

Causal Inference Book: Exercises – R code

*Book by M. A. Hernán and J. M. Robins, R code by Joy Shi and Sean McGrath, R
Markdown code by Tom Palmer*

2019-10-02

Contents

Preface	v
Packages to install	v
Downloading the datasets	v
 R code	 3
11. Why model?	3
Program 11.1	3
Program 11.2	5
Program 11.3	6
 12. IP Weighting and Marginal Structural Models	 9
Program 12.1	9
Program 12.2	12
Program 12.3	15
Program 12.4	18
Program 12.5	19
Program 12.6	21
Program 12.7	24
 13. Standardization and the parametric G-formula	 31
Program 13.1	31
Program 13.2	33
Program 13.3	35
Program 13.4	37
 14. G-estimation of Structural Nested Models	 41
Program 14.1	41
Program 14.2	43
Program 14.3	48
 15. Outcome regression and propensity scores	 51
Program 15.1	51
Program 15.2	57
Program 15.3	61
Program 15.4	67
 16. Instrumental variables estimation	 73

Program 16.1	73
Program 16.2	74
Program 16.3	74
Program 16.4	75
Program 16.5	77
17. Causal survival analysis	79
Program 17.1	79
Program 17.2	80
Program 17.3	82
Program 17.4	85
Program 17.5	88
R session information	93

Preface

This book presents code examples from the Causal Inference Book by Hernán and Robins, 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.

Packages to install

To install the R packages required for this book please copy/fork the repository and run:

```
# install.packages('devtools') # uncomment if devtools not  
# installed  
devtools::install_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()  
dataurls[[1]] <- "https://cdn1.sph.harvard.edu/wp-content/uploads/sites/1268/2012/10/nhefs_sas.zip"  
dataurls[[2]] <- "https://cdn1.sph.harvard.edu/wp-content/uploads/sites/1268/2012/10/nhefs_stata.zip"  
dataurls[[3]] <- "https://cdn1.sph.harvard.edu/wp-content/uploads/sites/1268/2017/01/nhefs_excel.zip"  
dataurls[[4]] <- "https://cdn1.sph.harvard.edu/wp-content/uploads/sites/1268/1268/20/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

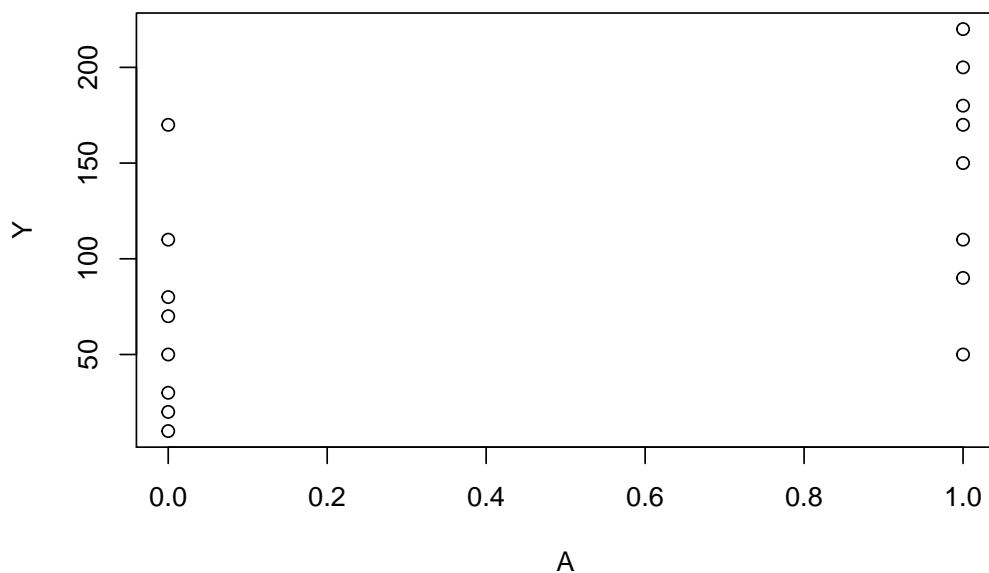
11. Why model?

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, 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)
```



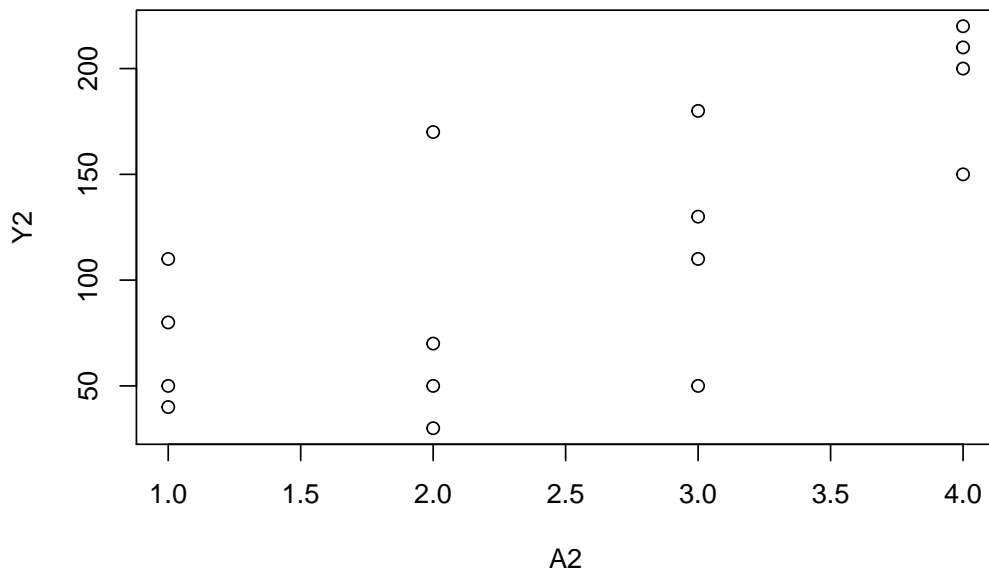
```
summary(Y[A == 0])
```

##	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
```

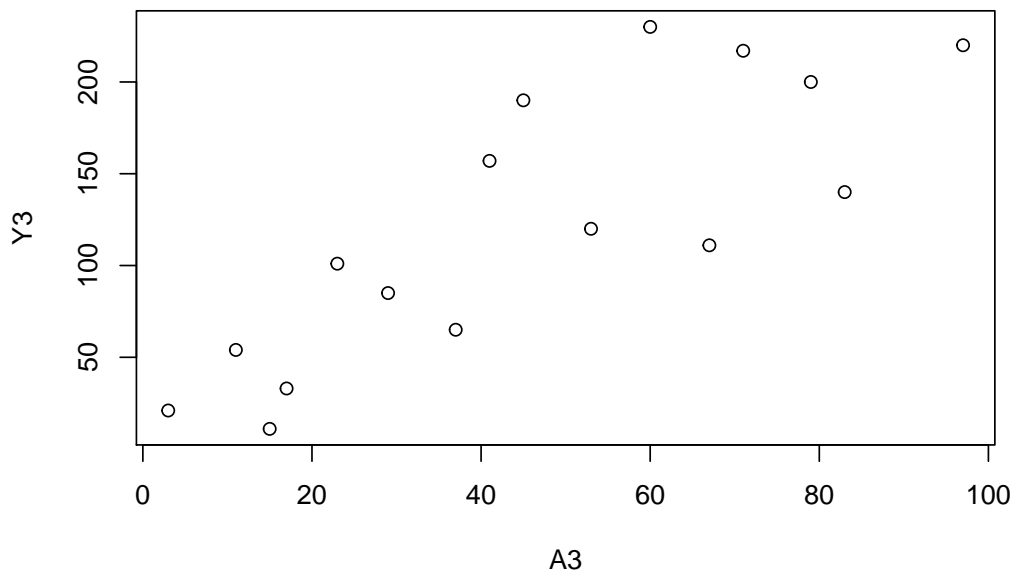
```
summary(Y2[A2 == 4])
```

```
##      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)  
##  
## Deviance Residuals:  
##      Min       1Q   Median       3Q      Max   
## -61.930  -30.564   -5.741   30.653   77.225   
##  
## 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)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -96.250  -40.000   3.125   35.938  102.500
##
## 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)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q        Max
## -65.27  -34.41   13.21   26.11   64.36
##
## 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$qsmk, nehs.nmv$race), 1)

##
##      0      1
##    0 0.85382631 0.14617369
##    1 0.91066998 0.08933002

table(nhefs.nmv$qsmk, nehs.nmv$education)

##
##      1    2    3    4    5
##    0 210 266 480  92 115
##    1  81  74 157  29  62

prop.table(table(nhefs.nmv$qsmk, nehs.nmv$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 = nhefs.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 = nhefs.nmv)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.5127  -0.7907  -0.6387   0.9832   2.3729
##
## 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 = dhefs.nmv,
  weights = w,
  id = seqn,
  corstr = "independence"
)
summary(msm.w)

##
## Call:
## geeglm(formula = wt82_71 ~ qsmk, data = dhefs.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
##
## Estimated Scale Parameters:
##              Estimate Std. err
## (Intercept)   65.06   4.221
##
## Correlation: Structure = independenceNumber 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(dhefs.nmv$age[dhefs.nmv$race == 0 & dhefs.nmv$sex == 1],
      dhefs.nmv$qsmk[dhefs.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)
```

```
##
```

```
## Deviance Residuals:
```

```
##      Min       1Q   Median       3Q      Max
```

```
## -1.513  -0.791  -0.639   0.983   2.373
```

```
##
```

```
## 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 = nhfs.nmv)
summary(numer.fit)

##
## Call:
## glm(formula = qsmk ~ 1, family = binomial(), data = nhfs.nmv)
##
## Deviance Residuals:
##    Min       1Q   Median       3Q      Max
## -0.771  -0.771  -0.771   1.648   1.648
##
## 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.terr 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
##
## Estimated Scale Parameters:
##              Estimate Std.terr
## (Intercept)    60.7    3.71
##
## Correlation: Structure = independenceNumber 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$sw ~ nhefs.nmv$sex + nhefs.nmv$qsmk)

##           nhefs.nmv$qsmk
## nhefs.nmv$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 = nhefs.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 = nhefs.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 = nhefs.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
##
## (Intercept)          ***
## smkintensity82_71      ***
## I(smkintensity82_71 * smkintensity82_71)
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Estimated Scale Parameters:
##              Estimate Std.err
## (Intercept)    60.5      4.5
##
## Correlation: Structure = independenceNumber 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
##
## Estimated Scale Parameters:
##              Estimate Std.err
## (Intercept)         1 0.0678
##
## Correlation: Structure = independence 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)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.513  -0.791  -0.639   0.983   2.373
##
## 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)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.825  -0.825  -0.719   1.576   1.720
##
## 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
```

```
msem.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(msem.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
```

```
##
```

```
## Estimated Scale Parameters:
```

```
##              Estimate Std.err
## (Intercept)      60.8    3.71
```

```
##
```

```
## Correlation: Structure = independenceNumber of clusters: 1566 Maximum cluster size: 1
```

```
beta <- coef(msem.emm)
```

```
SE <- coef(summary(msem.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$qsmk, nehefs$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 = nehefs
  )
```

```
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 = nehefs)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.465  -0.804  -0.646   1.058   2.355
##
## 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)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.781  -0.781  -0.781   1.635   1.635
##
## 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)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.097  -0.287  -0.207  -0.157   2.996
##
## 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)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.347  -0.254  -0.254  -0.254   2.628
##
## 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
##
## Estimated Scale Parameters:
##              Estimate Std.err
## (Intercept)   61.8    3.83
##

```

```
## Correlation: Structure = independenceNumber 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)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -42.056  -4.171  -0.343   3.891  44.606
##
```

```

## 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
##
## (Intercept)
## qsmk                **
## sex                 **
## race
## age                 *
## I(age * age)        ***
## as.factor(education)2
## as.factor(education)3
## as.factor(education)4
## as.factor(education)5
## smokeintensity
## I(smokeintensity * smokeintensity)
## smokeyrs
## I(smokeyrs * smokeyrs)
## as.factor(exercise)1
## as.factor(exercise)2
## as.factor(active)1    *
## as.factor(active)2
## wt71
## I(wt71 * wt71)        .
## qsmk:smokeintensity
## ---
## 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> <dbl>
## 1 0.342 0 0 0 26 4 15 12
## # ... with 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)
A <- c(0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1)
Y <- c(0, 1, 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)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.66667  -0.25000   0.04167   0.33333   0.75000
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.500e-01  2.552e-01   0.980   0.342
## A           -4.164e-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 <- dhefs # 2nd copy: treatment set to 0, outcome to missing
interv0$interv <- 0
interv0$qsmk <- 0
interv0$wt82_71 <- NA

interv1 <- dhefs # 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)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -42.056  -4.171  -0.343   3.891  44.606
##
## 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
##
## (Intercept)
## qsmk              **
## sex              **
## race
## age              *
## I(age * age)     ***
## as.factor(education)2
## as.factor(education)3
## as.factor(education)4
## as.factor(education)5
## smokeintensity
## I(smokeintensity * smokeintensity)
```

```

## smokeyrs
## I(smokeyrs * smokeyrs)
## as.factor(exercise)1
## as.factor(exercise)2
## as.factor(active)1      *
## as.factor(active)2
## wt71
## I(wt71 * wt71)          .
## I(qsmk * smokeintensity)
## ---
## 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 = nhfs,
  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
## 1      Observed 2.56188497106103 0.196821258955131
## 2      No Treatment 1.65212306626746 0.156859659349977
## 3      Treatment 5.11474489549347 0.453115935788573
## 4 Treatment - No Treatment 3.46262182922601 0.406373056272699
##           ll           ul
## 1 2.17612239211714 2.94764755000492
## 2 1.34468378331428 1.95956234922064
## 3 4.2266539805267 6.00283581046023
## 4 2.66614527464405 4.25909838380796
```


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)

## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
## Loading required package: ggplot2

##
## 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 : -41.28047 -30.50192 -30.05007 -29.02579 -25.97056
## highest:  34.01780  36.96925  37.65051  47.51130  48.53839

# 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)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.9959   0.1571   0.2069   0.2868   1.0967
##
## 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 cens
```

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
##
## (Intercept)                .
## sex                        ***
## race                       ***
## age                        *
## I(age * age)
## as.factor(education)2
## as.factor(education)3
## as.factor(education)4
## as.factor(education)5      *
## smokeintensity             ***
## I(smokeintensity * smokeintensity) ***
## smokeyrs                   **
## I(smokeyrs * smokeyrs)      .
## as.factor(exercise)1       .
## as.factor(exercise)2       *
## as.factor(active)1
## as.factor(active)2
## wt71
## I(wt71 * wt71)
## Hpsi
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Estimated Scale Parameters:
##           Estimate Std.err
## (Intercept)  0.9969 0.06717
##
## Correlation: Structure = independenceNumber of clusters: 1566 Maximum cluster size: 1

```

G-estimation: Checking multiple possible values of psi

```

#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

```

[illegible]

```
## 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)
```

```
##
```

```
## Deviance Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -1.529  -0.808  -0.650   1.029   2.417
```

```
##
```

```
## 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$pqsmk <- predict(logit.est, neufs, type = "response")
describe(nhefs$pqsmk)
```

```
## neufs$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 : 0.05145 0.05157 0.05438 0.05583 0.05931
## highest: 0.67208 0.68643 0.71391 0.73330 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 <- neufs[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)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -42.056  -4.171  -0.343   3.891  44.606
##
## 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
##
## (Intercept)
## qsmk                **
## sex                 **
## race
## age                 *
## I(age * age)        ***
## as.factor(education)2
## as.factor(education)3
## as.factor(education)4 .
## as.factor(education)5
## smokeintensity
## I(smokeintensity * smokeintensity)
## smokeyrs
## I(smokeyrs * smokeyrs)
## as.factor(exercise)1
## as.factor(exercise)2
## as.factor(active)1   *
## as.factor(active)2
## wt71
## I(wt71 * wt71)      .
## I(qsmk * smokeintensity)
## ---
## 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
## Effect of Quitting Smoking at Smokeintensity of 5          0   1   0
## Effect of Quitting Smoking at Smokeintensity of 40          0   1   0
##
##                                     race age I(age * age)
## Effect of Quitting Smoking at Smokeintensity of 5          0   0          0
## Effect of Quitting Smoking at Smokeintensity of 40          0   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) + smokeyr + I(smokeyr * smokeyr) + as.factor(exercise) +
## as.factor(active) + wt71 + I(wt71 * wt71) + I(qsmk * smokeintensity),
## data = nhefs)
##
## Linear Hypotheses:

```

```

##                                     Estimate
## Effect of Quitting Smoking at Smokeintensity of 5 == 0    2.7929
## Effect of Quitting Smoking at Smokeintensity of 40 == 0    4.4261
##                                     Std. Error z value
## Effect of Quitting Smoking at Smokeintensity of 5 == 0      0.6683   4.179
## Effect of Quitting Smoking at Smokeintensity of 40 == 0      0.8478   5.221
##                                     Pr(>|z|)
## Effect of Quitting Smoking at Smokeintensity of 5 == 0  5.84e-05 ***
## Effect of Quitting Smoking at Smokeintensity of 40 == 0  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:
##                                     Estimate lwr
## Effect of Quitting Smoking at Smokeintensity of 5 == 0  2.7929   1.3039
## Effect of Quitting Smoking at Smokeintensity of 40 == 0  4.4261   2.5372
##                                     upr
## Effect of Quitting Smoking at Smokeintensity of 5 == 0  4.2819
## Effect of Quitting Smoking at Smokeintensity of 40 == 0  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)
##

```

```

## Deviance Residuals:
##      Min        1Q    Median        3Q        Max
## -42.332   -4.216   -0.318    3.807   44.668
##
## 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
##
## (Intercept)
## qsmk          ***
## sex           **
## race
## age          *
## I(age * age)  ***
## as.factor(education)2
## as.factor(education)3
## as.factor(education)4 .
## as.factor(education)5
## smokeintensity
## I(smokeintensity * smokeintensity)
## smokeyrs
## I(smokeyrs * smokeyrs)
## as.factor(exercise)1
## as.factor(exercise)2
## as.factor(active)1    *
## as.factor(active)2
## wt71
## I(wt71 * wt71)      .
## ---
## 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 = dhefs)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.4646  -0.8044  -0.6460   1.0578   2.3550
##
## 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
##
## (Intercept)
## sex                        ***
## race                       ***
## age                        *
## I(age * age)
## as.factor(education)2
## as.factor(education)3
## as.factor(education)4
## as.factor(education)5      .
## smokeintensity             ***
## I(smokeintensity * smokeintensity) **
## smokeyrs                   **
## I(smokeyrs * smokeyrs)      .
## as.factor(exercise)1        .
## as.factor(exercise)2        *
## as.factor(active)1
## as.factor(active)2
## wt71
## I(wt71 * wt71)
## ---
## 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")

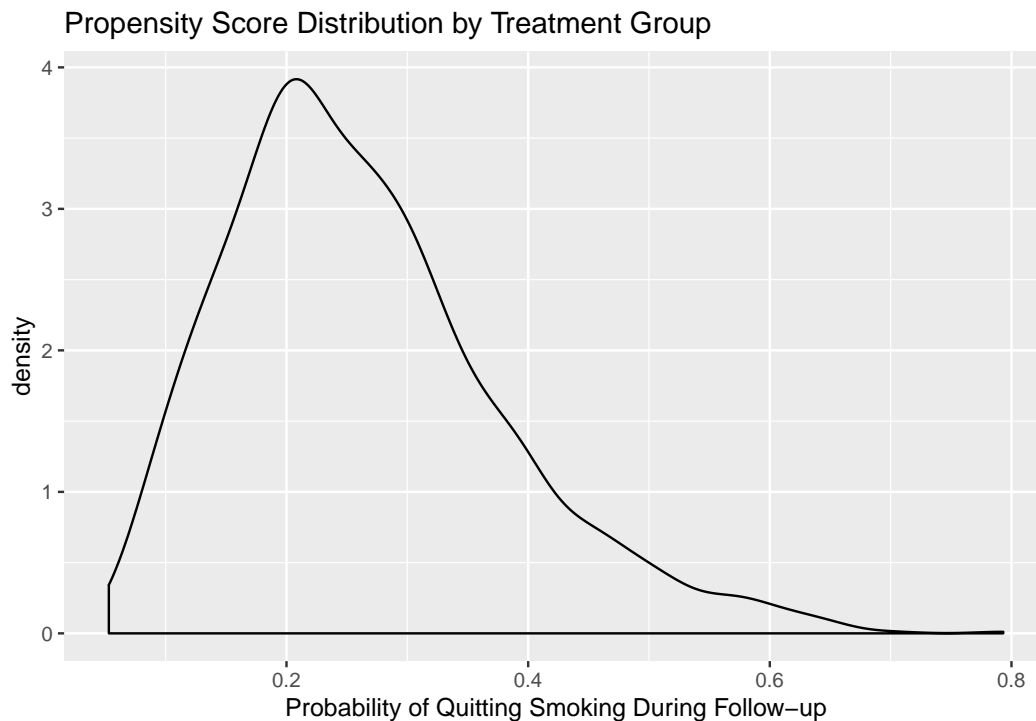
```



```
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')
```

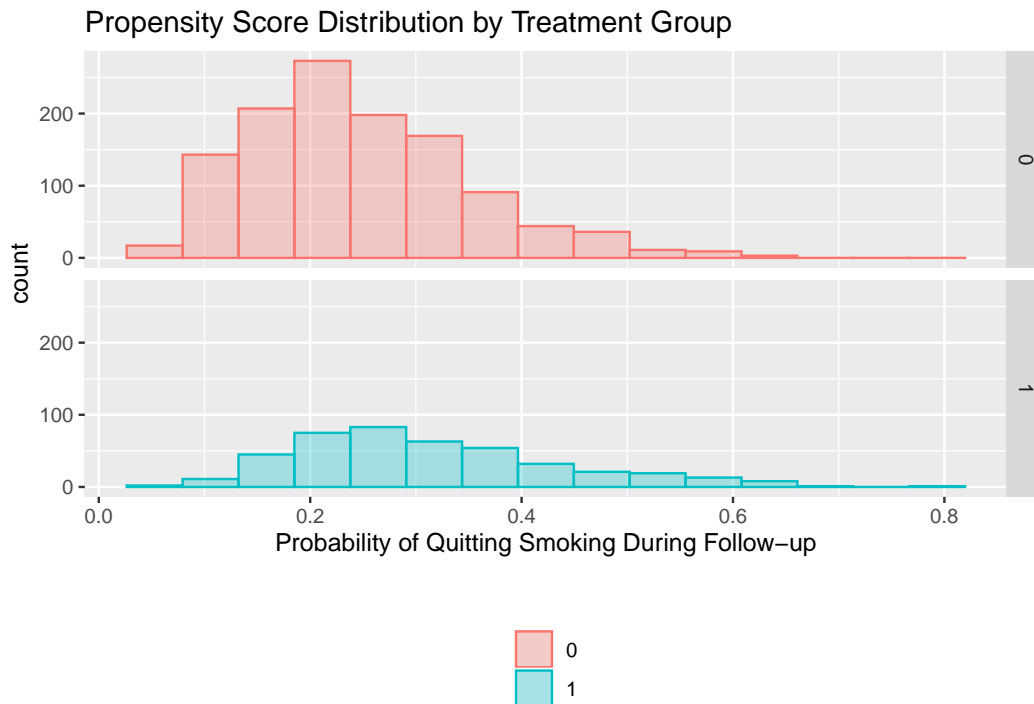


```
# alternative plot with histograms  
nhefs <- nhefs %>% mutate(qsmklabel = ifelse(qsmk == 1,  
  yes = 'Quit Smoking 1971-1982',  
  no = 'Did Not Quit Smoking 1971-1982'))  
ggplot(nhefs, aes(x = ps, fill = as.factor(qsmk), color = as.factor(qsmk))) +  
  geom_histogram(alpha = 0.3, position = 'identity', 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

```

nhfs %>%
  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, nhefs$qsmk))

##
## Descriptive statistics by group
## : 1
## : 0
##   vars   n mean   sd median trimmed  mad  min  max range  skew kurtosis
## X1      1 151  0.1 0.02   0.11     0.1 0.02 0.05 0.13  0.08 -0.55   -0.53
##   se
## X1      0
## -----
## : 2
## : 0
##   vars   n mean   sd median trimmed  mad  min  max range  skew kurtosis
## X1      1 136  0.15 0.01   0.15     0.15 0.01 0.13 0.17  0.04 -0.04   -1.23
##   se
## X1      0
## -----
## : 3
## : 0
##   vars   n mean   sd median trimmed  mad  min  max range  skew kurtosis
## X1      1 134  0.18 0.01   0.18     0.18 0.01 0.17 0.19  0.03 -0.08   -1.34
##   se
## X1      0
## -----
## : 4
## : 0
##   vars   n mean   sd median trimmed  mad  min  max range  skew kurtosis
## X1      1 129  0.21 0.01   0.21     0.21 0.01 0.19 0.22  0.02 -0.04   -1.13
##   se
```

```

## X1 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
## X1 1 117 0.26 0.01 0.26 0.26 0.01 0.25 0.27 0.03 -0.11 -1.29
## se
## X1 0
## -----
## : 7
## : 0
## vars n mean sd median trimmed mad min max range skew kurtosis
## X1 1 120 0.29 0.01 0.29 0.29 0.01 0.27 0.31 0.03 -0.23 -1.19
## se
## X1 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
## X1 1 86 0.49 0.06 0.47 0.48 0.05 0.42 0.66 0.24 1.1 0.47
## se
## X1 0.01
## -----
## : 1
## : 1
## vars n mean sd median trimmed mad min max range skew kurtosis
## X1 1 12 0.1 0.02 0.11 0.1 0.03 0.06 0.13 0.07 -0.5 -1.36
## se
## X1 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
## X1     1 77 0.52 0.08   0.51   0.51 0.08 0.42 0.79  0.38 0.88   0.81
##       se
## X1 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 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 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 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 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 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 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 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 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 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 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)
##
## Deviance Residuals:
## Min 1Q Median 3Q Max
## -43.543 -3.932 -0.085 4.233 46.773
##
## 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
## 1      Observed 2.63384609228479 0.194461632915776
## 2      No Treatment 1.71983636149843 0.238038648388876
## 3      Treatment 5.35072300362993 0.439399620871625
## 4 Treatment - No Treatment 3.63088664213151 0.422213269021722
##              ll              ul

```

```
## 1 2.25270829539502 3.01498388917456
## 2 1.25328918372763 2.18638353926922
## 3 4.48951557190099 6.21193043535887
## 4 2.80336384105401 4.458409443209

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

##
## Call:
## glm(formula = wt82_71 ~ qsmk + ps, data = nhefs)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -43.314  -4.006  -0.068   4.244  47.158
##
## 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  
## 2.580653 , 4.514927
```

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 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 ψ
- 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 = dhefs.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
##
## Estimated Scale Parameters:
##             Estimate Std.err
## (Intercept)      1 0.7607
##
## Correlation: Structure = independenceNumber 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 = dhefs.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$yrdrth-83)*12+nhefs$modth) # yrdrth 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

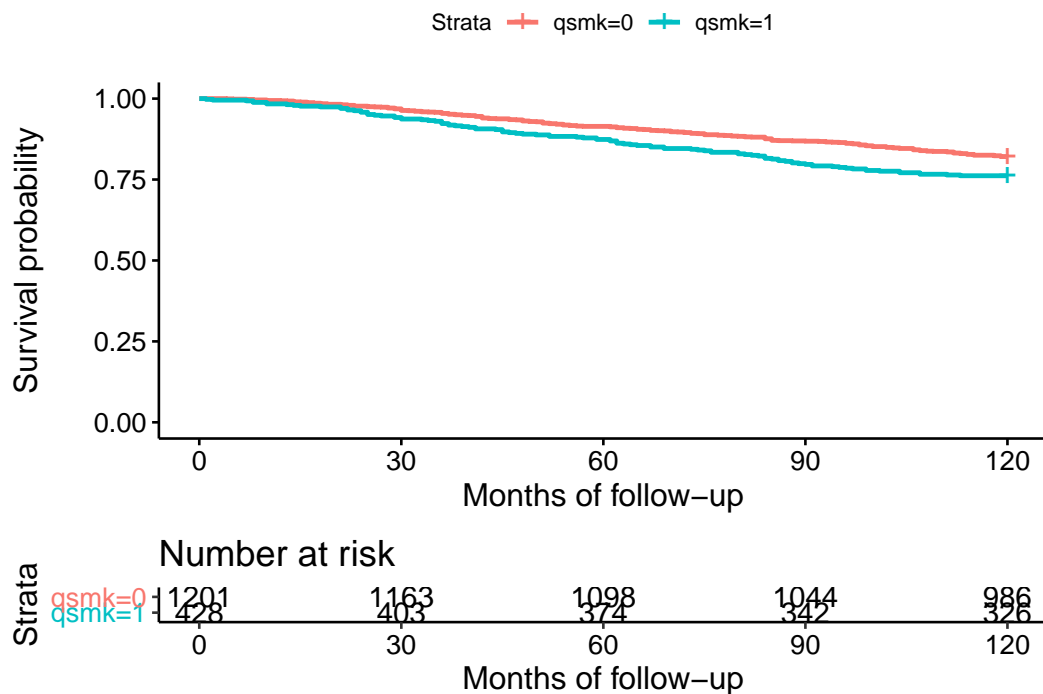
## Loading required package: magrittr

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")
nhefs.surv <- expandRows(nhefs, "survtime", drop=F)
nhefs.surv$time <- sequence(rle(nhefs.surv$seqn)$lengths)-1
nhefs.surv$event <- ifelse(nhefs.surv$time==nhefs.surv$survtime-1 &
                           nhefs.surv$death==1, 1, 0)
nhefs.surv$timesq <- nhefs.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=nhefs.surv)
summary(hazards.model)

```

```

##
## Call:
## glm(formula = event == 0 ~ qsmk + I(qsmk * time) + I(qsmk * timesq) +
##     time + timesq, family = binomial(), data = dhefs.surv)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.7253   0.0546   0.0601   0.0625   0.0783
##
## 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")

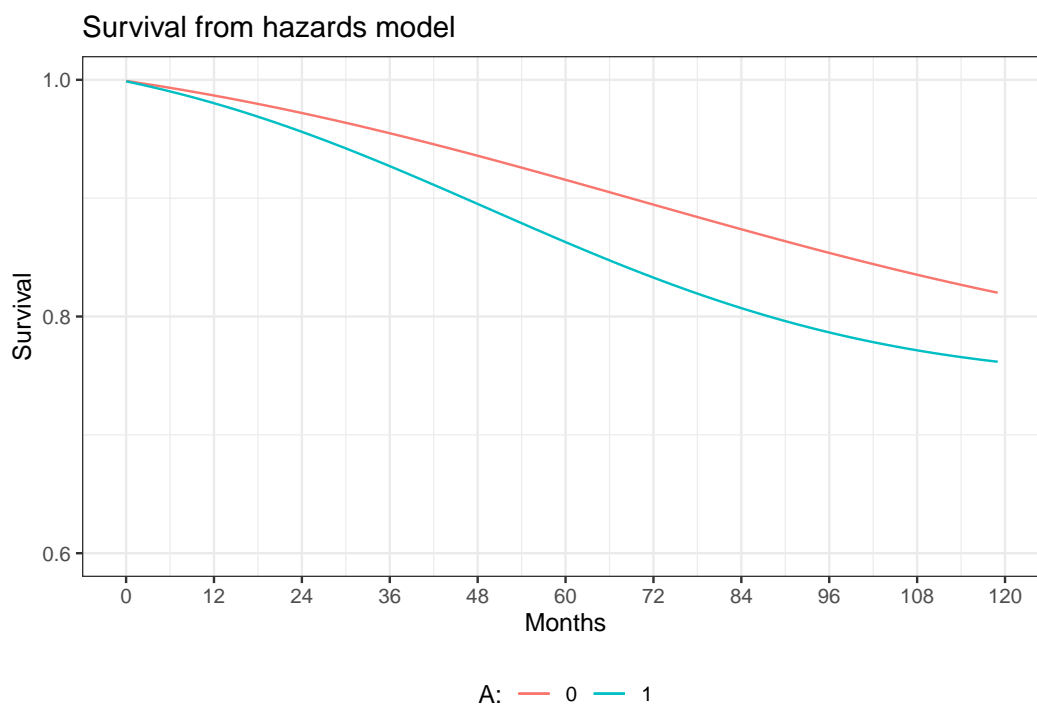
# assignment of estimated (1-hazard) to each person-month */
qsmk0$p.noevent0 <- predict(hazards.model, qsmk0, type="response")
qsmk1$p.noevent1 <- predict(hazards.model, qsmk1, type="response")

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

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

```

```
# 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)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -7.1859   0.0528   0.0595   0.0640   0.1452
##
## 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))

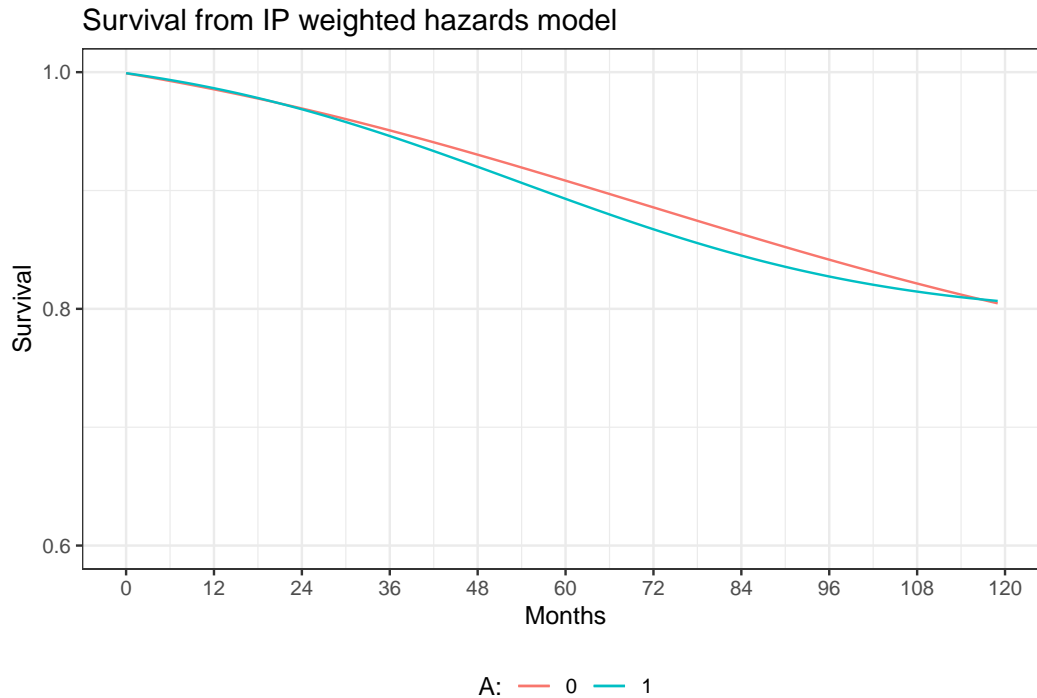
colnames(ipw.qsmk0) <- c("time", "qsmk", "timesq")
colnames(ipw.qsmk1) <- c("time", "qsmk", "timesq")

# 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$survdif <- ipw.graph$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 = dhefs.surv)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -4.3160   0.0244   0.0395   0.0640   0.3303
```

```

##
## 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
##
## (Intercept) ***
## qsmk
## I(qsmk * time)
## I(qsmk * timesq)
## time **
## timesq .
## sex **
## race
## age
## I(age * age)
## as.factor(education)2
## as.factor(education)3 **
## as.factor(education)4
## as.factor(education)5
## smokeintensity
## I(smokeintensity * smokeintensity)
## smkintensity82_71
## smokeyrs
## I(smokeyrs * smokeyrs)
## as.factor(exercise)1

```

```

## as.factor(exercise)2
## as.factor(active)1
## as.factor(active)2
## wt71 **
## I(wt71 * wt71) ***
## ---
## 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

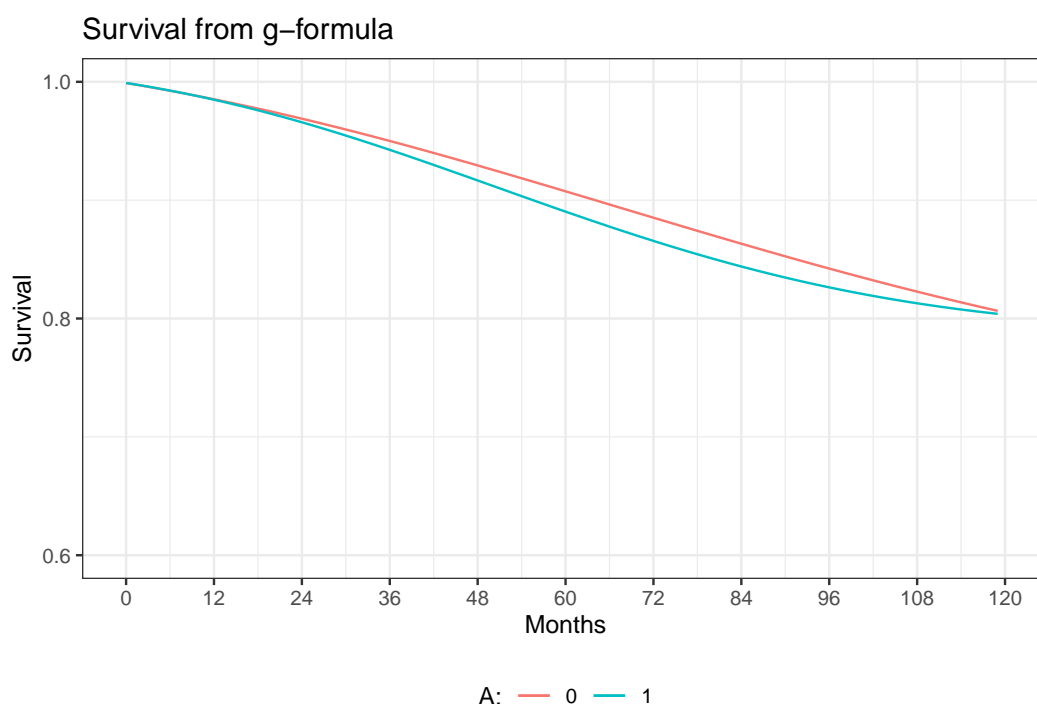
gf.qsmk0.surv <- gf.qsmk0 %>% group_by(seqn) %>% mutate(surv0 = cumprod(p.noevent0))
gf.qsmk1.surv <- gf.qsmk1 %>% group_by(seqn) %>% mutate(surv1 = cumprod(p.noevent1))

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$survdif <- gf.graph$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$yrdth-83)*12+nhefs$modth) # * yrdth ranges from 83

# 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
psilow <- psi1
testlow <- objfunc
countlow <- 0
while (testlow < 3.84 & countlow < 100){
  psilow <- psilow - increm
  testlow <- sumeef(psilow)
  countlow <- countlow + 1
}

# Upper bound of 95% CI
psihigh <- psi1
testhigh <- objfunc
counthigh <- 0
while (testhigh < 3.84 & counthigh < 100){
  psihigh <- psihigh + increm
  testhigh <- sumeef(psihigh)
  counthigh <- counthigh + 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
```


R session information

For reproducibility.

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

```
## - Session info -----
## setting value
## version R version 3.6.1 (2019-07-05)
## os      Windows 10 x64
## system  x86_64, mingw32
## ui      RTerm
## language (EN)
## collate English_United Kingdom.1252
## ctype   English_United Kingdom.1252
## tz      Europe/London
## date    2019-10-02
##
## - Packages -----
## package      * version date      lib source
## assertthat   0.2.1   2019-03-21 [1] CRAN (R 3.6.0)
## bookdown     0.14    2019-10-01 [1] CRAN (R 3.6.1)
## cli          1.1.0   2019-03-19 [1] CRAN (R 3.6.0)
## crayon       1.3.4   2017-09-16 [1] CRAN (R 3.6.0)
## digest       0.6.21  2019-09-20 [1] CRAN (R 3.6.1)
## evaluate     0.14    2019-05-28 [1] CRAN (R 3.6.0)
## htmltools    0.3.6   2017-04-28 [1] CRAN (R 3.6.0)
## knitr        1.25    2019-09-18 [1] CRAN (R 3.6.1)
## magrittr     1.5     2014-11-22 [1] CRAN (R 3.6.0)
## Rcpp         1.0.2   2019-07-25 [1] CRAN (R 3.6.1)
## rmarkdown    1.16    2019-10-01 [1] CRAN (R 3.6.1)
## sessioninfo  1.1.1   2018-11-05 [1] CRAN (R 3.6.0)
## stringi      1.4.3   2019-03-12 [1] CRAN (R 3.6.0)
## stringr      1.4.0   2019-02-10 [1] CRAN (R 3.6.0)
## withr        2.1.2   2018-03-15 [1] CRAN (R 3.6.0)
## xfun         0.10    2019-10-01 [1] CRAN (R 3.6.1)
## yaml         2.2.0   2018-07-25 [1] CRAN (R 3.6.0)
##
## [1] C:/Users/palmertm/library
## [2] C:/Program Files/R/R-3.6.1/library
```

