fflchain: RRPlot Demo

Jose Tamez

2023-05-18

Contents

1 RRPLOTS and flchain **1**
1.1 Exploring Raw Features with RRPlot 2
1.2 Reporting the Metrics 16
1.3 Cox Modeling 18
1.4 Cox Model Performance 19

```
library(survival)
library(FRESA.CAD)
```

```
## Loading required package: Rcpp
```

```
## Loading required package: stringr
```

```
## Loading required package: miscTools
```

```
## Loading required package: Hmisc
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
## Loading required package: pROC
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
#library(corrplot)
#source("~/GitHub/FRESA.CAD/R/RRPlot.R")
op <- par(no.readonly = TRUE)
pander::panderOptions('digits', 3)
pander::panderOptions('keep.trailing.zeros',TRUE)
```

1 RRPLOTS and flchain

```
odata <- flchain
odata$chapter <- NULL
pander::pander(table(odata$death))
```

0	1
5705	2169

```
rownames(odata) <- c(1:nrow(odata))
data <- as.data.frame(model.matrix(Surv(futime,death)~.,odata))

data$`(Intercept)` <- NULL

dataFL <- as.data.frame(cbind(time=odata[rownames(data),"futime"],
                              status=odata[rownames(data),"death"],
                              data))
pander::pander(table(dataFL$status))
```

0	1
4562	1962

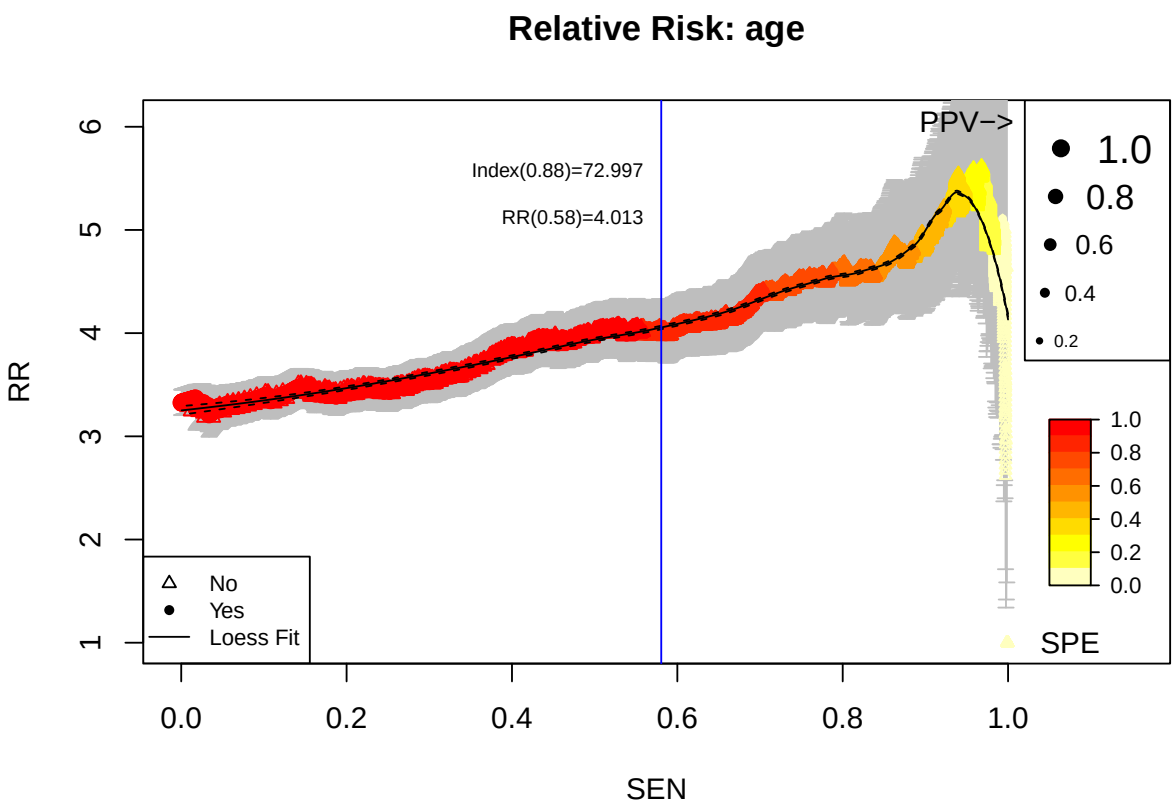
```
dataFL$time <- dataFL$time/365
```

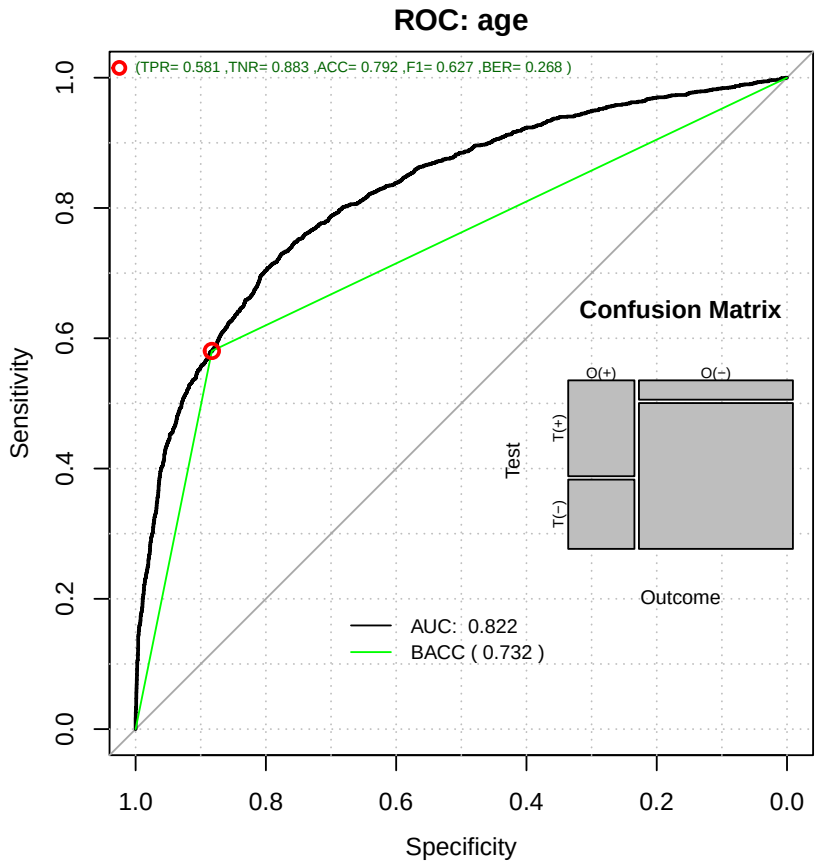
1.1 Exploring Raw Features with RRPlot

```
convar <- colnames(dataFL)[lapply(apply(dataFL,2,unique),length) > 10]
convar <- convar[convar != "time"]
topvar <- univariate_BinEnsemble(dataFL[,c("status",convar)],"status")
pander::pander(topvar)
```

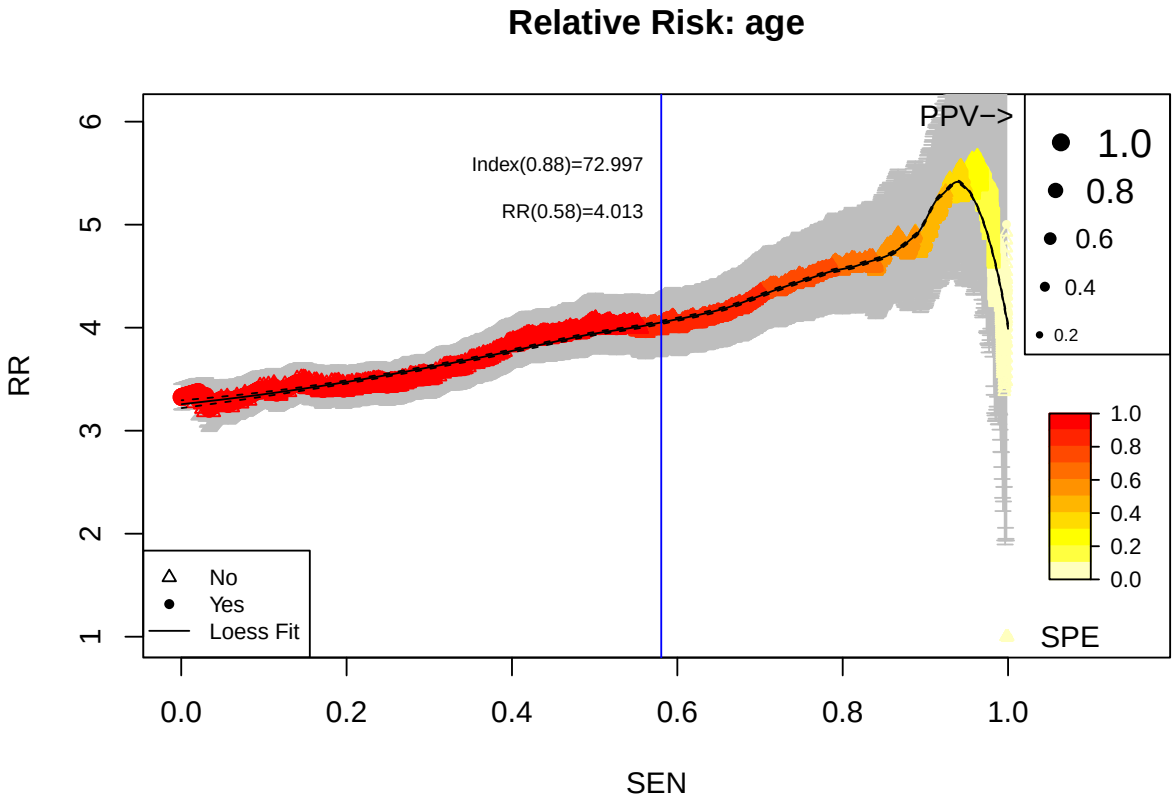
age	kappa	lambda	creatinine
0	0	0	0

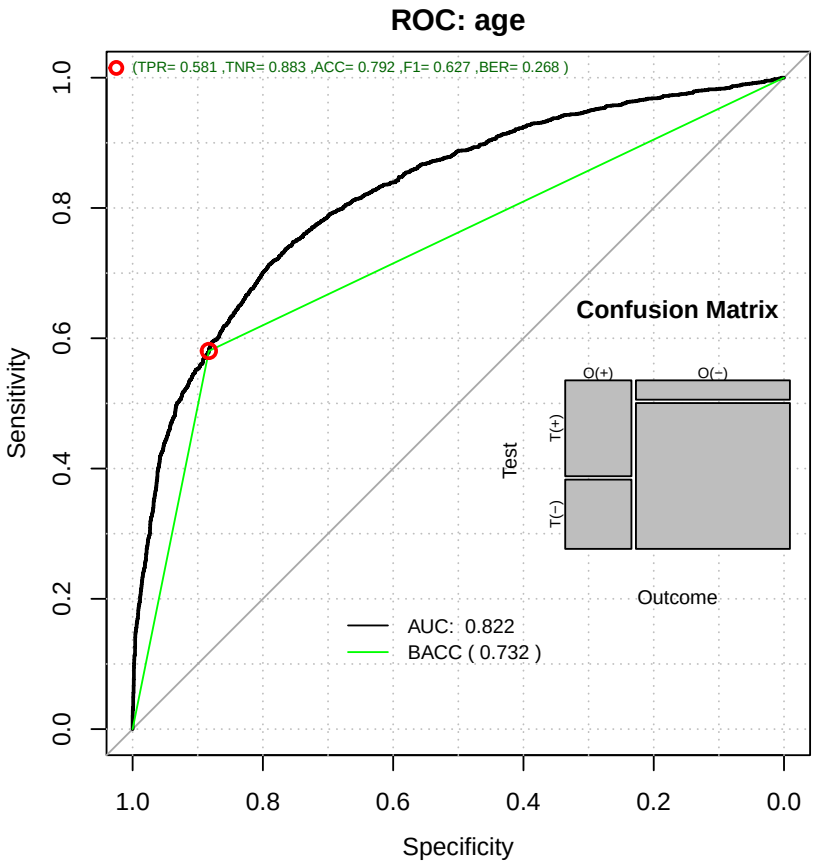
```
topv <- min(5,length(topvar))
topFive <- names(topvar)[1:topv]

topFeature <- RRPlot(cbind(dataFL$status,dataFL[,topFive[1]]),
                     title=topFive[1])
```

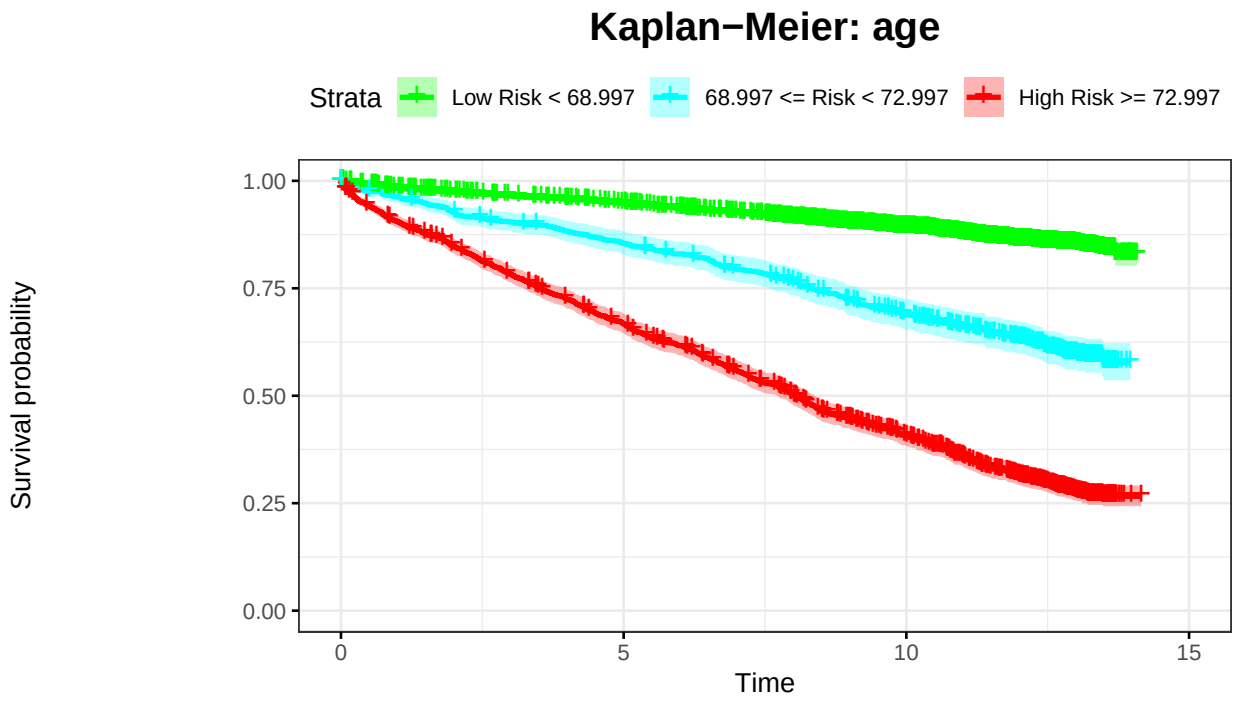

Relative Risk: age
Index(0.88)=72.997
RR(0.58)=4.013
PPV–>
1.0
0.8
0.6
0.4
0.2
1.0
0.8
0.6
0.4
0.2
0.0
No
Yes
Loess Fit
SPE
RR
SEN
1
2
3
4
5
6
0.0
0.2
0.4
0.6
0.8
1.0




```
  par(op)

## With Survival Analysis
RRanalysis <- list();
idx <- 1
for (topf in topFive)
{
  RRanalysis[[idx]] <- RRPlot(cbind(dataFL$status,dataFL[,topf]),
                  timetoEvent=dataFL$time,
                  atRate=c(0.90,0.80),
                  title=topf)
  idx <- idx + 1
  par(op)
}
```

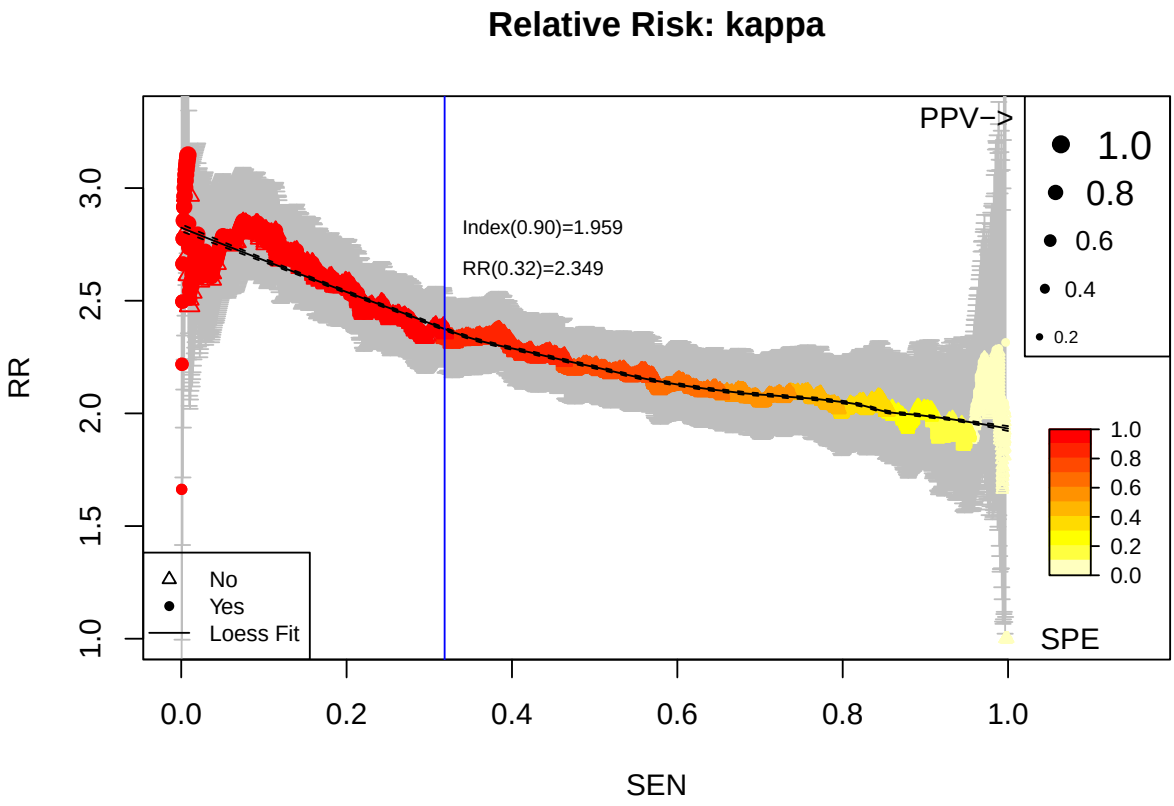

Relative Risk: age
Index(0.88)=72.997
RR(0.58)=4.013
PPV−>
1.0
0.8
0.6
0.4
0.2
1.0
0.8
0.6
0.4
0.2
0.0
SPE
No
Yes
Loess Fit
RR
SEN
1
2
3
4
5
6
0.0
0.2
0.4
0.6
0.8
1.0

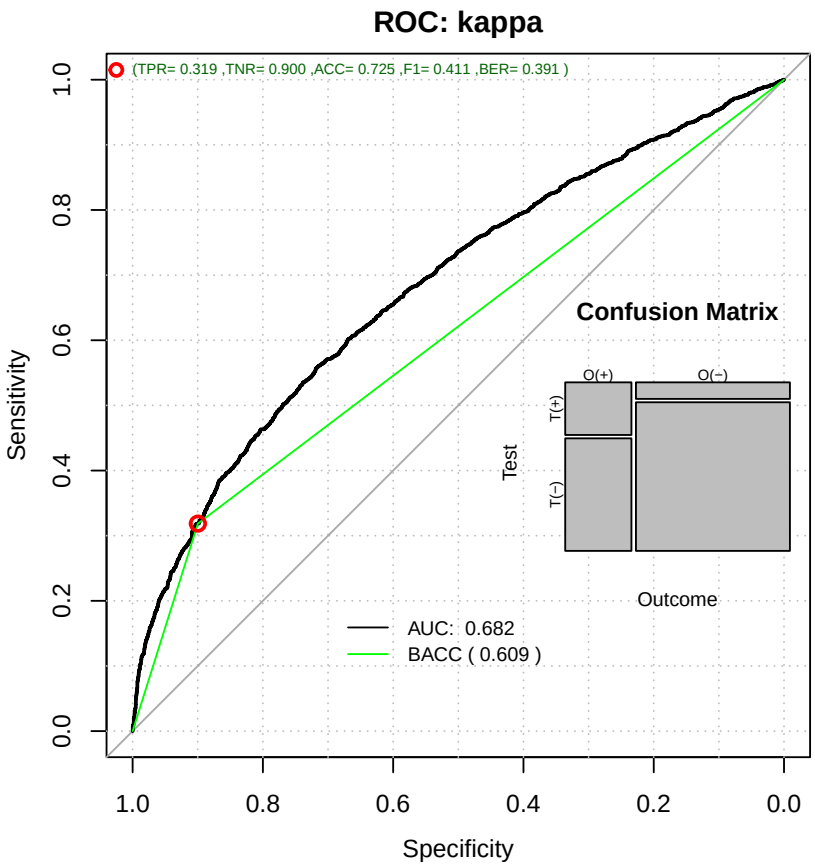

ROC: age
(TPR= 0.581 ,TNR= 0.883 ,ACC= 0.792 ,F1= 0.627 ,BER= 0.268)
Sensitivity
1.0
0.8
0.6
0.4
0.2
0.0
Specificity
1.0
0.8
0.6
0.4
0.2
0.0
Confusion Matrix
O(+)
O(−)
T(+)
T(−)
Test
Outcome
AUC: 0.822
BACC (0.732)

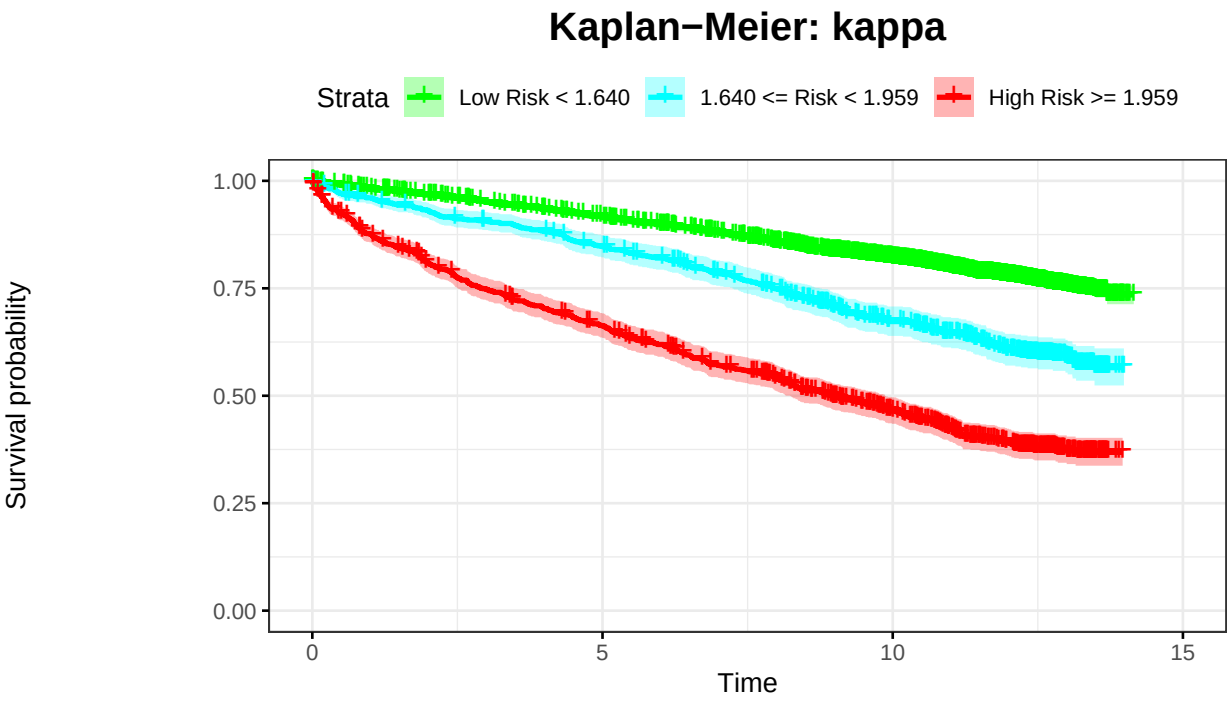


Number at risk

Low Risk < 68.997	4169	3839	3195	0
68.997 <= Risk < 72.997	682	575	430	0
High Risk >= 72.997	1673	1094	607	0

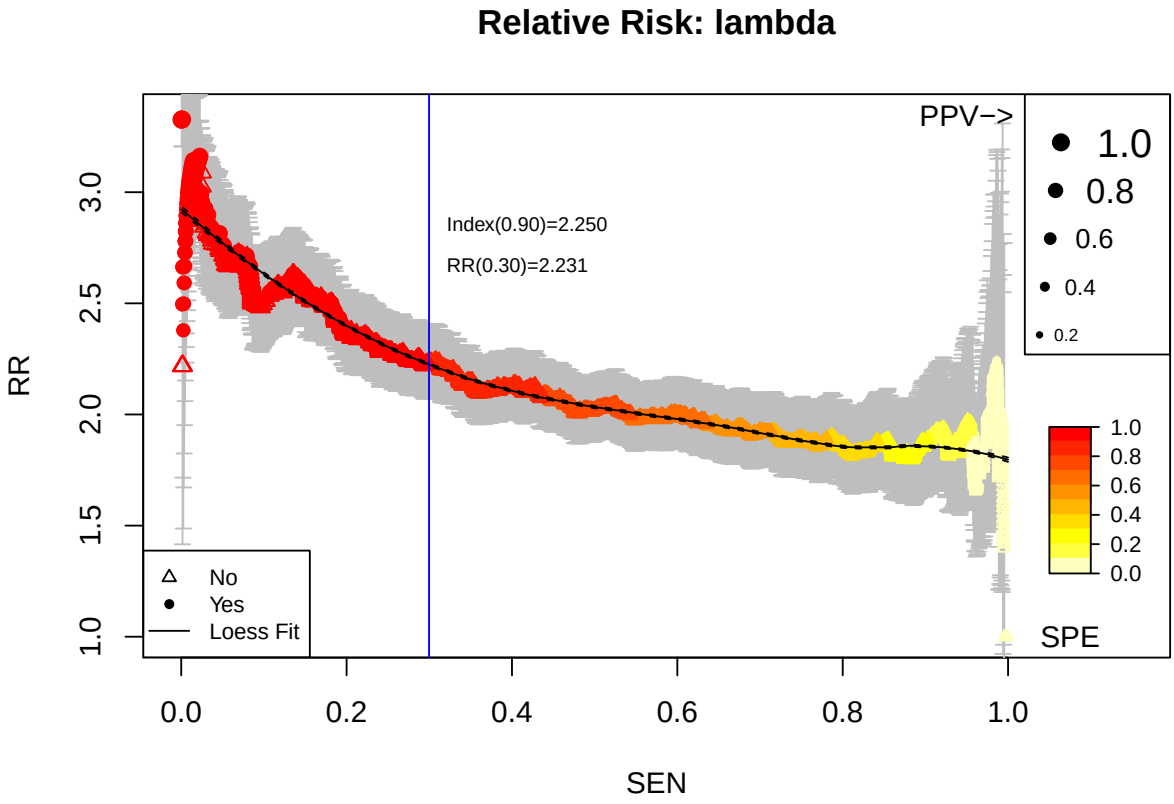

Relative Risk: kappa
PPV–>
1.0
0.8
0.6
0.4
0.2
Index(0.90)=1.959
RR(0.32)=2.349
1.0
0.8
0.6
0.4
0.2
0.0
SPE
No
Yes
Loess Fit
RR
1.0
1.5
2.0
2.5
3.0
0.0
0.2
0.4
0.6
0.8
1.0
SEN

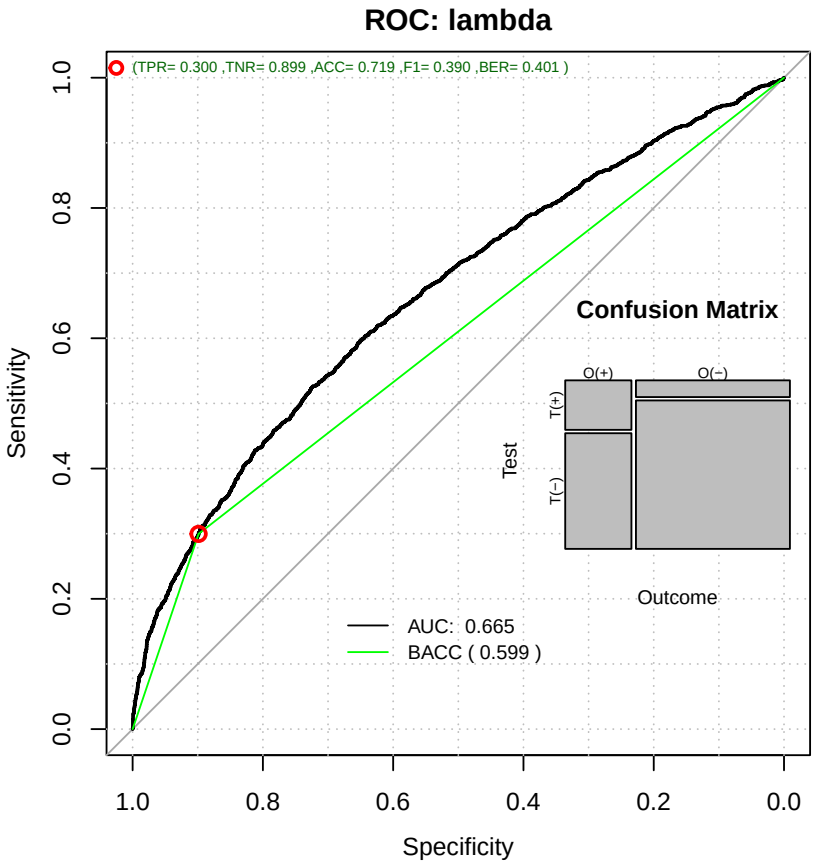

ROC: kappa
(TPR= 0.319 ,TNR= 0.900 ,ACC= 0.725 ,F1= 0.411 ,BER= 0.391)
Sensitivity
1.0
0.8
0.6
0.4
0.2
0.0
Specificity
1.0
0.8
0.6
0.4
0.2
0.0
Confusion Matrix
O(+)
O(−)
T(+)
T(−)
Test
Outcome
AUC: 0.682
BACC (0.609)

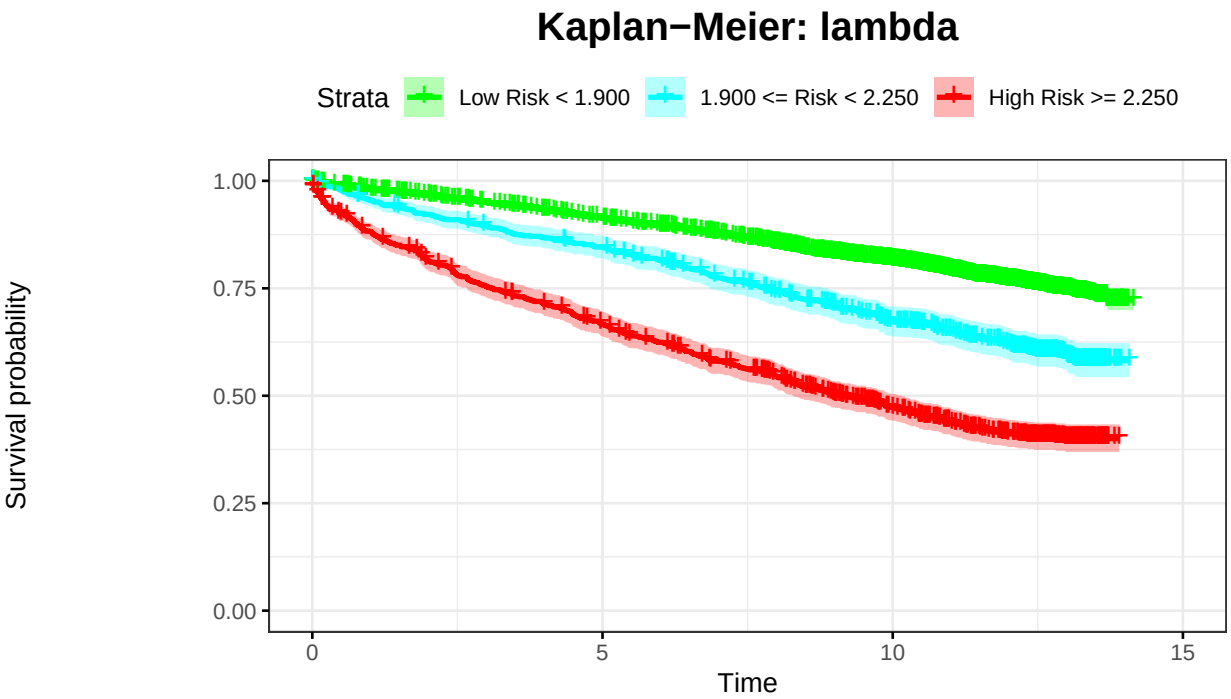


Number at risk

Low Risk < 1.640	4681	4184	3399	0
1.640 <= Risk < 1.959	761	632	428	0
High Risk >= 1.959	1082	692	405	0

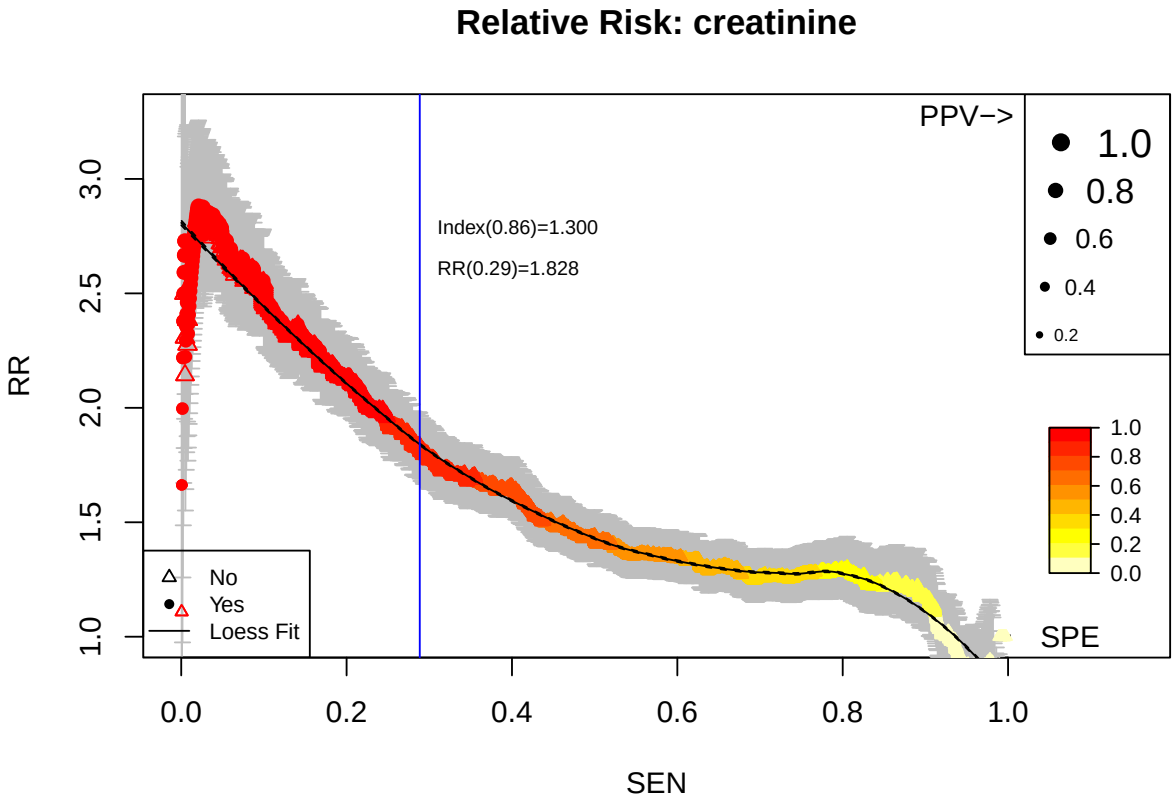

Relative Risk: lambda
PPV−>
1.0
0.8
0.6
0.4
0.2
Index(0.90)=2.250
RR(0.30)=2.231
1.0
0.8
0.6
0.4
0.2
0.0
SPE
No
Yes
Loess Fit
RR
1.0
1.5
2.0
2.5
3.0
0.0
0.2
0.4
0.6
0.8
1.0
SEN

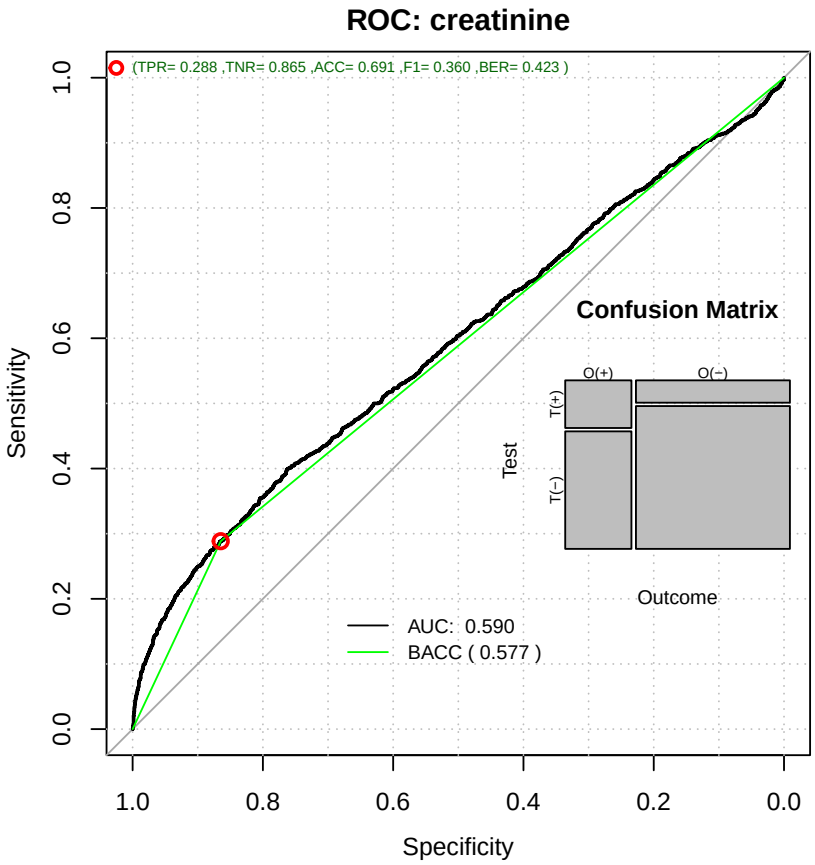

ROC: lambda
(TPR= 0.300 ,TNR= 0.899 ,ACC= 0.719 ,F1= 0.390 ,BER= 0.401)
Sensitivity
1.0
0.8
0.6
0.4
0.2
0.0
Specificity
1.0
0.8
0.6
0.4
0.2
0.0
Confusion Matrix
O(+)
O(−)
T(+)
T(−)
Test
Outcome
AUC: 0.665
BACC (0.599)

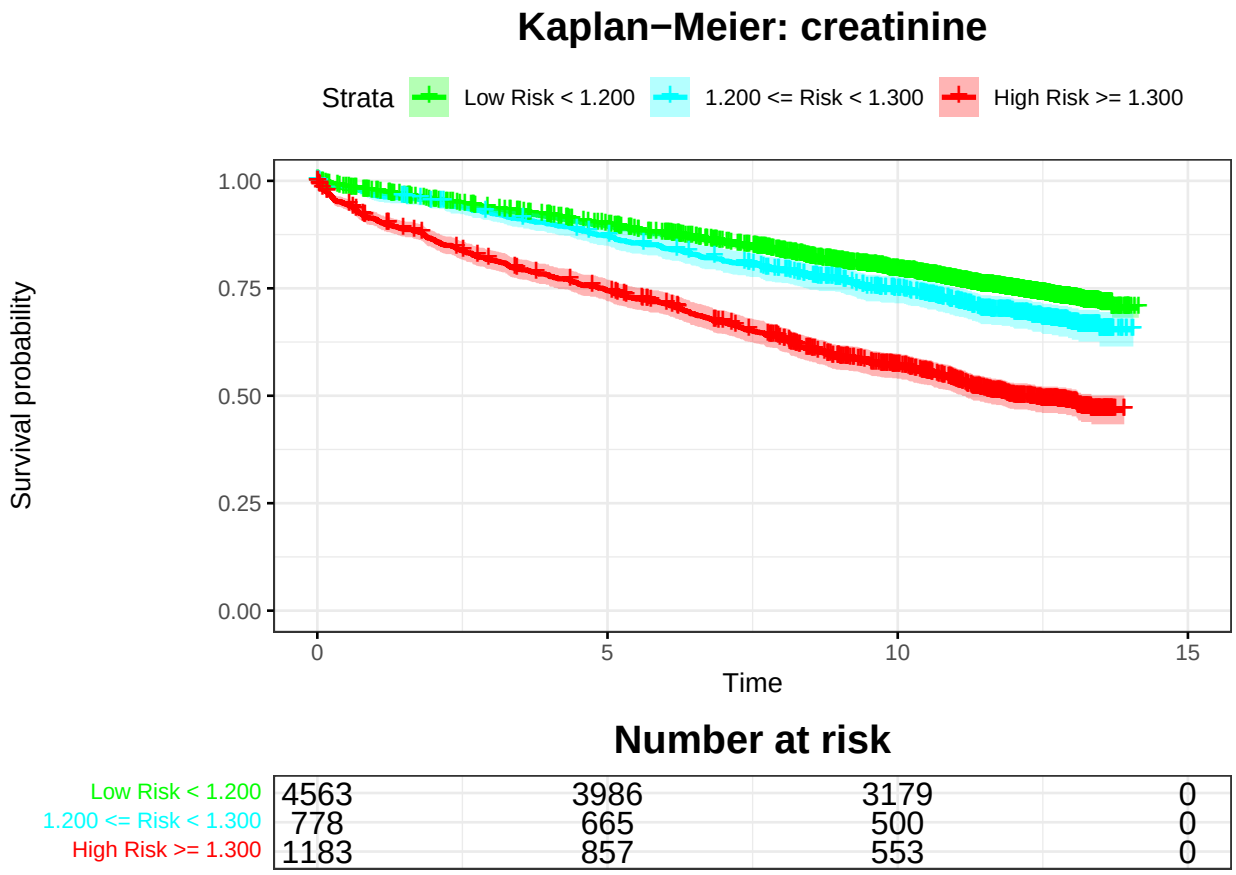


Number at risk

Low Risk < 1.900	4741	4219	3404	0
1.900 <= Risk < 2.250	733	611	424	0
High Risk >= 2.250	1050	678	404	0


Relative Risk: creatinine
PPV–>
1.0
0.8
0.6
0.4
0.2
Index(0.86)=1.300
RR(0.29)=1.828
1.0
0.8
0.6
0.4
0.2
0.0
SPE
No
Yes
Loess Fit
RR
1.0
1.5
2.0
2.5
3.0
0.0
0.2
0.4
0.6
0.8
1.0
SEN


ROC: creatinine
(TPR= 0.288 ,TNR= 0.865 ,ACC= 0.691 ,F1= 0.360 ,BER= 0.423)
Sensitivity
1.0
0.8
0.6
0.4
0.2
0.0
Specificity
1.0
0.8
0.6
0.4
0.2
0.0
Confusion Matrix
O(+)
O(−)
T(+)
T(−)
Test
Outcome
AUC: 0.590
BACC (0.577)




```
names(RRanalysis) <- topFive
```

1.2 Reporting the Metrics

```
pander::pander(t(RRanalysis[[1]]$keyPoints),caption="Threshold values")
```

Table 4: Threshold values

	@:0.9	@:0.8	@MAX_BACC	@MAX_RR	@SPE100
Thr	73.000	69.000	69.000	54.000	50.00000
RR	4.013	4.399	4.405	5.672	1.00000
RR_LCI	3.740	4.045	4.051	4.529	0.00000
RR_UCI	4.305	4.783	4.790	7.104	0.00000
SEN	0.581	0.713	0.713	0.963	1.00000
SPE	0.883	0.790	0.791	0.241	0.00395
BACC	0.732	0.752	0.752	0.602	0.50197

```
ROCAUC <- NULL
CstatCI <- NULL
RRatios <- NULL
LogRangp <- NULL
Sensitivity <- NULL
Specificity <- NULL
```

```
for (topf in topFive)
{
  CstatCI <- rbind(CstatCI,RRanalysis[[topf]]$c.index$cstatCI)
  RRatios <- rbind(RRatios,RRanalysis[[topf]]$RR_atP)
  LogRangp <- rbind(LogRangp,RRanalysis[[topf]]$surdif$pvalue)
  Sensitivity <- rbind(Sensitivity,RRanalysis[[topf]]$ROCAnalysis$sensitivity)
  Specificity <- rbind(Specificity,RRanalysis[[topf]]$ROCAnalysis$specificity)
  ROCAUC <- rbind(ROCAUC,RRanalysis[[topf]]$ROCAnalysis$aucs)
}
rownames(CstatCI) <- topFive
rownames(LogRangp) <- topFive
rownames(Sensitivity) <- topFive
rownames(Specificity) <- topFive
rownames(ROCAUC) <- topFive

pander::pander(ROCAUC)
```

	est	lower	upper
age	0.822	0.811	0.834
kappa	0.682	0.668	0.697
lambda	0.665	0.650	0.680
creatinine	0.590	0.575	0.606

```
pander::pander(CstatCI)
```

	mean.C Index	median	lower	upper
age	0.775	0.775	0.764	0.785
kappa	0.671	0.671	0.659	0.684
lambda	0.657	0.657	0.644	0.670
creatinine	0.587	0.587	0.573	0.601

```
pander::pander(LogRangp)
```

age	0.00e+00
kappa	4.90e-175
lambda	4.41e-145
creatinine	2.67e-67

```
pander::pander(Sensitivity)
```

	est	lower	upper
age	0.581	0.558	0.602
kappa	0.319	0.298	0.340
lambda	0.300	0.279	0.321
creatinine	0.288	0.269	0.309

```
pander::pander(Specificity)
```

	est	lower	upper
age	0.883	0.873	0.892
kappa	0.900	0.891	0.908
lambda	0.899	0.890	0.907
creatinine	0.865	0.854	0.875

```
meanMatrix <- cbind(ROCAUC[,1],CstatCI[,1],Sensitivity[,1],Specificity[,1])
colnames(meanMatrix) <- c("ROCAUC","C-Stat","Sen","Spe")
pander::pander(meanMatrix)
```

	ROCAUC	C-Stat	Sen	Spe
age	0.822	0.775	0.581	0.883
kappa	0.682	0.671	0.319	0.900
lambda	0.665	0.657	0.300	0.899
creatinine	0.590	0.587	0.288	0.865

1.2.1 Train Test Set

```
trainsamples <- sample(nrow(dataFL),0.5*nrow(dataFL))
dataFLTrain <- dataFL[trainsamples,]
dataFLTest <- dataFL[-trainsamples,]


pander::pander(table(dataFLTrain$status))
```

0	1
2288	974

```
pander::pander(table(dataFLTest$status))
```

0	1
2274	988

1.3 Cox Modeling

```
ml <- BSWiMS.model(Surv(time,status)~.,data=dataFLTrain,loops=0)
sm <- summary(ml)
pander::pander(sm$coefficients)
```

Table 13: Table continues below

	Estimate	lower	mean	upper	u.Accuracy	r.Accuracy
age	0.1005	0.0759	0.0814	0.0883	0.738	0.615
lambda	0.1962	0.1326	0.1699	0.2793	0.654	0.725

	Estimate	lower	mean	upper	u.Accuracy	r.Accuracy
sexM	0.2053	0.0461	0.1789	0.2833	0.487	0.731
flc.grp	0.0572	0.0122	0.0450	0.0707	0.596	0.729
mgus	-0.6036	-14.2252	-0.5733	0.1287	0.310	0.727

Table 14: Table continues below

	full.Accuracy	u.AUC	r.AUC	full.AUC	IDI	NRI
age	0.727	0.755	0.626	0.751	0.200233	0.9091
lambda	0.727	0.620	0.750	0.751	0.003799	-0.0174
sexM	0.727	0.515	0.755	0.751	0.002258	-0.0612
flc.grp	0.727	0.619	0.752	0.751	0.003000	0.2497
mgus	0.727	0.507	0.752	0.751	0.000668	-0.0639

	z.IDI	z.NRI	Delta.AUC	Frequency
age	28.02	27.60	-0.12576	1
lambda	3.58	-0.46	-0.00168	1
sexM	3.11	-1.62	0.00365	1
flc.grp	3.04	6.64	0.00035	1
mgus	1.96	-4.09	0.00095	1

1.4 Cox Model Performance

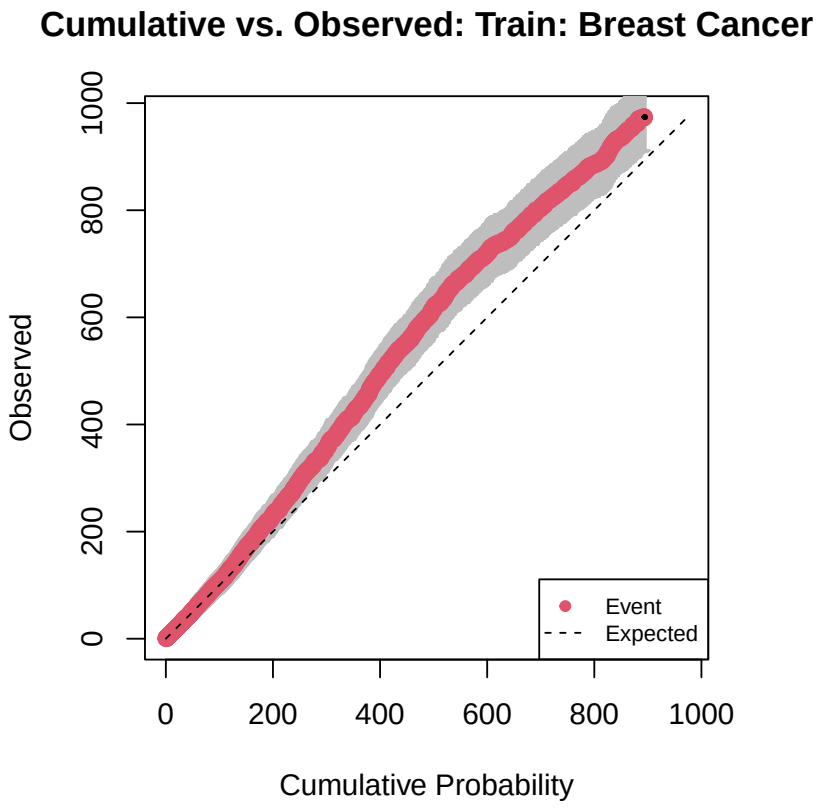
1.4.1 The evaluation of the raw Cox model with RRPlot()

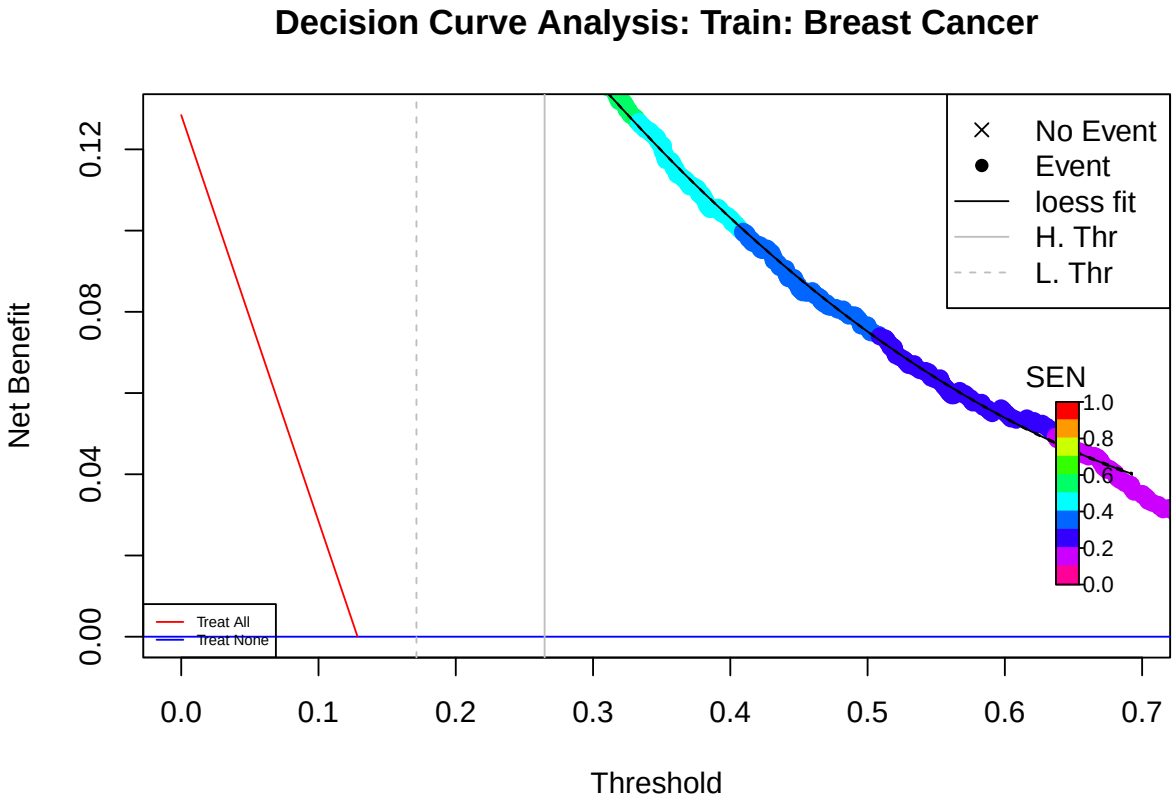
```
timeinterval <- 5

h0 <- sum(dataFLTrain$status & dataFLTrain$time <= timeinterval)
h0 <- h0/sum((dataFLTrain$time > timeinterval) | (dataFLTrain$status==1))

pander::pander(t(c(h0=h0,timeinterval=timeinterval)),caption="Initial Parameters")
```

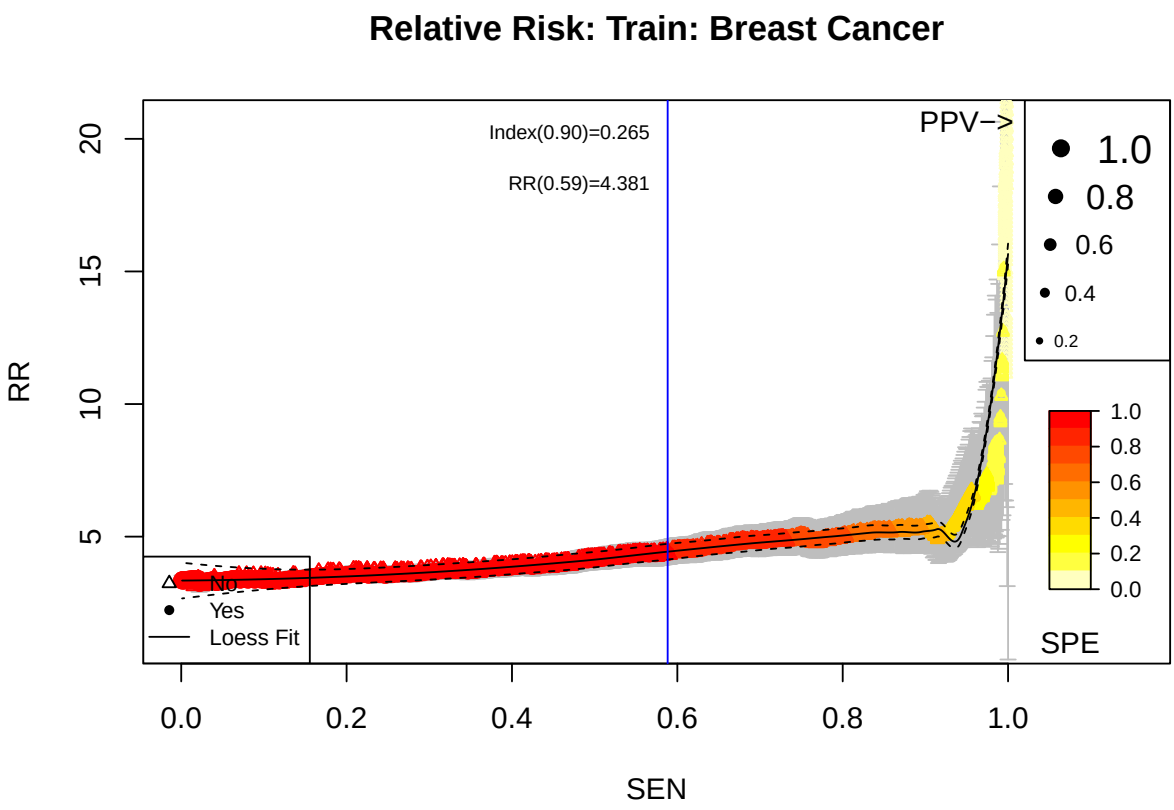
Table 16: Initial Parameters

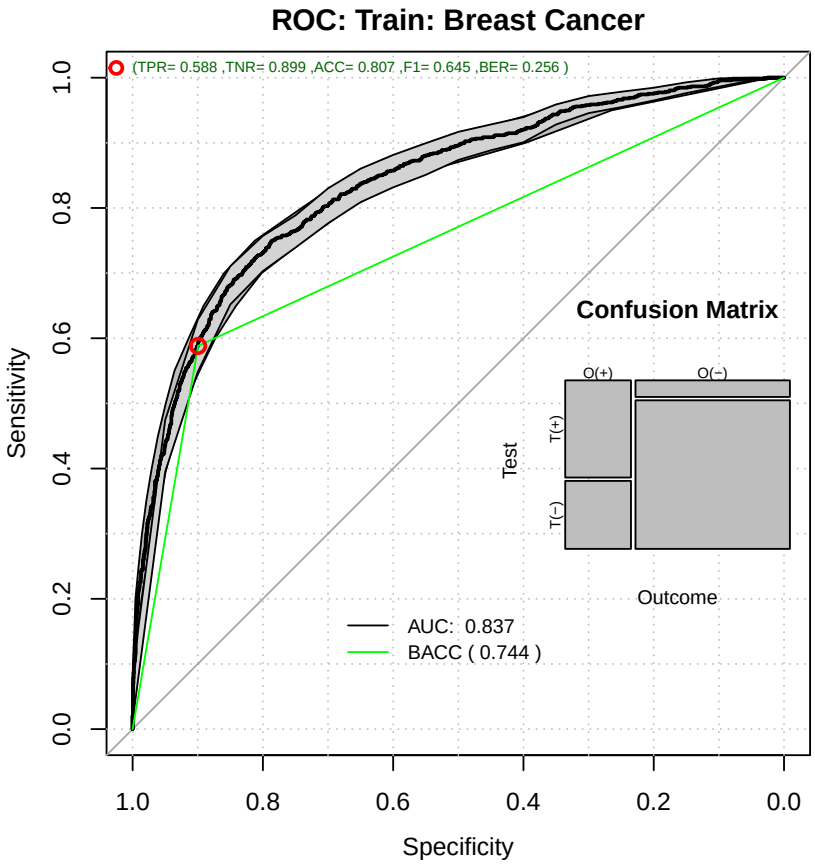
h0	timeinterval
0.131	5

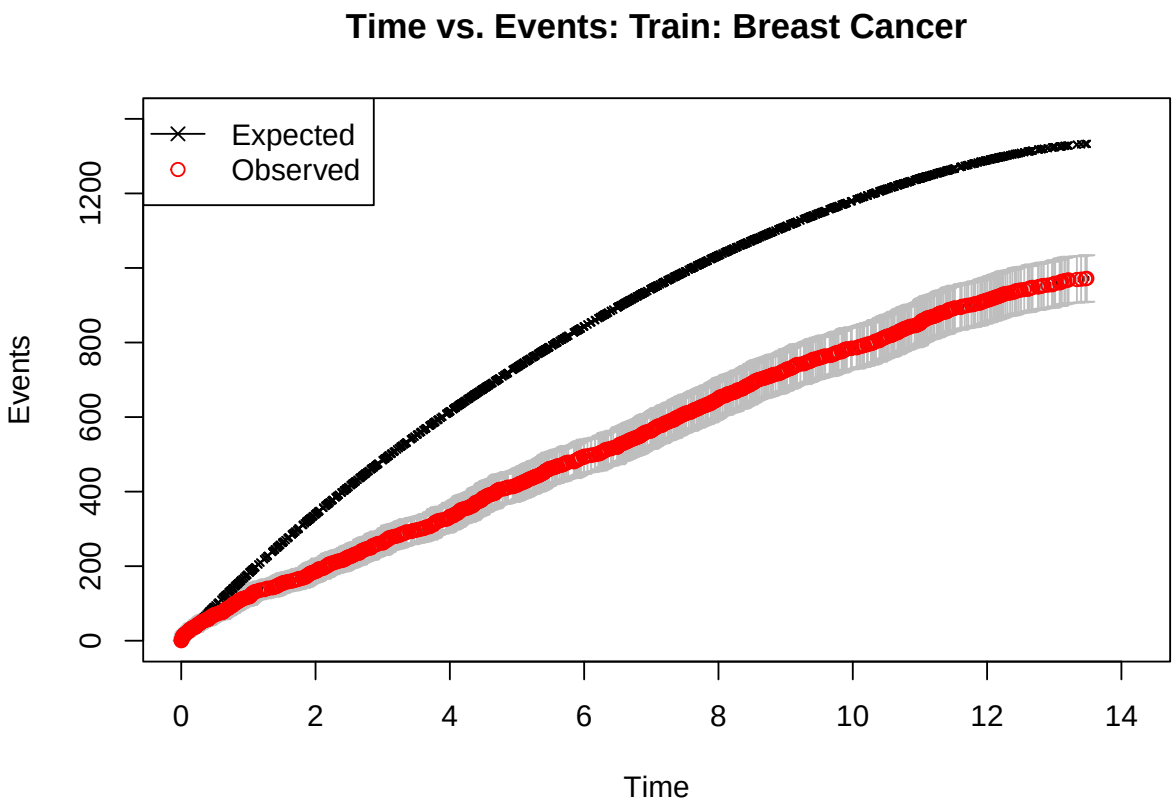
```
index <- predict(ml,dataFLTrain)
rdata <- cbind(dataFLTrain$status,ppoisGzero(index,h0))

rrAnalysisTrain <- RRPlot(rdata,atRate=c(0.90,0.80),
                      timetoEvent=dataFLTrain$time,
                      title="Train: Breast Cancer",
                      ysurvlim=c(0.00,1.0),
                      riskTimeInterval=timeinterval)
```

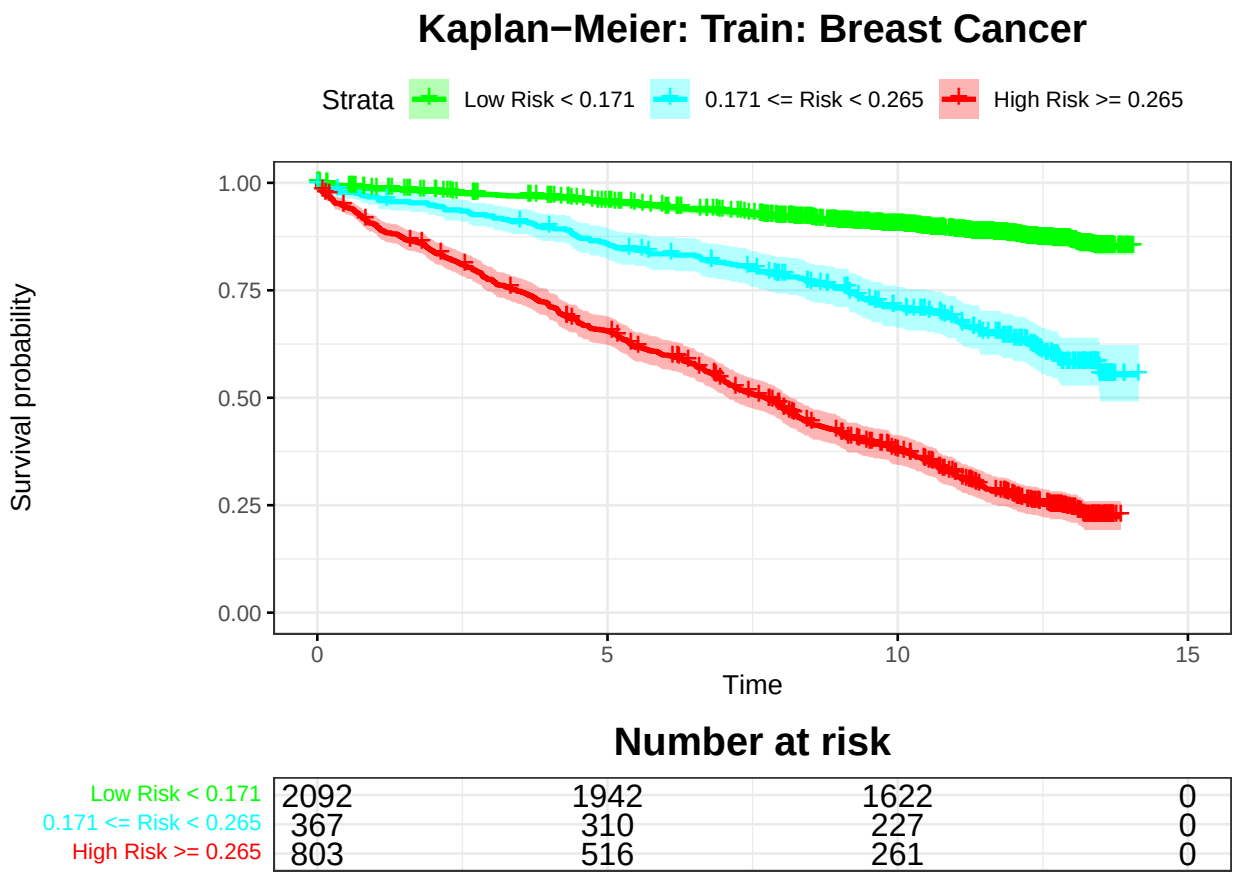

Cumulative vs. Observed: Train: Breast Cancer
Observed
0
200
400
600
800
1000
Event
Expected
0
200
400
600
800
1000
Cumulative Probability


Decision Curve Analysis: Train: Breast Cancer
Net Benefit
0.00
0.04
0.08
0.12
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Threshold
No Event
Event
loess fit
H. Thr
L. Thr
SEN
1.0
0.8
0.6
0.4
0.2
0.0
Treat All
Treat None


Relative Risk: Train: Breast Cancer
Index(0.90)=0.265
RR(0.59)=4.381
PPV–>
1.0
0.8
0.6
0.4
0.2
SPE
No
Yes
Loess Fit
RR
SEN
5
10
15
20
0.0
0.2
0.4
0.6
0.8
1.0


ROC: Train: Breast Cancer
(TPR= 0.588 ,TNR= 0.899 ,ACC= 0.807 ,F1= 0.645 ,BER= 0.256)
Sensitivity
Specificity
1.0
0.8
0.6
0.4
0.2
0.0
Confusion Matrix
O(+)
O(−)
T(+)
T(−)
Test
Outcome
AUC: 0.837
BACC (0.744)


Time vs. Events: Train: Breast Cancer
Expected
Observed
Events
0
200
400
600
800
1200
0
2
4
6
8
10
12
14
Time



1.4.2 Time to event

```
toinclude <- rdata[,1]==1
obstiemToEvent <- dataFLTrain[,"time"]
summary(obstiemToEvent)
```

Min. 1st Qu. Median Mean 3rd Qu. Max. 0.000 7.995 11.760 10.028 13.071 14.153

```
tmin<-min(obstiemToEvent)
if (tmin < 0.01) tmin <- 0.01
tmax<-max(obstiemToEvent)
sum(toinclude)
```

[1] 974

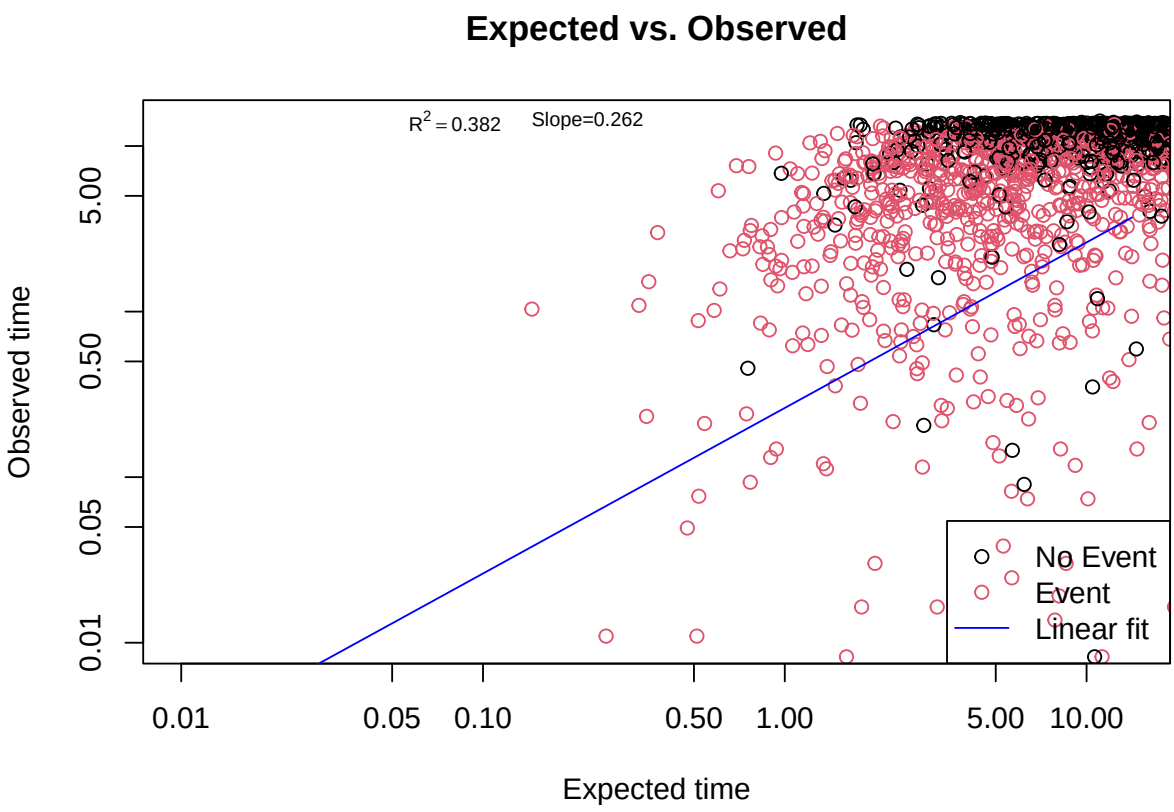
```
timetoEvent <- meanTimeToEvent(rdata[,2],timeinterval)
timetoEvent[is.infinite(timetoEvent)] <- 3*tmax
timetoEvent[timetoEvent > 3*tmax] <- 3*tmax
timetoEvent[timetoEvent < tmin] <- tmin

lmfit <- lm(obstiemToEvent[toinclude]~0+timetoEvent[toinclude])
sm <- summary(lmfit)
pander::pander(sm)
```

	Estimate	Std. Error	t value	Pr(>|t|)
timetoEvent[toinclude]	0.262	0.0107	24.5	7.34e-104

Table 18: Fitting linear model: obstiemToEvent[toinclude] ~ 0 + timetoEvent[toinclude]

Observations	Residual Std. Error	R^2	Adjusted R^2
974	5.58	0.382	0.382

```
plot(timetoEvent,obstiemToEvent,
     col=1+rdata[,1],
     xlab="Expected time",
     ylab="Observed time",
     main="Expected vs. Observed",
     xlim=c(tmin,tmax),
     ylim=c(tmin,tmax),
     log="xy")
lines(x=c(tmin,tmax),y=lmfit$coefficients*c(tmin,tmax),lty=1,col="blue")
txt <- bquote(paste(R^2 == .(round(sm$r.squared,3))))
text(tmin+0.005*(tmax-tmin),tmax,txt,cex=0.7)
text(tmin+0.015*(tmax-tmin),tmax,sprintf("Slope=%4.3f",sm$coefficients[1]),cex=0.7)
legend("bottomright",legend=c("No Event","Event","Linear fit"),
             pch=c(1,1,-1),
             col=c(1,2,"blue"),
             lty=c(-1,-1,1)
             )
```




```
MADerror2 <- mean(abs(timetoEvent[toinclude]-obstiemToEvent[toinclude]))
pander::pander(MADerror2)
```

8.48

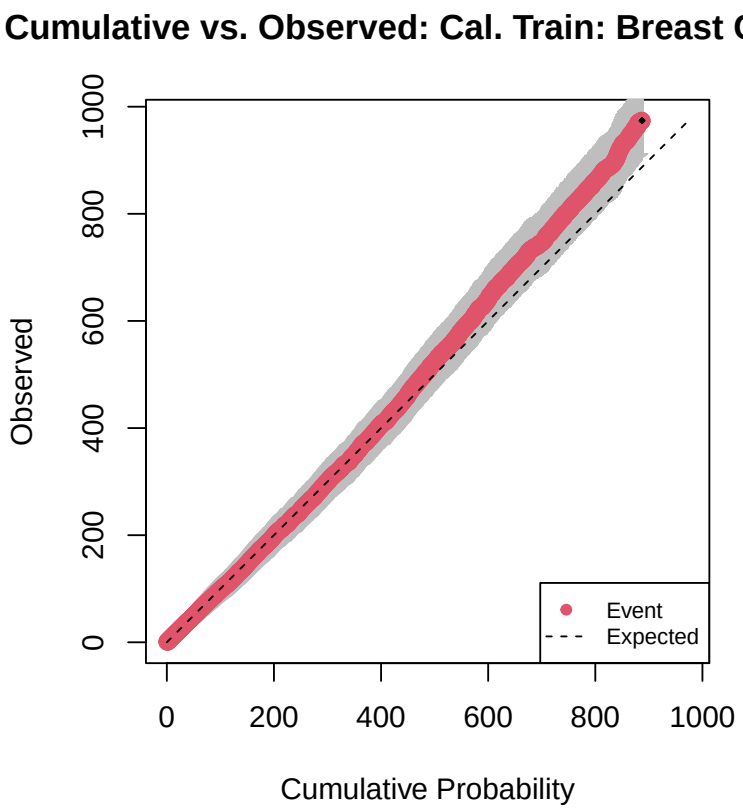
1.4.3 Cox Calibration

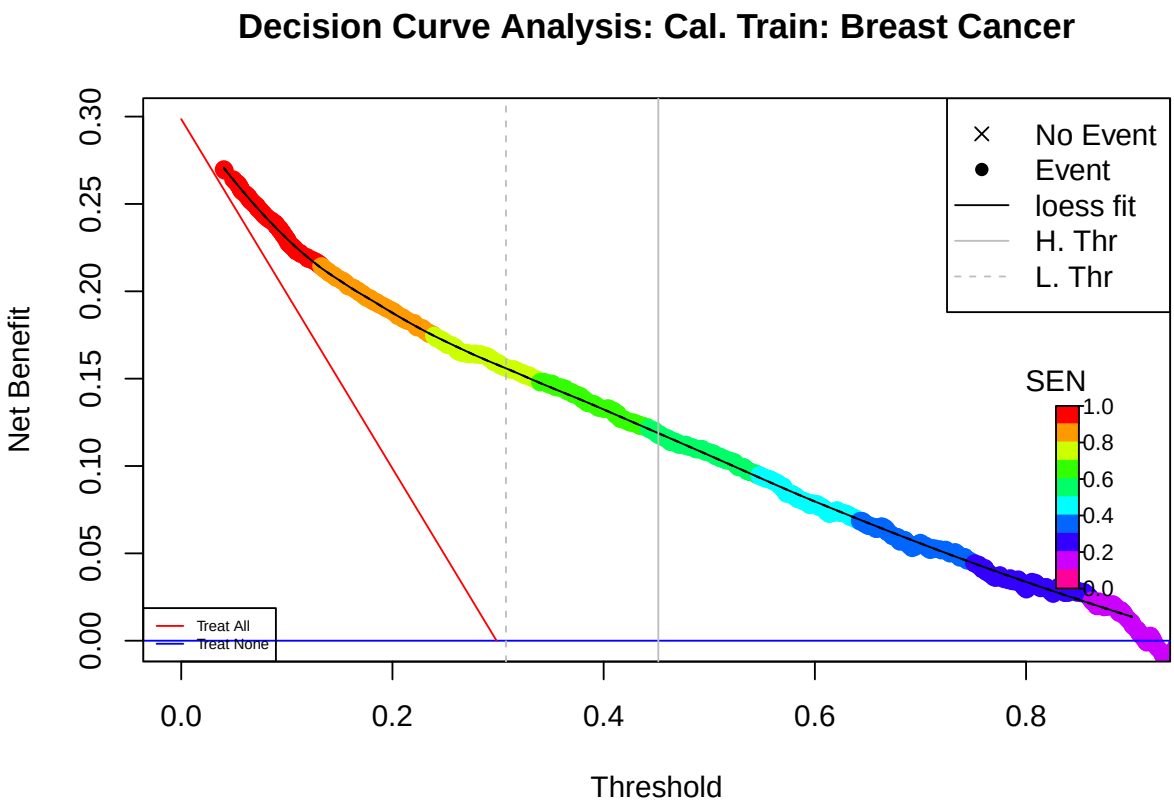
```
op <- par(no.readonly = TRUE)


calprob <- CoxRiskCalibration(ml,dataFLTrain,"status","time")


pander::pander(c(h0=calprob$h0,
                 Gain=calprob$hazardGain,
                 DeltaTime=calprob$timeInterval),
               caption="Cox Calibration Parameters")
```

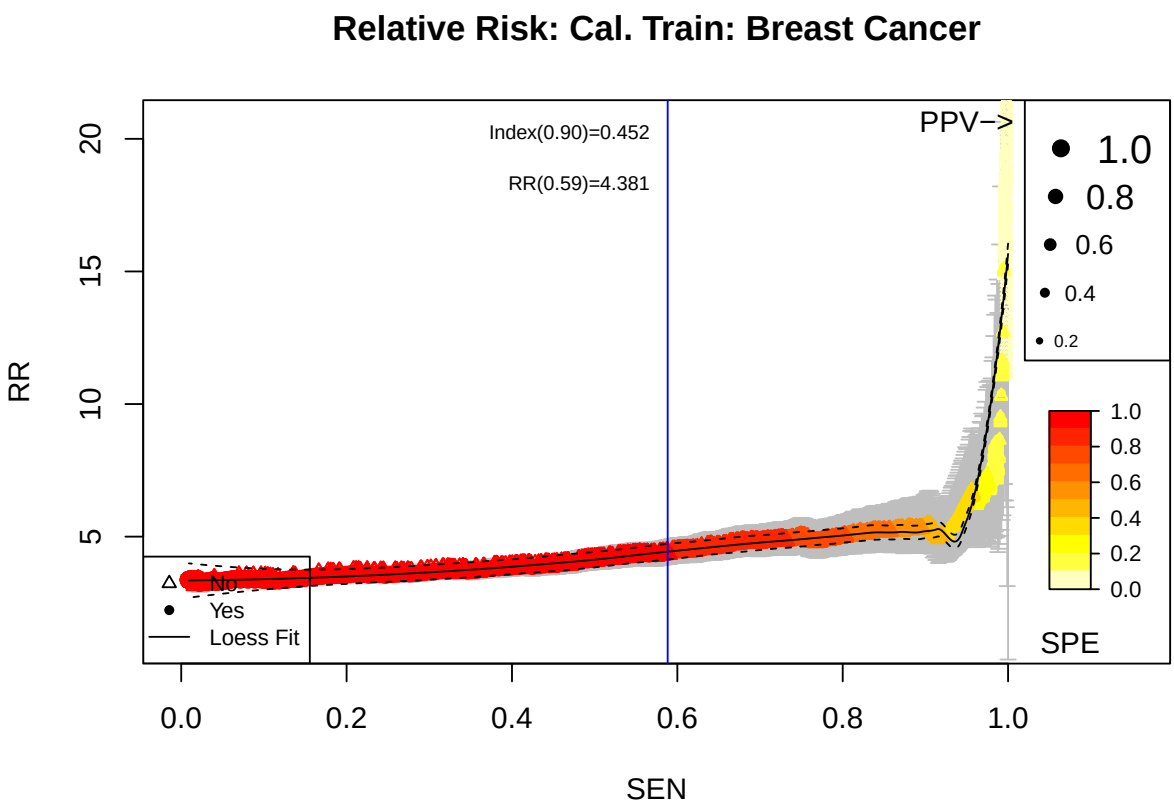
h0	Gain	DeltaTime
0.257	0.688	15.8

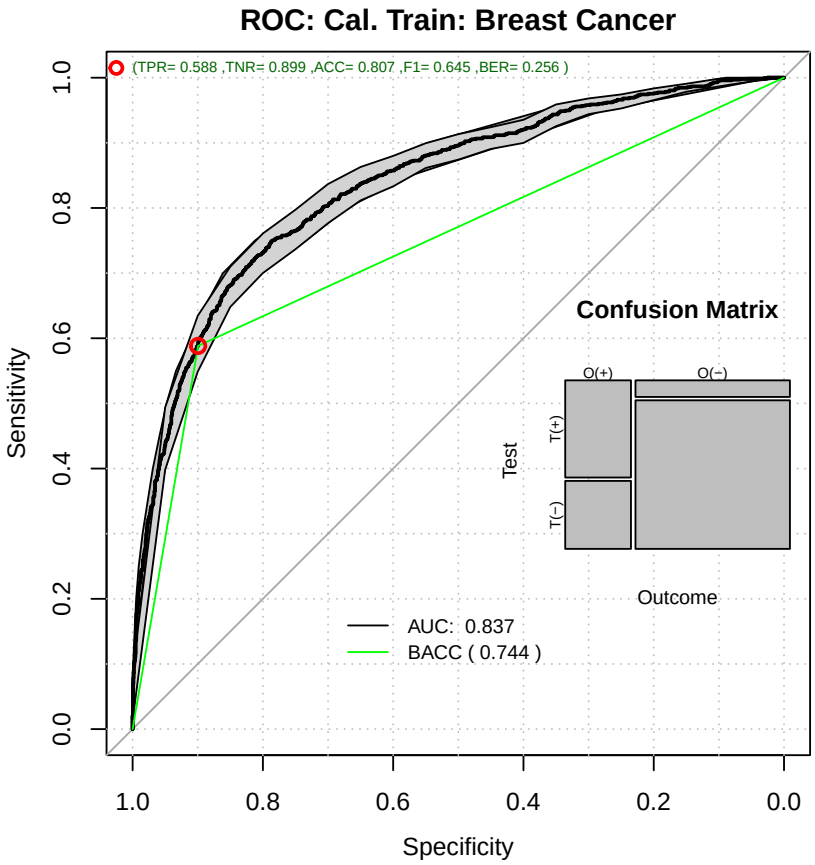
1.4.4 The RRplot() of the calibrated model

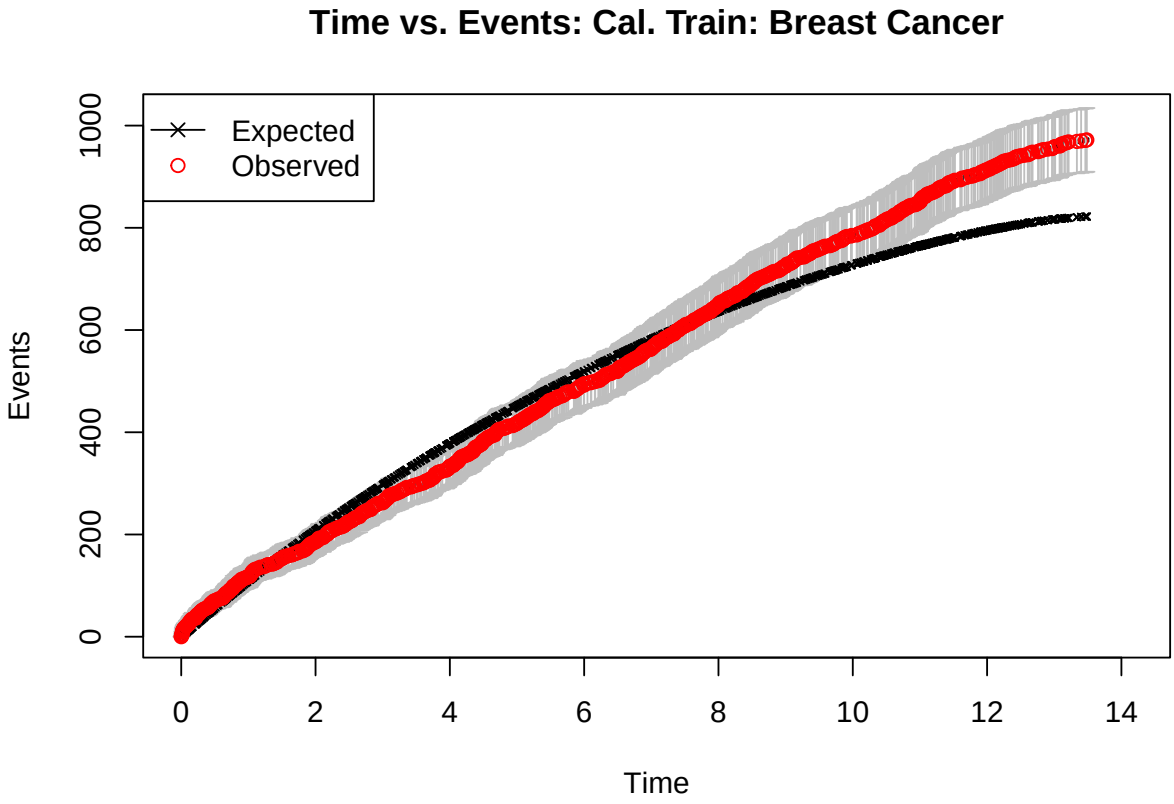
```
h0 <- calprob$h0
timeinterval <- calprob$timeInterval;

rdata <- cbind(dataFLTrain$status,calprob$prob)


rrAnalysisTrain <- RRPlot(rdata,atRate=c(0.90,0.80),
                      timetoEvent=dataFLTrain$time,
                      title="Cal. Train: Breast Cancer",
                      ysurvlim=c(0.00,1.0),
                      riskTimeInterval=timeinterval)
```

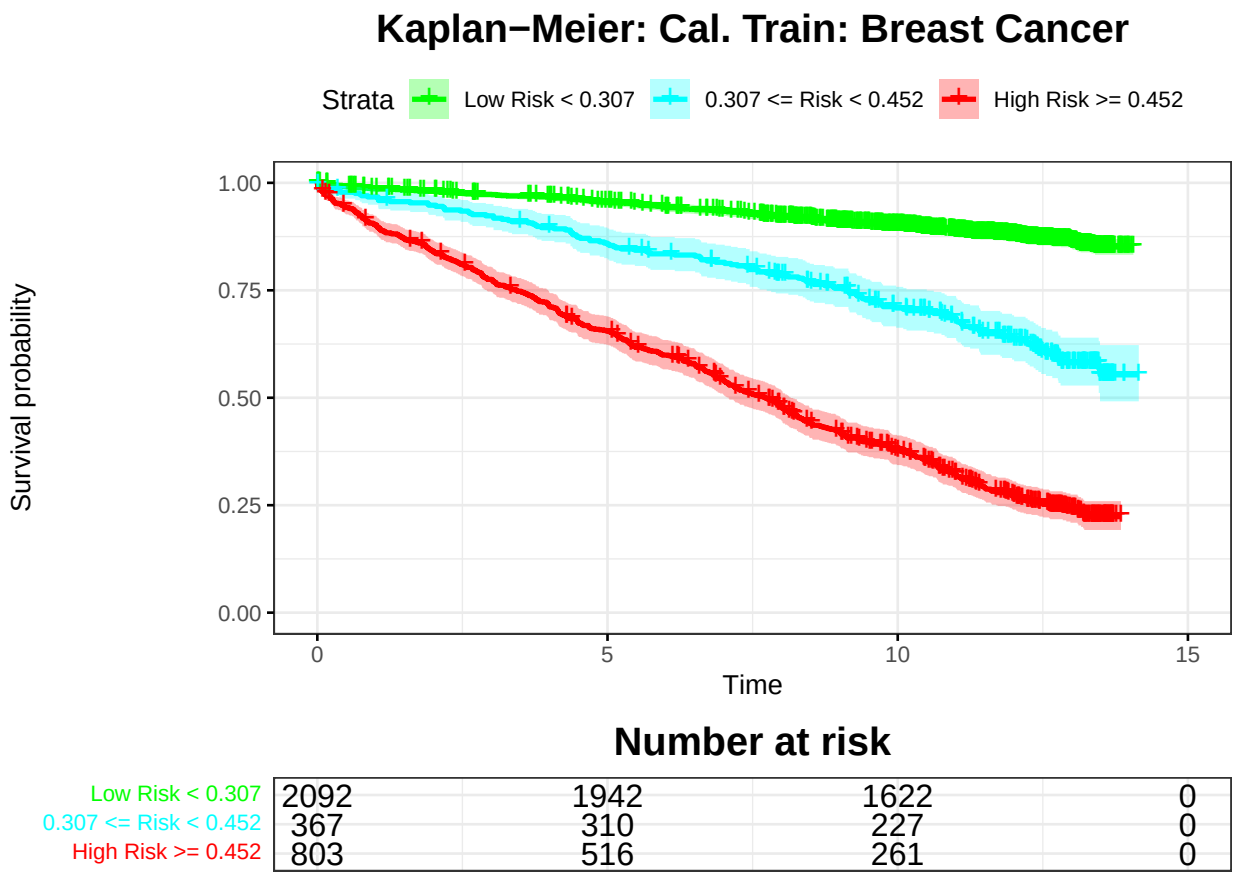



Decision Curve Analysis: Cal. Train: Breast Cancer
Net Benefit
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.0
0.2
0.4
0.6
0.8
Threshold
No Event
Event
loess fit
H. Thr
L. Thr
SEN
1.0
0.8
0.6
0.4
0.2
0.0
Treat All
Treat None


Relative Risk: Cal. Train: Breast Cancer
Index(0.90)=0.452
RR(0.59)=4.381
PPV–>
1.0
0.8
0.6
0.4
0.2
1.0
0.8
0.6
0.4
0.2
0.0
SPE
No
Yes
Loess Fit
RR
5
10
15
20
0.0
0.2
0.4
0.6
0.8
1.0
SEN


ROC: Cal. Train: Breast Cancer
(TPR= 0.588 ,TNR= 0.899 ,ACC= 0.807 ,F1= 0.645 ,BER= 0.256)
Sensitivity
Specificity
1.0
0.8
0.6
0.4
0.2
0.0
Confusion Matrix
O(+)
O(−)
T(+)
T(−)
Test
Outcome
AUC: 0.837
BACC (0.744)


Time vs. Events: Cal. Train: Breast Cancer
Expected
Observed
Events
0
200
400
600
800
1000
0
2
4
6
8
10
12
14
Time



1.4.5 Time to event after calibration

```
timetoEvent <- meanTimeToEvent(rdata[,2],timeinterval)
tmax<-max(c(obstiemToEvent))
lmfit <- lm(obstiemToEvent[toinclude]~0+timetoEvent[toinclude])
sm <- summary(lmfit)
pander::pander(sm)
```

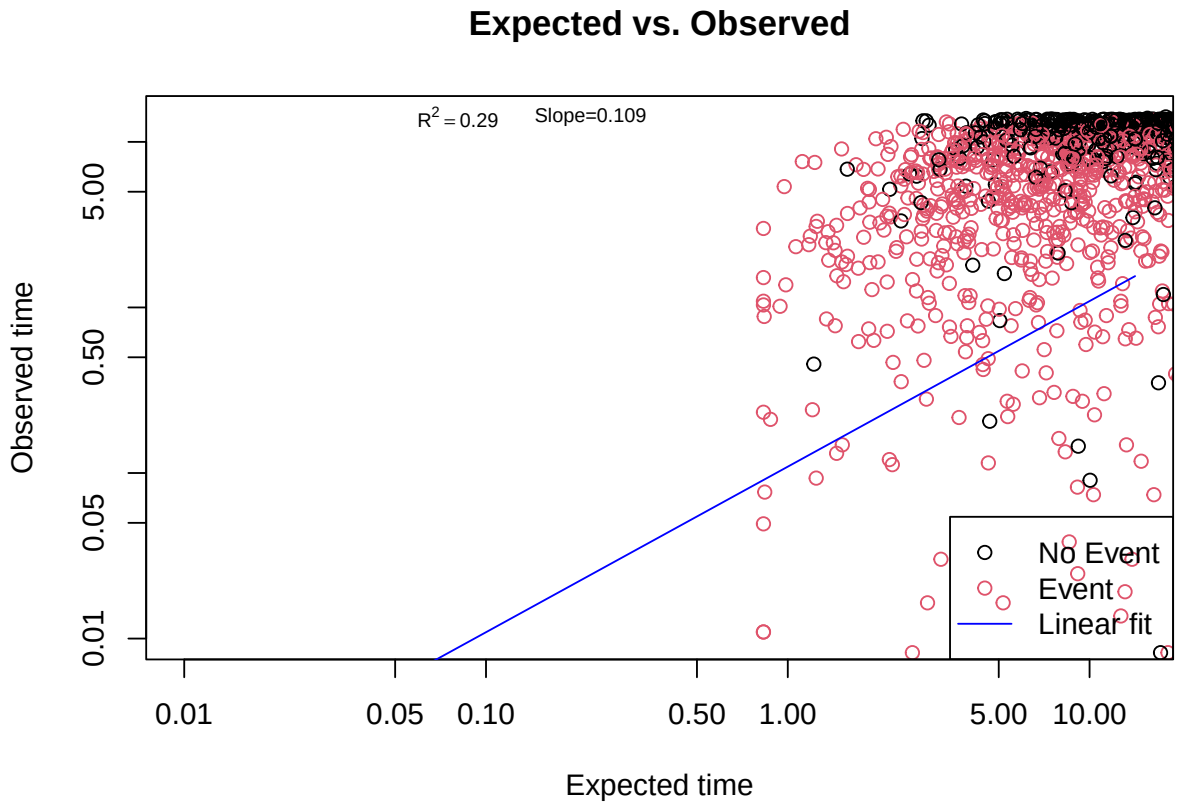
	Estimate	Std. Error	t value	Pr(>|t|)
timetoEvent[toinclude]	0.109	0.00548	19.9	2.17e-74

Table 21: Fitting linear model: obstiemToEvent[toinclude] ~ 0 + timetoEvent[toinclude]

Observations	Residual Std. Error	R^2	Adjusted R^2
974	5.98	0.29	0.289

```
plot(timetoEvent,obstiemToEvent,
     col=1+rdata[,1],
     xlab="Expected time",
     ylab="Observed time",
```

```
     main="Expected vs. Observed",
     xlim=c(tmin,tmax),
     ylim=c(tmin,tmax),
     log="xy")
lines(x=c(tmin,tmax),y=lmfit$coefficients*c(tmin,tmax),lty=1,col="blue")
txt <- bquote(paste(R^2 == .(round(sm$r.squared,3))))
text(tmin+0.005*(tmax-tmin),tmax,txt,cex=0.7)
text(tmin+0.015*(tmax-tmin),tmax,sprintf("Slope=%4.3f",sm$coefficients[1]),cex=0.7)
legend("bottomright",legend=c("No Event","Event","Linear fit"),
             pch=c(1,1,-1),
             col=c(1,2,"blue"),
             lty=c(-1,-1,1)
             )
```




```
MADerror2 <- c(MADerror2,mean(abs(timetoEvent[toinclude]-obstiemToEvent[toinclude])))
pander::pander(MADerror2)
```

8.48 and *16.64*