Stat331FinalProject

Yaolun Yin, Yanyi Wang

April 15, 2020

1 Summary

The object of this report is to investigate which explanatory variables have impact on CHD risk scores. We first explore the data set by printing summary, pair plot and VIF scores of explanatory variables. Then, we generated 2 candidate models by automating and manual selection. These two models are both diagnosed by different types of residual plots, leverage, and influence measures. Finally, we pick one model by performing cross-validation.

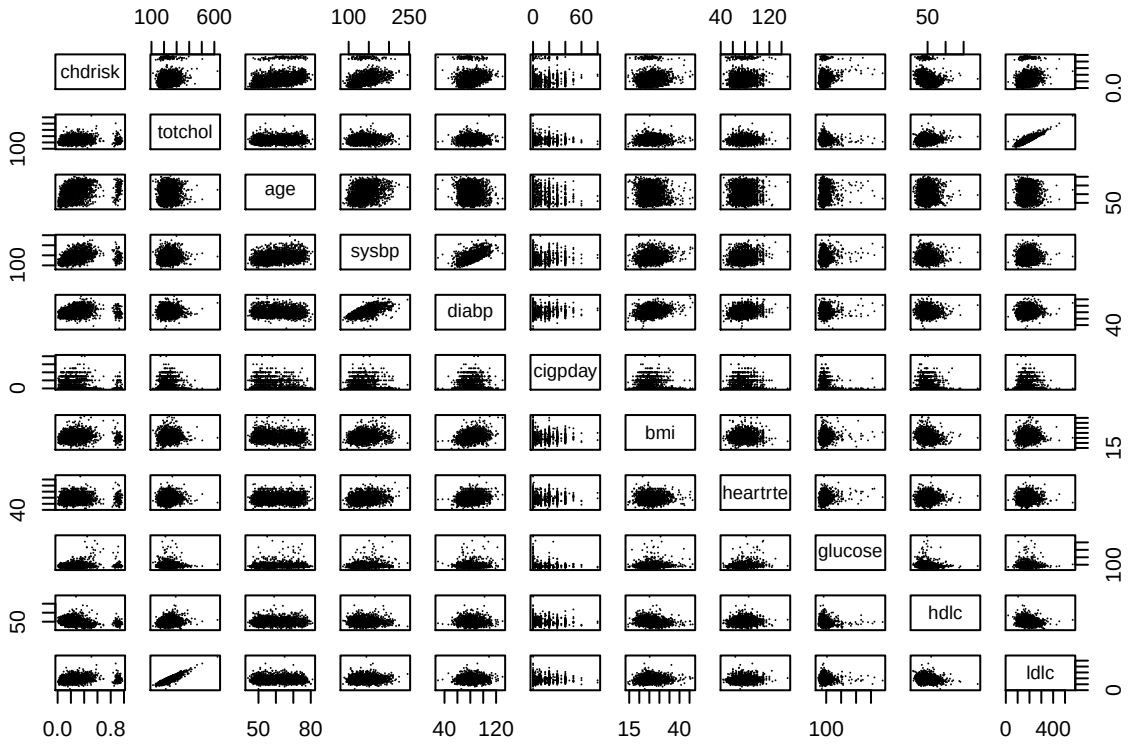
2 Descriptive Analysis

```
# summary of fhsd
summary(fhsd$chdrisk)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0050  0.1320  0.2240  0.2655  0.3448  0.9770
```

The mean of chdrisk is 0.2655 and 3/4 of the data is smaller than 0.35. Thus, we expect most normal values with some outliers.

```
# display the pair plots of continuous variables
pairs(~chdrisk + totchol + age + sysbp + diabp
      + cigpday + bmi + heartrte + glucose+hdlc+ldlc,
      data = fhsd, pch=16, cex = 0.2)
```



From the pair plot, we cannot find a linear relationship between CHD risk (chdrisk) and another continuous variable.

The variable 'totchol' and 'ldlc' have VIF > 10, so we removed them.

3 Candidate Models

Automated Selection:

```
if (!params$load_pairs) {
  # forward selection
  Mfwd <- step(object = M0, # starting point model
  scope = list(lower = M0, upper = Mmax), # smallest and largest model
  direction = "forward",
  trace = FALSE)

  # backward elimination
  Mback <- step(object = Mmax, # starting point model
                scope = list(lower = M0, upper = Mmax),
                direction = "backward", trace = FALSE)

  # stepwise calculation
  Mstep <- step(object = Mstart,
              scope = list(lower = M0, upper = Mmax),
              direction = "both", trace = FALSE)

  saveRDS(list(Mfwd = Mfwd, Mback = Mback, Mstep = Mstep), file = "lm_back.rds")
} else {
  tmp <- readRDS("lm_back.rds")
  Mfwd <- tmp$Mfwd
  Mback <- tmp$Mback
```

```
  Mstep <- tmp$Mstep
  rm(tmp) # optionally remove tmp from workspace
}
```

```
# compare 3 models using AIC method
AICscore <- c(AIC(Mfwd),AIC(Mback),AIC(Mstep))
names(AICscore) <- c("Mfwd","Mback","Mstep")
AICscore
```

```
##     Mfwd    Mback    Mstep
## 3481.231 3466.234 3468.200
```

AICscores: Mback < Mstep < Mfwd. We can choose Mstep because it has a smaller AIC value than Mfwd. Mback and Mstep have similar AIC values, but Mback has slower running time than Mstep. Thus, Mstep is the best choice among 3 models.

Next, we can construct the second candidate model. We start by writing the sum of all main effect and non-linear effect.

```
#model with only main effect and non-linear effect
Mman <- lm(formula = log(chdrisk) - log(1 - chdrisk) ~ sex + age +
             sysbp + cursmoke + diabp + diabetes + prevmi + prevhyp +
             prevstrk + glucose + cigpday + I(hdlc^2) + I(diabp^2) +
             I(bmi^2) + I(heartrte^2) + I(glucose^2), data=fhsdm)
# do the f-test between Mman and Mstep
anova(Mman,Mstep)
```

```
## Analysis of Variance Table
##
## Model 1: log(chdrisk) - log(1 - chdrisk) ~ sex + age + sysbp + cursmoke +
##     diabp + diabetes + prevmi + prevhyp + prevstrk + glucose +
##     cigpday + I(hdlc^2) + I(diabp^2) + I(bmi^2) + I(heartrte^2) +
##     I(glucose^2)
## Model 2: log(chdrisk) - log(1 - chdrisk) ~ sex + age + sysbp + diabp +
##     cursmoke + cigpday + bmi + diabetes + bpmeds + heartrte +
##     glucose + prevmi + prevstrk + prevhyp + hdlc + I(hdlc^2) +
##     I(diabp^2) + I(bmi^2) + I(heartrte^2) + I(glucose^2) + sysbp:prevmi +
##     diabetes:prevmi + sysbp:prevhyp + diabp:hdlc + sysbp:diabetes +
##     prevmi:hdlc + age:heartrte + diabp:bmi + sex:glucose + prevhyp:hdlc +
##     diabp:heartrte + age:prevhyp + cigpday:hdlc + bmi:prevhyp +
##     age:cursmoke + diabp:glucose + prevmi:prevhyp + prevmi:prevstrk +
##     diabp:prevhyp + diabp:cursmoke + age:bmi + sex:age + sysbp:bpmeds +
##     diabetes:hdlc + age:hdlc + cigpday:glucose + bpmeds:hdlc +
##     cigpday:heartrte + cursmoke:hdlc + heartrte:hdlc + cursmoke:bpmeds +
##     diabetes:glucose + bmi:prevmi
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   2289 758.54
## 2   2252 579.22 37    179.32 18.843 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on the anova table, F value is huge. Thus, there is a strong evidence against null hypothesis. The Mman and Mstep have significant difference. We need to improve this model by adding some interaction effect with p-value < 0.05.

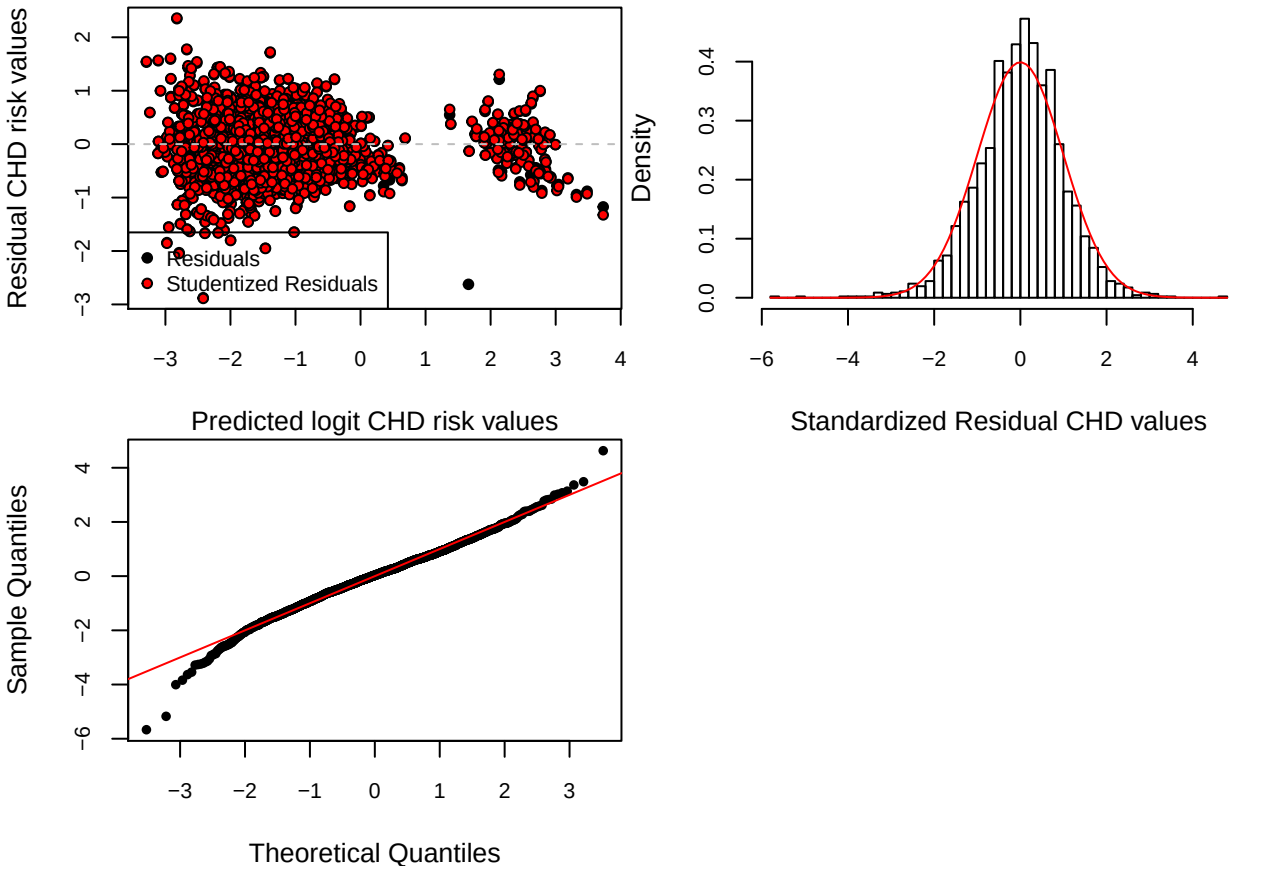
```
# model with only main effect and non-linear effect + Interaction
Mman <- lm(formula = log(chdrisk) - log(1 - chdrisk) ~ sex + age +
```

```
              sysbp + cursmoke + diabp + diabetes + prevmi + prevhyp +
              prevstrk + glucose + cigpday + I(hdlc^2) + I(diabp^2) +
              I(bmi^2) + I(heartrte^2) + I(glucose^2) + sysbp:prevmi +
              diabetes:prevmi + sysbp:prevhyp + diabp:hdlc + sysbp:diabetes +
              prevmi:hdlc + prevhyp:hdlc + diabp:heartrte + age:prevhyp
              + cigpday:hdlc + bmi:prevhyp + diabp:cursmoke +
              diabp:glucose + prevmi:prevhyp + diabp:prevhyp +
              cigpday:glucose,
data = fhsdm)
```

4 Model Diagnostics

Mstep diagnostic:

```
par(mfrow = c(2,2), mar= c(4,4,.1,.1))
# plotting residual vs fitted
threePlots(Mstep)
```



Checking assumptions:

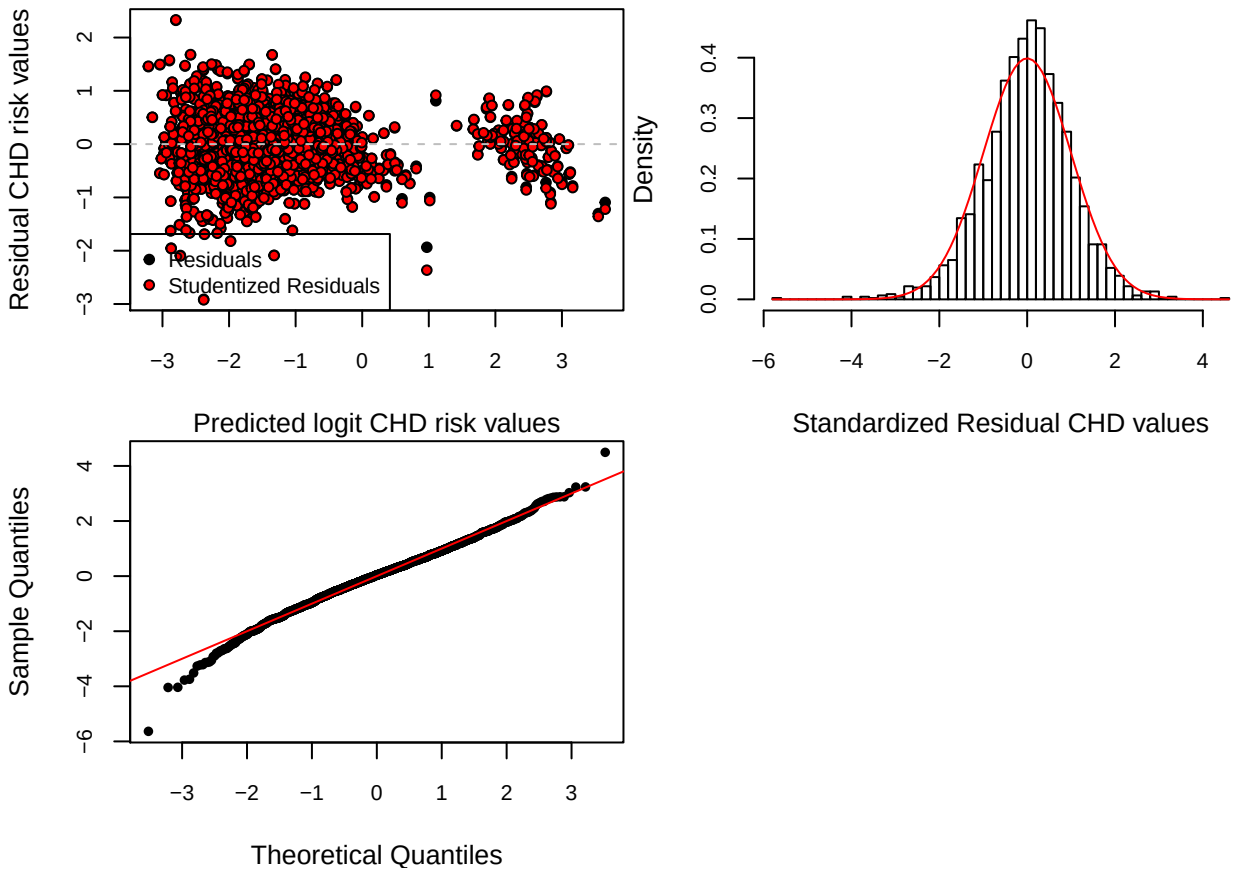
1. Zero mean: The model is satisfactory because all the observations are uniformly distributed near the 0 line.
2. Constant Variance: The model is satisfactory because the variance of points are bounded within [-2,2] except for 2 obervations.
3. Histogram: The model is satisfactory because the residual follows standard normal distribution. This is

because the curve fits the bell shape.

4. Normality: the model is satisfactory because the points lie more or less along a straight line in the qqplot.

Mman diagnostic:

```
par(mfrow=c(2,2),mar=c(4,4,.1,.1))
# plotting residual
threePlots(Mman)
```



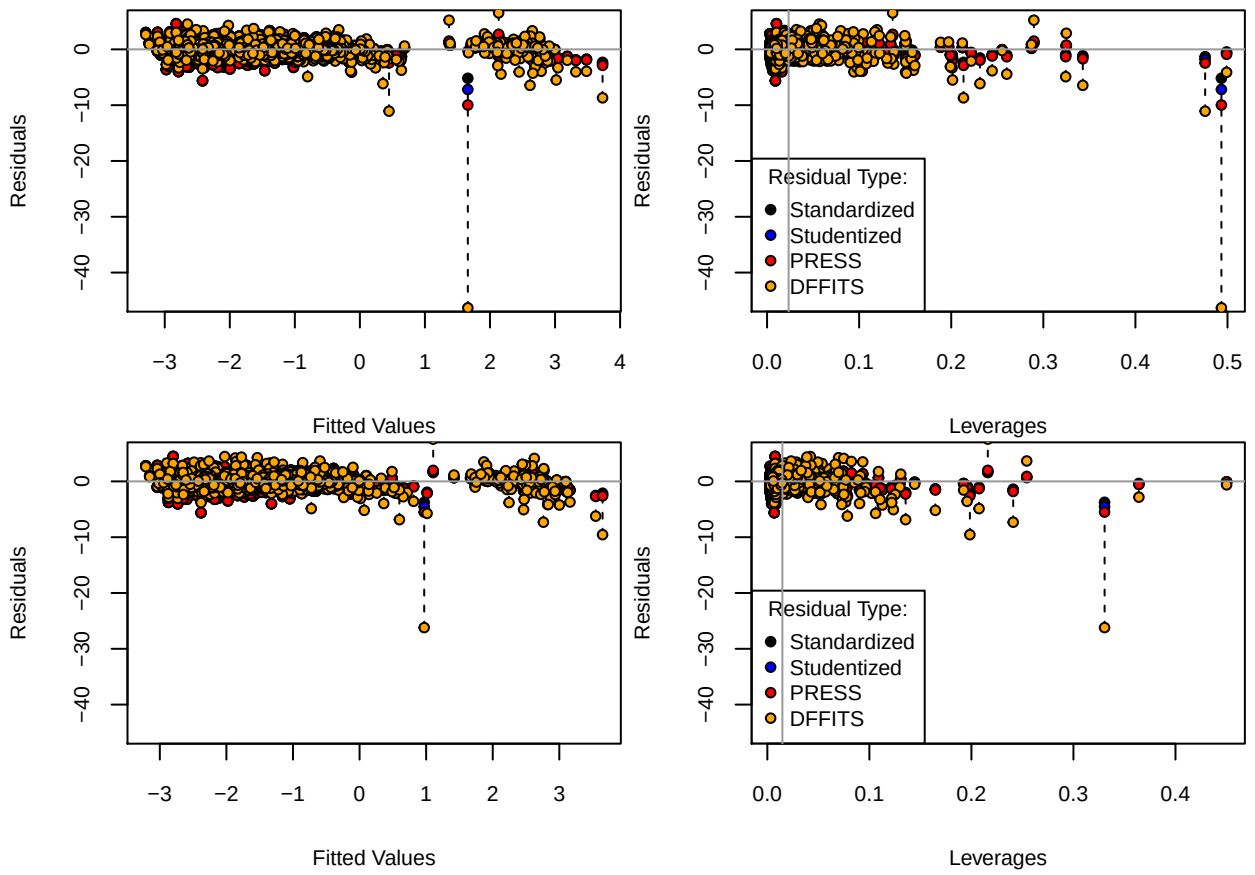
Checking assumptions:

1. Zero mean: The model is satisfactory because all the observations are uniformly distributed near the 0 line.
2. Constant Variance: The model is satisfactory because the variance of points are bounded within [-2,2] except for 2 obervations.
3. Histogram: The model is satisfactory because the residual follows standard normal distribution. This is because the curve fits the bell shape.
4. Normality: the model is satisfactory because the points lie more or less along a straight line in the qqplot.

5 Leverage and Influence

```
par(mfrow = c(2,2), mar = c(4,4,.1,.1))
# plotting leverage for Mstep
```

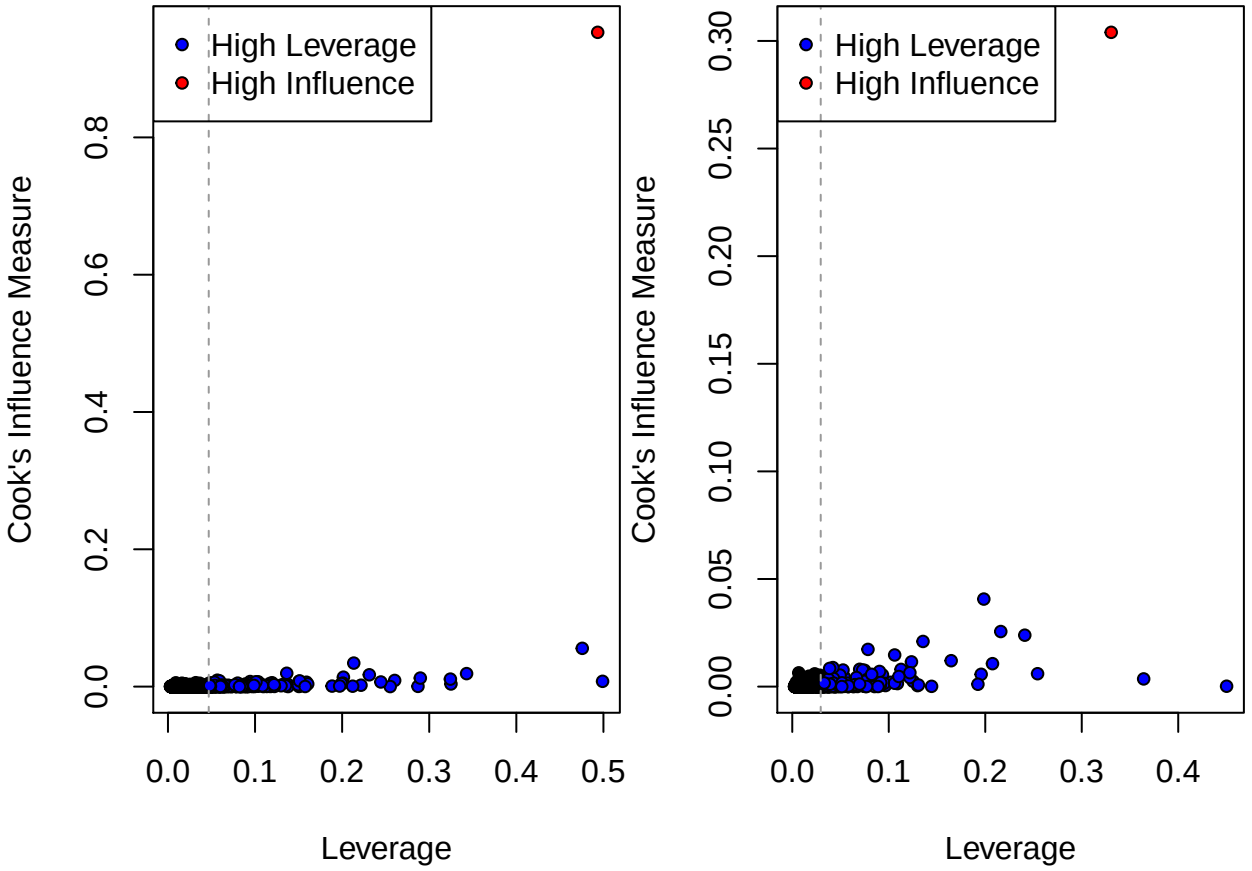
```
leveragePlot(Mstep)
# plotting leverage for Mman
leveragePlot(Mman)
```



Note: The first row is Mstep

We used 4 types of residuals, which are standardized, studentized, PRESS and DFFITS. Although there exist some outliers, we cannot see much difference between the first and the second row. Therefore, we cannot decide which model would be better.

```
par(mfrow = c(1,2), mar=c(4,4,.1,.1))
# plotting cook's distance influence measure
cookDisPlot(Mstep)
# plotting cook's distance influence measure
cookDisPlot(Mman)
```



Note: the left plot is for Mstep.

For Mstep, one of the observations is almost 8 times the Cook's distance of the others(in red) and many observations have high leverage(in blue).

For Mman, one of the observations is almost 3 times the Cook's distance of the others(in red) and many observations have high leverage(in blue).

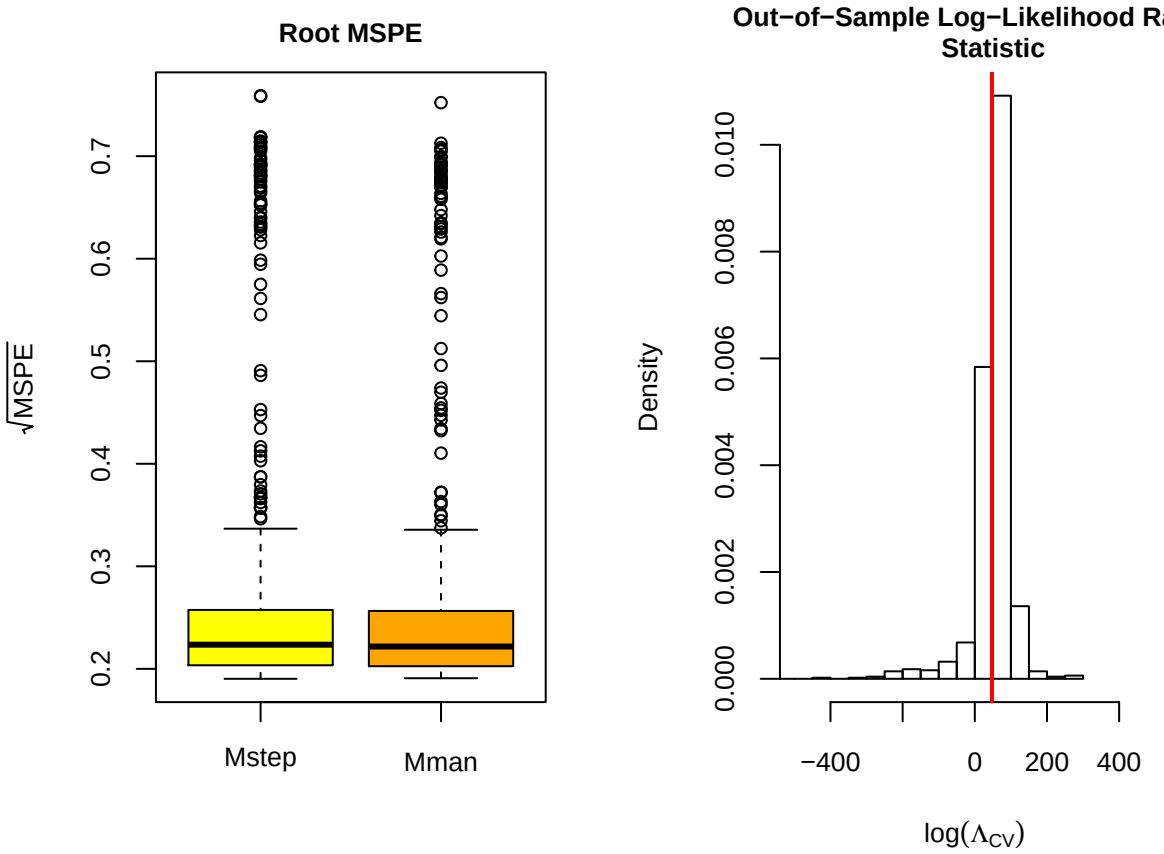
We cannot pick a model by comparing the Cook's distance

6 Model Selection

Perform cross-validation between Mstep and Mman:

```
# compare
par(mfrow = c(1,2), mar = c(4, 4, 2.1, 2.1))
cex <- .8
boxplot(x = list(rmspe1, rmspe2), names = Mnames,
        main = "Root MSPE",
        ylab = expression(sqrt(MSPE)),
        ## ylab = expression(SSE[CV]),
        col = c("yellow", "orange"),
        cex = cex, cex.lab = cex, cex.axis = cex, cex.main = cex)
lambda <- lambda1 - lambda2
hist(lambda, breaks = 50, freq = FALSE,
     main = "Out-of-Sample Log-Likelihood Ratio
     Statistic",
     xlab = expression(log(Lambda[CV])),
```

```
     xlim = c(-500,500),
     cex = cex, cex.lab = cex, cex.axis = cex, cex.main = cex)
abline(v = mean(lambda), col = "red", lwd = 2)
```



From the box plot, the 2 models have similar rMSPE. However, Mstep has a much higher likelihood according to out-of-sample MLE. Therefore, we choose Mstep as our final answer.

```
# The final answer is Mstep
summary(Mstep)
```

```
##
## Call:
## lm(formula = log(chdrisk) - log(1 - chdrisk) ~ sex + age + sysbp +
##     diabp + cursmoke + cigpday + bmi + diabetes + bpmeds + heartrte +
##     glucose + prevmi + prevstrk + prevhyp + hdlc + I(hdlc^2) +
##     I(diabp^2) + I(bmi^2) + I(heartrte^2) + I(glucose^2) + sysbp:prevmi +
##     diabetes:prevmi + sysbp:prevhyp + diabp:hdlc + sysbp:diabetes +
##     prevmi:hdlc + age:heartrte + diabp:bmi + sex:glucose + prevhyp:hdlc +
##     diabp:heartrte + age:prevhyp + cigpday:hdlc + bmi:prevhyp +
##     age:cursmoke + diabp:glucose + prevmi:prevhyp + prevmi:prevstrk +
##     diabp:prevhyp + diabp:cursmoke + age:bmi + sex:age + sysbp:bpmeds +
##     diabetes:hdlc + age:hdlc + cigpday:glucose + bpmeds:hdlc +
##     cigpday:heartrte + cursmoke:hdlc + heartrte:hdlc + cursmoke:bpmeds +
##     diabetes:glucose + bmi:prevmi, data = fhsdm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -2.8744 -0.2950  0.0156  0.3216  2.3473
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -6.390e+00  1.232e+00  -5.189 2.31e-07 ***
## sexMale                9.495e-01  1.763e-01   5.386 7.94e-08 ***
## age                    6.480e-02  1.404e-02   4.616 4.12e-06 ***
## sysbp                  1.541e-02  1.815e-03   8.491  < 2e-16 ***
## diabp                 -3.639e-02  1.190e-02  -3.059 0.002248 **
## cursmokeYes            7.755e-01  2.941e-01   2.637 0.008422 **
## cigpday                1.713e-02  8.413e-03   2.036 0.041904 *
## bmi                   -8.153e-02  3.971e-02  -2.053 0.040141 *
## diabetesYes            1.262e+00  3.393e-01   3.719 0.000205 ***
## bpmedsYes              5.103e-01  2.696e-01   1.893 0.058505 .
## heartrte               1.940e-02  1.100e-02   1.764 0.077825 .
## glucose                4.773e-03  2.284e-03   2.090 0.036739 *
## prevmiYes              5.133e+00  4.730e-01  10.851  < 2e-16 ***
## prevstrkYes            2.208e-01  8.540e-02   2.585 0.009806 **
## prevhypYes             3.922e+00  4.119e-01   9.522  < 2e-16 ***
## hdlc                  -2.520e-03  9.681e-03  -0.260 0.794653
## I(hdlc^2)              2.984e-04  2.013e-05  14.822  < 2e-16 ***
## I(diabp^2)             6.381e-04  7.142e-05   8.934  < 2e-16 ***
## I(bmi^2)               2.614e-03  4.701e-04   5.561 3.00e-08 ***
## I(heartrte^2)          1.741e-04  4.754e-05   3.662 0.000256 ***
## I(glucose^2)           1.289e-05  4.586e-06   2.810 0.004994 **
## sysbp:prevmiYes       -1.217e-02  2.704e-03  -4.500 7.15e-06 ***
## diabetesYes:prevmiYes -7.250e-01  1.432e-01  -5.064 4.44e-07 ***
## sysbp:prevhypYes      -8.309e-03  2.081e-03  -3.993 6.72e-05 ***
## diabp:hdlc            -2.145e-04  7.367e-05  -2.911 0.003637 **
## sysbp:diabetesYes     -5.970e-03  1.816e-03  -3.288 0.001025 **
## prevmiYes:hdlc         1.632e-02  4.067e-03   4.013 6.20e-05 ***
## age:heartrte          -3.043e-04  1.058e-04  -2.876 0.004063 **
## diabp:bmi             -8.035e-04  2.897e-04  -2.773 0.005596 **
## sexMale:glucose       -1.922e-03  7.472e-04  -2.572 0.010161 *
## prevhypYes:hdlc       -6.220e-03  1.886e-03  -3.298 0.000988 ***
## diabp:heartrte        -2.682e-04  8.138e-05  -3.295 0.000999 ***
## age:prevhypYes        -1.492e-02  3.247e-03  -4.595 4.57e-06 ***
## cigpday:hdlc          -3.521e-04  1.024e-04  -3.440 0.000592 ***
## bmi:prevhypYes        -2.178e-02  7.718e-03  -2.821 0.004823 **
## age:cursmokeYes       -7.519e-03  3.061e-03  -2.456 0.014112 *
## diabp:glucose         -6.419e-05  2.749e-05  -2.335 0.019608 *
## prevmiYes:prevhypYes  -2.733e-01  1.392e-01  -1.963 0.049764 *
## prevmiYes:prevstrkYes -4.061e-01  2.101e-01  -1.932 0.053432 .
## diabp:prevhypYes      -9.506e-03  3.733e-03  -2.547 0.010938 *
## diabp:cursmokeYes     -5.885e-03  2.200e-03  -2.675 0.007522 **
## age:bmi                6.347e-04  3.722e-04   1.705 0.088256 .
## sexMale:age           -6.724e-03  2.779e-03  -2.420 0.015613 *
## sysbp:bpmedsYes       -3.440e-03  1.586e-03  -2.169 0.030181 *
## diabetesYes:hdlc       4.885e-03  2.680e-03   1.823 0.068494 .
## age:hdlc              -1.901e-04  9.977e-05  -1.905 0.056854 .
## cigpday:glucose       -8.082e-05  4.045e-05  -1.998 0.045806 *
## bpmedsYes:hdlc         3.514e-03  2.090e-03   1.681 0.092807 .
## cigpday:heartrte       1.305e-04  8.239e-05   1.584 0.113374
## cursmokeYes:hdlc       4.132e-03  2.503e-03   1.651 0.098891 .
```

```
## heartrte:hdlc         -1.006e-04  6.026e-05  -1.670 0.095056 .
## cursmokeYes:bpmedsYes -1.100e-01  7.244e-02  -1.518 0.129183
## diabetesYes:glucose   -1.945e-03  1.332e-03  -1.460 0.144428
## bmi:prevmiYes         -1.851e-02  1.323e-02  -1.399 0.161850
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5072 on 2252 degrees of freedom
## Multiple R-squared:  0.8236, Adjusted R-squared:  0.8195
## F-statistic: 198.4 on 53 and 2252 DF,  p-value: < 2.2e-16
```

7 Discussion

The most important factors associated with high CHD risk are Systolic blood pressure, glucose, smoking, diabetes, and high density lipoprotein cholesterol. The factors for low CHD risk are Diastolic blood pressure.

The recommendations are eating less lipoprotein cholesterol, drinking less and going on a diet to keep the blood pressure at a normal level. In addition, it is important to eat less sugar to lower the casual serum glucose. For smokers, they should consume fewer cigarettes.

There are some variables with high p-values but still retained in the final model. For example, heartrate has p-value 0.08. This may be caused by the data. In this data set, the patients with high CHD risk happen to have fast heart rate. Another reason is interaction effect. For example, high density lipoprotein cholesterol and smokers are a significant factors, but the interaction between them may not be important.

There is one observation with high influence labelled in red. We believe that it is an outlier and therefore can be removed.

One possible deficency is that our model has some variables with high p-value. These variables may cause some noise in prediction.

8 Appendix

```
# read the data
fhsd <- read.csv("~/Desktop/UW/3B/stat331/project/fhs.csv")
```

```
# load the external R script containing code chunks
# include means the code gets run but never displayed
#knitr::read_chunk("~/Desktop/UW/3B/stat331/project/external_code.R")
```

```
# use parames
rmarkdown::render("~/Desktop/UW/3B/stat331/project/Stat331FinalProject.Rmd",
                  params = list(load_pairs = FALSE))
```

```
# summary of fhsd
summary(fhsd$chdrisk)

# display the pair plots of continuous variables
pairs(~chdrisk + totchol + age + sysbp + diabp
      + cigpday + bmi + heartrte + glucose+hdlc+ldlc,
      data = fhsd, pch=16, cex = 0.2)

#calculate VIF using correlation matrix
X <- model.matrix(lm(chdrisk ~ . - 1 - sex - cursmoke -
                       diabetes - bpmeds - prevmi - prevstrk - prevhyp, data = fhsd))

# correlation matrix
C <- cor(X)

# calculate the VIF
vif <- diag(solve(C))
vif
```

```
# pick the significant variables.
drops <- c("totchol","ldlc")
fhsdm <- fhsd[,!(names(fhsd) %in% drops)]

# Model 1: automated model selection
# define the maximal model. We removed the variables mentioned in the previous section.
Mmax <- lm(log(chdrisk) - log(1-chdrisk) ~ (.)^2, data = fhsdm)

# check if there is any na
beta.max <- coef(Mmax)
names(beta.max)[is.na(beta.max)]

# new max model(add nonlinear effect)
Mmax <- lm(log(chdrisk) - log(1-chdrisk) ~ (.)^2 + I(age^2) + I(diabp^2) + I(bmi^2) +
             I(heartrte^2) + I(glucose^2) + I(hdlc^2) - cursmoke:cigpday -
             bpmeds:prevhyp, data = fhsdm)

# check the NA Mmax
anyNA(coef(Mmax))

# Method 1: automated selection
# new M0
M0 <- lm(log(chdrisk) - log(1-chdrisk) ~ 1, data = fhsdm)
```

```r
# new Mstart
Mstart <- lm(log(chdrisk) - log(1-chdrisk) ~ ., data = fhsdm)

if (!params$load_pairs) {
  # forward selection
  Mfwd <- step(object = M0, # starting point model
  scope = list(lower = M0, upper = Mmax), # smallest and largest model
  direction = "forward",
  trace = FALSE)

  # backward elimination
  Mback <- step(object = Mmax, # starting point model
                scope = list(lower = M0, upper = Mmax),
                direction = "backward", trace = FALSE)

  # stepwise calculation
  Mstep <- step(object = Mstart,
              scope = list(lower = M0, upper = Mmax),
              direction = "both", trace = FALSE)

  saveRDS(list(Mfwd = Mfwd, Mback = Mback, Mstep = Mstep), file = "lm_back.rds")
} else {
  tmp <- readRDS("lm_back.rds")
  Mfwd <- tmp$Mfwd
  Mback <- tmp$Mback
  Mstep <- tmp$Mstep
  rm(tmp) # optionally remove tmp from workspace
}

# compare 3 models using AIC method
AICscore <- c(AIC(Mfwd),AIC(Mback),AIC(Mstep))
names(AICscore) <- c("Mfwd","Mback","Mstep")
AICscore

#model with only main effect and non-linear effect
Mman <- lm(formula = log(chdrisk) - log(1 - chdrisk) ~ sex + age +
             sysbp + cursmoke + diabp + diabetes + prevmi + prevhyp +
             prevstrk + glucose + cigpday + I(hdlc^2) + I(diabp^2) +
             I(bmi^2) + I(heartrte^2) + I(glucose^2), data=fhsdm)
# do the f-test between Mman and Mstep
anova(Mman,Mstep)

# model with only main effect and non-linear effect + Interaction
Mman <- lm(formula = log(chdrisk) - log(1 - chdrisk) ~ sex + age +
             sysbp + cursmoke + diabp + diabetes + prevmi + prevhyp +
             prevstrk + glucose + cigpday + I(hdlc^2) + I(diabp^2) +
             I(bmi^2) + I(heartrte^2) + I(glucose^2) + sysbp:prevmi +
             diabetes:prevmi + sysbp:prevhyp + diabp:hdlc + sysbp:diabetes +
             prevmi:hdlc + prevhyp:hdlc + diabp:heartrte + age:prevhyp
             + cigpday:hdlc + bmi:prevhyp + diabp:cursmoke +
             diabp:glucose + prevmi:prevhyp + diabp:prevhyp +
             cigpday:glucose,
data = fhsdm)
```

```r
### useful functions

# the function for plotting the Residuals vs Fitted Values plot
threePlots <- function(M) {
  res <- residuals(M) # usual residuals
  X <- model.matrix(M) # design matrix
  H <- X %*% solve(crossprod(X), t(X)) # Hat matrix
  h <- diag(H)
  range(h - hatvalues(M)) # R way of calculating these
  # plot the Residuals vs Fitted Values
  res.stu <- resid(M)/sqrt(1-h) # studentized residuals, but on the data   scale
  cex <- .8 # controls the size of the points and labels
  #par(mar = c(4,4,.5,.1))
  plot(predict(M), res, pch = 21, bg = "black", cex = cex, cex.axis = cex,
       xlab = "Predicted logit CHD risk values", ylab = "Residual CHD risk values")
  points(predict(M), res.stu, pch = 21, bg = "red", cex = cex)
  abline(h = 0, lty = 2, col = "grey") # add horizontal line at 0
  legend(x = "bottomleft", c("Residuals", "Studentized Residuals"),
  pch = 21, pt.bg = c("black", "red"), pt.cex = cex, cex = cex)

  # the function for plotting the residual distribution
  # plot standardized residuals
  sigma.hat <- sigma(M)
  cex <- .8
  #par(mfrow = c(1,2), mar = c(4,4,.1,.1))
  res <- residuals(M) # usual residuals
  # histogram
  hist(res/sigma.hat, breaks = 50, freq = FALSE, cex.axis = cex,
     xlab = "Standardized Residual CHD values", main = "")
  curve(dnorm(x), col = "red", add = TRUE) # theoretical normal curve
  # qq-plot
  qqnorm(res/sigma.hat, main = "", pch = 16, cex = cex, cex.axis = cex)
  abline(a = 0, b = 1, col = "red") # add 45 degree line
}

leveragePlot <- function(M) {
  # calculate all types of residuals
  res.ord <- resid(M) # ordinary residuals e_i = y_i - x_i'beta_hat
  # standardized (units of sd's)
  sigma.hat <- sigma(M)
  res.stan <- res.ord/sigma.hat
  # studentized (account for different variances of e_i's)
  # compute leverages
  # the long way
  X <- model.matrix(M) # design matrix
  H <- X %*% solve(crossprod(X), t(X)) # Hat matrix
  head(diag(H))
  h <- hatvalues(M) # the R way
  range(h - diag(H))

  # studentized residuals
  res.stud <- res.ord/(sigma.hat * sqrt(1-h))
```

```r
# press residuals (obs minus leave-one-out fit)
res.press <- res.ord/(1-h)

# dffits residuals (difference between full and L1-out predictions)
# standardized version of sqrt(h)/(1-h) * e
res.dffits <- dffits(M)

# residual vs fitted for all types of residuals

yhat <- predict(M)

# collect all residuals
Resid <- data.frame(stan = res.stan,
                    stud = res.stud,
                    press = res.press,
                    dffits = res.dffits)

# standardize residuals by making them all equal at average leverage
# that is:
# at hbar = mean(h),
# * res.stud = res.stan
# * res.press = res.stan
# * res.dffits = res.stan

hbar <- ncol(model.matrix(M))/nobs(M) # hbar = p/n
Resid <- within(Resid, {
  stud <- stud * sqrt(1-hbar)
  press <- press * (1-hbar)/sigma.hat
  dffits <- dffits * (1-hbar)/sqrt(hbar)
})

# plot
#par(mfrow = c(1,2), mar = c(4,4,1,.1))
clrs <- c("black", "blue", "red", "orange")
pch <- 21
cex <- .8
# (1) residuals vs predicted values
# empty plot to get axes
plot(x = 0, type = "n", # empty plot to get the axis range
    xlim = range(yhat),
    # ylim = range(Resid)
    ylim = c(-45,5), cex.lab = cex, cex.axis = cex,
    xlab = "Fitted Values", ylab = "Residuals")
# add dotted lines between residuals to enhance visibility
res.y0 <- apply(Resid, 1, min)
res.y1 <- apply(Resid, 1, max)
segments(x0 = yhat, y0 = res.y0, y1 = res.y1, lty = 2)
# add points
for(ii in 1:4) {
  points(yhat, Resid[,ii], pch = pch, cex = cex, bg = clrs[ii])
}
abline(h = 0, col = "grey60")
# (2) residuals vs leverage
```

```
  plot(x = 0, type = "n", # empty plot to get the axis range
      xlim = range(h),
      ylim = c(-45,5), cex.lab = cex, cex.axis = cex,
      xlab = "Leverages", ylab = "Residuals")
  segments(x0 = h, y0 = res.y0, y1 = res.y1, lty = 2)
  for(ii in 1:4) {
    points(h, Resid[,ii], pch = pch, cex = cex, bg = clrs[ii])
  }
  legend("bottomleft",
        legend = c("Standardized", "Studentized", "PRESS", "DFFITS"),
        pch = 21, pt.cex = cex, cex = cex,
        pt.bg = c("black", "blue", "red", "orange"),
        title = "Residual Type:")
  abline(v = hbar, col = "grey60")
  abline(h = 0, col = "grey60")
}

# the function for plotting influence measure (cook's distance)
cookDisPlot <- function(M) {
  h <- hatvalues(M)
  hbar <- length(coef(M))/nobs(M)
  D <- cooks.distance(M) # Cook's distance

  # flag some of high leverage and high influence points
  infl.ind <- D == max(D) # top influence point
  lev.ind <- h > 2*hbar # leverage more than 2x the average

  #par(mfrow = c(1,1), mar = c(4,4,1,1))
  # color vector:
  # * black: normal point
  # * red: high influence point
  # * blue: high leverage point
  n <- nobs(M)
  clrs <- rep("black", len = n)
  clrs[lev.ind] <- "blue"
  clrs[infl.ind] <- "red"
  cex <- .8
  plot(h, D, pch = 21, bg = clrs, cex = cex,
      xlab = "Leverage", ylab = "Cook's Influence Measure")
  abline(v = 2*hbar, col = "grey60", lty = 2) # 2x average leverage
  legend("topleft", legend = c("High Leverage", "High Influence"), pch = 21,
        pt.bg = c("blue", "red"), pt.cex = cex)
}
```

```
# Mstep plot residuals
par(mfrow = c(2,2), mar= c(4,4,.1,.1))
threePlots(Mstep)

# Mman plot residuals
par(mfrow=c(2,2),mar=c(4,4,.1,.1))
threePlots(Mman)

# leverage
par(mfrow = c(2,2), mar = c(4,4,.1,.1))
```

```r
# plotting leverage for Mstep
leveragePlot(Mstep)
# plotting leverage for Mman
leveragePlot(Mman)

# influence
par(mfrow = c(1,2), mar=c(4,4,.1,.1))
# plotting cook's distance influence measure
cookDisPlot(Mstep)
# plotting cook's distance influence measure
cookDisPlot(Mman)
```

```r
require(statmod)
logitnorm_mean <- function(mu, sigma) {
  v = 1/(1+exp(-mu))
  alpha1 <- 1/(sigma^2 * (1-v))
  alpha2 <- 1/(v*sigma^2)
  gqp <- gauss.quad.prob(n = 10, dist = "beta", alpha = alpha1, beta = alpha2)
  x <- gqp$nodes # (x_1, ..., x_10)
  w <- gqp$weights # (w_1, ..., w_10)
  g_x = dnorm(log(x) - log(1-x), mean = mu, sd = sigma, log = TRUE) -
    log(1-x) - dbeta(x,shape1 = alpha1, shape2 = alpha2, log = TRUE)
  expo_g_x <- exp(g_x)
  w %*% expo_g_x
}

# compare Mfwd to Mstep
M1 <- Mstep
M2 <- Mman
Mnames <- expression("Mstep", "Mman")

# number of cross-validation replications
nreps <- 1e3

ntot <- nrow(fhsdm) # total number of observations
ntrain <- 500 # for fitting MLE's
ntest <- ntot-ntrain # for out-of-sample prediction

# storage space
rmspe1 <- rep(NA, nreps) # rmspe for M1
rmspe2 <- rep(NA, nreps) # rmspe for M2
lambda1 <- rep(NA, nreps) # out-of-sample log-likelihood for M1
lambda2 <- rep(NA, nreps) # out-of-sample log-likelihood for M2

for(ii in 1:nreps) {
  if(ii%%100 == 0) message("ii = ", ii)
  train.ind <- sample(ntot, ntrain) # training observations
  # long-form cross-validation
  ## M1.cv <- lm(math ~ read + prog + race + ses + locus + read:prog + prog:ses,
  ##          data = hsbm, subset = train.ind)
  ## M2.cv <- lm(math ~ race + ses + sch + prog + locus + concept +
  #            mot + read + ses:sch + ses:concept + prog:read,
    #          data = hsbm, subset = train.ind)
    # using R functions
```

```
  M1.cv <- update(M1, subset = train.ind)
  M2.cv <- update(M2, subset = train.ind)
  # compute mu and sigma
  M1.mu <- predict(M1.cv, newdata = fhsdm[-train.ind,])
  M2.mu <- predict(M2.cv, newdata = fhsdm[-train.ind,])
  M1.sigma <- sqrt(sum(resid(M1.cv)^2)/ntrain) # MLE of sigma
  M2.sigma <- sqrt(sum(resid(M2.cv)^2)/ntrain)

  # cross-validation residuals
  M1.res <- fhsdm$chdrisk[-train.ind] - # test observations
      logitnorm_mean(M1.mu[ii],M1.sigma) # prediction with training data
  M2.res <- fhsdm$chdrisk[-train.ind] -
       logitnorm_mean(M2.mu[ii], M2.sigma)
  # mspe for each model
  rmspe1[ii] <- sqrt(mean(M1.res^2))
  rmspe2[ii] <- sqrt(mean(M2.res^2))
  # out-of-sample log-likelihoods
  #M1.sigma <- sqrt(sum(resid(M1.cv)^2)/ntrain) # MLE of sigma
  #M2.sigma <- sqrt(sum(resid(M2.cv)^2)/ntrain)
  # since res = y - pred, dnorm(y, pred, sd) = dnorm(res, 0, sd)
  lambda1[ii] <- sum(dnorm(M1.res, sd = M1.sigma, log = TRUE))
  lambda2[ii] <- sum(dnorm(M2.res, sd = M2.sigma, log = TRUE))
}

# compare
par(mfrow = c(1,2), mar = c(4, 4, 2.1, 2.1))
cex <- .8
boxplot(x = list(rmspe1, rmspe2), names = Mnames,
        main = "Root MSPE",
        ylab = expression(sqrt(MSPE)),
        ## ylab = expression(SSE[CV]),
        col = c("yellow", "orange"),
        cex = cex, cex.lab = cex, cex.axis = cex, cex.main = cex)
lambda <- lambda1 - lambda2
hist(lambda, breaks = 50, freq = FALSE,
     main = "Out-of-Sample Log-Likelihood Ratio
     Statistic",
     xlab = expression(log(Lambda[CV])),
     xlim = c(-500,500),
     cex = cex, cex.lab = cex, cex.axis = cex, cex.main = cex)
abline(v = mean(lambda), col = "red", lwd = 2)

# The final answer is Mstep
summary(Mstep)
```