604 .   |
| sex                                | -5.137e-01 | 1.536e-01 | 11.193 | 0.000821 *** |
| race                               | -8.609e-01 | 2.099e-01 | 16.826 | 4.10e-05 *** |
| age                                | 1.152e-01  | 5.020e-02 | 5.263  | 0.021779 *   |
| I(age * age)                       | -7.593e-04 | 5.296e-04 | 2.056  | 0.151619     |
| as.factor(education)2              | -2.894e-02 | 1.964e-01 | 0.022  | 0.882859     |
| as.factor(education)3              | 8.771e-02  | 1.726e-01 | 0.258  | 0.611329     |
| as.factor(education)4              | 6.637e-02  | 2.698e-01 | 0.061  | 0.805645     |
| as.factor(education)5              | 4.711e-01  | 2.247e-01 | 4.395  | 0.036036 *   |
| smokeintensity                     | -7.834e-02 | 1.464e-02 | 28.635 | 8.74e-08 *** |
| I(smokeintensity * smokeintensity) | 1.072e-03  | 2.650e-04 | 16.368 | 5.21e-05 *** |
| smokeyrs                           | -7.111e-02 | 2.639e-02 | 7.261  | 0.007047 **  |
| I(smokeyrs * smokeyrs)             | 8.153e-04  | 4.490e-04 | 3.298  | 0.069384 .   |
| as.factor(exercise)1               | 3.363e-01  | 1.828e-01 | 3.384  | 0.065844 .   |
| as.factor(exercise)2               | 3.800e-01  | 1.889e-01 | 4.049  | 0.044187 *   |
| as.factor(active)1                 | 3.412e-02  | 1.339e-01 | 0.065  | 0.798778     |
| as.factor(active)2                 | 2.135e-01  | 2.121e-01 | 1.012  | 0.314308     |
| wt71                               | -7.661e-03 | 2.562e-02 | 0.089  | 0.764963     |
| I(wt71 * wt71)                     | 8.655e-05  | 1.582e-04 | 0.299  | 0.584233     |
| Hpsi                               | -1.903e-06 | 8.839e-03 | 0.000  | 0.999828     |


## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = independence
## Estimated Scale Parameters:
##
##


|             | Estimate | Std.err |
|-------------|----------|---------|
| (Intercept) | 0.9969   | 0.06717 |


## Number of clusters: 1566 Maximum cluster size: 1

```

G-estimation: Checking multiple possible values of ψ

```
#install.packages("geepack")
grid <- seq(from = 2,to = 5, by = 0.1)
j = 0
Hpsi.coefs <- cbind(rep(NA,length(grid)), rep(NA, length(grid)))
colnames(Hpsi.coefs) <- c("Estimate", "p-value")

for (i in grid){
  psi = i
  j = j+1
  nhefs$Hpsi <- nhefs$wt82_71 - psi * nhefs$qsmk

  gest.fit <- geeglm(qsmk ~ sex + race + age + I(age*age) + as.factor(education)
    + smokeintensity + I(smokeintensity*smokeintensity) + smokeyrs
    + I(smokeyrs*smokeyrs) + as.factor(exercise) + as.factor(active)
    + wt71 + I(wt71*wt71) + Hpsi, family=binomial, data=nhefs,
    weights=wc, id=seqn, corstr="independence")
  Hpsi.coefs[j,1] <- summary(gest.fit)$coefficients["Hpsi", "Estimate"]
  Hpsi.coefs[j,2] <- summary(gest.fit)$coefficients["Hpsi", "Pr(>|W|)"]
}

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

Hpsi.coefs

```
##           Estimate p-value
## [1,] 0.0267219 0.001772
## [2,] 0.0248946 0.003580
## [3,] 0.0230655 0.006963
## [4,] 0.0212344 0.013026
## [5,] 0.0194009 0.023417
## [6,] 0.0175647 0.040430
## [7,] 0.0157254 0.067015
## [8,] 0.0138827 0.106626
## [9,] 0.0120362 0.162877
## [10,] 0.0101857 0.238979
## [11,] 0.0083308 0.337048
## [12,] 0.0064713 0.457433
## [13,] 0.0046069 0.598235
## [14,] 0.0027374 0.755204
## [15,] 0.0008624 0.922101
## [16,] -0.0010181 0.908537
## [17,] -0.0029044 0.744362
## [18,] -0.0047967 0.592188
## [19,] -0.0066950 0.457169
## [20,] -0.0085997 0.342360
## [21,] -0.0105107 0.248681
```

```
## [22,] -0.0124282 0.175239
## [23,] -0.0143523 0.119841
## [24,] -0.0162831 0.079580
## [25,] -0.0182206 0.051347
## [26,] -0.0201649 0.032218
## [27,] -0.0221160 0.019675
## [28,] -0.0240740 0.011706
## [29,] -0.0260389 0.006792
## [30,] -0.0280106 0.003847
## [31,] -0.0299893 0.002129
```

Program 14.3

- G-estimation for 2-parameter structural nested mean model
- Closed form estimator
- Data from NHEFS

G-estimation: Closed form estimator linear mean models

```
logit.est <- glm(qsmk ~ sex + race + age + I(age^2) + as.factor(education)
+ smokeintensity + I(smokeintensity^2) + smokeyrs
+ I(smokeyrs^2) + as.factor(exercise) + as.factor(active)
+ wt71 + I(wt71^2), data = nhefs, weight = wc,
family = binomial())
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
summary(logit.est)
##
## Call:
## glm(formula = qsmk ~ sex + race + age + I(age^2) + as.factor(education) +
##      smokeintensity + I(smokeintensity^2) + smokeyrs + I(smokeyrs^2) +
##      as.factor(exercise) + as.factor(active) + wt71 + I(wt71^2),
##      family = binomial(), data = nhefs, weights = wc)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.40e+00   1.31e+00  -1.83  0.06743 .
## sex             -5.14e-01   1.50e-01  -3.42  0.00062 ***
## race            -8.61e-01   2.06e-01  -4.18  2.9e-05 ***
## age              1.15e-01   4.95e-02   2.33  0.01992 *
## I(age^2)        -7.59e-04   5.14e-04  -1.48  0.13953
## as.factor(education)2 -2.89e-02  1.93e-01  -0.15  0.88079
## as.factor(education)3  8.77e-02  1.73e-01   0.51  0.61244
## as.factor(education)4  6.64e-02  2.66e-01   0.25  0.80301
## as.factor(education)5  4.71e-01  2.21e-01   2.13  0.03314 *
## smokeintensity    -7.83e-02  1.49e-02  -5.27  1.4e-07 ***
## I(smokeintensity^2)  1.07e-03  2.78e-04   3.85  0.00012 ***
## smokeyrs         -7.11e-02  2.71e-02  -2.63  0.00862 **
## I(smokeyrs^2)       8.15e-04  4.45e-04   1.83  0.06722 .
## as.factor(exercise)1  3.36e-01  1.75e-01   1.92  0.05467 .
## as.factor(exercise)2  3.80e-01  1.82e-01   2.09  0.03637 *
## as.factor(active)1    3.41e-02  1.30e-01   0.26  0.79337
## as.factor(active)2    2.13e-01  2.06e-01   1.04  0.30033
```

```
## wt71 -7.66e-03 2.46e-02 -0.31 0.75530
## I(wt71^2) 8.66e-05 1.51e-04 0.57 0.56586
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 1872.2 on 1565 degrees of freedom
## Residual deviance: 1755.6 on 1547 degrees of freedom
## (63 observations deleted due to missingness)
## AIC: 1719
##
## Number of Fisher Scoring iterations: 4
nhefs$pqsmk <- predict(logit.est, nhefs, type = "response")
describe(nhefs$pqsmk)
## nhefs$pqsmk
## n missing distinct Info Mean Gmd .05 .10
## 1629 0 1629 1 0.2622 0.1302 0.1015 0.1261
## .25 .50 .75 .90 .95
## 0.1780 0.2426 0.3251 0.4221 0.4965
##
## lowest : 0.0514466 0.0515703 0.0543802 0.0558308 0.0593059
## highest: 0.672083 0.686432 0.713913 0.733299 0.78914
summary(nhefs$pqsmk)
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 0.0514 0.1780 0.2426 0.2622 0.3251 0.7891

# solve sum(w_c * H(psi) * (qsmk - E[qsmk | L])) = 0
# for a single psi and H(psi) = wt82_71 - psi * qsmk
# this can be solved as
# psi = sum(w_c * wt82_71 * (qsmk - pqsmk)) / sum(w_c * qsmk * (qsmk - pqsmk))

nhefs.c <- nhefs[which(!is.na(nhefs$wt82)),]
with(nhefs.c, sum(wc*wt82_71*(qsmk-pqsmk)) / sum(wc*qsmk*(qsmk - pqsmk)))
## [1] 3.446
```

G-estimation: Closed form estimator for 2-parameter model

```
diff = with(nhefs.c, qsmk - pqsmk)
diff2 = with(nhefs.c, wc * diff)

lhs = matrix(0,2,2)
lhs[1,1] = with(nhefs.c, sum(qsmk * diff2))
lhs[1,2] = with(nhefs.c, sum(qsmk * smokeintensity * diff2))
lhs[2,1] = with(nhefs.c, sum(qsmk * smokeintensity * diff2))
lhs[2,2] = with(nhefs.c, sum(qsmk * smokeintensity * smokeintensity * diff2))

rhs = matrix(0,2,1)
rhs[1] = with(nhefs.c, sum(wt82_71 * diff2))
rhs[2] = with(nhefs.c, sum(wt82_71 * smokeintensity * diff2))

psi = t(solve(lhs,rhs))
```

```
psi
##      [,1]      [,2]
## [1,] 2.859 0.03004
```

15. Outcome regression and propensity scores

Program 15.1

- Estimating the average causal effect within levels of confounders under the assumption of effect-measure modification by smoking intensity ONLY
- Data from NHEFS

```
library(here)

#install.packages("readxl") # install package if required
library("readxl")

nhefs <- read_excel(here("data", "NHEFS.xls"))
nhefs$cens <- ifelse(is.na(nhefs$wt82), 1, 0)

# regression on covariates, allowing for some effect modification
fit <- glm(wt82_71 ~ qsmk + sex + race + age + I(age*age) + as.factor(education)
          + smokeintensity + I(smokeintensity*smokeintensity) + smokeyrs
          + I(smokeyrs*smokeyrs) + as.factor(exercise) + as.factor(active)
          + wt71 + I(wt71*wt71) + I(qsmk*smokeintensity), data=nhefs)

summary(fit)
##
## Call:
## glm(formula = wt82_71 ~ qsmk + sex + race + age + I(age * age) +
##      as.factor(education) + smokeintensity + I(smokeintensity *
##      smokeintensity) + smokeyrs + I(smokeyrs * smokeyrs) + as.factor(exercise) +
##      as.factor(active) + wt71 + I(wt71 * wt71) + I(qsmk * smokeintensity),
##      data = nhefs)
##
## Coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.5881657    4.3130359   -0.368  0.712756
## qsmk             2.5595941    0.8091486    3.163  0.001590 **
## sex            -1.4302717    0.4689576   -3.050  0.002328 **
## race             0.5601096    0.5818888    0.963  0.335913
## age             0.3596353    0.1633188    2.202  0.027809 *
## I(age * age)   -0.0061010    0.0017261   -3.534  0.000421 ***
## as.factor(education)2    0.7904440    0.6070005    1.302  0.193038
## as.factor(education)3    0.5563124    0.5561016    1.000  0.317284
## as.factor(education)4    1.4915695    0.8322704    1.792  0.073301 .
## as.factor(education)5   -0.1949770    0.7413692   -0.263  0.792589
```

```

## smokeintensity          0.0491365  0.0517254   0.950  0.342287
## I(smokeintensity * smokeintensity) -0.0009907  0.0009380  -1.056  0.291097
## smokeyrs                0.1343686  0.0917122   1.465  0.143094
## I(smokeyrs * smokeyrs)    -0.0018664  0.0015437  -1.209  0.226830
## as.factor(exercise)1     0.2959754  0.5351533   0.553  0.580298
## as.factor(exercise)2     0.3539128  0.5588587   0.633  0.526646
## as.factor(active)1      -0.9475695  0.4099344  -2.312  0.020935 *
## as.factor(active)2     -0.2613779  0.6845577  -0.382  0.702647
## wt71                    0.0455018  0.0833709   0.546  0.585299
## I(wt71 * wt71)          -0.0009653  0.0005247  -1.840  0.066001 .
## I(qsmk * smokeintensity)  0.0466628  0.0351448   1.328  0.184463
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 53.5683)
##
## Null deviance: 97176 on 1565 degrees of freedom
## Residual deviance: 82763 on 1545 degrees of freedom
## (63 observations deleted due to missingness)
## AIC: 10701
##
## Number of Fisher Scoring iterations: 2

# (step 1) build the contrast matrix with all zeros
# this function builds the blank matrix
# install.packages("multcomp") # install packages if necessary
library("multcomp")
## Loading required package: mvtnorm
## Loading required package: survival
## Loading required package: TH.data
## Loading required package: MASS
##
## Attaching package: 'TH.data'
## The following object is masked from 'package:MASS':
##
## geyser
makeContrastMatrix <- function(model, nrow, names) {
  m <- matrix(0, nrow = nrow, ncol = length(coef(model)))
  colnames(m) <- names(coef(model))
  rownames(m) <- names
  return(m)
}
K1 <-
  makeContrastMatrix(
    fit,
    2,
    c(
      'Effect of Quitting Smoking at Smokeintensity of 5',
      'Effect of Quitting Smoking at Smokeintensity of 40'
    )
  )
# (step 2) fill in the relevant non-zero elements

```



```

K1[1:2, 'qsmk'] <- 1
K1[1:2, 'I(qsmk * smokeintensity)'] <- c(5, 40)

# (step 3) check the contrast matrix
K1
##
## (Intercept) qsmk sex race
## Effect of Quitting Smoking at Smokeintensity of 5 0 1 0 0
## Effect of Quitting Smoking at Smokeintensity of 40 0 1 0 0
##
## age I(age * age)
## Effect of Quitting Smoking at Smokeintensity of 5 0 0
## Effect of Quitting Smoking at Smokeintensity of 40 0 0
##
## as.factor(education)2
## Effect of Quitting Smoking at Smokeintensity of 5 0
## Effect of Quitting Smoking at Smokeintensity of 40 0
##
## as.factor(education)3
## Effect of Quitting Smoking at Smokeintensity of 5 0
## Effect of Quitting Smoking at Smokeintensity of 40 0
##
## as.factor(education)4
## Effect of Quitting Smoking at Smokeintensity of 5 0
## Effect of Quitting Smoking at Smokeintensity of 40 0
##
## as.factor(education)5
## Effect of Quitting Smoking at Smokeintensity of 5 0
## Effect of Quitting Smoking at Smokeintensity of 40 0
##
## smokeintensity
## Effect of Quitting Smoking at Smokeintensity of 5 0
## Effect of Quitting Smoking at Smokeintensity of 40 0
##
## I(smokeintensity * smokeintensity)
## Effect of Quitting Smoking at Smokeintensity of 5 0
## Effect of Quitting Smoking at Smokeintensity of 40 0
##
## smokeyr
## Effect of Quitting Smoking at Smokeintensity of 5 0
## Effect of Quitting Smoking at Smokeintensity of 40 0
##
## I(smokeyr * smokeyr)
## Effect of Quitting Smoking at Smokeintensity of 5 0
## Effect of Quitting Smoking at Smokeintensity of 40 0
##
## as.factor(exercise)1
## Effect of Quitting Smoking at Smokeintensity of 5 0
## Effect of Quitting Smoking at Smokeintensity of 40 0
##
## as.factor(exercise)2
## Effect of Quitting Smoking at Smokeintensity of 5 0
## Effect of Quitting Smoking at Smokeintensity of 40 0
##
## as.factor(active)1
## Effect of Quitting Smoking at Smokeintensity of 5 0
## Effect of Quitting Smoking at Smokeintensity of 40 0
##
## as.factor(active)2 wt71
## Effect of Quitting Smoking at Smokeintensity of 5 0 0
## Effect of Quitting Smoking at Smokeintensity of 40 0 0
##
## I(wt71 * wt71)
## Effect of Quitting Smoking at Smokeintensity of 5 0
## Effect of Quitting Smoking at Smokeintensity of 40 0
##
## I(qsmk * smokeintensity)
## Effect of Quitting Smoking at Smokeintensity of 5 5

```

```
## Effect of Quitting Smoking at Smokeintensity of 40 40

# (step 4) estimate the contrasts, get tests and confidence intervals for them
estimates1 <- glht(fit, K1)
summary(estimates1)
##
## Simultaneous Tests for General Linear Hypotheses
##
## Fit: glm(formula = wt82_71 ~ qsmk + sex + race + age + I(age * age) +
## as.factor(education) + smokeintensity + I(smokeintensity *
## smokeintensity) + smokeyrs + I(smokeyrs * smokeyrs) + as.factor(exercise) +
## as.factor(active) + wt71 + I(wt71 * wt71) + I(qsmk * smokeintensity),
## data = nhefs)
##
## Linear Hypotheses:
##
## Effect of Quitting Smoking at Smokeintensity of 5 == 0 2.7929 0.6683
## Effect of Quitting Smoking at Smokeintensity of 40 == 0 4.4261 0.8478
## z value Pr(>|z|)
## Effect of Quitting Smoking at Smokeintensity of 5 == 0 4.179 5.84e-05 ***
## Effect of Quitting Smoking at Smokeintensity of 40 == 0 5.221 3.56e-07 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
confint(estimates1)
##
## Simultaneous Confidence Intervals
##
## Fit: glm(formula = wt82_71 ~ qsmk + sex + race + age + I(age * age) +
## as.factor(education) + smokeintensity + I(smokeintensity *
## smokeintensity) + smokeyrs + I(smokeyrs * smokeyrs) + as.factor(exercise) +
## as.factor(active) + wt71 + I(wt71 * wt71) + I(qsmk * smokeintensity),
## data = nhefs)
##
## Quantile = 2.2281
## 95% family-wise confidence level
##
## Linear Hypotheses:
##
## Effect of Quitting Smoking at Smokeintensity of 5 == 0 2.7929 1.3039 4.2819
## Effect of Quitting Smoking at Smokeintensity of 40 == 0 4.4261 2.5372 6.3151

# regression on covariates, not allowing for effect modification
fit2 <- glm(wt82_71 ~ qsmk + sex + race + age + I(age*age) + as.factor(education)
+ smokeintensity + I(smokeintensity*smokeintensity) + smokeyrs
+ I(smokeyrs*smokeyrs) + as.factor(exercise) + as.factor(active)
+ wt71 + I(wt71*wt71), data=nhefs)

summary(fit2)
##
## Call:
```

```
## glm(formula = wt82_71 ~ qsmk + sex + race + age + I(age * age) +
##      as.factor(education) + smokeintensity + I(smokeintensity *
##      smokeintensity) + smokeyrs + I(smokeyrs * smokeyrs) + as.factor(exercise) +
##      as.factor(active) + wt71 + I(wt71 * wt71), data = nhefs)
##
## Coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1.6586176    4.3137734   -0.384 0.700666
## qsmk              3.4626218    0.4384543    7.897 5.36e-15 ***
## sex             -1.4650496    0.4683410   -3.128 0.001792 **
## race              0.5864117    0.5816949    1.008 0.313560
## age              0.3626624    0.1633431    2.220 0.026546 *
## I(age * age)     -0.0061377    0.0017263   -3.555 0.000389 ***
## as.factor(education)2    0.8185263    0.6067815    1.349 0.177546
## as.factor(education)3    0.5715004    0.5561211    1.028 0.304273
## as.factor(education)4    1.5085173    0.8323778    1.812 0.070134 .
## as.factor(education)5   -0.1708264    0.7413289   -0.230 0.817786
## smokeintensity      0.0651533    0.0503115    1.295 0.195514
## I(smokeintensity * smokeintensity) -0.0010468    0.0009373   -1.117 0.264261
## smokeyrs          0.1333931    0.0917319    1.454 0.146104
## I(smokeyrs * smokeyrs) -0.0018270    0.0015438   -1.183 0.236818
## as.factor(exercise)1     0.3206824    0.5349616    0.599 0.548961
## as.factor(exercise)2     0.3628786    0.5589557    0.649 0.516300
## as.factor(active)1      -0.9429574    0.4100208   -2.300 0.021593 *
## as.factor(active)2      -0.2580374    0.6847219   -0.377 0.706337
## wt71              0.0373642    0.0831658    0.449 0.653297
## I(wt71 * wt71)        -0.0009158    0.0005235   -1.749 0.080426 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 53.59474)
##
##      Null deviance: 97176  on 1565  degrees of freedom
## Residual deviance: 82857  on 1546  degrees of freedom
##      (63 observations deleted due to missingness)
## AIC: 10701
##
## Number of Fisher Scoring iterations: 2
```

Program 15.2

- Estimating and plotting the propensity score
- Data from NHEFS

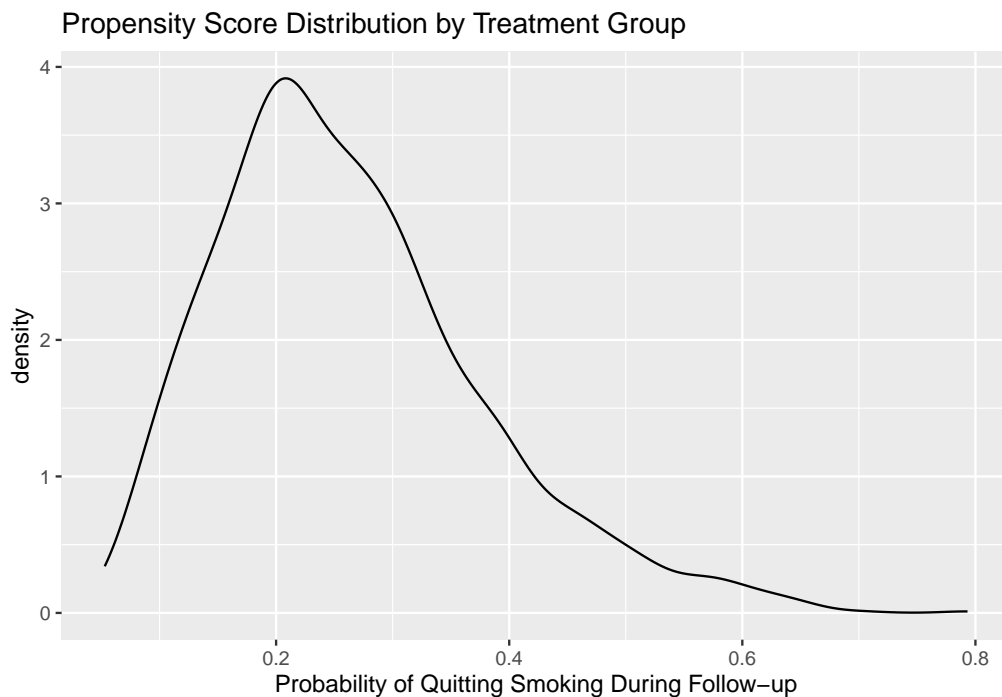
```
fit3 <- glm(qsmk ~ sex + race + age + I(age*age) + as.factor(education)
+ smokeintensity + I(smokeintensity*smokeintensity) + smokeyrs
+ I(smokeyrs*smokeyrs) + as.factor(exercise) + as.factor(active)
+ wt71 + I(wt71*wt71), data=nhefs, family=binomial())
summary(fit3)
##
## Call:
## glm(formula = qsmk ~ sex + race + age + I(age * age) + as.factor(education) +
```

```
##      smokeintensity + I(smokeintensity * smokeintensity) + smokeyrs +
##      I(smokeyrs * smokeyrs) + as.factor(exercise) + as.factor(active) +
##      wt71 + I(wt71 * wt71), family = binomial(), data = nhefs)
##
## Coefficients:
##
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.9889022    1.2412792  -1.602  0.109089
## sex              -0.5075218    0.1482316  -3.424  0.000617 ***
## race             -0.8502312    0.2058720  -4.130  3.63e-05 ***
## age               0.1030132    0.0488996   2.107  0.035150 *
## I(age * age)     -0.0006052    0.0005074  -1.193  0.232973
## as.factor(education)2 -0.0983203    0.1906553  -0.516  0.606066
## as.factor(education)3  0.0156987    0.1707139   0.092  0.926730
## as.factor(education)4 -0.0425260    0.2642761  -0.161  0.872160
## as.factor(education)5  0.3796632    0.2203947   1.723  0.084952 .
## smokeintensity    -0.0651561    0.0147589  -4.415  1.01e-05 ***
## I(smokeintensity * smokeintensity) 0.0008461    0.0002758   3.067  0.002160 **
## smokeyrs          -0.0733708    0.0269958  -2.718  0.006571 **
## I(smokeyrs * smokeyrs) 0.0008384    0.0004435   1.891  0.058669 .
## as.factor(exercise)1  0.2914117    0.1735543   1.679  0.093136 .
## as.factor(exercise)2  0.3550517    0.1799293   1.973  0.048463 *
## as.factor(active)1    0.0108754    0.1298320   0.084  0.933243
## as.factor(active)2    0.0683123    0.2087269   0.327  0.743455
## wt71              -0.0128478    0.0222829  -0.577  0.564226
## I(wt71 * wt71)       0.0001209    0.0001352   0.895  0.370957
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1876.3  on 1628  degrees of freedom
## Residual deviance: 1766.7  on 1610  degrees of freedom
## AIC: 1804.7
##
## Number of Fisher Scoring iterations: 4
nhefs$ps <- predict(fit3, nhefs, type="response")

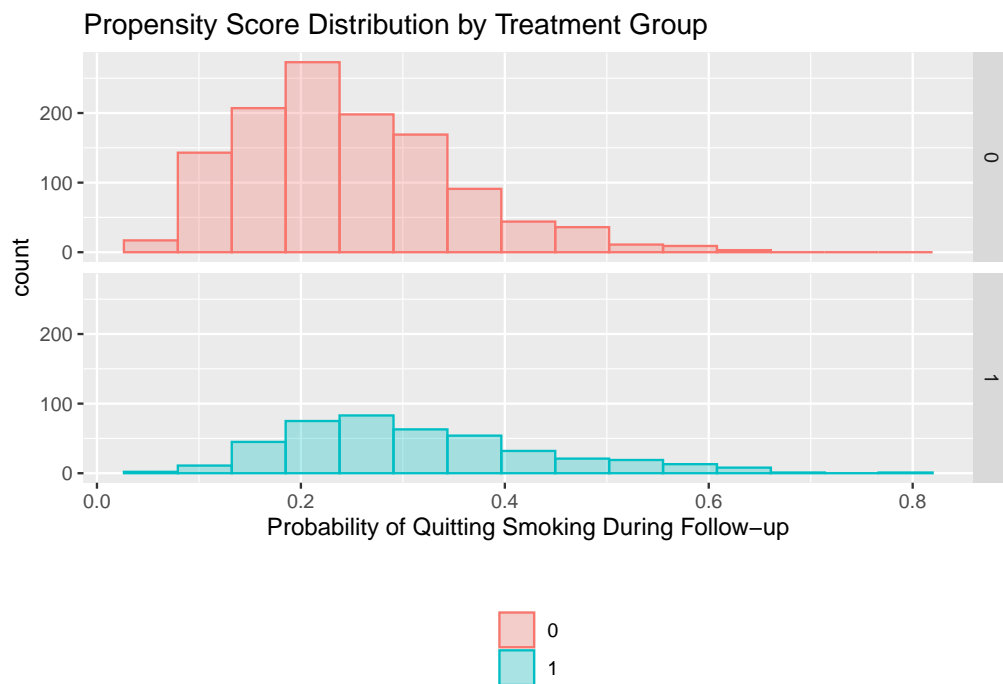
summary(nhefs$ps[nhefs$qsmk==0])
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.05298 0.16949 0.22747 0.24504 0.30441 0.65788
summary(nhefs$ps[nhefs$qsmk==1])
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.06248 0.22046 0.28897 0.31240 0.38122 0.79320

# # plotting the estimated propensity score
# install.packages("ggplot2") # install packages if necessary
# install.packages("dplyr")
library("ggplot2")
library("dplyr")
##
## Attaching package: 'dplyr'
## The following object is masked from 'package:MASS':
```

```
##
##   select
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
ggplot(nhefs, aes(x = ps, fill = qsmk)) + geom_density(alpha = 0.2) +
  xlab('Probability of Quitting Smoking During Follow-up') +
  ggtitle('Propensity Score Distribution by Treatment Group') +
  scale_fill_discrete('') +
  theme(legend.position = 'bottom', legend.direction = 'vertical')
## Warning: The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
## the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
## variable into a factor?
```



```
# alternative plot with histograms
nhefs <- nhefs %>% mutate(qsmklabel = ifelse(qsmk == 1,
  yes = 'Quit Smoking 1971-1982',
  no = 'Did Not Quit Smoking 1971-1982'))
ggplot(nhefs, aes(x = ps, fill = as.factor(qsmk), color = as.factor(qsmk))) +
  geom_histogram(alpha = 0.3, position = 'identity', bins=15) +
  facet_grid(as.factor(qsmk) ~ .) +
  xlab('Probability of Quitting Smoking During Follow-up') +
  ggtitle('Propensity Score Distribution by Treatment Group') +
  scale_fill_discrete('') +
  scale_color_discrete('') +
  theme(legend.position = 'bottom', legend.direction = 'vertical')
```



```
# attempt to reproduce plot from the book
nhefs %>%
  mutate(ps.grp = round(ps/0.05) * 0.05) %>%
  group_by(qsmk, ps.grp) %>%
  summarize(n = n()) %>%
  ungroup() %>%
  mutate(n2 = ifelse(qsmk == 0, yes = n, no = -1*n)) %>%
  ggplot(aes(x = ps.grp, y = n2, fill = as.factor(qsmk))) +
  geom_bar(stat = 'identity', position = 'identity') +
  geom_text(aes(label = n, x = ps.grp, y = n2 + ifelse(qsmk == 0, 8, -8))) +
  xlab('Probability of Quitting Smoking During Follow-up') +
  ylab('N') +
  ggtitle('Propensity Score Distribution by Treatment Group') +
  scale_fill_discrete('') +
  scale_x_continuous(breaks = seq(0, 1, 0.05)) +
  theme(legend.position = 'bottom', legend.direction = 'vertical',
        axis.ticks.y = element_blank(),
        axis.text.y = element_blank())
```

Program 15.3

- Stratification on the propensity score
- Data from NHEFS

```
# calculation of deciles
nhefs$ps.dec <- cut(nhefs$ps,
                    breaks=c(quantile(nhefs$ps, probs=seq(0,1,0.1))),
                    labels=seq(1:10),
                    include.lowest=TRUE)

#install.packages("psych") # install package if required
library("psych")
```

```
##
## Attaching package: 'psych'
## The following objects are masked from 'package:ggplot2':
##
##      %+, alpha
describeBy(nhefs$ps, list(nhefs$ps.dec, nehs$qsmk))
##
## Descriptive statistics by group
## : 1
## : 0
##      vars    n mean    sd median trimmed  mad   min   max range  skew kurtosis se
## X1      1 151  0.1 0.02   0.11      0.1 0.02 0.05 0.13  0.08 -0.55   -0.53  0
## -----
## : 2
## : 0
##      vars    n mean    sd median trimmed  mad   min   max range  skew kurtosis se
## X1      1 136  0.15 0.01   0.15      0.15 0.01 0.13 0.17  0.04 -0.04   -1.23  0
## -----
## : 3
## : 0
##      vars    n mean    sd median trimmed  mad   min   max range  skew kurtosis se
## X1      1 134  0.18 0.01   0.18      0.18 0.01 0.17 0.19  0.03 -0.08   -1.34  0
## -----
## : 4
## : 0
##      vars    n mean    sd median trimmed  mad   min   max range  skew kurtosis se
## X1      1 129  0.21 0.01   0.21      0.21 0.01 0.19 0.22  0.02 -0.04   -1.13  0
## -----
## : 5
## : 0
##      vars    n mean    sd median trimmed  mad   min   max range  skew kurtosis se
## X1      1 120  0.23 0.01   0.23      0.23 0.01 0.22 0.25  0.03 0.24   -1.22  0
## -----
## : 6
## : 0
##      vars    n mean    sd median trimmed  mad   min   max range  skew kurtosis se
## X1      1 117  0.26 0.01   0.26      0.26 0.01 0.25 0.27  0.03 -0.11   -1.29  0
## -----
## : 7
## : 0
##      vars    n mean    sd median trimmed  mad   min   max range  skew kurtosis se
## X1      1 120  0.29 0.01   0.29      0.29 0.01 0.27 0.31  0.03 -0.23   -1.19  0
## -----
## : 8
## : 0
##      vars    n mean    sd median trimmed  mad   min   max range  skew kurtosis se
## X1      1 112  0.33 0.01   0.33      0.33 0.02 0.31 0.35  0.04 0.15   -1.1  0
## -----
## : 9
## : 0
##      vars    n mean    sd median trimmed  mad   min   max range  skew kurtosis se
## X1      1  96  0.38 0.02   0.38      0.38 0.02 0.35 0.42  0.06 0.13   -1.15  0
```

```

## -----
## : 10
## : 0
##   vars  n mean   sd median trimmed  mad  min  max range skew kurtosis  se
## X1     1 86 0.49 0.06   0.47   0.48 0.05 0.42 0.66 0.24 1.1    0.47 0.01
## -----
## : 1
## : 1
##   vars  n mean   sd median trimmed  mad  min  max range skew kurtosis  se
## X1     1 12 0.1 0.02   0.11   0.1 0.03 0.06 0.13 0.07 -0.5   -1.36 0.01
## -----
## : 2
## : 1
##   vars  n mean   sd median trimmed  mad  min  max range skew kurtosis  se
## X1     1 27 0.15 0.01   0.15   0.15 0.01 0.13 0.17 0.03 -0.03   -1.34 0
## -----
## : 3
## : 1
##   vars  n mean   sd median trimmed  mad  min  max range skew kurtosis  se
## X1     1 29 0.18 0.01   0.18   0.18 0.01 0.17 0.19 0.03 0.01   -1.34 0
## -----
## : 4
## : 1
##   vars  n mean   sd median trimmed  mad  min  max range skew kurtosis  se
## X1     1 34 0.21 0.01   0.21   0.21 0.01 0.19 0.22 0.02 -0.31   -1.23 0
## -----
## : 5
## : 1
##   vars  n mean   sd median trimmed  mad  min  max range skew kurtosis  se
## X1     1 43 0.23 0.01   0.23   0.23 0.01 0.22 0.25 0.03 0.11   -1.23 0
## -----
## : 6
## : 1
##   vars  n mean   sd median trimmed  mad  min  max range skew kurtosis  se
## X1     1 45 0.26 0.01   0.26   0.26 0.01 0.25 0.27 0.03 0.2    -1.12 0
## -----
## : 7
## : 1
##   vars  n mean   sd median trimmed  mad  min  max range skew kurtosis  se
## X1     1 43 0.29 0.01   0.29   0.29 0.01 0.27 0.31 0.03 0.16   -1.25 0
## -----
## : 8
## : 1
##   vars  n mean   sd median trimmed  mad  min  max range skew kurtosis  se
## X1     1 51 0.33 0.01   0.33   0.33 0.02 0.31 0.35 0.04 0.11   -1.19 0
## -----
## : 9
## : 1
##   vars  n mean   sd median trimmed  mad  min  max range skew kurtosis  se
## X1     1 67 0.38 0.02   0.38   0.38 0.03 0.35 0.42 0.06 0.19   -1.27 0
## -----
## : 10

```



```
## : 1
##      vars  n mean   sd median trimmed  mad   min   max range skew kurtosis   se
## X1      1 77 0.52 0.08   0.51    0.51 0.08 0.42 0.79   0.38 0.88    0.81 0.01

# function to create deciles easily
decile <- function(x) {
  return(factor(quantcut(x, seq(0, 1, 0.1), labels = FALSE)))
}

# regression on PS deciles, allowing for effect modification
for (deciles in c(1:10)) {
  print(t.test(wt82_71~qsmk, data=nhefs[which(nhefs$ps.dec==deciles),]))
}

##
## Welch Two Sample t-test
##
## data:  wt82_71 by qsmk
## t = 0.0060506, df = 11.571, p-value = 0.9953
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
## -5.283903  5.313210
## sample estimates:
## mean in group 0 mean in group 1
##      3.995205      3.980551
##
##
## Welch Two Sample t-test
##
## data:  wt82_71 by qsmk
## t = -3.1117, df = 37.365, p-value = 0.003556
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
## -6.849335 -1.448161
## sample estimates:
## mean in group 0 mean in group 1
##      2.904679      7.053426
##
##
## Welch Two Sample t-test
##
## data:  wt82_71 by qsmk
## t = -4.5301, df = 35.79, p-value = 6.317e-05
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
## -9.474961 -3.613990
## sample estimates:
## mean in group 0 mean in group 1
##      2.612094      9.156570
##
##
## Welch Two Sample t-test
##
```

```

## data: wt82_71 by qsmk
## t = -1.4117, df = 45.444, p-value = 0.1648
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
## -5.6831731 0.9985715
## sample estimates:
## mean in group 0 mean in group 1
## 3.474679 5.816979
##
##
## Welch Two Sample t-test
##
## data: wt82_71 by qsmk
## t = -3.1371, df = 74.249, p-value = 0.002446
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
## -6.753621 -1.507087
## sample estimates:
## mean in group 0 mean in group 1
## 2.098800 6.229154
##
##
## Welch Two Sample t-test
##
## data: wt82_71 by qsmk
## t = -2.1677, df = 50.665, p-value = 0.0349
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
## -8.7516605 -0.3350127
## sample estimates:
## mean in group 0 mean in group 1
## 1.847004 6.390340
##
##
## Welch Two Sample t-test
##
## data: wt82_71 by qsmk
## t = -3.3155, df = 84.724, p-value = 0.001348
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
## -6.904207 -1.727590
## sample estimates:
## mean in group 0 mean in group 1
## 1.560048 5.875946
##
##
## Welch Two Sample t-test
##
## data: wt82_71 by qsmk
## t = -2.664, df = 75.306, p-value = 0.009441
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:

```

```

## -6.2396014 -0.9005605
## sample estimates:
## mean in group 0 mean in group 1
##      0.2846851      3.8547661
##
##
## Welch Two Sample t-test
##
## data: wt82_71 by qsmk
## t = -1.9122, df = 129.12, p-value = 0.05806
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
## -4.68143608 0.07973698
## sample estimates:
## mean in group 0 mean in group 1
##      -0.8954482      1.4054014
##
##
## Welch Two Sample t-test
##
## data: wt82_71 by qsmk
## t = -1.5925, df = 142.72, p-value = 0.1135
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
## -5.0209284 0.5404697
## sample estimates:
## mean in group 0 mean in group 1
##      -0.5043766      1.7358528

# regression on PS deciles, not allowing for effect modification
fit.psdec <- glm(wt82_71 ~ qsmk + as.factor(ps.dec), data = nhefs)
summary(fit.psdec)
##
## Call:
## glm(formula = wt82_71 ~ qsmk + as.factor(ps.dec), data = nhefs)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.7505      0.6089   6.159 9.29e-10 ***
## qsmk              3.5005      0.4571   7.659 3.28e-14 ***
## as.factor(ps.dec)2 -0.7391      0.8611  -0.858  0.3908
## as.factor(ps.dec)3 -0.6182      0.8612  -0.718  0.4730
## as.factor(ps.dec)4 -0.5204      0.8584  -0.606  0.5444
## as.factor(ps.dec)5 -1.4884      0.8590  -1.733  0.0834 .
## as.factor(ps.dec)6 -1.6227      0.8675  -1.871  0.0616 .
## as.factor(ps.dec)7 -1.9853      0.8681  -2.287  0.0223 *
## as.factor(ps.dec)8 -3.4447      0.8749  -3.937 8.61e-05 ***
## as.factor(ps.dec)9 -5.1544      0.8848  -5.825 6.91e-09 ***
## as.factor(ps.dec)10 -4.8403      0.8828  -5.483 4.87e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##

```

```
## (Dispersion parameter for gaussian family taken to be 58.42297)
##
##      Null deviance: 97176  on 1565  degrees of freedom
## Residual deviance: 90848  on 1555  degrees of freedom
## (63 observations deleted due to missingness)
## AIC: 10827
##
## Number of Fisher Scoring iterations: 2
confint.lm(fit.psdec)
##              2.5 %      97.5 %
## (Intercept)    2.556098  4.94486263
## qsmk           2.603953  4.39700504
## as.factor(ps.dec)2 -2.428074  0.94982494
## as.factor(ps.dec)3 -2.307454  1.07103569
## as.factor(ps.dec)4 -2.204103  1.16333143
## as.factor(ps.dec)5 -3.173337  0.19657938
## as.factor(ps.dec)6 -3.324345  0.07893027
## as.factor(ps.dec)7 -3.688043 -0.28248110
## as.factor(ps.dec)8 -5.160862 -1.72860113
## as.factor(ps.dec)9 -6.889923 -3.41883853
## as.factor(ps.dec)10 -6.571789 -3.10873731
```

Program 15.4

- Standardization using the propensity score
- Data from NHEFS

```
#install.packages("boot") # install package if required
library("boot")
##
## Attaching package: 'boot'
## The following object is masked from 'package:psych':
##
##      logit
## The following object is masked from 'package:survival':
##
##      aml

# standardization by propensity score, agnostic regarding effect modification
std.ps <- function(data, indices) {
  d <- data[indices,] # 1st copy: equal to original one`
  # calculating propensity scores
  ps.fit <- glm(qsmk ~ sex + race + age + I(age*age)
    + as.factor(education) + smokeintensity
    + I(smokeintensity*smokeintensity) + smokeyrs
    + I(smokeyrs*smokeyrs) + as.factor(exercise)
    + as.factor(active) + wt71 + I(wt71*wt71),
    data=d, family=binomial())
  d$pscore <- predict(ps.fit, d, type="response")

  # create a dataset with 3 copies of each subject
  d$interv <- -1 # 1st copy: equal to original one`
```

```

d0 <- d # 2nd copy: treatment set to 0, outcome to missing
d0$interv <- 0
d0$qsmk <- 0
d0$wt82_71 <- NA
d1 <- d # 3rd copy: treatment set to 1, outcome to missing
d1$interv <- 1
d1$qsmk <- 1
d1$wt82_71 <- NA
d.onesample <- rbind(d, d0, d1) # combining datasets

std.fit <- glm(wt82_71 ~ qsmk + pscore + I(qsmk*pscore), data=d.onesample)
d.onesample$predicted_meanY <- predict(std.fit, d.onesample)

# estimate mean outcome in each of the groups interv=-1, interv=0, and interv=1
return(c(mean(d.onesample$predicted_meanY[d.onesample$interv==1]),
          mean(d.onesample$predicted_meanY[d.onesample$interv==0]),
          mean(d.onesample$predicted_meanY[d.onesample$interv==1]),
          mean(d.onesample$predicted_meanY[d.onesample$interv==1]) -
            mean(d.onesample$predicted_meanY[d.onesample$interv==0])))
}

# bootstrap
results <- boot(data=nhefs, statistic=std.ps, R=5)

# generating confidence intervals
se <- c(sd(results$t[,1]), sd(results$t[,2]),
        sd(results$t[,3]), sd(results$t[,4]))
mean <- results$t0
ll <- mean - qnorm(0.975)*se
ul <- mean + qnorm(0.975)*se

bootstrap <- data.frame(cbind(c("Observed", "No Treatment", "Treatment",
                                "Treatment - No Treatment"), mean, se, ll, ul))

bootstrap
##              V1              mean              se              ll
## 1      Observed 2.63384609228479 0.111607927294825 2.41509857439777
## 2      No Treatment 1.71983636149845 0.116141968455712 1.49220228623167
## 3      Treatment 5.35072300362985 0.295288817731863 4.77196755583798
## 4 Treatment - No Treatment 3.6308866421314 0.385151512975718 2.87600354810788
##              ul
## 1 2.85259361017181
## 2 1.94747043676523
## 3 5.92947845142171
## 4 4.38576973615491

# regression on the propensity score (linear term)
model6 <- glm(wt82_71 ~ qsmk + ps, data = dhefs) # p.qsmk
summary(model6)
##
## Call:
## glm(formula = wt82_71 ~ qsmk + ps, data = dhefs)
##
## Coefficients:

```

```
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.5945      0.4831  11.581 < 2e-16 ***
## qsmk        3.5506      0.4573   7.765 1.47e-14 ***
## ps        -14.8218      1.7576  -8.433 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 58.28455)
##
##      Null deviance: 97176  on 1565  degrees of freedom
## Residual deviance: 91099  on 1563  degrees of freedom
## (63 observations deleted due to missingness)
## AIC: 10815
##
## Number of Fisher Scoring iterations: 2

# standarization on the propensity score
# (step 1) create two new datasets, one with all treated and one with all untreated
treated <- nhefs
treated$qsmk <- 1

untreated <- nhefs
untreated$qsmk <- 0

# (step 2) predict values for everyone in each new dataset based on above model
treated$pred.y <- predict(model6, treated)
untreated$pred.y <- predict(model6, untreated)

# (step 3) compare mean weight loss had all been treated vs. that had all been untreated
mean1 <- mean(treated$pred.y, na.rm = TRUE)
mean0 <- mean(untreated$pred.y, na.rm = TRUE)
mean1
## [1] 5.250824
mean0
## [1] 1.700228
mean1 - mean0
## [1] 3.550596

# (step 4) bootstrap a confidence interval
# number of bootstraps
nboot <- 100
# set up a matrix to store results
boots <- data.frame(i = 1:nboot,
                    mean1 = NA,
                    mean0 = NA,
                    difference = NA)
# loop to perform the bootstrapping
nhefs <- subset(nhefs, !is.na(ps) & !is.na(wt82_71)) # p.qsmk
for(i in 1:nboot) {
  # sample with replacement
  sampl <- nhefs[sample(1:nrow(nhefs), nrow(nhefs), replace = TRUE), ]

```

```

# fit the model in the bootstrap sample
bootmod <- glm(wt82_71 ~ qsmk + ps, data = sampl) # ps

# create new datasets
sampl.treated <- sampl %>%
  mutate(qsmk = 1)

sampl.untreated <- sampl %>%
  mutate(qsmk = 0)

# predict values
sampl.treated$pred.y <- predict(bootmod, sampl.treated)
sampl.untreated$pred.y <- predict(bootmod, sampl.untreated)

# output results
boots[i, 'mean1'] <- mean(sampl.treated$pred.y, na.rm = TRUE)
boots[i, 'mean0'] <- mean(sampl.untreated$pred.y, na.rm = TRUE)
boots[i, 'difference'] <- boots[i, 'mean1'] - boots[i, 'mean0']

# once loop is done, print the results
if(i == nboot) {
  cat('95% CI for the causal mean difference\n')
  cat(mean(boots$difference) - 1.96*sd(boots$difference),
      ', ',
      mean(boots$difference) + 1.96*sd(boots$difference))
}
}
## 95% CI for the causal mean difference
## 2.653562 , 4.514129

```

A more flexible and elegant way to do this is to write a function to perform the model fitting, prediction, bootstrapping, and reporting all at once.

16. Instrumental variables estimation

Program 16.1

- Estimating the average causal using the standard IV estimator via the calculation of sample averages
- Data from NHEFS

```
library(here)

#install.packages("readxl") # install package if required
library("readxl")
nhefs <- read_excel(here("data", "NHEFS.xls"))

# some preprocessing of the data
nhefs$cens <- ifelse(is.na(nhefs$wt82), 1, 0)
summary(nhefs$price82)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
##  1.452   1.740   1.815   1.806   1.868   2.103       92

# for simplicity, ignore subjects with missing outcome or missing instrument
nhefs.iv <- nhefs[which(!is.na(nhefs$wt82) & !is.na(nhefs$price82)),]
nhefs.iv$highprice <- ifelse(nhefs.iv$price82>=1.5, 1, 0)

table(nhefs.iv$highprice, nhefs.iv$qsmk)
##
##           0      1
##  0    33      8
##  1 1065    370

t.test(wt82_71 ~ highprice, data=nhefs.iv)
##
## Welch Two Sample t-test
##
## data:  wt82_71 by highprice
## t = -0.10179, df = 41.644, p-value = 0.9194
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -3.130588  2.830010
## sample estimates:
## mean in group 0 mean in group 1
##      2.535729      2.686018
```

Program 16.2

- Estimating the average causal effect using the standard IV estimator via two-stage-least-squares regression
- Data from NHEFS

```
#install.packages("sem") # install package if required
library(sem)

model1 <- tsls(wt82_71 ~ qsmk, ~ highprice, data = nhefs.iv)
summary(model1)
##
## 2SLS Estimates
##
## Model Formula: wt82_71 ~ qsmk
##
## Instruments: ~highprice
##
## Residuals:
##      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
## -43.34863  -4.00206  -0.02712   0.00000   4.17040  46.47022
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.068164   5.085098  0.40671  0.68428
## qsmk         2.396270  19.840037  0.12078  0.90388
##
## Residual standard error: 7.8561141 on 1474 degrees of freedom
confint(model1) # note the wide confidence intervals
##              2.5 %    97.5 %
## (Intercept) -7.898445 12.03477
## qsmk        -36.489487 41.28203
```

Program 16.3

- Estimating the average causal using the standard IV estimator via additive marginal structural models
- Data from NHEFS
- G-estimation: Checking one possible value of psi
- See Chapter 14 for program that checks several values and computes 95% confidence intervals

```
nhefs.iv$psi <- 2.396
nhefs.iv$Hpsi <- nhefs.iv$wt82_71-nhefs.iv$psi*nhefs.iv$qsmk

#install.packages("geepack") # install package if required
library("geepack")
g.est <- geeglm(highprice ~ Hpsi, data=nhefs.iv, id=seqn, family=binomial(),
               corstr="independence")
summary(g.est)
##
## Call:
## geeglm(formula = highprice ~ Hpsi, family = binomial(), data = nhefs.iv,
##       id = seqn, corstr = "independence")
##
```

```
## Coefficients:
##           Estimate Std. err Wald Pr(>|W|)
## (Intercept) 3.555e+00 1.652e-01 463.1 <2e-16 ***
## Hpsi        2.748e-07 2.273e-02  0.0      1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = independence
## Estimated Scale Parameters:
##
##           Estimate Std. err
## (Intercept)          1  0.7607
## Number of clusters:   1476 Maximum cluster size: 1

beta <- coef(g.est)
SE <- coef(summary(g.est))[,2]
lcl <- beta-qnorm(0.975)*SE
ucl <- beta+qnorm(0.975)*SE
cbind(beta, lcl, ucl)
##           beta      lcl      ucl
## (Intercept) 3.555e+00 3.23152 3.87917
## Hpsi        2.748e-07 -0.04456 0.04456
```

Program 16.4

- Estimating the average causal using the standard IV estimator with alternative proposed instruments
- Data from NHEFS

```
summary(tsls(wt82_71 ~ qsmk, ~ ifelse(price82 >= 1.6, 1, 0), data = nhefs.iv))
##
## 2SLS Estimates
##
## Model Formula: wt82_71 ~ qsmk
##
## Instruments: ~ifelse(price82 >= 1.6, 1, 0)
##
## Residuals:
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    -55.6   -13.5     7.6     0.0   12.5   56.4
##
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -7.89      42.25  -0.187  0.852
## qsmk           41.28     164.95   0.250  0.802
##
## Residual standard error: 18.6055 on 1474 degrees of freedom
summary(tsls(wt82_71 ~ qsmk, ~ ifelse(price82 >= 1.7, 1, 0), data = nhefs.iv))
##
## 2SLS Estimates
##
## Model Formula: wt82_71 ~ qsmk
##
```

```

## Instruments: ~ifelse(price82 >= 1.7, 1, 0)
##
## Residuals:
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    -54.4   -13.4    -8.4     0.0   18.1   75.3
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    13.16     48.08   0.274   0.784
## qsmk          -40.91    187.74  -0.218   0.828
##
## Residual standard error: 20.591 on 1474 degrees of freedom
summary(tsls(wt82_71 ~ qsmk, ~ ifelse(price82 >= 1.8, 1, 0), data = nhefs.iv))
##
## 2SLS Estimates
##
## Model Formula: wt82_71 ~ qsmk
##
## Instruments: ~ifelse(price82 >= 1.8, 1, 0)
##
## Residuals:
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    -49.37   -8.31   -3.44     0.00    7.27   60.53
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)     8.086     7.288   1.110   0.267
## qsmk           -21.103    28.428  -0.742   0.458
##
## Residual standard error: 13.0188 on 1474 degrees of freedom
summary(tsls(wt82_71 ~ qsmk, ~ ifelse(price82 >= 1.9, 1, 0), data = nhefs.iv))
##
## 2SLS Estimates
##
## Model Formula: wt82_71 ~ qsmk
##
## Instruments: ~ifelse(price82 >= 1.9, 1, 0)
##
## Residuals:
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    -47.24   -6.33   -1.43     0.00    5.52   54.36
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)     5.963     6.067   0.983   0.326
## qsmk           -12.811    23.667  -0.541   0.588
##
## Residual standard error: 10.3637 on 1474 degrees of freedom

```

Program 16.5

- Estimating the average causal using the standard IV estimator
- Conditional on baseline covariates
- Data from NHEFS

```

model2 <- tsls(wt82_71 ~ qsmk + sex + race + age + smokeintensity + smokeyrs +
               as.factor(exercise) + as.factor(active) + wt71,
               ~ highprice + sex + race + age + smokeintensity + smokeyrs + as.factor(exercise) +
               as.factor(active) + wt71, data = nhefs.iv)
summary(model2)
##
## 2SLS Estimates
##
## Model Formula: wt82_71 ~ qsmk + sex + race + age + smokeintensity + smokeyrs +
##               as.factor(exercise) + as.factor(active) + wt71
##
## Instruments: ~highprice + sex + race + age + smokeintensity + smokeyrs + as.factor(exercise) +
##               as.factor(active) + wt71
##
## Residuals:
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## -42.23   -4.29    -0.62     0.00     3.87    46.74
##
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    17.280330   2.335402   7.399 2.3e-13 ***
## qsmk           -1.042295   29.987369  -0.035  0.9723
## sex            -1.644393    2.630831  -0.625  0.5320
## race           -0.183255    4.650386  -0.039  0.9686
## age            -0.163640    0.240548  -0.680  0.4964
## smokeintensity  0.005767    0.145504   0.040  0.9684
## smokeyrs        0.025836    0.161421   0.160  0.8729
## as.factor(exercise)1  0.498748    2.171239   0.230  0.8184
## as.factor(exercise)2  0.581834    2.183148   0.267  0.7899
## as.factor(active)1  -1.170145    0.607466  -1.926  0.0543 .
## as.factor(active)2  -0.512284    1.308451  -0.392  0.6955
## wt71            -0.097949    0.036271  -2.701  0.0070 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.7162 on 1464 degrees of freedom

```


17. Causal survival analysis

Program 17.1

- Nonparametric estimation of survival curves
- Data from NHEFS

```
library(here)

library("readxl")
nhefs <- read_excel(here("data","NHEFS.xls"))

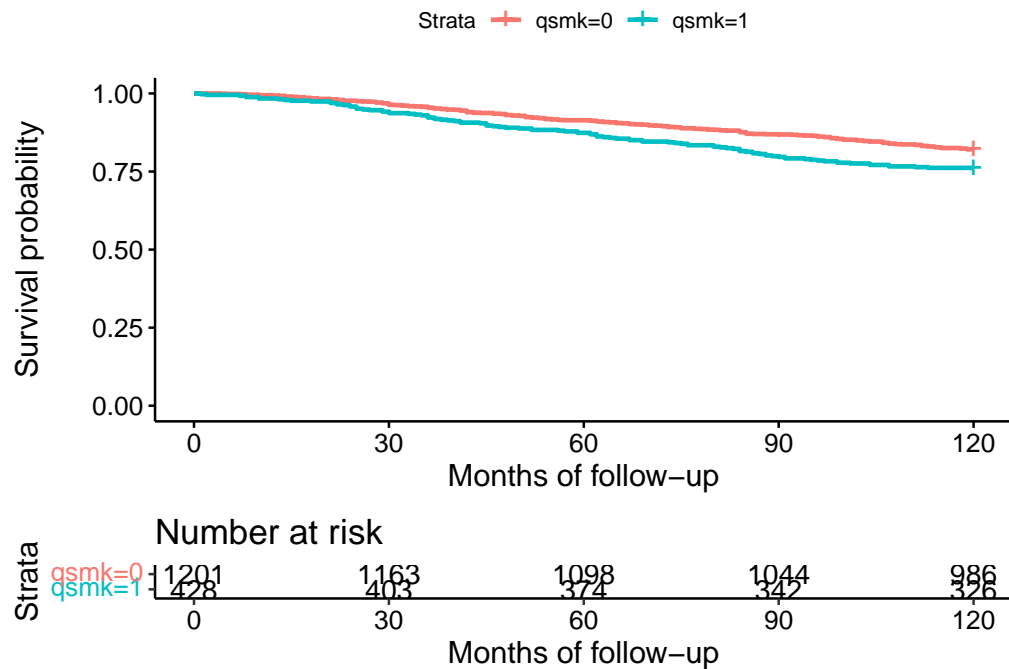
# some preprocessing of the data
nhefs$survtime <- ifelse(nhefs$death==0, 120,
                        (nhefs$yrdth-83)*12+nhefs$modth) # yrdth ranges from 83 to 92

table(nhefs$death, nhefs$qsmk)
##
##      0    1
## 0 985 326
## 1 216 102
summary(nhefs[which(nhefs$death==1),]$survtime)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.00  35.00   61.00  61.14   86.75  120.00

#install.packages("survival")
#install.packages("ggplot2") # for plots
#install.packages("survminer") # for plots
library("survival")
library("ggplot2")
library("survminer")
## Loading required package: ggpubr
##
## Attaching package: 'survminer'
## The following object is masked from 'package:survival':
##
##      myeloma
survdif(Surv(survtime, death) ~ qsmk, data=nhefs)
## Call:
## survdif(formula = Surv(survtime, death) ~ qsmk, data = nhefs)
##
##      N Observed Expected (O-E) ^2/E (O-E) ^2/V
## qsmk=0 1201      216   237.5    1.95    7.73
## qsmk=1  428      102    80.5    5.76    7.73
```

```
##
## Chisq= 7.7 on 1 degrees of freedom, p= 0.005

fit <- survfit(Surv(survtime, death) ~ qsmk, data=nhefs)
ggsurvplot(fit, data = dhefs, xlab="Months of follow-up",
  ylab="Survival probability",
  main="Product-Limit Survival Estimates", risk.table = TRUE)
```



Program 17.2

- Parametric estimation of survival curves via hazards model
- Data from NHEFS

```
# creation of person-month data
#install.packages("splitstackshape")
library("splitstackshape")
dhefs.surv <- expandRows(dhefs, "survtime", drop=F)
dhefs.surv$time <- sequence(rle(dhefs.surv$seqn)$lengths)-1
dhefs.surv$event <- ifelse(dhefs.surv$time==dhefs.surv$survtime-1 &
  dhefs.surv$death==1, 1, 0)
dhefs.surv$timesq <- dhefs.surv$time^2

# fit of parametric hazards model
hazards.model <- glm(event==0 ~ qsmk + I(qsmk*time) + I(qsmk*timesq) +
  time + timesq, family=binomial(), data=dhefs.surv)
summary(hazards.model)
##
## Call:
## glm(formula = event == 0 ~ qsmk + I(qsmk * time) + I(qsmk * timesq) +
##     time + timesq, family = binomial(), data = dhefs.surv)
##
## Coefficients:
```



```
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    6.996e+00  2.309e-01  30.292  <2e-16 ***
## qsmk          -3.355e-01  3.970e-01  -0.845   0.3981
## I(qsmk * time) -1.208e-02  1.503e-02  -0.804   0.4215
## I(qsmk * timesq) 1.612e-04  1.246e-04   1.293   0.1960
## time          -1.960e-02  8.413e-03  -2.329   0.0198 *
## timesq         1.256e-04  6.686e-05   1.878   0.0604 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##    Null deviance: 4655.3  on 176763  degrees of freedom
## Residual deviance: 4631.3  on 176758  degrees of freedom
## AIC: 4643.3
##
## Number of Fisher Scoring iterations: 9

# creation of dataset with all time points under each treatment level
qsmk0 <- data.frame(cbind(seq(0, 119),0,(seq(0, 119))^2))
qsmk1 <- data.frame(cbind(seq(0, 119),1,(seq(0, 119))^2))

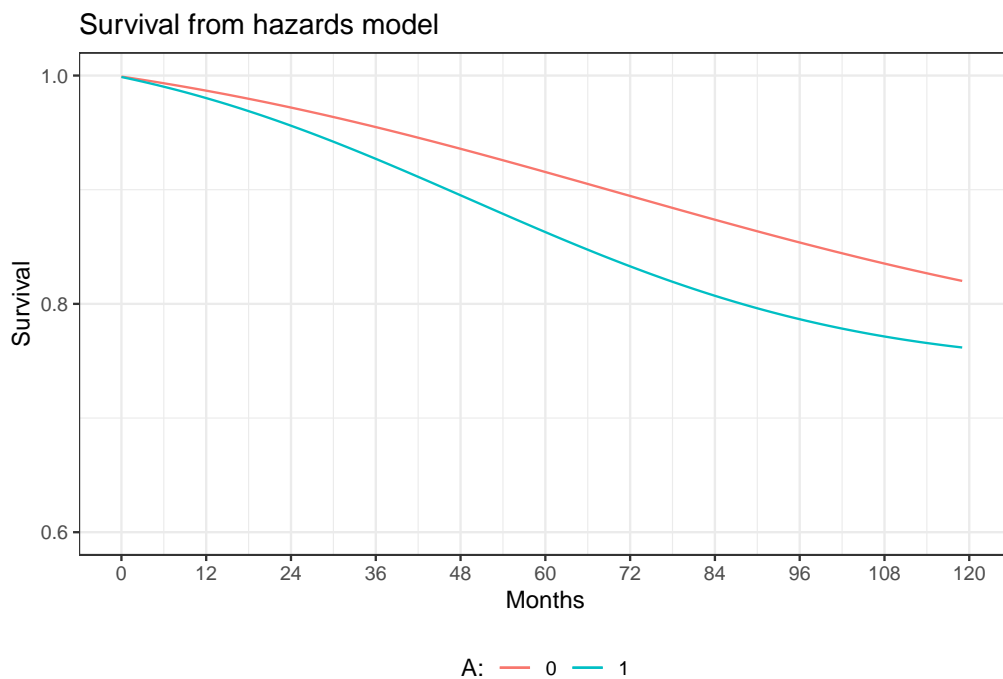
colnames(qsmk0) <- c("time", "qsmk", "timesq")
colnames(qsmk1) <- c("time", "qsmk", "timesq")

# assignment of estimated (1-hazard) to each person-month */
qsmk0$p.noevent0 <- predict(hazards.model, qsmk0, type="response")
qsmk1$p.noevent1 <- predict(hazards.model, qsmk1, type="response")

# computation of survival for each person-month
qsmk0$surv0 <- cumprod(qsmk0$p.noevent0)
qsmk1$surv1 <- cumprod(qsmk1$p.noevent1)

# some data management to plot estimated survival curves
hazards.graph <- merge(qsmk0, qsmk1, by=c("time", "timesq"))
hazards.graph$survdiff <- hazards.graph$surv1-hazards.graph$surv0

# plot
ggplot(hazards.graph, aes(x=time, y=surv)) +
  geom_line(aes(y = surv0, colour = "0")) +
  geom_line(aes(y = surv1, colour = "1")) +
  xlab("Months") +
  scale_x_continuous(limits = c(0, 120), breaks=seq(0,120,12)) +
  scale_y_continuous(limits=c(0.6, 1), breaks=seq(0.6, 1, 0.2)) +
  ylab("Survival") +
  ggtitle("Survival from hazards model") +
  labs(colour="A:") +
  theme_bw() +
  theme(legend.position="bottom")
```



Program 17.3

- Estimation of survival curves via IP weighted hazards model
- Data from NHEFS

```
# estimation of denominator of ip weights
p.denom <- glm(qsmk ~ sex + race + age + I(age*age) + as.factor(education)
               + smokeintensity + I(smokeintensity*smokeintensity)
               + smokeyrs + I(smokeyrs*smokeyrs) + as.factor(exercise)
               + as.factor(active) + wt71 + I(wt71*wt71),
               data=nhefs, family=binomial())
nhefs$pd.qsmk <- predict(p.denom, nhefs, type="response")

# estimation of numerator of ip weights
p.num <- glm(qsmk ~ 1, data=nhefs, family=binomial())
nhefs$pn.qsmk <- predict(p.num, nhefs, type="response")

# computation of estimated weights
nhefs$sw.a <- ifelse(nhefs$qsmk==1, nhefs$pn.qsmk/nhefs$pd.qsmk,
                    (1-nhefs$pn.qsmk)/(1-nhefs$pd.qsmk))
summary(nhefs$sw.a)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.3312  0.8640  0.9504  0.9991  1.0755  4.2054

# creation of person-month data
nhefs.ipw <- expandRows(nhefs, "survtime", drop=F)
nhefs.ipw$time <- sequence(rle(nhefs.ipw$seqn)$lengths)-1
nhefs.ipw$event <- ifelse(nhefs.ipw$time==nhefs.ipw$survtime-1 &
                          nhefs.ipw$death==1, 1, 0)
nhefs.ipw$timesq <- nhefs.ipw$time^2

# fit of weighted hazards model
```

```

ipw.model <- glm(event==0 ~ qsmk + I(qsmk*time) + I(qsmk*timesq) +
                 time + timesq, family=binomial(), weight=sw.a,
                 data=nhefs.ipw)
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
summary(ipw.model)
##
## Call:
## glm(formula = event == 0 ~ qsmk + I(qsmk * time) + I(qsmk * timesq) +
##      time + timesq, family = binomial(), data = nhefs.ipw, weights = sw.a)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   6.897e+00  2.208e-01  31.242  <2e-16 ***
## qsmk          1.794e-01  4.399e-01   0.408   0.6834
## I(qsmk * time) -1.895e-02  1.640e-02  -1.155   0.2481
## I(qsmk * timesq) 2.103e-04  1.352e-04   1.556   0.1198
## time         -1.889e-02  8.053e-03  -2.345   0.0190 *
## timesq        1.181e-04  6.399e-05   1.846   0.0649 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 4643.9  on 176763  degrees of freedom
## Residual deviance: 4626.2  on 176758  degrees of freedom
## AIC: 4633.5
##
## Number of Fisher Scoring iterations: 9

# creation of survival curves
ipw.qsmk0 <- data.frame(cbind(seq(0, 119),0,(seq(0, 119))^2))
ipw.qsmk1 <- data.frame(cbind(seq(0, 119),1,(seq(0, 119))^2))

colnames(ipw.qsmk0) <- c("time", "qsmk", "timesq")
colnames(ipw.qsmk1) <- c("time", "qsmk", "timesq")

# assignment of estimated (1-hazard) to each person-month */
ipw.qsmk0$p.noevent0 <- predict(ipw.model, ipw.qsmk0, type="response")
ipw.qsmk1$p.noevent1 <- predict(ipw.model, ipw.qsmk1, type="response")

# computation of survival for each person-month
ipw.qsmk0$urv0 <- cumprod(ipw.qsmk0$p.noevent0)
ipw.qsmk1$urv1 <- cumprod(ipw.qsmk1$p.noevent1)

# some data management to plot estimated survival curves
ipw.graph <- merge(ipw.qsmk0, ipw.qsmk1, by=c("time", "timesq"))
ipw.graph$urvdiff <- ipw.graph$urv1-ipw.graph$urv0

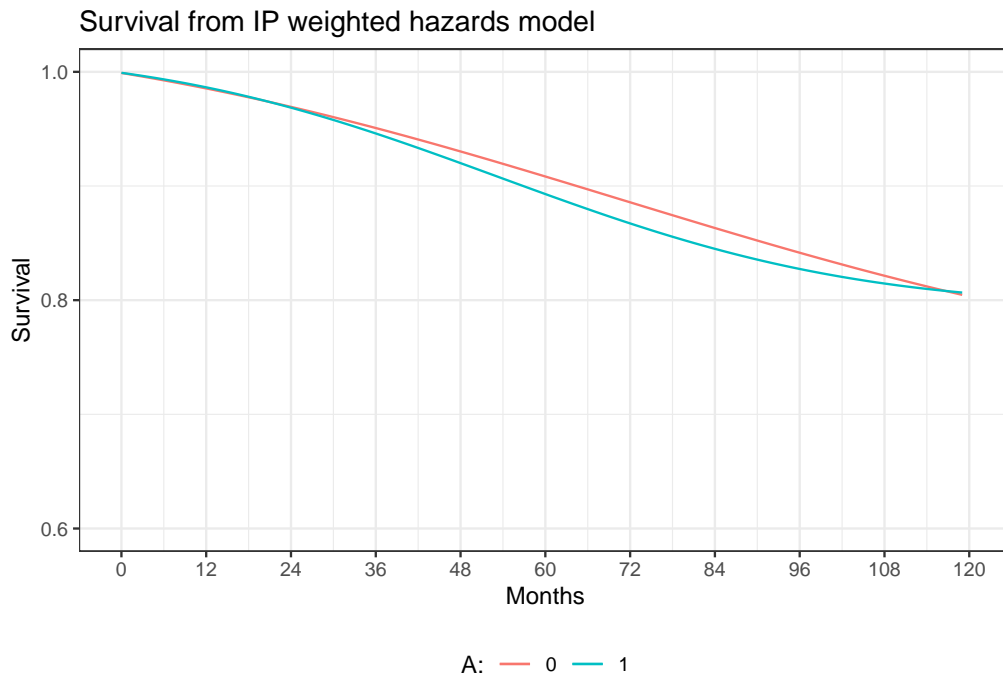
# plot
ggplot(ipw.graph, aes(x=time, y=surv)) +
  geom_line(aes(y = surv0, colour = "0")) +
  geom_line(aes(y = surv1, colour = "1")) +

```

```

xlab("Months") +
scale_x_continuous(limits = c(0, 120), breaks=seq(0,120,12)) +
scale_y_continuous(limits=c(0.6, 1), breaks=seq(0.6, 1, 0.2)) +
ylab("Survival") +
ggtitle("Survival from IP weighted hazards model") +
labs(colour="A:") +
theme_bw() +
theme(legend.position="bottom")

```



Program 17.4

- Estimating of survival curves via g-formula
- Data from NHEFS

```

# fit of hazards model with covariates
gf.model <- glm(event==0 ~ qsmk + I(qsmk*time) + I(qsmk*timesq)
               + time + timesq + sex + race + age + I(age*age)
               + as.factor(education) + smokeintensity
               + I(smokeintensity*smokeintensity) + smkintensity82_71
               + smokeyrs + I(smokeyrs*smokeyrs) + as.factor(exercise)
               + as.factor(active) + wt71 + I(wt71*wt71),
               data=nhefs.surv, family=binomial())
summary(gf.model)
##
## Call:
## glm(formula = event == 0 ~ qsmk + I(qsmk * time) + I(qsmk * timesq) +
##      time + timesq + sex + race + age + I(age * age) + as.factor(education) +
##      smokeintensity + I(smokeintensity * smokeintensity) + smkintensity82_71 +
##      smokeyrs + I(smokeyrs * smokeyrs) + as.factor(exercise) +
##      as.factor(active) + wt71 + I(wt71 * wt71), family = binomial(),
##      data = nhefs.surv)
##

```

```
## Coefficients:
##
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)      9.272e+00  1.379e+00   6.724 1.76e-11 ***
## qsmk             5.959e-02  4.154e-01   0.143 0.885924
## I(qsmk * time)   -1.485e-02  1.506e-02  -0.987 0.323824
## I(qsmk * timesq)  1.702e-04  1.245e-04   1.367 0.171643
## time            -2.270e-02  8.437e-03  -2.690 0.007142 **
## timesq           1.174e-04  6.709e-05   1.751 0.080020 .
## sex              4.368e-01  1.409e-01   3.101 0.001930 **
## race             -5.240e-02  1.734e-01  -0.302 0.762572
## age              -8.750e-02  5.907e-02  -1.481 0.138536
## I(age * age)      8.128e-05  5.470e-04   0.149 0.881865
## as.factor(education)2  1.401e-01  1.566e-01   0.895 0.370980
## as.factor(education)3  4.335e-01  1.526e-01   2.841 0.004502 **
## as.factor(education)4  2.350e-01  2.790e-01   0.842 0.399750
## as.factor(education)5  3.750e-01  2.386e-01   1.571 0.116115
## smokeintensity    -1.626e-03  1.430e-02  -0.114 0.909431
## I(smokeintensity * smokeintensity) -7.182e-05  2.390e-04  -0.301 0.763741
## smkintensity82_71   -1.686e-03  6.501e-03  -0.259 0.795399
## smokeyrs          -1.677e-02  3.065e-02  -0.547 0.584153
## I(smokeyrs * smokeyrs) -5.280e-05  4.244e-04  -0.124 0.900997
## as.factor(exercise)1  1.469e-01  1.792e-01   0.820 0.412300
## as.factor(exercise)2 -1.504e-01  1.762e-01  -0.854 0.393177
## as.factor(active)1   -1.601e-01  1.300e-01  -1.232 0.218048
## as.factor(active)2   -2.294e-01  1.877e-01  -1.222 0.221766
## wt71              6.222e-02  1.902e-02   3.271 0.001073 **
## I(wt71 * wt71)      -4.046e-04  1.129e-04  -3.584 0.000338 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 4655.3 on 176763 degrees of freedom
## Residual deviance: 4185.7 on 176739 degrees of freedom
## AIC: 4235.7
##
## Number of Fisher Scoring iterations: 10

# creation of dataset with all time points for
# each individual under each treatment level
gf.qsmk0 <- expandRows(nhefs, count=120, count.is.col=F)
gf.qsmk0$time <- rep(seq(0, 119), nrow(nhefs))
gf.qsmk0$timesq <- gf.qsmk0$time^2
gf.qsmk0$qsmk <- 0

gf.qsmk1 <- gf.qsmk0
gf.qsmk1$qsmk <- 1

gf.qsmk0$p.noevent0 <- predict(gf.model, gf.qsmk0, type="response")
gf.qsmk1$p.noevent1 <- predict(gf.model, gf.qsmk1, type="response")

#install.packages("dplyr")
```

```

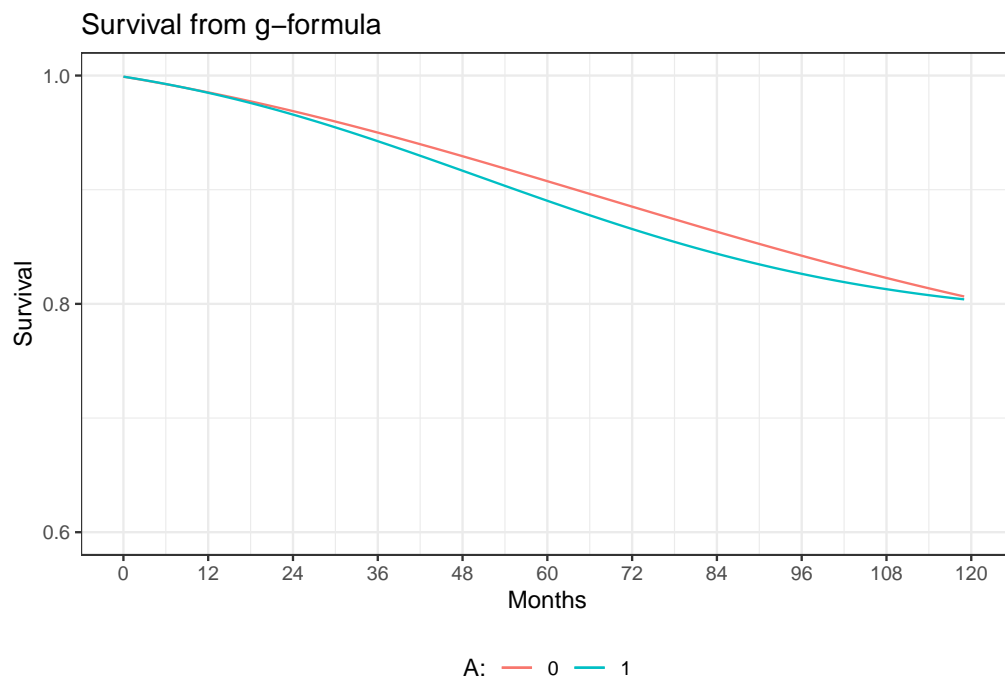
library("dplyr")
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
## filter, lag
## The following objects are masked from 'package:base':
##
## intersect, setdiff, setequal, union
gf.qsmk0.surv <- gf.qsmk0 %>% group_by(seqn) %>% mutate(surv0 = cumprod(p.noevent0))
gf.qsmk1.surv <- gf.qsmk1 %>% group_by(seqn) %>% mutate(surv1 = cumprod(p.noevent1))

gf.surv0 <-
  aggregate(gf.qsmk0.surv,
            by = list(gf.qsmk0.surv$time),
            FUN = mean)[c("qsmk", "time", "surv0")]
gf.surv1 <-
  aggregate(gf.qsmk1.surv,
            by = list(gf.qsmk1.surv$time),
            FUN = mean)[c("qsmk", "time", "surv1")]

gf.graph <- merge(gf.surv0, gf.surv1, by=c("time"))
gf.graph$survdifff <- gf.graph$surv1-gf.graph$surv0

# plot
ggplot(gf.graph, aes(x=time, y=surv)) +
  geom_line(aes(y = surv0, colour = "0")) +
  geom_line(aes(y = surv1, colour = "1")) +
  xlab("Months") +
  scale_x_continuous(limits = c(0, 120), breaks=seq(0,120,12)) +
  scale_y_continuous(limits=c(0.6, 1), breaks=seq(0.6, 1, 0.2)) +
  ylab("Survival") +
  ggtitle("Survival from g-formula") +
  labs(colour="A:") +
  theme_bw() +
  theme(legend.position="bottom")

```



Program 17.5

- Estimating of median survival time ratio via a structural nested AFT model
- Data from NHEFS

```
# some preprocessing of the data
nhefs <- read_excel(here("data", "NHEFS.xls"))
nhefs$survtime <-
  ifelse(nhefs$death == 0, NA, (nhefs$yrdeath - 83) * 12 + nhefs$month)
# * yrdeath ranges from 83 to 92

# model to estimate E[A|L]
modelA <- glm(qsmk ~ sex + race + age + I(age*age)
  + as.factor(education) + smokeintensity
  + I(smokeintensity*smokeintensity) + smokeyrs
  + I(smokeyrs*smokeyrs) + as.factor(exercise)
  + as.factor(active) + wt71 + I(wt71*wt71),
  data=nhefs, family=binomial())

nhefs$p.qsmk <- predict(modelA, nhefs, type="response")
d <- nhefs[!is.na(nhefs$survtime),] # select only those with observed death time
n <- nrow(d)

# define the estimating function that needs to be minimized
sumeef <- function(psi){

  # creation of delta indicator
  if (psi>=0){
    delta <- ifelse(d$qsmk==0 |
      (d$qsmk==1 & psi <= log(120/d$survtime)),
      1, 0)
  } else if (psi < 0) {
```

```

    delta <- ifelse(d$qsmk==1 |
                    (d$qsmk==0 & psi > log(d$survtime/120)), 1, 0)
  }

  smat <- delta*(d$qsmk-d$p.qsmk)
  sval <- sum(smat, na.rm=T)
  save <- sval/n
  smat <- smat - rep(save, n)

  # covariance
  sigma <- t(smat) %*% smat
  if (sigma == 0){
    sigma <- 1e-16
  }
  estimateq <- sval*solve(sigma)*t(sval)
  return(estimateq)
}

res <- optimize(sumeef, interval = c(-0.2,0.2))
psi1 <- res$minimum
objfunc <- as.numeric(res$objective)

# Use simple bisection method to find estimates of lower and upper 95% confidence bounds
incred <- 0.1
for_conf <- function(x){
  return(sumeef(x) - 3.84)
}

if (objfunc < 3.84){
  # Find estimate of where sumeef(x) > 3.84

  # Lower bound of 95% CI
  psi_low <- psi1
  test_low <- objfunc
  count_low <- 0
  while (test_low < 3.84 & count_low < 100){
    psi_low <- psi_low - increm
    test_low <- sumeef(psi_low)
    count_low <- count_low + 1
  }

  # Upper bound of 95% CI
  psi_high <- psi1
  test_high <- objfunc
  count_high <- 0
  while (test_high < 3.84 & count_high < 100){
    psi_high <- psi_high + increm
    test_high <- sumeef(psi_high)
    count_high <- count_high + 1
  }
}

```



```

# Better estimate using bisection method
if ((testhigh > 3.84) & (testlow > 3.84)){

  # Bisection method
  left <- psi1
  fleft <- objfunc - 3.84
  right <- psihigh
  fright <- testhigh - 3.84
  middle <- (left + right) / 2
  fmiddle <- for_conf(middle)
  count <- 0
  diff <- right - left

  while (!(abs(fmiddle) < 0.0001 | diff < 0.0001 | count > 100)){
    test <- fmiddle * fleft
    if (test < 0){
      right <- middle
      fright <- fmiddle
    } else {
      left <- middle
      fleft <- fmiddle
    }
    middle <- (left + right) / 2
    fmiddle <- for_conf(middle)
    count <- count + 1
    diff <- right - left
  }

  psi_high <- middle
  objfunc_high <- fmiddle + 3.84

  # lower bound of 95% CI
  left <- psilow
  fleft <- testlow - 3.84
  right <- psi1
  fright <- objfunc - 3.84
  middle <- (left + right) / 2
  fmiddle <- for_conf(middle)
  count <- 0
  diff <- right - left

  while(!(abs(fmiddle) < 0.0001 | diff < 0.0001 | count > 100)){
    test <- fmiddle * fleft
    if (test < 0){
      right <- middle
      fright <- fmiddle
    } else {
      left <- middle
      fleft <- fmiddle
    }
    middle <- (left + right) / 2
    fmiddle <- for_conf(middle)
  }

```

```
    diff <- right - left
    count <- count + 1
  }
  psi_low <- middle
  objfunc_low <- fmiddle + 3.84
  psi <- psi1
}
}
c(psi, psi_low, psi_high)
## [1] -0.05041591 -0.22312099  0.33312901
```

Session information: R

For reproducibility.

```
# install.packages("sessioninfo")
sessioninfo::session_info()

## - Session info -----
## setting value
## version R version 4.3.0 (2023-04-21 ucrt)
## os Windows 11 x64 (build 22621)
## system x86_64, mingw32
## ui RTerm
## language (EN)
## collate English_United Kingdom.utf8
## ctype English_United Kingdom.utf8
## tz Europe/London
## date 2023-06-16
## pandoc 3.1.1 @ C:/Program Files/RStudio/resources/app/bin/quarto/bin/tools/ (via rmarkdown)
##
## - Packages -----
## package * version date (UTC) lib source
## bookdown 0.34 2023-05-09 [1] CRAN (R 4.3.0)
## cli 3.6.1 2023-03-23 [1] CRAN (R 4.3.0)
## digest 0.6.31 2022-12-11 [1] CRAN (R 4.3.0)
## evaluate 0.21 2023-05-05 [1] CRAN (R 4.3.0)
## fastmap 1.1.1 2023-02-24 [1] CRAN (R 4.3.0)
## htmltools 0.5.5 2023-03-23 [1] CRAN (R 4.3.0)
## knitr 1.43 2023-05-25 [1] CRAN (R 4.3.0)
## rlang 1.1.1 2023-04-28 [1] CRAN (R 4.3.0)
## rmarkdown 2.22 2023-06-01 [1] CRAN (R 4.3.0)
## rstudioapi 0.14 2022-08-22 [1] CRAN (R 4.3.0)
## sessioninfo 1.2.2 2021-12-06 [1] CRAN (R 4.3.0)
## xfun 0.39 2023-04-20 [1] CRAN (R 4.3.0)
## yaml 2.3.7 2023-01-23 [1] CRAN (R 4.3.0)
##
## [1] C:/Users/tom/AppData/Local/R/win-library/4.3
## [2] C:/Program Files/R/R-4.3.0/library
##
## -----
```


Stata code

11. Why model: Stata

```
library(Statamarkdown)
```

```
do dependency
```

```
checking extremes consistency and verifying not already installed...
all files already exist and are up to date.
```

```
checking tomata consistency and verifying not already installed...
all files already exist and are up to date.
```

```
end of do-file
```

```
/*****
Stata code for Causal Inference: What If by Miguel Hernan & Jamie Robins
Date: 10/10/2019
Author: Eleanor Murray
For errors contact: ejmurray@bu.edu
*****/
```

Program 11.1

- Figures 11.1, 11.2, and 11.3
- Sample averages by treatment level

```
clear
```

```
**Figure 11.1**
```

```
*create the dataset*
```

```
input A Y
```

```
1 200
```

```
1 150
```

```
1 220
```

```
1 110
```

```
1 50
```

```
1 180
```

```
1 90
```

```
1 170
```

```
0 170
```

```
0 30
```

```
0 70
```

```
0 110
```

```
0 80
```

```
0 50
```

```

0 10
0 20
end

*Save the data*
qui save ./data/fig1, replace

*Build the scatterplot*
scatter Y A, ylab(0(50)250) xlab(0 1) xscale(range(-0.5 1.5))
qui gr export figs/stata-fig-11-1.png, replace

*Output the mean values for Y in each level of A*
bysort A: sum Y

```

```

      A      Y
1. 1 200
2. 1 150
3. 1 220
4. 1 110
5. 1 50
6. 1 180
7. 1 90
8. 1 170
9. 0 170
10. 0 30
11. 0 70
12. 0 110
13. 0 80
14. 0 50
15. 0 10
16. 0 20
17. end

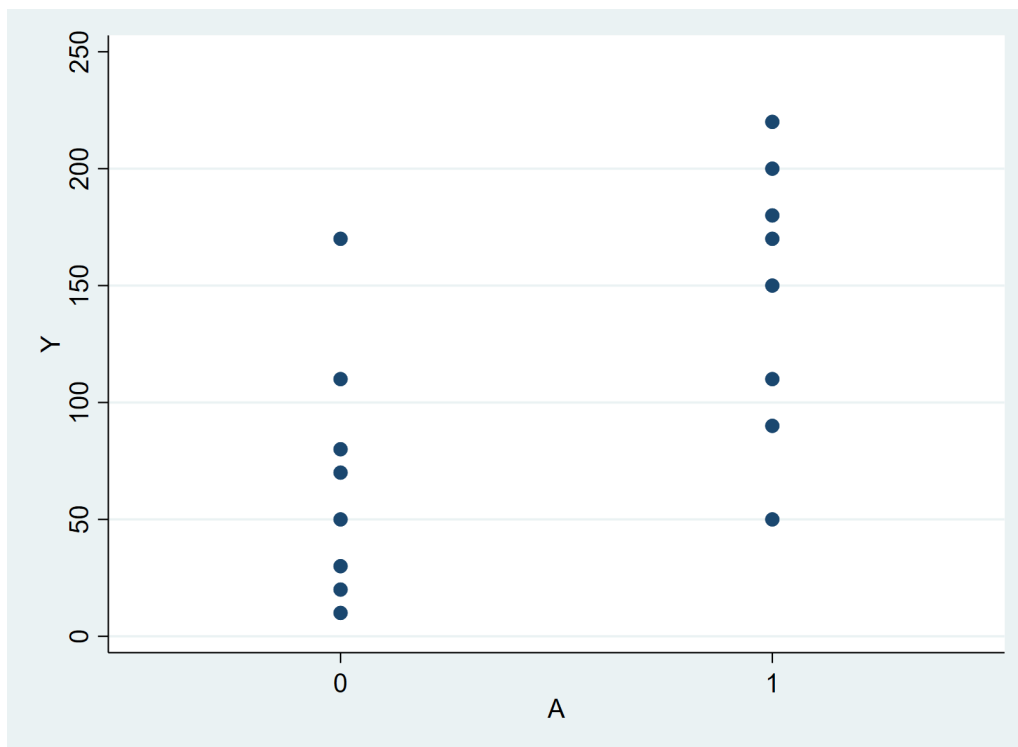
```

-> A = 0

Variable	Obs	Mean	Std. dev.	Min	Max
Y	8	67.5	53.11712	10	170

-> A = 1

Variable	Obs	Mean	Std. dev.	Min	Max
Y	8	146.25	58.2942	50	220



```

*Clear the workspace to be able to use a new dataset*
clear

**Figure 11.2**
input A Y
1 110
1 80
1 50
1 40
2 170
2 30
2 70
2 50
3 110
3 50
3 180
3 130
4 200
4 150
4 220
4 210
end

qui save ./data/fig2, replace

scatter Y A, ylab(0(50)250) xlab(0(1)4) xscale(range(0 4.5))
qui gr export figs/stata-fig-11-2.png, replace

bysort A: sum Y

```

A

Y

```

1. 1 110
2. 1 80
3. 1 50
4. 1 40
5. 2 170
6. 2 30
7. 2 70
8. 2 50
9. 3 110
10. 3 50
11. 3 180
12. 3 130
13. 4 200
14. 4 150
15. 4 220
16. 4 210
17. end

```

```
-----
-> A = 1
```

Variable	Obs	Mean	Std. dev.	Min	Max
Y	4	70	31.62278	40	110

```
-----
-> A = 2
```

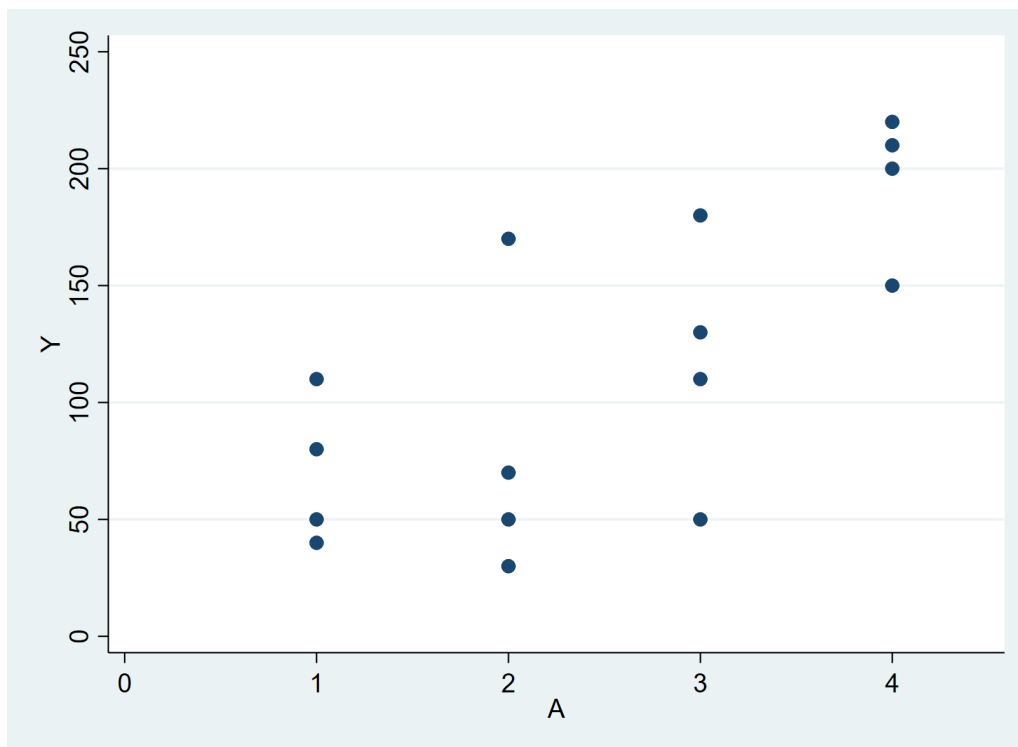
Variable	Obs	Mean	Std. dev.	Min	Max
Y	4	80	62.18253	30	170

```
-----
-> A = 3
```

Variable	Obs	Mean	Std. dev.	Min	Max
Y	4	117.5	53.77422	50	180

```
-----
-> A = 4
```

Variable	Obs	Mean	Std. dev.	Min	Max
Y	4	195	31.09126	150	220



```
clear

**Figure 11.3**
input A Y
3 21
11 54
17 33
23 101
29 85
37 65
41 157
53 120
67 111
79 200
83 140
97 220
60 230
71 217
15 11
45 190
end

qui save ./data/fig3, replace

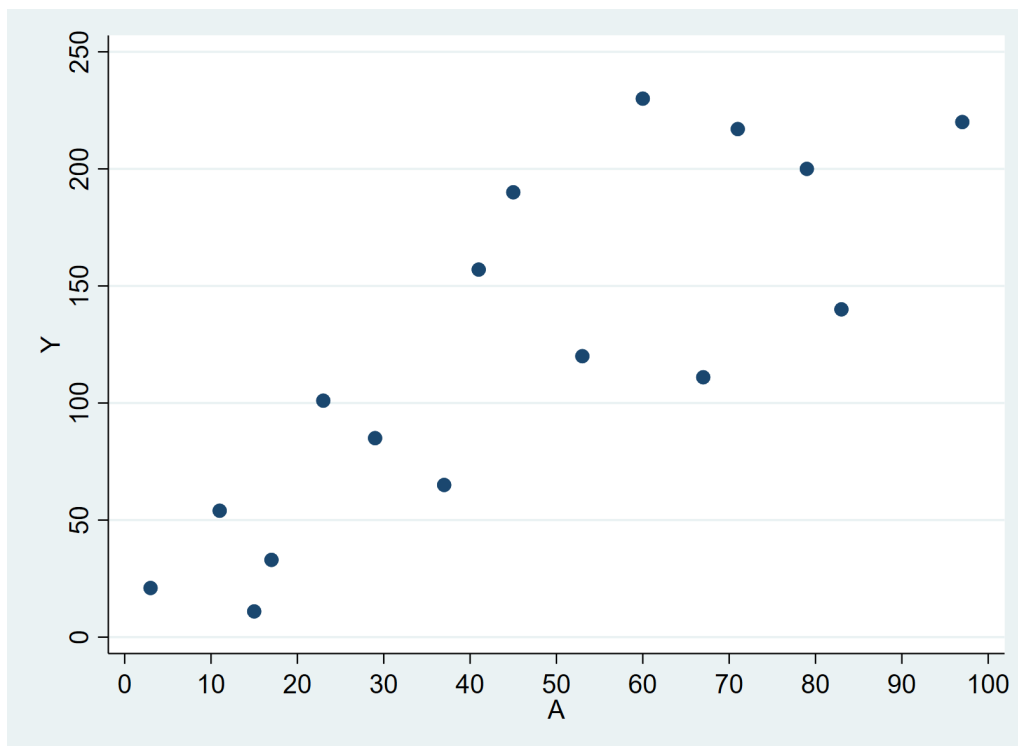
scatter Y A, ylab(0(50)250) xlab(0(10)100) xscale(range(0 100))
qui gr export figs/stata-fig-11-3.png, replace
```

	A	Y
1.	3	21
2.	11	54
3.	17	33

```

4. 23      101
5. 29      85
6. 37      65
7. 41      157
8. 53      120
9. 67      111
10. 79     200
11. 83     140
12. 97     220
13. 60     230
14. 71     217
15. 15      11
16. 45     190
17. end

```



Program 11.2

- 2-parameter linear model
- Creates Figure 11.4, parameter estimates with 95% confidence intervals from Section 11.2, and parameter estimates with 95% confidence intervals from Section 11.3

```

**Section 11.2: parametric estimators**
*Reload data
use ./data/fig3, clear

*Plot the data*
scatter Y A, ylab(0(50)250) xlab(0(10)100) xscale(range(0 100))

*Fit the regression model*
regress Y A, noheader cformat(%5.2f)

```

```
*Output the estimated mean Y value when A = 90*
```

```
lincom _b[_cons] + 90*_b[A]
```

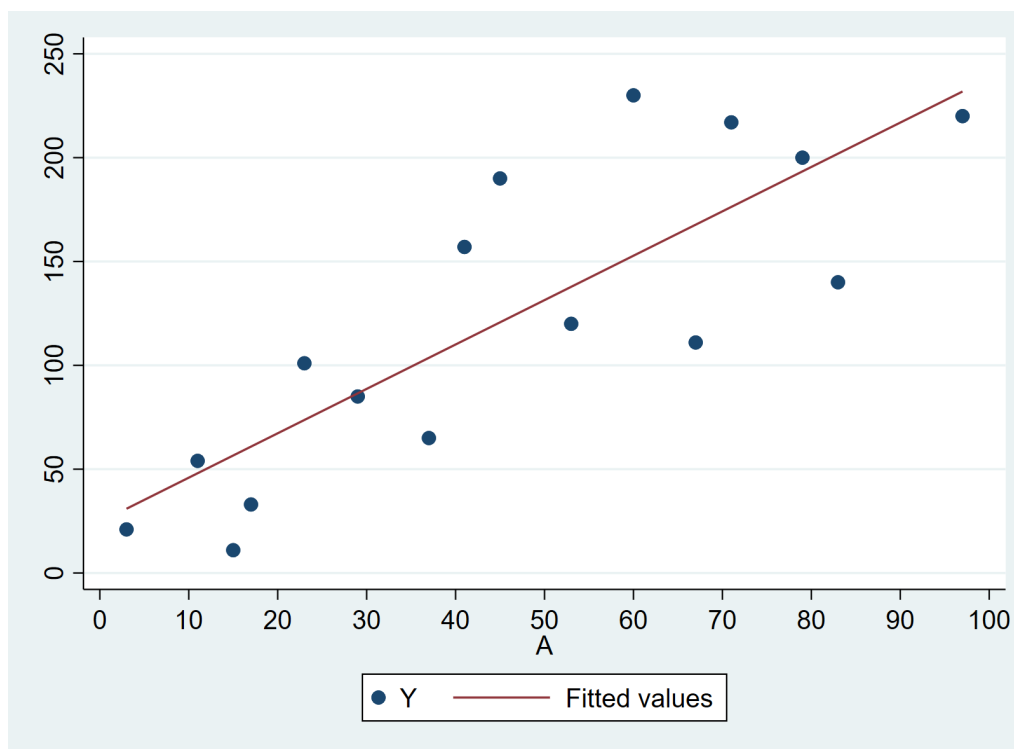
```
*Plot the data with the regression line: Fig 11.4*
```

```
scatter Y A, ylab(0(50)250) xlabel(0(10)100) xscale(range(0 100)) || lfit Y A
qui gr export figs/stata-fig-11-4.png, replace
```

Y	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
A	2.14	0.40	5.35	0.000	1.28	2.99
_cons	24.55	21.33	1.15	0.269	-21.20	70.29

```
( 1) 90*A + _cons = 0
```

Y	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
(1)	216.89	20.8614	10.40	0.000	172.1468	261.6333



```
**Section 11.3: non-parametric estimation*
```

```
* Reload the data
```

```
use ./data/fig1, clear
```

```
*Fit the regression model*
```

```
regress Y A, noheader cformat(%5.2f)
```

```
*E[Y|A=1]*
```

```
di 67.50 + 78.75
```

	Y	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
	A	78.75	27.88	2.82	0.014	18.95	138.55
	_cons	67.50	19.72	3.42	0.004	25.21	109.79

146.25

Program 11.3

- 3-parameter linear model
- Creates Figure 11.5 and Parameter estimates for Section 11.4

```
* Reload the data
use ./data/fig3, clear

*Create the product term*
gen Asq = A*A

*Fit the regression model*
regress Y A Asq, noheader cformat(%5.2f)

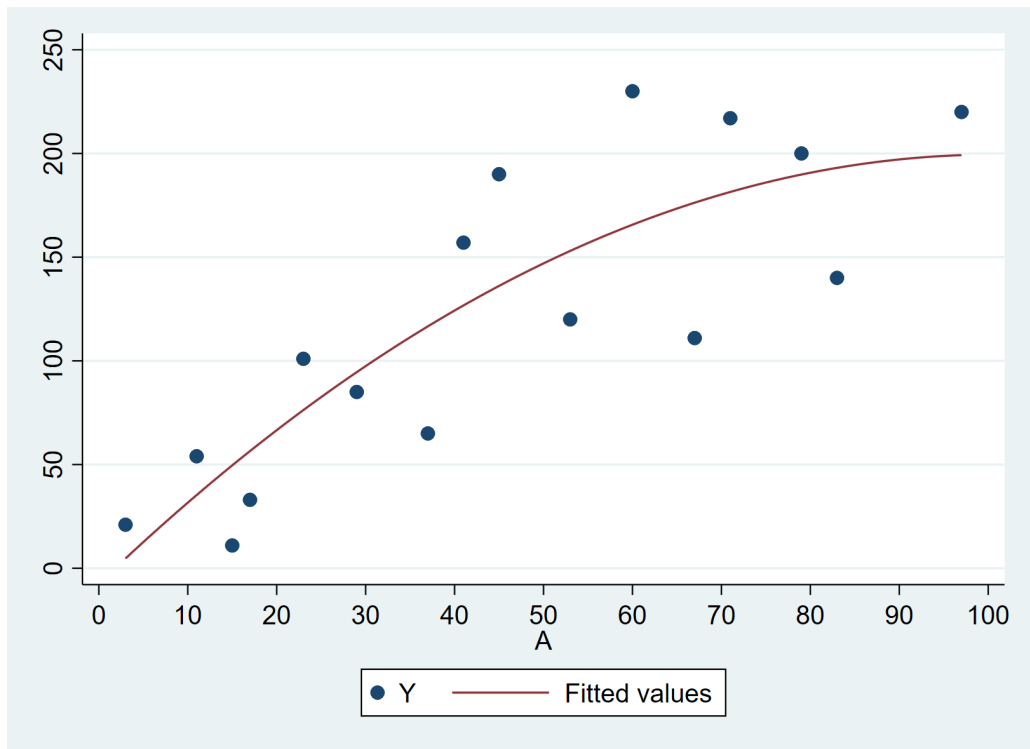
*Output the estimated mean Y value when A = 90*
lincom _b[_cons] + 90*_b[A] + 90*90*_b[Asq]

*Plot the data with the regression line: Fig 11.5*
scatter Y A, ylab(0(50)250) xlab(0(10)100) xscale(range(0 100)) || qfit Y A
qui gr export figs/stata-fig-11-5.png, replace
```

	Y	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
	A	4.11	1.53	2.68	0.019	0.80	7.41
	Asq	-0.02	0.02	-1.33	0.206	-0.05	0.01
	_cons	-7.41	31.75	-0.23	0.819	-75.99	61.18

(1) 90*A + 8100*Asq + _cons = 0

	Y	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
	(1)	197.1269	25.16157	7.83	0.000	142.7687	251.4852



12. IP Weighting and Marginal Structural Models: Stata

```
library(Statamarkdown)
```

```
/******  
Stata code for Causal Inference: What If by Miguel Hernan & Jamie Robins  
Date: 10/10/2019  
Author: Eleanor Murray  
For errors contact: ejmurray@bu.edu  
******/
```

Program 12.1

- Descriptive statistics from NHEFS data (Table 12.1)

```
use ./data/nhefs, clear  
  
/*Provisionally ignore subjects with missing values for follow-up weight*/  
/*Sample size after exclusion: N = 1566*/  
drop if wt82==.  
  
/* Calculate mean weight change in those with and without smoking cessation*/  
label define qsmk 0 "No smoking cessation" 1 "Smoking cessation"  
label values qsmk qsmk  
by qsmk, sort: egen years = mean(age) if age < .  
label var years "Age, years"  
by qsmk, sort: egen male = mean(100 * (sex==0)) if sex < .  
label var male "Men, %"  
by qsmk, sort: egen white = mean(100 * (race==0)) if race < .  
label var white "White, %"  
by qsmk, sort: egen university = mean(100 * (education == 5)) if education < .  
label var university "University, %"  
by qsmk, sort: egen kg = mean(wt71) if wt71 < .  
label var kg "Weight, kg"  
by qsmk, sort: egen cigs = mean(smokeintensity) if smokeintensity < .  
label var cigs "Cigarettes/day"  
by qsmk, sort: egen meansmkyrs = mean(smokeyrs) if smokeyrs < .  
label var kg "Years smoking"  
by qsmk, sort: egen noexer = mean(100 * (exercise == 2)) if exercise < .  
label var noexer "Little/no exercise"  
by qsmk, sort: egen inactive = mean(100 * (active==2)) if active < .
```

```
label var inactive "Inactive daily life"
qui save ./data/nhefs-formatted, replace
```

(63 observations deleted)

```
use ./data/nhefs-formatted, clear
```

```
/*Output table*/
```

```
foreach var of varlist years male white university kg cigs meansmkyrs noexer inactive {
    tabdisp qsmk, cell(`var') format(%3.1f)
}
```

```
2. tabdisp qsmk, cell(`var') format(%3.1f)
3. }
```

```
-----
quit smoking between |
baseline and 1982    | Age, years
-----+-----
No smoking cessation |      42.8
Smoking cessation   |      46.2
-----
```

```
-----
quit smoking between |
baseline and 1982    |      Men, %
-----+-----
No smoking cessation |      46.6
Smoking cessation   |      54.6
-----
```

```
-----
quit smoking between |
baseline and 1982    |      White, %
-----+-----
No smoking cessation |      85.4
Smoking cessation   |      91.1
-----
```

```
-----
quit smoking between |
baseline and 1982    | University, %
-----+-----
No smoking cessation |      9.9
Smoking cessation   |     15.4
-----
```

```
-----
quit smoking between |
baseline and 1982    | Years smoking
-----+-----
No smoking cessation |      70.3
Smoking cessation   |      72.4
-----
```

```

-----
quit smoking between |
baseline and 1982    | Cigarettes/day
-----+-----
No smoking cessation |          21.2
  Smoking cessation  |          18.6
-----

```

```

-----
quit smoking between |
baseline and 1982    | meansmkyrs
-----+-----
No smoking cessation |          24.1
  Smoking cessation  |          26.0
-----

```

```

-----
quit smoking between |
baseline and 1982    | Little/no exercise
-----+-----
No smoking cessation |          37.9
  Smoking cessation  |          40.7
-----

```

```

-----
quit smoking between |
baseline and 1982    | Inactive daily life
-----+-----
No smoking cessation |           8.9
  Smoking cessation  |          11.2
-----

```

Program 12.2

- Estimating IP weights for Section 12.2
- Data from NHEFS

```

use ./data/nhefs-formatted, clear

/*Fit a logistic model for the IP weights*/
logit qsmk sex race c.age##c.age ib(last).education c.smokeintensity##c.smokeintensity ///
c.smokeyrs##c.smokeyrs ib(last).exercise ib(last).active c.wt71##c.wt71

/*Output predicted conditional probability of quitting smoking for each individual*/
predict p_qsmk, pr

/*Generate nonstabilized weights as P(A=1|covariates) if A = 1 and */
/* 1-P(A=1|covariates) if A = 0*/
gen w=.
replace w=1/p_qsmk if qsmk==1
replace w=1/(1-p_qsmk) if qsmk==0
/*Check the mean of the weights; we expect it to be close to 2.0*/

```

```
summarize w
```

```
/*Fit marginal structural model in the pseudopopulation*/
/*Weights assigned using pweight = w*/
/*Robust standard errors using cluster() option where 'seqn' is the ID variable*/
regress wt82_71 qsmk [pweight=w], cluster(seqn)
```

```
Iteration 0: log likelihood = -893.02712
Iteration 1: log likelihood = -839.70016
Iteration 2: log likelihood = -838.45045
Iteration 3: log likelihood = -838.44842
Iteration 4: log likelihood = -838.44842
```

Logistic regression

Number of obs = 1,566
LR chi2(18) = 109.16
Prob > chi2 = 0.0000
Pseudo R2 = 0.0611

Log likelihood = -838.44842

	qsmk	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
sex		-.5274782	.1540497	-3.42	0.001	-.82941	-.2255463
race		-.8392636	.2100668	-4.00	0.000	-1.250987	-.4275404
age		.1212052	.0512663	2.36	0.018	.0207251	.2216853
c.age#c.age		-.0008246	.0005361	-1.54	0.124	-.0018753	.0002262
education							
1		-.4759606	.2262238	-2.10	0.035	-.9193511	-.0325701
2		-.5047361	.217597	-2.32	0.020	-.9312184	-.0782538
3		-.3895288	.1914353	-2.03	0.042	-.7647351	-.0143226
4		-.4123596	.2772868	-1.49	0.137	-.9558318	.1311126
smokeintensity		-.0772704	.0152499	-5.07	0.000	-.1071596	-.0473812
c.smokeintensity#							
c.smokeintensity		.0010451	.0002866	3.65	0.000	.0004835	.0016068
smokeyrs		-.0735966	.0277775	-2.65	0.008	-.1280395	-.0191538
c.smokeyrs#							
c.smokeyrs		.0008441	.0004632	1.82	0.068	-.0000637	.0017519
exercise							
0		-.395704	.1872401	-2.11	0.035	-.7626878	-.0287201
1		-.0408635	.1382674	-0.30	0.768	-.3118627	.2301357
active							
0		-.176784	.2149721	-0.82	0.411	-.5981215	.2445535
1		-.1448395	.2111472	-0.69	0.493	-.5586806	.2690015
wt71		-.0152357	.0263161	-0.58	0.563	-.0668144	.036343

c.wt71#c.wt71		.0001352	.0001632	0.83	0.407	-.0001846	.000455
_cons		-1.19407	1.398493	-0.85	0.393	-3.935066	1.546925

(1,566 missing values generated)

(403 real changes made)

(1,163 real changes made)

Variable		Obs	Mean	Std. dev.	Min	Max
w		1,566	1.996284	1.474787	1.053742	16.70009

(sum of wgt is 3,126.18084549904)

Linear regression		Number of obs	=	1,566
		F(1, 1565)	=	42.81
		Prob > F	=	0.0000
		R-squared	=	0.0435
		Root MSE	=	8.0713

(Std. err. adjusted for 1,566 clusters in seqn)

			Robust			
wt82_71		Coefficient	std. err.	t	P> t	[95% conf. interval]
qsmk		3.440535	.5258294	6.54	0.000	2.409131 4.47194
_cons		1.779978	.2248742	7.92	0.000	1.338892 2.221065

Program 12.3

- Estimating stabilized IP weights for Section 12.3
- Data from NHEFS

```
use ./data/nhefs-formatted, clear

/*Fit a logistic model for the denominator of the IP weights and predict the */
/* conditional probability of smoking */
logit qsmk sex race c.age##c.age ib(last).education c.smokeintensity##c.smokeintensity ///
c.smokeyrs##c.smokeyrs ib(last).exercise ib(last).active c.wt71##c.wt71
predict pd_qsmk, pr

/*Fit a logistic model for the numerator of ip weights and predict Pr(A=1) */
logit qsmk
predict pn_qsmk, pr

/*Generate stabilized weights as f(A)/f(A|L)*/
gen sw_a=.
replace sw_a=pn_qsmk/pd_qsmk if qsmk==1
```

```

replace sw_a=(1-pn_qsmk)/(1-pd_qsmk) if qsmk==0

/*Check distribution of the stabilized weights*/
summarize sw_a

/*Fit marginal structural model in the pseudopopulation*/
regress wt82_71 qsmk [pweight=sw_a], cluster(seqn)

/*****
FINE POINT 12.2
Checking positivity
*****/

/*Check for missing values within strata of covariates, for example: */
tab age qsmk if race==0 & sex==1 & wt82!=.
tab age qsmk if race==1 & sex==1 & wt82!=.

```

```

Iteration 0:  log likelihood = -893.02712
Iteration 1:  log likelihood = -839.70016
Iteration 2:  log likelihood = -838.45045
Iteration 3:  log likelihood = -838.44842
Iteration 4:  log likelihood = -838.44842

```

Logistic regression

Number of obs = 1,566

LR chi2(18) = 109.16

Prob > chi2 = 0.0000

Pseudo R2 = 0.0611

Log likelihood = -838.44842

	qsmk	Coefficient	Std. err.	z	P> z	[95% conf. interval]

	sex	-.5274782	.1540497	-3.42	0.001	-.82941 -.2255463
	race	-.8392636	.2100668	-4.00	0.000	-1.250987 -.4275404
	age	.1212052	.0512663	2.36	0.018	.0207251 .2216853
	c.age#c.age	-.0008246	.0005361	-1.54	0.124	-.0018753 .0002262
	education					
	1	-.4759606	.2262238	-2.10	0.035	-.9193511 -.0325701
	2	-.5047361	.217597	-2.32	0.020	-.9312184 -.0782538
	3	-.3895288	.1914353	-2.03	0.042	-.7647351 -.0143226
	4	-.4123596	.2772868	-1.49	0.137	-.9558318 .1311126
	smokeintensity	-.0772704	.0152499	-5.07	0.000	-.1071596 -.0473812
	c.smokeintensity#					
	c.smokeintensity	.0010451	.0002866	3.65	0.000	.0004835 .0016068
	smokeyr	-.0735966	.0277775	-2.65	0.008	-.1280395 -.0191538
	c.smokeyr#					
	c.smokeyr	.0008441	.0004632	1.82	0.068	-.0000637 .0017519

exercise							
0		-.395704	.1872401	-2.11	0.035	-.7626878	-.0287201
1		-.0408635	.1382674	-0.30	0.768	-.3118627	.2301357
active							
0		-.176784	.2149721	-0.82	0.411	-.5981215	.2445535
1		-.1448395	.2111472	-0.69	0.493	-.5586806	.2690015
wt71		-.0152357	.0263161	-0.58	0.563	-.0668144	.036343
c.wt71#c.wt71		.0001352	.0001632	0.83	0.407	-.0001846	.000455
_cons		-1.19407	1.398493	-0.85	0.393	-3.935066	1.546925

Iteration 0: log likelihood = -893.02712

Iteration 1: log likelihood = -893.02712

Logistic regression

Number of obs = 1,566

LR chi2(0) = 0.00

Prob > chi2 = .

Pseudo R2 = 0.0000

Log likelihood = -893.02712

qsmk		Coefficient	Std. err.	z	P> z	[95% conf. interval]
-----+-----						
_cons		-1.059822	.0578034	-18.33	0.000	-1.173114 - .946529

(1,566 missing values generated)

(403 real changes made)

(1,163 real changes made)

Variable		Obs	Mean	Std. dev.	Min	Max
-----+-----						
sw_a		1,566	.9988444	.2882233	.3312489	4.297662

(sum of wgt is 1,564.19025221467)

Linear regression

Number of obs = 1,566

F(1, 1565) = 42.81

Prob > F = 0.0000

R-squared = 0.0359

Root MSE = 7.7972

(Std. err. adjusted for 1,566 clusters in seqn)

| Robust

wt82_71	Coefficient	std. err.	t	P> t	[95% conf. interval]	
qsmk	3.440535	.5258294	6.54	0.000	2.409131	4.47194
_cons	1.779978	.2248742	7.92	0.000	1.338892	2.221065

quit smoking between baseline and 1982			
age	No smokin	Smoking c	Total
25	24	3	27
26	14	5	19
27	18	2	20
28	20	5	25
29	15	4	19
30	14	5	19
31	11	5	16
32	14	7	21
33	12	3	15
34	22	5	27
35	16	5	21
36	13	3	16
37	14	1	15
38	6	2	8
39	19	4	23
40	10	4	14
41	13	3	16
42	16	3	19
43	14	3	17
44	9	4	13
45	12	5	17
46	19	4	23
47	19	4	23
48	19	4	23
49	11	3	14
50	18	4	22
51	9	3	12
52	11	3	14
53	11	4	15
54	17	9	26
55	9	4	13
56	8	7	15
57	9	2	11
58	8	4	12
59	5	4	9
60	5	4	9
61	5	2	7
62	6	5	11
63	3	3	6
64	7	1	8
65	3	2	5
66	4	0	4
67	2	0	2

69	6	2	8
70	2	1	3
71	0	1	1
72	2	2	4
74	0	1	1

Total	524	164	688

quit smoking between baseline and 1982			
age	No smokin	Smoking c	Total

25	3	1	4
26	3	0	3
28	3	1	4
29	1	0	1
30	4	0	4
31	3	0	3
32	8	0	8
33	2	0	2
34	2	1	3
35	3	0	3
36	5	0	5
37	3	1	4
38	4	2	6
39	1	1	2
40	2	2	4
41	3	0	3
42	3	0	3
43	4	2	6
44	3	0	3
45	1	3	4
46	5	0	5
47	3	0	3
48	4	0	4
49	1	1	2
50	2	0	2
51	4	0	4
52	1	0	1
53	2	0	2
54	2	0	2
55	3	0	3
56	2	1	3
57	2	1	3
61	1	1	2
67	1	0	1
68	1	0	1
69	2	0	2
70	0	1	1

Total	97	19	116

Program 12.4

- Estimating the parameters of a marginal structural mean model with a continuous treatment Data from NHEFS
- Section 12.4

```
use ./data/nhefs-formatted, clear

* drop sw_a

/*Analysis restricted to subjects reporting <=25 cig/day at baseline: N = 1162*/
keep if smokeintensity <=25

/*Fit a linear model for the denominator of the IP weights and calculate the */
/* mean expected smoking intensity*/
regress smkintensity82_71 sex race c.age##c.age ib(last).education ///
c.smokeintensity##c.smokeintensity c.smokeyrs##c.smokeyrs ///
ib(last).exercise ib(last).active c.wt71##c.wt71
quietly predict p_den

/*Generate the denisty of the denomiator expectation using the mean expected */
/* smoking intensity and the residuals, assuming a normal distribution*/
/*Note: The regress command in Stata saves the root mean squared error for the */
/* immediate regression as e(rmse), thus there is no need to calculate it again. */
gen dens_den = normalden(smkintensity82_71, p_den, e(rmse))

/*Fit a linear model for the numerator of ip weights, calculate the mean */
/* expected value, and generate the density*/
quietly regress smkintensity82_71
quietly predict p_num
gen dens_num = normalden( smkintensity82_71, p_num, e(rmse))

/*Generate the final stabilized weights from the estimated numerator and */
/* denominator, and check the weights distribution*/
gen sw_a=dens_num/dens_den
summarize sw_a

/*Fit a marginal structural model in the pseudopopulation*/
regress wt82_71 c.smkintensity82_71##c.smkintensity82_71 [pweight=sw_a], cluster(seqn)

/*Output the estimated mean Y value when smoke intensity is unchanged from */
/* baseline to 1982 */
lincom _b[_cons]

/*Output the estimated mean Y value when smoke intensity increases by 20 from */
/* baseline to 1982*/
lincom _b[_cons] + 20*_b[smkintensity82_71 ] + ///
400*_b[c.smkintensity82_71#c.smkintensity82_71]
```

(404 observations deleted)

Source	SS	df	MS	Number of obs	=	1,162
-----+-----				F(18, 1143)	=	5.39
Model	9956.95654	18	553.164252	Prob > F	=	0.0000

Residual		117260.18	1,143	102.589834	R-squared	=	0.0783
-----+					Adj R-squared	=	0.0638
Total		127217.137	1,161	109.575484	Root MSE	=	10.129

smkintensity82_71		Coefficient	Std. err.	t	P> t	[95% conf. interval]	
-----+							
sex		1.087021	.7425694	1.46	0.144	-.3699308 2.543973	
race		.2319789	.8434739	0.28	0.783	-1.422952 1.88691	
age		-.8099902	.2555388	-3.17	0.002	-1.311368 -.3086124	
c.age#c.age		.0066545	.0026849	2.48	0.013	.0013865 .0119224	
education							
1		1.508097	1.184063	1.27	0.203	-.8150843 3.831278	
2		2.02692	1.133772	1.79	0.074	-.1975876 4.251428	
3		2.240314	1.022556	2.19	0.029	.2340167 4.246611	
4		2.528767	1.44702	1.75	0.081	-.3103458 5.36788	
smokeintensity		-.3589684	.2246653	-1.60	0.110	-.799771 .0818342	
c.smokeintensity#							
c.smokeintensity		.0019582	.0085753	0.23	0.819	-.0148668 .0187832	
smokeyrs		.3857088	.1416765	2.72	0.007	.1077336 .6636841	
c.smokeyrs#							
c.smokeyrs		-.0054871	.0023837	-2.30	0.022	-.0101641 -.0008101	
exercise							
0		1.996904	.9080421	2.20	0.028	.215288 3.778521	
1		.988812	.6929239	1.43	0.154	-.3707334 2.348357	
active							
0		.8451341	1.098573	0.77	0.442	-1.310312 3.000581	
1		.800114	1.08438	0.74	0.461	-1.327485 2.927712	
wt71		-.0656882	.136955	-0.48	0.632	-.3343996 .2030232	
c.wt71#c.wt71		.0005711	.000877	0.65	0.515	-.0011496 .0022918	
_cons		16.86761	7.109189	2.37	0.018	2.91909 30.81614	

Variable		Obs	Mean	Std. dev.	Min	Max
-----+						
sw_a		1,162	.9968057	.3222937	.1938336	5.102339

(sum of wgt is 1,158.28818286955)

Linear regression	Number of obs	=	1,162
	F(2, 1161)	=	12.75
	Prob > F	=	0.0000

R-squared = 0.0233
Root MSE = 7.7864

(Std. err. adjusted for 1,162 clusters in seqn)

		Robust				
wt82_71	Coefficient	std. err.	t	P> t	[95% conf. interval]	
smkintensity82_71	-.1089889	.0315762	-3.45	0.001	-.1709417	-.0470361
c. smkintensity82_71#						
c.smkintensity82_71	.0026949	.0024203	1.11	0.266	-.0020537	.0074436
_cons	2.004525	.295502	6.78	0.000	1.424747	2.584302

(1) _cons = 0

wt82_71	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
(1)	2.004525	.295502	6.78	0.000	1.424747	2.584302

(1) 20*smkintensity82_71 + 400*c.smkintensity82_71#c.smkintensity82_71 + _cons = 0

wt82_71	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
(1)	.9027234	1.310533	0.69	0.491	-1.668554	3.474001

Program 12.5

- Estimating the parameters of a marginal structural logistic model
- Data from NHEFS
- Section 12.4

```
use ./data/nhefs, clear

/*Provisionally ignore subjects with missing values for follow-up weight*/
/*Sample size after exclusion: N = 1566*/
drop if wt82==.

/*Estimate the stabilized weights for quitting smoking as in PROGRAM 12.3*/
/*Fit a logistic model for the denominator of the IP weights and predict the */
/* conditional probability of smoking*/
logit qsmk sex race c.age##c.age ib(last).education c.smokeintensity##c.smokeintensity ///
c.smokeyrs##c.smokeyrs ib(last).exercise ib(last).active c.wt71##c.wt71
predict pd_qsmk, pr
```

```

/*Fit a logistic model for the numerator of ip weights and predict Pr(A=1) */
logit qsmk
predict pn_qsmk, pr
/*Generate stabilized weights as f(A)/f(A|L)*/
gen sw_a=.
replace sw_a=pn_qsmk/pd_qsmk if qsmk==1
replace sw_a=(1-pn_qsmk)/(1-pd_qsmk) if qsmk==0
summarize sw_a

/*Fit marginal structural model in the pseudopopulation*/
/*NOTE: Stata has two commands for logistic regression, logit and logistic*/
/*Using logistic allows us to output the odds ratios directly*/
/*We can also output odds ratios from the logit command using the or option */
/* (default logit output is regression coefficients*/
logistic death qsmk [pweight=sw_a], cluster(seqn)

```

(63 observations deleted)

```

Iteration 0:  log likelihood = -893.02712
Iteration 1:  log likelihood = -839.70016
Iteration 2:  log likelihood = -838.45045
Iteration 3:  log likelihood = -838.44842
Iteration 4:  log likelihood = -838.44842

```

Logistic regression

```

Number of obs = 1,566
LR chi2(18)    = 109.16
Prob > chi2    = 0.0000
Pseudo R2     = 0.0611

```

Log likelihood = -838.44842

	qsmk	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
sex		-.5274782	.1540497	-3.42	0.001	-.82941	-.2255463
race		-.8392636	.2100668	-4.00	0.000	-1.250987	-.4275404
age		.1212052	.0512663	2.36	0.018	.0207251	.2216853
c.age#c.age		-.0008246	.0005361	-1.54	0.124	-.0018753	.0002262
education							
1		-.4759606	.2262238	-2.10	0.035	-.9193511	-.0325701
2		-.5047361	.217597	-2.32	0.020	-.9312184	-.0782538
3		-.3895288	.1914353	-2.03	0.042	-.7647351	-.0143226
4		-.4123596	.2772868	-1.49	0.137	-.9558318	.1311126
smokeintensity		-.0772704	.0152499	-5.07	0.000	-.1071596	-.0473812
c.smokeintensity#							
c.smokeintensity		.0010451	.0002866	3.65	0.000	.0004835	.0016068
smokeyrs		-.0735966	.0277775	-2.65	0.008	-.1280395	-.0191538
c.smokeyrs#							

c.smokeyrs		.0008441	.0004632	1.82	0.068	-.0000637	.0017519
exercise							
0		-.395704	.1872401	-2.11	0.035	-.7626878	-.0287201
1		-.0408635	.1382674	-0.30	0.768	-.3118627	.2301357
active							
0		-.176784	.2149721	-0.82	0.411	-.5981215	.2445535
1		-.1448395	.2111472	-0.69	0.493	-.5586806	.2690015
wt71		-.0152357	.0263161	-0.58	0.563	-.0668144	.036343
c.wt71#c.wt71		.0001352	.0001632	0.83	0.407	-.0001846	.000455
_cons		-1.19407	1.398493	-0.85	0.393	-3.935066	1.546925

Iteration 0: log likelihood = -893.02712
Iteration 1: log likelihood = -893.02712

Logistic regression	Number of obs =	1,566
	LR chi2(0) =	-0.00
	Prob > chi2 =	.
Log likelihood = -893.02712	Pseudo R2 =	-0.0000

qsmk		Coefficient	Std. err.	z	P> z	[95% conf. interval]
_cons		-1.059822	.0578034	-18.33	0.000	-1.173114 - .946529

(1,566 missing values generated)

(403 real changes made)

(1,163 real changes made)

Variable		Obs	Mean	Std. dev.	Min	Max
sw_a		1,566	.9988444	.2882233	.3312489	4.297662

Logistic regression	Number of obs =	1,566
	Wald chi2(1) =	0.04
	Prob > chi2 =	0.8482
Log pseudolikelihood = -749.11596	Pseudo R2 =	0.0000

(Std. err. adjusted for 1,566 clusters in seqn)

		Robust
--	--	--------

	death	Odds ratio	std. err.	z	P> z	[95% conf. interval]	
	qsmk	1.030578	.1621842	0.19	0.848	.7570517	1.402931
	_cons	.2252711	.0177882	-18.88	0.000	.1929707	.2629781

Note: _cons estimates baseline odds.

Program 12.6

- Assessing effect modification by sex using a marginal structural mean model
- Data from NHEFS
- Section 12.5

```
use ./data/nhefs, clear

* drop pd_qsmk pn_qsmk sw_a

/*Check distribution of sex*/
tab sex

/*Fit logistic model for the denominator of IP weights, as in PROGRAM 12.3 */
logit qsmk sex race c.age##c.age ib(last).education c.smokeintensity##c.smokeintensity ///
c.smokeyrs##c.smokeyrs ib(last).exercise ib(last).active c.wt71##c.wt71
predict pd_qsmk, pr

/*Fit logistic model for the numerator of IP weights, no including sex */
logit qsmk sex
predict pn_qsmk, pr

/*Generate IP weights as before*/
gen sw_a=.
replace sw_a=pn_qsmk/pd_qsmk if qsmk==1
replace sw_a=(1-pn_qsmk)/(1-pd_qsmk) if qsmk==0

summarize sw_a

/*Fit marginal structural model in the pseudopopulation, including interaction */
/* term between quitting smoking and sex*/
regress wt82_71 qsmk##sex [pw=sw_a], cluster(seqn)
```

sex	Freq.	Percent	Cum.
0	799	49.05	49.05
1	830	50.95	100.00
Total	1,629	100.00	

```
Iteration 0:  log likelihood = -938.14308
Iteration 1:  log likelihood = -884.53806
Iteration 2:  log likelihood = -883.35064
Iteration 3:  log likelihood = -883.34876
Iteration 4:  log likelihood = -883.34876
```

Logistic regression

Number of obs = 1,629

LR chi2(18) = 109.59

Prob > chi2 = 0.0000

Pseudo R2 = 0.0584

Log likelihood = -883.34876

qsmk	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
sex	-.5075218	.1482316	-3.42	0.001	-.7980505	-.2169932
race	-.8502312	.2058722	-4.13	0.000	-1.253733	-.4467292
age	.1030132	.0488996	2.11	0.035	.0071718	.1988547
c.age#c.age	-.0006052	.0005074	-1.19	0.233	-.0015998	.0003893
education						
1	-.3796632	.2203948	-1.72	0.085	-.811629	.0523026
2	-.4779835	.2141771	-2.23	0.026	-.8977629	-.0582041
3	-.3639645	.1885776	-1.93	0.054	-.7335698	.0056409
4	-.4221892	.2717235	-1.55	0.120	-.9547574	.110379
smokeintensity	-.0651561	.0147589	-4.41	0.000	-.0940831	-.0362292
c.smokeintensity#						
c.smokeintensity	.0008461	.0002758	3.07	0.002	.0003054	.0013867
smokeyrs	-.0733708	.0269958	-2.72	0.007	-.1262816	-.02046
c.smokeyrs#						
c.smokeyrs	.0008384	.0004435	1.89	0.059	-.0000307	.0017076
exercise						
0	-.3550517	.1799293	-1.97	0.048	-.7077067	-.0023967
1	-.06364	.1351256	-0.47	0.638	-.3284812	.2012013
active						
0	-.0683123	.2087269	-0.33	0.743	-.4774095	.3407849
1	-.057437	.2039967	-0.28	0.778	-.4572632	.3423892
wt71	-.0128478	.0222829	-0.58	0.564	-.0565214	.0308258
c.wt71#c.wt71	.0001209	.0001352	0.89	0.371	-.000144	.0003859
_cons	-1.185875	1.263142	-0.94	0.348	-3.661588	1.289838

Iteration 0: log likelihood = -938.14308

Iteration 1: log likelihood = -933.49896

Iteration 2: log likelihood = -933.49126

Iteration 3: log likelihood = -933.49126


```

Logistic regression
Log likelihood = -933.49126
Number of obs = 1,629
LR chi2(1) = 9.30
Prob > chi2 = 0.0023
Pseudo R2 = 0.0050

```

	qsmk	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
sex		-.3441893	.1131341	-3.04	0.002	-.565928	-.1224506
_cons		-.8634417	.0774517	-11.15	0.000	-1.015244	-.7116391

(1,629 missing values generated)

(428 real changes made)

(1,201 real changes made)

Variable		Obs	Mean	Std. dev.	Min	Max
sw_a		1,629	.9991318	.2636164	.2901148	3.683352

(sum of wgt is 1,562.01032829285)

```

Linear regression
Number of obs = 1,566
F(3, 1565) = 16.31
Prob > F = 0.0000
R-squared = 0.0379
Root MSE = 7.8024

```

(Std. err. adjusted for 1,566 clusters in seqn)

		Robust				
wt82_71		Coefficient	std. err.	t	P> t	[95% conf. interval]
1.qsmk		3.60623	.6576053	5.48	0.000	2.31635 4.89611
1.sex		-.0040025	.4496206	-0.01	0.993	-.8859246 .8779197
qsmk#sex						
1 1		-.161224	1.036143	-0.16	0.876	-2.1936 1.871152
_cons		1.759045	.3102511	5.67	0.000	1.150494 2.367597

Program 12.7

- Estimating IP weights to adjust for selection bias due to censoring
- Data from NHEFS
- Section 12.6

```
use ./data/nhefs, clear
```

```

/*Analysis including all individuals regardless of missing wt82 status: N=1629*/
/*Generate censoring indicator: C = 1 if wt82 missing*/
gen byte cens = (wt82 == .)

/*Check distribution of censoring by quitting smoking and baseline weight*/
tab cens qsmk, column
bys cens: summarize wt71

/*Fit logistic regression model for the denominator of IP weight for A*/
logit qsmk sex race c.age##c.age ib(last).education c.smokeintensity##c.smokeintensity ///
c.smokeyrs##c.smokeyrs ib(last).exercise ib(last).active c.wt71##c.wt71
predict pd_qsmk, pr

/*Fit logistic regression model for the numerator of IP weights for A*/
logit qsmk
predict pn_qsmk, pr

/*Fit logistic regression model for the denominator of IP weights for C, */
/* including quitting smoking*/
logit cens qsmk sex race c.age##c.age ib(last).education ///
c.smokeintensity##c.smokeintensity c.smokeyrs##c.smokeyrs ib(last).exercise ///
ib(last).active c.wt71##c.wt71
predict pd_cens, pr

/*Fit logistic regression model for the numerator of IP weights for C, */
/* including quitting smoking */
logit cens qsmk
predict pn_cens, pr

/*Generate the stabilized weights for A (sw_a)*/
gen sw_a=.
replace sw_a=pn_qsmk/pd_qsmk if qsmk==1
replace sw_a=(1-pn_qsmk)/(1-pd_qsmk) if qsmk==0

/*Generate the stabilized weights for C (sw_c)*/
/*NOTE: the conditional probability estimates generated by our logistic models */
/* for C represent the conditional probability of being censored (C=1)*/
/*We want weights for the conditional probability of being uncensored, Pr(C=0|A,L)*/
gen sw_c=.
replace sw_c=(1-pn_cens)/(1-pd_cens) if cens==0

/*Generate the final stabilized weights and check distribution*/
gen sw=sw_a*sw_c
summarize sw

/*Fit marginal structural model in the pseudopopulation*/
regress wt82_71 qsmk [pw=sw], cluster(seqn)

```

```

| Key |
|-----|
| frequency |
| column percentage |
+-----+

```

quit smoking between baseline and 1982			
cens	0	1	Total
0	1,163	403	1,566
	96.84	94.16	96.13
1	38	25	63
	3.16	5.84	3.87
Total	1,201	428	1,629
	100.00	100.00	100.00

-> cens = 0

Variable	Obs	Mean	Std. dev.	Min	Max
wt71	1,566	70.83092	15.3149	39.58	151.73

-> cens = 1

Variable	Obs	Mean	Std. dev.	Min	Max
wt71	63	76.55079	23.3326	36.17	169.19

Iteration 0: log likelihood = -938.14308
 Iteration 1: log likelihood = -884.53806
 Iteration 2: log likelihood = -883.35064
 Iteration 3: log likelihood = -883.34876
 Iteration 4: log likelihood = -883.34876

Logistic regression

Number of obs = 1,629
 LR chi2(18) = 109.59
 Prob > chi2 = 0.0000
 Pseudo R2 = 0.0584

Log likelihood = -883.34876

qsmk	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
sex	-.5075218	.1482316	-3.42	0.001	-.7980505	-.2169932
race	-.8502312	.2058722	-4.13	0.000	-1.253733	-.4467292
age	.1030132	.0488996	2.11	0.035	.0071718	.1988547
c.age#c.age	-.0006052	.0005074	-1.19	0.233	-.0015998	.0003893
education						
1	-.3796632	.2203948	-1.72	0.085	-.811629	.0523026

2		-.4779835	.2141771	-2.23	0.026	-.8977629	-.0582041
3		-.3639645	.1885776	-1.93	0.054	-.7335698	.0056409
4		-.4221892	.2717235	-1.55	0.120	-.9547574	.110379
smokeintensity		-.0651561	.0147589	-4.41	0.000	-.0940831	-.0362292
c.smokeintensity#							
c.smokeintensity		.0008461	.0002758	3.07	0.002	.0003054	.0013867
smokeyrs		-.0733708	.0269958	-2.72	0.007	-.1262816	-.02046
c.smokeyrs#							
c.smokeyrs		.0008384	.0004435	1.89	0.059	-.0000307	.0017076
exercise							
0		-.3550517	.1799293	-1.97	0.048	-.7077067	-.0023967
1		-.06364	.1351256	-0.47	0.638	-.3284812	.2012013
active							
0		-.0683123	.2087269	-0.33	0.743	-.4774095	.3407849
1		-.057437	.2039967	-0.28	0.778	-.4572632	.3423892
wt71		-.0128478	.0222829	-0.58	0.564	-.0565214	.0308258
c.wt71#c.wt71		.0001209	.0001352	0.89	0.371	-.000144	.0003859
_cons		-1.185875	1.263142	-0.94	0.348	-3.661588	1.289838

Iteration 0: log likelihood = -938.14308

Iteration 1: log likelihood = -938.14308

Logistic regression

Number of obs = 1,629

LR chi2(0) = 0.00

Prob > chi2 = .

Pseudo R2 = 0.0000

Log likelihood = -938.14308

	qsmk	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
_cons		-1.031787	.0562947	-18.33	0.000	-1.142122	-.9214511

Iteration 0: log likelihood = -266.67873

Iteration 1: log likelihood = -238.48654

Iteration 2: log likelihood = -232.82848

Iteration 3: log likelihood = -232.68043

Iteration 4: log likelihood = -232.67999

Iteration 5: log likelihood = -232.67999

Logistic regression

Number of obs = 1,629

LR chi2(19) = 68.00

Prob > chi2 = 0.0000

Pseudo R2 = 0.1275

Log likelihood = -232.67999

	cens	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
qsmk		.5168674	.2877162	1.80	0.072	-.0470459	1.080781
sex		.0573131	.3302775	0.17	0.862	-.590019	.7046452
race		-.0122715	.4524888	-0.03	0.978	-.8991332	.8745902
age		-.2697293	.1174647	-2.30	0.022	-.4999559	-.0395027
c.age#c.age		.0028837	.0011135	2.59	0.010	.0007012	.0050661
education							
1		.3823818	.5601808	0.68	0.495	-.7155523	1.480316
2		-.0584066	.5749586	-0.10	0.919	-1.185305	1.068491
3		.2176937	.5225008	0.42	0.677	-.8063891	1.241776
4		.5208288	.6678735	0.78	0.435	-.7881792	1.829837
smokeintensity		.0157119	.0347319	0.45	0.651	-.0523614	.0837851
c.smokeintensity#							
c.smokeintensity		-.0001133	.0006058	-0.19	0.852	-.0013007	.0010742
smokekeys		.0785973	.0749576	1.05	0.294	-.0683169	.2255116
c.smokekeys#							
c.smokekeys		-.0005569	.0010318	-0.54	0.589	-.0025791	.0014653
exercise							
0		.583989	.3723133	1.57	0.117	-.1457317	1.31371
1		-.3874824	.3439133	-1.13	0.260	-1.06154	.2865754
active							
0		-.7065829	.3964577	-1.78	0.075	-1.483626	.0704599
1		-.9540614	.3893181	-2.45	0.014	-1.717111	-.1910119
wt71		-.0878871	.0400115	-2.20	0.028	-.1663082	-.0094659
c.wt71#c.wt71		.0006351	.0002257	2.81	0.005	.0001927	.0010775
_cons		3.754678	2.651222	1.42	0.157	-1.441622	8.950978

Iteration 0: log likelihood = -266.67873

Iteration 1: log likelihood = -264.00252

Iteration 2: log likelihood = -263.88028

Iteration 3: log likelihood = -263.88009

Iteration 4: log likelihood = -263.88009

Logistic regression

Number of obs = 1,629

LR chi2(1) = 5.60

Prob > chi2 = 0.0180

Log likelihood = -263.88009

Pseudo R2 = 0.0105

	cens	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
	qsmk	.6411113	.2639262	2.43	0.015	.1238255	1.158397
	_cons	-3.421172	.1648503	-20.75	0.000	-3.744273	-3.098071

(1,629 missing values generated)

(428 real changes made)

(1,201 real changes made)

(1,629 missing values generated)

(1,566 real changes made)

(63 missing values generated)

Variable	Obs	Mean	Std. dev.	Min	Max
sw	1,566	.9962351	.2819583	.3546469	4.093113

(sum of wgt is 1,560.10419079661)

Linear regression

Number of obs = 1,566

F(1, 1565) = 44.19

Prob > F = 0.0000

R-squared = 0.0363

Root MSE = 7.8652

(Std. err. adjusted for 1,566 clusters in seqn)

			Robust			
	wt82_71	Coefficient	std. err.	t	P> t	[95% conf. interval]
	qsmk	3.496493	.5259796	6.65	0.000	2.464794 4.528192
	_cons	1.66199	.2328986	7.14	0.000	1.205164 2.118816

13. Standardization and the parametric G-formula: Stata

```
library(Statamarkdown)
```

```
/******  
Stata code for Causal Inference: What If by Miguel Hernan & Jamie Robins  
Date: 10/10/2019  
Author: Eleanor Murray  
For errors contact: ejmurray@bu.edu  
******/
```

Program 13.1

- Estimating the mean outcome within levels of treatment and confounders: Data from NHEFS
- Section 13.2

```
use ./data/nhefs-formatted, clear  
  
/* Estimate the conditional mean outcome within strata of quitting  
smoking and covariates, among the uncensored */  
glm wt82_71 qsmk sex race c.age##c.age ib(last).education ///  
    c.smokeintensity##c.smokeintensity c.smokeyrs##c.smokeyrs ///  
    ib(last).exercise ib(last).active c.wt71##c.wt71 ///  
    qsmk##c.smokeintensity  
predict meanY  
summarize meanY  
  
/*Look at the predicted value for subject ID = 24770*/  
list meanY if seqn == 24770  
  
/*Observed mean outcome for comparison */  
summarize wt82_71
```

note: 1.qsmk omitted because of collinearity.

note: smokeintensity omitted because of collinearity.

Iteration 0: log likelihood = -5328.5765

Generalized linear models	Number of obs	=	1,566
Optimization : ML	Residual df	=	1,545
	Scale parameter	=	53.5683

Deviance = 82763.02862 (1/df) Deviance = 53.5683
Pearson = 82763.02862 (1/df) Pearson = 53.5683

Variance function: $V(u) = 1$ [Gaussian]
Link function : $g(u) = u$ [Identity]

Log likelihood = -5328.576456 AIC = 6.832154
BIC = 71397.58

		OIM					
	wt82_71	Coefficient	std. err.	z	P> z	[95% conf. interval]	

	qsmk	2.559594	.8091486	3.16	0.002	.973692	4.145496
	sex	-1.430272	.4689576	-3.05	0.002	-2.349412	-.5111317
	race	.5601096	.5818888	0.96	0.336	-.5803714	1.700591
	age	.3596353	.1633188	2.20	0.028	.0395364	.6797342
	c.age#c.age	-.006101	.0017261	-3.53	0.000	-.0094841	-.0027178
	education						
	1	.194977	.7413692	0.26	0.793	-1.25808	1.648034
	2	.9854211	.7012116	1.41	0.160	-.3889285	2.359771
	3	.7512894	.6339153	1.19	0.236	-.4911617	1.993741
	4	1.686547	.8716593	1.93	0.053	-.0218744	3.394967
	smokeintensity	.0491365	.0517254	0.95	0.342	-.0522435	.1505165
	c.smokeintensity#						
	c.smokeintensity	-.0009907	.000938	-1.06	0.291	-.0028292	.0008479
	smokeyrs	.1343686	.0917122	1.47	0.143	-.045384	.3141212
	c.smokeyrs#						
	c.smokeyrs	-.0018664	.0015437	-1.21	0.227	-.0048921	.0011592
	exercise						
	0	-.3539128	.5588587	-0.63	0.527	-1.449256	.7414301
	1	-.0579374	.4316468	-0.13	0.893	-.9039497	.7880749
	active						
	0	.2613779	.6845577	0.38	0.703	-1.08033	1.603086
	1	-.6861916	.6739131	-1.02	0.309	-2.007037	.6346539
	wt71	.0455018	.0833709	0.55	0.585	-.1179022	.2089058
	c.wt71#c.wt71	-.0009653	.0005247	-1.84	0.066	-.0019937	.0000631
	qsmk						
Smoking cessation		0	(omitted)				
smokeintensity		0	(omitted)				
	qsmk#						


```

c.smokeintensity |
Smoking cessation | .0466628 .0351448 1.33 0.184 -.0222197 .1155453
                  |
                  _cons | -1.690608 4.388883 -0.39 0.700 -10.29266 6.911444
-----

```

(option mu assumed; predicted mean wt82_71)

```

Variable |      Obs      Mean   Std. dev.      Min      Max
-----+-----
meanY    |    1,566     2.6383   3.034683  -10.87582   9.876489

```

```

+-----+
| meanY |
|-----|
960. | .3421569 |
+-----+

```

```

Variable |      Obs      Mean   Std. dev.      Min      Max
-----+-----
wt82_71  |    1,566     2.6383   7.879913  -41.28047  48.53839

```

Program 13.2

- Standardizing the mean outcome to the baseline confounders
- Data from Table 2.2
- Section 13.3

```

clear
input str10 ID L A Y
"Rheia"    0 0 0
"Kronos"   0 0 1
"Demeter"  0 0 0
"Hades"    0 0 0
"Hestia"   0 1 0
"Poseidon" 0 1 0
"Hera"     0 1 0
"Zeus"     0 1 1
"Artemis"  1 0 1
"Apollo"   1 0 1
"Leto"     1 0 0
"Ares"     1 1 1
"Athena"   1 1 1
"Hephaestus" 1 1 1
"Aphrodite" 1 1 1
"Cyclope"  1 1 1
"Persephone" 1 1 1
"Hermes"   1 1 0
"Hebe"     1 1 0
"Dionysus" 1 1 0
end

```

```
/* i. Data set up for standardization:
```

```

- create 3 copies of each subject first,
- duplicate the dataset and create a variable `interv` which indicates
which copy is the duplicate (interv =1) */
expand 2, generate(interv)

/* Next, duplicate the original copy (interv = 0) again, and create
another variable 'interv2' to indicate the copy */
expand 2 if interv == 0, generate(interv2)

/* Now, change the value of 'interv' to -1 in one of the copies so that
there are unique values of interv for each copy */
replace interv = -1 if interv2 ==1
drop interv2

/* Check that the data has the structure you want:
- there should be 1566 people in each of the 3 levels of interv*/
tab interv

/* Two of the copies will be for computing the standardized result
for these two copies (interv = 0 and interv = 1), set the outcome to
missing and force qsmk to either 0 or 1, respectively.
You may need to edit this part of the code for your outcome and exposure variables */
replace Y = . if interv != -1
replace A = 0 if interv == 0
replace A = 1 if interv == 1

/* Check that the data has the structure you want:
for interv = -1, some people quit and some do not;
for interv = 0 or 1, noone quits or everyone quits, respectively */
by interv, sort: summarize A

*ii.Estimation in original sample*
*Now, we do a parametric regression with the covariates we want to adjust for*
*You may need to edit this part of the code for the variables you want.*
*Because the copies have missing Y, this will only run the regression in the
*original copy.*
*The double hash between A & L creates a regression model with A and L and a
* product term between A and L*
regress Y A##L

*Ask Stata for expected values - Stata will give you expected values for all
* copies, not just the original ones*
predict predY, xb

*Now ask for a summary of these values by intervention*
*These are the standardized outcome estimates: you can subtract them to get the
* standardized difference*
by interv, sort: summarize predY

*iii.OPTIONAL: Output standardized point estimates and difference*
*The summary from the last command gives you the standardized estimates*
*We can stop there, or we can ask Stata to calculate the standardized difference

```

```

* and display all the results in a simple table*
*The code below can be used as-is without changing any variable names*
*The option "quietly" asks Stata not to display the output of some intermediate
* calculations*
*You can delete this option if you want to see what is happening step-by-step*
quietly summarize predY if(interv == -1)
matrix input observe = (-1,`r(mean)')
quietly summarize predY if(interv == 0)
matrix observe = (observe \0,`r(mean)')
quietly summarize predY if(interv == 1)
matrix observe = (observe \1,`r(mean)')
matrix observe = (observe \., observe[3,2]-observe[2,2])

*Add some row/column descriptions and print results to screen*
matrix rownames observe = observed E(Y(a=0)) E(Y(a=1)) difference
matrix colnames observe = interv value
matrix list observe

*to interpret these results:*
*row 1, column 2, is the observed mean outcome value in our original sample*
*row 2, column 2, is the mean outcome value if everyone had not quit smoking*
*row 3, column 2, is the mean outcome value if everyone had quit smoking*
*row 4, column 2, is the mean difference outcome value if everyone had quit
* smoking compared to if everyone had not quit smoking*

```

	ID	L	A	Y
1.	"Rheia"	0	0	0
2.	"Kronos"	0	0	1
3.	"Demeter"	0	0	0
4.	"Hades"	0	0	0
5.	"Hestia"	0	1	0
6.	"Poseidon"	0	1	0
7.	"Hera"	0	1	0
8.	"Zeus"	0	1	1
9.	"Artemis"	1	0	1
10.	"Apollo"	1	0	1
11.	"Leto"	1	0	0
12.	"Ares"	1	1	1
13.	"Athena"	1	1	1
14.	"Hephaestus"	1	1	1
15.	"Aphrodite"	1	1	1
16.	"Cyclope"	1	1	1
17.	"Persephone"	1	1	1
18.	"Hermes"	1	1	0
19.	"Hebe"	1	1	0
20.	"Dionysus"	1	1	0
21.	end			

(20 observations created)

(20 observations created)

(20 real changes made)

Expanded observation type	Freq.	Percent	Cum.
-1	20	33.33	33.33
Original observation	20	33.33	66.67
Duplicated observation	20	33.33	100.00
Total	60	100.00	

(40 real changes made, 40 to missing)

(13 real changes made)

(7 real changes made)

-> interv = -1

Variable	Obs	Mean	Std. dev.	Min	Max
A	20	.65	.4893605	0	1

-> interv = Original

Variable	Obs	Mean	Std. dev.	Min	Max
A	20	0	0	0	0

-> interv = Duplicat

Variable	Obs	Mean	Std. dev.	Min	Max
A	20	1	0	1	1

Source	SS	df	MS	Number of obs	=	20
Model	.833333333	3	.277777778	F(3, 16)	=	1.07
Residual	4.16666667	16	.260416667	Prob > F	=	0.3909
Total	5	19	.263157895	R-squared	=	0.1667
				Adj R-squared	=	0.0104
				Root MSE	=	.51031

Y	Coefficient	Std. err.	t	P> t	[95% conf. interval]
1.A	1.05e-16	.3608439	0.00	1.000	-.7649549 .7649549
1.L	.4166667	.389756	1.07	0.301	-.4095791 1.242912

A#L							
1 1		-5.83e-17	.4959325	-0.00	1.000	-1.05133	1.05133
_cons		.25	.2551552	0.98	0.342	-.2909048	.7909048

```
-> interv = -1
```

Variable		Obs	Mean	Std. dev.	Min	Max
predY		20	.5	.209427	.25	.6666667

```
-> interv = Original
```

Variable		Obs	Mean	Std. dev.	Min	Max
predY		20	.5	.209427	.25	.6666667

```
-> interv = Duplicat
```

Variable		Obs	Mean	Std. dev.	Min	Max
predY		20	.5	.209427	.25	.6666667

```
observe[4,2]
```

	interv	value
observed	-1	.50000001
E(Y(a=0))	0	.50000001
E(Y(a=1))	1	.50000001
difference	.	0

Program 13.3

- Standardizing the mean outcome to the baseline confounders:
- Data from NHEFS
- Section 13.3

```

use ./data/nhefs-formatted, clear

*i.Data set up for standardization: create 3 copies of each subject*
*first, duplicate the dataset and create a variable 'interv' which indicates
* which copy is the duplicate (interv =1)
expand 2, generate(interv)

*next, duplicate the original copy (interv = 0) again, and create another
* variable 'interv2' to indicate the copy
expand 2 if interv == 0, generate(interv2)

*now, change the value of 'interv' to -1 in one of the copies so that there are
* unique values of interv for each copy*
replace interv = -1 if interv2 ==1
drop interv2

*check that the data has the structure you want: there should be 1566 people in
* each of the 3 levels of interv*
tab interv

*two of the copies will be for computing the standardized result*
*for these two copies (interv = 0 and interv = 1), set the outcome to missing
* and force qsmk to either 0 or 1, respectively*
*you may need to edit this part of the code for your outcome and exposure variables*
replace wt82_71 = . if interv != -1
replace qsmk = 0 if interv == 0
replace qsmk = 1 if interv == 1

*check that the data has the structure you want: for interv = -1, some people
* quit and some do not; for interv = 0 or 1, noone quits or everyone quits, respectively*
by interv, sort: summarize qsmk

*ii.Estimation in original sample*
*Now, we do a parametric regression with the covariates we want to adjust for*
*You may need to edit this part of the code for the variables you want.*
*Because the copies have missing wt82_71, this will only run the regression in
* the original copy*
regress wt82_71 qsmk sex race c.age##c.age ib(last).education ///
c.smokeintensity##c.smokeintensity c.smokeyrs##c.smokeyrs ///
ib(last).exercise ib(last).active c.wt71##c.wt71 qsmk#c.smokeintensity

*Ask Stata for expected values - Stata will give you expected values for all
* copies, not just the original ones*
predict predY, xb

*Now ask for a summary of these values by intervention*
*These are the standardized outcome estimates: you can subtract them to get the
* standardized difference*
by interv, sort: summarize predY

/* iii.OPTIONAL: Output standardized point estimates and difference
- The summary from the last command gives you the

```

```

standardized estimates
- We can stop there, or we can ask Stata to calculate the
standardized difference and display all the results
in a simple table
- The code below can be used as-is without changing any
variable names
- The option `quietly` asks Stata not to display the output of
some intermediate calculations
- You can delete this option if you want to see what is
happening step-by-step */
quietly summarize predY if(interv == -1)
matrix input observe = (-1,`r(mean)')
quietly summarize predY if(interv == 0)
matrix observe = (observe \0,`r(mean)')
quietly summarize predY if(interv == 1)
matrix observe = (observe \1,`r(mean)')
matrix observe = (observe \., observe[3,2]-observe[2,2])

* Add some row/column descriptions and print results to screen
matrix rownames observe = observed E(Y(a=0)) E(Y(a=1)) difference
matrix colnames observe = interv value
matrix list observe

/* To interpret these results:
- row 1, column 2, is the observed mean outcome value
in our original sample
- row 2, column 2, is the mean outcome value
if everyone had not quit smoking
- row 3, column 2, is the mean outcome value
if everyone had quit smoking
- row 4, column 2, is the mean difference outcome value
if everyone had quit smoking compared to if everyone
had not quit smoking */

/* Addition due to way Statamarkdown works
i.e. each code chunk is a separate Stata session */
mata observe = st_matrix("observe")
mata mata matsave ./data/observe observe, replace

*drop the copies*
drop if interv != -1
gen meanY_b =.
qui save ./data/nhefs_std, replace

```

(1,566 observations created)

(1,566 observations created)

(1,566 real changes made)

Expanded observation |

	type	Freq.	Percent	Cum.
	-1	1,566	33.33	33.33
Original observation		1,566	33.33	66.67
Duplicated observation		1,566	33.33	100.00
	Total	4,698	100.00	

(3,132 real changes made, 3,132 to missing)

(403 real changes made)

(1,163 real changes made)

-> interv = -1

Variable	Obs	Mean	Std. dev.	Min	Max
qsmk	1,566	.2573436	.4373099	0	1

-> interv = Original

Variable	Obs	Mean	Std. dev.	Min	Max
qsmk	1,566	0	0	0	0

-> interv = Duplicat

Variable	Obs	Mean	Std. dev.	Min	Max
qsmk	1,566	1	0	1	1

Source	SS	df	MS	Number of obs	=	1,566
				F(20, 1545)	=	13.45
Model	14412.558	20	720.6279	Prob > F	=	0.0000
Residual	82763.0286	1,545	53.5683033	R-squared	=	0.1483
				Adj R-squared	=	0.1373
Total	97175.5866	1,565	62.0930266	Root MSE	=	7.319

	wt82_71	Coefficient	Std. err.	t	P> t	[95% conf. interval]
	qsmk	2.559594	.8091486	3.16	0.002	.9724486 4.14674
	sex	-1.430272	.4689576	-3.05	0.002	-2.350132 -.5104111
	race	.5601096	.5818888	0.96	0.336	-.5812656 1.701485
	age	.3596353	.1633188	2.20	0.028	.0392854 .6799851
	c.age#c.age	-.006101	.0017261	-3.53	0.000	-.0094868 -.0027151

education							
1		.194977	.7413692	0.26	0.793	-1.259219	1.649173
2		.9854211	.7012116	1.41	0.160	-.390006	2.360848
3		.7512894	.6339153	1.19	0.236	-.4921358	1.994715
4		1.686547	.8716593	1.93	0.053	-.0232138	3.396307
smokeintensity		.0491365	.0517254	0.95	0.342	-.052323	.1505959
c.smokeintensity#							
c.smokeintensity		-.0009907	.000938	-1.06	0.291	-.0028306	.0008493
smokeyrs		.1343686	.0917122	1.47	0.143	-.045525	.3142621
c.smokeyrs#							
c.smokeyrs		-.0018664	.0015437	-1.21	0.227	-.0048944	.0011616
exercise							
0		-.3539128	.5588587	-0.63	0.527	-1.450114	.7422889
1		-.0579374	.4316468	-0.13	0.893	-.904613	.7887381
active							
0		.2613779	.6845577	0.38	0.703	-1.081382	1.604138
1		-.6861916	.6739131	-1.02	0.309	-2.008073	.6356894
wt71		.0455018	.0833709	0.55	0.585	-.1180303	.2090339
c.wt71#c.wt71		-.0009653	.0005247	-1.84	0.066	-.0019945	.0000639
qsmk#							
c.smokeintensity							
Smoking cessation		.0466628	.0351448	1.33	0.184	-.0222737	.1155993
_cons		-1.690608	4.388883	-0.39	0.700	-10.2994	6.918188

-> interv = -1

Variable		Obs	Mean	Std. dev.	Min	Max
-----+						
predY		1,566	2.6383	3.034683	-10.87582	9.876489

-> interv = Original

Variable		Obs	Mean	Std. dev.	Min	Max
-----+						
predY		1,566	1.756213	2.826271	-11.83737	6.733498

-> interv = Duplicat

Variable	Obs	Mean	Std. dev.	Min	Max
-----+-----					
predY	1,566	5.273587	2.920532	-9.091126	11.0506

```
observe[4,2]
      interv      value
observed      -1  2.6382998
E(Y(a=0))       0  1.7562131
E(Y(a=1))       1  5.2735873
difference       .  3.5173742
```

```
(saving observe[4,2])
file ./data/observe.mmat saved
```

```
(3,132 observations deleted)
```

```
(1,566 missing values generated)
```

Program 13.4

- Computing the 95% confidence interval of the standardized means and their difference: Data from NHEFS
- Section 13.3

```
*Run program 13.3 to obtain point estimates, and then the code below*
```

```
capture program drop bootstdz
```

```
program define bootstdz, rclass
use ./data/nhefs_std, clear
```

```
preserve
```

```
* Draw bootstrap sample from original observations
bsample
```

```
/* Create copies with each value of qsmk in bootstrap sample.
First, duplicate the dataset and create a variable `interv` which
indicates which copy is the duplicate (interv =1)*/
expand 2, generate(interv_b)
```

```

/* Next, duplicate the original copy (interv = 0) again, and create
another variable `interv2` to indicate the copy*/
expand 2 if interv_b == 0, generate(interv2_b)

/* Now, change the value of interv to -1 in one of the copies so that
there are unique values of interv for each copy*/
replace interv_b = -1 if interv2_b == 1
drop interv2_b

/* Two of the copies will be for computing the standardized result.
For these two copies (interv = 0 and interv = 1), set the outcome to
missing and force qsmk to either 0 or 1, respectively*/
replace wt82_71 = . if interv_b != -1
replace qsmk = 0 if interv_b == 0
replace qsmk = 1 if interv_b == 1

* Run regression
regress wt82_71 qsmk sex race c.age##c.age ib(last).education ///
    c.smokeintensity##c.smokeintensity c.smokeyrs##c.smokeyrs ///
    ib(last).exercise ib(last).active c.wt71##c.wt71 ///
    qsmk#c.smokeintensity

/* Ask Stata for expected values.
Stata will give you expected values for all copies, not just the
original ones*/
predict predY_b, xb
summarize predY_b if interv_b == 0
return scalar boot_0 = r(mean)
summarize predY_b if interv_b == 1
return scalar boot_1 = r(mean)
return scalar boot_diff = return(boot_1) - return(boot_0)
drop meanY_b

restore

end

/* Then we use the `simulate` command to run the bootstraps as many
times as we want.
Start with reps(10) to make sure your code runs, and then change to
reps(1000) to generate your final CIs.*/
simulate EY_a0=r(boot_0) EY_a1 = r(boot_1) ///
    difference = r(boot_diff), reps(10) seed(1): bootstdz

/* Next, format the point estimate to allow Stata to calculate our
standard errors and confidence intervals*/

* Addition: read back in the observe matrix
mata mata matuse ./data/observe, replace
mata st_matrix("observe", observe)

matrix pe = observe[2..4, 2]'

```

```
matrix list pe
```

```
/* Finally, the bstat command generates valid 95% confidence intervals
under the normal approximation using our bootstrap results.
The default results use a normal approximation to calculate the
confidence intervals.
Note, n contains the original sample size of your data before censoring*/
bstat, stat(pe) n(1629)
```

12.

```
Command: bootstdz
      EY_a0: r(boot_0)
      EY_a1: r(boot_1)
difference: r(boot_diff)
```

Simulations (10)

```
-----+--- 1 ----+--- 2 ----+--- 3 ----+--- 4 ----+--- 5
.....
```

(loading observe[4,2])

```
pe[1,3]
```

```
      r2      r3      r4
c2 1.7562131 5.2735873 3.5173742
```

Bootstrap results

Number of obs = 1,629
Replications = 10

		Observed	Bootstrap			Normal-based
		coefficient	std. err.	z	P> z	[95% conf. interval]

EY_a0		1.756213	.2157234	8.14	0.000	1.333403 2.179023
EY_a1		5.273587	.4999001	10.55	0.000	4.293801 6.253374
difference		3.517374	.538932	6.53	0.000	2.461087 4.573662

14. G-estimation of Structural Nested Models: Stata

```
library(Statamarkdown)
```

```
/******  
Stata code for Causal Inference: What If by Miguel Hernan & Jamie Robins  
Date: 10/10/2019  
Author: Eleanor Murray  
For errors contact: ejmurray@bu.edu  
******/
```

Program 14.1

- Ranks of extreme observations
- Data from NHEFS
- Section 14.4

```
/*For Stata 15 or later, first install the extremes function using this code:*/  
* ssc install extremes  
  
*Data preprocessing**  
  
use ./data/nhefs, clear  
gen byte cens = (wt82 == .)  
  
/*Ranking of extreme observations*/  
extremes wt82_71 seqn  
  
/*Estimate unstabilized censoring weights for use in g-estimation models*/  
glm cens qsmk sex race c.age##c.age ib(last).education ///  
    c.smokeintensity##c.smokeintensity c.smokeyrs##c.smokeyrs ///  
    ib(last).exercise ib(last).active c.wt71##c.wt71 ///  
    , family(binomial)  
predict pr_cens  
gen w_cens = 1/(1-pr_cens)  
replace w_cens = . if cens == 1  
/*observations with cens = 1 contribute to censoring models but not outcome model*/  
summarize w_cens  
  
/*Analyses restricted to N=1566*/  
drop if wt82 == .
```

```
summarize wt82_71
```

```
save ./data/nhefs-wcens, replace
```

```
| obs:      wt82_71    seqn |
|-----|
| 1329.   -41.28046982  23321 |
|  527.   -30.50192161  13593 |
| 1515.   -30.05007421  24363 |
|  204.   -29.02579305   5412 |
| 1067.   -25.97055814  21897 |
+-----+
```

```
+-----+
|  205.    34.01779932   5415 |
| 1145.    36.96925111  22342 |
|   64.    37.65051215   1769 |
|  260.    47.51130337   6928 |
| 1367.    48.53838568  23522 |
+-----+
```

```
Iteration 0:  log likelihood = -292.45812
Iteration 1:  log likelihood = -233.5099
Iteration 2:  log likelihood = -232.68635
Iteration 3:  log likelihood = -232.68
Iteration 4:  log likelihood = -232.67999
```

Generalized linear models

Optimization : ML

Deviance = 465.3599898

Pearson = 1654.648193

Number of obs = 1,629

Residual df = 1,609

Scale parameter = 1

(1/df) Deviance = .2892231

(1/df) Pearson = 1.028371

Variance function: $V(u) = u*(1-u)$

[Bernoulli]

Link function : $g(u) = \ln(u/(1-u))$

[Logit]

Log likelihood = -232.6799949

AIC = .3102271

BIC = -11434.36

```
-----
|                               OIM
|      cens | Coefficient std. err.    z    P>|z|    [95% conf. interval]
|-----+-----|
| qsmk | .5168674 .2877162    1.80    0.072    -.0470459    1.080781
| sex | .0573131 .3302775    0.17    0.862    -.590019    .7046452
| race | -.0122715 .4524888   -0.03    0.978    -.8991332    .8745902
| age | -.2697293 .1174647   -2.30    0.022    -.4999558    -.0395027
|
| c.age#c.age | .0028837 .0011135    2.59    0.010    .0007012    .0050661
|
| education |
| 1 | .3823818 .5601808    0.68    0.495    -.7155523    1.480316
|-----
```

2		-.0584066	.5749586	-0.10	0.919	-1.185305	1.068491
3		.2176937	.5225008	0.42	0.677	-.8063891	1.241776
4		.5208288	.6678735	0.78	0.435	-.7881792	1.829837
smokeintensity		.0157119	.0347319	0.45	0.651	-.0523614	.0837851
c.smokeintensity#							
c.smokeintensity		-.0001133	.0006058	-0.19	0.852	-.0013007	.0010742
smokeyrs		.0785973	.0749576	1.05	0.294	-.068317	.2255116
c.smokeyrs#							
c.smokeyrs		-.0005569	.0010318	-0.54	0.589	-.0025791	.0014653
exercise							
0		.583989	.3723133	1.57	0.117	-.1457317	1.31371
1		-.3874824	.3439133	-1.13	0.260	-1.06154	.2865753
active							
0		-.7065829	.3964577	-1.78	0.075	-1.483626	.0704599
1		-.9540614	.3893181	-2.45	0.014	-1.717111	-.1910119
wt71		-.0878871	.0400115	-2.20	0.028	-.1663082	-.0094659
c.wt71#c.wt71		.0006351	.0002257	2.81	0.005	.0001927	.0010775
_cons		3.754678	2.651222	1.42	0.157	-1.441622	8.950978

(option mu assumed; predicted mean cens)

(63 real changes made, 63 to missing)

Variable		Obs	Mean	Std. dev.	Min	Max
w_cens		1,566	1.039197	.05646	1.001814	1.824624

(63 observations deleted)

Variable		Obs	Mean	Std. dev.	Min	Max
wt82_71		1,566	2.6383	7.879913	-41.28047	48.53839

file ./data/nhefs-wcens.dta saved

Program 14.2

- G-estimation of a 1-parameter structural nested mean model
- Brute force search
- Data from NHEFS
- Section 14.5

```

use ./data/nhefs-wcens, clear

/*Generate test value of Psi = 3.446*/
gen psi = 3.446

/*Generate H(Psi) for each individual using test value of Psi and
their own values of weight change and smoking status*/
gen Hpsi = wt82_71 - psi * qsmk

/*Fit a model for smoking status, given confounders and H(Psi) value,
with censoring weights and display H(Psi) coefficient*/
logit qsmk sex race c.age##c.age ib(last).education ///
  c.smokeintensity##c.smokeintensity c.smokeyrs##c.smokeyrs ///
  ib(last).exercise ib(last).active c.wt71##c.wt71 Hpsi ///
  [pw = w_cens], cluster(seqn)
di _b[Hpsi]

/*G-estimation*/
/*Checking multiple possible values of psi*/
cap noi drop psi Hpsi

local seq_start = 2
local seq_end = 5
local seq_by = 0.1 // Setting seq_by = 0.01 will yield the result 3.46
local seq_len = (`seq_end' - `seq_start') / `seq_by' + 1

matrix results = J(`seq_len', 4, 0)

qui gen psi = .
qui gen Hpsi = .

local j = 0

forvalues i = `seq_start'(`seq_by')`seq_end' {
  local j = `j' + 1
  qui replace psi = `i'
  qui replace Hpsi = wt82_71 - psi * qsmk
  quietly logit qsmk sex race c.age##c.age ///
    ib(last).education c.smokeintensity##c.smokeintensity ///
    c.smokeyrs##c.smokeyrs ib(last).exercise ib(last).active ///
    c.wt71##c.wt71 Hpsi ///
    [pw = w_cens], cluster(seqn)
  matrix p_mat = r(table)
  matrix p_mat = p_mat["pvalue", "qsmk:Hpsi"]
  local p = p_mat[1,1]
  local b = _b[Hpsi]
  di "coeff", %6.3f `b', "is generated from psi", %4.1f `i'
  matrix results[`j',1] = `i'
  matrix results[`j',2] = `b'
  matrix results[`j',3] = abs(`b')
  matrix results[`j',4] = `p'
}

```



```

matrix colnames results = "psi" "B(Hpsi)" "AbsB(Hpsi)" "pvalue"
mat li results

mata
res = st_matrix("results")
for(i=1; i<= rows(res); i++) {
    if (res[i,3] == colmin(res[,3])) res[i,1]
}
end
* Setting seq_by = 0.01 will yield the result 3.46

```

```

Iteration 0: log pseudolikelihood = -936.10067
Iteration 1: log pseudolikelihood = -879.13942
Iteration 2: log pseudolikelihood = -877.82647
Iteration 3: log pseudolikelihood = -877.82423
Iteration 4: log pseudolikelihood = -877.82423

```

Logistic regression

Number of obs = 1,566

Wald chi2(19) = 106.13

Prob > chi2 = 0.0000

Log pseudolikelihood = -877.82423

Pseudo R2 = 0.0623

(Std. err. adjusted for 1,566 clusters in seqn)

	qsmk	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
sex		-.5137324	.1536024	-3.34	0.001	-.8147876	-.2126772
race		-.8608912	.2099415	-4.10	0.000	-1.272369	-.4494133
age		.1151589	.0502116	2.29	0.022	.016746	.2135718
c.age#c.age		-.0007593	.0005297	-1.43	0.152	-.0017976	.000279
education							
1		-.4710855	.2247701	-2.10	0.036	-.9116268	-.0305441
2		-.5000231	.2208583	-2.26	0.024	-.9328974	-.0671487
3		-.3833788	.195914	-1.96	0.050	-.7673632	.0006056
4		-.4047116	.2836068	-1.43	0.154	-.9605707	.1511476
smokeintensity		-.0783425	.014645	-5.35	0.000	-.1070461	-.0496389
c.smokeintensity#							
c.smokeintensity		.0010722	.0002651	4.04	0.000	.0005526	.0015917
smokeyrs		-.0711097	.026398	-2.69	0.007	-.1228488	-.0193705
c.smokeyrs#							
c.smokeyrs		.0008153	.0004491	1.82	0.069	-.000065	.0016955
exercise							
0		-.3800465	.1889205	-2.01	0.044	-.7503238	-.0097692
1		-.0437043	.1372725	-0.32	0.750	-.3127534	.2253447

active							
0		-.2134552	.2122025	-1.01	0.314	-.6293645	.2024541
1		-.1793327	.207151	-0.87	0.387	-.5853412	.2266758
wt71		-.0076607	.0256319	-0.30	0.765	-.0578983	.0425769
c.wt71#c.wt71		.0000866	.0001582	0.55	0.584	-.0002236	.0003967
Hpsi		-1.90e-06	.0088414	-0.00	1.000	-.0173307	.0173269
_cons		-1.338367	1.359613	-0.98	0.325	-4.00316	1.326426

-1.905e-06

```

6.      matrix p_mat = r(table)
7.      matrix p_mat = p_mat["pvalue","qsmk:Hpsi"]
8.      local p = p_mat[1,1]
9.      local b = _b[Hpsi]
10.     di "coeff", %6.3f `b', "is generated from psi", %4.1f `i'
11.     matrix results[`j',1]= `i'
12.     matrix results[`j',2]= `b'
13.     matrix results[`j',3]= abs(`b')
14.     matrix results[`j',4]= `p'
15. }
coeff 0.027 is generated from psi 2.0
coeff 0.025 is generated from psi 2.1
coeff 0.023 is generated from psi 2.2
coeff 0.021 is generated from psi 2.3
coeff 0.019 is generated from psi 2.4
coeff 0.018 is generated from psi 2.5
coeff 0.016 is generated from psi 2.6
coeff 0.014 is generated from psi 2.7
coeff 0.012 is generated from psi 2.8
coeff 0.010 is generated from psi 2.9
coeff 0.008 is generated from psi 3.0
coeff 0.006 is generated from psi 3.1
coeff 0.005 is generated from psi 3.2
coeff 0.003 is generated from psi 3.3
coeff 0.001 is generated from psi 3.4
coeff -0.001 is generated from psi 3.5
coeff -0.003 is generated from psi 3.6
coeff -0.005 is generated from psi 3.7
coeff -0.007 is generated from psi 3.8
coeff -0.009 is generated from psi 3.9

```

```

coeff -0.011 is generated from psi 4.0
coeff -0.012 is generated from psi 4.1
coeff -0.014 is generated from psi 4.2
coeff -0.016 is generated from psi 4.3
coeff -0.018 is generated from psi 4.4
coeff -0.020 is generated from psi 4.5
coeff -0.022 is generated from psi 4.6
coeff -0.024 is generated from psi 4.7
coeff -0.026 is generated from psi 4.8
coeff -0.028 is generated from psi 4.9
coeff -0.030 is generated from psi 5.0

```

```
results[31,4]
```

	psi	B(Hpsi)	AbsB(Hpsi)	pvalue
r1	2	.02672188	.02672188	.00177849
r2	2.1	.02489456	.02489456	.00359089
r3	2.2	.02306552	.02306552	.00698119
r4	2.3	.02123444	.02123444	.01305479
r5	2.4	.01940095	.01940095	.02346121
r6	2.5	.01756472	.01756472	.04049437
r7	2.6	.0157254	.0157254	.06710192
r8	2.7	.01388267	.01388267	.10673812
r9	2.8	.0120362	.0120362	.16301154
r10	2.9	.01018567	.01018567	.23912864
r11	3	.00833081	.00833081	.33720241
r12	3.1	.00647131	.00647131	.45757692
r13	3.2	.0046069	.0046069	.59835195
r14	3.3	.00273736	.00273736	.75528009
r15	3.4	.00086243	.00086243	.92212566
r16	3.5	-.00101809	.00101809	.90856559
r17	3.6	-.00290439	.00290439	.7444406
r18	3.7	-.00479666	.00479666	.59230593
r19	3.8	-.00669505	.00669505	.45731304
r20	3.9	-.00859969	.00859969	.3425138
r21	4	-.01051072	.01051072	.2488326
r22	4.1	-.01242824	.01242824	.17537691
r23	4.2	-.01435235	.01435235	.1199593
r24	4.3	-.01628313	.01628313	.07967563
r25	4.4	-.01822063	.01822063	.05142147
r26	4.5	-.02016492	.02016492	.03227271
r27	4.6	-.02211603	.02211603	.01971433
r28	4.7	-.02407401	.02407401	.01173271
r29	4.8	-.02603888	.02603888	.00680955
r30	4.9	-.02801063	.02801063	.00385828
r31	5	-.02998926	.02998926	.00213639

```

----- mata (type end to exit) -----
: res = st_matrix("results")

: for(i=1; i<= rows(res); i++) {
>   if (res[i,3] == colmin(res[,3])) res[i,1]

```

```
> }
3.4

: end
```

Program 14.3

- G-estimation for 2-parameter structural nested mean model
- Closed form estimator
- Data from NHEFS
- Section 14.6

```
use ./data/nhefs-wcens, clear

/*create weights*/
logit qsmk sex race c.age##c.age ib(last).education ///
    c.smokeintensity##c.smokeintensity c.smokeyrs##c.smokeyrs ///
    ib(last).exercise ib(last).active c.wt71##c.wt71 ///
    [pw = w_cens], cluster(seqn)
predict pr_qsmk
summarize pr_qsmk

/* Closed form estimator linear mean models */
* ssc install tomata
putmata *, replace
mata: diff = qsmk - pr_qsmk
mata: part1 = w_cens :* wt82_71 :* diff
mata: part2 = w_cens :* qsmk :* diff
mata: psi = sum(part1)/sum(part2)

/** Closed form estimator for 2-parameter model */
mata
diff = qsmk - pr_qsmk
diff2 = w_cens :* diff

lhs = J(2,2, 0)
lhs[1,1] = sum( qsmk :* diff2)
lhs[1,2] = sum( qsmk :* smokeintensity :* diff2 )
lhs[2,1] = sum( qsmk :* smokeintensity :* diff2)
lhs[2,2] = sum( qsmk :* smokeintensity :* smokeintensity :* diff2 )

rhs = J(2,1,0)
rhs[1] = sum(wt82_71 :* diff2 )
rhs[2] = sum(wt82_71 :* smokeintensity :* diff2 )

psi = (lusolve(lhs, rhs))'
psi
psi = (invsym(lhs'lhs)*lhs'rhs)'
psi
end
```

```
Iteration 0:    log pseudolikelihood = -936.10067
```

Iteration 1: log pseudolikelihood = -879.13943
 Iteration 2: log pseudolikelihood = -877.82647
 Iteration 3: log pseudolikelihood = -877.82423
 Iteration 4: log pseudolikelihood = -877.82423

Logistic regression

Number of obs = 1,566

Wald chi2(18) = 106.13

Prob > chi2 = 0.0000

Log pseudolikelihood = -877.82423

Pseudo R2 = 0.0623

(Std. err. adjusted for 1,566 clusters in seqn)

		Robust				
	qsmk	Coefficient	std. err.	z	P> z	[95% conf. interval]
sex		-.5137295	.1533507	-3.35	0.001	-.8142913 -.2131677
race		-.8608919	.2099555	-4.10	0.000	-1.272397 -.4493867
age		.1151581	.0503079	2.29	0.022	.0165564 .2137598
c.age#c.age		-.0007593	.00053	-1.43	0.152	-.0017981 .0002795
education						
1		-.4710854	.2247796	-2.10	0.036	-.9116454 -.0305255
2		-.5000247	.220776	-2.26	0.024	-.9327378 -.0673116
3		-.3833802	.1954991	-1.96	0.050	-.7665515 -.0002089
4		-.4047148	.2833093	-1.43	0.153	-.9599908 .1505613
smokeintensity		-.0783426	.0146634	-5.34	0.000	-.1070824 -.0496029
c.smokeintensity#						
c.smokeintensity		.0010722	.0002655	4.04	0.000	.0005518 .0015925
smokeyrs		-.0711099	.0263523	-2.70	0.007	-.1227596 -.0194602
c.smokeyrs#						
c.smokeyrs		.0008153	.0004486	1.82	0.069	-.0000639 .0016945
exercise						
0		-.3800461	.1890123	-2.01	0.044	-.7505034 -.0095887
1		-.0437044	.137269	-0.32	0.750	-.3127467 .225338
active						
0		-.2134564	.2121759	-1.01	0.314	-.6293135 .2024007
1		-.1793322	.2070848	-0.87	0.386	-.5852109 .2265466
wt71		-.0076609	.0255841	-0.30	0.765	-.0578048 .042483
c.wt71#c.wt71		.0000866	.0001572	0.55	0.582	-.0002216 .0003947
_cons		-1.338358	1.359289	-0.98	0.325	-4.002516 1.3258

(option pr assumed; Pr(qsmk))

Variable	Obs	Mean	Std. dev.	Min	Max
pr_qsmk	1,566	.2607709	.1177584	.0514466	.7891403

(68 vectors posted)

```

----- mata (type end to exit) -----
: diff = qsmk - pr_qsmk

: diff2 = w_cens :* diff

:
: lhs = J(2,2, 0)

: lhs[1,1] = sum( qsmk :* diff2)

: lhs[1,2] = sum( qsmk :* smokeintensity :* diff2 )

: lhs[2,1] = sum( qsmk :* smokeintensity :* diff2)

: lhs[2,2] = sum( qsmk :* smokeintensity :* smokeintensity :* diff2 )

:
: rhs = J(2,1,0)

: rhs[1] = sum(wt82_71 :* diff2 )

: rhs[2] = sum(wt82_71 :* smokeintensity :* diff2 )

:
: psi = (lusolve(lhs, rhs))'

: psi
      1          2
+-----+
1 |  2.859470362   .0300412816 |
+-----+

: psi = (invsym(lhs'lhs)*lhs'rhs)'

: psi
      1          2
+-----+
1 |  2.859470362   .0300412816 |
+-----+

: end
-----

```

15. Outcome regression and propensity scores: Stata

```
library(Statamarkdown)
```

```
/******  
Stata code for Causal Inference: What If by Miguel Hernan & Jamie Robins  
Date: 10/10/2019  
Author: Eleanor Murray  
For errors contact: ejmurray@bu.edu  
******/
```

Program 15.1

- Estimating the average causal effect within levels of confounders under the assumption of effect-measure modification by smoking intensity ONLY
- Data from NHEFS
- Section 15.1

```
use ./data/nhefs-formatted, clear  
  
/* Generate smoking intensity among smokers product term */  
gen qsmkintensity = qsmk*smokeintensity  
  
* Regression on covariates, allowing for some effect modification  
regress wt82_71 qsmk qsmkintensity ///  
    c.smokeintensity##c.smokeintensity sex race c.age##c.age ///  
    ib(last).education c.smokeyrs##c.smokeyrs ///  
    ib(last).exercise ib(last).active c.wt71##c.wt71  
  
/* Display the estimated mean difference between quitting and  
    not quitting value when smoke intensity = 5 cigarettes/ day */  
lincom 1*_b[qsmk] + 5*1*_b[qsmkintensity]  
  
/* Display the estimated mean difference between quitting and  
    not quitting value when smoke intensity = 40 cigarettes/ day */  
lincom 1*_b[qsmk] + 40*1*_b[qsmkintensity]  
  
/* Regression on covariates, with no product terms */  
regress wt82_71 qsmk c.smokeintensity##c.smokeintensity ///  
    sex race c.age##c.age ///  
    ib(last).education c.smokeyrs##c.smokeyrs ///
```

ib(last).exercise ib(last).active c.wt71##c.wt71

Source	SS	df	MS	Number of obs	=	1,566
Model	14412.558	20	720.6279	F(20, 1545)	=	13.45
Residual	82763.0286	1,545	53.5683033	Prob > F	=	0.0000
				R-squared	=	0.1483
				Adj R-squared	=	0.1373
Total	97175.5866	1,565	62.0930266	Root MSE	=	7.319

wt82_71	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
qsmk	2.559594	.8091486	3.16	0.002	.9724486	4.14674
qsmkintensity	.0466628	.0351448	1.33	0.184	-.0222737	.1155993
smokeintensity	.0491365	.0517254	0.95	0.342	-.052323	.1505959
c.smokeintensity#						
c.smokeintensity	-.0009907	.000938	-1.06	0.291	-.0028306	.0008493
sex	-1.430272	.4689576	-3.05	0.002	-2.350132	-.5104111
race	.5601096	.5818888	0.96	0.336	-.5812656	1.701485
age	.3596353	.1633188	2.20	0.028	.0392854	.6799851
c.age#c.age	-.006101	.0017261	-3.53	0.000	-.0094868	-.0027151
education						
1	.194977	.7413692	0.26	0.793	-1.259219	1.649173
2	.9854211	.7012116	1.41	0.160	-.390006	2.360848
3	.7512894	.6339153	1.19	0.236	-.4921358	1.994715
4	1.686547	.8716593	1.93	0.053	-.0232138	3.396307
smokeysrs	.1343686	.0917122	1.47	0.143	-.045525	.3142621
c.smokeysrs#						
c.smokeysrs	-.0018664	.0015437	-1.21	0.227	-.0048944	.0011616
exercise						
0	-.3539128	.5588587	-0.63	0.527	-1.450114	.7422889
1	-.0579374	.4316468	-0.13	0.893	-.904613	.7887381
active						
0	.2613779	.6845577	0.38	0.703	-1.081382	1.604138
1	-.6861916	.6739131	-1.02	0.309	-2.008073	.6356894
wt71	.0455018	.0833709	0.55	0.585	-.1180303	.2090339
c.wt71#c.wt71	-.0009653	.0005247	-1.84	0.066	-.0019945	.0000639
_cons	-1.690608	4.388883	-0.39	0.700	-10.2994	6.918188

(1) qsmk + 5*qsmkintensity = 0

wt82_71	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
(1)	2.792908	.6682596	4.18	0.000	1.482117	4.1037

(1) qsmk + 40*qsmkintensity = 0

wt82_71	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
(1)	4.426108	.8477818	5.22	0.000	2.763183	6.089032

Source	SS	df	MS	Number of obs	=	1,566
				F(19, 1546)	=	14.06
Model	14318.1239	19	753.58547	Prob > F	=	0.0000
Residual	82857.4627	1,546	53.5947365	R-squared	=	0.1473
				Adj R-squared	=	0.1369
Total	97175.5866	1,565	62.0930266	Root MSE	=	7.3208

wt82_71	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
qsmk	3.462622	.4384543	7.90	0.000	2.602594	4.32265
smokeintensity	.0651533	.0503115	1.29	0.196	-.0335327	.1638392
c.smokeintensity#						
c.smokeintensity	-.0010468	.0009373	-1.12	0.264	-.0028853	.0007918
sex	-1.46505	.468341	-3.13	0.002	-2.3837	-.5463989
race	.5864117	.5816949	1.01	0.314	-.5545827	1.727406
age	.3626624	.1633431	2.22	0.027	.0422649	.6830599
c.age#c.age	-.0061377	.0017263	-3.56	0.000	-.0095239	-.0027515
education						
1	.1708264	.7413289	0.23	0.818	-1.28329	1.624943
2	.9893527	.7013784	1.41	0.159	-.3864007	2.365106
3	.7423268	.6340357	1.17	0.242	-.501334	1.985988
4	1.679344	.8718575	1.93	0.054	-.0308044	3.389492
smokeyrs	.1333931	.0917319	1.45	0.146	-.0465389	.3133252
c.smokeyrs#						
c.smokeyrs	-.001827	.0015438	-1.18	0.237	-.0048552	.0012012
exercise						
0	-.3628786	.5589557	-0.65	0.516	-1.45927	.7335129
1	-.0421962	.4315904	-0.10	0.922	-.8887606	.8043683

active							
0		.2580374	.6847219	0.38	0.706	-1.085044	1.601119
1		-.68492	.6740787	-1.02	0.310	-2.007125	.6372851
wt71		.0373642	.0831658	0.45	0.653	-.1257655	.200494
c.wt71#c.wt71		-.0009158	.0005235	-1.75	0.080	-.0019427	.0001111
_cons		-1.724603	4.389891	-0.39	0.694	-10.33537	6.886166

Prorgam 15.2

- Estimating and plotting the propensity score
- Data from NHEFS
- Section 15.2

```
use ./data/nhefs-formatted, clear

/*Fit a model for the exposure, quitting smoking*/
logit qsmk sex race c.age##c.age ib(last).education ///
    c.smokeintensity##c.smokeintensity ///
    c.smokeyrs##c.smokeyrs ib(last).exercise ib(last).active ///
    c.wt71##c.wt71

/*Estimate the propensity score, P(Qsmk|Covariates)*/
predict ps, pr

/*Check the distribution of the propensity score*/
bys qsmk: summarize ps

/*Return extreme values of propensity score:
    note, for Stata versions 15 and above, start by installing extremes*/
* ssc install extremes
extremes ps seqn
bys qsmk: extremes ps seqn

save ./data/nhefs-ps, replace

/*Plotting the estimated propensity score*/
histogram ps, width(0.05) start(0.025) ///
    frequency fcolor(none) lcolor(black) ///
    lpattern(solid) addlabel ///
    addlabopts(mlabcolor(black) mlabposition(12) ///
    mlabangle(zero)) ///
    ytitle(No. Subjects) ylabel(#4) ///
    xtitle(Estimated Propensity Score) xlabel(#15) ///
    by(, title(Estimated Propensity Score Distribution) ///
    subtitle(By Quit Smoking Status)) ///
    by(, legend(off)) ///
    by(qsmk, style(compact) colfirst) ///
    subtitle(, size(small) box bexpand)
qui gr export ./figs/stata-fig-15-2.png, replace
```

Iteration 0: log likelihood = -893.02712
 Iteration 1: log likelihood = -839.70016
 Iteration 2: log likelihood = -838.45045
 Iteration 3: log likelihood = -838.44842
 Iteration 4: log likelihood = -838.44842

Logistic regression

Number of obs = 1,566

LR chi2(18) = 109.16

Prob > chi2 = 0.0000

Pseudo R2 = 0.0611

Log likelihood = -838.44842

qsmk	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
sex	-.5274782	.1540497	-3.42	0.001	-.82941	-.2255463
race	-.8392636	.2100668	-4.00	0.000	-1.250987	-.4275404
age	.1212052	.0512663	2.36	0.018	.0207251	.2216853
c.age#c.age	-.0008246	.0005361	-1.54	0.124	-.0018753	.0002262
education						
1	-.4759606	.2262238	-2.10	0.035	-.9193511	-.0325701
2	-.5047361	.217597	-2.32	0.020	-.9312184	-.0782538
3	-.3895288	.1914353	-2.03	0.042	-.7647351	-.0143226
4	-.4123596	.2772868	-1.49	0.137	-.9558318	.1311126
smokeintensity	-.0772704	.0152499	-5.07	0.000	-.1071596	-.0473812
c.smokeintensity#						
c.smokeintensity	.0010451	.0002866	3.65	0.000	.0004835	.0016068
smokekeys	-.0735966	.0277775	-2.65	0.008	-.1280395	-.0191538
c.smokekeys#						
c.smokekeys	.0008441	.0004632	1.82	0.068	-.0000637	.0017519
exercise						
0	-.395704	.1872401	-2.11	0.035	-.7626878	-.0287201
1	-.0408635	.1382674	-0.30	0.768	-.3118627	.2301357
active						
0	-.176784	.2149721	-0.82	0.411	-.5981215	.2445535
1	-.1448395	.2111472	-0.69	0.493	-.5586806	.2690015
wt71	-.0152357	.0263161	-0.58	0.563	-.0668144	.036343
c.wt71#c.wt71	.0001352	.0001632	0.83	0.407	-.0001846	.000455
_cons	-1.19407	1.398493	-0.85	0.393	-3.935066	1.546925

```
-----
```

```
-> qsmk = No smoking cessation
```

Variable	Obs	Mean	Std. dev.	Min	Max
ps	1,163	.2392928	.1056545	.0510008	.6814955

```
-----
```

```
-> qsmk = Smoking cessation
```

Variable	Obs	Mean	Std. dev.	Min	Max
ps	403	.3094353	.1290642	.0598799	.7768887

```
+-----+
```

obs:	ps	seqn
979.	.0510008	22941
945.	.0527126	1769
1023.	.0558418	21140
115.	.0558752	2522
478.	.0567372	12639

```
+-----+
```

```
+-----+
```

1173.	.6659576	22272
1033.	.6814955	22773
1551.	.7166381	14983
1494.	.7200644	24817
1303.	.7768887	24949

```
+-----+
```

```
-----
```

```
-> qsmk = No smoking cessation
```

```
+-----+
```

obs:	ps	seqn
979.	.0510008	22941
945.	.0527126	1769
1023.	.0558418	21140
115.	.0558752	2522
478.	.0567372	12639

```
+-----+
```

```
+-----+
```

463.	.6337243	17096
812.	.6345721	17768
707.	.6440308	19147
623.	.6566707	21983

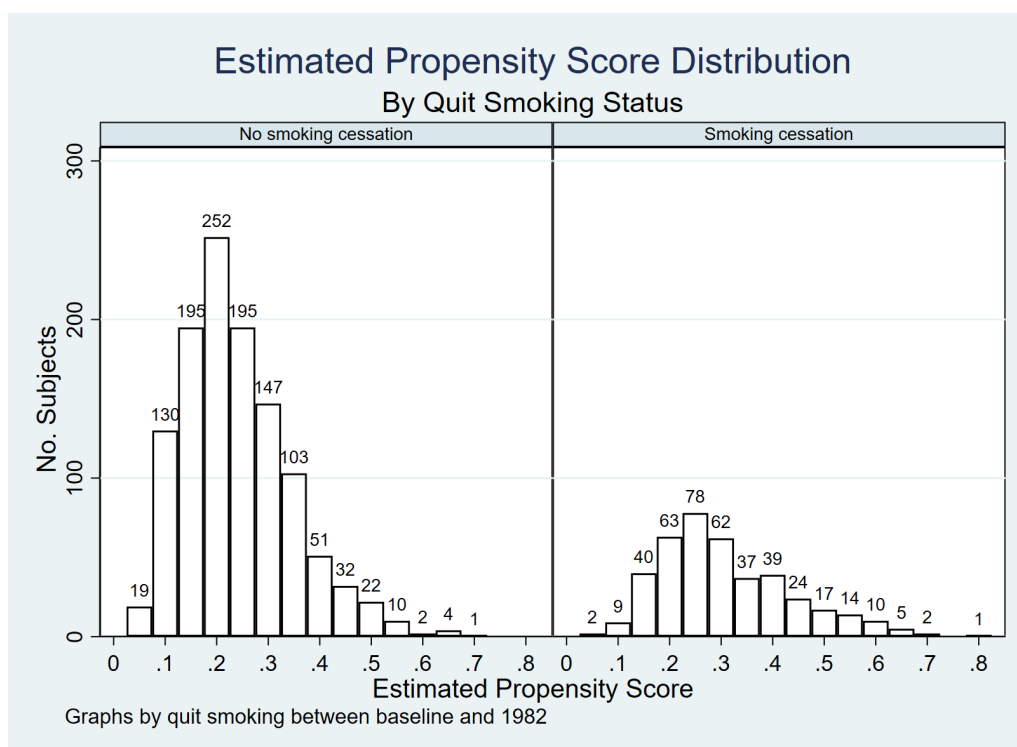
```
| 1033.    .6814955    22773 |
+-----+
```

-> qsmk = Smoking cessation

```
+-----+
| obs:      ps    seqn |
+-----+
| 1223.    .0598799    4289 |
| 1283.    .0600822    23550 |
| 1253.    .0806089    24306 |
| 1467.    .0821677    22904 |
| 1165.    .1021875    24584 |
+-----+
```

```
+-----+
| 1399.    .635695    17738 |
| 1173.    .6659576    22272 |
| 1551.    .7166381    14983 |
| 1494.    .7200644    24817 |
| 1303.    .7768887    24949 |
+-----+
```

file ./data/nhefs-ps.dta saved



Program 15.3

- Stratification and outcome regression using deciles of the propensity score
- Data from NHEFS
- Section 15.3

- Note: Stata decides borderline cutpoints differently from SAS, so, despite identically distributed propensity scores, the results of regression using deciles are not an exact match with the book.

```
use ./data/nhefs-ps, clear
```

```
/*Calculation of deciles of ps*/
```

```
xtile ps_dec = ps, nq(10)
```

```
by ps_dec, sort: summarize ps
```

```
/*Stratification on PS deciles, allowing for effect modification*/
```

```
/*Note: Stata compares qsmk 0 vs qsmk 1, so the coefficients are reversed  
relative to the book*/
```

```
by ps_dec: ttest wt82_71, by(qsmk)
```

```
/*Regression on PS deciles, with no product terms*/
```

```
regress wt82_71 qsmk ib(last).ps_dec
```

```
-> ps_dec = 1
```

Variable	Obs	Mean	Std. dev.	Min	Max
ps	157	.0976251	.0185215	.0510008	.1240482

```
-> ps_dec = 2
```

Variable	Obs	Mean	Std. dev.	Min	Max
ps	157	.1430792	.0107751	.1241923	.1603558

```
-> ps_dec = 3
```

Variable	Obs	Mean	Std. dev.	Min	Max
ps	156	.1750423	.008773	.1606041	.1893271

```
-> ps_dec = 4
```

Variable	Obs	Mean	Std. dev.	Min	Max
ps	157	.2014066	.0062403	.189365	.2121815

```
-> ps_dec = 5
```

Variable	Obs	Mean	Std. dev.	Min	Max
ps	156	.2245376	.0073655	.2123068	.237184

```
-> ps_dec = 6
```

Variable	Obs	Mean	Std. dev.	Min	Max
ps	157	.2515298	.0078777	.2377578	.2655718

Variable	Obs	Mean	Std. dev.	Min	Max
ps	157	.2827476	.0099986	.2655724	.2994968

Variable	Obs	Mean	Std. dev.	Min	Max
ps	156	.3204104	.0125102	.2997581	.3438773

Variable	Obs	Mean	Std. dev.	Min	Max
ps	157	.375637	.0221347	.3439862	.4174631

Variable	Obs	Mean	Std. dev.	Min	Max
ps	156	.5026508	.0733494	.4176717	.7768887

```
-> ps_dec = 1
```

Group	Obs	Mean	Std. err.	Std. dev.	[95% conf. interval]	
No smoki	146	3.74236	.6531341	7.891849	2.451467	5.033253
Smoking	11	3.949703	2.332995	7.737668	-1.248533	9.14794
Combined	157	3.756887	.6270464	7.856869	2.51829	4.995484
diff		-.2073431	2.464411		-5.075509	4.660822

Ha: diff < 0	Ha: diff != 0	Ha: diff > 0
Pr(T < t) = 0.4665	Pr(T > t) = 0.9331	Pr(T > t) = 0.5335

-> ps_dec = 2

Two-sample t test with equal variances

```
-----
  Group |      Obs      Mean   Std. err.   Std. dev.   [95% conf. interval]
-----+-----
No smoki |     134    2.813019    .589056    6.818816    1.647889    3.978149
Smoking  |      23    7.726944    1.260784    6.046508    5.112237    10.34165
-----+-----
Combined |     157    3.532893    .5519826   6.916322    2.442569    4.623217
-----+-----
      diff |          -4.913925    1.515494              -7.907613   -1.920237
-----

      diff = mean(No smoki) - mean(Smoking)              t =   -3.2425
H0: diff = 0                      Degrees of freedom =      155

      Ha: diff < 0              Ha: diff != 0              Ha: diff > 0
Pr(T < t) = 0.0007      Pr(|T| > |t|) = 0.0015      Pr(T > t) = 0.9993
-----
```

-> ps_dec = 3

Two-sample t test with equal variances

```
-----
  Group |      Obs      Mean   Std. err.   Std. dev.   [95% conf. interval]
-----+-----
No smoki |     128    3.25684    .5334655   6.035473    2.201209    4.312472
Smoking  |      28    7.954974    1.418184    7.504324    5.045101    10.86485
-----+-----
Combined |     156    4.100095    .5245749   6.551938    3.063857    5.136334
-----+-----
      diff |          -4.698134    1.318074              -7.301973   -2.094294
-----

      diff = mean(No smoki) - mean(Smoking)              t =   -3.5644
H0: diff = 0                      Degrees of freedom =      154

      Ha: diff < 0              Ha: diff != 0              Ha: diff > 0
Pr(T < t) = 0.0002      Pr(|T| > |t|) = 0.0005      Pr(T > t) = 0.9998
-----
```

-> ps_dec = 4

Two-sample t test with equal variances

```
-----
  Group |      Obs      Mean   Std. err.   Std. dev.   [95% conf. interval]
-----+-----
No smoki |     121    3.393929    .5267602   5.794362    2.350981    4.436877
Smoking  |      36    5.676072    1.543143    9.258861    2.543324    8.808819
-----+-----
Combined |     157    3.917223    .5412091   6.78133    2.848179    4.986266
-----+-----
```



```

diff |          -2.282143    1.278494          -4.807663    .2433778
-----
diff = mean(No smoki) - mean(Smoking)          t = -1.7850
H0: diff = 0          Degrees of freedom = 155

```

```

Ha: diff < 0          Ha: diff != 0          Ha: diff > 0
Pr(T < t) = 0.0381    Pr(|T| > |t|) = 0.0762    Pr(T > t) = 0.9619

```

```

-----
-> ps_dec = 5

```

Two-sample t test with equal variances

```

-----
Group |      Obs      Mean   Std. err.   Std. dev.   [95% conf. interval]
-----+-----
No smoki |      119    1.368438   .8042619    8.773461   -.2242199    2.961095
Smoking |       37    5.195421   1.388723    8.44727    2.378961    8.011881
-----+-----
Combined |      156    2.27612   .7063778    8.822656   .8807499    3.671489
-----+-----
diff |          -3.826983    1.637279          -7.061407   -.592559
-----

```

```

diff = mean(No smoki) - mean(Smoking)          t = -2.3374
H0: diff = 0          Degrees of freedom = 154

```

```

Ha: diff < 0          Ha: diff != 0          Ha: diff > 0
Pr(T < t) = 0.0104    Pr(|T| > |t|) = 0.0207    Pr(T > t) = 0.9896

```

```

-----
-> ps_dec = 6

```

Two-sample t test with equal variances

```

-----
Group |      Obs      Mean   Std. err.   Std. dev.   [95% conf. interval]
-----+-----
No smoki |      112    2.25564   .6850004    7.249362   .8982664    3.613014
Smoking |       45    7.199088   1.724899   11.57097    3.722782   10.67539
-----+-----
Combined |      157    3.672552   .7146582    8.954642    2.260897    5.084207
-----+-----
diff |          -4.943447    1.535024          -7.975714   -1.911181
-----

```

```

diff = mean(No smoki) - mean(Smoking)          t = -3.2204
H0: diff = 0          Degrees of freedom = 155

```

```

Ha: diff < 0          Ha: diff != 0          Ha: diff > 0
Pr(T < t) = 0.0008    Pr(|T| > |t|) = 0.0016    Pr(T > t) = 0.9992

```

```

-----
-> ps_dec = 7

```

Two-sample t test with equal variances

```

      Group |      Obs      Mean   Std. err.   Std. dev.   [95% conf. interval]
-----+-----
No smoki |      116   .7948483   .7916172   8.525978   -.773193    2.36289
Smoking |       41   6.646091   1.00182   6.414778   4.621337    8.670844
-----+-----
Combined |      157   2.32288   .6714693   8.413486   .9965349    3.649225
-----+-----
      diff |          -5.851242    1.45977          -8.734853   -2.967632
-----+-----
      diff = mean(No smoki) - mean(Smoking)          t =  -4.0083
H0: diff = 0          Degrees of freedom =      155

      Ha: diff < 0          Ha: diff != 0          Ha: diff > 0
Pr(T < t) = 0.0000          Pr(|T| > |t|) = 0.0001          Pr(T > t) = 1.0000

```

```

-----
-> ps_dec = 8

```

Two-sample t test with equal variances

```

      Group |      Obs      Mean   Std. err.   Std. dev.   [95% conf. interval]
-----+-----
No smoki |      107   1.063848   .5840159   6.041107   -.0940204    2.221716
Smoking |       49   3.116263   1.113479   7.794356   .8774626    5.355063
-----+-----
Combined |      156   1.708517   .5352016   6.684666   .6512864    2.765747
-----+-----
      diff |          -2.052415    1.144914          -4.31418    .2093492
-----+-----
      diff = mean(No smoki) - mean(Smoking)          t =  -1.7926
H0: diff = 0          Degrees of freedom =      154

      Ha: diff < 0          Ha: diff != 0          Ha: diff > 0
Pr(T < t) = 0.0375          Pr(|T| > |t|) = 0.0750          Pr(T > t) = 0.9625

```

```

-----
-> ps_dec = 9

```

Two-sample t test with equal variances

```

      Group |      Obs      Mean   Std. err.   Std. dev.   [95% conf. interval]
-----+-----
No smoki |      100  -.0292906   .7637396   7.637396  -1.544716    1.486134
Smoking |       57   .9112647   .9969309   7.526663  -1.085828    2.908357
-----+-----
Combined |      157   .3121849   .6054898   7.586766  -.8838316    1.508201
-----+-----
      diff |          -.9405554    1.26092          -3.43136    1.550249
-----+-----
      diff = mean(No smoki) - mean(Smoking)          t =  -0.7459
H0: diff = 0          Degrees of freedom =      155

      Ha: diff < 0          Ha: diff != 0          Ha: diff > 0

```

Pr(T < t) = 0.2284 Pr(|T| > |t|) = 0.4568 Pr(T > t) = 0.7716

-> ps_dec = 10

Two-sample t test with equal variances

Group	Obs	Mean	Std. err.	Std. dev.	[95% conf. interval]	
No smoki	80	-.768504	.9224756	8.250872	-2.604646	1.067638
Smoking	76	2.39532	1.053132	9.180992	.2973737	4.493267
Combined	156	.7728463	.7071067	8.831759	-.6239631	2.169656
diff		-3.163824	1.396178		-5.921957	-.405692

diff = mean(No smoki) - mean(Smoking) t = -2.2661
H0: diff = 0 Degrees of freedom = 154

Ha: diff < 0 Ha: diff != 0 Ha: diff > 0
Pr(T < t) = 0.0124 Pr(|T| > |t|) = 0.0248 Pr(T > t) = 0.9876

Source	SS	df	MS	Number of obs	=	1,566
Model	5799.7817	10	579.97817	F(10, 1555)	=	9.87
Residual	91375.8049	1,555	58.7625755	Prob > F	=	0.0000
Total	97175.5866	1,565	62.0930266	R-squared	=	0.0597
				Adj R-squared	=	0.0536
				Root MSE	=	7.6657

wt82_71	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
qsmk	3.356927	.4580399	7.33	0.000	2.458486	4.255368
ps_dec						
1	4.384269	.8873947	4.94	0.000	2.643652	6.124885
2	3.903694	.8805212	4.43	0.000	2.17656	5.630828
3	4.36015	.8793345	4.96	0.000	2.635343	6.084956
4	4.010061	.8745966	4.59	0.000	2.294548	5.725575
5	2.342505	.8754878	2.68	0.008	.6252438	4.059766
6	3.572955	.8714389	4.10	0.000	1.863636	5.282275
7	2.30881	.8727462	2.65	0.008	.5969261	4.020693
8	1.516677	.8715796	1.74	0.082	-.1929182	3.226273
9	-.0439923	.8684465	-0.05	0.960	-1.747442	1.659457
_cons	-.8625798	.6530529	-1.32	0.187	-2.143537	.4183773

Program 15.4

- Standardization and outcome regression using the propensity score
- Data from NHEFS

- Section 15.3

```

use ./data/nhefs-formatted, clear

/*Estimate the propensity score*/
logit qsmk sex race c.age##c.age ib(last).education ///
    c.smokeintensity##c.smokeintensity ///
    c.smokeyrs##c.smokeyrs ib(last).exercise ///
    ib(last).active c.wt71##c.wt71
predict ps, pr

/*Expand the dataset for standardization*/
expand 2, generate(interv)
expand 2 if interv == 0, generate(interv2)
replace interv = -1 if interv2 == 1
drop interv2
tab interv
replace wt82_71 = . if interv != -1
replace qsmk = 0 if interv == 0
replace qsmk = 1 if interv == 1
by interv, sort: summarize qsmk

/*Regression on the propensity score, allowing for effect modification*/
regress wt82_71 qsmk##c.ps
predict predY, xb
by interv, sort: summarize predY

quietly summarize predY if(interv == -1)
matrix input observe = (-1,`r(mean)')
quietly summarize predY if(interv == 0)
matrix observe = (observe \0,`r(mean)')
quietly summarize predY if(interv == 1)
matrix observe = (observe \1,`r(mean)')
matrix observe = (observe \., observe[3,2]-observe[2,2])
matrix rownames observe = observed E(Y(a=0)) E(Y(a=1)) difference
matrix colnames observe = interv value
matrix list observe

/*bootstrap program*/
drop if interv != -1
gen meanY_b =.
qui save ./data/nhefs_std, replace

capture program drop bootstdz

program define bootstdz, rclass
use ./data/nhefs_std, clear
preserve
bsample
/*Create 2 new copies of the data.
Set the outcome AND the exposure to missing in the copies*/
expand 2, generate(interv_b)
expand 2 if interv_b == 0, generate(interv2_b)

```

```

qui replace interv_b = -1 if interv2_b ==1
qui drop interv2_b
qui replace wt82_71 = . if interv_b != -1
qui replace qsmk = . if interv_b != -1

/*Fit the propensity score in the original data
(where qsmk is not missing) and generate predictions for everyone*/
logit qsmk sex race c.age##c.age ib(last).education ///
    c.smokeintensity##c.smokeintensity ///
    c.smokeyrs##c.smokeyrs ib(last).exercise ib(last).active ///
    c.wt71##c.wt71
predict ps_b, pr

/*Set the exposure to 0 for everyone in copy 0,
and 1 to everyone for copy 1*/
qui replace qsmk = 0 if interv_b == 0
qui replace qsmk = 1 if interv_b == 1

/*Fit the outcome regression in the original data
(where wt82_71 is not missing) and
generate predictions for everyone*/
regress wt82_71 qsmk##c.ps
predict predY_b, xb

/*Summarize the predictions in each set of copies*/
summarize predY_b if interv_b == 0
return scalar boot_0 = r(mean)
summarize predY_b if interv_b == 1
return scalar boot_1 = r(mean)
return scalar boot_diff = return(boot_1) - return(boot_0)
qui drop meanY_b
restore
end

/*Then we use the `simulate` command to run the bootstraps
as many times as we want.
Start with reps(10) to make sure your code runs,
and then change to reps(1000) to generate your final CIs*/
simulate EY_a0=r(boot_0) EY_a1 = r(boot_1) ///
    difference = r(boot_diff), reps(500) seed(1): bootstdz

matrix pe = observe[2..4, 2]'
matrix list pe
bstat, stat(pe) n(1629)
estat bootstrap, p

```

```

Iteration 0:  log likelihood = -893.02712
Iteration 1:  log likelihood = -839.70016
Iteration 2:  log likelihood = -838.45045
Iteration 3:  log likelihood = -838.44842
Iteration 4:  log likelihood = -838.44842

```

Logistic regression

Number of obs = 1,566

Log likelihood = -838.44842

LR chi2(18) = 109.16
 Prob > chi2 = 0.0000
 Pseudo R2 = 0.0611

qsmk	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
sex	-.5274782	.1540497	-3.42	0.001	-.82941	-.2255463
race	-.8392636	.2100668	-4.00	0.000	-1.250987	-.4275404
age	.1212052	.0512663	2.36	0.018	.0207251	.2216853
c.age#c.age	-.0008246	.0005361	-1.54	0.124	-.0018753	.0002262
education						
1	-.4759606	.2262238	-2.10	0.035	-.9193511	-.0325701
2	-.5047361	.217597	-2.32	0.020	-.9312184	-.0782538
3	-.3895288	.1914353	-2.03	0.042	-.7647351	-.0143226
4	-.4123596	.2772868	-1.49	0.137	-.9558318	.1311126
smokeintensity	-.0772704	.0152499	-5.07	0.000	-.1071596	-.0473812
c.smokeintensity#						
c.smokeintensity	.0010451	.0002866	3.65	0.000	.0004835	.0016068
smokeyrs	-.0735966	.0277775	-2.65	0.008	-.1280395	-.0191538
c.smokeyrs#						
c.smokeyrs	.0008441	.0004632	1.82	0.068	-.0000637	.0017519
exercise						
0	-.395704	.1872401	-2.11	0.035	-.7626878	-.0287201
1	-.0408635	.1382674	-0.30	0.768	-.3118627	.2301357
active						
0	-.176784	.2149721	-0.82	0.411	-.5981215	.2445535
1	-.1448395	.2111472	-0.69	0.493	-.5586806	.2690015
wt71	-.0152357	.0263161	-0.58	0.563	-.0668144	.036343
c.wt71#c.wt71	.0001352	.0001632	0.83	0.407	-.0001846	.000455
_cons	-1.19407	1.398493	-0.85	0.393	-3.935066	1.546925

(1,566 observations created)

(1,566 observations created)

(1,566 real changes made)

Expanded observation type	Freq.	Percent	Cum.
-1	1,566	33.33	33.33
Original observation	1,566	33.33	66.67
Duplicated observation	1,566	33.33	100.00
Total	4,698	100.00	

(3,132 real changes made, 3,132 to missing)

(403 real changes made)

(1,163 real changes made)

-> interv = -1

Variable	Obs	Mean	Std. dev.	Min	Max
qsmk	1,566	.2573436	.4373099	0	1

-> interv = Original

Variable	Obs	Mean	Std. dev.	Min	Max
qsmk	1,566	0	0	0	0

-> interv = Duplicat

Variable	Obs	Mean	Std. dev.	Min	Max
qsmk	1,566	1	0	1	1

Source	SS	df	MS	Number of obs	=	1,566
Model	5287.31428	3	1762.43809	F(3, 1562)	=	29.96
Residual	91888.2723	1,562	58.827319	Prob > F	=	0.0000
Total	97175.5866	1,565	62.0930266	R-squared	=	0.0544
				Adj R-squared	=	0.0526
				Root MSE	=	7.6699

wt82_71	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
qsmk						
Smoking cessation	4.036457	1.13904	3.54	0.000	1.80225	6.270665
ps	-12.3319	2.129602	-5.79	0.000	-16.50908	-8.154716
qsmk#c.ps						
Smoking cessation	-2.038829	3.649684	-0.56	0.576	-9.197625	5.119967

_cons		4.935432	.5570216	8.86	0.000	3.842843 6.028021

-> interv = -1

Variable		Obs	Mean	Std. dev.	Min	Max
predY		1,566	2.6383	1.838063	-3.4687	8.111371

-> interv = Original

Variable		Obs	Mean	Std. dev.	Min	Max
predY		1,566	1.761898	1.433264	-4.645079	4.306496

-> interv = Duplicat

Variable		Obs	Mean	Std. dev.	Min	Max
predY		1,566	5.273676	1.670225	-2.192565	8.238971

observe[4,2]

	interv	value
observed	-1	2.6382998
E(Y(a=0))	0	1.7618979
E(Y(a=1))	1	5.2736757
difference	.	3.5117778

(3,132 observations deleted)

(1,566 missing values generated)

11. predict ps_b, pr
12.


```

Command: bootstdz
      EY_a0: r(boot_0)
      EY_a1: r(boot_1)
difference: r(boot_diff)

```

Simulations (500)

```

-----+----- 1 -----+----- 2 -----+----- 3 -----+----- 4 -----+----- 5
..... 50
..... 100
..... 150
..... 200
..... 250
..... 300
..... 350
..... 400
..... 450
..... 500

```

pe[1,3]

```

      E(Y(a=0))  E(Y(a=1))  difference
value  1.7618979  5.2736757  3.5117778

```

Bootstrap results

Number of obs = 1,629
Replications = 500

	Observed coefficient	Bootstrap std. err.	z	P> z	Normal-based [95% conf. interval]	
EY_a0	1.761898	.2255637	7.81	0.000	1.319801	2.203995
EY_a1	5.273676	.4695378	11.23	0.000	4.353399	6.193953
difference	3.511778	.4970789	7.06	0.000	2.537521	4.486035

Bootstrap results

Number of obs = 1,629
Replications = 500

	Observed coefficient	Bias	Bootstrap std. err.	[95% conf. interval]		
EY_a0	1.7618979	.0026735	.22556365	1.269908	2.186845	(P)
EY_a1	5.2736757	-.0049491	.46953779	4.34944	6.109205	(P)
difference	3.5117778	-.0076226	.49707894	2.466025	4.424034	(P)

Key: P: Percentile

16. Instrumental variables estimation: Stata

```
library(Statamarkdown)
```

```
/******  
Stata code for Causal Inference: What If by Miguel Hernan & Jamie Robins  
Date: 10/10/2019  
Author: Eleanor Murray  
For errors contact: ejmurray@bu.edu  
******/
```

Program 16.1

- Estimating the average causal effect using the standard IV estimator via the calculation of sample averages
- Data from NHEFS
- Section 16.2

```
use ./data/nhefs-formatted, clear  
  
summarize price82  
  
/* ignore subjects with missing outcome or missing instrument for simplicity*/  
foreach var of varlist wt82 price82 {  
    drop if `var'==.  
}  
  
/*Create categorical instrument*/  
gen byte highprice = (price82 > 1.5 & price82 < .)  
  
save ./data/nhefs-highprice, replace  
  
/*Calculate P[Z|A=a]*/  
tab highprice qsmk, row  
  
/*Calculate P[Y|Z=z]*/  
ttest wt82_71, by(highprice)  
  
/*Final IV estimate, OPTION 1: Hand calculations*/  
/*Numerator: num = E[Y|Z=1] - E[Y|Z=0] = 2.686 - 2.536 = 0.150*/  
/*Denominator: denom = P[A=1|Z=1] - P[A=1|Z=0] = 0.258 - 0.195 = 0.063 */
```

```

/*IV estimator: E[Ya=1] - E[Ya=0] =
(E[Y|Z=1]-E[Y|Z=0])/(P[A=1|Z=1]-P[A=1|Z=0]) = 0.150/0.063 = 2.397*/
display "Numerator, E[Y|Z=1] - E[Y|Z=0] =", 2.686 - 2.536
display "Denominator: denom = P[A=1|Z=1] - P[A=1|Z=0] =", 0.258 - 0.195
display "IV estimator =", 0.150/0.063

/*OPTION 2 2: automated calculation of instrument*/
/*Calculate P[A=1|Z=z], for each value of the instrument,
and store in a matrix*/
quietly summarize qsmk if (highprice==0)
matrix input pa = (`r(mean)')
quietly summarize qsmk if (highprice==1)
matrix pa = (pa ,`r(mean)')
matrix list pa

/*Calculate P[Y|Z=z], for each value of the instrument,
and store in a second matrix*/
quietly summarize wt82_71 if (highprice==0)
matrix input ey = (`r(mean)')
quietly summarize wt82_71 if (highprice==1)
matrix ey = (ey ,`r(mean)')
matrix list ey

/*Using Stata's built-in matrix manipulation feature (Mata),
calculate numerator, denominator and IV estimator*/
*Numerator: num = E[Y|Z=1] - E[Y|Z=0]*mata
*Denominator: denom = P[A=1|Z=1] - P[A=1|Z=0]*
*IV estimator: iv_est = IV estimate of E[Ya=1] - E[Ya=0] *
mata
pa = st_matrix("pa")
ey = st_matrix("ey")
num = ey[1,2] - ey[1,1]
denom = pa[1,2] - pa[1,1]
iv_est = num / denom
num
denom
st_numscalar("iv_est", iv_est)
end
di scalar(iv_est)

```

Variable	Obs	Mean	Std. dev.	Min	Max
price82	1,476	1.805989	.1301703	1.451904	2.103027

(0 observations deleted)
(90 observations deleted)

file ./data/nhefs-highprice.dta saved

```

+-----+
| Key      |

```

```
|-----|
| frequency |
| row percentage |
+-----+
```

quit smoking between baseline and 1982			
highprice	No smokin	Smoking c	Total
0	33	8	41
	80.49	19.51	100.00
1	1,065	370	1,435
	74.22	25.78	100.00
Total	1,098	378	1,476
	74.39	25.61	100.00

Two-sample t test with equal variances

Group	Obs	Mean	Std. err.	Std. dev.	[95% conf. interval]	
0	41	2.535729	1.461629	9.358993	-.4183336	5.489792
1	1,435	2.686018	.2084888	7.897848	2.277042	3.094994
Combined	1,476	2.681843	.2066282	7.938395	2.276527	3.087159
diff		-.1502887	1.257776		-2.617509	2.316932

```
diff = mean(0) - mean(1)                                t = -0.1195
H0: diff = 0                                             Degrees of freedom = 1474
```

```
Ha: diff < 0                Ha: diff != 0                Ha: diff > 0
Pr(T < t) = 0.4525          Pr(|T| > |t|) = 0.9049          Pr(T > t) = 0.5475
```

Numerator, $E[Y|Z=1] - E[Y|Z=0] = .15$

Denominator: $\text{denom} = P[A=1|Z=1] - P[A=1|Z=0] = .063$

IV estimator = 2.3809524

```
pa[1,2]
      c1      c2
r1 .19512195 .25783972
```

```

ey[1,2]
           c1      c2
r1    2.535729  2.6860178

----- mata (type end to exit) -----
: pa = st_matrix("pa")

: ey = st_matrix("ey")

: num = ey[1,2] - ey[1,1]

: denom = pa[1,2] - pa[1,1]

: iv_est = num / denom

: num
    .1502887173

: denom
    .06271777

: st_numscalar("iv_est", iv_est)

: end
-----

2.3962701

```

Program 16.2

- Estimating the average causal effect using the standard IV estimator via two-stage-least-squares regression
- Data from NHEFS
- Section 16.2

```

use ./data/nhefs-highprice, clear

/* ivregress fits the model in two stages:
- first model: qsmk = highprice
- second model: wt82_71 = predicted_qsmk */
ivregress 2sls wt82_71 (qsmk = highprice)

```

Instrumental variables 2SLS regression	Number of obs	=	1,476
	Wald chi2(1)	=	0.01
	Prob > chi2	=	0.9038
	R-squared	=	0.0213
	Root MSE	=	7.8508

```

-----
      wt82_71 | Coefficient  Std. err.      z    P>|z|    [95% conf. interval]

```

qsmk		2.39627	19.82659	0.12	0.904	-36.46313	41.25567
_cons		2.068164	5.081652	0.41	0.684	-7.89169	12.02802

Instrumented: qsmk
Instruments: highprice

Program 16.3

- Estimating the average causal effect using the standard IV estimator via an additive marginal structural model
- Data from NHEFS
- Checking one possible value of psi.
- See Chapter 14 for program that checks several values and computes 95% confidence intervals
- Section 16.2

```
use ./data/nhefs-highprice, clear

gen psi = 2.396
gen hspi = wt82_71 - psi*qsmk

logit highprice hspi
```

Iteration 0: log likelihood = -187.34948
Iteration 1: log likelihood = -187.34948

Logistic regression	Number of obs = 1,476
	LR chi2(1) = 0.00
	Prob > chi2 = 1.0000
Log likelihood = -187.34948	Pseudo R2 = 0.0000

highprice		Coefficient	Std. err.	z	P> z	[95% conf. interval]
hspi		2.75e-07	.0201749	0.00	1.000	-.0395419 .0395424
_cons		3.555347	.1637931	21.71	0.000	3.234319 3.876376

Program 16.4

- Estimating the average causal effect using the standard IV estimator based on alternative proposed instruments
- Data from NHEFS
- Section 16.5

```
use ./data/nhefs-highprice, clear

/*Instrument cut-point: 1.6*/
replace highprice = .
replace highprice = (price82 >1.6 & price82 < .)

ivregress 2sls wt82_71 (qsmk = highprice)
```

```

/*Instrument cut-point: 1.7*/
replace highprice = .
replace highprice = (price82 >1.7 & price82 < .)

ivregress 2sls wt82_71 (qsmk = highprice)

/*Instrument cut-point: 1.8*/
replace highprice = .
replace highprice = (price82 >1.8 & price82 < .)

ivregress 2sls wt82_71 (qsmk = highprice)

/*Instrument cut-point: 1.9*/
replace highprice = .
replace highprice = (price82 >1.9 & price82 < .)

ivregress 2sls wt82_71 (qsmk = highprice)

```

(1,476 real changes made, 1,476 to missing)

(1,476 real changes made)

Instrumental variables 2SLS regression	Number of obs	=	1,476
	Wald chi2(1)	=	0.06
	Prob > chi2	=	0.8023
	R-squared	=	.
	Root MSE	=	18.593

wt82_71	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
qsmk	41.28124	164.8417	0.25	0.802	-281.8026	364.365
_cons	-7.890182	42.21833	-0.19	0.852	-90.63659	74.85623

Instrumented: qsmk
Instruments: highprice

(1,476 real changes made, 1,476 to missing)

(1,476 real changes made)

Instrumental variables 2SLS regression	Number of obs	=	1,476
	Wald chi2(1)	=	0.05
	Prob > chi2	=	0.8274
	R-squared	=	.
	Root MSE	=	20.577

wt82_71	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
---------	-------------	-----------	---	------	----------------------	--

qsmk		-40.91185	187.6162	-0.22	0.827	-408.6328	326.8091
_cons		13.15927	48.05103	0.27	0.784	-81.01901	107.3375

Instrumented: qsmk

Instruments: highprice

(1,476 real changes made, 1,476 to missing)

(1,476 real changes made)

Instrumental variables 2SLS regression	Number of obs	=	1,476
	Wald chi2(1)	=	0.55
	Prob > chi2	=	0.4576
	R-squared	=	.
	Root MSE	=	13.01

wt82_71		Coefficient	Std. err.	z	P> z	[95% conf. interval]
qsmk		-21.10342	28.40885	-0.74	0.458	-76.78374 34.57691
_cons		8.086377	7.283314	1.11	0.267	-6.188657 22.36141

Instrumented: qsmk

Instruments: highprice

(1,476 real changes made, 1,476 to missing)

(1,476 real changes made)

Instrumental variables 2SLS regression	Number of obs	=	1,476
	Wald chi2(1)	=	0.29
	Prob > chi2	=	0.5880
	R-squared	=	.
	Root MSE	=	10.357

wt82_71		Coefficient	Std. err.	z	P> z	[95% conf. interval]
qsmk		-12.81141	23.65099	-0.54	0.588	-59.16649 33.54368
_cons		5.962813	6.062956	0.98	0.325	-5.920362 17.84599

Instrumented: qsmk

Instruments: highprice

Program 16.5

- Estimating the average causal effect using the standard IV estimator conditional on baseline covariates
- Data from NHEFS
- Section 16.5

```

use ./data/nhefs-highprice, clear

replace highprice = .
replace highprice = (price82 >1.5 & price82 < .)

ivregress 2sls wt82_71 sex race c.age c.smokeintensity ///
    c.smokeyrs i.exercise i.active c.wt71 ///
    (qsmk = highprice)

```

(1,476 real changes made, 1,476 to missing)

(1,476 real changes made)

Instrumental variables 2SLS regression	Number of obs	=	1,476
	Wald chi2(11)	=	135.18
	Prob > chi2	=	0.0000
	R-squared	=	0.0622
	Root MSE	=	7.6848

	wt82_71	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
qsmk		-1.042295	29.86522	-0.03	0.972	-59.57705	57.49246
sex		-1.644393	2.620115	-0.63	0.530	-6.779724	3.490938
race		-.1832546	4.631443	-0.04	0.968	-9.260716	8.894207
age		-.16364	.2395678	-0.68	0.495	-.6331844	.3059043
smokeintensity		.0057669	.144911	0.04	0.968	-.2782534	.2897872
smokeyrs		.0258357	.1607639	0.16	0.872	-.2892558	.3409271
exercise							
1		.4987479	2.162395	0.23	0.818	-3.739469	4.736964
2		.5818337	2.174255	0.27	0.789	-3.679628	4.843296
active							
1		-1.170145	.6049921	-1.93	0.053	-2.355908	.0156176
2		-.5122842	1.303121	-0.39	0.694	-3.066355	2.041787
wt71		-.0979493	.036123	-2.71	0.007	-.168749	-.0271496
_cons		17.28033	2.32589	7.43	0.000	12.72167	21.83899

Instrumented: qsmk

Instruments: sex race age smokeintensity smokeyrs 1.exercise 2.exercise
1.active 2.active wt71 highprice

17. Causal survival analysis: Stata

```
library(Statamarkdown)
```

```
/******  
Stata code for Causal Inference: What If by Miguel Hernan & Jamie Robins  
Date: 10/10/2019  
Author: Eleanor Murray  
For errors contact: ejmurray@bu.edu  
******/
```

Program 17.1

- Nonparametric estimation of survival curves
- Data from NHEFS
- Section 17.1

```
use ./data/nhefs-formatted, clear  
  
/*Some preprocessing of the data*/  
gen survtime = .  
replace survtime = 120 if death == 0  
replace survtime = (yrnth - 83)*12 + month if death ==1  
* yrnth ranges from 83 to 92*  
  
tab death qsmk  
  
/*Kaplan-Meier graph of observed survival over time, by quitting smoking*/  
*For now, we use the stset function in Stata*  
stset survtime, failure(death=1)  
sts graph, by(qsmk) xlabel(0(12)120)  
qui gr export ./figs/stata-fig-17-1.png, replace
```

(1,566 missing values generated)

(1,275 real changes made)

(291 real changes made)

death	
between quit smoking between	
1983 and baseline and 1982	
1992 No smokin Smoking c	Total
-----+-----+-----	

0	963	312	1,275
1	200	91	291
<hr/>			
Total	1,163	403	1,566

Survival-time data settings

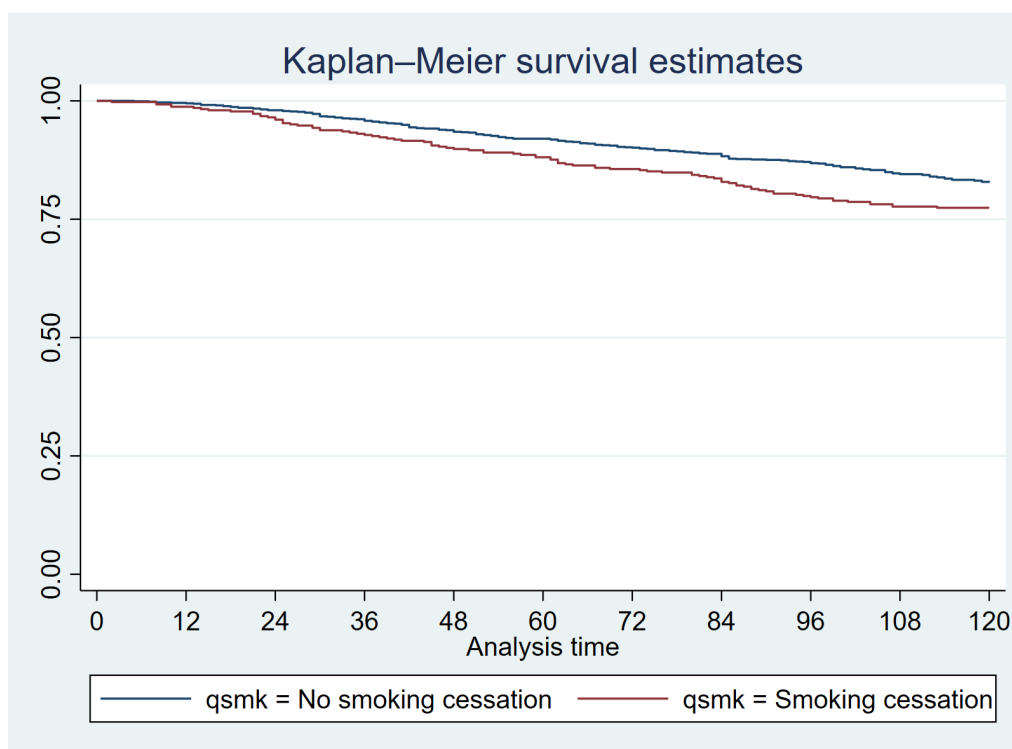
Failure event: death==1
Observed time interval: (0, survtime]
Exit on or before: failure

1,566 total observations
0 exclusions

1,566 observations remaining, representing
291 failures in single-record/single-failure data
171,076 total analysis time at risk and under observation

At risk from t = 0
Earliest observed entry t = 0
Last observed exit t = 120

Failure _d: death==1
Analysis time _t: survtime



Program 17.2

- Parametric estimation of survival curves via hazards model
- Data from NHEFS
- Section 17.1

- Generates Figure 17.4

```

/**Create person-month dataset for survival analyses**/

/* We want our new dataset to include 1 observation per person
per month alive, starting at time = 0.
Individuals who survive to the end of follow-up will have
119 time points
Individuals who die will have survtime - 1 time points*/

use ./data/nhefs-formatted, clear

gen survtime = .
replace survtime = 120 if death == 0
replace survtime = (yrdth - 83)*12 + modth if death ==1

*expand data to person-time*
gen time = 0
expand survtime if time == 0
bysort seqn: replace time = _n - 1

*Create event variable*
gen event = 0
replace event = 1 if time == survtime - 1 & death == 1
tab event

*Create time-squared variable for analyses*
gen timesq = time*time

*Save the dataset to your working directory for future use*
qui save ./data/nhefs_surv, replace

/**Hazard ratios**/
use ./data/nhefs_surv, clear

*Fit a pooled logistic hazards model *
logistic event qsmk qsmk#c.time qsmk#c.time#c.time ///
    c.time c.time#c.time

/**Survival curves: run regression then do:**/

*Create a dataset with all time points under each treatment level*
*Re-expand data with rows for all timepoints*
drop if time != 0
expand 120 if time ==0
bysort seqn: replace time = _n - 1

/*Create 2 copies of each subject, and set outcome to missing
and treatment -- use only the newobs*/
expand 2 , generate(interv)
replace qsmk = interv

/*Generate predicted event and survival probabilities

```

```

for each person each month in copies*/
predict pevent_k, pr
gen psurv_k = 1-pevent_k
keep seqn time qsmk interv psurv_k

*Within copies, generate predicted survival over time*
*Remember, survival is the product of conditional survival probabilities in each interval*
sort seqn interv time
gen _t = time + 1
gen psurv = psurv_k if _t ==1
bysort seqn interv: replace psurv = psurv_k*psurv[_t-1] if _t >1

*Display 10-year standardized survival, under interventions*
*Note: since time starts at 0, month 119 is 10-year survival*
by interv, sort: summarize psurv if time == 119

*Graph of standardized survival over time, under interventions*
/*Note, we want our graph to start at 100% survival,
so add an extra time point with P(surv) = 1*/
expand 2 if time ==0, generate(newtime)
replace psurv = 1 if newtime == 1
gen time2 = 0 if newtime ==1
replace time2 = time + 1 if newtime == 0

/*Separate the survival probabilities to allow plotting by
intervention on qsmk*/
separate psurv, by(interv)

*Plot the curves*
twoway (line psurv0 time2, sort) ///
      (line psurv1 time2, sort) if interv > -1 ///
      , ylabel(0.5(0.1)1.0) xlabel(0(12)120) ///
      ytitle("Survival probability") xtitle("Months of follow-up") ///
      legend(label(1 "A=0") label(2 "A=1"))
qui gr export ./figs/stata-fig-17-2.png, replace

```

(1,566 missing values generated)

(1,275 real changes made)

(291 real changes made)

(169,510 observations created)

(169510 real changes made)

(291 real changes made)

event	Freq.	Percent	Cum.
-----+-----			
0	170,785	99.83	99.83

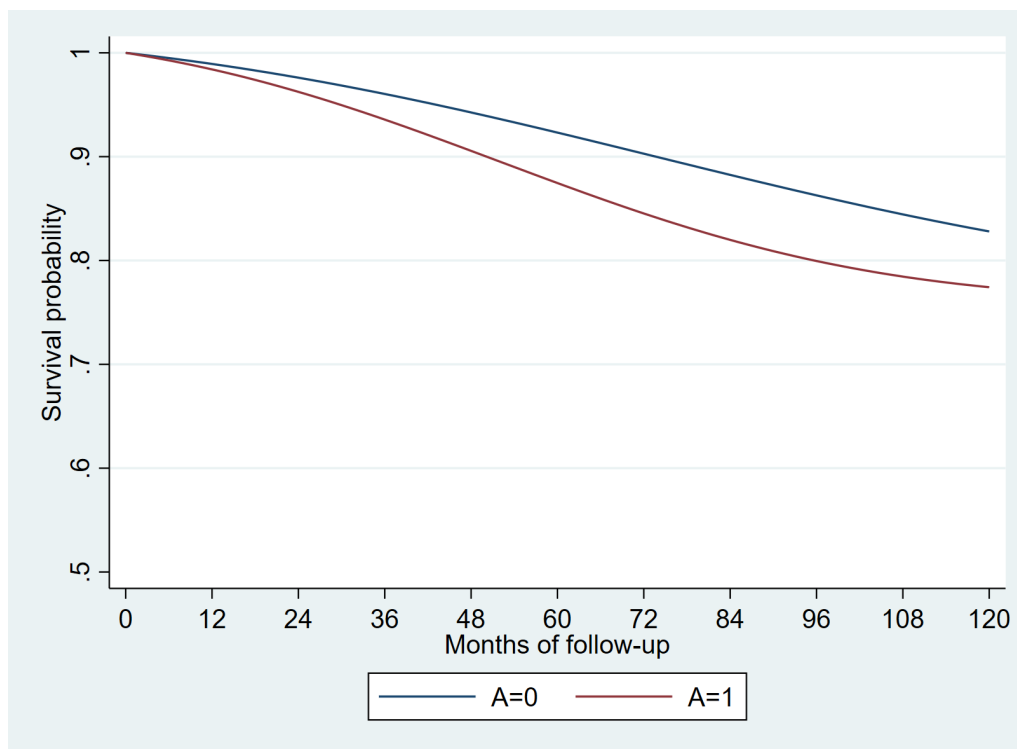
1		291	0.17	100.00
-----+-----				
Total		171,076	100.00	

```

file ./data/nhefs_surv.dta cannot be modified or erased; likely cause is read-only
    directory or file
r(608);

end of do-file
r(608);

```



Program 17.3

- Estimation of survival curves via IP weighted hazards model
- Data from NHEFS
- Section 17.4
- Generates Figure 17.6

```

use ./data/nhefs_surv, clear

keep seqn event qsmk time sex race age education ///
    smokeintensity smkintensity82_71 smokeyrs ///
    exercise active wt71
preserve

*Estimate weights*
logit qsmk sex race c.age##c.age ib(last).education ///
    c.smokeintensity##c.smokeintensity ///
    c.smokeyrs##c.smokeyrs ib(last).exercise ///
    ib(last).active c.wt71##c.wt71 if time == 0

```

```

predict p_qsmk, pr

logit qsmk if time ==0
predict num, pr
gen sw=num/p_qsmk if qsmk==1
replace sw=(1-num)/(1-p_qsmk) if qsmk==0
summarize sw

*IP weighted survival by smoking cessation*
logit event qsmk qsmk#c.time qsmk#c.time#c.time ///
    c.time c.time#c.time [pweight=sw] , cluster(seqn)

*Create a dataset with all time points under each treatment level*
*Re-expand data with rows for all timepoints*
drop if time != 0
expand 120 if time ==0
bysort seqn: replace time = _n - 1

/*Create 2 copies of each subject, and set outcome
to missing and treatment -- use only the newobs*/
expand 2 , generate(interv)
replace qsmk = interv

/*Generate predicted event and survival probabilities
for each person each month in copies*/
predict pevent_k, pr
gen psurv_k = 1-pevent_k
keep seqn time qsmk interv psurv_k

*Within copies, generate predicted survival over time*
/*Remember, survival is the product of conditional survival
probabilities in each interval*/
sort seqn interv time
gen _t = time + 1
gen psurv = psurv_k if _t ==1
bysort seqn interv: replace psurv = psurv_k*psurv[_t-1] if _t >1

*Display 10-year standardized survival, under interventions*
*Note: since time starts at 0, month 119 is 10-year survival*
by interv, sort: summarize psurv if time == 119

quietly summarize psurv if(interv==0 & time ==119)
matrix input observe = (0,`r(mean)')
quietly summarize psurv if(interv==1 & time ==119)
matrix observe = (observe \1,`r(mean)')
matrix observe = (observe \3, observe[2,2]-observe[1,2])
matrix list observe

*Graph of standardized survival over time, under interventions*
/*Note: since our outcome model has no covariates,
we can plot psurv directly.
If we had covariates we would need to stratify or average across the values*/

```



```

expand 2 if time ==0, generate(newtime)
replace psurv = 1 if newtime == 1
gen time2 = 0 if newtime ==1
replace time2 = time + 1 if newtime == 0
separate psurv, by(interv)
twoway (line psurv0 time2, sort) ///
      (line psurv1 time2, sort) if interv > -1 ///
      , ylabel(0.5(0.1)1.0) xlabel(0(12)120) ///
      ytitle("Survival probability") xtitle("Months of follow-up") ///
      legend(label(1 "A=0") label(2 "A=1"))
qui gr export ./figs/stata-fig-17-3.png, replace

*remove extra timepoint*
drop if newtime == 1
drop time2

restore

**Bootstraps**
qui save ./data/nhefs_std1 , replace

capture program drop bootipw_surv

program define bootipw_surv , rclass
use ./data/nhefs_std1 , clear
preserve
bsample, cluster(seqn) idcluster(newseqn)

logit qsmk sex race c.age##c.age ib(last).education ///
      c.smokeintensity##c.smokeintensity ///
      c.smokeyrs##c.smokeyrs ib(last).exercise ib(last).active ///
      c.wt71##c.wt71 if time == 0
predict p_qsmk, pr

logit qsmk if time ==0
predict num, pr

gen sw=num/p_qsmk if qsmk==1
replace sw=(1-num)/(1-p_qsmk) if qsmk==0

logit event qsmk qsmk#c.time qsmk#c.time#c.time ///
      c.time c.time#c.time [pweight=sw], cluster(newseqn)

drop if time != 0
expand 120 if time ==0
bysort newseqn: replace time = _n - 1
expand 2 , generate(interv_b)
replace qsmk = interv_b

predict pevent_k, pr
gen psurv_k = 1-pevent_k
keep newseqn time qsmk interv_b psurv_k

```

```

sort newseqn interv_b time
gen _t = time + 1
gen psurv = psurv_k if _t ==1
bysort newseqn interv_b: ///
    replace psurv = psurv_k*psurv[_t-1] if _t >1
drop if time != 119
bysort interv_b: egen meanS_b = mean(psurv)
keep newseqn qsmk meanS_b
drop if newseqn != 1 /* only need one pair */

drop newseqn

return scalar boot_0 = meanS_b[1]
return scalar boot_1 = meanS_b[2]
return scalar boot_diff = return(boot_1) - return(boot_0)
restore
end

set rmsg on
simulate PrY_a0 = r(boot_0) PrY_a1 = r(boot_1) ///
    difference=r(boot_diff), reps(10) seed(1): bootipw_surv
set rmsg off

matrix pe = observe[1..3, 2]'
bstat, stat(pe) n(1629)

```

```

Iteration 0:  log likelihood = -893.02712
Iteration 1:  log likelihood = -839.70016
Iteration 2:  log likelihood = -838.45045
Iteration 3:  log likelihood = -838.44842
Iteration 4:  log likelihood = -838.44842

```

Logistic regression

```

Number of obs = 1,566
LR chi2(18)    = 109.16
Prob > chi2    = 0.0000
Pseudo R2     = 0.0611

```

Log likelihood = -838.44842

	qsmk	Coefficient	Std. err.	z	P> z	[95% conf. interval]

	sex	-.5274782	.1540497	-3.42	0.001	-.82941 -.2255463
	race	-.8392636	.2100668	-4.00	0.000	-1.250987 -.4275404
	age	.1212052	.0512663	2.36	0.018	.0207251 .2216853
	c.age#c.age	-.0008246	.0005361	-1.54	0.124	-.0018753 .0002262
	education					
	1	-.4759606	.2262238	-2.10	0.035	-.9193511 -.0325701
	2	-.5047361	.217597	-2.32	0.020	-.9312184 -.0782538
	3	-.3895288	.1914353	-2.03	0.042	-.7647351 -.0143226
	4	-.4123596	.2772868	-1.49	0.137	-.9558318 .1311126

smokeintensity		-.0772704	.0152499	-5.07	0.000	-.1071596	-.0473812
c.smokeintensity#							
c.smokeintensity		.0010451	.0002866	3.65	0.000	.0004835	.0016068
smokeyrs		-.0735966	.0277775	-2.65	0.008	-.1280395	-.0191538
c.smokeyrs#							
c.smokeyrs		.0008441	.0004632	1.82	0.068	-.0000637	.0017519
exercise							
0		-.395704	.1872401	-2.11	0.035	-.7626878	-.0287201
1		-.0408635	.1382674	-0.30	0.768	-.3118627	.2301357
active							
0		-.176784	.2149721	-0.82	0.411	-.5981215	.2445535
1		-.1448395	.2111472	-0.69	0.493	-.5586806	.2690015
wt71		-.0152357	.0263161	-0.58	0.563	-.0668144	.036343
c.wt71#c.wt71		.0001352	.0001632	0.83	0.407	-.0001846	.000455
_cons		-1.19407	1.398493	-0.85	0.393	-3.935066	1.546925

Iteration 0: log likelihood = -893.02712
Iteration 1: log likelihood = -893.02712

Logistic regression	Number of obs =	1,566
	LR chi2(0) =	-0.00
	Prob > chi2 =	.
Log likelihood = -893.02712	Pseudo R2 =	-0.0000

qsmk	Coefficient	Std. err.	z	P> z	[95% conf. interval]
_cons	-1.059822	.0578034	-18.33	0.000	-1.173114 - .946529

(128,481 missing values generated)

(128,481 real changes made)

Variable	Obs	Mean	Std. dev.	Min	Max
sw	171,076	1.000509	.2851505	.3312489	4.297662

Iteration 0: log pseudolikelihood = -2136.3671
Iteration 1: log pseudolikelihood = -2127.0974

Iteration 2: log pseudolikelihood = -2126.8556
 Iteration 3: log pseudolikelihood = -2126.8554

Logistic regression

Number of obs = 171,076
 Wald chi2(5) = 22.74
 Prob > chi2 = 0.0004
 Pseudo R2 = 0.0045

Log pseudolikelihood = -2126.8554

(Std. err. adjusted for 1,566 clusters in seqn)

event	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
qsmk	-.1301273	.4186673	-0.31	0.756	-.9507002	.6904456
qsmk#c.time						
Smoking cessation	.01916	.0151318	1.27	0.205	-.0104978	.0488178
qsmk#c.time#c.time						
Smoking cessation	-.0002152	.0001213	-1.77	0.076	-.0004528	.0000225
time	.0208179	.0077769	2.68	0.007	.0055754	.0360604
c.time#c.time	-.0001278	.0000643	-1.99	0.047	-.0002537	-1.84e-06
_cons	-7.038847	.2142855	-32.85	0.000	-7.458839	-6.618855

(169,510 observations deleted)

(186,354 observations created)

(186354 real changes made)

(187,920 observations created)

(187,920 real changes made)

(372,708 missing values generated)

(372708 real changes made)

-> interv = Original

Variable	Obs	Mean	Std. dev.	Min	Max
----------	-----	------	-----------	-----	-----

```

psurv |      1,566      .8161003          0      .8161003      .8161003
-----
-> interv = Duplicat

Variable |      Obs      Mean      Std. dev.      Min      Max
-----+-----
psurv |      1,566      .8116784          0      .8116784      .8116784

observe[3,2]
      c1      c2
r1      0      .8161003
r2      1      .81167841
r3      3      -.00442189

(3,132 observations created)

(3,132 real changes made)

(375,840 missing values generated)

(375,840 real changes made)

Variable      Storage      Display      Value
  name      type      format      label      Variable label
-----
psurv0      float      %9.0g      psurv, interv == Original observation
psurv1      float      %9.0g      psurv, interv == Duplicated observation

(3,132 observations deleted)

5. predict p_qsmk, pr
6.
11.
23. drop if time != 119
24. bysort interv_b: egen meanS_b = mean(psurv)
25. keep newseqn qsmk meanS_b
26. drop if newseqn != 1 /* only need one pair */
27.

```

```
r; t=0.00 22:41:01
```

```
Command: bootipw_surv
PrY_a0: r(boot_0)
PrY_a1: r(boot_1)
difference: r(boot_diff)
```

```
Simulations (10)
```

```
-----+----- 1 -----+----- 2 -----+----- 3 -----+----- 4 -----+----- 5
```

```
.....
```

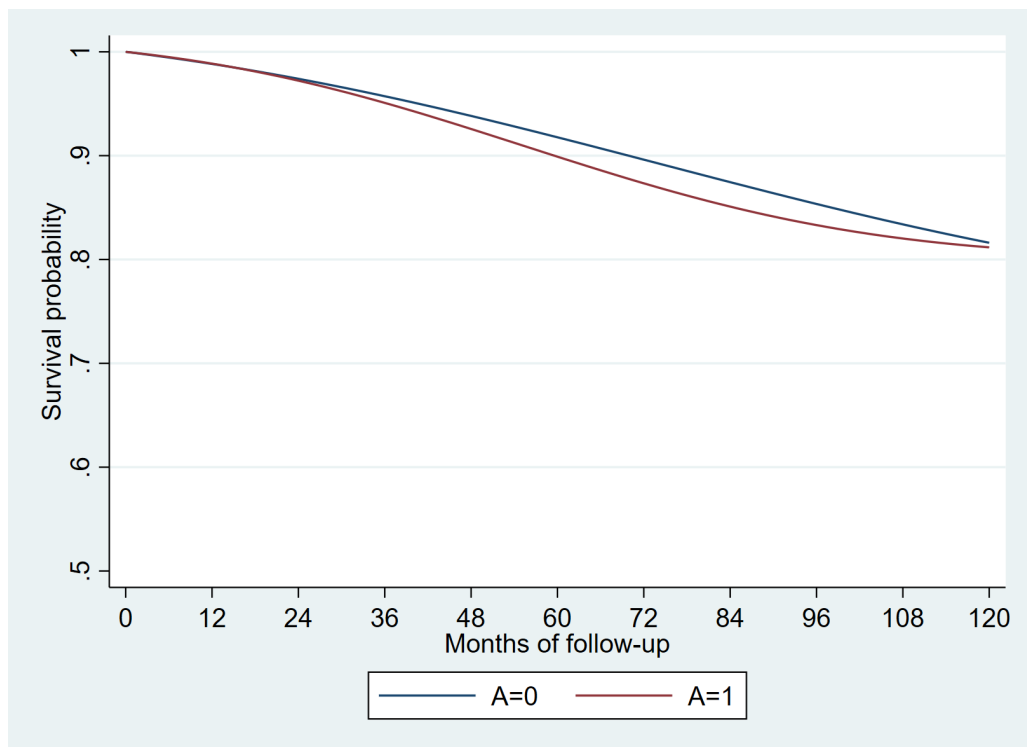
```
r; t=20.32 22:41:21
```

Bootstrap results

Number of obs = 1,629

Replications = 10

	Observed	Bootstrap			Normal-based	
	coefficient	std. err.	z	P> z	[95% conf. interval]	
PrY_a0	.8161003	.0093124	87.64	0.000	.7978484	.8343522
PrY_a1	.8116784	.0237581	34.16	0.000	.7651133	.8582435
difference	-.0044219	.0225007	-0.20	0.844	-.0485224	.0396786



Program 17.4

- Estimating of survival curves via g-formula

- Data from NHEFS
- Section 17.5
- Generates Figure 17.7

```

use ./data/nhefs_surv, clear

keep seqn event qsmk time sex race age education ///
    smokeintensity smkintensity82_71 smokeyrs exercise ///
    active wt71
preserve

quietly logistic event qsmk qsmk#c.time ///
    qsmk#c.time#c.time time c.time#c.time ///
    sex race c.age##c.age ib(last).education ///
    c.smokeintensity##c.smokeintensity ///
    c.smokeyrs##c.smokeyrs ib(last).exercise ib(last).active ///
    c.wt71##c.wt71 , cluster(seqn)

drop if time != 0
expand 120 if time ==0
bysort seqn: replace time = _n - 1
expand 2 , generate(interv)
replace qsmk = interv
predict pevent_k, pr
gen psurv_k = 1-pevent_k
keep seqn time qsmk interv psurv_k
sort seqn interv time
gen _t = time + 1
gen psurv = psurv_k if _t ==1
bysort seqn interv: replace psurv = psurv_k*psurv[_t-1] if _t >1
by interv, sort: summarize psurv if time == 119

keep qsmk interv psurv time

bysort interv : egen meanS = mean(psurv) if time == 119
by interv: summarize meanS

quietly summarize meanS if(qsmk==0 & time ==119)
matrix input observe = ( 0,`r(mean)')
quietly summarize meanS if(qsmk==1 & time ==119)
matrix observe = (observe \1,`r(mean)')
matrix observe = (observe \2, observe[2,2]-observe[1,2])
*Add some row/column descriptions and print results to screen*
matrix rownames observe = P(Y(a=0)=1) P(Y(a=1)=1) difference
matrix colnames observe = interv survival

*Graph standardized survival over time, under interventions*
/*Note: unlike in Program 17.3, we now have covariates
so we first need to average survival across strata*/
bysort interv time : egen meanS_t = mean(psurv)

*Now we can continue with the graph*
expand 2 if time ==0, generate(newtime)

```

```

replace meanS_t = 1 if newtime == 1
gen time2 = 0 if newtime ==1
replace time2 = time + 1 if newtime == 0
separate meanS_t, by(interv)

twoway (line meanS_t0 time2, sort) ///
      (line meanS_t1 time2, sort) ///
      , ylabel(0.5(0.1)1.0) xlabel(0(12)120) ///
      ytitle("Survival probability") xtitle("Months of follow-up") ///
      legend(label(1 "A=0") label(2 "A=1"))
gr export ./figs/stata-fig-17-4.png, replace

*remove extra timepoint*
drop if newtime == 1

restore

*Bootstraps*
qui save ./data/nhefs_std2 , replace

capture program drop bootstdz_surv

program define bootstdz_surv , rclass
use ./data/nhefs_std2 , clear
preserve

bsample, cluster(seqn) idcluster(newseqn)
logistic event qsmk qsmk#c.time qsmk#c.time#c.time ///
      time c.time#c.time ///
      sex race c.age##c.age ib(last).education ///
      c.smokeintensity##c.smokeintensity c.smkintensity82_71 ///
      c.smokeyrs##c.smokeyrs ib(last).exercise ib(last).active ///
      c.wt71##c.wt71
drop if time != 0
/*only predict on new version of data */
expand 120 if time ==0
bysort newseqn: replace time = _n - 1
expand 2 , generate(interv_b)
replace qsmk = interv_b
predict pevent_k, pr
gen psurv_k = 1-pevent_k
keep newseqn time qsmk psurv_k
sort newseqn qsmk time
gen _t = time + 1
gen psurv = psurv_k if _t ==1
bysort newseqn qsmk: replace psurv = psurv_k*psurv[_t-1] if _t >1
drop if time != 119 /* keep only last observation */
keep newseqn qsmk psurv
/* if time is in data for complete graph add time to bysort */
bysort qsmk : egen meanS_b = mean(psurv)
keep newseqn qsmk meanS_b
drop if newseqn != 1 /* only need one pair */

```



```

drop newseqn

return scalar boot_0 = meanS_b[1]
return scalar boot_1 = meanS_b[2]
return scalar boot_diff = return(boot_1) - return(boot_0)
restore
end

set rmsg on
simulate PrY_a0 = r(boot_0) PrY_a1 = r(boot_1) ///
    difference=r(boot_diff), reps(10) seed(1): bootstdz_surv
set rmsg off

matrix pe = observe[1..3, 2]'
bstat, stat(pe) n(1629)

```

(169,510 observations deleted)

(186,354 observations created)

(186354 real changes made)

(187,920 observations created)

(187,920 real changes made)

(372,708 missing values generated)

(372708 real changes made)

-> interv = Original

Variable	Obs	Mean	Std. dev.	Min	Max
psurv	1,566	.8160697	.2014345	.014127	.9903372

-> interv = Duplicat

Variable	Obs	Mean	Std. dev.	Min	Max
psurv	1,566	.811763	.2044758	.0123403	.9900259

(372,708 missing values generated)

```
-----
-> interv = Original
```

Variable	Obs	Mean	Std. dev.	Min	Max
meanS	1,566	.8160697	0	.8160697	.8160697

```
-----
-> interv = Duplicat
```

Variable	Obs	Mean	Std. dev.	Min	Max
meanS	1,566	.8117629	0	.8117629	.8117629

```
(3,132 observations created)
```

```
(3,132 real changes made)
```

```
(375,840 missing values generated)
```

```
(375,840 real changes made)
```

Variable name	Storage type	Display format	Value label	Variable label
meanS_t0	float	%9.0g		meanS_t, interv == Original observation
meanS_t1	float	%9.0g		meanS_t, interv == Duplicated observation

```
file ./figs/stata-fig-17-4.png saved as PNG format
```

```
(3,132 observations deleted)
```

```
5. drop if time != 0
```

```
6. /*only predict on new version of data */
```

```
r; t=0.00 22:41:28
```

```

Command: bootstdz_surv
PrY_a0: r(boot_0)
PrY_a1: r(boot_1)
difference: r(boot_diff)

```

Simulations (10)

```

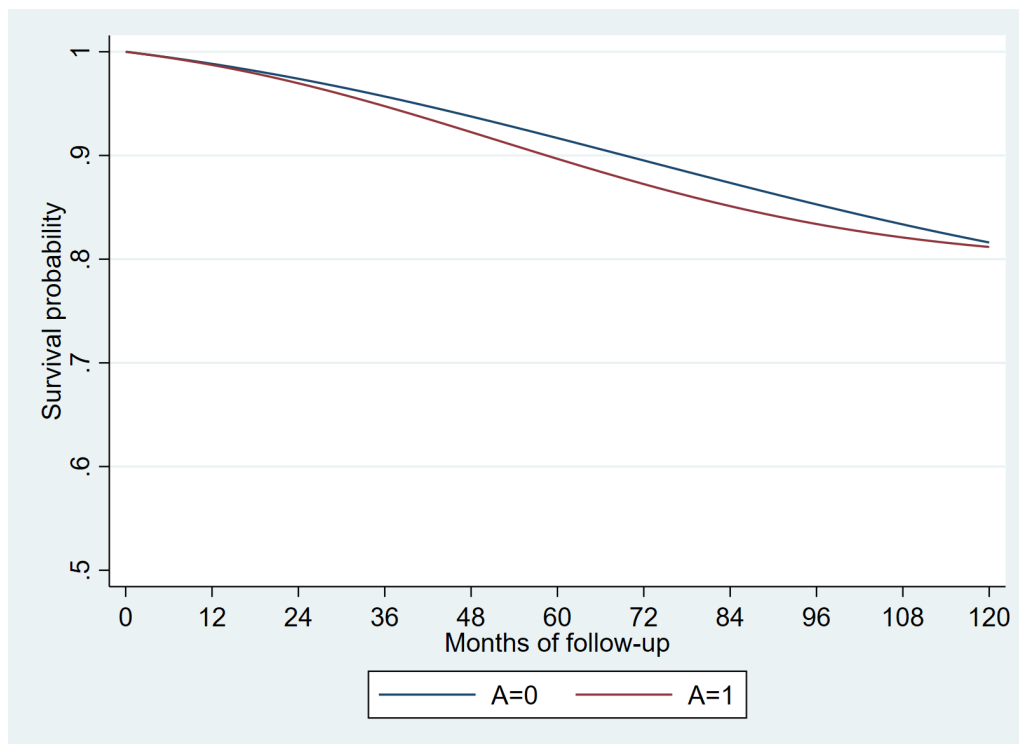
-----+--- 1 -----+--- 2 -----+--- 3 -----+--- 4 -----+--- 5
.....
r; t=21.41 22:41:49

```

Bootstrap results

Number of obs = 1,629
Replications = 10

	Observed	Bootstrap			Normal-based	
	coefficient	std. err.	z	P> z	[95% conf. interval]	
PrY_a0	.8160697	.0087193	93.59	0.000	.7989802	.8331593
PrY_a1	.8117629	.0292177	27.78	0.000	.7544973	.8690286
difference	-.0043068	.0307674	-0.14	0.889	-.0646099	.0559963



Session information: Stata

```
library(Statamarkdown)
```

For reproducibility.

```
about
```

Stata/MP 17.0 for Windows (64-bit x86-64)
Revision 16 May 2023
Copyright 1985-2021 StataCorp LLC

Total physical memory: 32.00 GB
Available physical memory: 9.16 GB

Stata license: Unlimited-user 2-core network, expiring 21 Jan 2024
Serial number: 501709378202
Licensed to: Tom Palmer
University of Bristol

```
# install.packages("sessioninfo")
sessioninfo::session_info()
- Session info -----
setting value
version R version 4.3.0 (2023-04-21 ucrt)
os Windows 11 x64 (build 22621)
system x86_64, mingw32
ui RTerm
language (EN)
collate English_United Kingdom.utf8
ctype English_United Kingdom.utf8
tz Europe/London
date 2023-06-16
pandoc 3.1.1 @ C:/Program Files/RStudio/resources/app/bin/quarto/bin/tools/ (via rmarkdown)

- Packages -----
package * version date (UTC) lib source
bookdown 0.34 2023-05-09 [1] CRAN (R 4.3.0)
cli 3.6.1 2023-03-23 [1] CRAN (R 4.3.0)
digest 0.6.31 2022-12-11 [1] CRAN (R 4.3.0)
evaluate 0.21 2023-05-05 [1] CRAN (R 4.3.0)
fastmap 1.1.1 2023-02-24 [1] CRAN (R 4.3.0)
htmltools 0.5.5 2023-03-23 [1] CRAN (R 4.3.0)
knitr 1.43 2023-05-25 [1] CRAN (R 4.3.0)
rlang 1.1.1 2023-04-28 [1] CRAN (R 4.3.0)
```

```
rmarkdown      2.22    2023-06-01 [1] CRAN (R 4.3.0)
rstudioapi     0.14    2022-08-22 [1] CRAN (R 4.3.0)
sessioninfo    1.2.2    2021-12-06 [1] CRAN (R 4.3.0)
Statamarkdown * 0.8.0    2023-06-01 [1] CRAN (R 4.3.0)
xfun           0.39     2023-04-20 [1] CRAN (R 4.3.0)
yaml           2.3.7    2023-01-23 [1] CRAN (R 4.3.0)
```

```
[1] C:/Users/tom/AppData/Local/R/win-library/4.3
```

```
[2] C:/Program Files/R/R-4.3.0/library
```

Bibliography

Miguel A Hernán and James M Robins. *Causal Inference: What If*. Boca Raton: Chapman & Hall/CRC, 2020.

