```
# Try to make a small program, illustrating R2 in
# linear, logistic, and Cox models
setwd("f:\\book\\R\\Rdata")
library(Hmisc)
library(Design)
library(foreign)

# Import samples from GUSTO-I
sample5 <- as.data.frame(read.spss('f:/Book/SPSS/sample5.sav'))
sample4 <- as.data.frame(read.spss('f:/Book/SPSS/sample4.sav'))
gusto <- as.data.frame(read.spss('f:/Book/SPSS/gusto.sav'))
gustoB <- as.data.frame(read.spss('f:/Book/SPSS/gustoB.sav'))

gustos  <- sample5
gustos  <- sample4  # choose one of the samples for illustration

gustos[1,]    # simple name for use during procedure
describe(gustos) # overview of the data

# Fit a full model in sample5; 17 predictors
full <- lrm(DAY30~SHO+A65+ANT+DIA+HYP+HRT+TTR+SEX+PMI+HEI+WEI+HTN+SMK+LIP+PAN+FAM+ST4,
                    data=gustos,x=T,y=T,linear.predictors=F)
# Fit a model with 8 previously specified predictors
full8 <- lrm(DAY30~SHO+A65+HIG+DIA+HYP+HRT+TTR+SEX, data=gustos,x=T,y=T,linear.predictors=T)
# Fit with continuous versions of age and shock (Killip)
full8 <- lrm(DAY30~KILLIP+AGE+HIG+DIA+HYP+HRT+TTR+SEX, data=gustos,x=T,y=T,linear.predictors=T,
maxit=99)

# Internal validation by bootstrapping
val.full8 <- validate(full8, B=200)

# Look at the results
full
full8

# Make predictions for gustoB, an independent part for validation
ruleLP  <- predict(full, gustoB, type="lp")
ruleLP <- predict(full8, gustoB, type="lp")
ruleLP[1:10] # Lp values negative, on average prob needs to be around 7%
probLP         <- plogis(ruleLP) # from linear predictor to probabilities
mean(probLP) # On average prob needs to be around 7%

# Use val.prob function for validation graph; val.prob.ci function is even nicer
val.full  <- val.prob(probLP,gustoB$DAY30, pl=T,smooth=T,logistic.cal=F, g=10,lim=c(0, 0.7),
                            xlab="Predicted probability according to full model",
                            ylab="Observed probability in external
data",riskdist='predicted',cex=1, statloc=F)
text(x=c(.3,.1,.3), y=c(.6,.5,.5),
  c(paste("C (ROC)=", round(val.full[2],2),sep=""),"",
    paste("Calibration slope=", round(val.full[13],2),sep="") ),adj=1)

# Perform bw stepwise selection
bw5    <- fastbw(full8, rule="p", type="individual", sls=0.05)
bw5
# Validate by bootstrapping, including the selection procedure
val.full8stepwise <- validate(full8, bw=T,rule="p", type="individual", sls=0.05, B=200)

# Fit stepwise selected model
step5   <- lrm.fit(y=full8$y, x=as.matrix(full8$x[,bw5$factors.kept]))
ruleLP5<- predict(step5, gustoB, type="lp")
ruleLP5[1:10]
probLP5       <- plogis(ruleLP5)
mean(probLP5)

# Use val.prob function for validation graph; val.prob.ci function is even nicer
val5  <- val.prob(probLP5,gustoB$DAY30, pl=T,smooth=T,logistic.cal=F, g=10,lim=c(0, 0.7),
                            xlab="Predicted probability according to stepwise model",
                            ylab="Observed probability in external
data",riskdist='predicted',cex=1, statloc=F)
text(x=c(.3,.1,.3), y=c(.6,.5,.5),
  c(paste("C (ROC)=", round(val5[2],2),sep=""),"",
    paste("Calibration slope=", round(val5[13],2),sep="") ),adj=1)

# Selection with p<.50
bw50   <- fastbw(full, rule="p", sls=0.5)


#### Validation in gustoB  #####
```

```
names(gustoB)
 [1] "DAY30"  "SEX"    "AGE"    "A65"    "KILLIP" "SHO"    "DIA"    "HYP"
 [9] "HRT"    "ANT"    "PMI"    "HEI"    "WEI"    "HTN"    "SMK"    "LIP"
[17] "PAN"    "FAM"    "STE"    "ST4"    "TTR"    "ESAMP"  "GRPL"   "GRPS"
[25] "REGL"   "HIG"
XB <- as.matrix(gustoB[,2:21])

# Order:                      "SEX" "AGE"  "A65" "KILLIP" "SHO"  "DIA"   "HYP"   "HRT"  "ANT"
"PMI" "HEI" "WEI" "HTN" "SMK"  "LIP" "PAN" "FAM" "STE" "ST4"  "TTR"
XB[1,]
rule   <- -9.9  +XB %*% (c(0, 0.063, 0   , 1.28    ,    0,   0    ,   1.29,    0,       0 , 0 ,
0 ,  0    , .78  , 0 ,0     ,0   , 0,    0 ,  1.40 ,   0))
rule[1:10]
prob   <- plogis(rule)
mean(prob) # Should be around 7%
val.prob(prob,gustoB$DAY30,pl=T,smooth=T,logistic.cal=F, g=10,lim=c(0, 0.8),
                     xlab="Predicted probability according to Monique model",
                     ylab="Observed probability",riskdist='predicted',cex=1)

# Order:                      "SEX" "AGE"  "A65" "KILLIP" "SHO"  "DIA"   "HYP"   "HRT"  "ANT"
"PMI" "HEI" "WEI" "HTN" "SMK"  "LIP" "PAN" "FAM" "STE" "ST4"  "TTR"
rule   <- -11.2  +XB %*% (c(0, 0.07,   0,   1.26  ,     0 ,    0 ,  1.25,   0 ,      0  ,    0,
0 ,  0    ,0.86  , 0   ,0     ,0   , 0,   0.24 ,  0,  0.73))
rule[1:10]
prob   <- 1/(1+exp(-rule))
mean(prob)
val.prob(prob,gustoB$DAY30,pl=T,smooth=T,logistic.cal=F, g=10,lim=c(0, 1),
                     xlab="Predicted probability according to Huub model",
                     ylab="Observed probability",riskdist='predicted',cex=1)


# Order:                      "SEX" "AGE"  "A65" "KILLIP" "SHO"  "DIA"   "HYP"   "HRT"  "ANT"
"PMI" "HEI" "WEI" "HTN" "SMK"  "LIP" "PAN" "FAM" "STE" "ST4"  "TTR"
rule   <- -10.2 +XB %*% (c(0, 0.064, 0,    1.18,     0 ,      0 ,    1.23,    0.52 ,   0  ,   0,
0 ,  0   ,0.7  , 0   ,0     ,0   , 0,    0 ,  1.27 , 0.58))
rule[1:10]
prob   <- 1/(1+exp(-rule))
mean(prob)
val.prob(prob,gustoB$DAY30,pl=T,smooth=T,logistic.cal=F, g=10,lim=c(0, 0.8),
                     xlab="Predicted probability according to Floris 1 model",
                     ylab="Observed probability",riskdist='predicted',cex=1)

# Order:                      "SEX" "AGE"  "A65" "KILLIP" "SHO"  "DIA"   "HYP"   "HRT"  "ANT"
"PMI" "HEI" "WEI" "HTN" "SMK"  "LIP" "PAN" "FAM" "STE" "ST4"  "TTR"
rule   <- -7.8 +XB %*% (c(0, 0.08,     0,   0.41,     0 ,      0.62 ,  0.8,   0.88,   0.56, 0.51,
0 , -0.02  , 0  , 0   , 0     ,0   , 0,    -.11 , 0 , 0.68))
rule[1:10]
prob   <- 1/(1+exp(-rule))
mean(prob)
val.prob(prob,gustoB$DAY30,pl=T,smooth=T,logistic.cal=F, g=10,lim=c(0, 0.8),
                     xlab="Predicted probability according to Floris 2 model",
                     ylab="Observed probability",riskdist='predicted',cex=1)

# Order:                   "SEX" "AGE"  "A65" "KILLIP" "SHO"  "DIA"   "HYP"   "HRT"  "ANT" "PMI"
"HEI" "WEI" "HTN" "SMK"  "LIP" "PAN" "FAM" "STE" "ST4" "TR"
rule   <- -10 +XB %*% (c(0,    0.073, 0,    1.2,     0 ,      0 ,    1.25,  0,     0,    0,    0,
0,    0.89   ,0    ,0   , 0,    0.21 ,  0,  0.66))
rule[1:10]
prob   <- 1/(1+exp(-rule))
mean(prob)
val.prob(prob,gustoB$DAY30,pl=T,smooth=T,logistic.cal=F, g=10,lim=c(0, 1),
                     xlab="Predicted probability according to UMCU model",
                     ylab="Observed probability",riskdist='predicted',cex=1)



############ OLD MATERIAL #######
# Bootstrap validation
val.full <- validate(full8,B=200)
val.full
val.bw5 <- validate(full8,B=200, bw=T,rule="p", type="individual", sls=0.05)
val.bw5
val.bw50 <- validate(full8,B=100, bw=T,rule="p",type="individual", sls=0.5)
val.bw50
val.bw5.cheat <- validate(step5,B=200)
val.bw5.cheat
# AIC
```

```
val.bwAIC <- validate(full8,B=200, bw=T,rule="aic")
val.bwAIC
val.bwAIC.cheat <- validate(step5,B=200)
val.bwAIC.cheat # validation without replaying selection per bootstrap sample

### Random permutations of gustos (sample5 or sample4) to approximate Ye's procedure
### First: Harrell's: method="random" in validate
val.fullr <- validate(full8,B=200,method="random")
val.fullr   # validation results randomly permuted
val.bw5r <- validate(full8,B=200, bw=T,rule="p", sls=0.05, method="random")
val.bw5r    # validation results randomly permuted

### Second: own program
R2sum <- R2sum5 <- j <- 0      # starting values
for (i in 1:1000) {       # repeat many times
gustos$DAY30r  <- sample(gustos$DAY30)     # outcome permuted
full8 <- lrm(DAY30r~SHO+A65+HIG+DIA+HYP+HRT+TTR+SEX,
          data=gustos,x=T,y=T,linear.predictors=F) # 8 predictor model
bw5    <- fastbw(full8, rule="p", sls=0.05)          # bw stepwise selection
R2sum  <- R2sum + full8$stat[10]                    # always calculate R2 full
if (length(bw5$factors.kept>0)) {                   # often no predictors selected
fitbw  <- lrm.fit(y=full8$y, x=as.matrix(full8$x[,bw5$factors.kept]))
R2sum5 <- R2sum5 + fitbw$stat[10]
j <- j+1  }                # if model fitted, add R2, keep track of number of fits

if (i%%10==0)                                       # print only per 10
cat(i,j, "Full R2",full8$stat[10], "Step R2",fitbw$stat[10],
    "Mean R2", R2sum / i, "Unconditional mean step R2", R2sum5 / i,
                       "Conditional mean step R2", R2sum5 / j, "\n")
                } # end for loop


## ROC curves ###
library(ROCR)
pred <- prediction(full8$linear.predictors, full8$y)
perf <- performance(pred,"tpr","fpr")
plot(perf)
lines(x=c(0,1), y=c(0,1), type='l')
perfauc <- performance(pred,"auc")
text(x=.1,y=.9,paste("AUC (c):",round(slot(perfauc,"y.values")[[1]],2)),adj=0)

## END OLD MATERIAL #######
```