1. I started to perform cross-validation with the SAheart dataset to see its output. With this code, I transferred the information of famhist to a series of 1's and 0's and placed that information in the last row.

```
> SAh = SAheart
> nrow(SAh)
[1] 462
> famhist = rep(1,462)
> famhist[SAh[,5]=="Absent"]=0
> famhist[1:10]
 [1] 1 0 1 1 1 1 0 1 1 1
> SAh1 = cbind(SAh,famhist)
```

After checking the output, I deleted the original famhist column.

```
> SAh1 = SAh1[,-5]
> head(SAh1)
  sbp tobacco  ldl adiposity typea obesity alcohol age chd famhist
1 160   12.00 5.73     23.11    49   25.30   97.20  52   1       1
2 144    0.01 4.41     28.61    55   28.87    2.06  63   1       0
3 118    0.08 3.48     32.28    52   29.14    3.81  46   0       1
4 170    7.50 6.41     38.03    51   31.99   24.26  58   1       1
5 134   13.60 3.50     27.78    60   25.99   57.34  49   1       1
6 132    6.20 6.47     36.21    62   30.77   14.14  45   0       1
```

Now I will perform cross-validation on the SAheart dataset.

```
> SHaCV.out = bestglm(SAh1,IC="CV",family=binomial)
Morgan-Tatar search since family is non-gaussian.
> SHaCV.out
CVd(d = 373, REP = 1000)
BICq equivalent for q in (0.0703191746407851, 0.895940044901826)
Best Model:
             Estimate  Std. Error   z value     Pr(>|z|)
(Intercept) -1.75407801 0.330944614 -5.300216 1.156656e-07
age          0.02485654 0.007500281  3.314082 9.194469e-04
chd          0.91518643 0.216342727  4.230262 2.334192e-05
```

Now that cross-validation was performed, I will use BIC to determine if both methods agree.

```
> SHaBIC.out = bestglm(SAh1,IC="BICq",family=binomial,TopModels=1)
Morgan-Tatar search since family is non-gaussian.
> SHaBIC.out
BICq(q = 0.25)
BICq equivalent for q in (0.0703191746407851, 0.895940044901826)
Best Model:
             Estimate  Std. Error   z value     Pr(>|z|)
(Intercept) -1.75407801 0.330944614 -5.300216 1.156656e-07
age          0.02485654 0.007500281  3.314082 9.194469e-04
chd          0.91518643 0.216342727  4.230262 2.334192e-05
```

According to the data, both methods agree. age I also observed the speed at which BIC processed the information was substantially quicker.

2. Now I am going to perform cross-validation and the BIC method for the Cardiac dataset. First I selected the 1st and 2nd as my x-variables, and selected the 24th in as my y-variable.

```
> CCV = cbind(Cardiac[,1:2],Cardiac[,24,drop=FALSE])
> head(CCV)
       bhr    basebp gender
1 0.5476190 0.8728814      0
2 0.3690476 1.1779661      0
3 0.3690476 1.1779661      0
4 0.5535714 1.0000000      1
5 0.5297619 0.8728814      0
6 0.3452381 0.8474576      0
```

First, I will use cross-validation for this data frame:

```
> CardiacCV.out = bestglm(CCV,IC="CV",family=binomial)
Morgan-Tatar search since family is non-gaussian.
> CardiacCV.out
CVd(d = 454, REP = 1000)
BICq equivalent for q in (0.109003416087217, 0.846952882132143)
Best Model:
             Estimate Std. Error   z value   Pr(>|z|)
(Intercept) -1.435895  0.5889720 -2.437968 0.01477008
basebp       1.634575  0.5130224  3.186168 0.00144171
```

Now I will use BIC and compare the two methods:

```
> CardiacBIC.out = bestglm(CCV,IC="BICq",family=binomial,TopModels=1)
Morgan-Tatar search since family is non-gaussian.
> CardiacBIC.out
BICq(q = 0.25)
BICq equivalent for q in (0.109003416087217, 0.846952882132143)
Best Model:
             Estimate Std. Error   z value   Pr(>|z|)
(Intercept) -1.435895  0.5889720 -2.437968 0.01477008
basebp       1.634575  0.5130224  3.186168 0.00144171
```

As mentioned before, both data is identical with BIC processing the data much faster.

3. I will use my BIC output from the SAheart dataset to create a classification space.

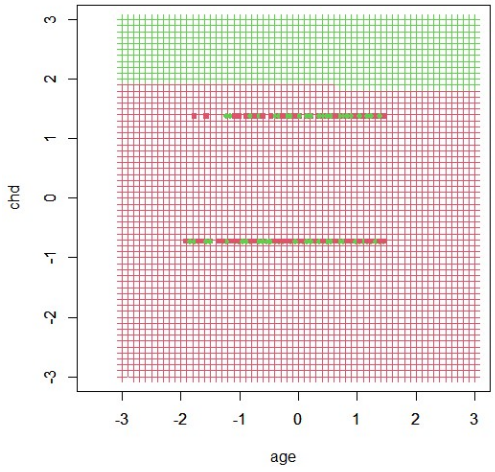
```
> SBest.out = bestglm(SAhl,IC="BICq",family=binomial)
Morgan-Tatar search since family is non-gaussian.
> SBest.out
BICq(q = 0.25)
BICq equivalent for q in (0.0703191746407851, 0.895940044901826)
Best Model:
               Estimate  Std. Error   z value     Pr(>|z|)
(Intercept) -1.75407801 0.330944614 -5.300216 1.156656e-07
age          0.02485654 0.007500281  3.314082 9.194469e-04
chd          0.91518643 0.216342727  4.230262 2.334192e-05
> glm.fit = glm(famhist~age+chd,data=SAhl,family=binomial)
> newdata=ProbeX
```

After, I created all the variables needed to make it work:

```
> X = SAh[,c(9,10)]
> head(X)
  age chd
1  52   1
2  63   1
3  46   0
```

```
> StdX = apply(X,2,scale)
> head(StdX)
            age         chd
[1,] 0.6286543  1.3723755
[2,] 1.3816170  1.3723755
[3,] 0.2179473 -0.7270864
[4,] 1.0393612  1.3723755
[5,] 0.4233008  1.3723755
[6,] 0.1494961 -0.7270864
> glm.probs = predict(glm.fit,newdata=dfX,type="response")
> length(glm.probs)
[1] 462
> famhist = SAhl[,10]
> StCard2 = as.data.frame(cbind(dfx,famhist))
> head(StCard2)
        age         chd famhist
1 0.6286543  1.3723755       1
2 1.3816170  1.3723755       0
3 0.2179473 -0.7270864       1
4 1.0393612  1.3723755       1
5 0.4233008  1.3723755       1
6 0.1494961 -0.7270864       1
> glm.fit2 = glm(famhist~age+chd,data=StCard2,family=binomial)
> names(dfX)
[1] "age" "chd"
> names(StProbeX)
[1] "dobEF" "phat"
> names(StProbeX)[1]="age"
> names(StProbeX)[2]="chd"
> glm.probe = predict(glm.fit,newdata=StProbeX,type="response")
> length(glm.probe)
[1] 3720
> glm.y = glm.probe
> glm.y[glm.probe>0.5]=1
> glm.y[glm.probe<0.5]=0
> ProbeGlm(ProbeX=StProbeX,ProbeYhat=c(glm.y),InX=dfX,InY=SAhl$
```

After following all the instructions, I received this classification space and plots:



I have no idea why this happened. Honestly, I'm not sure what to tweak to get a better result.

Appendix

```
points(dvnlr[,c(3,4)],col=risk.col,pch=100)
history(max.show=200)
save.image("C:\\Users\\whall\\Google Drive\\1 CUNY WORK\\0
BARUCH\\2020 Summer Baruch\\CIS3940\\WEEKLY
LECTURES\\LN7.RData")
q()
getwd()
install.packages("bestglm")
library(bestglm)
data(SAheart)
head(SAheart)
out = bestglm(SAheart,IC="BICq",t=1,family=binomial)
out
out = bestglm(SAheart,IC="BICq",family=binomial)
out
head(Hospital)
Xy = Hospital[,-1]
nrow(Xy)
head(Xy)
y=rep(1,50)
head(y)
y[Xy[,9]=="McCain"]=0
y[1:10]
Xy1=cbind(Xy,y)
head(Xy1)
Xy1=xY1[,-9]
Xy1=Xy1[,-9]
out = bestglm(Xy1,IC="BICq",family=binomial)
out
out = bestglm(Xy1,IC="BICq",TopModels=1)
out = bestglm(Xy1,IC="BICq",family=binomial,TopModels=1)
out
out$Susets$BICq
out$Subsets$BICq
plot(out$Subsets$BICq,type="b",xlab=Number of X-
Variables",ylab="BICq",main="Bias-Variance Trade-off")
```

```
plot(out$Subsets$BICq,type="b",xlab="Number of X-
Variables",ylab="BICq",main="Bias-Variance Trade-off")
bestglm
out$Subsets$BICq
GLM.fit=bestglm(Xy1,IC="BICq",family=binomial,TopModels=25)
y = GLM.fit$BestModels
x = apply(y[1:8],1,sum)
plot(x,y[,9],xlab="#of regressors",ylab="BICq")
dim(Cardiac)
head(Cardiac[,1:23])
plot(Cardiac[,1:23])
plot(Cardiac[,1:12])
Cardiac.Scrub15 = Cardiac.Scrub19[,4:19]
CV.out = bestglm(Cardiac.Scrub15,IC="CV",family=binomial)
CV.out
best.out = bestglm(Cardiac.Scrub15,IC="BICq",family=binomial,TopModels=1)
best.out
junk1 = as.data.frame(scale(Cardiac[,c(1,2,19)]))
head(junk1)
y=cbind(Cardiac[,1:2],Cardiac[,19])
head(y)
dim(y)
class(y)
y=cbind(Cardiac[,1:2],Cardiac[,19,drop=FALSE])
head(y)
pairs(y)
which.max(Cardiac[,1])
pairs(y)
head(Dcardiac.LGT)
glm.fit = glm(hardness~.,data=Dcardiac.LGT,family=binomial)
Dcardiac.LGT=cbind(Cardiac[,1:2],Cardiac[,19,drop=FALSE])
glm.fit = glm(hardness~.,data=Dcardiac.LGT,family=binomial)
glm.probs=predict(glm.fit,type="response")
glm.probs[1:5]
glm.forecast = Dcardiac.LGT$hardness
glm.forecast[glm.probs>0.5]=1
glm.forecast[glm.probs<0.5]=0
table(glm.forecast,Dcardiac.LGT$hardness)
best.out2 = bestglm(Cardiac.Scrub15,IC="BICq",family=binomial)
best.out2
glm.fit2 = glm(hardness~dobEF+phat,data=Cardiac.Scrub15,family=binomial)
head(glm.fit2)
newdata2 = ProbeX
X = Cardiac.Scrub15[,c(12,13)]
```

```
head(X)
StdX = apply(X,2,scale)
head(StdX)
class(StdX)
dfX = as.data.frame(StdX)
class(dfX)
glm.probs = predict(glm.fit,newdata=dfX,type="response")
glm.fit = glm(hardness~dobEF+phat,data=Cardiac.Scrub15,family=binomial)
glm.probs = predict(glm.fit,newdata=dfX,type="response")
lenght(glm.probs)
length(glm.probs)
head(SAHeart)
head(SAheart)
SAh = nrow(SAheart)
head(SAh)
head(SAheart)
SAh = SAheart
nrow(SAh)
y = rep(1,462)
y[SAh[,5]=="Present"]=1
y[1:10]
head(SAh)
y[SAh[,5]=="Absent"]=0
y[1:10]
SAh1 = cbind(SAh,y)
head(SAh1)
head(SAh1,10)
SAh = SAheart
nrow(SAh)
famhist = rep(1,462)
famhist[SAh[,5]=="Absent"]=0
famhist[1:10]
SAh1 = cbind(SAh,famhist)
head(SAh1)
SAh1 = SAh1[,-5]
head(SAh1)
SHaCV.out = bestglm(Sah1,IC="CV",family=binomial)
SHaCV.out = bestglm(SAh1,IC="CV",family=binomial)
SHaCV.out
SHaBIC.out = bestglm(SAh1,IC="BICq",family=binomial,TopModels=1)
SHaBIC.out
plot(SHaBIC.out)
SHaBIC.out = bestglm(SAh1,IC="BICq",family=binomial,TopModels=25)
SHaBIC.out
```

```
dim(Cardiac)
head(Cardiac)
head(Cardiac[,23])
head(Cardiac[,1:23])
head(Cardiac[,1:25])
head(Cardiac[,1:27])
head(Cardiac[,1:24])
CCV = Cardiac.Scrub15[,1:14]
head(CCV)
nrow(CCV)
gender = rep(1,557)
gender[CCV[,24]]
gender[CCV[,24]==0]=0
gender[CCV[,24]=="0"]=0
CCV = Cardiac.Scrub15[,8:23]
CCV = Cardiac.Scrub15[,8:20]
CCV = Cardiac.Scrub15[,1:14]
CCV = Cardiac.Scrub15[,-1:5]
CCV = Cardiac.Scrub15[,-1]
CCV = Cardiac.Scrub15[,-2]
CCV = Cardiac.Scrub15[,-3]
CCV = Cardiac.Scrub15[,-4]
CCV = Cardiac.Scrub15[,-5]
head(CCV)
head(Cardiac[,1:24])
head(Cardiac[,10:24])
CCV = Cardiac[,10:24]
head(CCV)
CCV = as.data.frame(scale(Cardiac[,c(1,2,24)]))
Head(CCV)
head(CCV)
CCV = cbind(Cardiac[,1:2],Cardiac[,24)])
CCV = cbind(Cardiac[,1:2],Cardiac[,24)]
CCV = cbind Cardiac[,1:2],Cardiac[,24)]
CCV = cbind(Cardiac[,1:2],Cardiac[,24,drop=FALSE])
head(CCV)
CardiacCV.out = bestglm(CCV,IC="CV",family=binomial)
CardiacCV.out
CardiacCV.out = bestglm(CCV,IC="BICq",family=binomial,TopModels=1)
CardiacCV.out = bestglm(CCV,IC="CV",family=binomial)
CardiacBIC.out = bestglm(CCV,IC="BICq",family=binomial,TopModels=1)
CardiacBIC.out
summary(Cardiac$gender)
table(Cardiac$gender)
```

```
SBest.out = bestglm(SAh1,IC="CV",family=binomial)
SBest.out = bestglm(SAh1,IC="BICq",family=binomial)
SBest.out
glm.fit = glm(famhist~age+chd,data=SAh1,family=binomial)
newdata=ProbeX
head(SAh1)
X = SAh[,c(8,9)]
head(X)
X = SAh[,c(9,10)]
head(X)
StdX = apply(X,2,scale)
head(StdX)
glm.probs = predict(glm.fit,newdata=StdX,type="response")
class(StdX)
dfx = as.data.frame(StdX)
dfX = as.data.frame(StdX)
class(dfX)
glm.probs = predict(glm.fit,newdata=dfX,type="response")
length(glm.probs)
SAh1[,11]
SAh1[,10]
head(SAh1)
famhist = SAh1[,10]
StCard2 = as.data.frame(cbind(dfx,famhist))
head(StCard2)
glm.fit2 = glm(famhist~age+chd,data=StCard2,family=binomial)
names(dfX)
names(StProbeX)
names(StProbeX)[1]="age"
names(StProbeX)[2]="chd"
glm.probe = predict(glm.fit,newdata=StProbeX,type="response")
length(glm.probe)
glm.y = glm.probe
glm.y[glm.probe>0.5]=1
glm.y[glm.probe<0.5]=0
ProbeGlm(ProbeX=StProbeX,ProbeYhat=c(glm.y),InX=dfX,InY=SAh1$famhist,
xr=c(-3.5,3),yr=c(-3,3))
history(max.show=200)
```