Data Analysis Homework 2

Jimmy Hickey

9/25/2020

1

We will proceed with the same that I used in homework 1.

```
library(DynTxRegime)
```

```
## Loading required package: modelObj
```

```
data <- read.csv(file ="cholesterol.dat.txt", header = TRUE, sep = ",")
data$A= data$trt

y = data$chol0 - data$chol6

lm = buildModelObj(model = ~A + exercise + wt + smoke + trig0 + age + gender +
                      A:exercise + A:wt + A:smoke + A:trig0 + A:age + A:gender,
                    solver.method = "lm",
                    predict.method = "predict.lm",
                    predict.args = list("type"="response"))
# adj R^2 = 0.889

# summary(fit(lm, data, y))
```

a. regression-based estimator

```
# From slide 35 of Halloway
moMain <- buildModelObj(model = ~exercise + wt + smoke + trig0 + age + gender,
        solver.method = 'lm',
        predict.method = 'predict.lm')

moCont <- buildModelObj(model = ~exercise + wt + smoke + trig0 + age + gender,
        solver.method = 'lm',
        predict.method = 'predict.lm')

qObj <- qLearn(moMain = moMain, moCont = moCont, iter = 0L,
          data = data, response = y, txName = 'A',
          verbose = TRUE)
```

```
## First step of the Q-Learning Algorithm.
##
## Outcome regression.
## Combined outcome regression model: ~ exercise+wt+smoke+trig0+age+gender + A + A:(exercise+wt+smoke+tr
## Regression analysis for Combined:
##
## Call:
## lm(formula = YinternalY ~ exercise + wt + smoke + trig0 + age +
##     gender + A + exercise:A + wt:A + smoke:A + trig0:A + age:A +
##     gender:A, data = data)
##
## Coefficients:
## (Intercept)     exercise           wt        smoke        trig0          age
##   3.246e+01    2.058e+01   -2.393e-01    2.908e+00   -1.671e-02    9.491e-03
##      gender            A   exercise:A         wt:A      smoke:A      trig0:A
##   5.239e-01   -2.615e+02   -2.108e+01    1.621e+00   -5.081e+00    3.512e-02
##       age:A     gender:A
##   2.488e-02    9.056e-01
##
##
## Recommended Treatments:
##   0   1
## 211 789
##
## Estimated value: 33.75671
```

```r
coef(object = qObj)
```

```
## $outcome
## $outcome$Combined
##   (Intercept)      exercise            wt         smoke         trig0
##  3.246193e+01  2.058365e+01 -2.392622e-01  2.907925e+00 -1.671489e-02
##           age        gender             A    exercise:A          wt:A
##  9.490884e-03  5.239228e-01 -2.614569e+02 -2.107722e+01  1.620786e+00
##       smoke:A       trig0:A         age:A      gender:A
## -5.081452e+00  3.511518e-02  2.488064e-02  9.055717e-01
```

```r
fitObj = fitObject(object = qObj)
fitObj
```

```
## $outcome
## $outcome$Combined
##
## Call:
## lm(formula = YinternalY ~ exercise + wt + smoke + trig0 + age +
##     gender + A + exercise:A + wt:A + smoke:A + trig0:A + age:A +
##     gender:A, data = data)
##
## Coefficients:
## (Intercept)     exercise           wt        smoke        trig0          age
##   3.246e+01    2.058e+01   -2.393e-01    2.908e+00   -1.671e-02    9.491e-03
##      gender            A   exercise:A         wt:A      smoke:A      trig0:A
##   5.239e-01   -2.615e+02   -2.108e+01    1.621e+00   -5.081e+00    3.512e-02
```

```
##       age:A     gender:A
##   2.488e-02    9.056e-01
```

```r
ot <- optTx(x = qObj)
table(ot$optimalTx)
```

```
##
##   0   1
## 211 789
```

```r
estimator(x = qObj)
```

```
## [1] 33.75671
```

b. restricted value search

```r
# slide 54 of Halloway
regimes = function(eta1, data)
{
  d1 = {data$wt > eta1}
  return(as.integer(x = d1))
}

# propensity model from hw1
propensity <- modelObj::buildModelObj(model = ~ age + wt + gender + exercise + smoke + trig0 + chol0,
    solver.method = 'glm',
    solver.args = list(family='binomial'),
    predict.method = 'predict.glm',
    predict.args = list(type='response'))

# optimal seq from slide 56 of Halloway
# notice we only need a propensity model for equation 3.42
vsObj <- optimalSeq(moPropen = propensity,
            moMain = NULL, moCont = NULL, iter = 0L,
            data = data, response = y, txName = 'A',
            regimes = regimes,
            Domains = matrix(data = c(110, 290), ncol = 2L),
            starting.values = c(0,0), pop.size = 1000,
            verbose = TRUE)
```

```
## IPW estimator will be used
## Value Search - Missing Data Perspective.
##
## Propensity for treatment regression.
## Regression analysis for moPropen:
##
## Call:  glm(formula = YinternalY ~ age + wt + gender + exercise + smoke +
##     trig0 + chol0, family = "binomial", data = data)
##
## Coefficients:
```

```
## (Intercept)          age           wt       gender     exercise        smoke
##  -2.9404887    0.0009917    0.0083114   -0.0929438    0.3816924   -0.0976649
##       trig0        chol0
##  -0.0009232    0.0061492
##
## Degrees of Freedom: 999 Total (i.e. Null);  992 Residual
## Null Deviance:       1386
## Residual Deviance: 1366  AIC: 1382
##
## Outcome regression.
## No outcome regression performed.

## Warning in (function (fn, nvars, max = FALSE, pop.size = 1000, max.generations =
## 100, : Ignoring 'starting.values' because length(staring.values)!=nvars

##
##
## Thu Oct 01 08:42:44 2020
## Domains:
##  1.100000e+02   <=  X1   <=    2.900000e+02
##
## Data Type: Floating Point
## Operators (code number, name, population)
##  (1) Cloning..........................  122
##  (2) Uniform Mutation.................  125
##  (3) Boundary Mutation................  125
##  (4) Non-Uniform Mutation.............  125
##  (5) Polytope Crossover...............  125
##  (6) Simple Crossover.................  126
##  (7) Whole Non-Uniform Mutation........  125
##  (8) Heuristic Crossover..............  126
##  (9) Local-Minimum Crossover..........  0
##
## HARD Maximum Number of Generations: 100
## Maximum Nonchanging Generations: 10
## Population size       : 1000
## Convergence Tolerance: 1.000000e-03
##
## Not Using the BFGS Derivative Based Optimizer on the Best Individual Each Generation.
## Not Checking Gradients before Stopping.
## Using Out of Bounds Individuals.
##
## Maximization Problem.
##
##
## Generation#      Solution Value
##
##       0  3.440739e+01
##
## 'wait.generations' limit reached.
## No significant improvement in 10 generations.
##
## Solution Fitness Value: 3.440739e+01
##
```

```
## Parameters at the Solution:
##
##  X[ 1] : 1.578440e+02
##
## Solution Found Generation 1
## Number of Generations Run 11
##
## Thu Oct 01 08:42:56 2020
## Total run time : 0 hours 0 minutes and 12 seconds
## Genetic Algorithm
## $value
## [1] 34.40739
##
## $par
## [1] 157.844
##
## $gradients
## [1] NA
##
## $generations
## [1] 11
##
## $peakgeneration
## [1] 1
##
## $popsize
## [1] 1000
##
## $operators
## [1] 122 125 125 125 125 126 125 126   0
##
##
## Recommended Treatments:
##   0   1
## 224 776
##
## Estimated value: 34.40739
```

```r
regimeCoef(vsObj)
```

```
##    eta1
## 157.844
```

```r
estimator(vsObj)
```

```
## [1] 34.40739
```

Notice that our optimal eta is $\eta = 157.8948$. This agrees with the graph that we got from homework 1 question 2 b!

c

```r
# notice that the difference here is we need our moMain and moCont are no longer null
vsObj2 <- optimalSeq(moPropen = propensity,
            moMain = moMain, moCont = moCont, iter = 0L,
            data = data, response = y, txName = 'A',
            regimes = regimes,
            Domains = matrix(data = c(110, 290), ncol = 2L),
            starting.values = c(0,0), pop.size = 1000,
            verbose = TRUE)
```

```
## Value Search - Missing Data Perspective.
##
## Propensity for treatment regression.
## Regression analysis for moPropen:
##
## Call:  glm(formula = YinternalY ~ age + wt + gender + exercise + smoke +
##     trig0 + chol0, family = "binomial", data = data)
##
## Coefficients:
## (Intercept)          age           wt       gender     exercise        smoke
##  -2.9404887    0.0009917    0.0083114   -0.0929438    0.3816924   -0.0976649
##       trig0        chol0
##  -0.0009232    0.0061492
##
## Degrees of Freedom: 999 Total (i.e. Null);  992 Residual
## Null Deviance:       1386
## Residual Deviance: 1366  AIC: 1382
##
## Outcome regression.
## Combined outcome regression model: ~ exercise+wt+smoke+trig0+age+gender + A + A:(exercise+wt+smoke+t
## Regression analysis for Combined:
##
## Call:
## lm(formula = YinternalY ~ exercise + wt + smoke + trig0 + age +
##     gender + A + exercise:A + wt:A + smoke:A + trig0:A + age:A +
##     gender:A, data = data)
##
## Coefficients:
## (Intercept)     exercise           wt        smoke        trig0          age
##   3.246e+01    2.058e+01   -2.393e-01    2.908e+00   -1.671e-02    9.491e-03
##      gender            A   exercise:A         wt:A      smoke:A      trig0:A
##   5.239e-01   -2.615e+02   -2.108e+01    1.621e+00   -5.081e+00    3.512e-02
##       age:A     gender:A
##   2.488e-02    9.056e-01

## Warning in (function (fn, nvars, max = FALSE, pop.size = 1000, max.generations =
## 100, : Ignoring 'starting.values' because length(staring.values)!=nvars

##
##
## Thu Oct 01 08:42:56 2020
```

```
## Domains:
##  1.100000e+02   <=  X1   <=    2.900000e+02
##
## Data Type: Floating Point
## Operators (code number, name, population)
##  (1) Cloning..........................  122
##  (2) Uniform Mutation.................  125
##  (3) Boundary Mutation................  125
##  (4) Non-Uniform Mutation.............  125
##  (5) Polytope Crossover...............  125
##  (6) Simple Crossover.................  126
##  (7) Whole Non-Uniform Mutation........  125
##  (8) Heuristic Crossover..............  126
##  (9) Local-Minimum Crossover..........  0
##
## HARD Maximum Number of Generations: 100
## Maximum Nonchanging Generations: 10
## Population size       : 1000
## Convergence Tolerance: 1.000000e-03
##
## Not Using the BFGS Derivative Based Optimizer on the Best Individual Each Generation.
## Not Checking Gradients before Stopping.
## Using Out of Bounds Individuals.
##
## Maximization Problem.
##
##
## Generation#      Solution Value
##
##       0  3.472935e+01
##
## 'wait.generations' limit reached.
## No significant improvement in 10 generations.
##
## Solution Fitness Value: 3.472935e+01
##
## Parameters at the Solution:
##
##  X[ 1] : 1.586857e+02
##
## Solution Found Generation 1
## Number of Generations Run 11
##
## Thu Oct 01 08:43:45 2020
## Total run time : 0 hours 0 minutes and 49 seconds
## Genetic Algorithm
## $value
## [1] 34.72935
##
## $par
## [1] 158.6857
##
## $gradients
## [1] NA
```

```
##
## $generations
## [1] 11
##
## $peakgeneration
## [1] 1
##
## $popsize
## [1] 1000
##
## $operators
## [1] 122 125 125 125 125 126 125 126   0
##
##
## Recommended Treatments:
##   0   1
## 227 773
##
## Estimated value: 34.72935
```

```
regimeCoef(vsObj2)
```

```
##     eta1
## 158.6857
```

```
estimator(vsObj2)
```

```
## [1] 34.72935
```

Notice that our optimal eta is $\eta = 158.5162$. This is similar to the estimate in (b) and also with the graph from homework 1.

d

```
require(rpart)
```

```
## Loading required package: rpart
```

```
moClass <- buildModelObj(model = ~exercise + wt + smoke + trig0 + age + gender,
  solver.method = 'rpart',
  predict.method = 'predict',
  predict.args = list(type = "class"))


clObj <- optimalClass(moPropen = propensity,
            moMain = moMain, moCont = moCont, iter = 0L,
            moClass = moClass,
            data = data, response = y, txName = 'A',
            verbose = TRUE)
```

```
## AIPW value estimator

## First step of the Classification Algorithm.

## Classification Perspective.

##
## Propensity for treatment regression.
## Regression analysis for moPropen:
##
## Call:  glm(formula = YinternalY ~ age + wt + gender + exercise + smoke +
##     trig0 + chol0, family = "binomial", data = data)
##
## Coefficients:
## (Intercept)          age           wt       gender     exercise        smoke
##  -2.9404887    0.0009917    0.0083114   -0.0929438    0.3816924   -0.0976649
##       trig0        chol0
##  -0.0009232    0.0061492
##
## Degrees of Freedom: 999 Total (i.e. Null);  992 Residual
## Null Deviance:       1386
## Residual Deviance: 1366  AIC: 1382
##
## Outcome regression.
## Combined outcome regression model: ~ exercise+wt+smoke+trig0+age+gender + A + A:(exercise+wt+smoke+t
## Regression analysis for Combined:
##
## Call:
## lm(formula = YinternalY ~ exercise + wt + smoke + trig0 + age +
##     gender + A + exercise:A + wt:A + smoke:A + trig0:A + age:A +
##     gender:A, data = data)
##
## Coefficients:
## (Intercept)     exercise           wt        smoke        trig0          age
##   3.246e+01    2.058e+01   -2.393e-01    2.908e+00   -1.671e-02    9.491e-03
##      gender            A   exercise:A         wt:A      smoke:A      trig0:A
##   5.239e-01   -2.615e+02   -2.108e+01    1.621e+00   -5.081e+00    3.512e-02
##       age:A     gender:A
##   2.488e-02    9.056e-01
##
##
## Classification Analysis
## Regression analysis for moClass:
## n= 1000
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 1000 0.138746100 1 (0.0252960136 0.9747039864)
##    2) wt< 158.55 227 0.010564510 0 (0.6448357626 0.3551642374) *
##    3) wt>=158.55 773 0.019682370 1 (0.0037134730 0.9962865270)
##      6) wt< 167.25 99 0.015095020 1 (0.0935823105 0.9064176895)
##       12) smoke>=0.5 24 0.002070983 0 (0.3282265705 0.6717734295) *
```

```
##       13) smoke< 0.5 75 0.008813891 1 (0.0619973915 0.9380026085)
##         26) exercise>=0.5 14 0.000556363 0 (0.5534174300 0.4465825700) *
##         27) exercise< 0.5 61 0.004534129 1 (0.0337280043 0.9662719957) *
##      7) wt>=167.25 674 0.004587345 1 (0.0008926608 0.9991073392) *
## Recommended Treatments:
##   0   1
## 265 735
##
## Estimated value: 35.17289
```

```r
coef(object = clObj)
```

```
## $propensity
##   (Intercept)           age            wt        gender      exercise
## -2.9404886577  0.0009916959  0.0083114322 -0.0929437626  0.3816924426
##         smoke         trig0         chol0
## -0.0976648828 -0.0009231713  0.0061491542
##
## $outcome
## $outcome$Combined
##   (Intercept)      exercise            wt         smoke         trig0
##  3.246193e+01  2.058365e+01 -2.392622e-01  2.907925e+00 -1.671489e-02
##           age        gender             A    exercise:A          wt:A
##  9.490884e-03  5.239228e-01 -2.614569e+02 -2.107722e+01  1.620786e+00
##       smoke:A       trig0:A         age:A      gender:A
## -5.081452e+00  3.511518e-02  2.488064e-02  9.055717e-01
```

```r
table(ot$optimalTx)
```

```
##
##   0   1
## 211 789
```

```r
estimator(x = clObj)
```

```
## [1] 35.17289
```

2

a

```r
ldl = read.table("LDL.dat.txt", header=FALSE)

# remove ID column
ldl = ldl[,-1]
names(ldl) = c("L1", "A1", "L2", "S2", "A2", "L3",
               "S3", "A3", "L4", "S4", "A4", "Y", "S5")

calc_betas = function(data, K){
```

```r
### Setting up variables in equations

  # number of datapoints
  n = dim(data)[1]

  # LDL measurements
  L = data[,c(1,3,6,9,12)]

  # Side effect experienced
  S = data[,c(4,7,10,13)]

  # Statin dose received
  A = data[,c(2,5,8,11)]

  # Y outcome vector
  # shoutout to Samsul for helping me build Y
  Y = as.numeric(t(cbind(L[,1],L[,2:5]-L[,1:4])))

  # X design matrix
  X = NULL

  for(i in 1:n){
    X = rbind(X,
              c(1, rep(0,6)))

    for(k in 2:(K+1)){
      X = rbind(X,
                c(0,
                  1 - S[i, k-1],
                  A[i, k-1]*(1-S[i,k-1]),
                  L[i,k-1]*(1-S[i,k-1]),
                  A[i,k-1]*L[i,k-1]*(1-S[i,k-1]),
                  S[i,k-1],
                  S[i,k-1]*A[i,k-1]))
    }
  }

  # fit linear model
  # -1 removes intercept
  lm = lm(Y ~ -1 + X)
  return(lm)
}

K = 4

betaslm = calc_betas(ldl, K)
betas = coef(betaslm)
sigmasq = (summary(betaslm)$sigma)^2


cat("==================\n
betas\n
==================")
```

```
## =================
##
## betas
##
## ==================
```

```r
print(betas)
```

```
##            X1           X2           X3           X4           X5
## 170.092400000  -6.112302738 -11.970236677  -0.003808885   0.013909115
##            X6           X7
##  -6.592123769  -7.052320675
```

```r
cat("=================\n
sigma^2\n
==================")
```

```
## =================
##
## sigma^2
##
## ==================
```

```r
print(sigmasq)
```

```
## [1] 144.3508
```

b

```r
calc_psis = function(data, K){

  # number of datapoints
  n = dim(data)[1]

  # LDL measurements
  L = data[,c(1,3,6,9,12)]

  # Side effect experienced
  S = data[,c(4,7,10,13)]

  # Statin dose received
  A = data[,c(2,5,8,11)]

  # Y outcome vector
  # take off side effects at 12 months
  # again, thanks Samsul!
  Ylogis = as.numeric(as.matrix(S[, -4]))

  # X design matrix for logistic regression
  Xlogis = NULL
```

```
  for(i in 1:n){
    for(k in 2:K){
      abark = sum(A[i, 1:(k-1)])
      Xlogis = rbind(Xlogis,
                     c(abark,
                       abark * L[i, k-1],
                       S[i, k-1] * A[i, k-1]
                       ))
    }

  }

  psifit = glm(Ylogis ~ Xlogis, family = binomial)
  psis = coef(psifit)
  return(psis)
}

psis = calc_psis(ldl, K)

cat("=================\n
psis\n
==================")
```

```
## =================
##
## psis
##
## ==================
```

```
print(psis)
```

```
##   (Intercept)       Xlogis1       Xlogis2       Xlogis3
## -2.5114156150 -0.1045054890  0.0006132861 -0.1179023947
```

c

```
logistic_func = function(x){
  return( exp(x) / (1 + exp(x)) )
}


gcomp = function(data, regime, K, M){
  bfit = calc_betas(data, K)
  betas = bfit$coefficients
  sigma = summary(bfit)$sigma

  psis = calc_psis(data, K)

  # it would be more efficient to make this of length M, but that isn't working
  y = NULL
```

```
  for(r in 1:M){
    L = rep(0, K+1)
    S = rep(0, K+1)
    A = rep(0, K)

    # random draw for L1
    L[1] = rnorm(n=1, mean=betas[1], sd=sigma)

    for(k in 2:K+1){
      # dose
      A[k-1] = regime(L, S, A, k-1)

      # Equation 3
      mu = L[k-1] + (betas[2] + betas[3]*A[k-1] + betas[4]*L[k-1] +
             betas[5]*A[k-1]*L[k-1]) * (1-S[k-1]) +
             (betas[6] + betas[7]*A[k-1])*S[k-1]
      L[k] = rnorm(n=1, mean=mu, sd=sigma)

      # Equation 4
      Acum = sum(A[1:k-1])
      prob = logistic_func( psis[1] + psis[2] * Acum +
                        psis[3] * Acum * L[k-1] + psis[4] * S[k-1] * A[k-1])
      S[k] = rbinom(n=1, size=1, prob=prob)
    }

    y = rbind(y,L[K+1])
  }

  return(mean(y))
}

bootstrap_gcomp = function(data, regime, K, M, rep){
  out = NULL
  nrow = dim(data[1])

  for(i in 1:rep){
    sample = data[sample(nrow, replace=TRUE),]
    out = rbind(out, gcomp(data, regime, K, M))
  }

  return(sd(out))
}

# STATIC REGIMES
stat_reg1 = function(L, S, A, dk){
  return(0)
}

stat_reg2 = function(L, S, A, dk){
  return(dk %in% c(4) )
}
```

```r
stat_reg3 = function(L, S, A, dk){
  return(dk %in% c(3, 4) )
}

stat_reg4 = function(L, S, A, dk){
  return(dk %in% c(2, 3, 4) )
}

stat_reg5 = function(L, S, A, dk){
  return(dk %in% c(1, 2, 3, 4) )
}

stat_reg6 = function(L, S, A, dk){
  return(dk %in% c(1, 2, 3) )
}

stat_reg7 = function(L, S, A, dk){
  return(dk %in% c(1,2) )
}

stat_reg8 = function(L, S, A, dk){
  return(dk %in% c(1) )
}

# regime 1
est1 = gcomp(data = ldl, regime = stat_reg1, K = 4, M = 1000)
sd1 = bootstrap_gcomp(data = ldl, regime = stat_reg1, K = 4, M = 1000, rep = 10)

cat("================\nregime 1\nestimate:\t", est1, "\nstderr:\t\t", sd1, "\n================")
```

```
## ================
## regime 1
## estimate:     -18.32332
## stderr:       0.4245063
## ================
```

```r
# regime 2
est2 = gcomp(data = ldl, regime = stat_reg2, K = 4, M = 1000)
sd2 = bootstrap_gcomp(data = ldl, regime = stat_reg2, K = 4, M = 1000, rep = 10)

cat("================\nregime 2\nestimate:\t", est2, "\nstderr:\t\t", sd2, "\n================")
```

```
## ================
## regime 2
## estimate:     -29.29085
## stderr:       0.506684
## ================
```

```r
# regime 3
est3 = gcomp(data = ldl, regime = stat_reg3, K = 4, M = 1000)
sd3 = bootstrap_gcomp(data = ldl, regime = stat_reg3, K = 4, M = 1000, rep = 10)

cat("================\nregime 3\nestimate:\t", est3, "\nstderr:\t\t", sd3, "\n================")
```

```
## ================
## regime 3
## estimate:     -42.70595
## stderr:       0.4611302
## ================
```

```r
# regime 4
est4 = gcomp(data = ldl, regime = stat_reg4, K = 4, M = 1000)
sd4 = bootstrap_gcomp(data = ldl, regime = stat_reg4, K = 4, M = 1000, rep = 10)

cat("================\nregime 4\nestimate:\t", est4, "\nstderr:\t\t", sd4, "\n================")
```

```
## ================
## regime 4
## estimate:     -52.79949
## stderr:       0.7116331
## ================
```

```r
# regime 5
est5 = gcomp(data = ldl, regime = stat_reg5, K = 4, M = 1000)
sd5 = bootstrap_gcomp(data = ldl, regime = stat_reg5, K = 4, M = 1000, rep = 10)

cat("================\nregime 5\nestimate:\t", est5, "\nstderr:\t\t", sd5, "\n================")
```

```
## ================
## regime 5
## estimate:     -55.4591
## stderr:       0.534377
## ================
```

```r
# regime 6
est6 = gcomp(data = ldl, regime = stat_reg6, K = 4, M = 1000)
sd6 = bootstrap_gcomp(data = ldl, regime = stat_reg6, K = 4, M = 1000, rep = 10)

cat("================\nregime 6\nestimate:\t", est6, "\nstderr:\t\t", sd6, "\n================")
```

```
## ================
## regime 6
## estimate:     -41.75875
## stderr:       0.4142625
## ================
```

```r
# regime 7
est7 = gcomp(data = ldl, regime = stat_reg7, K = 4, M = 1000)
sd7 = bootstrap_gcomp(data = ldl, regime = stat_reg7, K = 4, M = 1000, rep = 10)

cat("================\nregime 7\nestimate:\t", est7, "\nstderr:\t\t", sd7, "\n================")
```

```
## ================
## regime 7
## estimate:     -30.73871
## stderr:       0.361967
## ================
```

```r
# regime 8
est8 = gcomp(data = ldl, regime = stat_reg8, K = 4, M = 1000)
sd8 = bootstrap_gcomp(data = ldl, regime = stat_reg8, K = 4, M = 1000, rep = 10)

cat("================\nregime 8\nestimate:\t", est8, "\nstderr:\t\t", sd8, "\n================")
```

```
## ================
## regime 8
## estimate:     -17.9668
## stderr:       0.3712015
## ================
```

d

i

```r
regime_d1 = function(L, S, A, dk){
  # only 0 if the patient is currently having a side effect
  return(!S[dk])
}

estd1 = gcomp(data = ldl, regime = regime_d1, K = 4, M = 1000)
sdd1 = bootstrap_gcomp(data = ldl, regime = regime_d1, K = 4, M = 1000, rep = 10)

cat("================\nregime d1\nestimate:\t", estd1, "\nstderr:\t\t", sdd1, "\n================")
```

```
## ================
## regime d1
## estimate:     -53.14432
## stderr:       0.7164968
## ================
```

ii

```r
regime_d2 = function(L, S, A, dk){
  # 0 if the patient has ever had a side effect
  return(!(1 %in% dk))
}

estd2 = gcomp(data = ldl, regime = regime_d2, K = 4, M = 1000)
sdd2 = bootstrap_gcomp(data = ldl, regime = regime_d2, K = 4, M = 1000, rep = 10)

cat("================\nregime d2\nestimate:\t", estd2, "\nstderr:\t\t", sdd2, "\n================")
```

```
## ================
## regime d2
## estimate:     -55.12239
## stderr:       0.5880577
## ================
```

e

```
etas = seq(90, 200, 10)

for(i in 1:length(etas)){
  eta_i = etas[i]

  regime_eta = function(L, S, A, dk){
    return(S[dk] == 0 && L[dk] > eta_i)
  }

  estd2 = gcomp(data = ldl, regime = regime_d2, K = 4, M = 1000)
  sdd2 = bootstrap_gcomp(data = ldl, regime = regime_d2, K = 4, M = 1000, rep = 10)

  cat("================\nregime eta=", eta_i,"\nestimate:\t", estd2, "\nstderr:\t\t", sdd2, "\n=========

}
```

```
## ================
## regime eta= 90
## estimate:     -53.53442
## stderr:       0.5690198
## ================================
## regime eta= 100
## estimate:     -54.70017
## stderr:       0.6022783
## ================================
## regime eta= 110
## estimate:     -54.69046
## stderr:       0.7102287
## ================================
## regime eta= 120
## estimate:     -53.85089
## stderr:       0.6175448
## ================================
## regime eta= 130
## estimate:     -54.71301
## stderr:       0.8334253
## ================================
## regime eta= 140
## estimate:     -54.62439
## stderr:       0.6739823
## ================================
## regime eta= 150
## estimate:     -54.31422
## stderr:       0.6002285
## ================================
## regime eta= 160
## estimate:     -54.55069
## stderr:       0.7815729
## ================================
## regime eta= 170
## estimate:     -54.53063
```

```
## stderr:       0.7082882
## ================================
## regime eta= 180
## estimate:     -53.69631
## stderr:       0.448302
## ================================
## regime eta= 190
## estimate:     -54.35442
## stderr:       1.055824
## ================================
## regime eta= 200
## estimate:     -53.95874
## stderr:       0.3838845
## ================
```

f

3

a

```r
calc_gamma = function(data){
  out = matrix(0, nrow=4, ncol=3)

  gamma1_mod = glm(A1 ~ L1, data, family = "binomial")
  # add extra 0 because other temrs has an S factor
  out[1,] = c(gamma1_mod$coefficients, 0)

  gamma2_mod = glm(A2 ~ L2 + S2, data, family = "binomial")
  out[2,] = gamma2_mod$coefficients

  gamma3_mod = glm(A3 ~ L3 + S3, data, family = "binomial")
  out[3,] = gamma3_mod$coefficients

  gamma4_mod = glm(A4 ~ L4 + S4, data, family = "binomial")
  out[4,] = gamma4_mod$coefficients

  return(out)
}

gammas = calc_gamma(ldl)
colnames(gammas) = c("gamma_k1", "gamma_k2", "gamma_k3")
row.names(gammas) = c("1", "2", "3", "4")

print(gammas)
```

```
##     gamma_k1    gamma_k2     gamma_k3
## 1 -0.9918005 0.006118285  0.000000000
## 2 -0.4655975 0.003045530 -0.090446238
## 3 -0.4550628 0.002432314 -0.187644615
## 4 -0.5344407 0.002083461 -0.001472349
```

b

```
# Cd vector for equation 5.27 on slide 304
calc_cd = function(data, regime, K)
{
  n = dim(data)[1]

  L = cbind(data$L1, data$L2, data$L3, data$L4, data$Y)
  # again need 0s becasue there are no side effects at the beginning
  S = cbind(rep(0, n), data$S2, data$S3, data$S4, data$S5)
  A = cbind(data$A1, data$A2, data$A3, data$A4)


  cd_vec = rep(1, n)

  for(i in 1:n){

    for(k in 1:K){
        cd_vec[i] = cd_vec[i] * ( A[i,k] == regime(L[i,k], S[i,k], A[i,k], k))

    }
  }
  return(cd_vec)
}


# equation 5.27 on slide 304
calc_ipw = function(data, regime, K){

  n = dim(data)[1]

  Y = data$Y
  L = cbind(data$L1, data$L2, data$L3, data$L4)
  # again need 0s becasue there are no side effects at the beginning
  S = cbind(rep(0, n), data$S2, data$S3, data$S4, data$S5)
  A = cbind(data$A1, data$A2, data$A3, data$A4)

  cd = calc_cd(data, regime, K)
  gamma = calc_gamma(data)

  ipw_est = 0

  for(i in 1:n){
    num = cd[i]

    # only need to calculate if Cd == 1
    if(cd[i]){

      # mulitply by Yi
      num = Y[i]

      denom = 1
      # calculate denominator
```

```r
      # the product of the propensities in equation 7
      for(k in 1:K){
        val = gamma[k, 1] + gamma[k, 2] * L[i,k] + gamma[k, 3] * S[i,k]
        p = logistic_func(val)
        dk = regime(L[i,], S[i,], A[i,], k)
        denom = denom*(dk * p + (1-dk)*(1-p))
      }

      ipw_est = ipw_est + num / denom

    } # end if
  }

  return(ipw_est/n)
}


bootstrap_ipw = function(data, regime, K, rep){
  out = NULL
  nrow = dim(data)[1]

  for(i in 1:rep){
    sample = data[sample(nrow, replace=TRUE),]
    ipw = calc_ipw(sample, regime, K)
    out = rbind(out, ipw)
  }
  return(sd(out))
}


# regime 1
ipw_est1 = calc_ipw(data = ldl, regime = stat_reg1, K = 4)
ipw_sd1 = bootstrap_ipw(data = ldl, regime = stat_reg1, K = 4, rep = 100)

cat("================\nregime 1\nipw_estimate:\t", ipw_est1, "\nstderr:\t\t", ipw_sd1, "\n=============
```

```
## ================
## regime 1
## ipw_estimate:     135.6029
## stderr:       6.354277
## ================
```

```r
# regime 2
ipw_est2 = calc_ipw(data = ldl, regime = stat_reg2, K = 4)
ipw_sd2 = bootstrap_ipw(data = ldl, regime = stat_reg2, K = 4, rep = 100)

cat("================\nregime 2\nipw_estimate:\t", ipw_est2, "\nstderr:\t\t", ipw_sd2, "\n=============
```

```
## ================
## regime 2
## ipw_estimate:     144.0354
## stderr:       6.444685
## ================
```

```r
# regime 3
ipw_est3 = calc_ipw(data = ldl, regime = stat_reg3, K = 4)
ipw_sd3 = bootstrap_ipw(data = ldl, regime = stat_reg3, K = 4, rep = 100)

cat("================\nregime 3\nipw_estimate:\t", ipw_est3, "\nstderr:\t\t", ipw_sd3, "\n============
```

```
## ================
## regime 3
## ipw_estimate:     135.4788
## stderr:       6.76914
## ================
```

```r
# regime 4
ipw_est4 = calc_ipw(data = ldl, regime = stat_reg4, K = 4)
ipw_sd4 = bootstrap_ipw(data = ldl, regime = stat_reg4, K = 4, rep = 100)

cat("================\nregime 4\nipw_estimate:\t", ipw_est4, "\nstderr:\t\t", ipw_sd4, "\n============
```

```
## ================
## regime 4
## ipw_estimate:     113.821
## stderr:       6.259131
## ================
```

```r
# regime 5
ipw_est5 = calc_ipw(data = ldl, regime = stat_reg5, K = 4)
ipw_sd5 = bootstrap_ipw(data = ldl, regime = stat_reg5, K = 4, rep = 100)

cat("================\nregime 5\nipw_estimate:\t", ipw_est5, "\nstderr:\t\t", ipw_sd5, "\n============
```

```
## ================
## regime 5
## ipw_estimate:     101.6646
## stderr:       5.823647
## ================
```

```r
# regime 6
ipw_est6 = calc_ipw(data = ldl, regime = stat_reg6, K = 4)
ipw_sd6 = bootstrap_ipw(data = ldl, regime = stat_reg6, K = 4, rep = 100)

cat("================\nregime 6\nipw_estimate:\t", ipw_est6, "\nstderr:\t\t", ipw_sd6, "\n============
```

```
## ================
## regime 6
## ipw_estimate:     113.033
## stderr:       5.347491
## ================
```

```r
# regime 7
ipw_est7 = calc_ipw(data = ldl, regime = stat_reg7, K = 4)
ipw_sd7 = bootstrap_ipw(data = ldl, regime = stat_reg7, K = 4, rep = 100)

cat("================\nregime 7\nipw_estimate:\t", ipw_est7, "\nstderr:\t\t", ipw_sd7, "\n============
```

```
## ================
## regime 7
## ipw_estimate:     133.8875
## stderr:       5.558936
## ================
```

```
# regime 8
ipw_est8 = calc_ipw(data = ldl, regime = stat_reg8, K = 4)
ipw_sd8 = bootstrap_ipw(data = ldl, regime = stat_reg8, K = 4, rep = 100)

cat("================\nregime 8\nipw_estimate:\t", ipw_est8, "\nstderr:\t\t", ipw_sd8, "\n=============
```

```
## ================
## regime 8
## ipw_estimate:     133.7209
## stderr:       5.043448
## ================
```

```
# d1
ipw_estd1 = calc_ipw(data = ldl, regime = regime_d2, K = 4)
ipw_sdd1 = bootstrap_ipw(data = ldl, regime = regime_d2, K = 4,rep = 100)

cat("================\nregime d1\nipw_estimate:\t", ipw_estd1, "\nstderr:\t\t", ipw_sdd1, "\n==========
```

```
## ================
## regime d1
## ipw_estimate:     113.821
## stderr:       5.748335
## ================
```

```
ipw_estd2 = calc_ipw(data = ldl, regime = regime_d2, K = 4)
ipw_sdd2 = bootstrap_ipw(data = ldl, regime = regime_d2, K = 4, rep = 10)

cat("================\nregime d2\nipw_estimate:\t", ipw_estd2, "\nstderr:\t\t", ipw_sdd2, "\n==========
```

```
## ================
## regime d2
## ipw_estimate:     113.821
## stderr:       6.935966
## ================
```

```
etas = seq(90, 200, 10)

for(i in 1:length(etas)){
  eta_i = etas[i]

  regime_eta = function(L, S, A, dk){
    return(S[dk] == 0 && L[dk] > eta_i)
  }

  ipw_estd2 = calc_ipw(data = ldl, regime = regime_d2, K = 4)
```

```r
  ipw_sdd2 = bootstrap_ipw(data = ldl, regime = regime_d2, K = 4, rep = 100)

  cat("================\nregime eta=", eta_i,"\nipw_estimate:\t", ipw_estd2, "\nstderr:\t\t", ipw_sdd2,
}
```

```
## ================
## regime eta= 90
## ipw_estimate:     113.821
## stderr:       6.571104
## ================================
## regime eta= 100
## ipw_estimate:     113.821
## stderr:       6.023239
## ================================
## regime eta= 110
## ipw_estimate:     113.821
## stderr:       5.896799
## ================================
## regime eta= 120
## ipw_estimate:     113.821
## stderr:       6.361984
## ================================
## regime eta= 130
## ipw_estimate:     113.821
## stderr:       5.454488
## ================================
## regime eta= 140
## ipw_estimate:     113.821
## stderr:       5.697975
## ================================
## regime eta= 150
## ipw_estimate:     113.821
## stderr:       6.210356
## ================================
## regime eta= 160
## ipw_estimate:     113.821
## stderr:       5.371317
## ================================
## regime eta= 170
## ipw_estimate:     113.821
## stderr:       6.448903
## ================================
## regime eta= 180
## ipw_estimate:     113.821
## stderr:       6.08682
## ================================
## regime eta= 190
## ipw_estimate:     113.821
## stderr:       5.9452
## ================================
## regime eta= 200
## ipw_estimate:     113.821
```

```
## stderr:       6.288679
## ================
```

c

```r
# equation 5.33 on slide 314
calc_ipw_star = function(data, regime, K){
  n = dim(data)[1]

  Y = data$Y
  L = cbind(data$L1, data$L2, data$L3, data$L4)
  # again need 0s becasue there are no side effects at the beginning
  S = cbind(rep(0, n), data$S2, data$S3, data$S4, data$S5)
  A = cbind(data$A1, data$A2, data$A3, data$A4)

  cd = calc_cd(data, regime, K)
  gamma = calc_gamma(data)

  sum1 = 0
  sum2 = 0

  for(i in 1:n){
    num = cd[i]

    # only need to calculate if Cd == 1
    if(cd[i]){

      denom = 1
      # calculate denominator
      # the product of the propensities in equation 7
      for(k in 1:K){
        val = gamma[k, 1] + gamma[k, 2] * L[i,k] + gamma[k, 3] * S[i,k]
        p = logistic_func(val)
        dk = regime(L[i,], S[i,], A[i,], k)
        denom = denom*(dk * p + (1-dk)*(1-p))
      }

      sum1 = sum1 + cd[i] / denom
      sum2 = sum2 + cd[i] * Y[i] / denom
    } # end if
  }

  return(sum2 / sum1)
}

bootstrap_ipw_star = function(data, regime, K, rep){
  out = NULL
  nrow = dim(data)[1]

  for(i in 1:rep){
    sample = data[sample(nrow, replace=TRUE),]
    ipw_star = calc_ipw_star(sample, regime, K)
```

```r
    out = rbind(out, ipw_star)
  }
  return(sd(out))
}


# regime 1
ipw_star_est1 = calc_ipw_star(data = ldl, regime = stat_reg1, K = 4)
ipw_star_sd1 = bootstrap_ipw_star(data = ldl, regime = stat_reg1, K = 4, rep = 100)

cat("================\nregime 1\nipw_star_estimate:\t", ipw_star_est1, "\nstderr:\t\t\t", ipw_star_sd1,
```

```
## ================
## regime 1
## ipw_star_estimate:    143.8031
## stderr:           1.731405
## ================
```

```r
# regime 2
ipw_star_est2 = calc_ipw_star(data = ldl, regime = stat_reg2, K = 4)
ipw_star_sd2 = bootstrap_ipw_star(data = ldl, regime = stat_reg2, K = 4, rep = 100)

cat("================\nregime 2\nipw_star_estimate:\t", ipw_star_est2, "\nstderr:\t\t\t", ipw_star_sd2,
```

```
## ================
## regime 2
## ipw_star_estimate:    135.6302
## stderr:           1.516484
## ================
```

```r
# regime 3
ipw_star_est3 = calc_ipw_star(data = ldl, regime = stat_reg3, K = 4)
ipw_star_sd3 = bootstrap_ipw_star(data = ldl, regime = stat_reg3, K = 4, rep = 100)

cat("================\nregime 3\nipw_star_estimate:\t", ipw_star_est3, "\nstderr:\t\t\t", ipw_star_sd3,
```

```
## ================
## regime 3
## ipw_star_estimate:    124.2947
## stderr:           1.588352
## ================
```

```r
# regime 4
ipw_star_est4 = calc_ipw_star(data = ldl, regime = stat_reg4, K = 4)
ipw_star_sd4 = bootstrap_ipw_star(data = ldl, regime = stat_reg4, K = 4, rep = 100)

cat("================\nregime 4\nipw_star_estimate:\t", ipw_star_est4, "\nstderr:\t\t\t", ipw_star_sd4,
```

```
## ================
## regime 4
## ipw_star_estimate:    115.0639
## stderr:           1.518395
## ================
```

```r
# regime 5
ipw_star_est5 = calc_ipw_star(data = ldl, regime = stat_reg5, K = 4)
ipw_star_sd5 = bootstrap_ipw_star(data = ldl, regime = stat_reg5, K = 4, rep = 100)

cat("================\nregime 5\nipw_star_estimate:\t", ipw_star_est5, "\nstderr:\t\t\t", ipw_star_sd5,
```

```
## ================
## regime 5
## ipw_star_estimate:    106.0979
## stderr:            1.691251
## ================
```

```r
# regime 6
ipw_star_est6 = calc_ipw_star(data = ldl, regime = stat_reg6, K = 4)
ipw_star_sd6 = bootstrap_ipw_star(data = ldl, regime = stat_reg6, K = 4, rep = 100)

cat("================\nregime 6\nipw_star_estimate:\t", ipw_star_est6, "\nstderr:\t\t\t", ipw_star_sd6,
```

```
## ================
## regime 6
## ipw_star_estimate:    113.8119
## stderr:            1.22207
## ================
```

```r
# regime 7
ipw_star_est7 = calc_ipw_star(data = ldl, regime = stat_reg7, K = 4)
ipw_star_sd7 = bootstrap_ipw_star(data = ldl, regime = stat_reg7, K = 4, rep = 100)

cat("================\nregime 7\nipw_star_estimate:\t", ipw_star_est7, "\nstderr:\t\t\t", ipw_star_sd7,
```

```
## ================
## regime 7
## ipw_star_estimate:    123.1004
## stderr:            1.18902
## ================
```

```r
# regime 8
ipw_star_est8 = calc_ipw_star(data = ldl, regime = stat_reg8, K = 4)
ipw_star_sd8 = bootstrap_ipw_star(data = ldl, regime = stat_reg8, K = 4, rep = 100)

cat("================\nregime 8\nipw_star_estimate:\t", ipw_star_est8, "\nstderr:\t\t\t", ipw_star_sd8,
```

```
## ================
## regime 8
## ipw_star_estimate:    133.9061
## stderr:            1.193371
## ================
```

```r
# d1
ipw_star_estd1 = calc_ipw_star(data = ldl, regime = regime_d2, K = 4)
ipw_star_sdd1 = bootstrap_ipw_star(data = ldl, regime = regime_d2, K = 4,rep = 100)

cat("================\nregime d1\nipw_star_estimate:\t", ipw_star_estd1, "\nstderr:\t\t\t", ipw_star_sd
```

```
## ================
## regime d1
## ipw_star_estimate:    115.0639
## stderr:           1.643683
## ================
```

```
ipw_star_estd2 = calc_ipw_star(data = ldl, regime = regime_d2, K = 4)
ipw_star_sdd2 = bootstrap_ipw_star(data = ldl, regime = regime_d2, K = 4, rep = 10)

cat("================\nregime d2\nipw_star_estimate:\t", ipw_star_estd2, "\nstderr:\t\t\t", ipw_star_sd
```

```
## ================
## regime d2
## ipw_star_estimate:    115.0639
## stderr:           0.8493705
## ================
```

```
etas = seq(90, 200, 10)

for(i in 1:length(etas)){
  eta_i = etas[i]

  regime_eta = function(L, S, A, dk){
    return(S[dk] == 0 && L[dk] > eta_i)
  }

  ipw_star_estd2 = calc_ipw_star(data = ldl, regime = regime_d2, K = 4)
  ipw_star_sdd2 = bootstrap_ipw_star(data = ldl, regime = regime_d2, K = 4, rep = 100)

  cat("================\nregime eta=", eta_i,"\nipw_star_estimate:\t", ipw_star_estd2, "\nstderr:\t\t\t
}
```

```
## ================
## regime eta= 90
## ipw_star_estimate:    115.0639
## stderr:           1.542405
## ================================
## regime eta= 100
## ipw_star_estimate:    115.0639
## stderr:           1.767288
## ================================
## regime eta= 110
## ipw_star_estimate:    115.0639
## stderr:           1.714325
## ================================
## regime eta= 120
## ipw_star_estimate:    115.0639
## stderr:           1.568568
## ================================
## regime eta= 130
## ipw_star_estimate:    115.0639
## stderr:           1.684143
## ================================
```

```
## regime eta= 140
## ipw_star_estimate:    115.0639
## stderr:           1.531586
## ================================
## regime eta= 150
## ipw_star_estimate:    115.0639
## stderr:           1.762022
## ================================
## regime eta= 160
## ipw_star_estimate:    115.0639
## stderr:           1.657728
## ================================
## regime eta= 170
## ipw_star_estimate:    115.0639
## stderr:           1.658034
## ================================
## regime eta= 180
## ipw_star_estimate:    115.0639
## stderr:           1.641979
## ================================
## regime eta= 190
## ipw_star_estimate:    115.0639
## stderr:           1.634786
## ================================
## regime eta= 200
## ipw_star_estimate:    115.0639
## stderr:           1.622516
## ================
```

d

Further, the IPW and IPW* code ran significantly faster! Perhaps it is due to inefficiencies in my data definition matrix, but I had to limit the bootstrap to 10 repetitions each for gcomputation because 100 took too long to run.