

NSWPCN Predictor Training

February 15, 2015

1 Preparation

```
library(survival)

## Loading required package: splines

library(glmulti)

## Loading required package: rJava
## Loading required package: methods

library(flexsurv)
library(randomForestSRC)

## Loading required package: parallel
##
## randomForestSRC 1.5.5
##
## Type rfsrc.news() to see new features, changes, and bug fixes.
##

library(reshape2)
library(plyr)
library(ggplot2)

library(MASS)
library(boot)

##
## Attaching package: 'boot'
##
## The following object is masked from 'package:survival':
##
## aml

library(timeROC)

## Loading required package: pec
## Loading required package: mvtnorm
## Loading required package: timereg

load("03_NSWPCN_subset.rda")

library(RColorBrewer)
pal = brewer.pal(4, "Dark2")
names(pal) = c("GG", "CPH", "RSF", "KMO")
```

2 Cohort selection and transformation

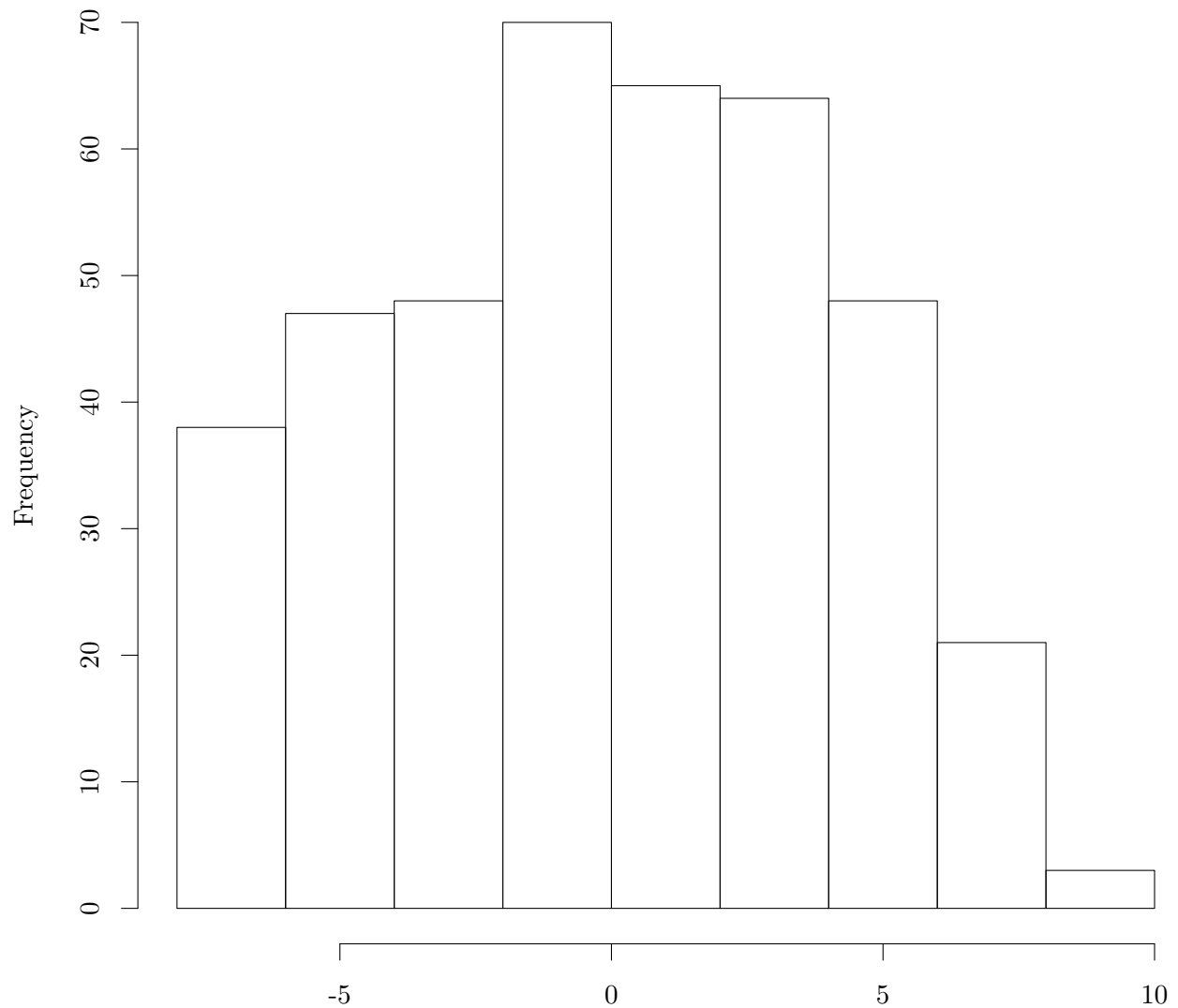
```
data$SexM = data$Patient.Sex == "M"
data$Ca199 = data$Path.Ca199.Preop > 100
data$DiagYearCent = as.numeric((data$History.Diagnosis.Date - median(data$History.Diagnosis.Date)) / 365)
data$Time = as.numeric(data$History.Death.Date - data$History.Diagnosis.Date)
data$DSD = data$History.DSDeath.Event == 1
data$AgeCent = data$History.Diagnosis.AgeAt.Cent
data$LocBody = data$Path.LocationBody
data$SizeCent = data$Path.Size.Cent
data$A2 = data$Molec.S100A2.DCThresh
data$A4 = data$Molec.S100A4.DCThresh

median(data$DiagYearCent)

## [1] 0

hist(data$DiagYearCent, main = "Histogram of Median-Centered Diagnosis Year", xlab = "")
```

Histogram of Median-Centered Diagnosis Year



```
temp = NA
temp = ls()
rm(list = temp[!(temp %in% c("pal", "data"))])

nrow(data)

## [1] 404

data = data[!is.na(data$Time) & !is.na(data$DSD) & !is.na(data$A2) & !is.na(data$A4) & !is.na(data$LocB) & !is.na(data$LocC),]
nrow(data)

## [1] 256

data = data[data$Time < 3000,] # Remove long-term survivors, which are very likely to be data censored
nrow(data)

## [1] 249
```

```

data.all = data
nrow(data.all)

## [1] 249

summary(data.all)

##      Patient.ID      Patient.Sex Cohort.ICGC      History.PreviousMalignancy
## Min.      : 4      F:126      Mode :logical      Mode :logical
## 1st Qu.: 305      M:123      FALSE:249      FALSE:227
## Median : 638      NA's :0      TRUE :22
## Mean      : 621      NA's :0
## 3rd Qu.:1031
## Max.      :1453
##
## History.FdrWithPancCancer History.FdrWithAnyCancer History.Diagnosis.Date
## Mode :logical      Mode :logical      Min.      :1994-03-09
## FALSE:239      FALSE:210      1st Qu.:1998-06-11
## TRUE :8      TRUE :39      Median :2001-07-28
## NA's :2      NA's :0      Mean      :2000-12-26
##      3rd Qu.:2003-06-26
##      Max.      :2006-08-14
##
## History.Diagnosis.AgeAt History.AlcoholLevel History.Smoking.Status
## Min.      :28.0      0:158      Never      :144
## 1st Qu.:62.0      1: 46      Ceased      : 51
## Median :69.0      2: 22      Current: 54
## Mean      :67.4      3: 23
## 3rd Qu.:75.0
## Max.      :87.0
##
## History.Smoking.PackYears History.Comorbid.Diabetes
## Min.      : 2.0      Mode :logical
## 1st Qu.:20.0      FALSE:186
## Median :27.5      TRUE :63
## Mean      :31.6      NA's :0
## 3rd Qu.:46.2
## Max.      :80.0
## NA's      :189
## History.Comorbid.ChronicPancreatitis History.Recurrence.Event
## Mode :logical      Min.      :0.00
## FALSE:238      1st Qu.:1.00
## TRUE :11      Median :1.00
## NA's :0      Mean      :0.96
##      3rd Qu.:1.00
##      Max.      :1.00
##
## History.Recurrence.Date History.DSDeath.Event History.Death.Date
## Min.      :1994-07-21      Min.      :0.000      Min.      :1995-01-12
## 1st Qu.:2000-01-08      1st Qu.:1.000      1st Qu.:1999-12-01
## Median :2002-06-03      Median :1.000      Median :2002-12-18
## Mean      :2002-03-22      Mean      :0.952      Mean      :2002-09-02
## 3rd Qu.:2005-02-04      3rd Qu.:1.000      3rd Qu.:2005-05-21
## Max.      :2009-01-29      Max.      :1.000      Max.      :2011-10-03
## NA's      :85

```

```

## History.Followup.Date History.Death.EventTimeDays Treat.Resected
## Min. :2009-10-24 Min. : 20 Mode:logical
## 1st Qu.:2009-10-24 1st Qu.: 270 TRUE:249
## Median :2009-10-24 Median : 479 NA's:0
## Mean :2009-11-30 Mean : 617
## 3rd Qu.:2009-10-24 3rd Qu.: 851
## Max. :2010-06-03 Max. :2701
## NA's :243
## Treat.ProcedureWhipple Treat.MarginPositive Treat.Chemo.Any
## Mode :logical Mode :logical Mode :logical
## FALSE:48 FALSE:145 FALSE:101
## TRUE :201 TRUE :104 TRUE :121
## NA's :0 NA's :0 NA's :27
##
##
##
## Treat.Chemo.Adjuvant Treat.Chemo.Adjuvant.GE3Cycles
## Mode :logical Mode :logical
## FALSE:175 FALSE:204
## TRUE :74 TRUE :45
## NA's :0 NA's :0
##
##
##
## Treat.Chemo.Palliative Treat.Chemo.PalliativeDC Treat.Chemo.GEM
## Mode :logical Mode :logical Mode :logical
## FALSE:1 FALSE:178 FALSE:156
## TRUE :66 TRUE :71 TRUE :92
## NA's :182 NA's :0 NA's :1
##
##
##
## Treat.Radio Path.LocationBody Path.Size Path.Bilirubin.Preop
## Mode :logical Mode :logical Min. : 8.0 Min. : 0.06
## FALSE:205 FALSE:201 1st Qu.:25.0 1st Qu.: 0.64
## TRUE :44 TRUE :48 Median :30.0 Median : 3.45
## NA's :0 NA's :0 Mean :33.6 Mean : 7.10
## 3rd Qu.:40.0 3rd Qu.:10.22
## Max. :90.0 Max. :45.03
## NA's :99
## Path.Ca199.Preop Path.Bilirubin.Postop Path.Ca199.Postop
## Min. : 1 Min. : 0.12 Min. : 1
## 1st Qu.: 67 1st Qu.: 0.47 1st Qu.: 15
## Median : 197 Median : 0.70 Median : 74
## Mean : 2701 Mean : 1.92 Mean : 1528
## 3rd Qu.: 802 3rd Qu.: 1.26 3rd Qu.: 271
## Max. :101075 Max. :25.38 Max. :31760
## NA's :168 NA's :106 NA's :143
## Path.Subtype Path.Differentiation Path.LN.Involved
## Adenosquamous: 18 1: 16 Min. : 0.00
## Large Cell : 0 2:162 1st Qu.: 0.00
## Mucinous : 5 3: 71 Median : 1.00
## NotSpecified : 39 4: 0 Mean : 1.72
## Papillary : 2 3rd Qu.: 2.00

```

```

## Tubular :185 Max. :12.00
## NA's :4
## Path.LN.Inspected Path.Invasion.Vascular Path.Invasion.Perineural
## Min. : 0.0 Mode :logical Mode :logical
## 1st Qu.: 5.0 FALSE:133 FALSE:63
## Median : 8.5 TRUE :116 TRUE :186
## Mean : 9.8 NA's :0 NA's :0
## 3rd Qu.:13.0
## Max. :52.0
## NA's :21
## Stage.pT Stage.pN Stage.pM Molec.BNIP3.NucInt Molec.BNIP3.CytoInt
## Tis: 0 N0 : 83 M0 :182 0 : 6 0 : 1
## T1 : 18 N1 :160 M1 : 9 1 :208 1 :130
## T2 : 34 NA's: 6 NA's: 58 2 : 21 2 : 76
## T3 :197 3 : 2 3 : 30
## T4 : 0 NA's: 12 NA's: 12
##
##
## Molec.CCND1.CytoLo Molec.CCND1.CytoHi Molec.CCND1.MembLo
## 0 :159 0 :75 0 :100
## 1 : 34 1 :90 1 : 71
## 2 : 4 2 :32 2 : 18
## 3 : 1 3 : 1 3 : 9
## NA's: 51 NA's:51 NA's: 51
##
##
## Molec.CCND1.MembHi Molec.Grb7.Int Molec.Grb7.Percent Molec.HCNT3PlusHENT1
## 0 :32 0 :51 Min. : 0.0 Mode :logical
## 1 :89 1 :94 1st Qu.: 3.0 FALSE:96
## 2 :46 2 :42 Median : 18.0 TRUE :98
## 3 :31 3 : 7 Mean : 31.1 NA's :55
## NA's:51 NA's:55 3rd Qu.: 55.0
## Max. :100.0
## NA's :55
##
## Molec.HENT1.Percent Molec.HENT1.Int Molec.HER2 Molec.HOXB2.Percent
## Min. : 0.0 0 : 19 Mode :logical Min. : 0.0
## 1st Qu.: 11.2 1 :117 FALSE:37 1st Qu.: 35.0
## Median : 42.5 2 : 53 TRUE :11 Median : 70.0
## Mean : 44.4 3 : 13 NA's :201 Mean : 60.8
## 3rd Qu.: 75.0 NA's: 47 3rd Qu.: 90.0
## Max. :100.0 Max. :100.0
## NA's :47 NA's :43
##
## Molec.HOXB2.Int Molec.RON.Int Molec.S100A2.Int Molec.S100A2.Percent
## 0 : 14 0 : 20 0:88 Min. : 0.0
## 1 :141 1 :111 1:63 1st Qu.: 0.0
## 2 : 36 2 : 64 2:57 Median : 10.0
## 3 : 15 3 : 10 3:41 Mean : 28.7
## NA's: 43 NA's: 44 3rd Qu.: 60.0
## Max. :100.0
##
## Molec.S100A2.StromaScore Molec.S100A4.CytoInt Molec.S100A4.CytoPercent
## Mode :logical 0:72 Min. : 0.0
## FALSE:183 1:93 1st Qu.: 0.0
## TRUE :22 2:43 Median : 10.0

```

```

## NA's :44          3:41          Mean   : 34.6
##                                     3rd Qu.: 75.0
##                                     Max.   :100.0
##
## Molec.S100A4.NucInt Molec.S100A4.NucPercent Stage.Overall
## 0:80                Min.   : 0.0          IIB    :120
## 1:68                1st Qu.: 0.0          IIA    : 43
## 2:65                Median : 5.0          IB     : 12
## 3:36                Mean    : 26.4         IV     : 9
##                    3rd Qu.: 60.0         IA     : 7
##                    Max.    :100.0        (Other): 0
##                                     NA's    : 58
## History.Death.Event Molec.S100A4.DCThresh Molec.S100A2.DCThresh
## Min.   :0.000      Mode :logical      Mode :logical
## 1st Qu.:1.000      FALSE:61          FALSE:209
## Median :1.000      TRUE :188          TRUE :40
## Mean    :0.984      NA's :0           NA's :0
## 3rd Qu.:1.000
## Max.    :1.000
##
## Stage.pT.Simplified Path.Ca199.Preop.Cent Path.Ca199.Postop.Cent
## T1 : 18            Min.   : -5.38        Min.   : -3.97
## T2 : 34            1st Qu.: -1.18        1st Qu.: -1.25
## T34:197           Median : -0.10        Median : 0.34
##                    Mean    : 0.01        Mean    : 0.57
##                    3rd Qu.: 1.31        3rd Qu.: 1.63
##                    Max.    : 6.14        Max.    : 6.40
##                    NA's    :168         NA's    :143
## History.Diagnosis.AgeAt.Cent History.Smoking.PackYears.Cent
## Min.   : -40.00      Min.   : -28.00
## 1st Qu.: -6.00      1st Qu.: -10.00
## Median : 1.00       Median : -2.50
## Mean    : -0.57     Mean    : 1.65
## 3rd Qu.: 7.00      3rd Qu.: 16.25
## Max.    : 19.00     Max.    : 50.00
##                    NA's    :189
## Path.Size.Cent      Path.Bilirubin.Preop.Cent Path.Bilirubin.Postop.Cent
## Min.   : -22.00     Min.   : -3.39      Min.   : -0.53
## 1st Qu.: -5.00     1st Qu.: -2.81     1st Qu.: -0.18
## Median : 0.00      Median : 0.00      Median : 0.06
## Mean    : 3.57     Mean    : 3.65     Mean    : 1.27
## 3rd Qu.: 10.00    3rd Qu.: 6.77     3rd Qu.: 0.61
## Max.    : 60.00    Max.    :41.58     Max.    :24.74
##                    NA's    :99          NA's    :106
## History.Diagnosis.Date.Cent Path.LN.InvolvedFraction Path.LN.Negative
## Min.   : -2867      Min.   : 0.000      Min.   : 0.00
## 1st Qu.: -1312     1st Qu.: 0.000     1st Qu.: 4.00
## Median : -169      Median : 0.143     Median : 7.00
## Mean    : -382     Mean    : 0.213     Mean    : 8.01
## 3rd Qu.: 529      3rd Qu.: 0.333     3rd Qu.:11.00
## Max.    : 1674     Max.    :1.000     Max.    :45.00
##                    NA's    :22          NA's    :21
## SexM              Ca199              DiagYearCent              Time
## Mode :logical     Mode :logical     Min.   : -7.849     Min.   : 20

```

```
## FALSE:126      FALSE:29      1st Qu.: -3.592    1st Qu.: 270
## TRUE :123      TRUE :52       Median : -0.463    Median : 478
## NA's :0        NA's :168      Mean  : -1.047     Mean   : 615
##                                     3rd Qu.: 1.448     3rd Qu.: 804
##                                     Max.   : 4.583     Max.   :2701
##
##      DSD          AgeCent      LocBody          SizeCent
## Mode :logical   Min.   : -40.00   Mode :logical   Min.   : -22.00
## FALSE:12        1st Qu.: -6.00   FALSE:201       1st Qu.: -5.00
## TRUE :237       Median :  1.00   TRUE :48        Median :  0.00
## NA's :0         Mean  : -0.57   NA's :0         Mean   :  3.57
##                 3rd Qu.:  7.00           3rd Qu.: 10.00
##                 Max.   : 19.00           Max.   : 60.00
##
##      A2          A4
## Mode :logical   Mode :logical
## FALSE:209       FALSE:61
## TRUE :40        TRUE :188
## NA's :0         NA's :0
##
##
##
```

3 Data splitting

There's going to be an awful lot of model manipulation and black magic going on. Create a holdout validation set for final model comparison and selection.

```
set.seed(20150201)
sel.val = sample.int(nrow(data), floor(nrow(data)/5))
sel.val = 1:nrow(data) %in% sel.val
mean(sel.val)

## [1] 0.1968

data.val = data[sel.val,,drop = FALSE]
data = data[!sel.val,,drop = FALSE]
nrow(data)

## [1] 200

nrow(data.val)

## [1] 49
```

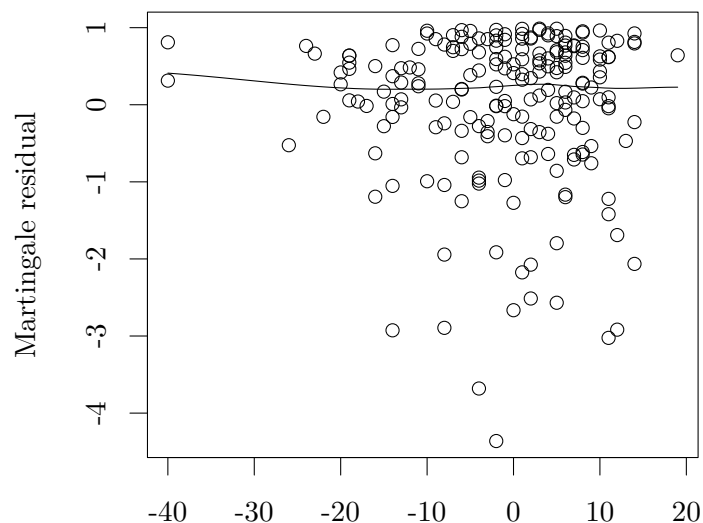
4 EDA

Use the CPH model as a convenient framework for EDA.

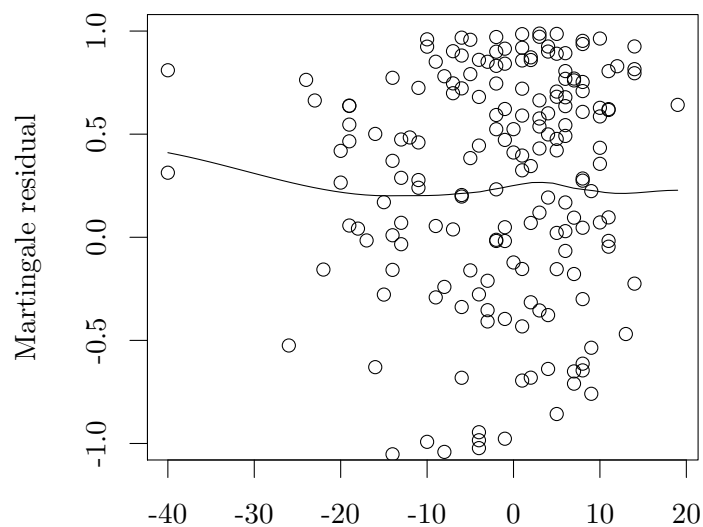
4.1 Functional form

Investigate functional form with martingale residuals.


```
fit.cph.NoAge = coxph(Surv(Time, DSD) ~ DiagYearCent + SexM + LocBody + SizeCent + A2 + A4, data = data)
scatter.smooth(data$AgeCent, resid(fit.cph.NoAge, type = "martingale"), xlab = "", ylab = "Martingale re
```

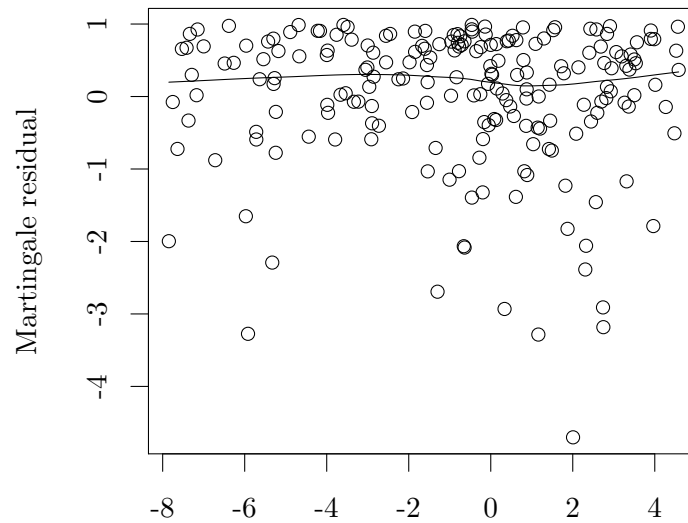


```
scatter.smooth(data$AgeCent, resid(fit.cph.NoAge, type = "martingale"), xlab = "", ylab = "Martingale re
```

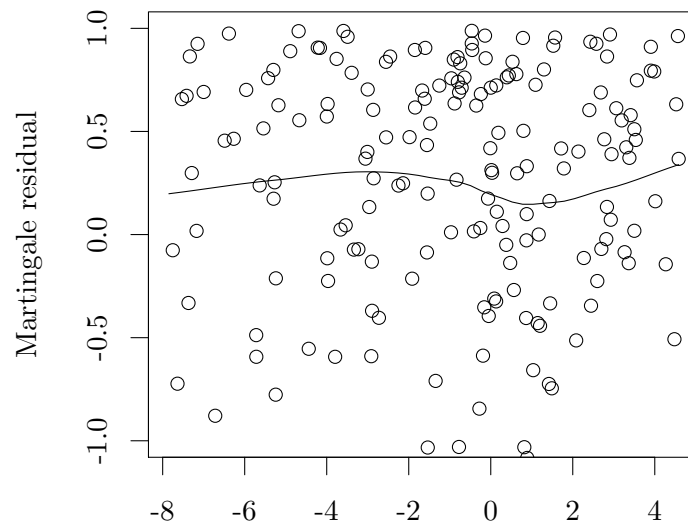


Close enough to linear.

```
fit.cph.NoDate = coxph(Surv(Time, DSD) ~ SexM + AgeCent + LocBody + SizeCent + A2 + A4, data = data)
scatter.smooth(data$DiagYearCent, resid(fit.cph.NoDate, type = "martingale"), xlab = "", ylab = "Marting
```

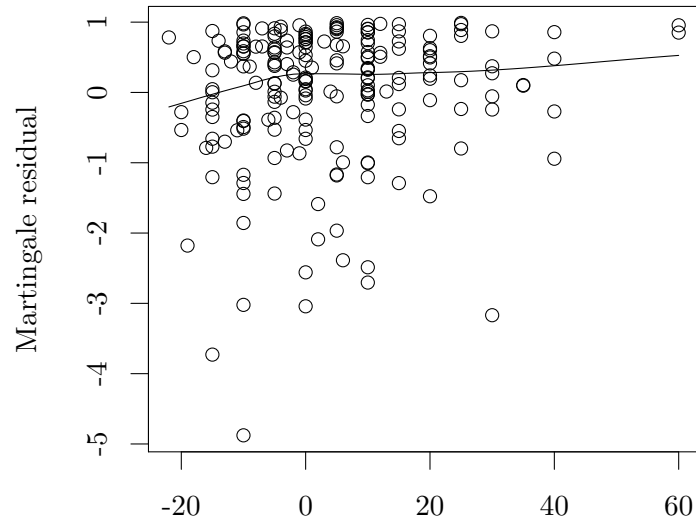


```
scatter.smooth(data$DiagYearCent, resid(fit.cph.NoDate, type = "martingale"), xlab = "", ylab = "Martingale")
```

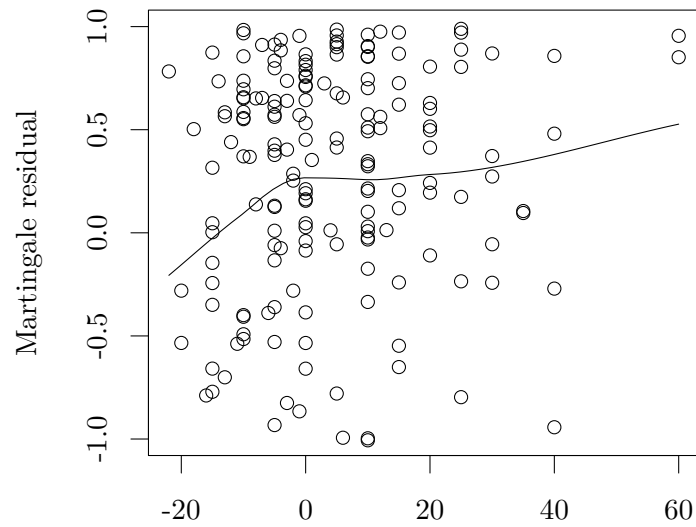


Doesn't appear to have much of an effect.

```
fit.cph.NoSize = coxph(Surv(Time, DSD) ~ DiagYearCent + SexM + AgeCent + LocBody + A2 + A4, data = data)
scatter.smooth(data$SizeCent, resid(fit.cph.NoSize, type = "martingale"), xlab = "", ylab = "Martingale")
```



```
scatter.smooth(data$SizeCent, resid(fit.cph.NoSize, type = "martingale"), xlab = "", ylab = "Martingale
```



The size relationship appears to have a knee, close to size == 0, around which the relationship is approximately linear.

Model size as: $SizeCent + SizeCentI(SizeCent > 0) \equiv SizeCent + SizeCent_+$

```
data$SizePlus = pmax(data$SizeCent, 0)
data.val$SizePlus = pmax(data.val$SizeCent, 0)
data.all$SizePlus = pmax(data.all$SizeCent, 0)
```

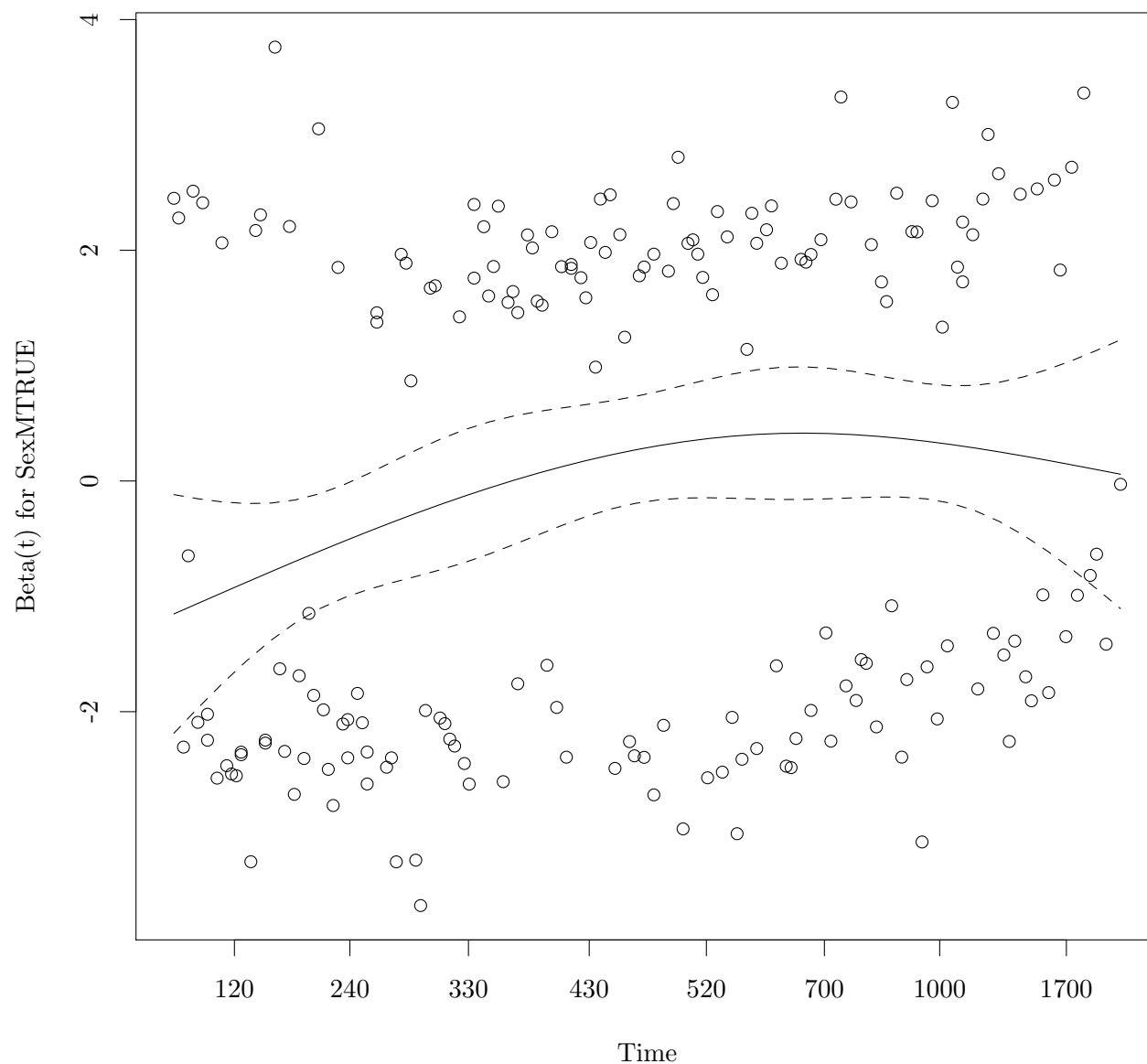
4.2 PH assumption: full model

```
fit.cph = coxph(Surv(Time, DSD) ~ SexM + AgeCent + LocBody + SizeCent + SizePlus + A2 + A4, data = data)
cox.zph(fit.cph)
```

```
##          rho    chisq      p
## SexMTRUE  0.17964  6.56115 0.0104
## AgeCent   -0.10574  2.40668 0.1208
## LocBodyTRUE -0.04856  0.37895 0.5382
```

```
## SizeCent      0.00231  0.00106 0.9740
## SizePlus     -0.01130  0.02666 0.8703
## A2TRUE       -0.03995  0.29907 0.5845
## A4TRUE       -0.08343  1.33308 0.2483
## GLOBAL              NA 13.17267 0.0680
```

```
plot(cox.zph(fit.cph)[1])
```



```
fit.cph = coxph(Surv(Time, DSD) ~ strata(SexM) + AgeCent + LocBody + SizeCent + SizePlus + A2 + A4, data)
cox.zph(fit.cph)
```

```
##           rho    chisq    p
## AgeCent    -0.11339  2.78186 0.0953
## LocBodyTRUE -0.04618  0.34177 0.5588
## SizeCent     0.00662  0.00857 0.9262
## SizePlus    -0.01329  0.03588 0.8498
## A2TRUE      -0.04361  0.35772 0.5498
```

```
## A4TRUE      -0.07985  1.25354  0.2629
## GLOBAL      NA  6.03352  0.4194
```

Using a threshold of 0.1 for the CPH tests, sex is stuffing things up. Stratification by sex makes good sense, given known variation in survival between the sexes. It would have been possible to model this with a Sex:Age term in an AFT model, but given this is CPH, a baseline change is needed.

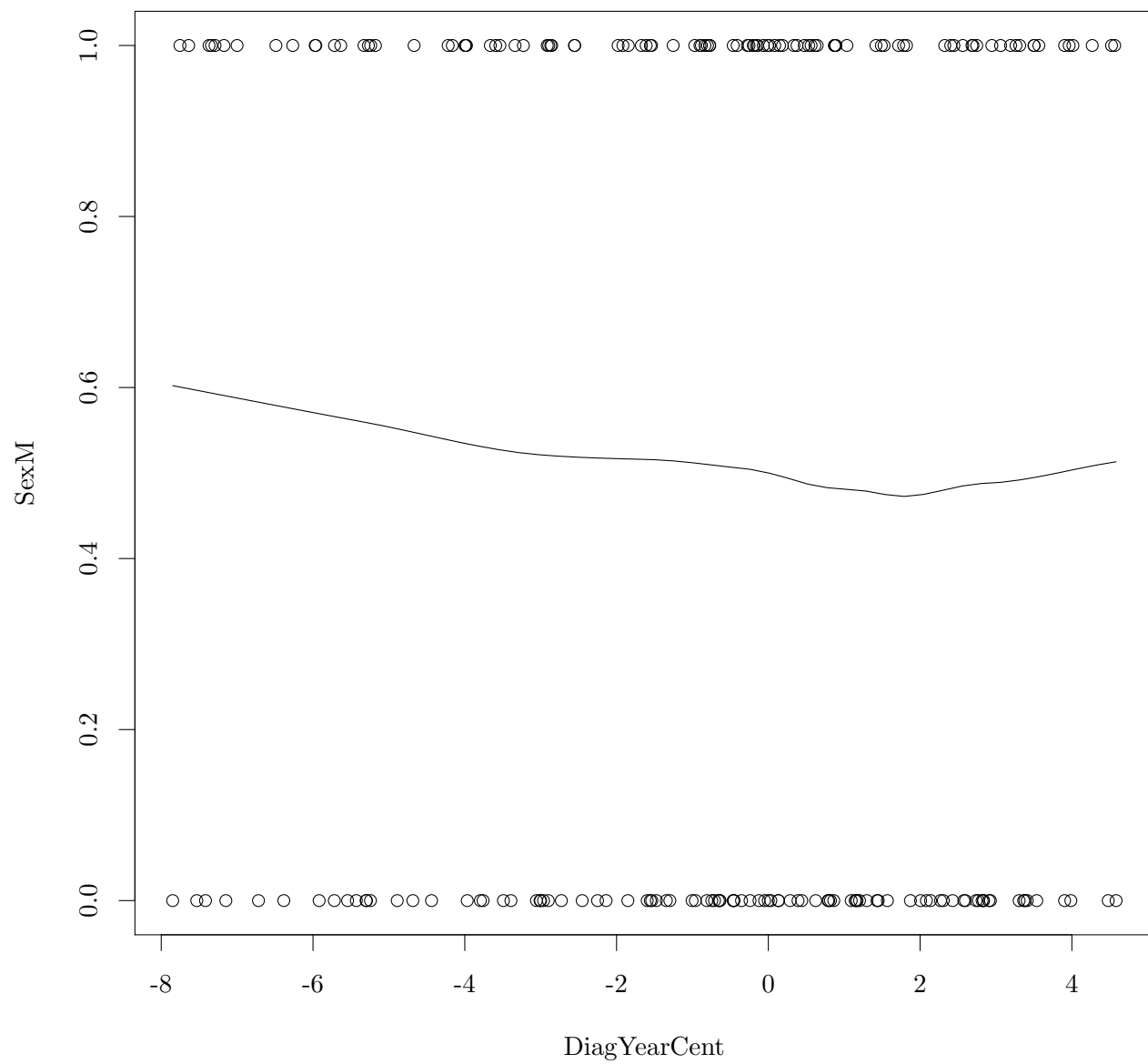
4.3 Date of diagnosis test

```
temp1 = coxph(Surv(Time, DSD) ~ strata(SexM) + AgeCent + LocBody + SizeCent + SizePlus + A2 + A4, data =
temp2 = coxph(Surv(Time, DSD) ~ strata(SexM) + AgeCent + LocBody + SizeCent + SizePlus + A2 + A4 + DiagYearCent
anova(temp1, temp2)

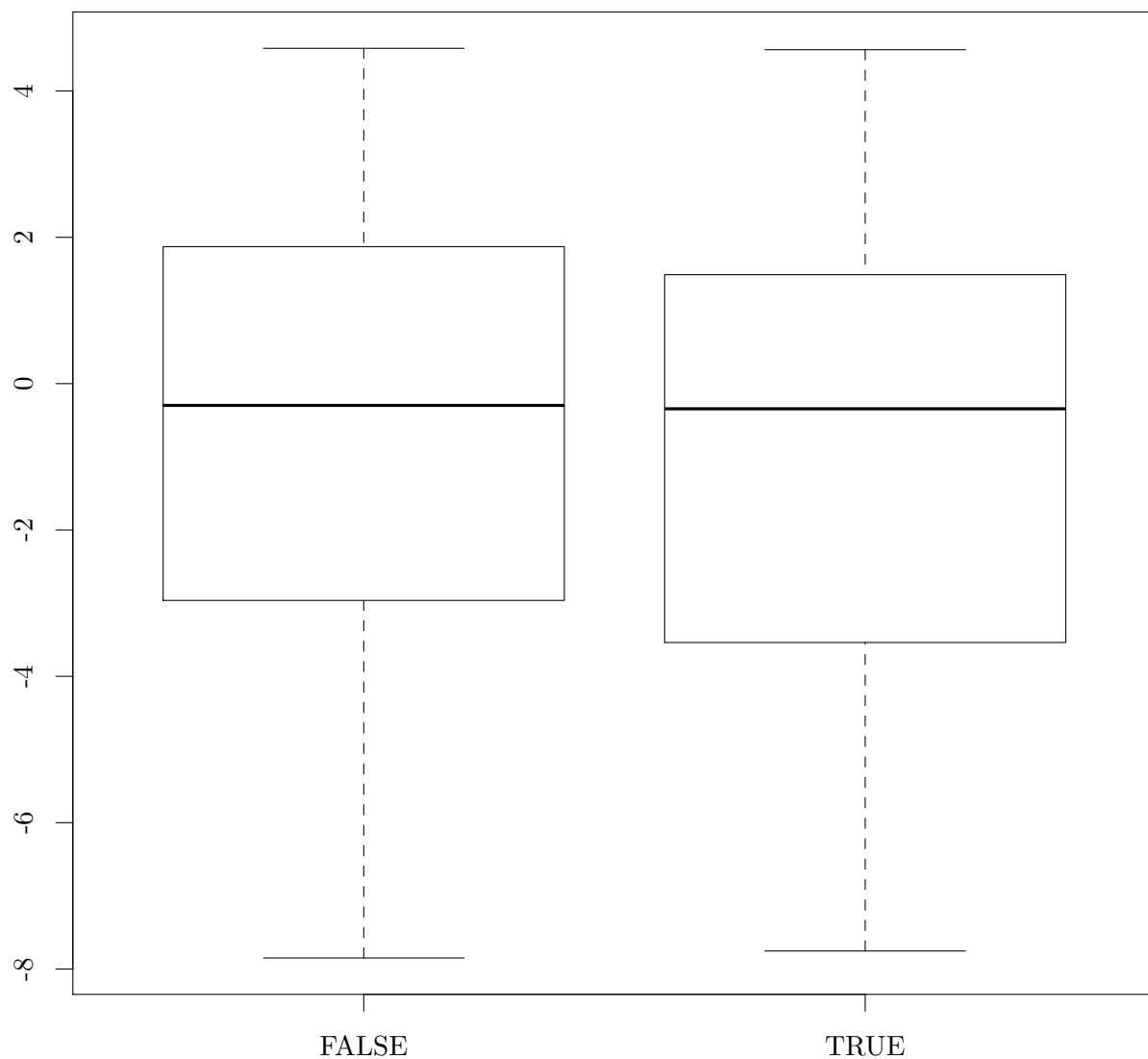
## Analysis of Deviance Table
## Cox model: response is Surv(Time, DSD)
## Model 1: ~ strata(SexM) + AgeCent + LocBody + SizeCent + SizePlus + A2 + A4
## Model 2: ~ strata(SexM) + AgeCent + LocBody + SizeCent + SizePlus + A2 + A4 + DiagYearCent
##      loglik Chisq Df P(>|Chi|)
## 1      -682
## 2      -682  0.86  1      0.35

library(energy)

scatter.smooth(data$DiagYearCent, data$SexM, xlab = "DiagYearCent", ylab = "SexM")
```



```
boxplot(DiagYearCent ~ SexM, data)
```



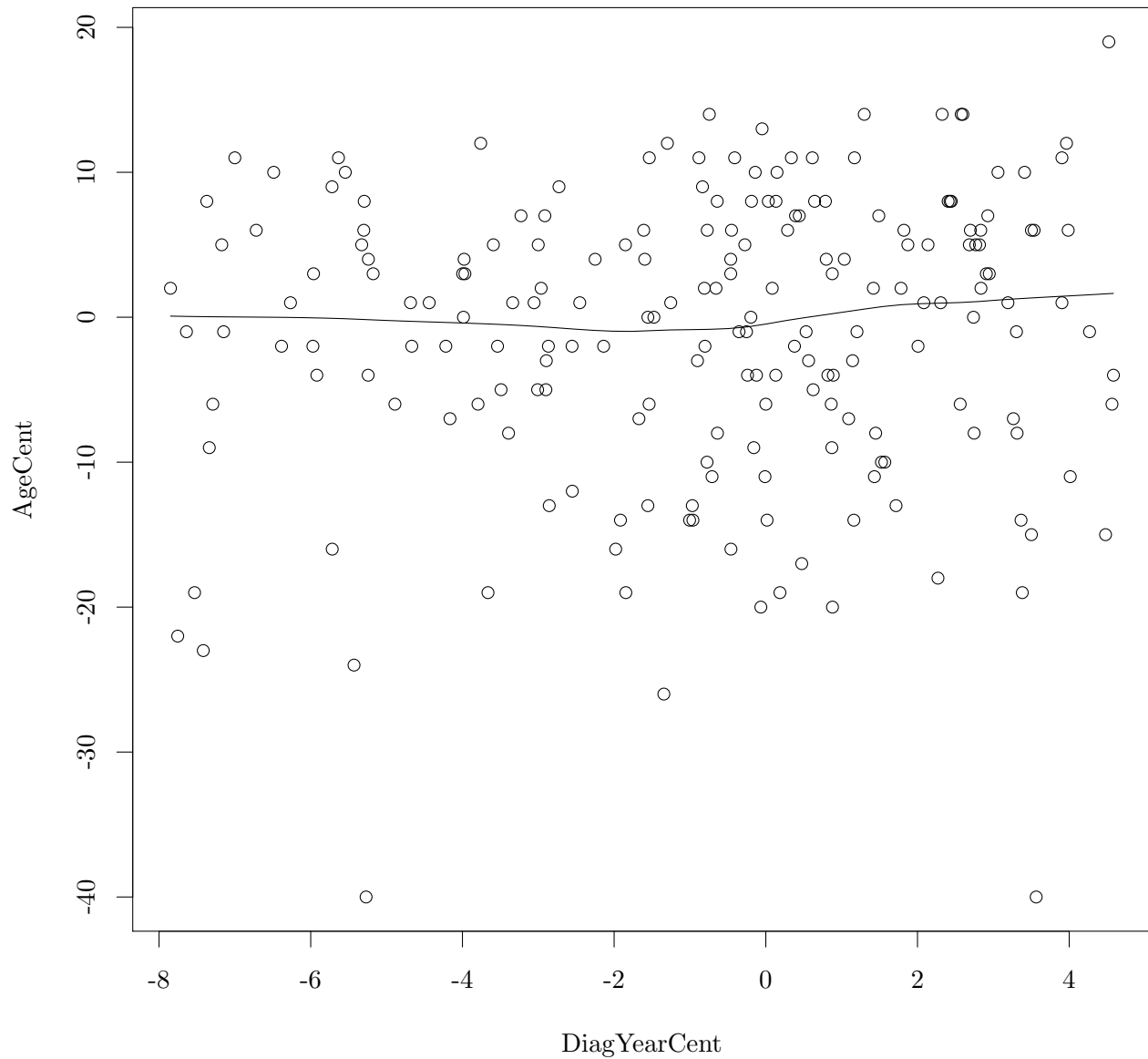
```
kruskal.test(data$DiagYearCent, data$SexM)

##
##  Kruskal-Wallis rank sum test
##
## data:  data$DiagYearCent and data$SexM
## Kruskal-Wallis chi-squared = 0.4306, df = 1, p-value = 0.5117

dcov.test(data$DiagYearCent, data$SexM, R = 499)

##
##  dCov test of independence
##
## data:  index 1, replicates 499
## nV^2 = 0.7729, p-value = 0.784
## sample estimates:
##      dCov
## 0.06217

scatter.smooth(data$DiagYearCent, data$AgeCent, xlab = "DiagYearCent", ylab = "AgeCent")
```



```
cor.test(data$DiagYearCent, data$AgeCent, method = "kendall")
```

```
##
##  Kendall's rank correlation tau
##
## data:  data$DiagYearCent and data$AgeCent
## z = 1.026, p-value = 0.3049
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##      tau
## 0.04952
```

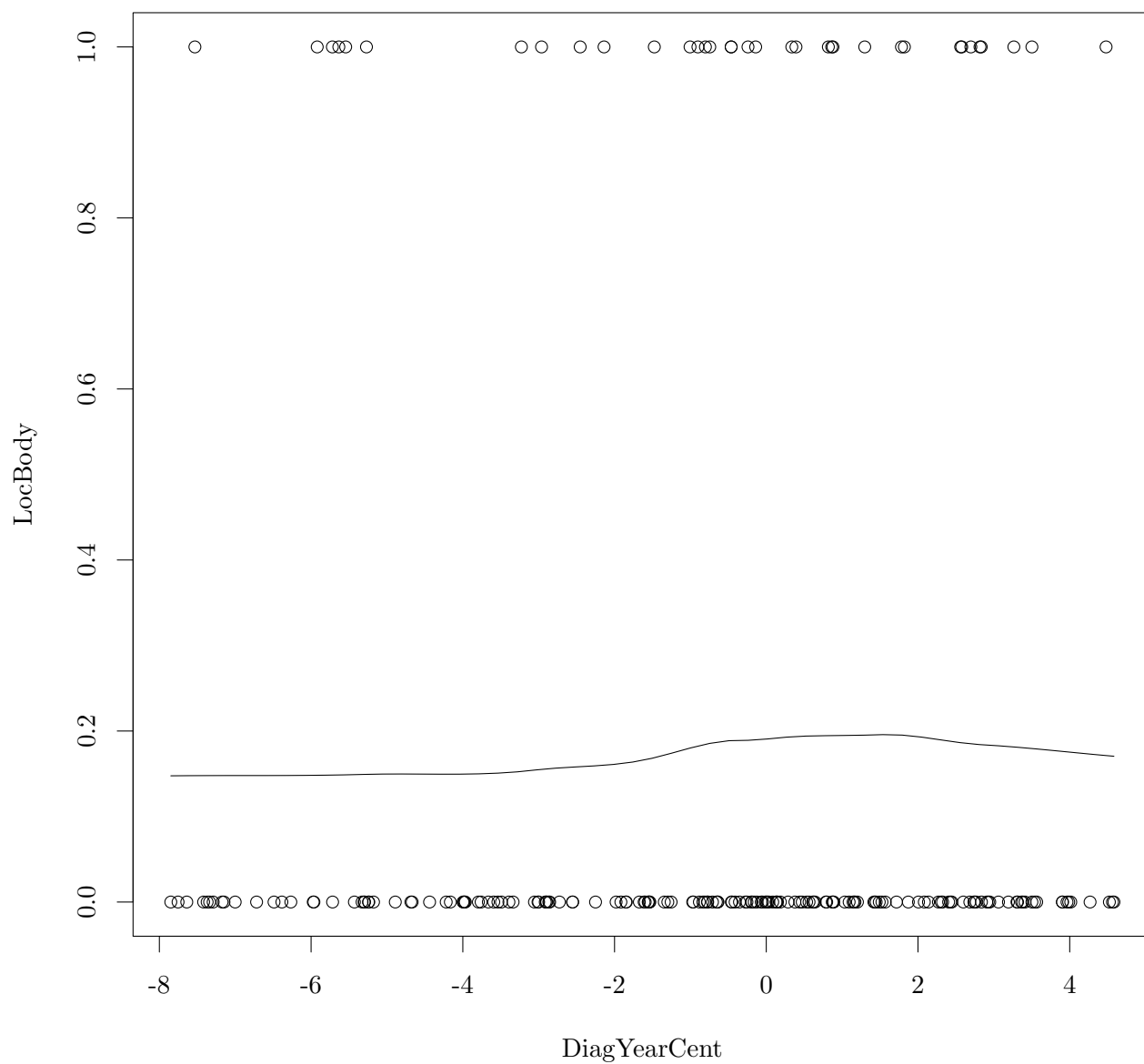
```
dcov.test(data$DiagYearCent, data$AgeCent, R = 499)
```

```
##
##  dCov test of independence
##
```

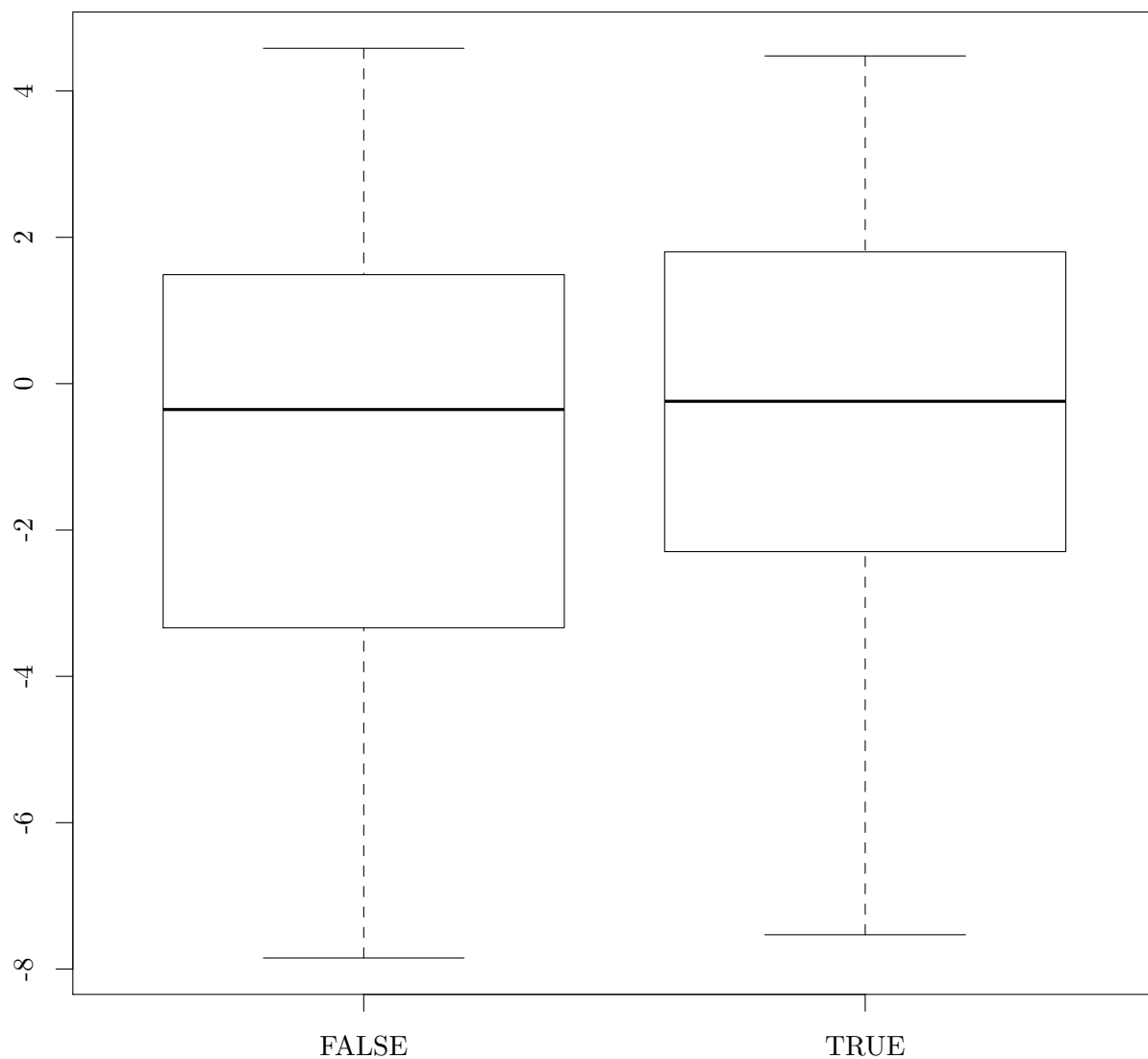


```
## data: index 1, replicates 499
## nV^2 = 36.72, p-value = 0.448
## sample estimates:
## dCov
## 0.4285
```

```
scatter.smooth(data$DiagYearCent, data$LocBody, xlab = "DiagYearCent", ylab = "LocBody")
```



```
boxplot(DiagYearCent ~ LocBody, data)
```



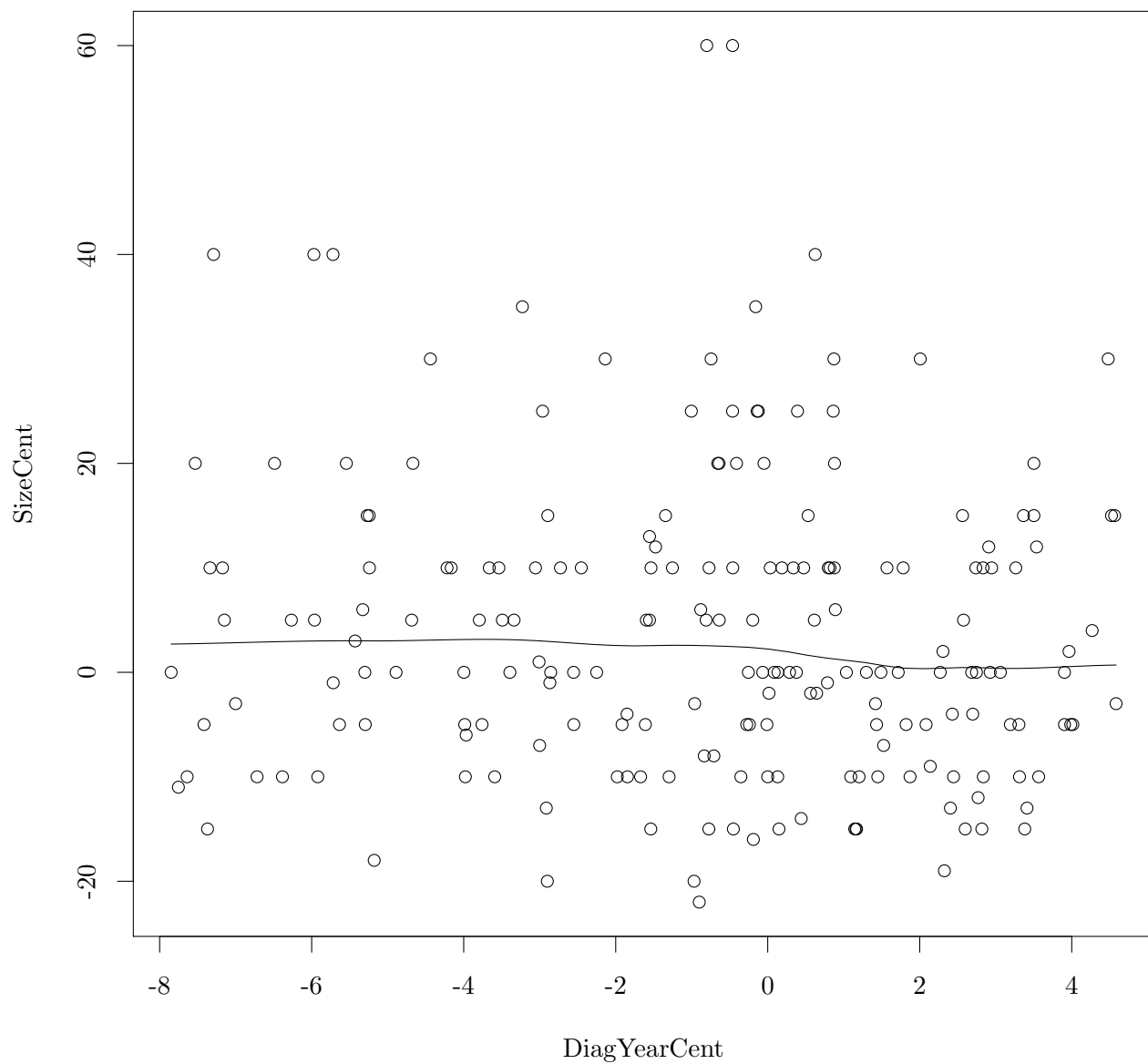
```
kruskal.test(data$DiagYearCent, data$LocBody)

##
##  Kruskal-Wallis rank sum test
##
## data:  data$DiagYearCent and data$LocBody
## Kruskal-Wallis chi-squared = 0.2357, df = 1, p-value = 0.6273

dcov.test(data$DiagYearCent, data$LocBody, R = 499)

##
##  dCov test of independence
##
## data:  index 1, replicates 499
## nV^2 = 0.4203, p-value = 0.812
## sample estimates:
##      dCov
## 0.04584

scatter.smooth(data$DiagYearCent, data$SizeCent, xlab = "DiagYearCent", ylab = "SizeCent")
```



```
cor.test(data$DiagYearCent, data$SizeCent, method = "kendall")
```

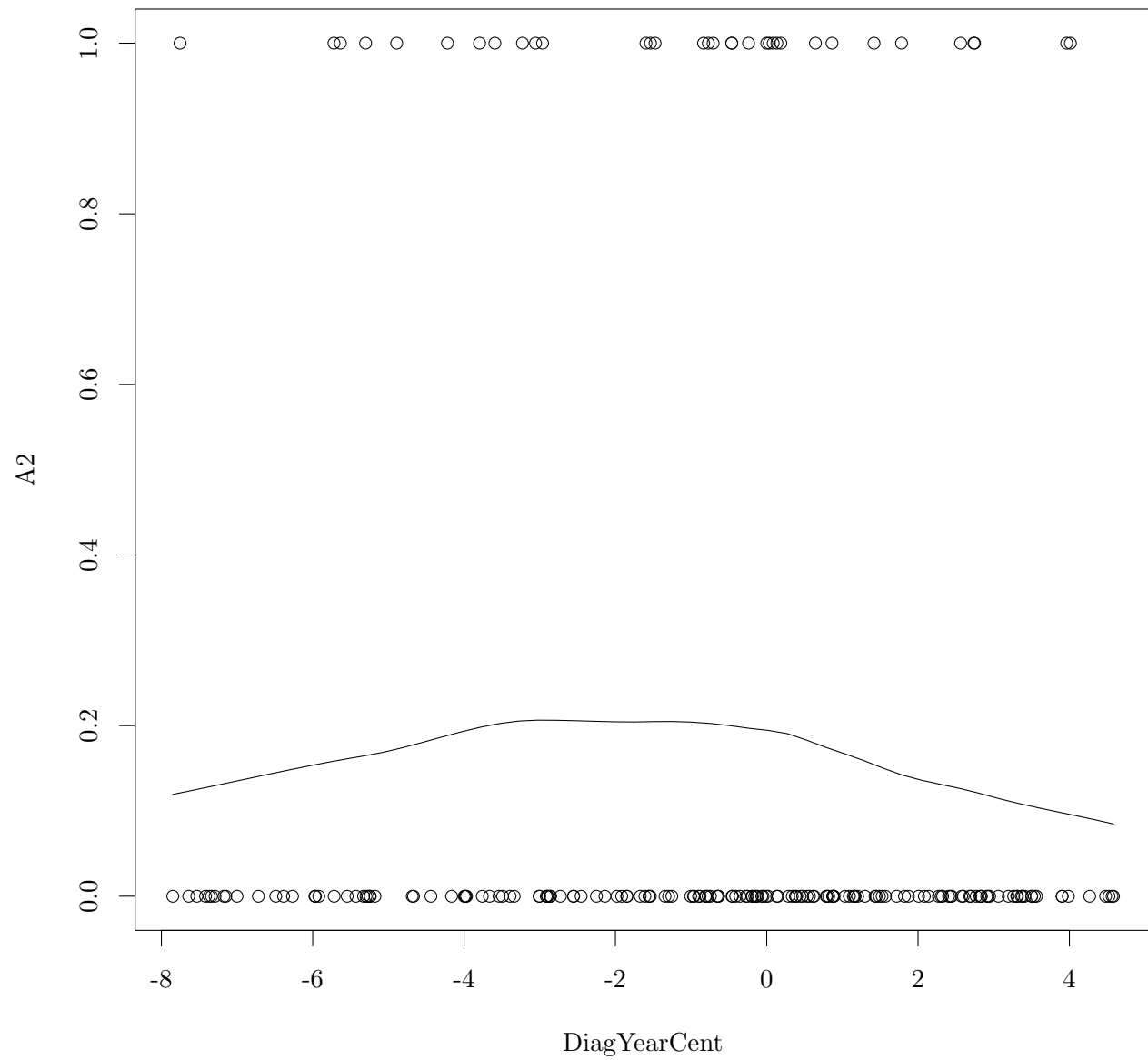
```
##
##  Kendall's rank correlation tau
##
## data:  data$DiagYearCent and data$SizeCent
## z = -1.095, p-value = 0.2737
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##      tau
## -0.05367
```

```
dcov.test(data$DiagYearCent, data$SizeCent, R = 499)
```

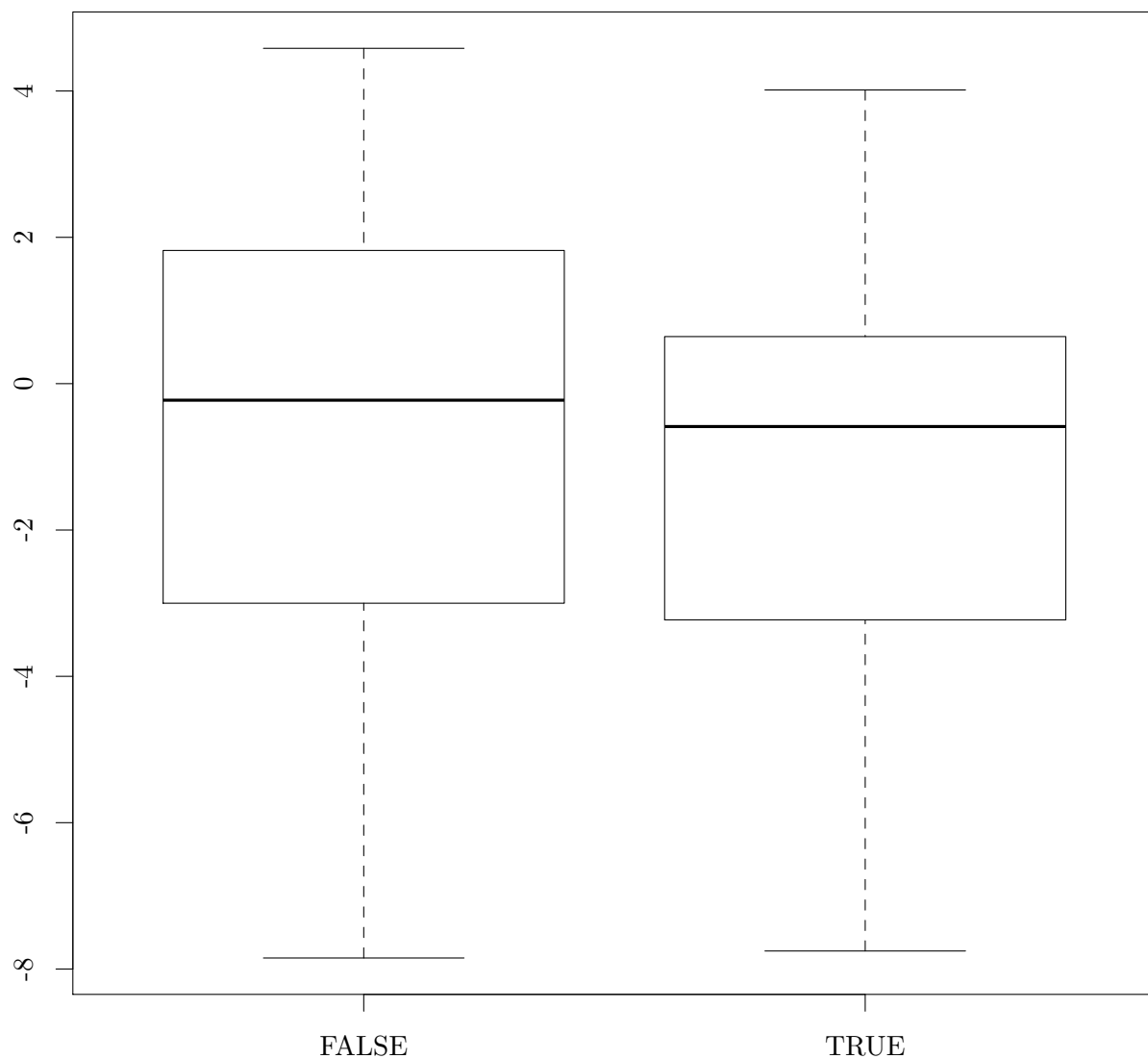
```
##
##  dCov test of independence
##
```

```
## data: index 1, replicates 499
## nV^2 = 59.67, p-value = 0.372
## sample estimates:
## dCov
## 0.5462
```

```
scatter.smooth(data$DiagYearCent, data$A2, xlab = "DiagYearCent", ylab = "A2")
```



```
boxplot(DiagYearCent ~ A2, data)
```



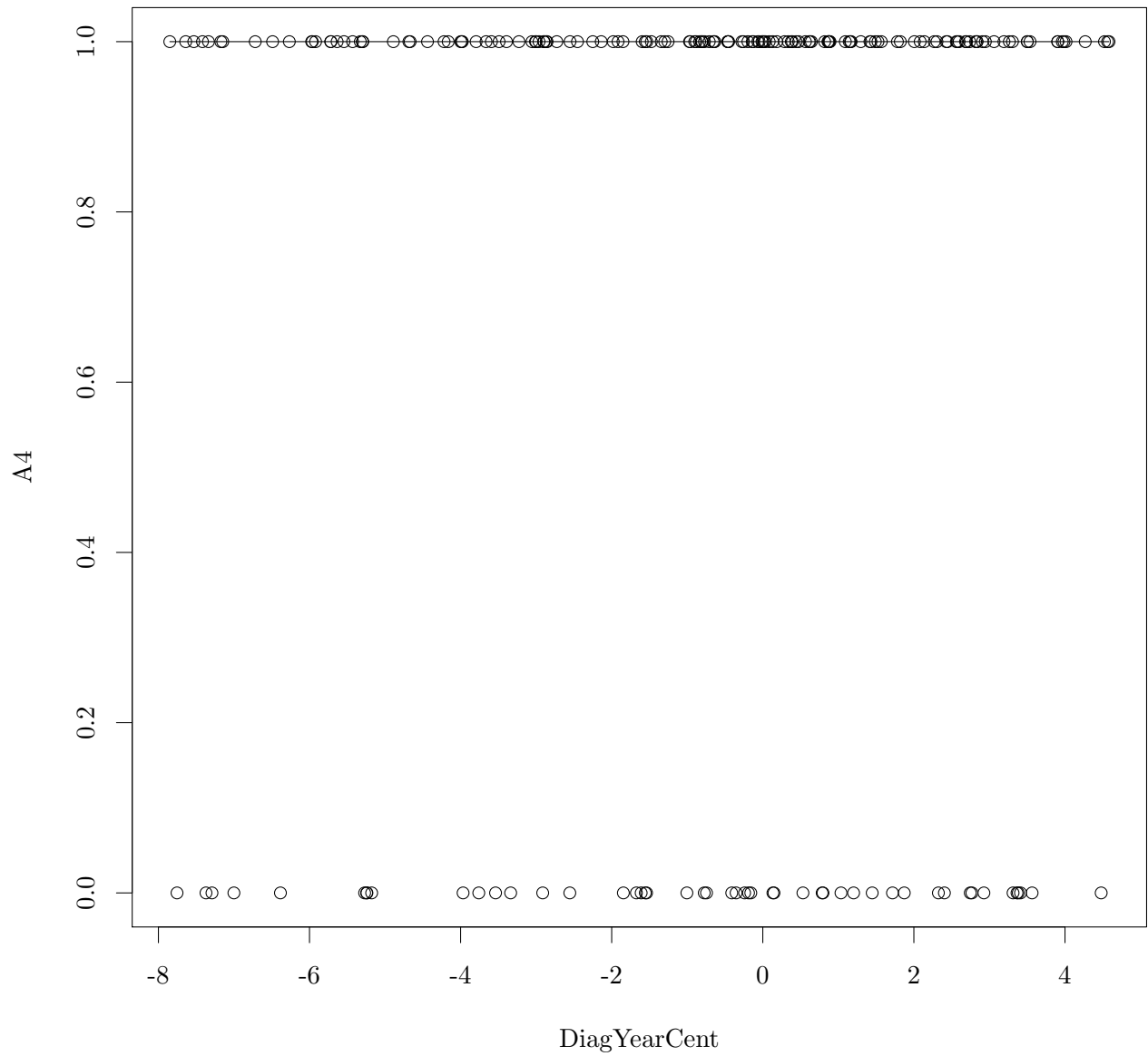
```
kruskal.test(data$DiagYearCent, data$A2)

##
##  Kruskal-Wallis rank sum test
##
## data:  data$DiagYearCent and data$A2
## Kruskal-Wallis chi-squared = 0.5693, df = 1, p-value = 0.4505

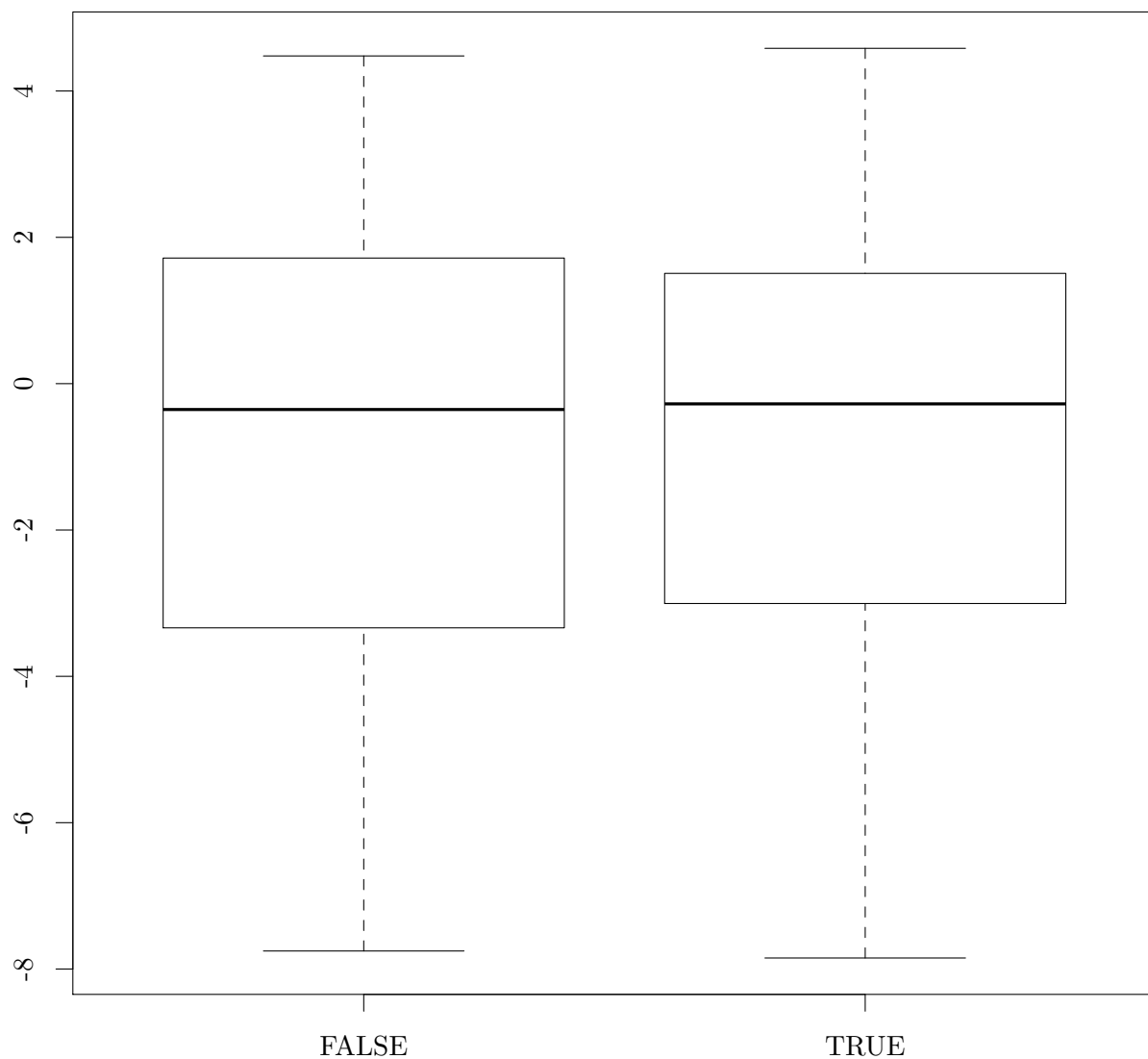
dcov.test(data$DiagYearCent, data$A2, R = 499)

##
##  dCov test of independence
##
## data:  index 1, replicates 499
## nV^2 = 0.6903, p-value = 0.558
## sample estimates:
##      dCov
## 0.05875

scatter.smooth(data$DiagYearCent, data$A4, xlab = "DiagYearCent", ylab = "A4")
```



```
boxplot(DiagYearCent ~ A4, data)
```



```
kruskal.test(data$DiagYearCent, data$A4)

##
##  Kruskal-Wallis rank sum test
##
## data:  data$DiagYearCent and data$A4
## Kruskal-Wallis chi-squared = 0.0055, df = 1, p-value = 0.9411

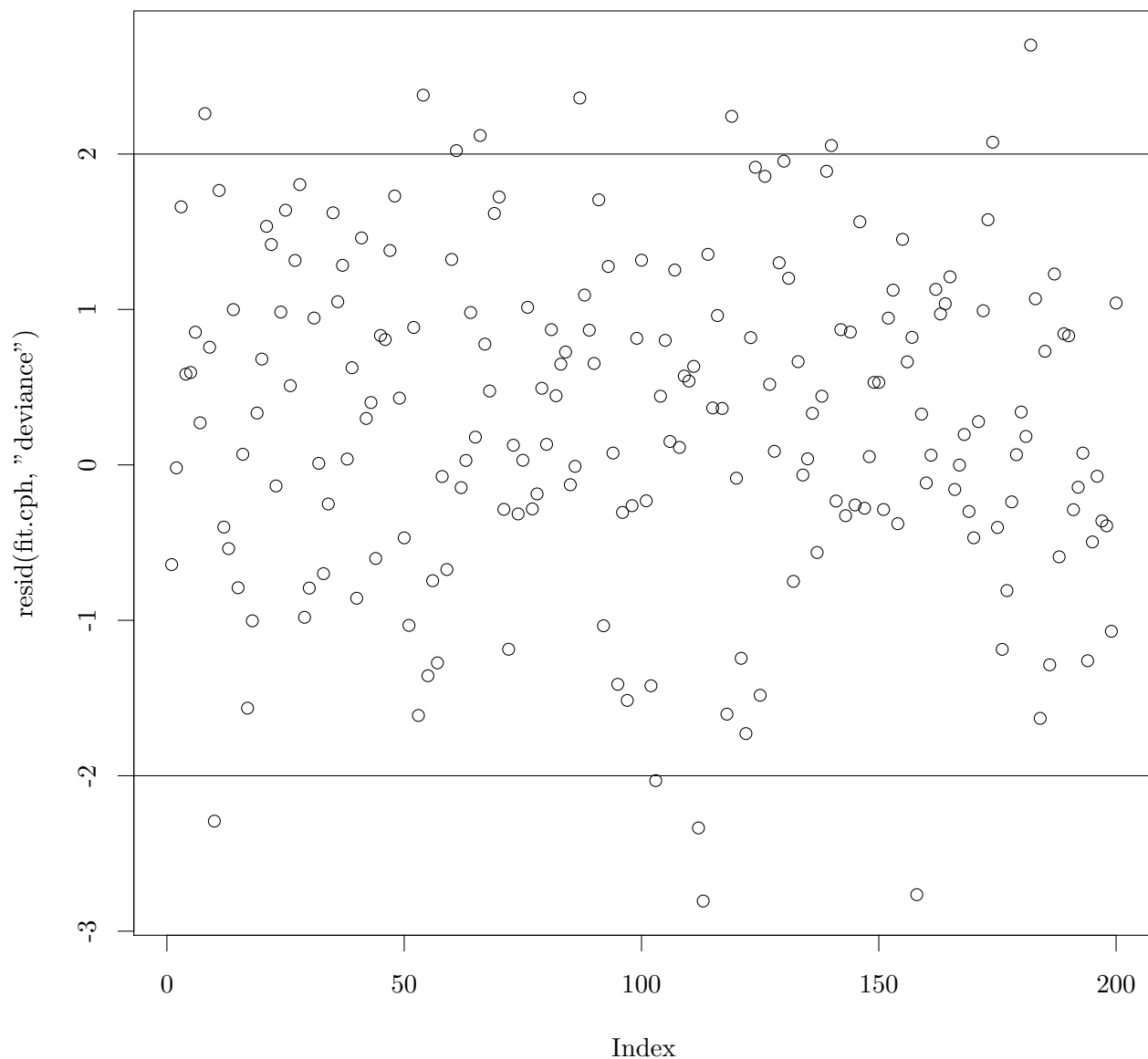
dcov.test(data$DiagYearCent, data$A4, R = 499)

##
##  dCov test of independence
##
## data:  index 1, replicates 499
## nV^2 = 0.1731, p-value = 0.998
## sample estimates:
##      dCov
## 0.02942
```

Not significant; good.

4.4 Outliers

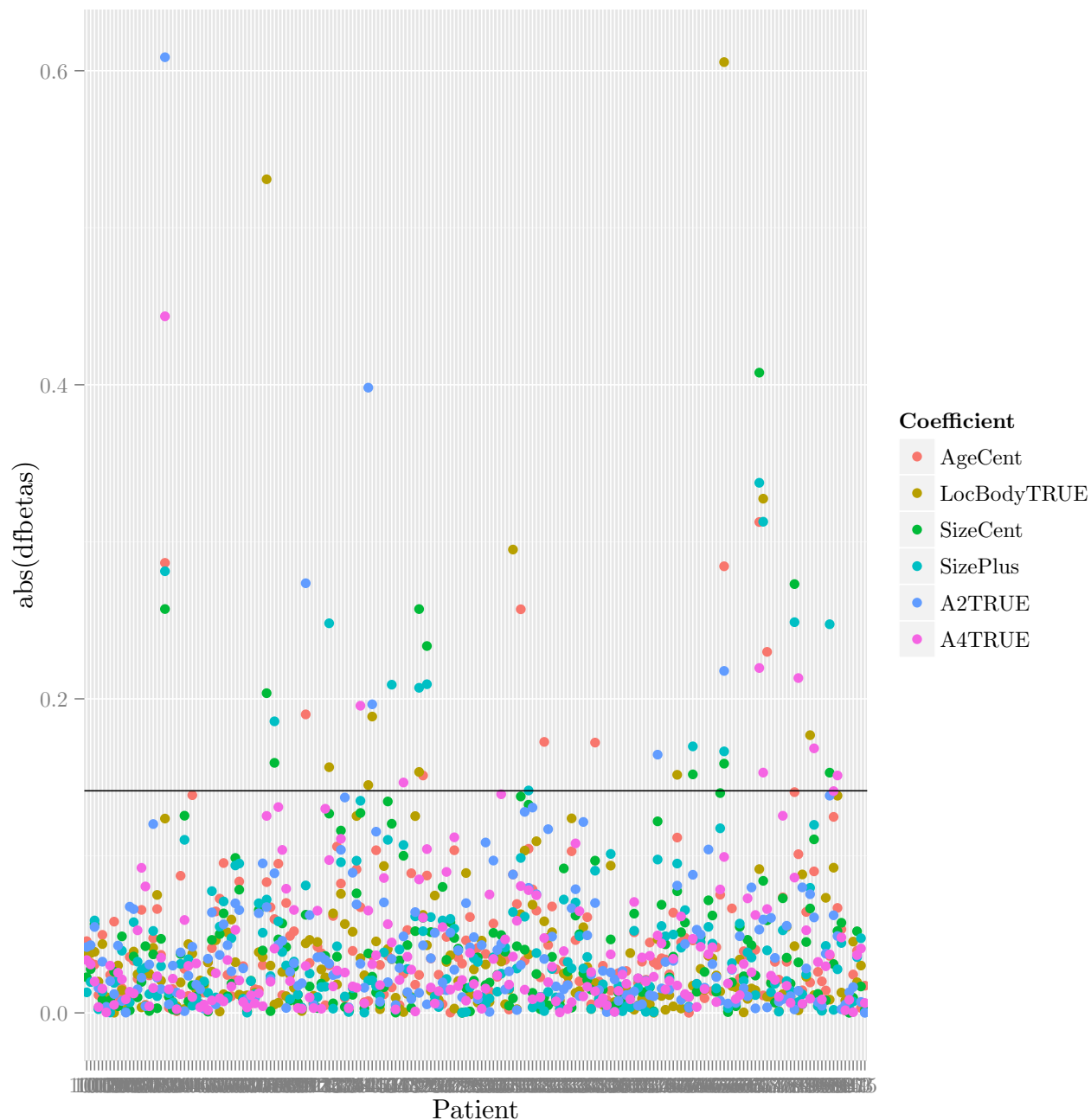
```
plot(resid(fit.cph, "deviance"))  
abline(h = c(-2, 2))
```



```
data$devresid = resid(fit.cph, type = "deviance")  
temp = data[abs(data$devresid) >= 2,]  
#temp[order(temp$Time),]  
  
temp = resid(fit.cph, type = "dfbetas")  
colnames(temp) = names(fit.cph$coefficients)  
temp = melt(temp)  
colnames(temp) = c("Patient", "Coefficient", "dfbetas")  
temp$Patient = gsub("NSWPCN_", "", temp$Patient)  
2/sqrt(nrow(data)) # The classic threshold for concern is 2/sqrt(n).  
## [1] 0.1414
```



```
ggplot(temp, aes(y = abs(dfbetas), x = Patient, col = Coefficient)) + geom_point() + geom_hline(yintercept = 0.15)
```



```
#sort(apply(abs(resid(fit.cph, type = "dfbetas")), 1, max), decreasing = TRUE)
sum(apply(abs(resid(fit.cph, type = "dfbetas")), 1, max) > 2/sqrt(nrow(data)))

## [1] 31

temp = resid(fit.cph, type = "dfbetas")
data$DFBETAS_max = apply(abs(temp), 1, max)
data$DFBETAS_vars = apply(abs(temp), 1, function(x) paste(attr(fit.cph$terms, "term.labels")[x > 2/sqrt(nrow(data))], collapse = ", "))
temp = data[data$DFBETAS_max >= 2/sqrt(nrow(data)) | abs(data$devresid) >= 2,]
#temp[order(temp$DFBETAS_max),]
```

Remove points with deviance residuals ≥ 2.5 , or DFBETAS ≥ 0.3 .

```
nrow(data)

## [1] 200

data = data[data$DFBETAS_max <= 0.3 & abs(data$devresid) <= 2.5,]
nrow(data)

## [1] 193

fit.cph = coxph(Surv(Time, DSD) ~ strata(SexM) + AgeCent + LocBody + SizeCent + SizePlus + A2 + A4, data)
```

4.5 EDA: Variable selection

```
nobs.coxph <- function(obj, ...) sum(obj$y[,2])
fit.cph.as.bic1 = glmulti(Surv(Time, DSD) ~ strata(SexM) + AgeCent + LocBody + SizeCent + SizePlus + A2

## Initialization...
## TASK: Exhaustive screening of candidate set.
## Fitting...
##
## After 50 models:
## Best model: Surv(Time,DSD)~1+A2+A4
## Crit= 1569.99720157408
## Mean crit= 1579.04206453807
##
## After 100 models:
## Best model: Surv(Time,DSD)~1+strata(SexM)+SizeCent+A2
## Crit= 1322.28966392719
## Mean crit= 1493.81514417481
##
## After 150 models:
## Best model: Surv(Time,DSD)~1+strata(SexM)+SizeCent+A2+A4
## Crit= 1319.12027767861
## Mean crit= 1416.9645603344
## Completed.

fit.cph.as.aicc1 = glmulti(Surv(Time, DSD) ~ strata(SexM) + AgeCent + LocBody + SizeCent + SizePlus + A2

## Initialization...
## TASK: Exhaustive screening of candidate set.
## Fitting...
##
## After 50 models:
## Best model: Surv(Time,DSD)~1+LocBody+SizeCent+A4
## Crit= 1562.92910743338
## Mean crit= 1570.63396981566
##
## After 100 models:
## Best model: Surv(Time,DSD)~1+strata(SexM)+LocBody+SizeCent+A2
## Crit= 1315.8613218026
## Mean crit= 1484.90325895394
##
```

```
## After 150 models:
## Best model: Surv(Time,DSD)~1+strata(SexM)+LocBody+SizeCent+A2+A4
## Crit= 1309.03451494962
## Mean crit= 1406.96604818801
## Completed.

rm(nobs.coxph)
summary(fit.cph.as.bic1)$bestmodel

## [1] "Surv(Time, DSD) ~ 1 + strata(SexM) + SizeCent + A2 + A4"

summary(fit.cph.as.aiccl1)$bestmodel

## [1] "Surv(Time, DSD) ~ 1 + strata(SexM) + LocBody + SizeCent + A2 + "
## [2] "      A4"
```

Also run BIC stepwise, because we can.

```
stepAIC(fit.cph, k = log(nrow(data)))

## Start:  AIC=1330
## Surv(Time, DSD) ~ strata(SexM) + AgeCent + LocBody + SizeCent +
##      SizePlus + A2 + A4
##
##           Df  AIC
## - SizePlus  1 1325
## - SizeCent  1 1326
## - AgeCent   1 1327
## - LocBody   1 1328
## <none>      1330
## - A4        1 1333
## - A2        1 1334
##
## Step:  AIC=1325
## Surv(Time, DSD) ~ strata(SexM) + AgeCent + LocBody + SizeCent +
##      A2 + A4
##
##           Df  AIC
## - AgeCent   1 1322
## - LocBody   1 1322
## - SizeCent  1 1324
## <none>      1325
## - A2        1 1329
## - A4        1 1330
##
## Step:  AIC=1322
## Surv(Time, DSD) ~ strata(SexM) + LocBody + SizeCent + A2 + A4
##
##           Df  AIC
## - LocBody   1 1319
## - SizeCent  1 1321
## <none>      1322
## - A2        1 1325
## - A4        1 1326
##
```

```

## Step: AIC=1319
## Surv(Time, DSD) ~ strata(SexM) + SizeCent + A2 + A4
##
##           Df  AIC
## <none>      1319
## - SizeCent  1 1322
## - A4        1 1322
## - A2        1 1324
## Call:
## coxph(formula = Surv(Time, DSD) ~ strata(SexM) + SizeCent + A2 +
##       A4, data = data)
##
##           coef exp(coef) se(coef)      z      p
## SizeCent 0.0159      1.02  0.00543 2.92 0.0035
## A2TRUE   0.7003      2.01  0.20650 3.39 0.0007
## A4TRUE   0.5154      1.67  0.18497 2.79 0.0053
##
## Likelihood ratio test=34.1 on 3 df, p=1.92e-07 n= 193, number of events= 184

stepAIC(fit.cph, k = 2)

## Start: AIC=1311
## Surv(Time, DSD) ~ strata(SexM) + AgeCent + LocBody + SizeCent +
##       SizePlus + A2 + A4
##
##           Df  AIC
## - SizePlus  1 1309
## - SizeCent  1 1310
## - AgeCent   1 1311
## <none>      1311
## - LocBody   1 1311
## - A4        1 1317
## - A2        1 1318
##
## Step: AIC=1309
## Surv(Time, DSD) ~ strata(SexM) + AgeCent + LocBody + SizeCent +
##       A2 + A4
##
##           Df  AIC
## - AgeCent   1 1309
## <none>      1309
## - LocBody   1 1309
## - SizeCent  1 1311
## - A2        1 1316
## - A4        1 1317
##
## Step: AIC=1309
## Surv(Time, DSD) ~ strata(SexM) + LocBody + SizeCent + A2 + A4
##
##           Df  AIC
## <none>      1309
## - LocBody   1 1309
## - SizeCent  1 1311

```

```
## - A2          1 1315
## - A4          1 1316
## Call:
## coxph(formula = Surv(Time, DSD) ~ strata(SexM) + LocBody + SizeCent +
##       A2 + A4, data = data)
##
##
##              coef exp(coef) se(coef)      z      p
## LocBodyTRUE 0.3806      1.46   0.2267 1.68 0.0930
## SizeCent    0.0126      1.01   0.0058 2.18 0.0290
## A2TRUE      0.6301      1.88   0.2120 2.97 0.0030
## A4TRUE      0.5312      1.70   0.1850 2.87 0.0041
##
## Likelihood ratio test=36.7  on 4 df, p=2.04e-07  n= 193, number of events= 184
```

4.6 Final Fits

```
fit.cph.as.bic = coxph(Surv(Time, DSD) ~ strata(SexM) + SizePlus + A2 + A4, data = data)
cox.zph(fit.cph.as.bic)

##              rho  chisq      p
## SizePlus    0.0212 0.0876 0.767
## A2TRUE      0.0340 0.2136 0.644
## A4TRUE     -0.0808 1.1972 0.274
## GLOBAL              NA 1.3865 0.709

fit.cph.as.aicc = coxph(Surv(Time, DSD) ~ strata(SexM)+AgeCent+LocBody+SizeCent+A2+A4+SizeCent:AgeCent+
cox.zph(fit.cph.as.aicc)

##              rho  chisq      p
## AgeCent              -0.16098 5.43356 0.0198
## LocBodyTRUE          0.03967 0.30863 0.5785
## SizeCent              0.00379 0.00275 0.9581
## A2TRUE                0.04060 0.34304 0.5581
## A4TRUE                -0.06803 0.84941 0.3567
## AgeCent:SizeCent      0.03856 0.28388 0.5942
## strata(SexM)SexM=TRUE:SizeCent 0.00853 0.01322 0.9085
## GLOBAL              NA 7.49932 0.3788

fit.cph.sw.bic = coxph(Surv(Time, DSD) ~ strata(SexM) + SizeCent + A2 + A4, data = data)
cox.zph(fit.cph.sw.bic)

##              rho  chisq      p
## SizeCent    0.0162 0.0507 0.822
## A2TRUE      0.0312 0.1797 0.672
## A4TRUE     -0.0874 1.4015 0.236
## GLOBAL              NA 1.4878 0.685

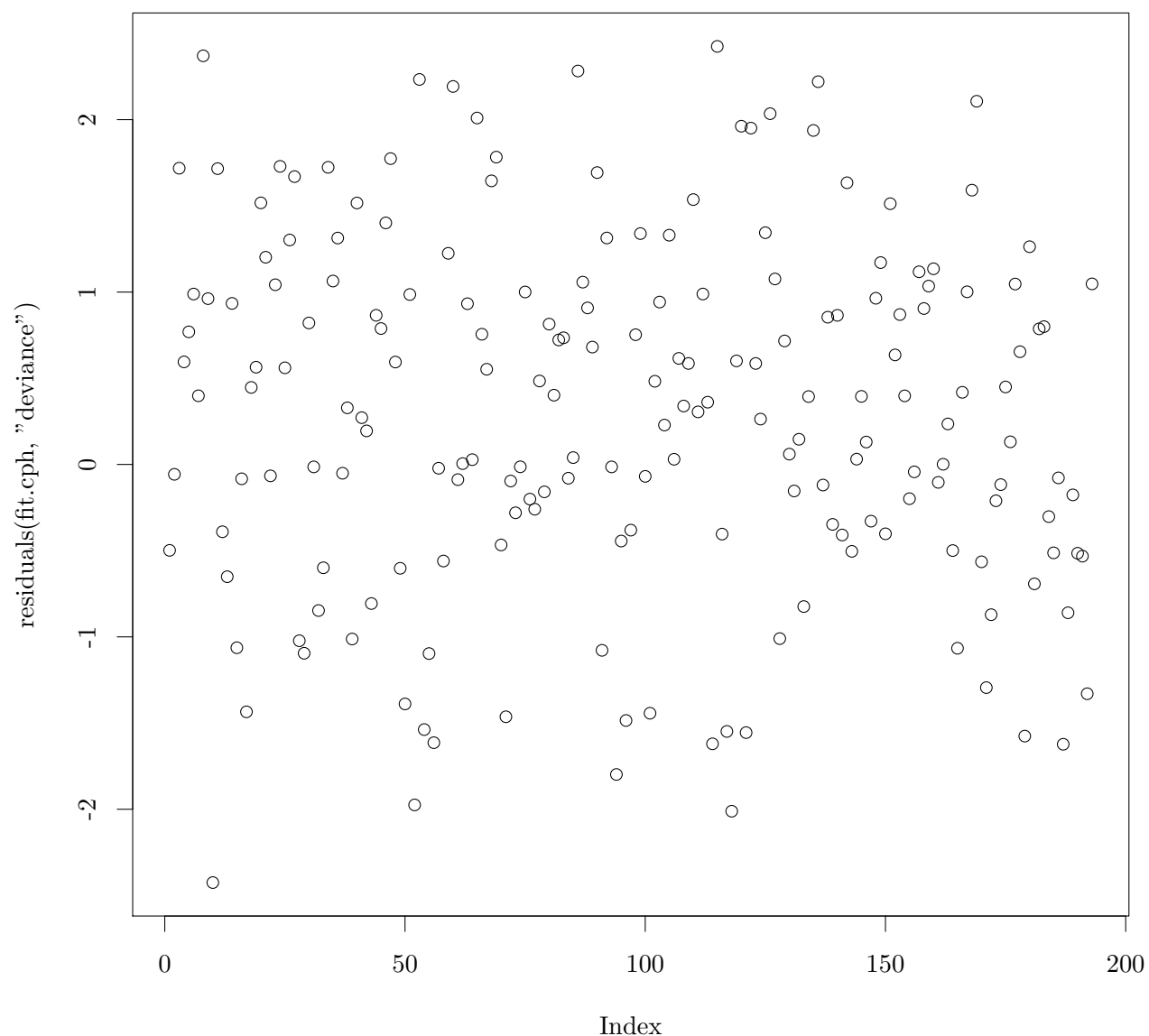
fit.cph.sw.aic = coxph(Surv(Time, DSD) ~ strata(SexM) + LocBody + SizeCent + A2 + A4, data = data)
cox.zph(fit.cph.sw.aic)

##              rho  chisq      p
## LocBodyTRUE 0.0180 0.0592 0.808
```

```
## SizeCent      0.0280 0.1465 0.702
## A2TRUE        0.0292 0.1636 0.686
## A4TRUE       -0.0839 1.2904 0.256
## GLOBAL         NA 1.6815 0.794
```

```
fit.cph = fit.cph.sw.aic
```

```
plot(residuals(fit.cph, "deviance"))
```



```
residuals(fit.cph, "deviance")[abs(residuals(fit.cph, "deviance")) >= 2]
```

```
## NSWPCN_125 NSWPCN_133 NSWPCN_315 NSWPCN_324 NSWPCN_333 NSWPCN_374
##      2.370      -2.425       2.233       2.193       2.009       2.282
## NSWPCN_779 NSWPCN_788 NSWPCN_799 NSWPCN_1017 NSWPCN_1165
##      2.425      -2.011       2.035       2.220       2.107
```

```
temp = sort(apply(abs(residuals(fit.cph, "dfbetas")), 1, max))
```

```

#temp
2/sqrt(nrow(data))

## [1] 0.144

mean(temp > 2/sqrt(nrow(data)))

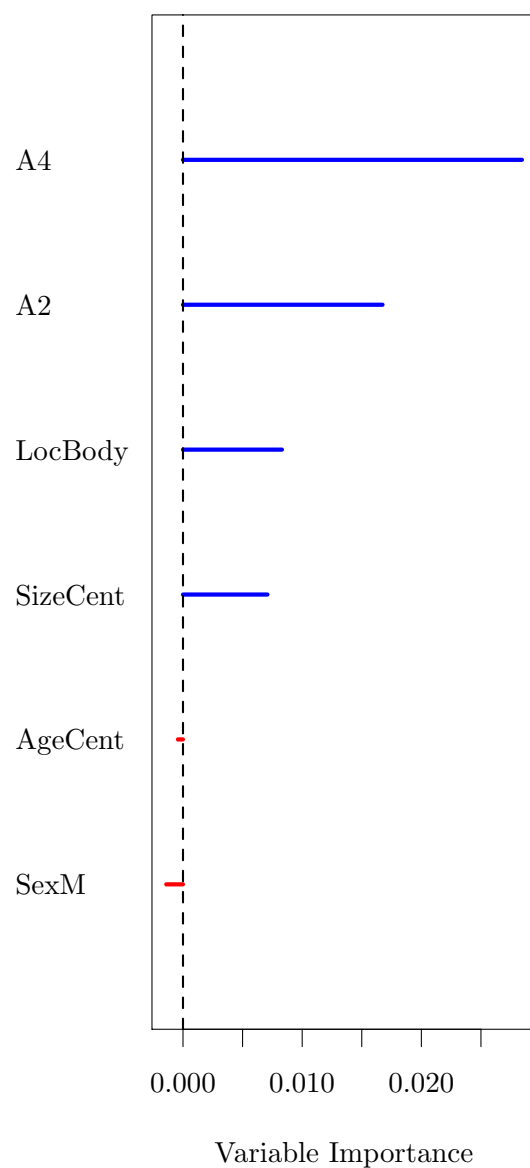
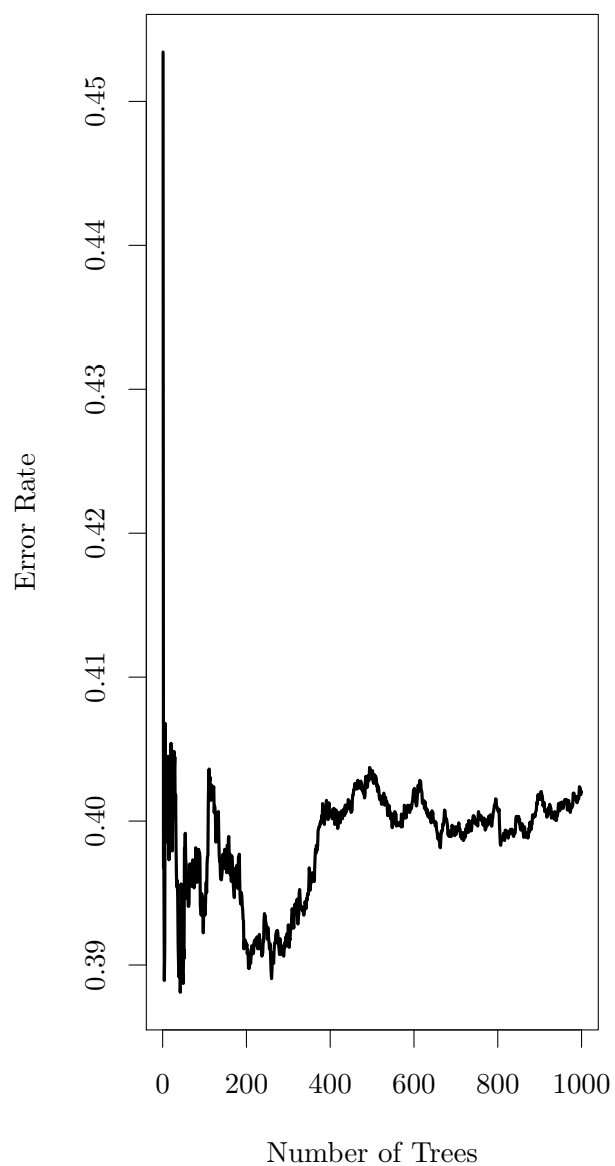
## [1] 0.1244

temp[temp > 2/sqrt(nrow(data))]

## NSWPCN_354 NSWPCN_445 NSWPCN_133 NSWPCN_374 NSWPCN_784 NSWPCN_777
## 0.1457 0.1524 0.1566 0.1580 0.1618 0.1637
## NSWPCN_195 NSWPCN_296 NSWPCN_267 NSWPCN_1155 NSWPCN_154 NSWPCN_794
## 0.1652 0.1674 0.1711 0.1804 0.1895 0.2037
## NSWPCN_802 NSWPCN_142 NSWPCN_799 NSWPCN_313 NSWPCN_192 NSWPCN_317
## 0.2056 0.2174 0.2178 0.2219 0.2225 0.2541
## NSWPCN_318 NSWPCN_788 NSWPCN_145 NSWPCN_1253 NSWPCN_1212 NSWPCN_310
## 0.2567 0.2749 0.3006 0.4234 0.4528 0.4926

set.seed(20150208)
fit.rsfc = rfsrc(Surv(Time, DSD) ~ SexM + AgeCent + LocBody + SizeCent + A2 + A4, data = data, mtry = 1,
plot(fit.rsfc)

```



```
##
##          Importance  Relative Imp
## A4          0.0284         1.0000
## A2          0.0167         0.5887
## LocBody     0.0083         0.2920
## SizeCent    0.0071         0.2492
## AgeCent    -0.0004        -0.0149
## SexM       -0.0014        -0.0494
```

```
fit.gg = flexsurvreg(Surv(Time, DSD) ~ SexM + LocBody + SizeCent + A2 + A4,
  anc = list(
    sigma = ~ SexM,
    Q = ~ SexM),
  data = data, dist = "gengamma")
```

```
fit.gg2 = flexsurvreg(Surv(Time, DSD) ~ SexM+AgeCent+LocBody+SizeCent+A2+A4+SizeCent:AgeCent+SexM:SizeCent)
```



```

    anc = list(
      sigma = ~ SexM,
      Q = ~ SexM),
    data = data, dist = "gengamma")

fit.gg$loglik
## [1] -1325
fit.gg2$loglik
## [1] -1321
pchisq(2*(fit.gg2$loglik - fit.gg$loglik), 3, lower.tail = FALSE)
## [1] 0.04837
AIC(fit.gg)
## [1] 2669
AIC(fit.gg2)
## [1] 2668
fit.gg
##
## Call:
## flexsurvreg(formula = Surv(Time, DSD) ~ SexM + LocBody + SizeCent +      A2 + A4, anc = list(sigma =
##
## Estimates:
##


|                    | data     | mean | est      | L95%     | U95%     | se      |
|--------------------|----------|------|----------|----------|----------|---------|
| ## mu              | NA       |      | 6.53611  | 6.19247  | 6.87976  | 0.17533 |
| ## sigma           | NA       |      | 0.78047  | 0.67245  | 0.90585  | 0.05932 |
| ## Q               | NA       |      | 0.11827  | -0.49632 | 0.73287  | 0.31357 |
| ## SexMTRUE        | 0.51813  |      | 0.28181  | -0.07256 | 0.63619  | 0.18081 |
| ## LocBodyTRUE     | 0.17098  |      | -0.20952 | -0.50577 | 0.08673  | 0.15115 |
| ## SizeCent        | 3.65285  |      | -0.00879 | -0.01600 | -0.00158 | 0.00368 |
| ## A2TRUE          | 0.16580  |      | -0.38962 | -0.65941 | -0.11983 | 0.13765 |
| ## A4TRUE          | 0.75130  |      | -0.39725 | -0.62687 | -0.16763 | 0.11716 |
| ## sigma(SexMTRUE) | 0.51813  |      | -0.26267 | -0.49374 | -0.03159 | 0.11790 |
| ## Q(SexMTRUE)     | 0.51813  |      | 0.48452  | -0.32987 | 1.29891  | 0.41551 |
| ##                 | exp(est) |      | L95%     |          | U95%     |         |
| ## mu              | NA       |      | NA       |          | NA       |         |
| ## sigma           | NA       |      | NA       |          | NA       |         |
| ## Q               | NA       |      | NA       |          | NA       |         |
| ## SexMTRUE        | 1.32553  |      | 0.93001  |          | 1.88927  |         |
| ## LocBodyTRUE     | 0.81097  |      | 0.60304  |          | 1.09060  |         |
| ## SizeCent        | 0.99124  |      | 0.98412  |          | 0.99842  |         |
| ## A2TRUE          | 0.67731  |      | 0.51715  |          | 0.88707  |         |
| ## A4TRUE          | 0.67217  |      | 0.53426  |          | 0.84567  |         |
| ## sigma(SexMTRUE) | 0.76900  |      | 0.61034  |          | 0.96890  |         |
| ## Q(SexMTRUE)     | 1.62340  |      | 0.71902  |          | 3.66531  |         |


##
## N = 193, Events: 184, Censored: 9
## Total time at risk: 114833
## Log-likelihood = -1325, df = 10
## AIC = 2669

```

```

fit.gg2

##
## Call:
## flexsurvreg(formula = Surv(Time, DSD) ~ SexM + AgeCent + LocBody +      SizeCent + A2 + A4 + SizeCent,
##
## Estimates:
##
##      data mean  est      L95%      U95%      se
## mu              NA  6.530218   6.184887   6.875549   0.176192
## sigma            NA  0.771216   0.660311   0.900749   0.061092
## Q                NA  0.228786  -0.410815   0.868387   0.326333
## SexMTRUE          0.518135  0.322116  -0.039753   0.683986   0.184631
## AgeCent          -1.067358  0.010352   0.000170   0.020534   0.005195
## LocBodyTRUE       0.170984 -0.271326  -0.558764   0.016113   0.146655
## SizeCent          3.652850 -0.004245  -0.015597   0.007107   0.005792
## A2TRUE            0.165803 -0.358631  -0.618603  -0.098660   0.132641
## A4TRUE            0.751295 -0.354054  -0.574822  -0.133287   0.112639
## AgeCent:SizeCent -8.896373 -0.000855  -0.001550  -0.000160   0.000354
## SexMTRUE:SizeCent  1.772021 -0.006910  -0.020503   0.006684   0.006936
## sigma(SexMTRUE)   0.518135 -0.334045  -0.602093  -0.065998   0.136762
## Q(SexMTRUE)       0.518135  0.550014  -0.328860   1.428889   0.448414
##
##      exp(est)  L95%      U95%
## mu              NA          NA          NA
## sigma            NA          NA          NA
## Q                NA          NA          NA
## SexMTRUE          1.380045  0.961027  1.981761
## AgeCent           1.010406  1.000170  1.020746
## LocBodyTRUE       0.762368  0.571915  1.016243
## SizeCent           0.995764  0.984524  1.007133
## A2TRUE            0.698632  0.538697  0.906051
## A4TRUE            0.701837  0.562805  0.875214
## AgeCent:SizeCent  0.999145  0.998452  0.999840
## SexMTRUE:SizeCent  0.993114  0.979706  1.006706
## sigma(SexMTRUE)   0.716021  0.547664  0.936133
## Q(SexMTRUE)       1.733278  0.719744  4.174059
##
## N = 193,  Events: 184,  Censored: 9
## Total time at risk: 114833
## Log-likelihood = -1321, df = 13
## AIC = 2668

```

5 Fit assessment

Plot fit stratified by sex, separate curves for A2, A4 status, at median (approx.) Size.

```

temp.grid = expand.grid(A4 = c(FALSE, TRUE), A2 = c(FALSE, TRUE), SexM = c(FALSE, TRUE), SizeCent = 0, A
temp.grid$ID = sprintf("SexM=%s, A2=% -5s, A4=% -5s, LocBody=%s", temp.grid$SexM, temp.grid$A2, temp.gr
temp.preds = summary(fit.gg, newdata = temp.grid, type = "survival", t = seq(0, 365*5, 30))
temp.preds2 = do.call(rbind, temp.preds)
temp.preds2$group = rep(gsub(".*ID=", "", names(temp.preds)), each = nrow(temp.preds[[1]]))
temp.preds.cox = survfit(fit.cph, newdata = temp.grid)
temp.preds.rsfc = predict(fit.rsfc, newdata = temp.grid)

```

```

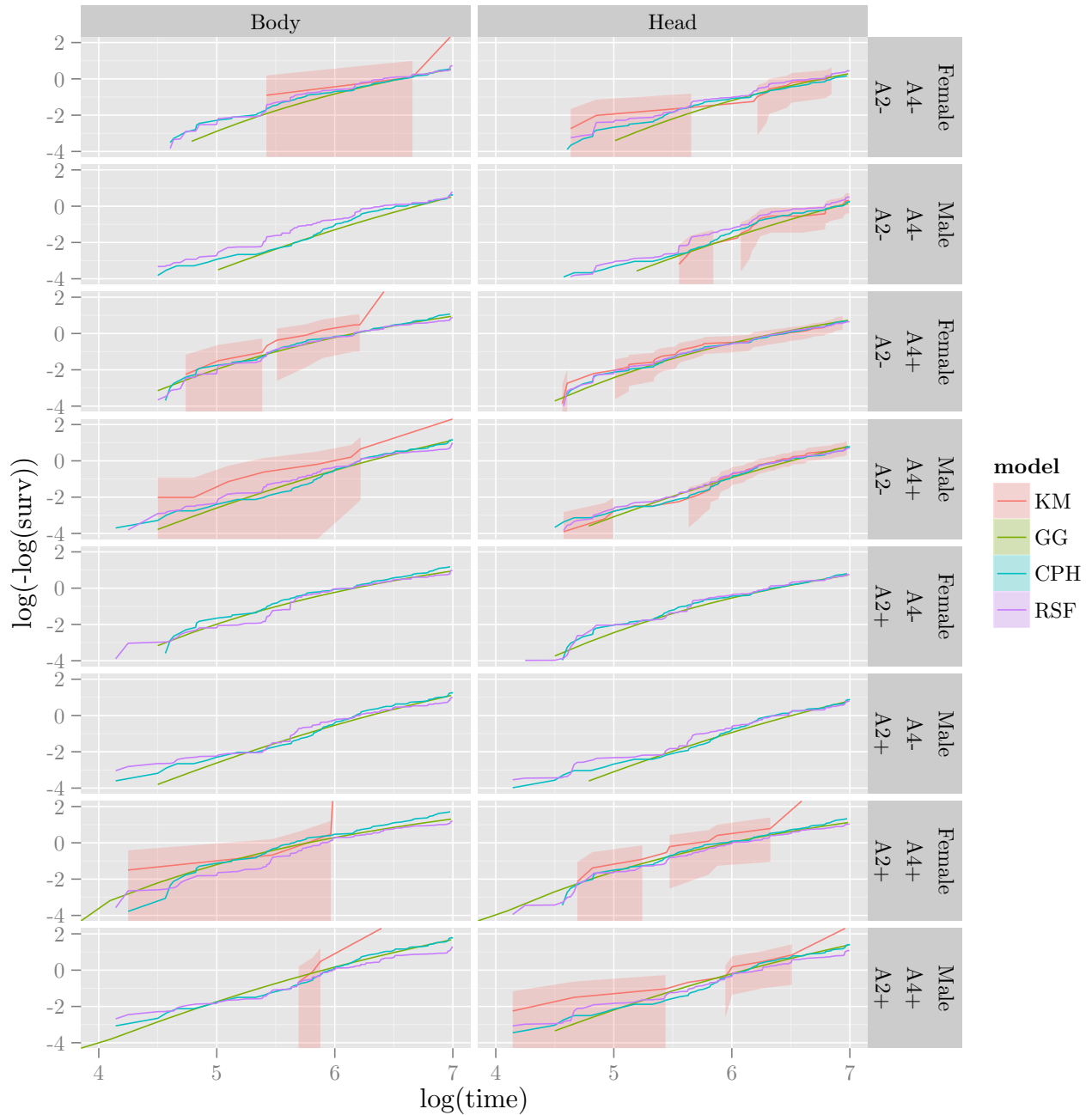
temp.survfit = survfit(Surv(Time, DSD) ~ SexM + A2 + A4 + LocBody, data)
temp.data = data.frame(time = temp.survfit$time, surv = temp.survfit$surv, upper = temp.survfit$lower, lower = temp.survfit$lower)
temp.data = rbind(temp.data, data.frame(time = temp.preds2$time, surv = temp.preds2$est, upper = temp.preds2$upper, lower = temp.preds2$lower))
temp.data = rbind(temp.data, data.frame(time = temp.preds.cox$time, surv = temp.preds.cox$surv, upper = temp.preds.cox$upper, lower = temp.preds.cox$lower))
temp.data = rbind(temp.data, data.frame(time = rep(temp.preds.rsrf$time.interest, each = nrow(temp.preds.rsrf)), surv = temp.preds.rsrf$est, upper = temp.preds.rsrf$upper, lower = temp.preds.rsrf$lower))

temp.data$Sex = c("Male", "Female")[grepl("SexM=FALSE", temp.data$group)+1]
temp.data$A2 = c("A2-", "A2+")[grepl("A2=TRUE", temp.data$group)+1]
temp.data$A4 = c("A4-", "A4+")[grepl("A4=TRUE", temp.data$group)+1]
temp.data$Location = c("Head", "Body")[grepl("LocBody=TRUE", temp.data$group)+1]

temp.data$lower[temp.data$model != "KM"] = NA
temp.data$upper[temp.data$model != "KM"] = NA
ggplot(temp.data, aes(x = log(time), y = log(-log(surv)), ymin = log(-log(lower)), ymax = log(-log(upper)))) +
  geom_ribbon(alpha = 0.25, colour = NA) +
  geom_line() +
  xlim(4, 7) + ylim(-4, 2) +
  facet_grid(A2 ~ A4 ~ Sex ~ Location)

## Warning: Removed 64 rows containing missing values (geom_path).
## Warning: Removed 70 rows containing missing values (geom_path).
## Warning: Removed 59 rows containing missing values (geom_path).
## Warning: Removed 69 rows containing missing values (geom_path).
## Warning: Removed 60 rows containing missing values (geom_path).
## Warning: Removed 70 rows containing missing values (geom_path).
## Warning: Removed 57 rows containing missing values (geom_path).
## Warning: Removed 66 rows containing missing values (geom_path).
## Warning: Removed 58 rows containing missing values (geom_path).
## Warning: Removed 59 rows containing missing values (geom_path).
## Warning: Removed 56 rows containing missing values (geom_path).
## Warning: Removed 56 rows containing missing values (geom_path).
## Warning: Removed 57 rows containing missing values (geom_path).
## Warning: Removed 58 rows containing missing values (geom_path).
## Warning: Removed 57 rows containing missing values (geom_path).
## Warning: Removed 56 rows containing missing values (geom_path).

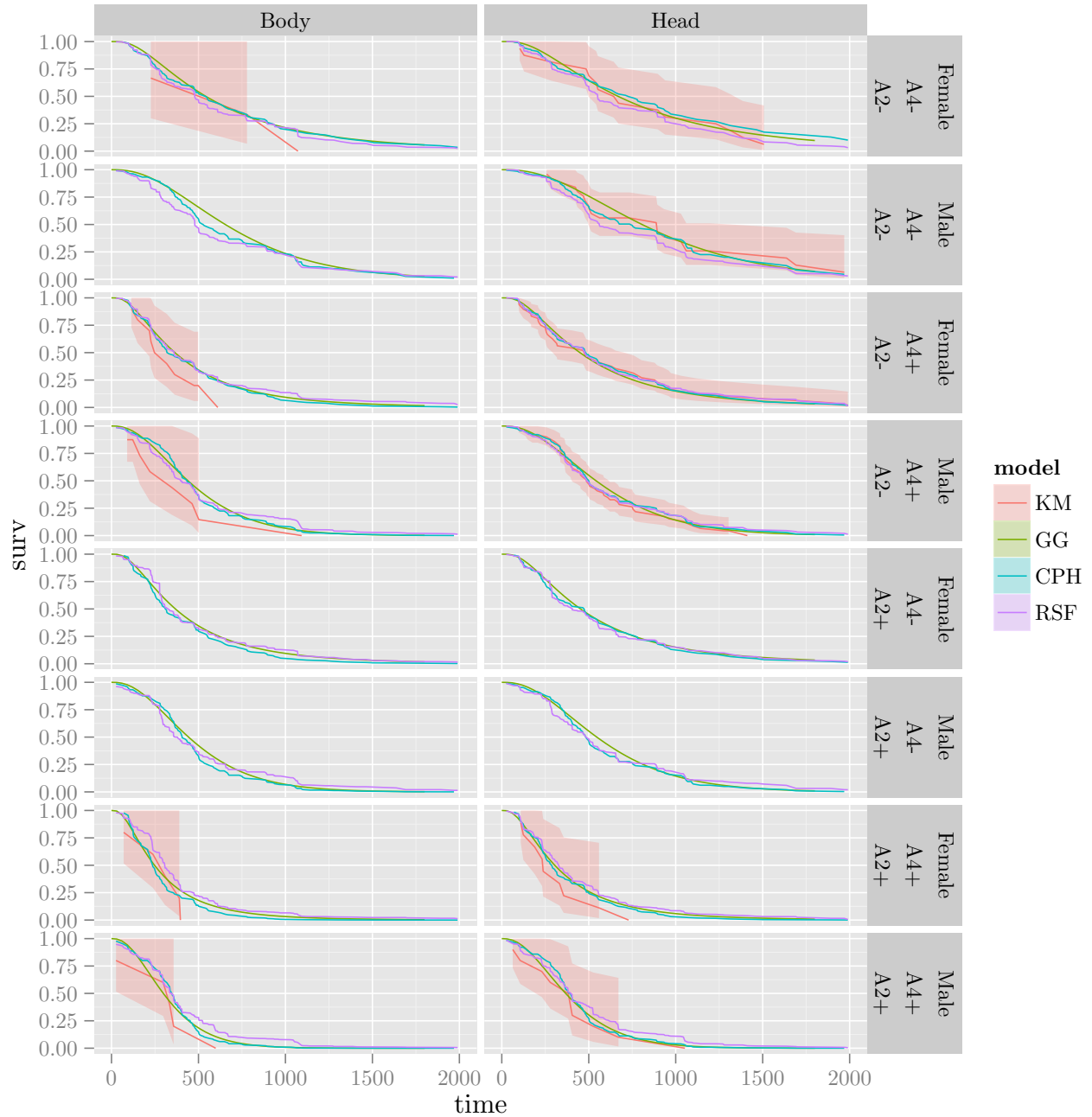
```



```
ggplot(temp.data, aes(x = time, y = surv, ymin = lower, ymax = upper, colour = model, fill = model)) +
  geom_ribbon(alpha = 0.25, colour = NA) +
  geom_line() + xlim(0, 2000) + ylim(0, 1) +
  facet_grid(A2 ~ A4 ~ Sex ~ Location)
```

```
## Warning: Removed 4 rows containing missing values (geom_path).
## Warning: Removed 5 rows containing missing values (geom_path).
## Warning: Removed 3 rows containing missing values (geom_path).
## Warning: Removed 4 rows containing missing values (geom_path).
## Warning: Removed 4 rows containing missing values (geom_path).
## Warning: Removed 5 rows containing missing values (geom_path).
## Warning: Removed 3 rows containing missing values (geom_path).
## Warning: Removed 3 rows containing missing values (geom_path).
```

```
## Warning: Removed 4 rows containing missing values (geom_path).
## Warning: Removed 4 rows containing missing values (geom_path).
## Warning: Removed 3 rows containing missing values (geom_path).
## Warning: Removed 3 rows containing missing values (geom_path).
## Warning: Removed 4 rows containing missing values (geom_path).
## Warning: Removed 4 rows containing missing values (geom_path).
## Warning: Removed 3 rows containing missing values (geom_path).
## Warning: Removed 3 rows containing missing values (geom_path).
```



```
temp.grid = expand.grid(A4 = c(FALSE, TRUE), A2 = c(FALSE, TRUE), SexM = c(FALSE, TRUE), SizeCent = 0, A
temp.grid$ID = sprintf("SexM=%s, A2=% -5s, A4=% -5s, LocBody=%s", temp.grid$SexM, temp.grid$A2, temp.gri
temp.preds = summary(fit.gg, newdata = temp.grid, type = "survival", t = seq(0, 365*5, 30))
temp.preds2 = do.call(rbind, temp.preds)
```

```

temp.preds2$group = rep(gsub(".*ID=", "", names(temp.preds)), each = nrow(temp.preds[[1]]))
temp.preds.cox = survfit(fit.cph, newdata = temp.grid)
temp.preds.rsfc = predict(fit.rsfc, newdata = temp.grid)

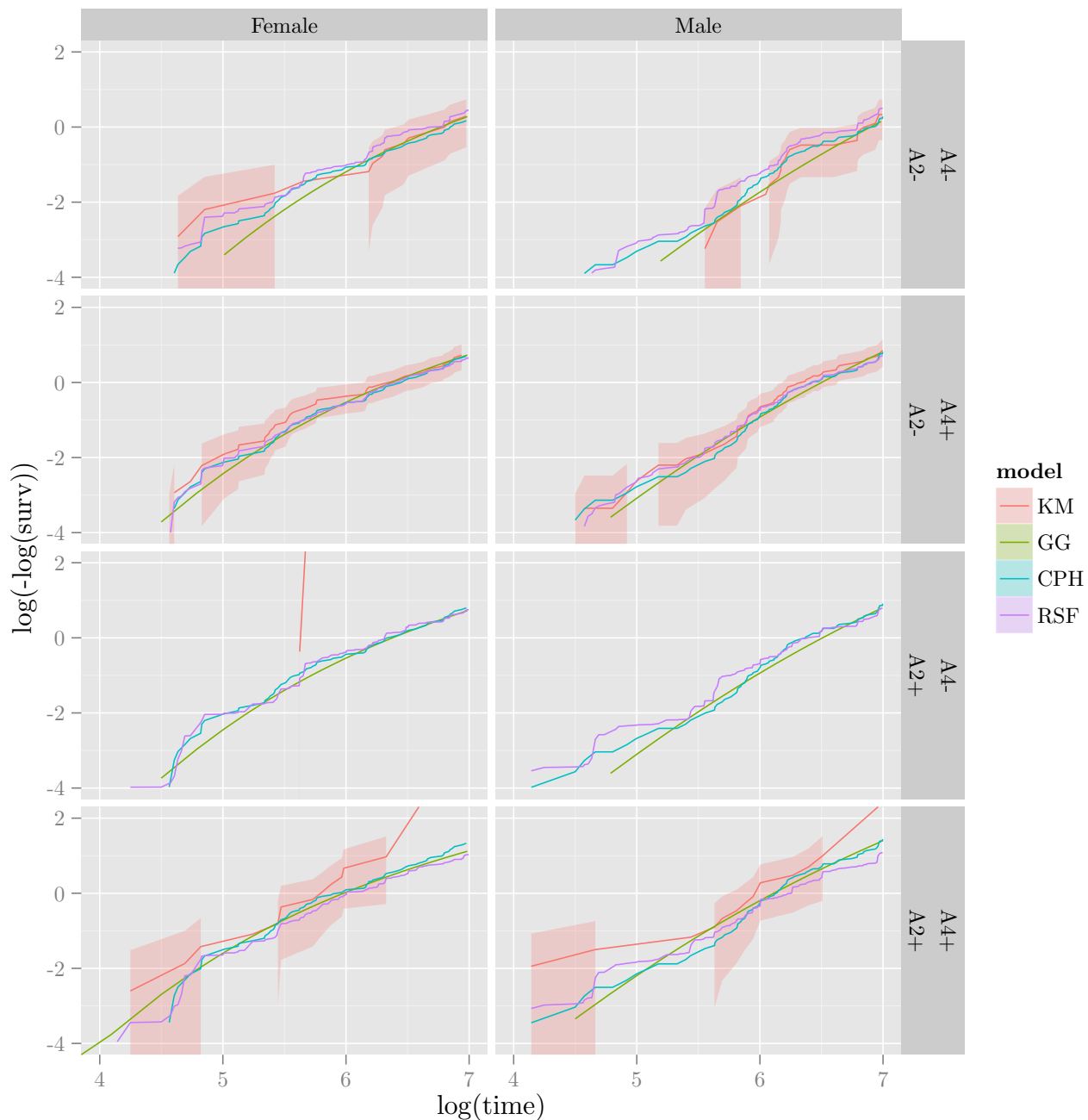
temp.survfit = survfit(Surv(Time, DSD) ~ SexM + A2 + A4, data)
temp.data = data.frame(time = temp.survfit$time, surv = temp.survfit$surv, upper = temp.survfit$lower, lower = temp.survfit$lower)
temp.data = rbind(temp.data, data.frame(time = temp.preds2$time, surv = temp.preds2$est, upper = temp.preds2$upper, lower = temp.preds2$lower))
temp.data = rbind(temp.data, data.frame(time = temp.preds.cox$time, surv = temp.preds.cox$surv, upper = temp.preds.cox$upper, lower = temp.preds.cox$lower))
temp.data = rbind(temp.data, data.frame(time = rep(temp.preds.rsfc$time.interest, each = nrow(temp.preds.rsfc)), surv = temp.preds.rsfc, upper = temp.preds.rsfc$upper, lower = temp.preds.rsfc$lower))

temp.data$Sex = c("Male", "Female")[grepl("SexM=FALSE", temp.data$group)+1]
temp.data$A2 = c("A2-", "A2+")[grepl("A2=TRUE", temp.data$group)+1]
temp.data$A4 = c("A4-", "A4+")[grepl("A4=TRUE", temp.data$group)+1]

temp.data$lower[temp.data$model != "KM"] = NA
temp.data$upper[temp.data$model != "KM"] = NA
ggplot(temp.data, aes(x = log(time), y = log(-log(surv)), ymin = log(-log(lower)), ymax = log(-log(upper)))) +
  geom_ribbon(alpha = 0.25, colour = NA) +
  geom_line() +
  xlim(4, 7) + ylim(-4, 2) +
  facet_grid(A2 ~ A4 ~ Sex)

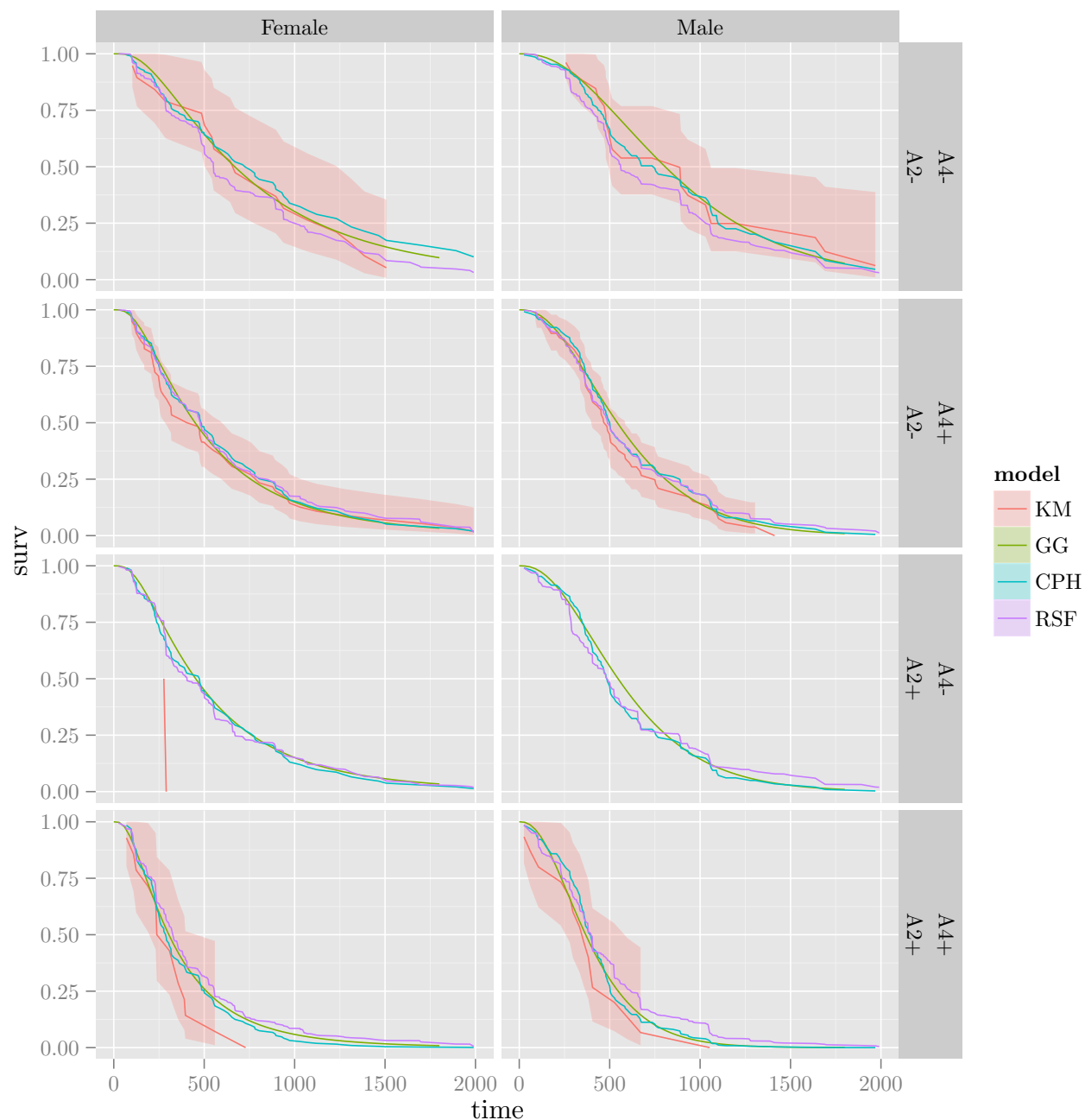
## Warning: Removed 70 rows containing missing values (geom_path).
## Warning: Removed 69 rows containing missing values (geom_path).
## Warning: Removed 71 rows containing missing values (geom_path).
## Warning: Removed 67 rows containing missing values (geom_path).
## Warning: Removed 59 rows containing missing values (geom_path).
## Warning: Removed 56 rows containing missing values (geom_path).
## Warning: Removed 58 rows containing missing values (geom_path).
## Warning: Removed 57 rows containing missing values (geom_path).

```



```
ggplot(temp.data, aes(x = time, y = surv, ymin = lower, ymax = upper, colour = model, fill = model)) +
  geom_ribbon(alpha = 0.25, colour = NA) +
  geom_line() + xlim(0, 2000) + ylim(0, 1) +
  facet_grid(A2 ~ A4 ~ Sex)
```

```
## Warning: Removed 5 rows containing missing values (geom_path).
## Warning: Removed 4 rows containing missing values (geom_path).
## Warning: Removed 5 rows containing missing values (geom_path).
## Warning: Removed 3 rows containing missing values (geom_path).
## Warning: Removed 4 rows containing missing values (geom_path).
## Warning: Removed 3 rows containing missing values (geom_path).
## Warning: Removed 4 rows containing missing values (geom_path).
## Warning: Removed 3 rows containing missing values (geom_path).
```



6 Model selection

It looks like that's as far as we can go with tweaking the fits. Time to put the different models against each other on the holdout data, and choose a winner.

DIY IBS, woo.

```
calcIBS = function(surv, pred, pred_times, max_time, min_time = 0)
{
  stopifnot(nrow(surv) == nrow(pred) && length(pred_times) == ncol(pred))

  n = nrow(surv)
  marg_survfit = survfit(surv ~ 1)
```



```

marg_censfit = survfit(Surv(surv[,1], !surv[,2]) ~ 1)
marg_surv_func = approxfun(marg_survfit$time, marg_survfit$surv, method = "constant", yleft = 1, yright = 0)
marg_cens_func = approxfun(marg_censfit$time, marg_censfit$surv, method = "constant", yleft = 1, yright = 0)

pred_funcs = apply(pred, 1, function(pat_preds) approxfun(pred_times, pat_preds, yleft = 1, yright = 0))

indiv_patient_bsc = function(pat_i, tstars)
{
  observed_time = surv[pat_i, 1]
  observed_event = surv[pat_i, 2]
  pred_func = pred_funcs[[pat_i]]
  category = 1*(observed_time <= tstars & observed_event) + 2*(observed_time > tstars) + 3*(observed_time >= 0 & !observed_event)
  bsc = rep(NA, length(tstars))
  bsc[category == 1] = pred_func(tstars[category == 1])^2 / marg_cens_func(observed_time)
  bsc[category == 2] = (1 - pred_func(tstars[category == 2]))^2 / marg_cens_func(tstars[category == 2])
  bsc[category == 3] = 0
  bsc
}

bsc_func = function(tstars) { rowMeans(sapply(1:n, function(pat_i) indiv_patient_bsc(pat_i, tstars))) }

weight_func = function(tstars) { (1 - marg_surv_func(tstars)) / (1 - marg_surv_func(max_time)) }

# Be slack and do trapezoidal int. with a fine grid. It should be possible
# to calculate the int. exactly but I cbfed.
int_grid = seq(min_time, max_time, length.out = 1e3)
bsc_vals = bsc_func(int_grid)
weight_vals = weight_func(int_grid)
int_vals = bsc_vals * weight_vals
ibsc = (2*sum(int_vals) - int_vals[1] - int_vals[length(int_vals)]) * (diff(range(int_grid))) /
length(int_grid)

return(list(bsc = bsc_vals, weights = weight_vals, eval_times = int_grid, ibsc = ibsc))
}

```

Calculate survival probability predictions for each of the models, on the validation data.

```

ibs_times = sort(unique(data.val$Time))
ibs_preds_gg = as.matrix(t(sapply(summary(fit.gg, newdata = data.val, type = "survival", t = ibs_times),
  function(t) survfit(fit.gg, newdata = data.val, type = "survival", t = t))))
ibs_preds_gg2 = as.matrix(t(sapply(summary(fit.gg2, newdata = data.val, type = "survival", t = ibs_times),
  function(t) survfit(fit.gg2, newdata = data.val, type = "survival", t = t))))
temp_cox_preds = survfit(fit.cph, newdata = data.val)
ibs_preds_cph = simplify2array(tapply(1:length(temp_cox_preds$time), rep(names(temp_cox_preds$strata), length(temp_cox_preds$time)),
  function(strat_i) approxfun(temp_cox_preds$time, temp_cox_preds$surv[strat_i], yleft = 1, yright = 0, method = "constant")),
  MARGINs = 2)
ibs_preds_cph = t(ibs_preds_cph[,rownames(data.val)])
temp_rsf_preds = predict(fit.rsf, newdata = data.val)
ibs_preds_rsf = t(apply(temp_rsf_preds$survival, 1, function(surv) approxfun(temp_rsf_preds$time, surv, yleft = 1, yright = 0, method = "constant")))
# Patients (from data.val) are in rows, times (from ibs_times) in columns.

# Add a no-information KM predictor
temp_km0 = survfit(Surv(Time, DSD) ~ 1, data)
ibs_preds_km0 = t(matrix(rep(approx(temp_km0$time, temp_km0$surv, xout = ibs_times, method = "constant", yleft = 1, yright = 0),
  length(ibs_times)), nrow = nrow(data.val)))
ibs_preds_all = list(gg = ibs_preds_gg, gg2 = ibs_preds_gg2, cph = ibs_preds_cph, rsf = ibs_preds_rsf, km0 = ibs_preds_km0)

```

```

val.prob.times = seq(0, max(data.val$Time), 1)

temp.coefs = coef(fit.gg)
val.linpred.gg = sapply(1:length(temp.coefs), function(coef_i) {
  # if (names(temp.coefs)[coef_i] == "SexMTRUE") {
  #   rep(0, nrow(data.val))
  # } else
  if (names(temp.coefs)[coef_i] %in% colnames(data.val)) {
    temp.coefs[coef_i] * data.val[,names(temp.coefs)[coef_i]]
  } else if (gsub("TRUE$", "", names(temp.coefs)[coef_i]) %in% colnames(data.val)) {
    temp.coefs[coef_i] * data.val[,gsub("TRUE$", "", names(temp.coefs)[coef_i])]
  } else {
    rep(0, nrow(data.val))
  } })
val.linpred.gg = -rowSums(val.linpred.gg) # Negate to bring into concordance with the direction of Co
temp = summary(fit.gg, newdata = data.val, ci = FALSE)
val.prob.gg = sapply(temp, function(x) approx(x[,1], x[,2], xout = val.prob.times, yleft = 1, yright = 0))
colnames(val.prob.gg) = rownames(data.val)

temp.coefs = coef(fit.gg2)
val.linpred.gg2 = sapply(1:length(temp.coefs), function(coef_i) {
  # if (names(temp.coefs)[coef_i] == "SexMTRUE") {
  #   rep(0, nrow(data.val))
  # } else
  if (names(temp.coefs)[coef_i] %in% colnames(data.val)) {
    temp.coefs[coef_i] * data.val[,names(temp.coefs)[coef_i]]
  } else if (gsub("TRUE$", "", names(temp.coefs)[coef_i]) %in% colnames(data.val)) {
    temp.coefs[coef_i] * data.val[,gsub("TRUE$", "", names(temp.coefs)[coef_i])]
  } else {
    rep(0, nrow(data.val))
  } })
val.linpred.gg2 = -rowSums(val.linpred.gg2) # Negate to bring into concordance with the direction of Co
temp = summary(fit.gg2, newdata = data.val, ci = FALSE)
val.prob.gg2 = sapply(temp, function(x) approx(x[,1], x[,2], xout = val.prob.times, yleft = 1, yright = 0))
colnames(val.prob.gg2) = rownames(data.val)

val.linpred.cph = predict(fit.cph, newdata = data.val)
temp = survfit(fit.cph, newdata = data.val)
val.prob.cph = simplify2array(tapply(1:length(temp$surv), rep(names(temp$strata), temp$strata), function(x) {
  temp = predict(fit.rsrf, newdata = data.val)
  # val.linpred.rsrf = temp$predicted
  # Median survival time:
  val.linpred.rsrf = apply(temp$survival, 1, function(s1) {
    sfunc = approxfun(temp$time.interest, s1, yleft = 1, yright = 0, rule = 2)
    med = uniroot(function(x) sfunc(x) - 0.5, lower = min(temp$time.interest), upper = max(temp$time.interest))
    med
  })
  val.linpred.rsrf = -val.linpred.rsrf
  val.prob.rsrf = apply(temp$survival, 1, function(s1) approx(temp$time.interest, s1, xout = val.prob.times))
  colnames(val.prob.rsrf) = rownames(data.val)

  summary(coxph(Surv(Time, DSD) ~ val.linpred.gg, data.val))

```

```
## Call:
## coxph(formula = Surv(Time, DSD) ~ val.linpred.gg, data = data.val)
##
##      n= 49, number of events= 49
##
##              coef exp(coef) se(coef)      z Pr(>|z|)
## val.linpred.gg 1.54      4.68      0.45 3.43      6e-04
##
##              exp(coef) exp(-coef) lower .95 upper .95
## val.linpred.gg      4.68      0.214      1.94      11.3
##
## Concordance= 0.673 (se = 0.05 )
## Rsquare= 0.216 (max possible= 0.997 )
## Likelihood ratio test= 11.9 on 1 df, p=0.000554
## Wald test              = 11.8 on 1 df, p=0.000599
## Score (logrank) test = 12.2 on 1 df, p=0.000485

summary(coxph(Surv(Time, DSD) ~ val.linpred.gg2, data.val))

## Call:
## coxph(formula = Surv(Time, DSD) ~ val.linpred.gg2, data = data.val)
##
##      n= 49, number of events= 49
##
##              coef exp(coef) se(coef)      z Pr(>|z|)
## val.linpred.gg2 1.78      5.93      0.51 3.49      0.00048
##
##              exp(coef) exp(-coef) lower .95 upper .95
## val.linpred.gg2      5.93      0.169      2.18      16.1
##
## Concordance= 0.668 (se = 0.05 )
## Rsquare= 0.216 (max possible= 0.997 )
## Likelihood ratio test= 11.9 on 1 df, p=0.000563
## Wald test              = 12.2 on 1 df, p=0.000483
## Score (logrank) test = 12.5 on 1 df, p=0.00041

summary(coxph(Surv(Time, DSD) ~ val.linpred.cph, data.val))

## Call:
## coxph(formula = Surv(Time, DSD) ~ val.linpred.cph, data = data.val)
##
##      n= 49, number of events= 49
##
##              coef exp(coef) se(coef)      z Pr(>|z|)
## val.linpred.cph 1.139      3.123      0.311 3.66      0.00025
##
##              exp(coef) exp(-coef) lower .95 upper .95
## val.linpred.cph      3.12      0.32      1.7      5.75
##
## Concordance= 0.65 (se = 0.05 )
## Rsquare= 0.236 (max possible= 0.997 )
## Likelihood ratio test= 13.2 on 1 df, p=0.000284
## Wald test              = 13.4 on 1 df, p=0.000252
## Score (logrank) test = 13.9 on 1 df, p=0.000192
```

```
summary(coxph(Surv(Time, DSD) ~ val.linpred.rsfs, data.val))

## Call:
## coxph(formula = Surv(Time, DSD) ~ val.linpred.rsfs, data = data.val)
##
##    n= 49, number of events= 49
##
##              coef exp(coef) se(coef)      z Pr(>|z|)
## val.linpred.rsfs 0.00811  1.00814  0.00209 3.87  0.00011
##
##              exp(coef) exp(-coef) lower .95 upper .95
## val.linpred.rsfs      1.01      0.992      1      1.01
##
## Concordance= 0.663 (se = 0.05 )
## Rsquare= 0.258 (max possible= 0.997 )
## Likelihood ratio test= 14.6 on 1 df,  p=0.000133
## Wald test               = 15 on 1 df,  p=0.000107
## Score (logrank) test = 15.5 on 1 df,  p=8.4e-05

anova(coxph(Surv(Time, DSD) ~ offset(val.linpred.gg) + val.linpred.gg, data.val))

## Analysis of Deviance Table
## Cox model: response is Surv(Time, DSD)
## Terms added sequentially (first to last)
##
##              loglik Chisq Df Pr(>|Chi|)
## NULL                  -139
## val.linpred.gg        -139  1.47  1      0.23

anova(coxph(Surv(Time, DSD) ~ offset(val.linpred.gg2) + val.linpred.gg2, data.val))

## Analysis of Deviance Table
## Cox model: response is Surv(Time, DSD)
## Terms added sequentially (first to last)
##
##              loglik Chisq Df Pr(>|Chi|)
## NULL                  -140
## val.linpred.gg2       -139  2.32  1      0.13

anova(coxph(Surv(Time, DSD) ~ offset(val.linpred.cph) + val.linpred.cph, data.val))

## Analysis of Deviance Table
## Cox model: response is Surv(Time, DSD)
## Terms added sequentially (first to last)
##
##              loglik Chisq Df Pr(>|Chi|)
## NULL                  -138
## val.linpred.cph       -138  0.2  1      0.66

anova(coxph(Surv(Time, DSD) ~ offset(val.linpred.rsfs) + val.linpred.rsfs, data.val))

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, : Ran out of
iterations and did not converge
## Error in fitter(X, Y, strats, offset, init, control, weights = weights, : NA/NaN/Inf in
foreign function call (arg 6)
```

```
summary(coxph(Surv(Time, DSD) ~ offset(val.linpred.gg) + SexM + AgeCent + LocBody + SizeCent + A2 + A4,
## Call:
## coxph(formula = Surv(Time, DSD) ~ offset(val.linpred.gg) + SexM +
##      AgeCent + LocBody + SizeCent + A2 + A4, data = data.val)
##
##      n= 49, number of events= 49
##
##              coef exp(coef) se(coef)      z Pr(>|z|)
## SexMTRUE      0.10665   1.11255  0.37675  0.28   0.78
## AgeCent      -0.00735   0.99268  0.02276 -0.32   0.75
## LocBodyTRUE   0.29902   1.34854  0.37945  0.79   0.43
## SizeCent      0.00391   1.00392  0.01002  0.39   0.70
## A2TRUE        0.30761   1.36017  0.49719  0.62   0.54
## A4TRUE        0.27581   1.31760  0.39889  0.69   0.49
##
##              exp(coef) exp(-coef) lower .95 upper .95
## SexMTRUE          1.113      0.899   0.532   2.33
## AgeCent           0.993      1.007   0.949   1.04
## LocBodyTRUE       1.349      0.742   0.641   2.84
## SizeCent          1.004      0.996   0.984   1.02
## A2TRUE            1.360      0.735   0.513   3.60
## A4TRUE            1.318      0.759   0.603   2.88
##
## Concordance= 0.672 (se = 0.05 )
## Rsquare= 0.064 (max possible= 0.997 )
## Likelihood ratio test= 3.25 on 6 df, p=0.777
## Wald test           = 3.3 on 6 df, p=0.77
## Score (logrank) test = 3.36 on 6 df, p=0.763

summary(coxph(Surv(Time, DSD) ~ offset(val.linpred.gg2) + SexM + AgeCent + LocBody + SizeCent + A2 + A4,
## Call:
## coxph(formula = Surv(Time, DSD) ~ offset(val.linpred.gg2) + SexM +
##      AgeCent + LocBody + SizeCent + A2 + A4, data = data.val)
##
##      n= 49, number of events= 49
##
##              coef exp(coef) se(coef)      z Pr(>|z|)
## SexMTRUE      0.14695   1.15830  0.37675  0.39   0.70
## AgeCent      0.00300   1.00301  0.02276  0.13   0.90
## LocBodyTRUE   0.23722   1.26772  0.37945  0.63   0.53
## SizeCent      0.00846   1.00849  0.01002  0.84   0.40
## A2TRUE        0.33860   1.40298  0.49719  0.68   0.50
## A4TRUE        0.31901   1.37576  0.39889  0.80   0.42
##
##              exp(coef) exp(-coef) lower .95 upper .95
## SexMTRUE          1.16      0.863   0.554   2.42
## AgeCent           1.00      0.997   0.959   1.05
## LocBodyTRUE       1.27      0.789   0.603   2.67
## SizeCent          1.01      0.992   0.989   1.03
## A2TRUE            1.40      0.713   0.529   3.72
## A4TRUE            1.38      0.727   0.630   3.01
##
## Concordance= 0.672 (se = 0.05 )
```

```
## Rsquare= 0.081 (max possible= 0.997 )
## Likelihood ratio test= 4.13 on 6 df, p=0.659
## Wald test = 4.14 on 6 df, p=0.658
## Score (logrank) test = 4.23 on 6 df, p=0.646

summary(coxph(Surv(Time, DSD) ~ offset(val.linpred.cph) + SexM + AgeCent + LocBody + SizeCent + A2 + A4, data = data.val))

## Call:
## coxph(formula = Surv(Time, DSD) ~ offset(val.linpred.cph) + SexM +
## AgeCent + LocBody + SizeCent + A2 + A4, data = data.val)
##
## n= 49, number of events= 49
##
##
```

	coef	exp(coef)	se(coef)	z	Pr(> z)
## SexMTRUE	-2.37e-01	7.89e-01	3.77e-01	-0.63	0.53
## AgeCent	-7.35e-03	9.93e-01	2.28e-02	-0.32	0.75
## LocBodyTRUE	1.28e-01	1.14e+00	3.79e-01	0.34	0.74
## SizeCent	5.99e-05	1.00e+00	1.00e-02	0.01	1.00
## A2TRUE	6.71e-02	1.07e+00	4.97e-01	0.13	0.89
## A4TRUE	1.42e-01	1.15e+00	3.99e-01	0.36	0.72

```
##
##
```

	exp(coef)	exp(-coef)	lower .95	upper .95
## SexMTRUE	0.789	1.267	0.377	1.65
## AgeCent	0.993	1.007	0.949	1.04
## LocBodyTRUE	1.137	0.880	0.540	2.39
## SizeCent	1.000	1.000	0.981	1.02
## A2TRUE	1.069	0.935	0.404	2.83
## A4TRUE	1.152	0.868	0.527	2.52

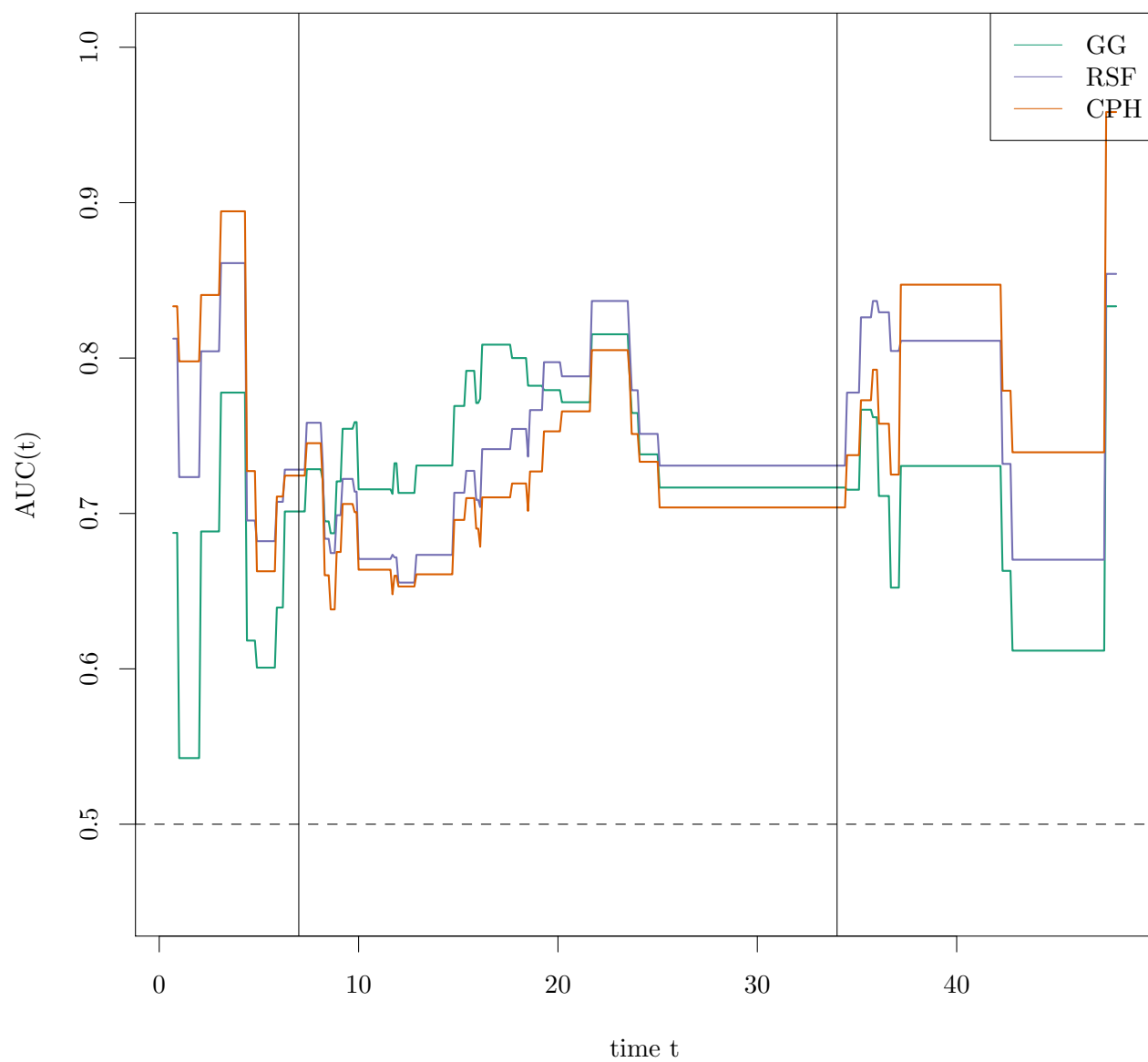
```
##
## Concordance= 0.672 (se = 0.05 )
## Rsquare= 0.015 (max possible= 0.996 )
## Likelihood ratio test= 0.73 on 6 df, p=0.994
## Wald test = 0.72 on 6 df, p=0.994
## Score (logrank) test = 0.72 on 6 df, p=0.994

summary(coxph(Surv(Time, DSD) ~ offset(val.linpred.rsfcph) + SexM + AgeCent + LocBody + SizeCent + A2 + A4, data = data.val))

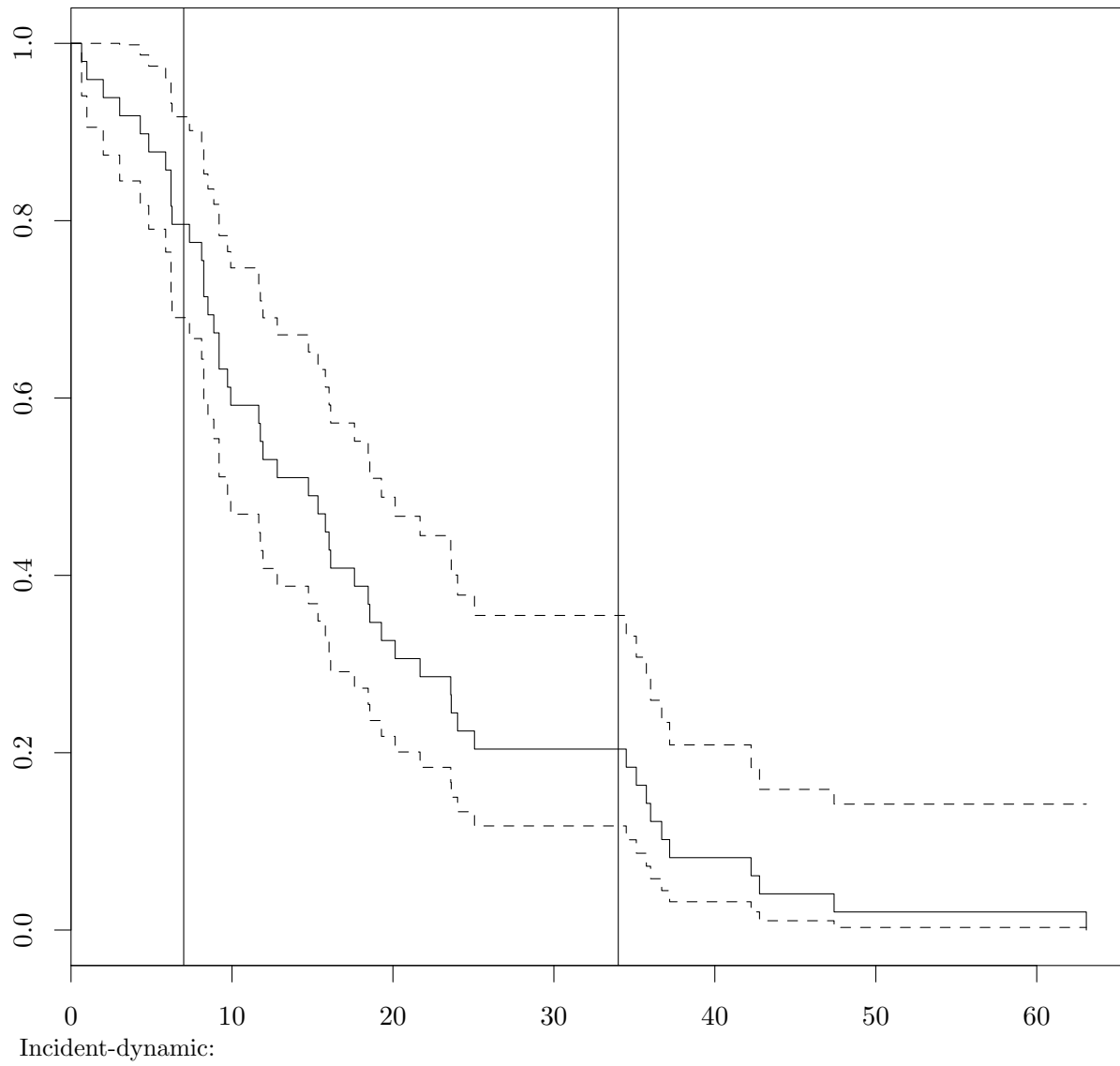
## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, : Ran out of
iterations and did not converge
## Error in fitter(X, Y, strats, offset, init, control, weights = weights, : NA/NaN/Inf in
foreign function call (arg 6)
```

Cumulative-dynamic:

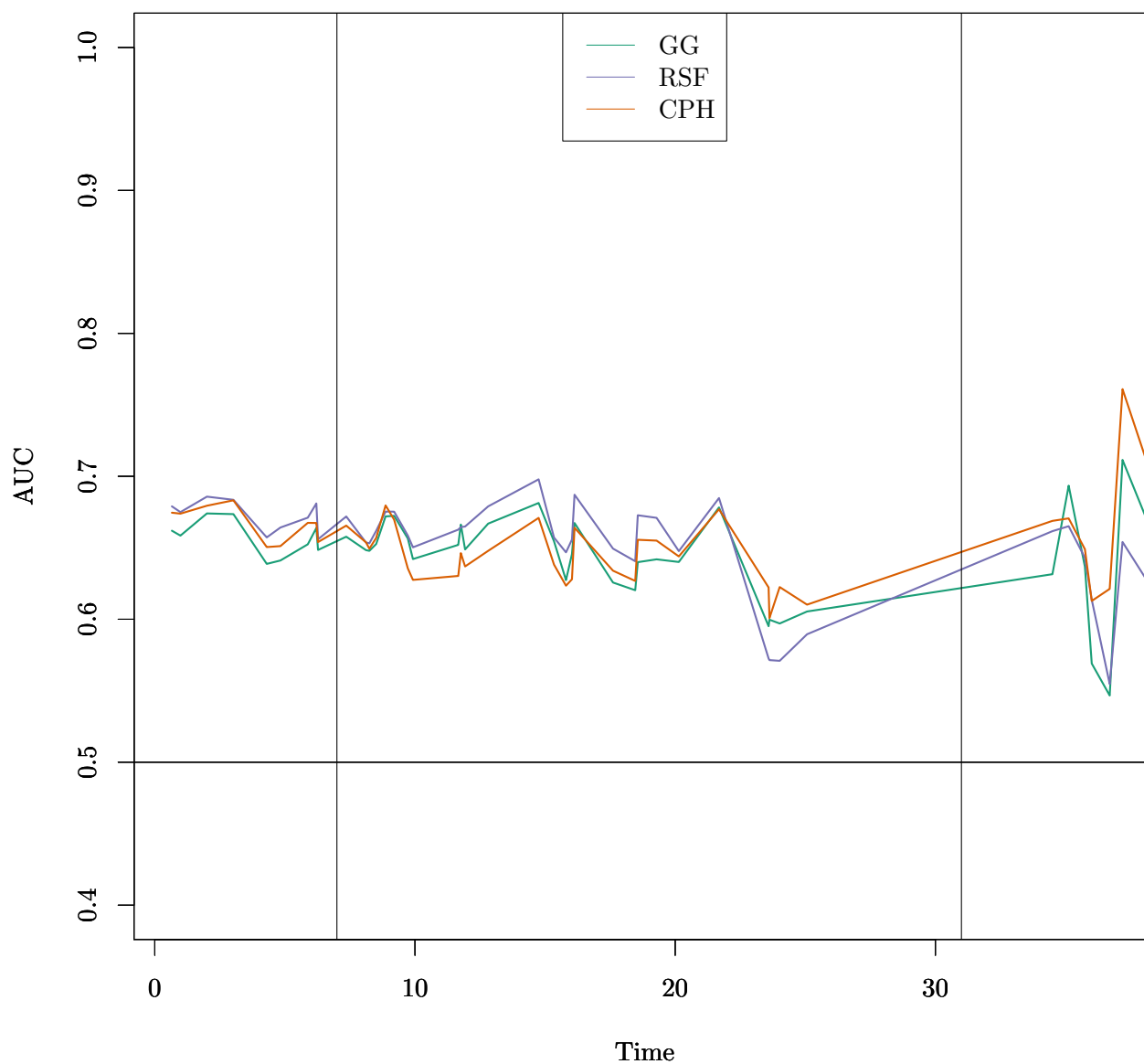
```
temp.times = seq(0.1, 48, 0.1)
temp.gg = timeROC(T = data.val$Time/365.25*12, delta = data.val$DSD*1, marker = val.linpred.gg, cause =
temp.gg2 = timeROC(T = data.val$Time/365.25*12, delta = data.val$DSD*1, marker = val.linpred.gg2, cause =
temp.rsfcph = timeROC(T = data.val$Time/365.25*12, delta = data.val$DSD*1, marker = val.linpred.rsfcph, cause =
temp.cph = timeROC(T = data.val$Time/365.25*12, delta = data.val$DSD*1, marker = val.linpred.cph, cause =
plotAUCcurve(temp.gg, conf.int = FALSE, add = FALSE, col = pal["GG"])
plotAUCcurve(temp.rsfcph, conf.int = FALSE, add = TRUE, col = pal["RSF"])
plotAUCcurve(temp.cph, conf.int = FALSE, add = TRUE, col = pal["CPH"])
legend("topright", legend = c("GG", "RSF", "CPH"), col = pal[c("GG", "RSF", "CPH")], lty = "solid")
abline(v = c(7, 34))
```



```
plot(survfit(Surv(data.val$Time/365.25*12, data.val$DSD) ~ 1))
abline(v = c(7, 34))
```



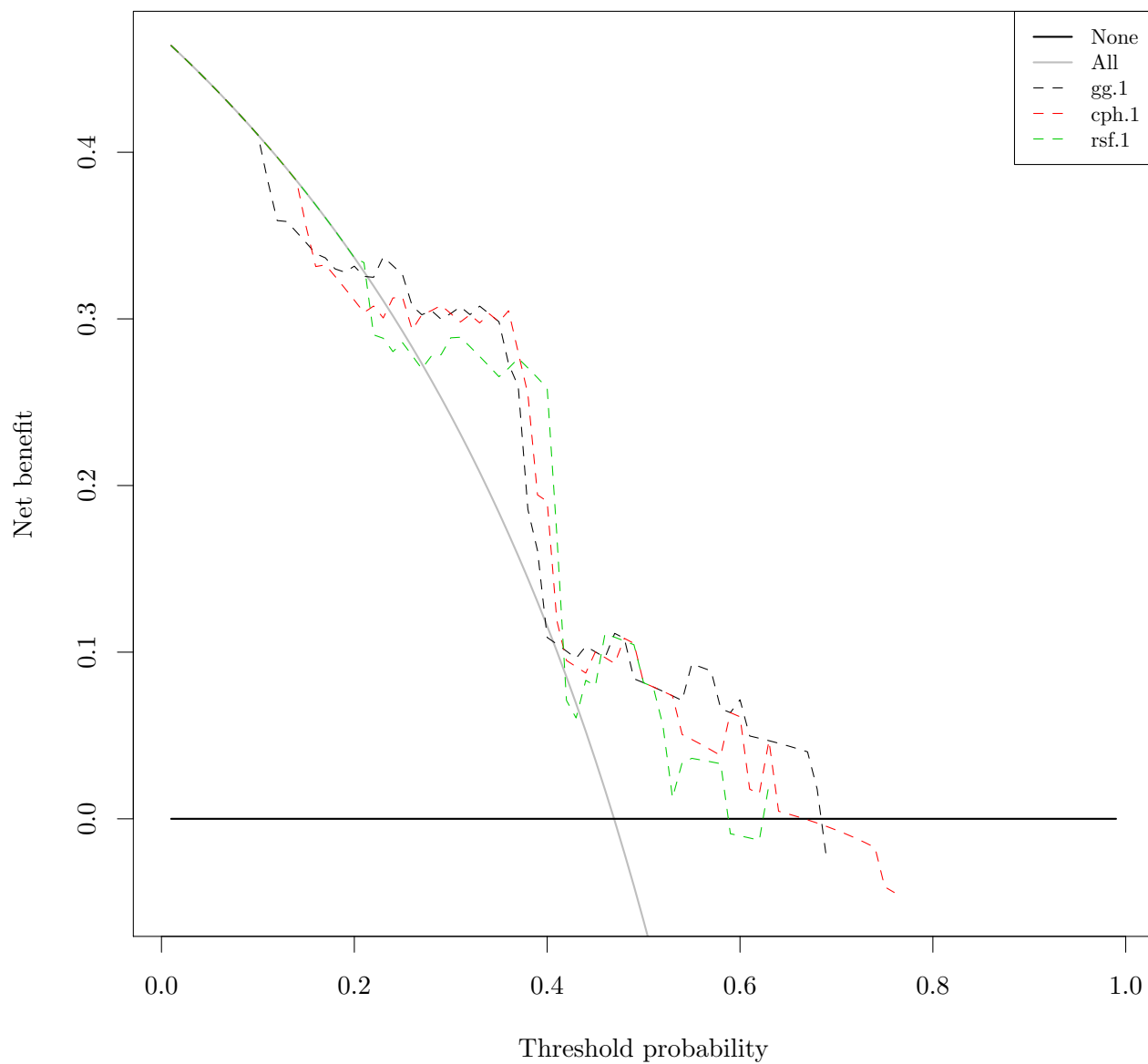
```
library(risksetROC)
invisible(risksetAUC(data.val$Time/365.25*12, status = data.val$DSD, marker = val.linpred.gg, tmax = 36,
par(new = TRUE)
invisible(risksetAUC(data.val$Time/365.25*12, status = data.val$DSD, marker = val.linpred.rsfcph, tmax = 36,
par(new = TRUE)
invisible(risksetAUC(data.val$Time/365.25*12, status = data.val$DSD, marker = val.linpred.cph, tmax = 36,
par(new = TRUE)
legend("top", legend = c("GG", "RSF", "CPH"), col = pal[c("GG", "RSF", "CPH")], lty = "solid")
abline(v = c(7, 31))
```

Decision curve analysis.

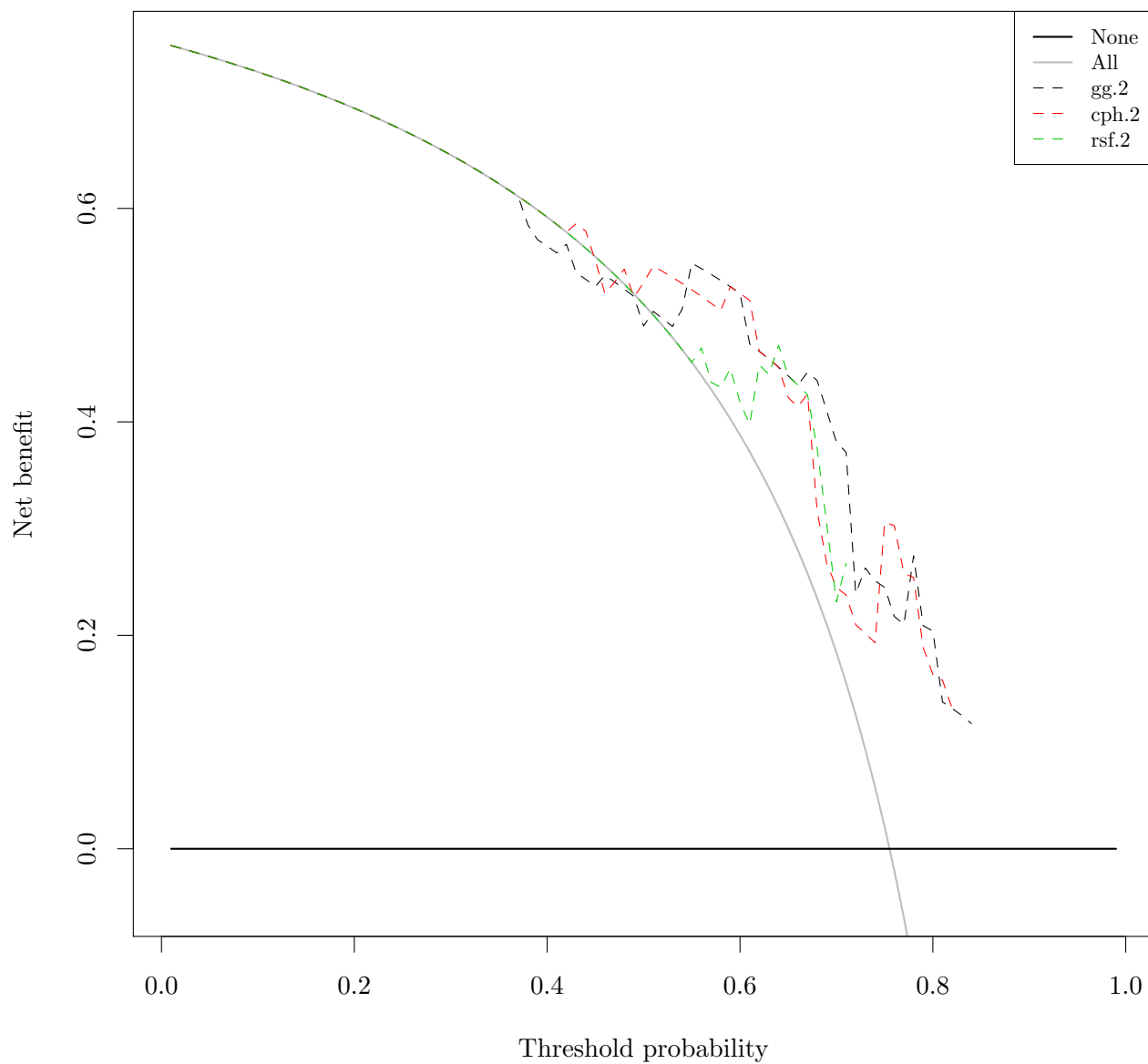
```
source("stdca.R")
temp.data = data.frame(Time = data.val$Time, DSD = data.val$DSD*1,
  gg.1 = 1-val.prob.gg[val.prob.times == 365,], gg.2 = 1-val.prob.gg[val.prob.times == 365*2,], gg.3 =
  cph.1 = 1-val.prob.cph[val.prob.times == 365,], cph.2 = 1-val.prob.cph[val.prob.times == 365*2,], cp
  rsf.1 = 1-val.prob.rsrf[val.prob.times == 365,], rsf.2 = 1-val.prob.rsrf[val.prob.times == 365*2,], rs
invisible(stdca(data = temp.data, outcome = "DSD", ttoutcome = "Time", predictors = c("gg.1", "cph.1", "rsf.1", "rsf.2", "cph.2", "gg.3")))

## [1] "gg.1: No observations with risk greater than 70% that have followup through the timepoint selected"
## [2] "cph.1: No observations with risk greater than 77% that have followup through the timepoint selected"
## [3] "rsf.1: No observations with risk greater than 64%, and therefore net benefit not calculable in this timepoint"
```

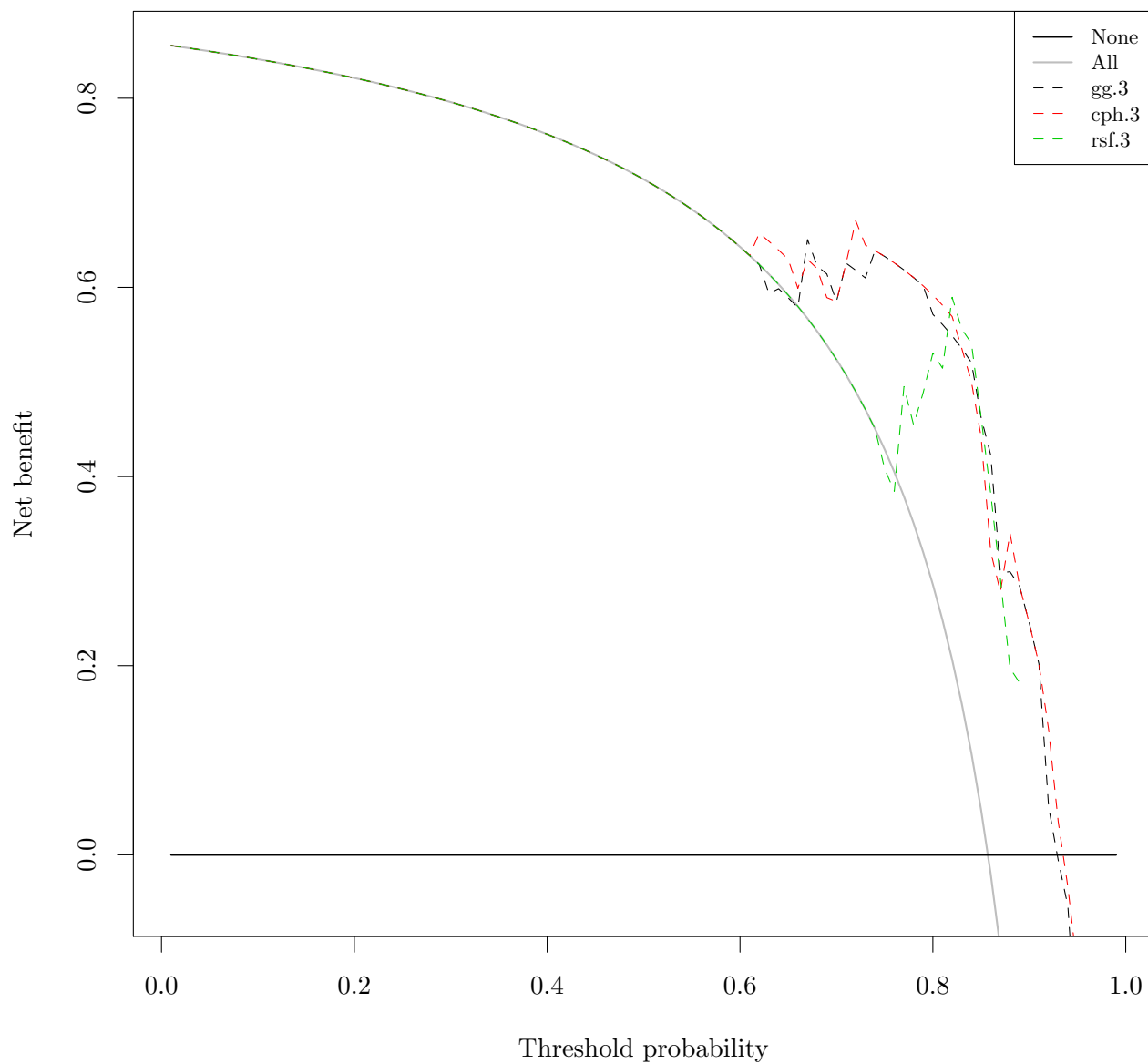


```
invisible(stdca(data = temp.data, outcome = "DSD", ttoutcome = "Time", predictors = c("gg.2", "cph.2", "rsf.2")))
```

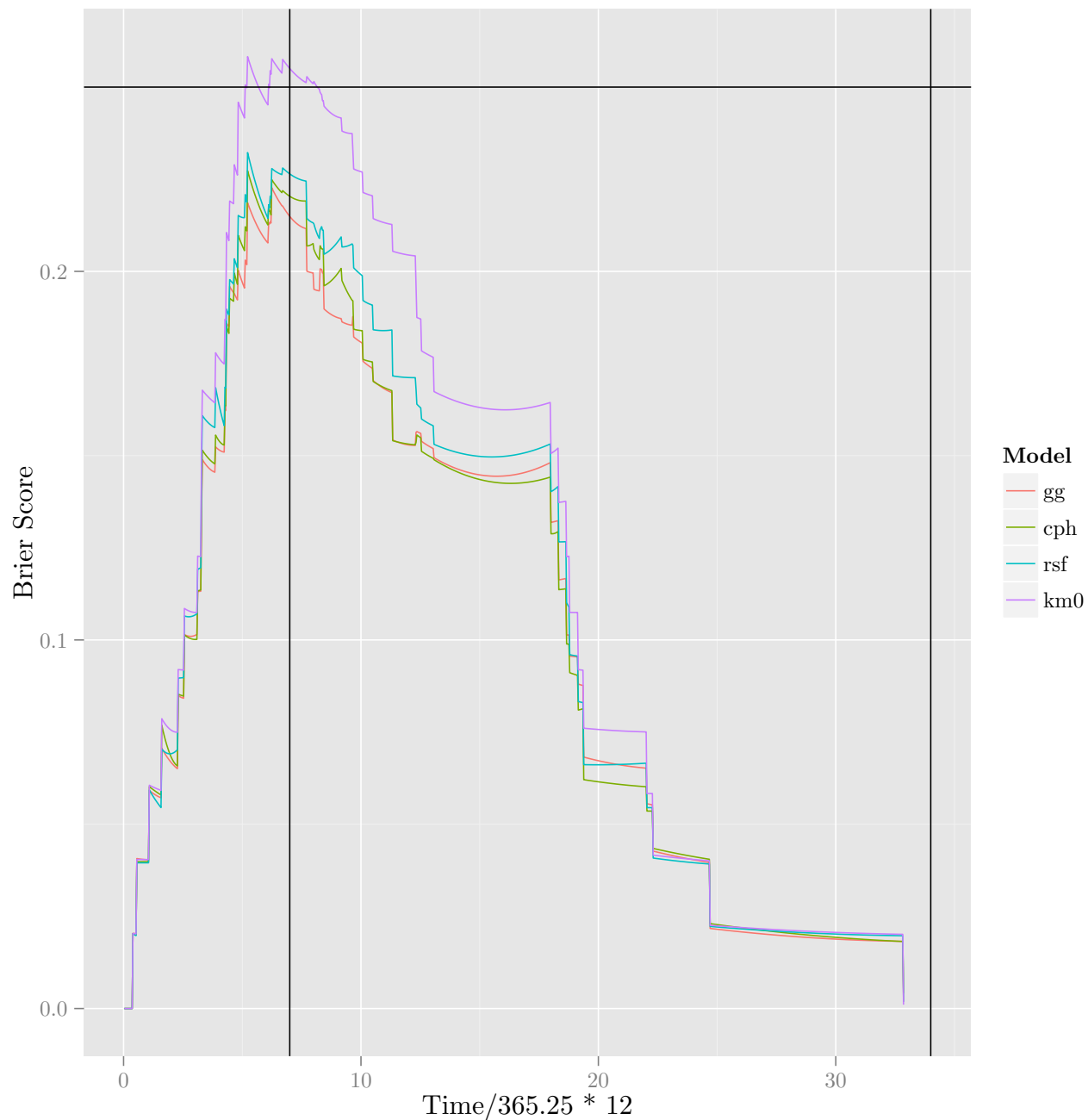
```
## [1] "gg.2: No observations with risk greater than 85% that have followup through the timepoint selected"
## [2] "cph.2: No observations with risk greater than 83% that have followup through the timepoint selected"
## [3] "rsf.2: No observations with risk greater than 72% that have followup through the timepoint selected"
```



```
invisible(stdca(data = temp.data, outcome = "DSD", ttoutcome = "Time", predictors = c("gg.3", "cph.3", "rsf.3"))
## [1] "gg.3: No observations with risk greater than 97% that have followup through the timepoint selected"
## [2] "cph.3: No observations with risk greater than 97% that have followup through the timepoint selected"
## [3] "rsf.3: No observations with risk greater than 90% that have followup through the timepoint selected"
```



```
temp = sapply(list(gg = ibs_preds_gg, cph = ibs_preds_cph, rsf = ibs_preds_rsrf, km0 = ibs_preds_km0), fun = function(x) {
  temp = melt(temp)
  colnames(temp) = c("Time", "Model", "BS")
  ggplot(temp, aes(x = Time/365.25*12, y = BS, colour = Model)) + geom_line() + ylab("Brier Score") + geom_vline(x = 0.85, colour = "black", linetype = "solid")
```



BCA bootstrapping on the differences.

```
set.seed(20150208)
ibsc_boots2 = boot(data.val, statistic = function(d, i) {
  gg = calcIBS(Surv(d$Time, d$DSD)[i,], ibs_preds_gg[i,], ibs_times, 34*365.25/12, 7*365.25/12)$ib
  cph = calcIBS(Surv(d$Time, d$DSD)[i,], ibs_preds_cph[i,], ibs_times, 34*365.25/12, 7*365.25/12)$ib
  rsf = calcIBS(Surv(d$Time, d$DSD)[i,], ibs_preds_rsf[i,], ibs_times, 34*365.25/12, 7*365.25/12)$ib
  km0 = calcIBS(Surv(d$Time, d$DSD)[i,], ibs_preds_km0[i,], ibs_times, 34*365.25/12, 7*365.25/12)$ib
  c(gg - km0, cph - km0, rsf - km0, gg - rsf, cph - rsf, gg - cph)
}, R = 1000)
ibsc_boots2_ci = t(sapply(1:length(ibsc_boots2$t0), function(i) boot.ci(ibsc_boots2, index = i, type = '
rownames(ibsc_boots2_ci) = c("gg-km0", "cph-km0", "rsf-km0", "gg-rsf", "cph-rsf", "gg-cph")
colnames(ibsc_boots2_ci) = c("level", "orderi1", "orderi2", "lci", "uci")
```

```

ibsc_boots2

##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = data.val, statistic = function(d, i) {
##   gg = calcIBS(Surv(d$Time, d$DSD)[i, ], ibs_preds_gg[i, ],
##   ibs_times, 34 * 365.25/12, 7 * 365.25/12)$ibs
##   cph = calcIBS(Surv(d$Time, d$DSD)[i, ], ibs_preds_cph[i,
##   ], ibs_times, 34 * 365.25/12, 7 * 365.25/12)$ibs
##   rsf = calcIBS(Surv(d$Time, d$DSD)[i, ], ibs_preds_rsfc[i,
##   ], ibs_times, 34 * 365.25/12, 7 * 365.25/12)$ibs
##   km0 = calcIBS(Surv(d$Time, d$DSD)[i, ], ibs_preds_km0[i,
##   ], ibs_times, 34 * 365.25/12, 7 * 365.25/12)$ibs
##   c(gg - km0, cph - km0, rsf - km0, gg - rsf, cph - rsf, gg -
##   cph)
## }, R = 1000)
##
##
## Bootstrap Statistics :
##   original   bias    std. error
## t1*   -21.062  0.78762      9.856
## t2*   -20.209  0.72053      9.039
## t3*   -14.505  0.34307      4.952
## t4*    -6.557  0.44455      5.798
## t5*    -5.704  0.37746      4.772
## t6*    -0.853  0.06709      2.123

ibsc_boots2_ci

##      level orderi1 orderi2      lci      uci
## gg-km0   0.95   19.71   969.3 -39.793 -2.523
## cph-km0   0.95   15.13   961.7 -38.853 -4.508
## rsf-km0   0.95   14.19   960.0 -24.557 -5.655
## gg-rsf    0.95   24.04   974.9 -17.721  5.620
## cph-rsf   0.95   16.32   963.5 -15.865  2.877
## gg-cph    0.95   37.22   985.5  -4.343  4.087

```

All models perform equivalently on the validation set. Select the simplest: gg.

Final model fitting:

```

temp = coxph(Surv(Time, DSD) ~ strata(SexM) + AgeCent + LocBody + SizeCent + SizePlus + A2 + A4, data =
sel = abs(resid(temp, type = "deviance")) <= 2.5 & apply(abs(resid(temp, type = "dfbetas")), 1, max) <=
data.all.polished = data.all[sel,]
nrow(data.all)

## [1] 249

nrow(data.all.polished)

## [1] 240

fit.final.gg = flexsurvreg(Surv(Time, DSD) ~ SexM + LocBody + SizeCent + A2 + A4,

```

```

anc = list(
  sigma = ~ SexM,
  Q = ~ SexM),
data = data.all.polished, dist = "gengamma")

fit.final.cph = coxph(Surv(Time, DSD) ~ strata(SexM) + LocBody + SizeCent + A2 + A4, data = data.all.polished,
set.seed(20150208)
fit.final.rsfc = rfsr(Surv(Time, DSD) ~ SexM + AgeCent + LocBody + SizeCent + A2 + A4, data = data.all.polished,
fit.final.km0 = survfit(Surv(Time, DSD) ~ 1, data.all)
saveRDS(list(gg = fit.final.gg, km0 = fit.final.km0, cph = fit.final.cph, rsf = fit.final.rsfc, data.train = data.all.polished,
fit.final.gg

##
## Call:
## flexsurvreg(formula = Surv(Time, DSD) ~ SexM + LocBody + SizeCent + A2 + A4, anc = list(sigma = ~ SexM, Q = ~ SexM),
##
## Estimates:
##
## data mean est L95% U95% se
## mu NA 6.47851 6.18670 6.77032 0.14889
## sigma NA 0.75029 0.65968 0.85335 0.04927
## Q NA 0.02879 -0.50416 0.56173 0.27192
## SexMTRUE 0.50000 0.37324 0.07777 0.66872 0.15076
## LocBodyTRUE 0.18333 -0.21498 -0.45459 0.02464 0.12226
## SizeCent 3.55833 -0.00887 -0.01480 -0.00295 0.00302
## A2TRUE 0.15417 -0.37292 -0.61497 -0.13088 0.12349
## A4TRUE 0.75000 -0.38434 -0.58916 -0.17952 0.10450
## sigma(SexMTRUE) 0.50000 -0.24520 -0.45420 -0.03621 0.10663
## Q(SexMTRUE) 0.50000 0.76301 0.07052 1.45551 0.35332
##
## exp(est) L95% U95%
## mu NA NA NA
## sigma NA NA NA
## Q NA NA NA
## SexMTRUE 1.45244 1.08087 1.95174
## LocBodyTRUE 0.80656 0.63471 1.02495
## SizeCent 0.99117 0.98531 0.99706
## A2TRUE 0.68872 0.54066 0.87732
## A4TRUE 0.68090 0.55479 0.83567
## sigma(SexMTRUE) 0.78255 0.63496 0.96444
## Q(SexMTRUE) 2.14473 1.07306 4.28668
##
## N = 240, Events: 231, Censored: 9
## Total time at risk: 141440
## Log-likelihood = -1658, df = 10
## AIC = 3337

fit.final.cph

## Call:
## coxph(formula = Surv(Time, DSD) ~ strata(SexM) + LocBody + SizeCent + A2 + A4, data = data.all.polished, model = TRUE, x = TRUE,
## y = TRUE)
##
##
##
## coef exp(coef) se(coef) z p

```

```
## LocBodyTRUE 0.402      1.50    0.1884 2.13 0.0330
## SizeCent    0.013      1.01    0.0049 2.64 0.0082
## A2TRUE      0.634      1.89    0.1946 3.26 0.0011
## A4TRUE      0.519      1.68    0.1637 3.17 0.0015
##
## Likelihood ratio test=47.1 on 4 df, p=1.42e-09 n= 240, number of events= 231
```

```
save.image("05_train_NSWPCN_2.rda")
```

7 Session information

```
sessionInfo()

## R version 3.1.1 (2014-07-10)
## Platform: x86_64-unknown-linux-gnu (64-bit)
##
## locale:
##  [1] LC_CTYPE=en_AU.UTF-8      LC_NUMERIC=C
##  [3] LC_TIME=en_AU.UTF-8      LC_COLLATE=en_AU.UTF-8
##  [5] LC_MONETARY=en_AU.UTF-8  LC_MESSAGES=en_AU.UTF-8
##  [7] LC_PAPER=en_AU.UTF-8     LC_NAME=en_AU.UTF-8
##  [9] LC_ADDRESS=en_AU.UTF-8   LC_TELEPHONE=en_AU.UTF-8
## [11] LC_MEASUREMENT=en_AU.UTF-8 LC_IDENTIFICATION=en_AU.UTF-8
##
## attached base packages:
## [1] parallel  methods    splines    stats      graphics  grDevices  utils
## [8] datasets  base
##
## other attached packages:
##  [1] risksetROC_1.0.4      energy_1.6.2          RColorBrewer_1.0-5
##  [4] timeROC_0.2           timereg_1.8.6         mvtnorm_1.0-1
##  [7] pec_2.4.4             boot_1.3-13          MASS_7.3-35
## [10] ggplot2_1.0.0         plyr_1.8.1            reshape2_1.4
## [13] randomForestSRC_1.5.5 flexsurv_0.5          glmulti_1.0.7
## [16] rJava_0.9-6           survival_2.37-7       tikzDevice_0.8.1
## [19] knitr_1.8
##
## loaded via a namespace (and not attached):
##  [1] codetools_0.2-9      colorspace_1.2-4      deSolve_1.11          digest_0.6.4
##  [5] evaluate_0.5.5       filehash_2.2-2        foreach_1.4.2         formatR_1.0
##  [9] grid_3.1.1           gtable_0.1.2          highr_0.4             iterators_1.0.7
## [13] labeling_0.3         lava_1.3              muhaz_1.2.6           munsell_0.4.2
## [17] prodlim_1.5.1        proto_0.3-10          Rcpp_0.11.3           scales_0.2.4
## [21] stringr_0.6.2        tools_3.1.1
```