STAT 331 Final Project

Krishna Prem Pasumarthy & Islam Amin

April 14, 2020

1 Summary

2 Descriptive Statistics

First, take a look at summary statistics of the Framingham Heart Study dataset.

Table 1: Summary Statistics

chdrisk	sex	totchol	age	sysbp	diabp	cursmoke	cigpday	bmi
Min. :0.0050	Female:1305	Min. :112.0	Min. :44.00	Min. : 86.0	Min. : 30.00	No :1504	Min. : 0.00	Min. :14.43
1st Qu.:0.1320	Male :1001	1st Qu.:207.0	1st Qu.:53.00	1st Qu.:122.5	1st Qu.: 73.00	Yes: 802	1st Qu.: 0.00	1st Qu.:23.22
Median :0.2240		Median :235.5	Median :60.00	Median :136.0	Median : 80.00		Median : 0.00	Median :25.40
Mean :0.2655		Mean :237.8	Mean :60.23	Mean :139.2	Mean : 81.07		Mean : 6.84	Mean :25.78
3rd Qu.:0.3448		3rd Qu.:265.0	3rd Qu.:67.00	3rd Qu.:153.0	3rd Qu.: 88.00		3rd Qu.:10.00	3rd Qu.:27.91
Max. :0.9770		Max. :625.0	Max. :81.00	Max. :246.0	Max. :130.00		Max. :80.00	Max. :46.52

diabetes	bpmeds	heartrte	glucose	prevmi	prevstrk	prevhyp	hdlc	ldlc
No :2142	No :1973	Min. : 44.00	Min. : 46.00	No :2189	No :2260	No : 957	Min. : 10.00	Min. : 20.0
Yes: 164	Yes: 333	1st Qu.: 70.00	1st Qu.: 75.00	Yes: 117	Yes: 46	Yes:1349	1st Qu.: 38.00	1st Qu.:152.0
		Median : 76.00	Median : 83.00				Median : 47.00	Median :180.0
		Mean : 77.61	Mean : 89.07				Mean : 48.89	Mean :183.1
		3rd Qu.: 85.00	3rd Qu.: 95.00				3rd Qu.: 57.00	3rd Qu.:210.0
		Max. :150.00	Max. :478.00				Max. :189.00	Max. :565.0

First observation we make from the summary is that the median and average ages are around 60, which means the survey seems to have been done on a relatively old group of people. We also have a significantly higher number of females in the study, almost 30% more than the number of males. This might affect the nature of the data to be skewed towards behaviours and physical attributes associated with females.

A further inspection of the expected coronary heart disease (CHD) risk against certain categorical variates, gives more insights.

For instance, if we take a look at expected CHD risk against whether or not an individual has hypertension, we get the following result:

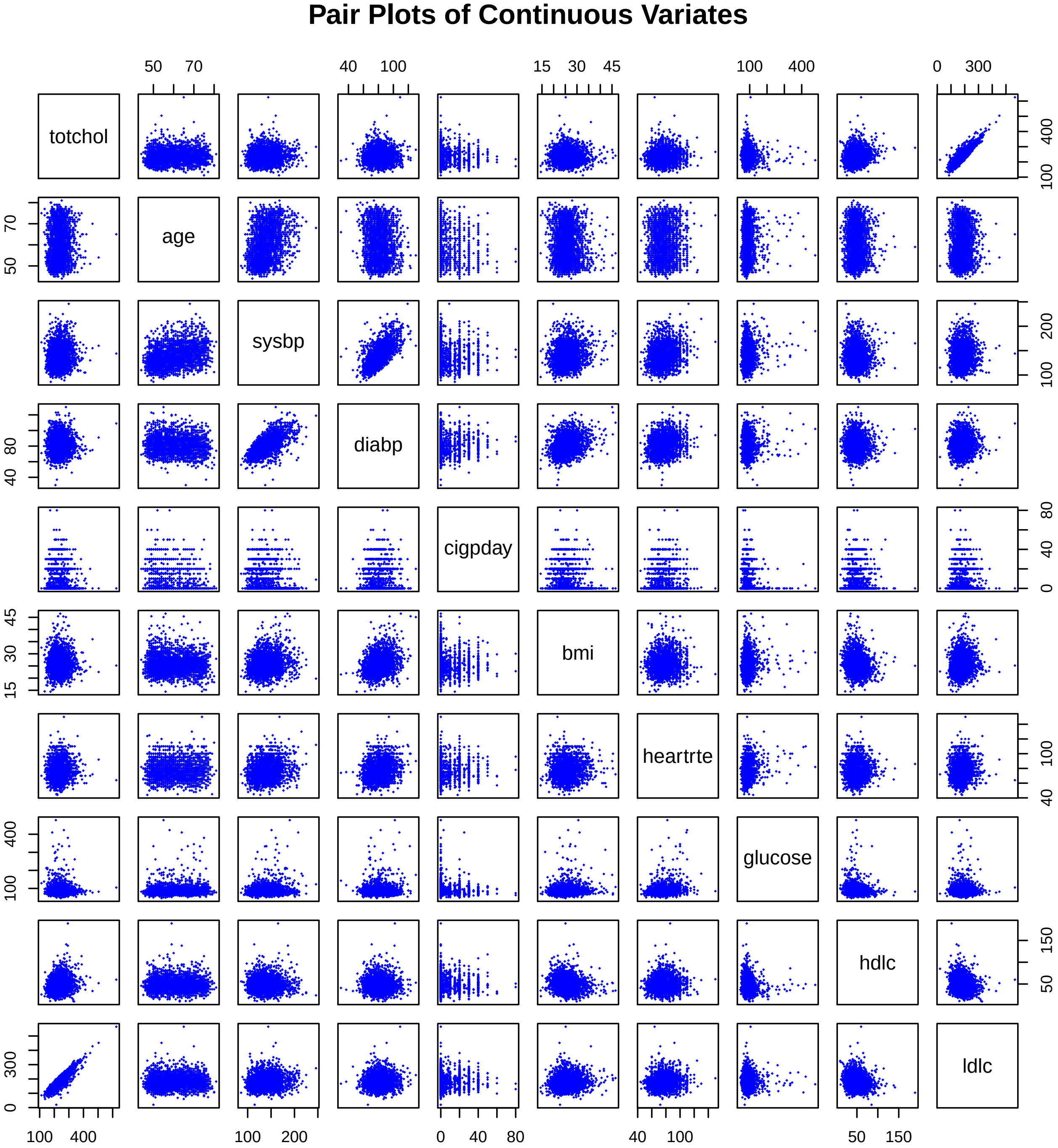
```
## fhsd$prevhyp: No
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.005   0.077   0.140   0.176   0.216   0.944
## ------------------------------------------------------------
## fhsd$prevhyp: Yes
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0320  0.1980  0.2890  0.3291  0.4010  0.9770
```

Indeed, we have that mean CHD risk given that a person has hypertension is significantly higher than the mean for people who did not have hypertension.

```
## fhsd$prevstrk: No
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0050  0.1300  0.2200  0.2611  0.3392  0.9770
## ------------------------------------------------------------
## fhsd$prevstrk: Yes
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.2020  0.3412  0.4410  0.4820  0.5060  0.9660 
```

Again, we see the same results with people who had a stroke before the study, with even a higher difference between the two groups.

Now take a look at pair plots of all numeric explanatory variates i.e. variates excluding response variate `chdrisk` and logical variates such as `cursmoke`.



From the pair plots, we can observe a strong correlation between low density lipoprotein cholesterol and serum total cholestrol. This correlation could be explained by the fact that there could be a relationship between the amount [TO BE CONTINUED]

Now take a look at the VIFs of these variates.

```
##      sexMale      totchol          age        sysbp        diabp  cursmokeYes
##     1.225191    10.634882     1.489926     2.918660     2.406411     2.978609
##      cigpday          bmi  diabetesYes    bpmedsYes     heartrte      glucose
##     2.973594     1.181865     1.286401     1.214744     1.105902     1.308923
##    prevmiYes prevstrkYes   prevhypYes         hdlc         ldlc
##     1.067134     1.045746     1.823014     2.287571    10.367649
```

[ADD COMMENTS]

3 Candidate Models

3.1 Automated Model Selection

```
library(gtools)
load_calcs = TRUE
# model with only intercept
M0 <- lm(I(logit(chdrisk)) ~ 1, data = fhsd)
Mmax <- lm(I(logit(chdrisk)) ~ (.)^2, data = fhsd)
# starting model for stepwise selection
Mstart <- lm(I(logit(chdrisk)) ~ ., data = fhsd)
# find model coefficients which are NA
beta.max <- coef(Mmax)
names(beta.max)[is.na(beta.max)]
```

```
## [1] "cursmokeYes:cigpday"  "bpmedsYes:prevhypYes"
```

```
# find the problem with the NA coeffs
kable(table(fhsd[c("cursmoke", "cigpday")]), "latex")
```

	0	1	2	3	4	5	6	7	8	9	10	12	14	15	16	17	18	19	20	23	25	26	27
No	1504	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Yes	0	16	18	34	11	18	24	9	18	5	76	3	3	50	6	1	8	1	279	1	14	1	1

```
kable(table(fhsd[c("bpmeds", "prevhyp")]), "latex")
```

	No	Yes
No	957	1016
Yes	0	333

```
# remove the coeffs with the problem and add quadratic terms for the continuous variables
Mmax <- lm(I(logit(chdrisk)) ~ (.)^2 - cursmoke:cigpday - bpmeds:prevhyp +
             I(totchol ^ 2) + I(sysbp ^ 2) + I(diabp ^ 2)
           + I(bmi ^ 2) + I(glucose ^ 2)
           + I(hdlc ^ 2) + I(ldlc ^ 2), data = fhsd)
anyNA(coef(Mmax)) # check if there are any remaining NAs
```

```
## [1] FALSE
```

```
if(!load_calcs){
  #forward model selection
  system.time({
    Mfwd <- step(object = M0,
                 scope = list(lower = M0, upper = Mmax),
                 direction = "forward", trace = FALSE)
  })

  #backward model selection
  system.time({
    Mback <- step(object = Mmax,
                  scope = list(lower = M0, upper = Mmax),
                  direction = "backward", trace = FALSE)
  })

  #stepwise model selection
  system.time({
    Mstep <- step(object = Mstart,
                  scope = list(lower = M0, upper = Mmax),
                  direction = "both", trace = FALSE)
  })
}
# the caching/loading block
if(!load_calcs) {
  saveRDS(list(Mfwd = Mfwd, Mback = Mback, Mstep = Mstep), file = "models_automated.rds")
} else {
  # just load the calculations
  tmp <- readRDS("models_automated.rds")
  Mfwd <- tmp$Mfwd
  Mback <- tmp$Mback
  Mstep <- tmp$Mstep
  rm(tmp) # optionally remove tmp from workspace
}
# Stepwise model selection
Mstep$call
```

```
## lm(formula = I(logit(chdrisk)) ~ sex + totchol + age + sysbp +
##     diabp + cursmoke + cigpday + bmi + diabetes + bpmeds + heartrte +
##     glucose + prevmi + prevstrk + prevhyp + hdlc + ldlc + I(hdlc^2) +
##     I(bmi^2) + I(diabp^2) + I(sysbp^2) + sysbp:prevmi + totchol:prevhyp +
##     diabetes:prevmi + prevhyp:ldlc + sysbp:prevhyp + totchol:heartrte +
##     sysbp:diabetes + diabp:bmi + diabp:hdlc + prevmi:hdlc + prevmi:prevhyp +
##     sex:glucose + age:ldlc + age:heartrte + cigpday:hdlc + bmi:ldlc +
##     totchol:hdlc + totchol:prevmi + sysbp:heartrte + sysbp:bpmeds +
##     cursmoke:hdlc + prevmi:prevstrk + diabetes:hdlc + sex:sysbp +
##     cigpday:glucose + heartrte:glucose + diabp:glucose + cursmoke:ldlc +
##     age:cigpday + age:hdlc + hdlc:ldlc + age:prevhyp + diabp:prevhyp +
##     diabp:cursmoke + diabp:cigpday + bmi:bpmeds + bpmeds:glucose +
##     age:prevmi + sex:ldlc + cigpday:heartrte + cigpday:prevmi +
##     glucose:prevmi + heartrte:prevmi + bpmeds:prevstrk, data = fhsd)
```

```
# Forward model selection
Mfwd$call
```

```
## lm(formula = I(logit(chdrisk)) ~ prevmi + sysbp + sex + age +
##     ldlc + prevhyp + diabetes + hdlc + I(hdlc^2) + cigpday +
##     I(bmi^2) + bmi + totchol + I(glucose^2) + I(sysbp^2) + bpmeds +
##     heartrte + cursmoke + prevstrk + prevmi:sysbp + sysbp:age +
##     prevhyp:hdlc + prevmi:diabetes + sysbp:prevhyp + prevhyp:totchol +
##     sysbp:diabetes + prevmi:hdlc + prevmi:prevhyp + age:ldlc +
##     age:cigpday + hdlc:cigpday + prevhyp:bmi + ldlc:bmi + prevmi:totchol +
##     ldlc:prevhyp + sysbp:bpmeds + sysbp:hdlc + hdlc:totchol +
##     totchol:heartrte + age:heartrte + diabetes:hdlc + sysbp:heartrte +
##     bmi:bpmeds + sysbp:sex + ldlc:hdlc + prevmi:bmi + age:bmi +
##     prevmi:age + sysbp:cursmoke + hdlc:cursmoke + ldlc:cursmoke +
##     prevmi:cigpday + sex:diabetes + prevmi:prevstrk, data = fhsd)
```

```
# Backward model selection
Mback$call
```

```
## lm(formula = I(logit(chdrisk)) ~ sex + totchol + age + sysbp +
##     diabp + cursmoke + cigpday + bmi + diabetes + bpmeds + heartrte +
##     glucose + prevmi + prevstrk + prevhyp + hdlc + ldlc + I(totchol^2) +
##     I(sysbp^2) + I(diabp^2) + I(bmi^2) + I(hdlc^2) + I(ldlc^2) +
##     sex:totchol + sex:sysbp + sex:glucose + sex:prevstrk + sex:prevhyp +
##     totchol:age + totchol:bpmeds + totchol:heartrte + totchol:prevmi +
##     totchol:prevstrk + totchol:prevhyp + totchol:hdlc + totchol:ldlc +
##     age:cursmoke + age:bmi + age:heartrte + age:prevmi + age:prevhyp +
##     age:hdlc + sysbp:diabetes + sysbp:bpmeds + sysbp:heartrte +
##     sysbp:prevmi + sysbp:prevhyp + diabp:cursmoke + diabp:cigpday +
##     diabp:bmi + diabp:glucose + diabp:prevhyp + diabp:hdlc +
##     cursmoke:bmi + cursmoke:hdlc + cursmoke:ldlc + cigpday:bmi +
##     cigpday:heartrte + cigpday:glucose + cigpday:prevmi + cigpday:hdlc +
##     bmi:prevmi + bmi:prevhyp + bmi:ldlc + diabetes:prevmi + diabetes:hdlc +
##     bpmeds:glucose + bpmeds:prevstrk + bpmeds:ldlc + heartrte:glucose +
##     heartrte:prevmi + glucose:prevmi + prevmi:prevhyp + prevmi:hdlc +
##     prevhyp:ldlc, data = fhsd)
```

```
beta.fwd = coef(Mfwd)
beta.back = coef(Mback)
beta.step = coef(Mstep)
identical(names(beta.fwd)[names(beta.fwd) %in% names(beta.back)], names(beta.fwd))
```

```
## [1] FALSE
```

```
identical(names(beta.fwd)[names(beta.fwd) %in% names(beta.step)], names(beta.fwd))
```

```
## [1] FALSE
```

```
identical(names(beta.back)[names(beta.back) %in% names(beta.step)], names(beta.back))
```

```
## [1] FALSE
```

3.2 Manual Model Selection

```
library(stringr) # For string operations
```

```
## Warning: package 'stringr' was built under R version 3.5.2
```

```
 table <- c() # Initialize empty vector
 names.table <- names(beta.step)                  # Obtain variate names in stepwise model
 names.table <- str_remove_all(names.table,"Yes") # Remove "Yes" from interactions
 names.table <- str_remove_all(names.table,"Male") # Remove "Male"
 # Perform F-tests with Mstep by removing one variate at a time
  for(i in names.table){
    # Obtain model without variate i
    mdl <- lm(as.formula(paste0("update(Mstep, . ~ . -", i,")")),data = fhsd)
   test <- anova(Mstep,mdl)               # F-Test between Stepwise and reduced model
   table <- cbind(table,test$`Pr(>F)`[2]) # Add corresponding p-value to the table
  }
 table <- as.data.frame(table)
 colnames(table) <- names.table            # Add appropriate column names to the table
sort(table,decreasing = TRUE)              # Arrange variates by decreasing significance
```

```
##   cigpday:heartrte bpmeds:prevstrk bpmeds:glucose diabp:cigpday   cigpday
## 1        0.1506282       0.1492283      0.1189197     0.1155989 0.1151079
##    sex:ldlc age:prevmi cigpday:prevmi hdlc:ldlc bmi:bpmeds prevmi:prevstrk
## 1 0.1141483  0.1097987      0.1051865 0.0923568  0.0855445      0.06997763
##   heartrte:prevmi glucose:prevmi I(sysbp^2) cursmoke:hdlc age:heartrte
## 1      0.06451949     0.05883116  0.0585469    0.05660935   0.05562064
##    age:hdlc cursmoke:ldlc  sex:sysbp sysbp:bpmeds   age:ldlc
## 1 0.0510796     0.0417893 0.03623249    0.0300776 0.02915113
##   cigpday:glucose prevmi:prevhyp       hdlc sex:glucose diabetes:hdlc
## 1       0.0291137     0.02242217 0.01880445  0.01702301    0.01394662
##   diabp:glucose    bmi:ldlc totchol:hdlc    bpmeds age:cigpday
## 1    0.01362058 0.009985489  0.009840662 0.0077735 0.006735591
##   heartrte:glucose    cursmoke totchol:prevmi sysbp:heartrte diabp:prevhyp
## 1      0.004772297 0.004188557    0.003609581    0.002926201   0.001409115
##   diabp:cursmoke prevhyp:ldlc          bmi  age:prevhyp sysbp:diabetes
## 1    0.001393474   0.00066789 0.0006664543 0.0005753017   0.0004931994
##     I(hdlc^2)   diabp:hdlc sysbp:prevhyp cigpday:hdlc  prevmi:hdlc
## 1 0.000320732 0.0001422969  0.0001292531 0.0001038006 7.056001e-05
##   diabetes:prevmi        diabp totchol:heartrte    diabp:bmi sysbp:prevmi
## 1    6.226049e-05 6.021714e-05     3.512093e-05 2.940165e-05 2.305381e-05
##            sex     heartrte          age totchol:prevhyp     I(bmi^2)
## 1 2.396724e-06 9.478088e-07 4.238229e-07    1.203731e-09 2.735937e-11
##     I(diabp^2)       prevmi      prevhyp
## 1 1.257752e-19 1.595006e-22 1.119628e-27
```

```
# Remove as many insignificant continuous variate interactions as possible
anova(Mstep, update(Mstep,. ~ . - cigpday:heartrte - diabp:cigpday))
```

```
## Analysis of Variance Table
##
## Model 1: I(logit(chdrisk)) ~ sex + totchol + age + sysbp + diabp + cursmoke +
##     cigpday + bmi + diabetes + bpmeds + heartrte + glucose +
##     prevmi + prevstrk + prevhyp + hdlc + ldlc + I(hdlc^2) + I(bmi^2) +
##     I(diabp^2) + I(sysbp^2) + sysbp:prevmi + totchol:prevhyp +
##     diabetes:prevmi + prevhyp:ldlc + sysbp:prevhyp + totchol:heartrte +
##     sysbp:diabetes + diabp:bmi + diabp:hdlc + prevmi:hdlc + prevmi:prevhyp +
##     sex:glucose + age:ldlc + age:heartrte + cigpday:hdlc + bmi:ldlc +
##     totchol:hdlc + totchol:prevmi + sysbp:heartrte + sysbp:bpmeds +
##     cursmoke:hdlc + prevmi:prevstrk + diabetes:hdlc + sex:sysbp +
##     cigpday:glucose + heartrte:glucose + diabp:glucose + cursmoke:ldlc +
##     age:cigpday + age:hdlc + hdlc:ldlc + age:prevhyp + diabp:prevhyp +
##     diabp:cursmoke + diabp:cigpday + bmi:bpmeds + bpmeds:glucose +
##     age:prevmi + sex:ldlc + cigpday:heartrte + cigpday:prevmi +
##     glucose:prevmi + heartrte:prevmi + bpmeds:prevstrk
## Model 2: I(logit(chdrisk)) ~ sex + totchol + age + sysbp + diabp + cursmoke +
##     cigpday + bmi + diabetes + bpmeds + heartrte + glucose +
##     prevmi + prevstrk + prevhyp + hdlc + ldlc + I(hdlc^2) + I(bmi^2) +
##     I(diabp^2) + I(sysbp^2) + sysbp:prevmi + totchol:prevhyp +
##     diabetes:prevmi + prevhyp:ldlc + sysbp:prevhyp + totchol:heartrte +
##     sysbp:diabetes + diabp:bmi + diabp:hdlc + prevmi:hdlc + prevmi:prevhyp +
##     sex:glucose + age:ldlc + age:heartrte + cigpday:hdlc + bmi:ldlc +
##     totchol:hdlc + totchol:prevmi + sysbp:heartrte + sysbp:bpmeds +
##     cursmoke:hdlc + prevmi:prevstrk + diabetes:hdlc + sex:sysbp +
##     cigpday:glucose + heartrte:glucose + diabp:glucose + cursmoke:ldlc +
##     age:cigpday + age:hdlc + hdlc:ldlc + age:prevhyp + diabp:prevhyp +
##     diabp:cursmoke + bmi:bpmeds + bpmeds:glucose + age:prevmi +
##     sex:ldlc + cigpday:prevmi + glucose:prevmi + heartrte:prevmi +
##     bpmeds:prevstrk
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1   2240 489.70
## 2   2242 490.84 -2   -1.1458 2.6205 0.07299 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#anova(Mstep, update(Mstep,. ~ . - cigpday:heartrte - diabp:cigpday -age:heartrte))
# Now remove less insignificant interactions
anova(Mstep, update(Mstep,. ~ . - cigpday:heartrte - diabp:cigpday - cigpday:heartrte
                    - bpmeds:prevstrk))
```

```
## Analysis of Variance Table
##
## Model 1: I(logit(chdrisk)) ~ sex + totchol + age + sysbp + diabp + cursmoke +
##     cigpday + bmi + diabetes + bpmeds + heartrte + glucose +
##     prevmi + prevstrk + prevhyp + hdlc + ldlc + I(hdlc^2) + I(bmi^2) +
##     I(diabp^2) + I(sysbp^2) + sysbp:prevmi + totchol:prevhyp +
##     diabetes:prevmi + prevhyp:ldlc + sysbp:prevhyp + totchol:heartrte +
##     sysbp:diabetes + diabp:bmi + diabp:hdlc + prevmi:hdlc + prevmi:prevhyp +
##     sex:glucose + age:ldlc + age:heartrte + cigpday:hdlc + bmi:ldlc +
##     totchol:hdlc + totchol:prevmi + sysbp:heartrte + sysbp:bpmeds +
##     cursmoke:hdlc + prevmi:prevstrk + diabetes:hdlc + sex:sysbp +
##     cigpday:glucose + heartrte:glucose + diabp:glucose + cursmoke:ldlc +
##     age:cigpday + age:hdlc + hdlc:ldlc + age:prevhyp + diabp:prevhyp +
##     diabp:cursmoke + diabp:cigpday + bmi:bpmeds + bpmeds:glucose +
##     age:prevmi + sex:ldlc + cigpday:heartrte + cigpday:prevmi +
##     glucose:prevmi + heartrte:prevmi + bpmeds:prevstrk
## Model 2: I(logit(chdrisk)) ~ sex + totchol + age + sysbp + diabp + cursmoke +
##     cigpday + bmi + diabetes + bpmeds + heartrte + glucose +
##     prevmi + prevstrk + prevhyp + hdlc + ldlc + I(hdlc^2) + I(bmi^2) +
##     I(diabp^2) + I(sysbp^2) + sysbp:prevmi + totchol:prevhyp +
##     diabetes:prevmi + prevhyp:ldlc + sysbp:prevhyp + totchol:heartrte +
##     sysbp:diabetes + diabp:bmi + diabp:hdlc + prevmi:hdlc + prevmi:prevhyp +
##     sex:glucose + age:ldlc + age:heartrte + cigpday:hdlc + bmi:ldlc +
##     totchol:hdlc + totchol:prevmi + sysbp:heartrte + sysbp:bpmeds +
##     cursmoke:hdlc + prevmi:prevstrk + diabetes:hdlc + sex:sysbp +
##     cigpday:glucose + heartrte:glucose + diabp:glucose + cursmoke:ldlc +
##     age:cigpday + age:hdlc + hdlc:ldlc + age:prevhyp + diabp:prevhyp +
##     diabp:cursmoke + bmi:bpmeds + bpmeds:glucose + age:prevmi +
##     sex:ldlc + cigpday:prevmi + glucose:prevmi + heartrte:prevmi
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1   2240 489.70
## 2   2243 491.35 -3   -1.6506 2.5168 0.05656 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Mmanual <- update(Mstep,. ~ . - cigpday:heartrte - diabp:cigpday - cigpday:heartrte
                    - bpmeds:prevstrk)     # Denotes manually constructed model
```

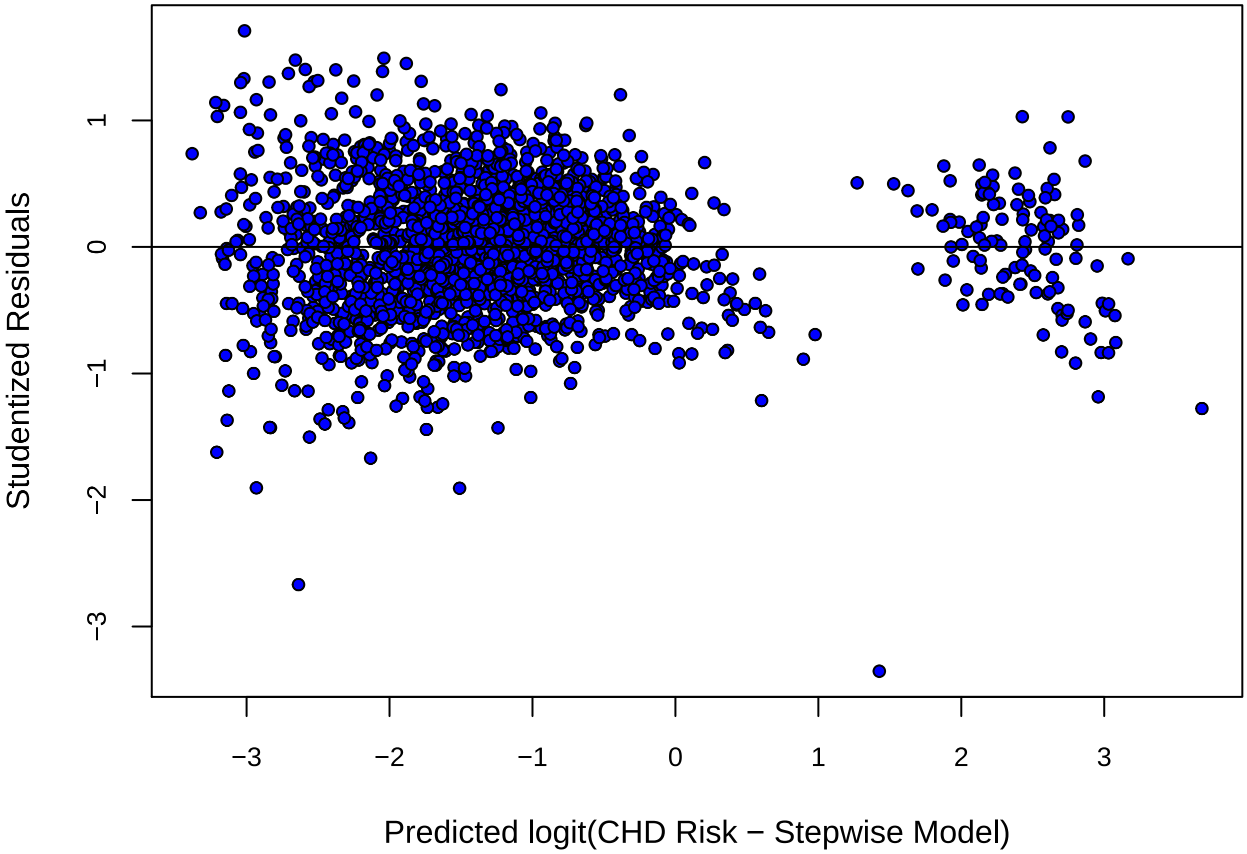
4 Model Diagnostics

4.1 Residual Plots

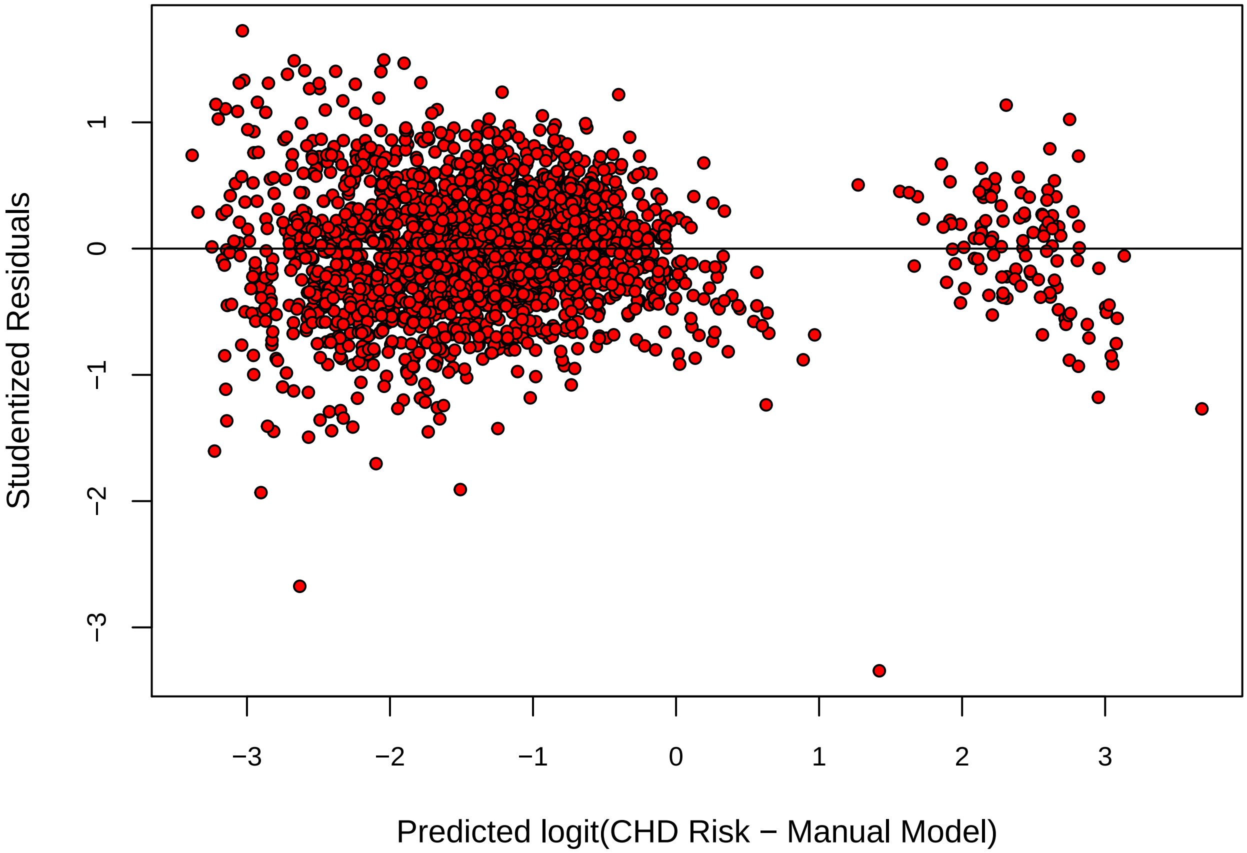
In this section we analyse the assumption that our residuals follow a normal distribution and check the homoscedasity assumption.

First, we have that the most normal looking residuals assuming that the model is true, would be the studentized residuals, so to check the homoscedasity we plot those values against the predicted values, as shown below:

```
# First we analyze Mstep
# get the hat values
h <- hatvalues(Mstep)
res.step <- resid(Mstep)/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(Mstep), res.step, pch = 21, bg = "blue", cex = cex, cex.axis = cex,xlab = "Predicted logit
abline(h = 0, lty = 1, col = "black") # add horizontal line at 0
```

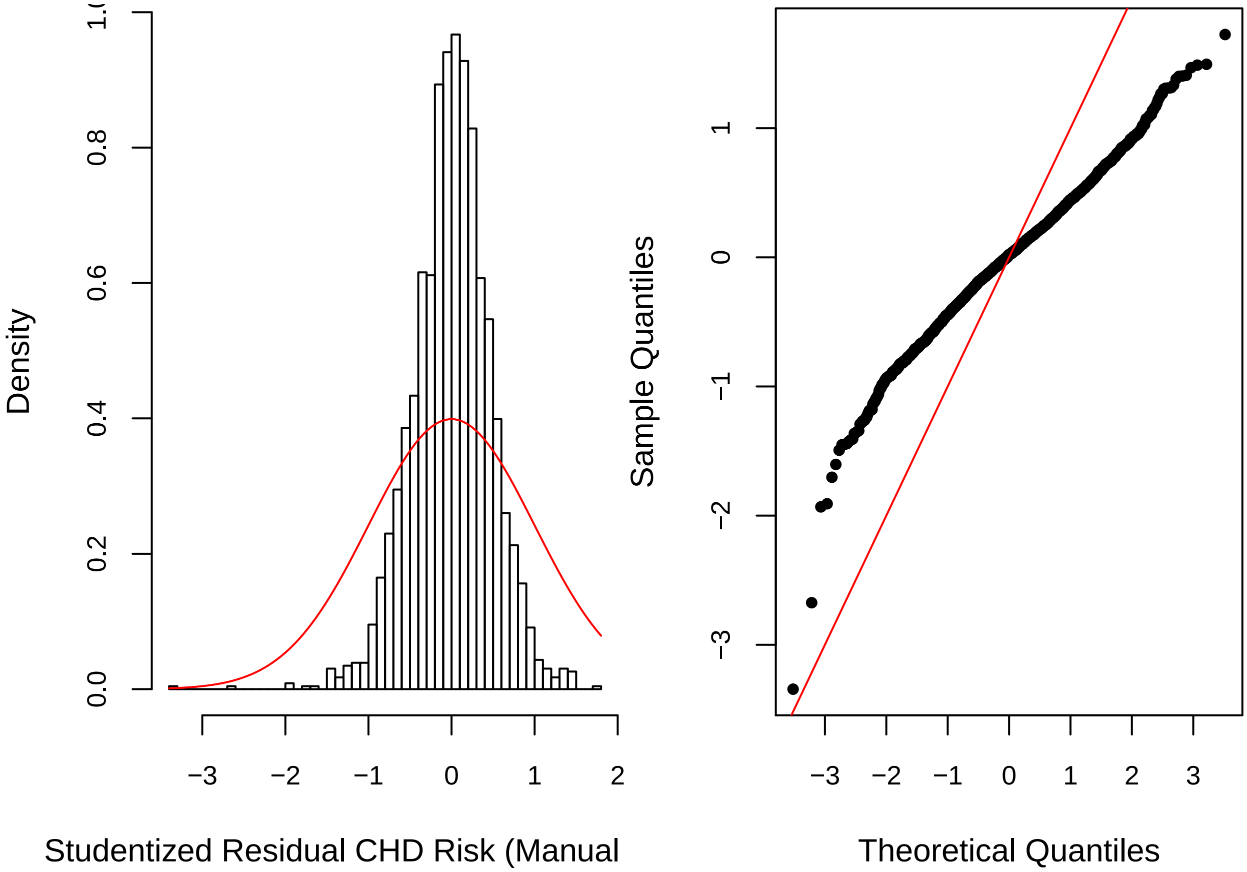


```
# Then we analyze Mdl_manual
# get the hat values
h <- hatvalues(Mmanual)
res.manual <- resid(Mmanual)/sqrt(1-h) # studentized residuals, but on the data scale
cex <- .8 # controls the size of the points and labels
par(mar = c(4,4,.5,.1))
plot(predict(Mmanual), res.manual, pch = 21, bg = "red", cex = cex, cex.axis = cex,xlab = "Predicted log
abline(h = 0, lty = 1, col = "black") # add horizontal line at 0
```

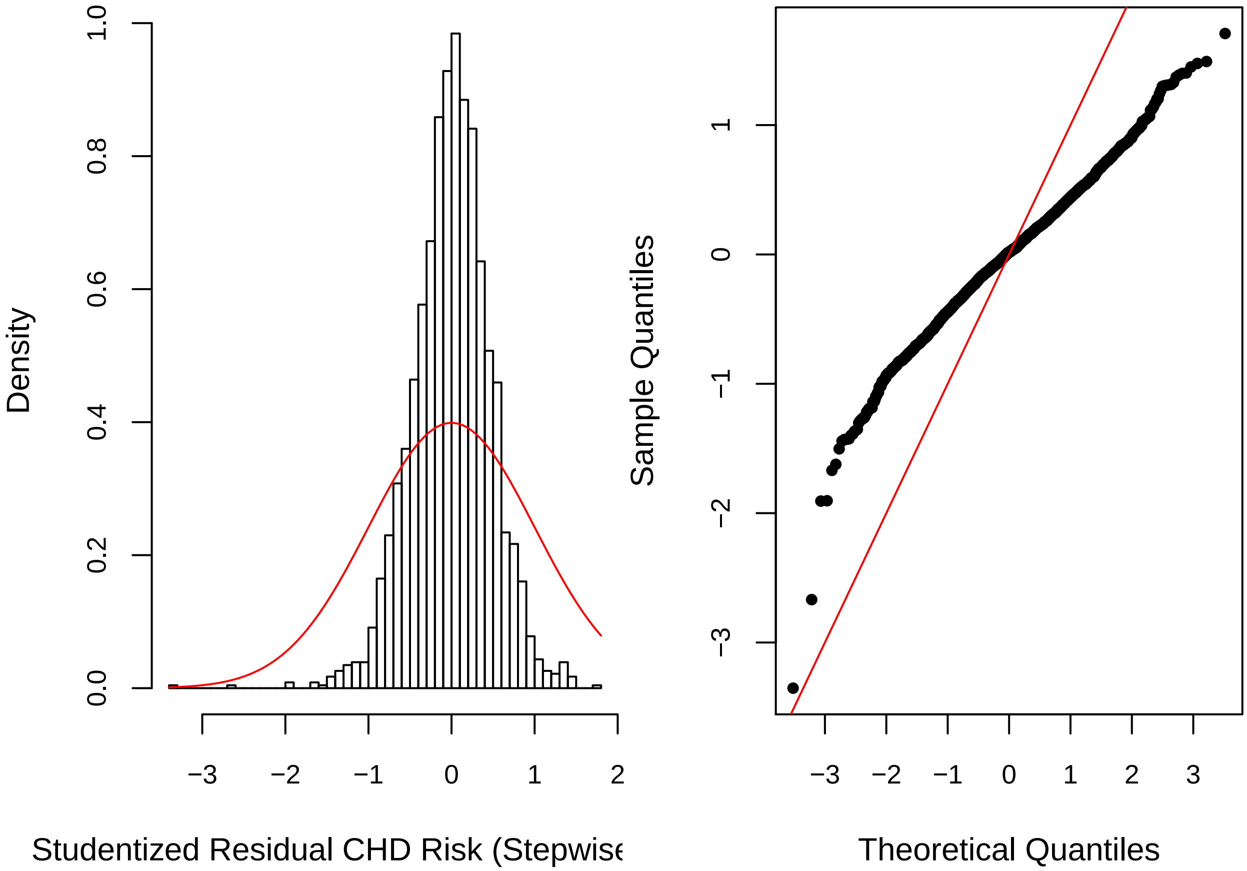


Then to check our assumption of normality of residuals we plot the residuals on a QQPlot and a histogram:

```
# plot standardized residuals
sigma.hat <- sigma(Mmanual)
cex <- .8
par(mfrow = c(1,2), mar = c(4,4,.1,.1))
# histogram
hist(res.manual, breaks = 50, freq = FALSE, cex.axis = cex,xlab = "Studentized Residual CHD Risk (Manual

curve(dnorm(x), col = "red", add = TRUE)
# theoretical normal curve
#qq-plot
qqnorm(res.manual, main = "", pch = 16, cex = cex, cex.axis = cex)
abline(a = 0, b = 1, col = "red") # add 45 degree line
```

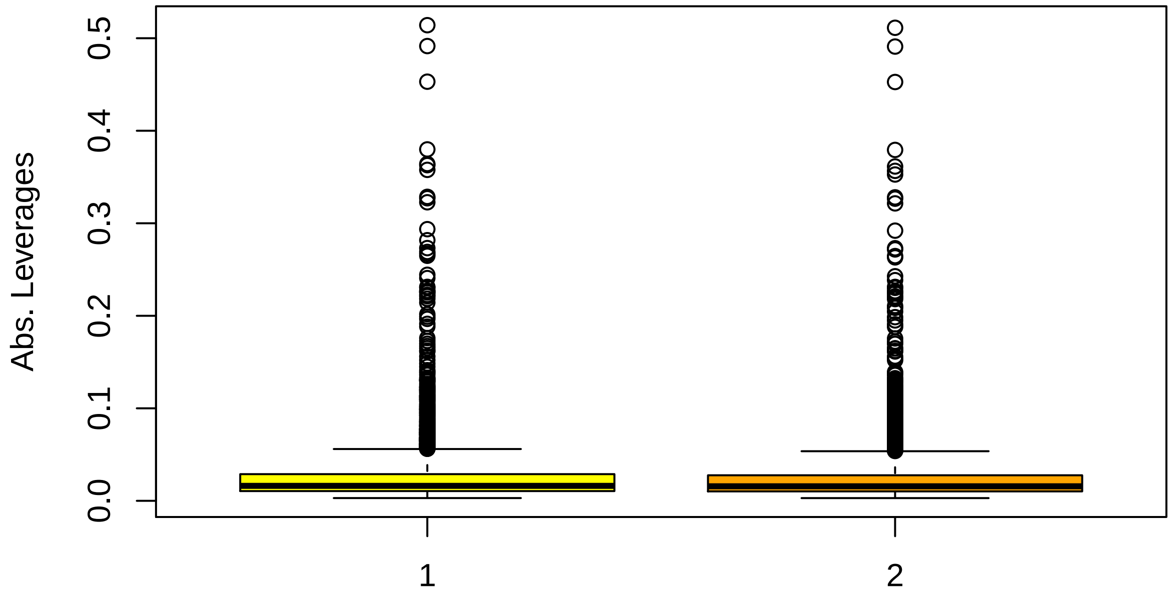


```r
# plot standardized residuals
sigma.hat <- sigma(Mstep)
cex <- .8
par(mfrow = c(1,2), mar = c(4,4,.1,.1))
# histogram
hist(res.step, breaks = 50, freq = FALSE, cex.axis = cex,xlab = "Studentized Residual CHD Risk (Stepwis

curve(dnorm(x), col = "red", add = TRUE)
# theoretical normal curve
#qq-plot
qqnorm(res.step, main = "", pch = 16, cex = cex, cex.axis = cex)
abline(a = 0, b = 1, col = "red") # add 45 degree line
```

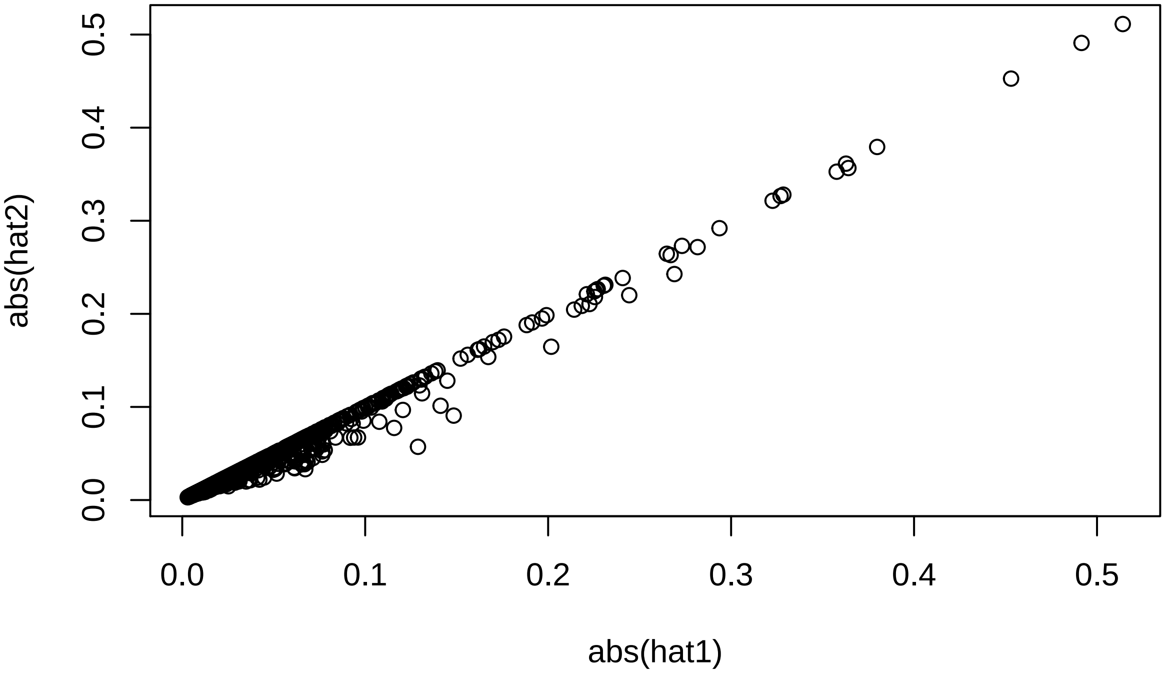


4.2 Leverage and Influence Measures

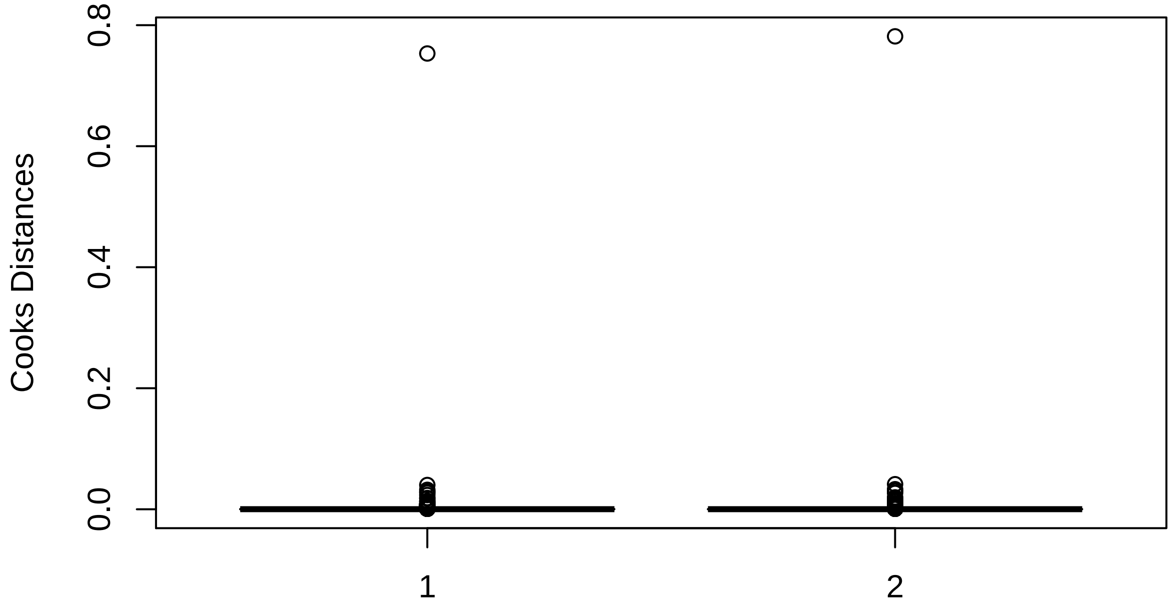
```
hat1 <- hatvalues(Mstep) # Leverages of stepwise model
hat2 <- hatvalues(Mmanual)

# Should be ideally close to 1 (as in course notes)
boxplot(x = list(abs(hat1), abs(hat2)),
        ylab = "Abs. Leverages", col = c("yellow", "orange"))
```



```
plot(abs(hat1),abs(hat2)) # Nearly linear
```



```
cook1 <- cooks.distance(Mstep)
cook2 <- cooks.distance(Mmanual)

 # Values should ideally be close to zero
boxplot(x = list(abs(cook1), abs(cook2)),
        ylab = "Cooks Distances", col = c("yellow", "orange"))
```



```
# Should this even be done since cooks is already done?
# dffits1 <- dffits(Mstep)
# dffits2 <- dffits(Mmanual)
#
# boxplot(x = list(abs(dffits1), abs(dffits2)),
#         ylab = "Abs. Leverages", col = c("yellow", "orange"))
#
# cooks.distance(Mstep)
```

5 Model Selection

5.1 Cross Validation

This is the written function

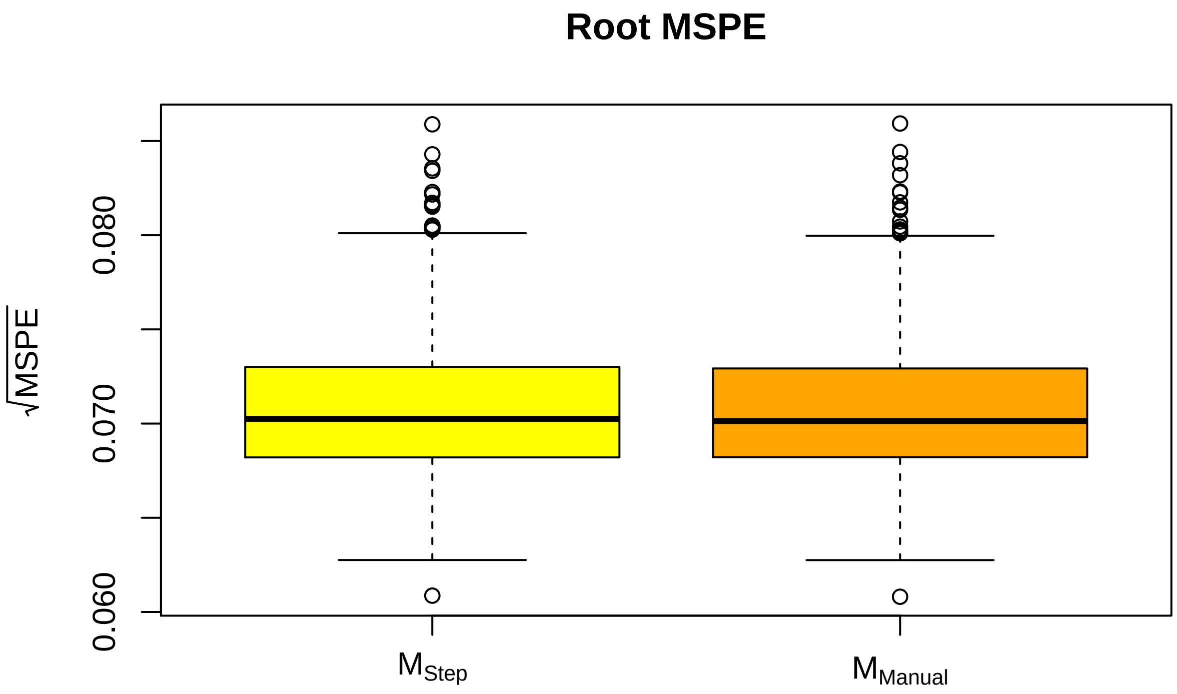
```
library(statmod) # Load this package for using gauss.quad.prob() function
```

```
## Warning: package 'statmod' was built under R version 3.5.2
```

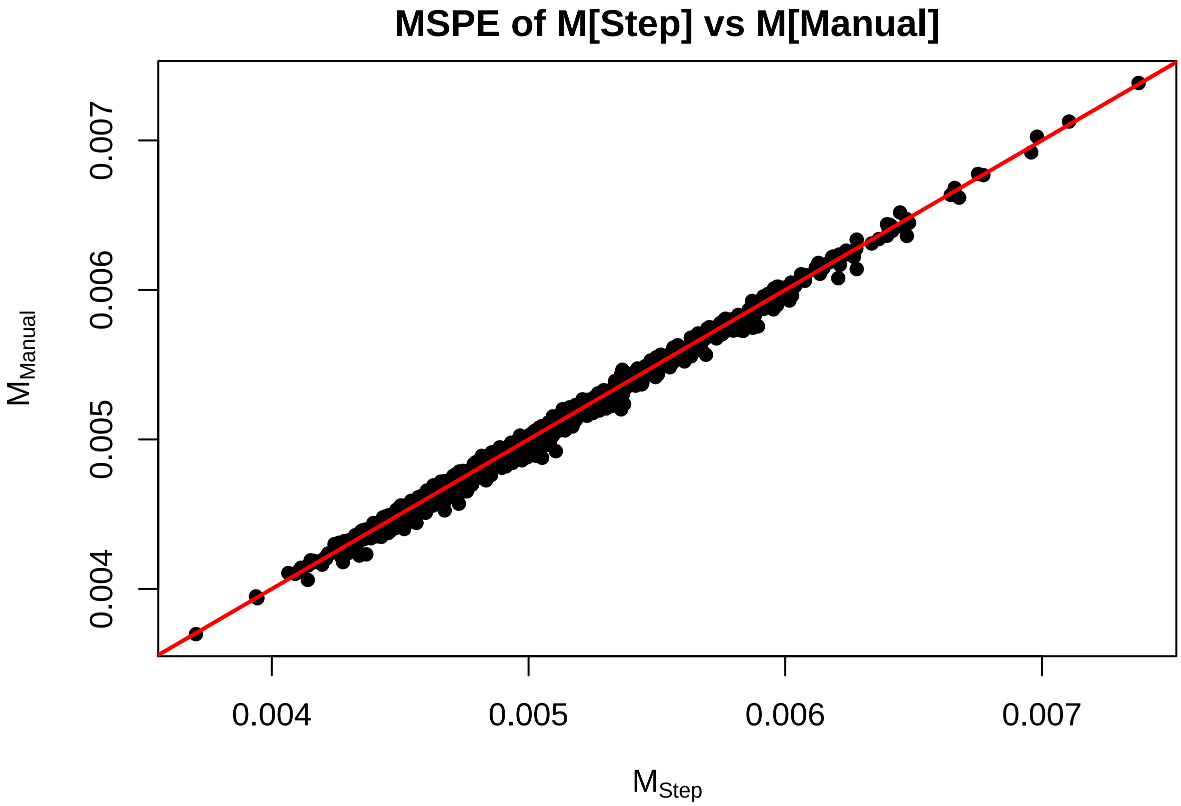
```
library(gtools) # Load this package for using the logit function

#' Following function calculates the mean of logit-normal distribution
#'
#' @param mu Mean of underlying normal distribution
#' @param sigma Standard deviation of underlying normal distribution
#'
#' @return A single number representing mean of the logit-normal distribution
#'
#' @details The calculation of w's and g(x)'s is vectorized
logitnorm_mean <- function(mu,sigma){
  v = 1/(1+ exp(-mu))           # Value passed into both shape parameters
  alpha_1 = 1/(sigma^2 * (1-v)) # Shape parameter 1
  alpha_2 = 1/(v * sigma^2) # Shape parameter 2
  # Calculate nodes and weights for Gaussian quadrature
  gqp <- gauss.quad.prob(n = 10,dist = "beta",alpha = alpha_1,beta = alpha_2)
  x <- gqp$nodes   # Extract the nodes into a vector
  w <- gqp$weights # Similarly the weights
  # Apply the function g (defined in the project description) onto the above x's
  g <- dnorm(logit(x),mean = mu,sd = sigma,log = TRUE) - log(1-x) -
       dbeta(x,shape1 = alpha_1,shape2 = alpha_2,log = TRUE)
  # Calculate and return the mean
  answer <- sum(w*exp(g))
  return(answer)
}

# For testing
mu <- c(0.7,3.2,-1.1)
sigma <- c(0.8,0.1,2.3)
sapply(1:3, function(i) logitnorm_mean(mu[i],sigma[i]))
```

```
## [1] 0.6491002 0.9606606 0.3530580
```

```
load_calcs = TRUE

# compare Mstep to Mmanual
M1 <- Mstep
M2 <- Mmanual
Mnames <- expression(M[Step], M[Manual])

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

ntot <- nrow(fhsd)   # total number of observations
ntrain <- 1800       # for fitting MLE's, roughly 80% of total
ntest <- ntot-ntrain # for out-of-sample prediction

# storage space
mspe1 <- rep(NA, nreps) # mspe for M1
mspe2 <- rep(NA, nreps) # mspe for M2

if (!load_calcs){
system.time({
  for(ii in 1:nreps) {
    train.ind <- sample(ntot, ntrain) # training observations

    # Update the models for this training set
    M1.cv <- update(M1, subset = train.ind)
    M2.cv <- update(M2, subset = train.ind)

    # MLE of sigma
    M1.sigma <- sqrt(sum(resid(M1.cv)^2)/ntrain)
    M2.sigma <- sqrt(sum(resid(M2.cv)^2)/ntrain)

    # predictions of logit(chdrisk) for test set
    predictions.M1 <- predict(M1.cv,newdata = fhsd[-train.ind,])
    predictions.M2 <- predict(M2.cv,newdata = fhsd[-train.ind,])

   # predictions of chdrisk for the test set
   values.M1 <- sapply(predictions.M1, function(i) logitnorm_mean(i,M1.sigma))
   values.M2 <- sapply(predictions.M2, function(i) logitnorm_mean(i,M2.sigma))

    M1.res <- fhsd$chdrisk[-train.ind] -   # test observations
              values.M1                    # prediction using training data
    M2.res <- fhsd$chdrisk[-train.ind] - values.M2

    # mspe for each model
    mspe1[ii] <- mean(M1.res^2)
    mspe2[ii] <- mean(M2.res^2)

  }
})
}

# the caching/loading block
if(!load_calcs) {
  saveRDS(list(mspe1 = mspe1,mspe2 = mspe2), file = "cross_validation_automated.rds")
} else {
  # just load the calculations
  tmp <- readRDS("cross_validation_automated.rds")
  mspe1 <- tmp$mspe1
  mspe2 <- tmp$mspe2
  rm(tmp) # optionally remove tmp from workspace
}

# compare Root MSPEs of both the models through boxplots
boxplot(x = list(sqrt(mspe1), sqrt(mspe2)), names = Mnames,
        main = "Root MSPE",
        ylab = expression(sqrt(MSPE)),
        col = c("yellow", "orange"))
```



```r
# compare predictions by training set
par(mar = c(5, 5, 2, 1))
plot(mspe1, mspe2, pch = 16,
     xlab = Mnames[1], ylab = Mnames[2],
     main = paste0("MSPE of ", Mnames[1]," vs ", Mnames[2])) # Fix this
abline(a = 0, b = 1, col= "red", lwd = 2)
```



6 Discussion