

Colon Cancer

Jose Tamez

2023-04-26

```
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")
source("~/GitHub/FRESA.CAD/R/PoissonEventRiskCalibration.R")
```

```
data(cancer)
colon <- subset(colon, etype==1)
colon$etype <- NULL
rownames(colon) <- colon$id
colon$id <- NULL
colon <- colon[complete.cases(colon),]
time <- colon$time
status <- colon$status
data <- colon
data$time <- NULL
data$study <- NULL
table(data$status)
```

0 1 442 446

```
#dataColon <- as.data.frame(model.matrix(status~*.,data))
dataColon <- as.data.frame(model.matrix(status~*age,data))
dataColon$` (Intercept)` <- NULL
dataColon$time <- time/365
dataColon$status <- status
colnames(dataColon) <-str_replace_all(colnames(dataColon),":","_")
colnames(dataColon) <-str_replace_all(colnames(dataColon),"\\.","_")
colnames(dataColon) <-str_replace_all(colnames(dataColon),"\\+","_")
data <- NULL

trainsamples <- sample(nrow(dataColon),0.7*nrow(dataColon))
dataColonTrain <- dataColon[trainsamples,]
dataColonTest <- dataColon[-trainsamples,]
```

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

0	1
313	308

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

0	1
129	138

Modeling

```
ml <- BSWiMS.model(Surv(time,status)~1,data=dataColonTrain,NumberofRepeats = 10)
```

```
[+++++++++++++++++++++++++++++++++++++-----]....
```

```
sm <- summary(ml)
pander::pander(sm$coefficients)
```

Table 3: Table continues below

	Estimate	lower	HR	upper	u.Accuracy
rxLev_5FU_age	-0.004031	0.9942	0.996	0.9977	0.5845
rxLev_5FU	-0.1961	0.7487	0.8219	0.9023	0.5845
age_node4	0.003271	1.002	1.003	1.005	0.5974
age_nodes	0.0003607	1	1	1.001	0.6006
node4	0.204	1.082	1.226	1.389	0.5974
nodes	0.03575	1.008	1.036	1.065	0.6055
extent	0.1219	1.027	1.13	1.243	0.525
adhere	0.08741	1.012	1.091	1.177	0.5314
differ	4.082e-07	1	1	1	0.5362
age	-0.001283	0.9976	0.9987	0.9999	0.5056
rxLev	9.671e-08	1	1	1	0.5378

Table 4: Table continues below

	r.Accuracy	full.Accuracy	u.AUC	r.AUC	full.AUC
rxLev_5FU_age	0.6009	0.6169	0.5858	0.5992	0.6163
rxLev_5FU	0.6017	0.6028	0.5858	0.6001	0.6031
age_node4	0.5917	0.6017	0.5955	0.5927	0.6022
age_nodes	0.5712	0.6103	0.5994	0.572	0.6099
node4	0.612	0.6142	0.5955	0.6122	0.6134
nodes	0.6046	0.6169	0.6043	0.6036	0.6163
extent	0.6023	0.6084	0.528	0.6013	0.6079
adhere	0.6075	0.611	0.5284	0.6064	0.61
differ	0.496	0.5362	0.5336	0.5	0.5336
age	0.5996	0.6156	0.5053	0.5987	0.6149
rxLev	0.496	0.5378	0.5363	0.5	0.5363

	IDI	NRI	z.IDI	z.NRI	Delta.AUC	Frequency
rxLev_5FU_age	0.02345	0.3431	4.443	4.572	0.0171	1
rxLev_5FU	0.02107	0.3422	4.198	4.558	0.002999	1
age_node4	0.02022	0.349	4.008	5.055	0.009442	1
age_nodes	0.01824	0.2767	3.912	3.637	0.03789	1
node4	0.01186	0.2969	3.309	4.366	0.001207	1
nodes	0.006881	0.1462	2.589	1.913	0.01266	1
extent	0.009369	0.1136	2.556	2.114	0.006613	0.9
adhere	0.00603	0.1181	2.419	2.182	0.003684	0.8
differ	0.00803	0.1342	2.241	2.246	0.03356	0.7
age	0.00557	0.1491	2.204	1.867	0.01617	0.2
rxLev	0.004556	0.1453	1.957	1.961	0.03632	0.2

Cox Model Performance

Here we evaluate the model using the `RRPlot()` function.

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

Here we will use the predicted event probability assuming a baseline hazard for events withing 5 years

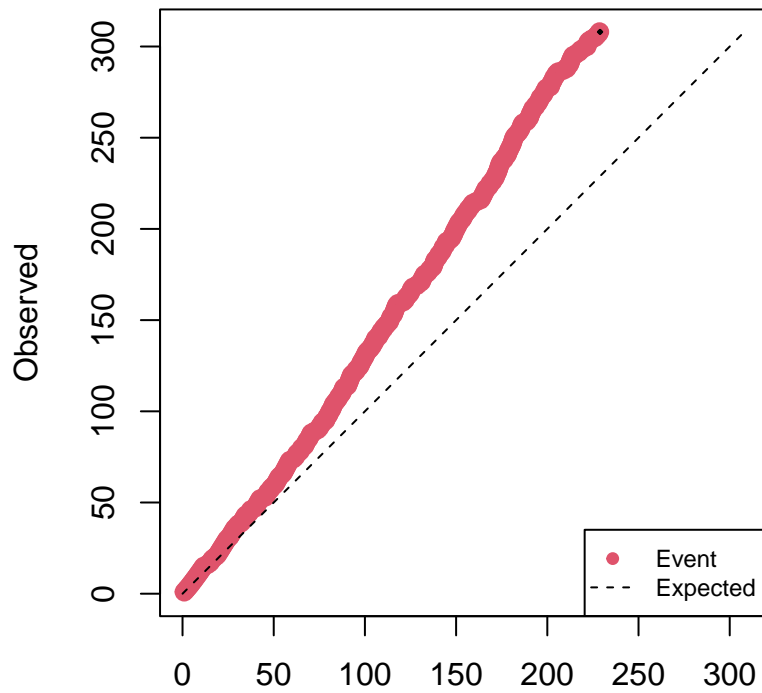
```
index <- predict(m1,dataColonTrain)
timeinterval <- 2*mean(subset(dataColonTrain,status==1)$time)

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

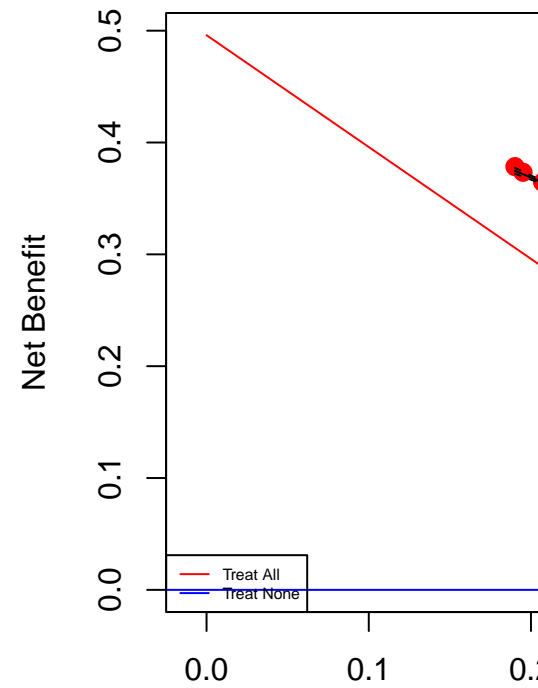
rdata <- cbind(dataColonTrain$status,ppoisGzero(index,h0))

rrAnalysisTrain <- RRPlot(rdata,atProb=c(0.90),
  timetoEvent=dataColonTrain$time,
  title="Raw Train: Colon Cancer",
  ysurvlim=c(0.00,1.0),
  riskTimeInterval=timeinterval)
```

Cumulative vs. Observed: Raw Train: Colon Cancer

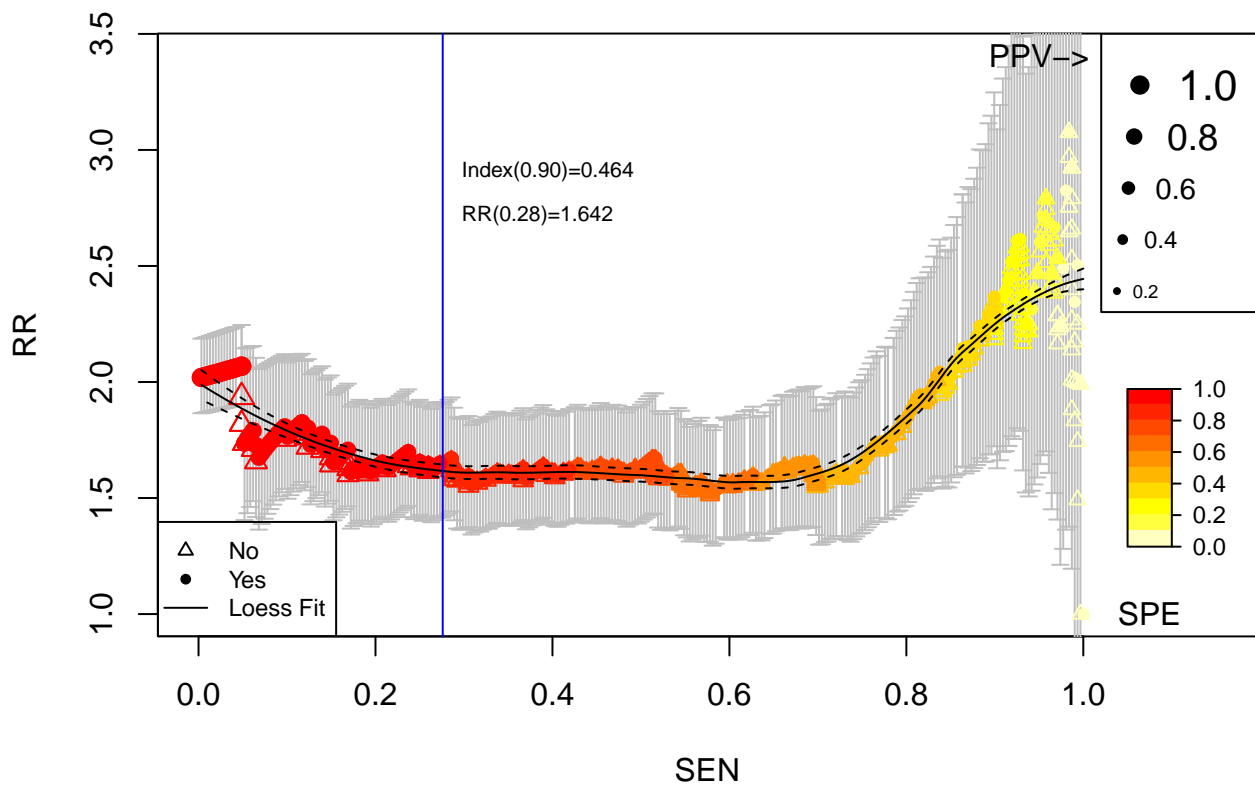


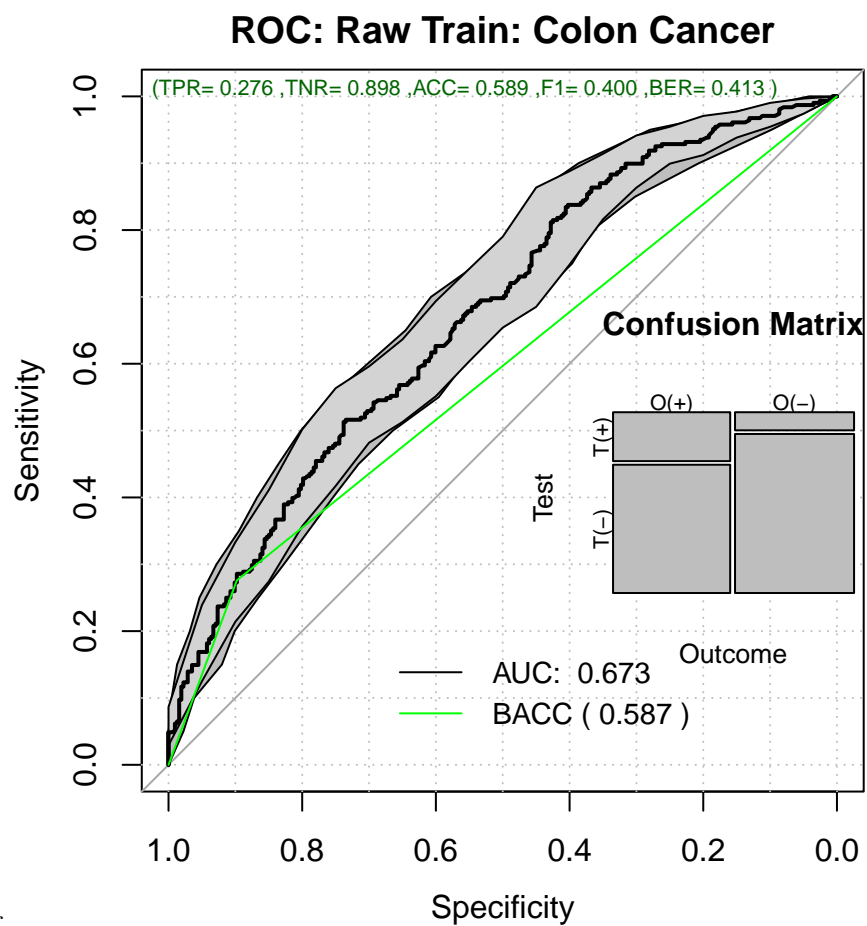
Decision Curve



Cumulative Probability

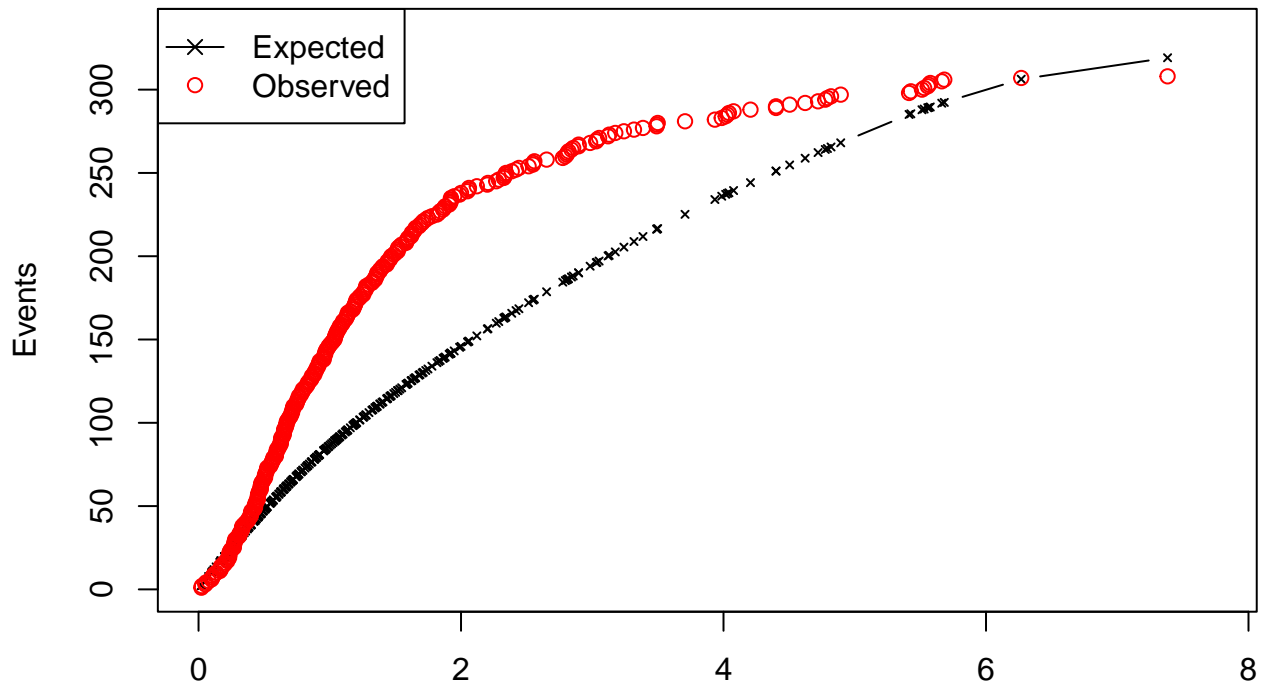
Relative Risk: Raw Train: Colon Cancer



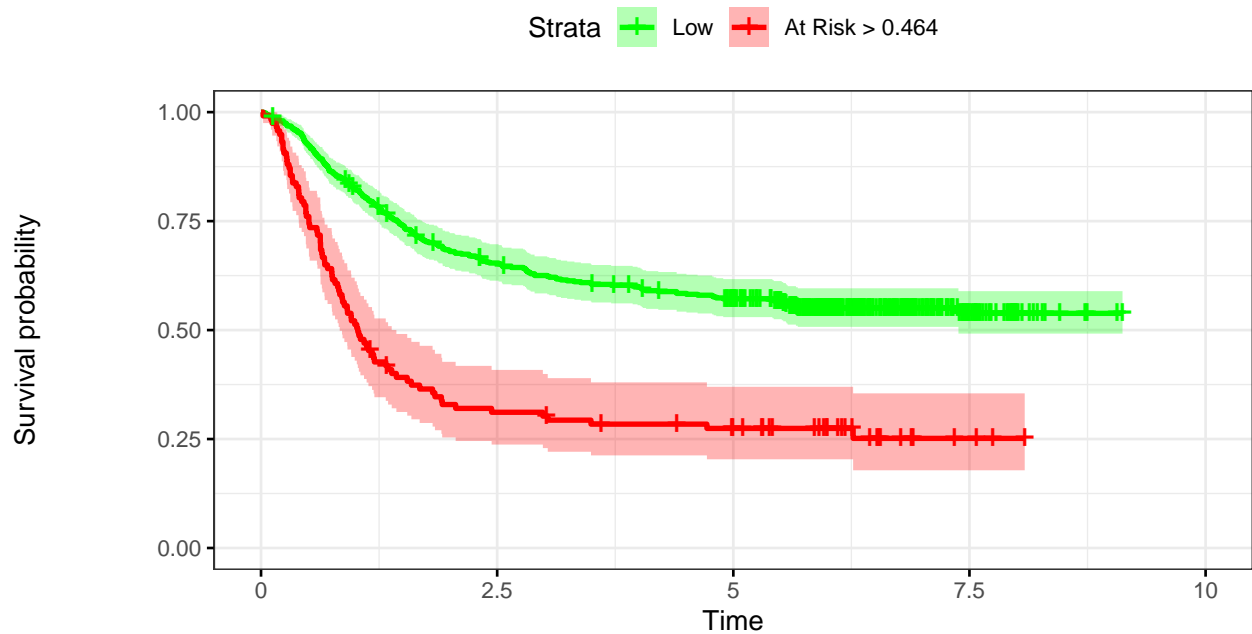


ROC: Raw Train: Colon Cancer

Time vs. Events: Raw Train: Colon Cancer



Kaplan–Meier: Raw Train: Colon Cancer



Number at risk

Low	504	321	267	38	0
At Risk > 0.464	117	35	26	3	0

Uncalibrated Performance Report

```
pander::pander(t(rrAnalysisTrain$OERatio),caption="O/E Ratio")
```

Table 6: O/E Ratio

est	lower	upper
0.9651	0.8603	1.079

```
pander::pander(t(rrAnalysisTrain$OE95ci),caption="O/E Ratio")
```

Table 7: O/E Ratio

mean	50%	2.5%	97.5%
1.479	1.479	1.453	1.508

```
pander::pander(t(rrAnalysisTrain$OAcum95ci),caption="O/Acum Ratio")
```

Table 8: O/Acum Ratio

mean	50%	2.5%	97.5%
1.311	1.312	1.306	1.316

```
pander::pander(rrAnalysisTrain$c.index$cstatCI,caption="C. Index")
```

mean.C Index	median	lower	upper
0.6525	0.6523	0.6214	0.681

```
pander::pander(t(rrAnalysisTrain$ROCAAnalysis$aucs),caption="ROC AUC")
```

Table 10: ROC AUC

est	lower	upper
0.6734	0.6316	0.7152

```
pander::pander((rrAnalysisTrain$ROCAAnalysis$sensitivity),caption="Sensitivity")
```

Table 11: Sensitivity

est	lower	upper
0.276	0.2268	0.3295

```
pander::pander((rrAnalysisTrain$ROCAAnalysis$specificity),caption="Specificity")
```

Table 12: Specificity

est	lower	upper
0.8978	0.8587	0.929

```
pander::pander(t(rrAnalysisTrain$thr_atP),caption="Probability Thresholds")
```

Table 13: Probability Thresholds

90%
0.4643

```
pander::pander(t(rrAnalysisTrain$RR_atP),caption="Risk Ratio")
```

Table 14: Risk Ratio

est	lower	upper
1.642	1.416	1.904

```
pander::pander(rrAnalysisTrain$sufdif,caption="Logrank test")
```

Table 15: Logrank test Chisq = 59.085144 on 1 degrees of freedom,
p = 0.000000

	N	Observed	Expected	(O-E)^2/E	(O-E)^2/V
class=0	504	223	268.1	7.585	59.09
class=1	117	85	39.9	50.96	59.09

Cox Calibration

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

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

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

h0	Gain	DeltaTime
0.6696	1.519	2.942

The RRplot() of the calibrated model


```

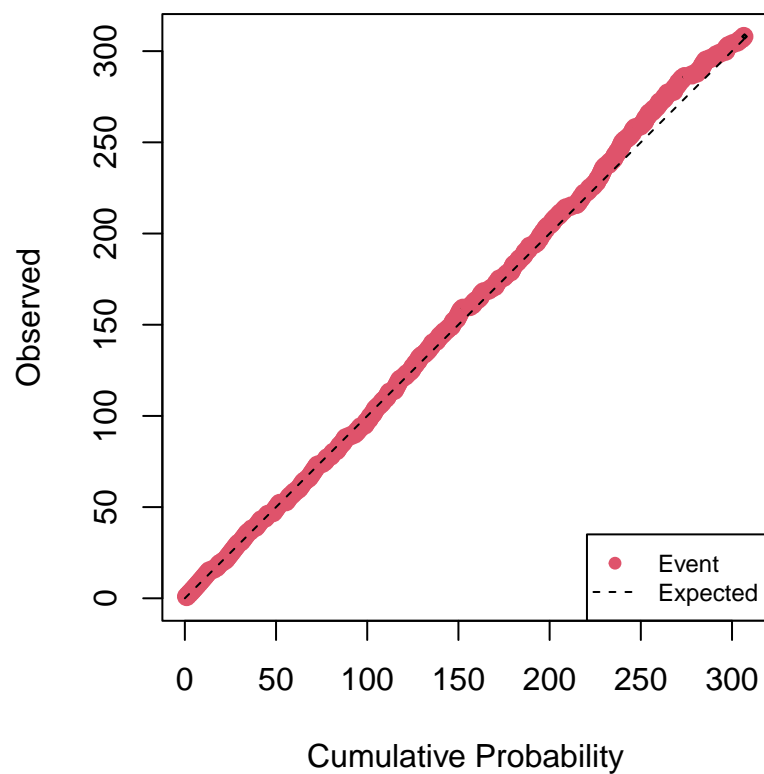
h0 <- calprob$h0
timeinterval <- calprob$timeInterval;

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

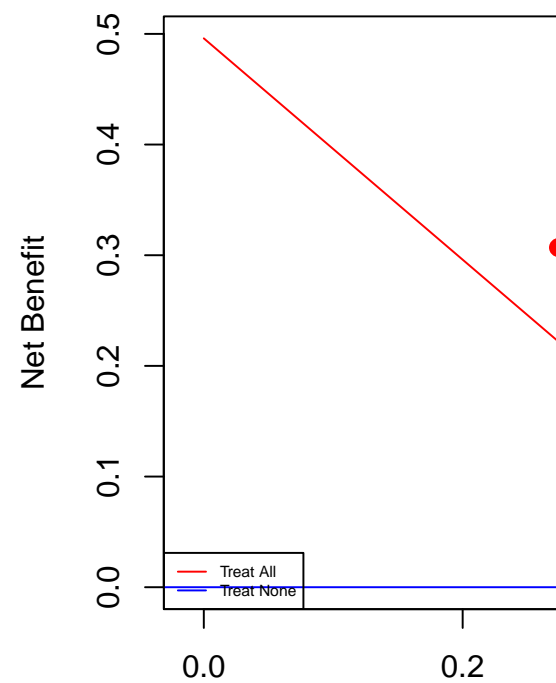
rrAnalysisTrain <- RRPlot(rdata,atProb=c(0.90),
                           timetoEvent=dataColonTrain$time,
                           title="Calibrated Train: Colon",
                           ysurvlim=c(0.00,1.0),
                           riskTimeInterval=timeinterval)

```

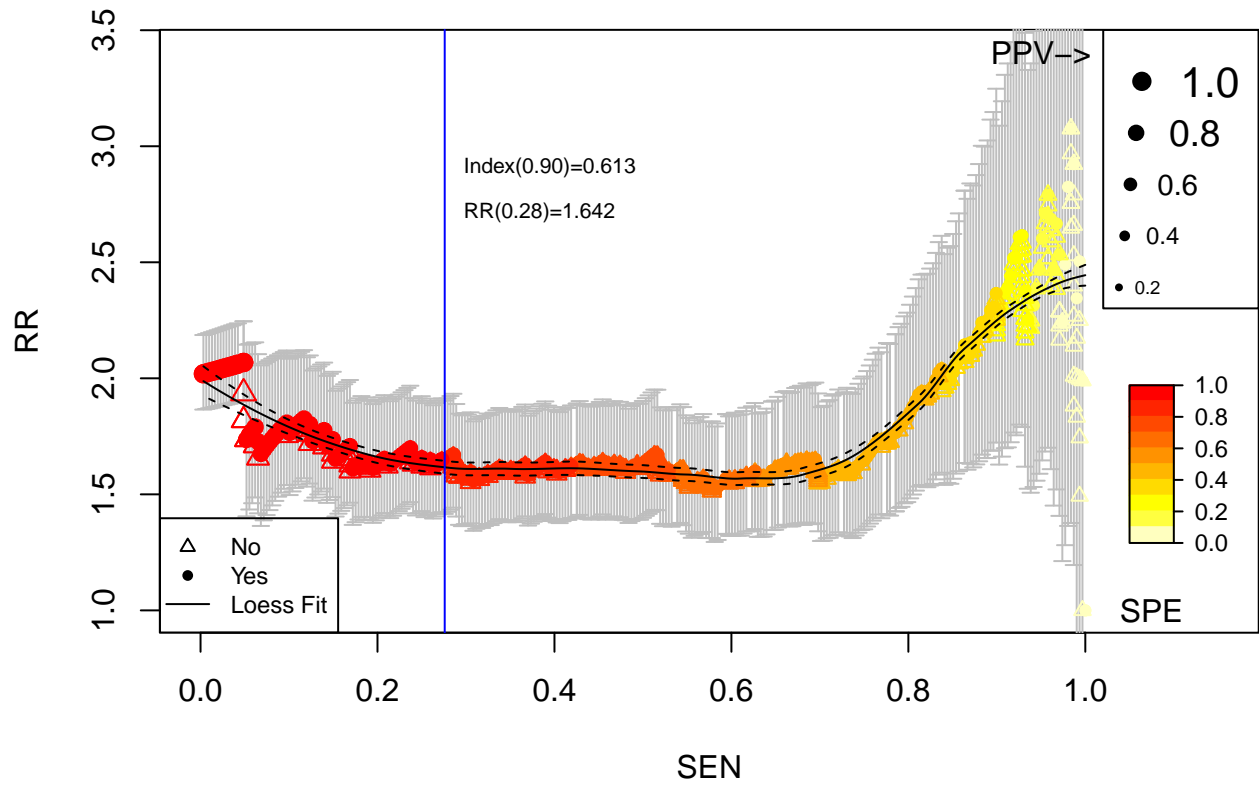
Cumulative vs. Observed: Calibrated Train: Colon

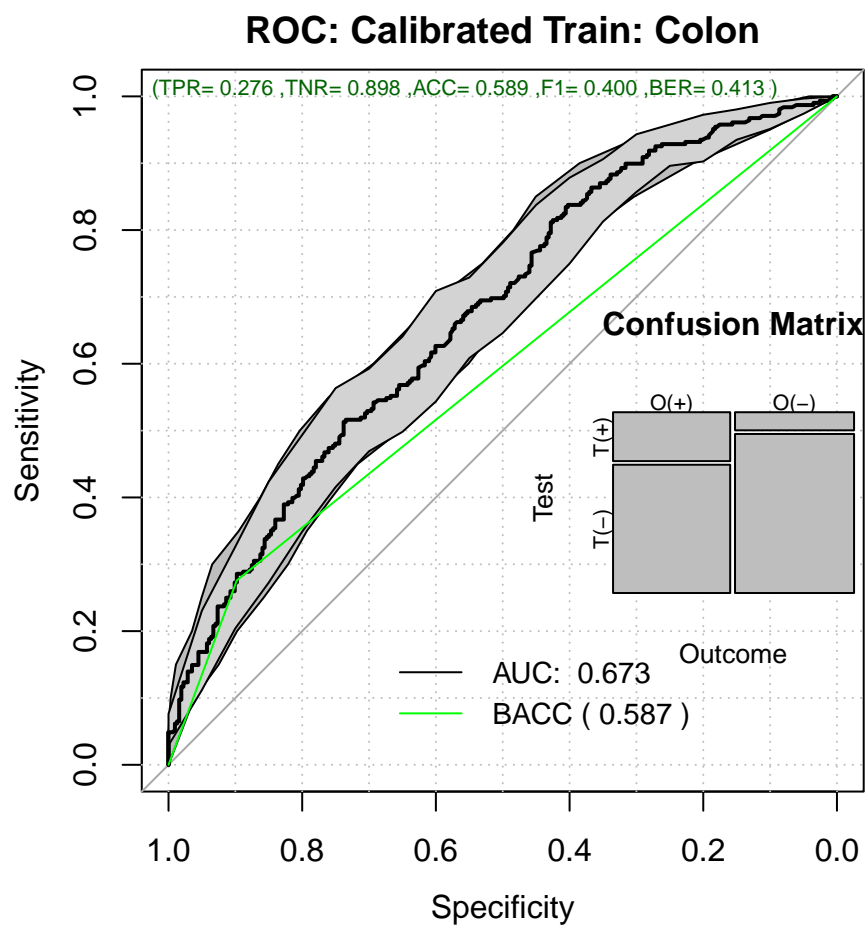


Decision Curve



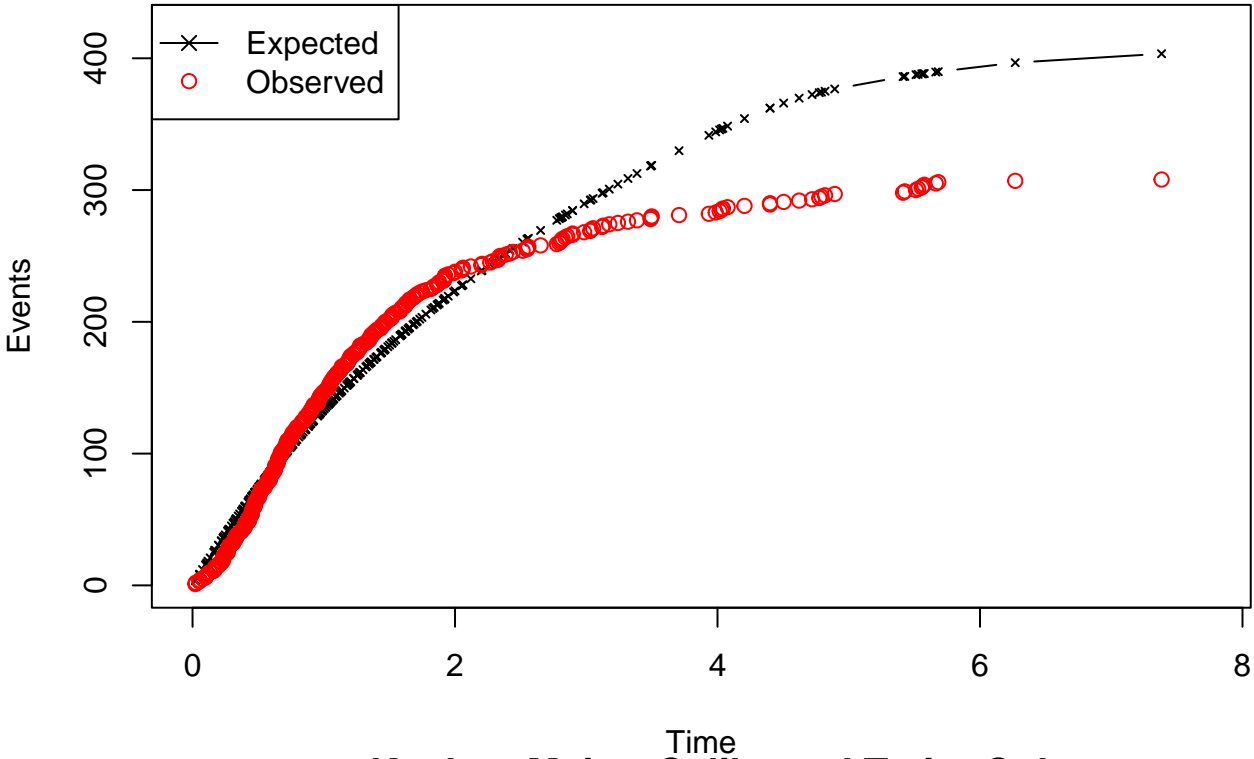
Relative Risk: Calibrated Train: Colon



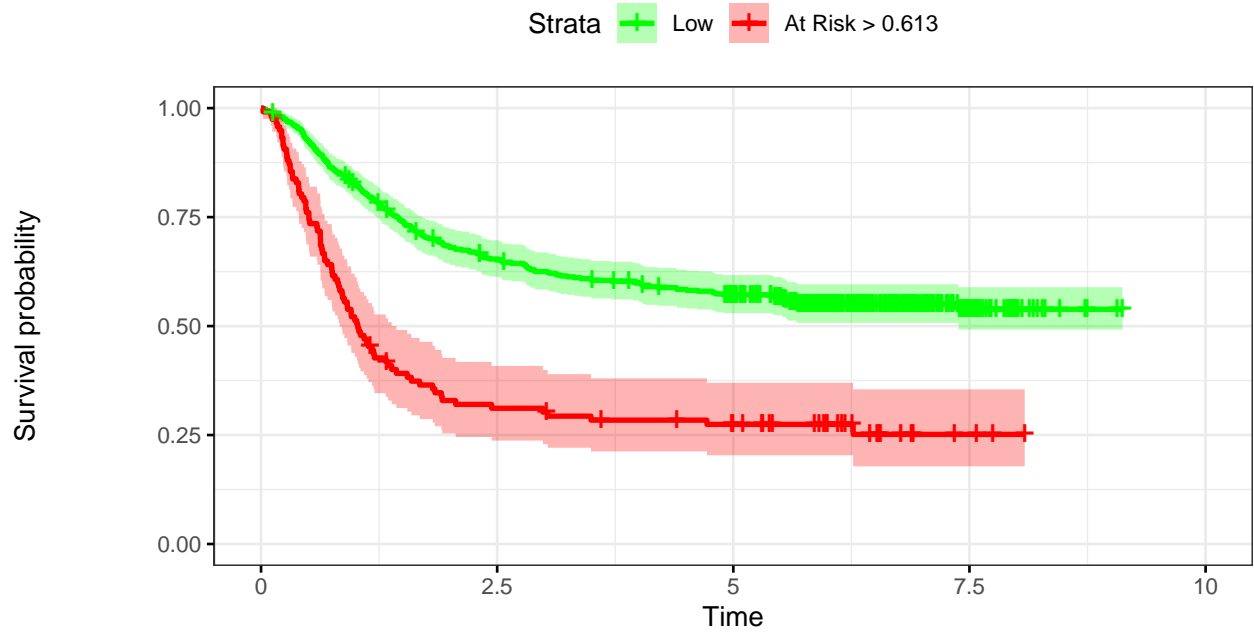


ROC: Calibrated Train: Colon

Time vs. Events: Calibrated Train: Colon



Kaplan–Meier: Calibrated Train: Colon



Number at risk

Low	504	321	267	38	0
At Risk > 0.613	117	35	26	3	0

Calibrated Train Performance

```
pander::pander(t(rrAnalysisTrain$OERatio),caption="O/E Ratio")
```

Table 17: O/E Ratio

est	lower	upper
0.7634	0.6805	0.8536

```
pander::pander(t(rrAnalysisTrain$OE95ci),caption="O/E Ratio")
```

Table 18: O/E Ratio

mean	50%	2.5%	97.5%
0.9684	0.9687	0.9507	0.9851

```
pander::pander(t(rrAnalysisTrain$OAcum95ci),caption="O/Acum Ratio")
```

Table 19: O/Acum Ratio

mean	50%	2.5%	97.5%
1.013	1.013	1.011	1.015

```
pander::pander(rrAnalysisTrain$c.index$cstatCI,caption="C. Index")
```

mean.C Index	median	lower	upper
0.6525	0.6524	0.6246	0.6798

```
pander::pander(t(rrAnalysisTrain$ROCAAnalysis$aucs),caption="ROC AUC")
```

Table 21: ROC AUC

est	lower	upper
0.6734	0.6316	0.7152

```
pander::pander((rrAnalysisTrain$ROCAAnalysis$sensitivity),caption="Sensitivity")
```

Table 22: Sensitivity

est	lower	upper
0.276	0.2268	0.3295

```
pander::pander((rrAnalysisTrain$ROCAAnalysis$specificity),caption="Specificity")
```

Table 23: Specificity

est	lower	upper
0.8978	0.8587	0.929

```
pander::pander(t(rrAnalysisTrain$thr_atP),caption="Probability Thresholds")
```

Table 24: Probability Thresholds

90%
0.6125

```
pander::pander(t(rrAnalysisTrain$RR_atP),caption="Risk Ratio")
```

Table 25: Risk Ratio

est	lower	upper
1.642	1.416	1.904

```
pander::pander(rrAnalysisTrain$sufdif,caption="Logrank test")
```

Table 26: Logrank test Chisq = 59.085144 on 1 degrees of freedom,
p = 0.000000

	N	Observed	Expected	(O-E)^2/E	(O-E)^2/V
class=0	504	223	268.1	7.585	59.09
class=1	117	85	39.9	50.96	59.09

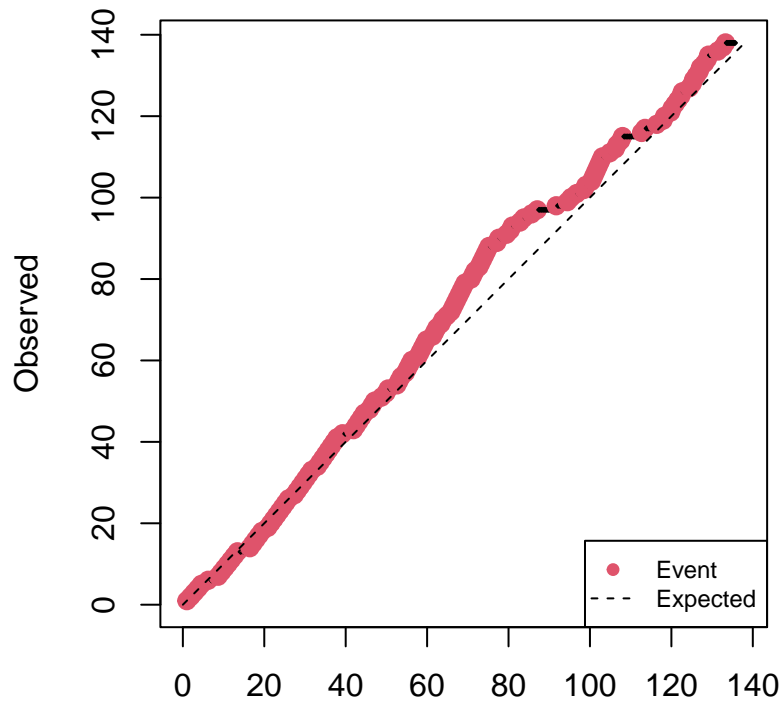
Evaluating on the test set

The calibrated h0 and timeinterval were estimated on the training set

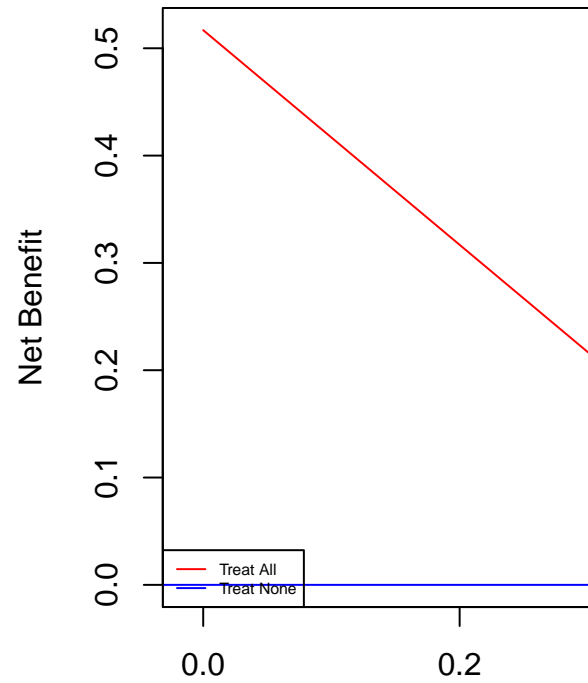
```
index <- predict(ml,dataColonTest)
rdata <- cbind(dataColonTest$status,ppoisGzero(index,h0))

rrAnalysisTest <- RRPlot(rdata,atThr = rrAnalysisTrain$thr_atP,
  timetoEvent=dataColonTest$time,
  title="Test: Colon Cancer",
  ysurvlim=c(0.00,1.0),
  riskTimeInterval=timeinterval)
```

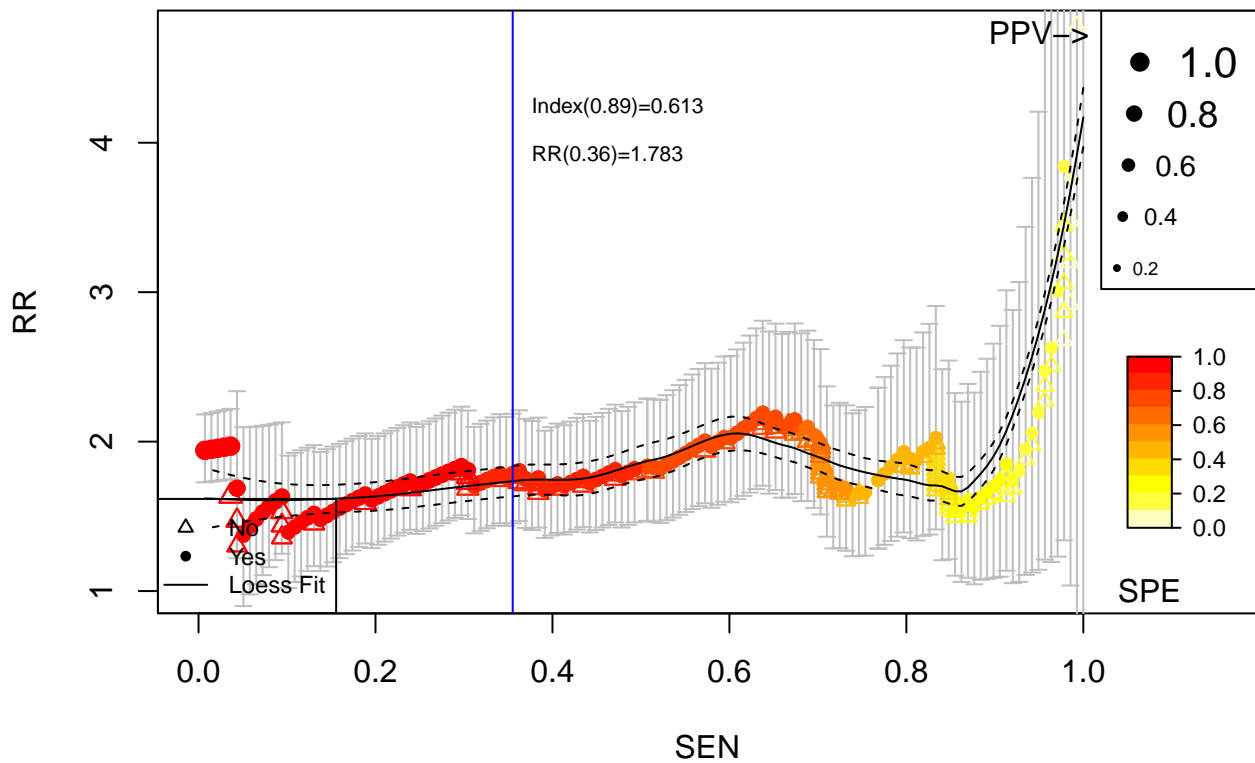
Cumulative vs. Observed: Test: Colon Cancer



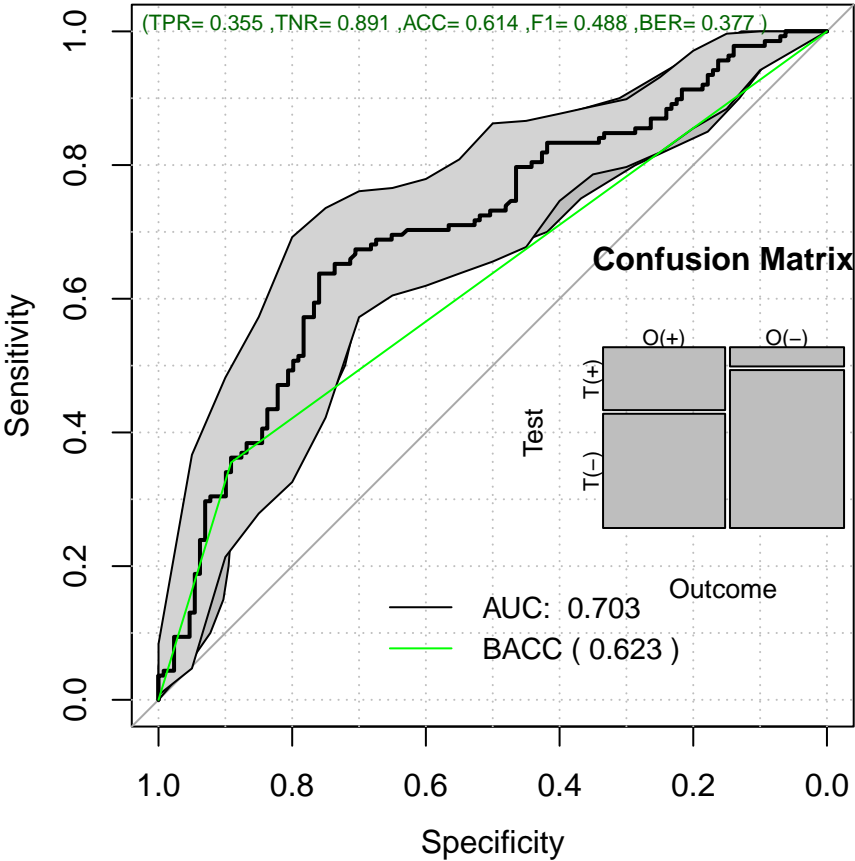
Decision Curve



Relative Risk: Test: Colon Cancer

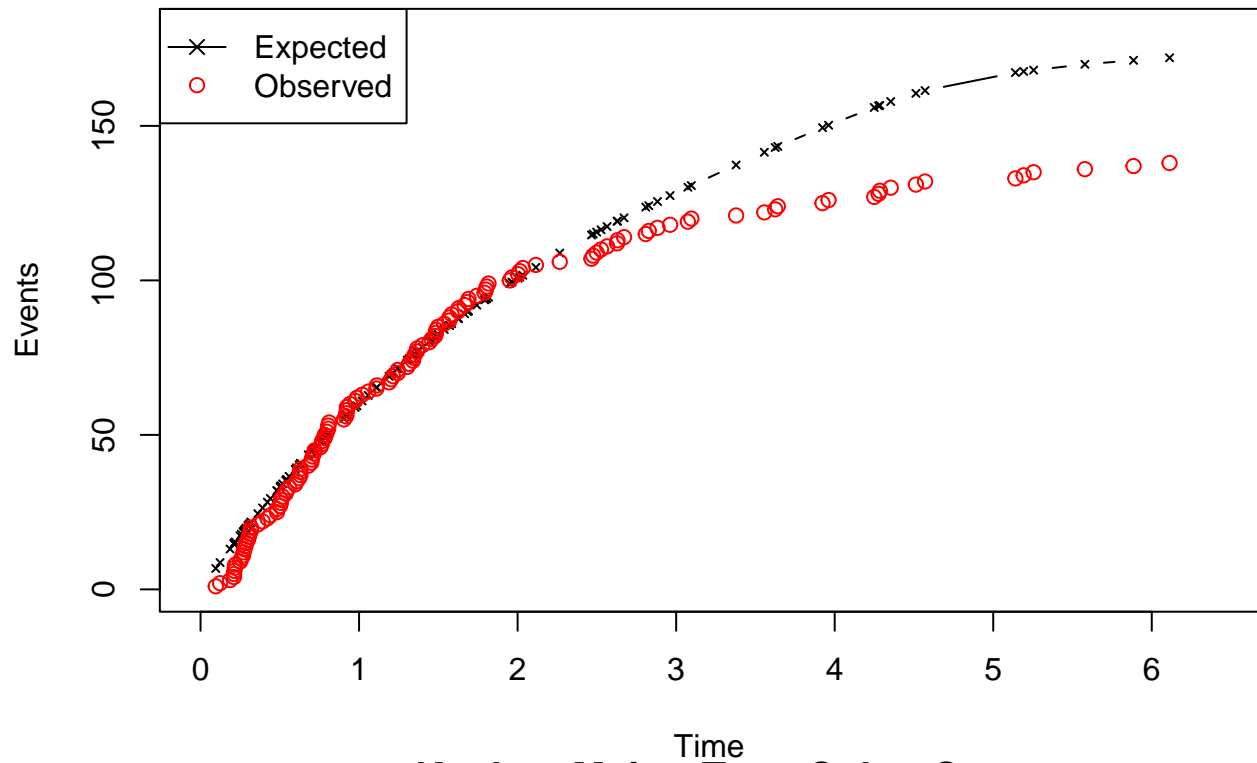


ROC: Test: Colon Cancer

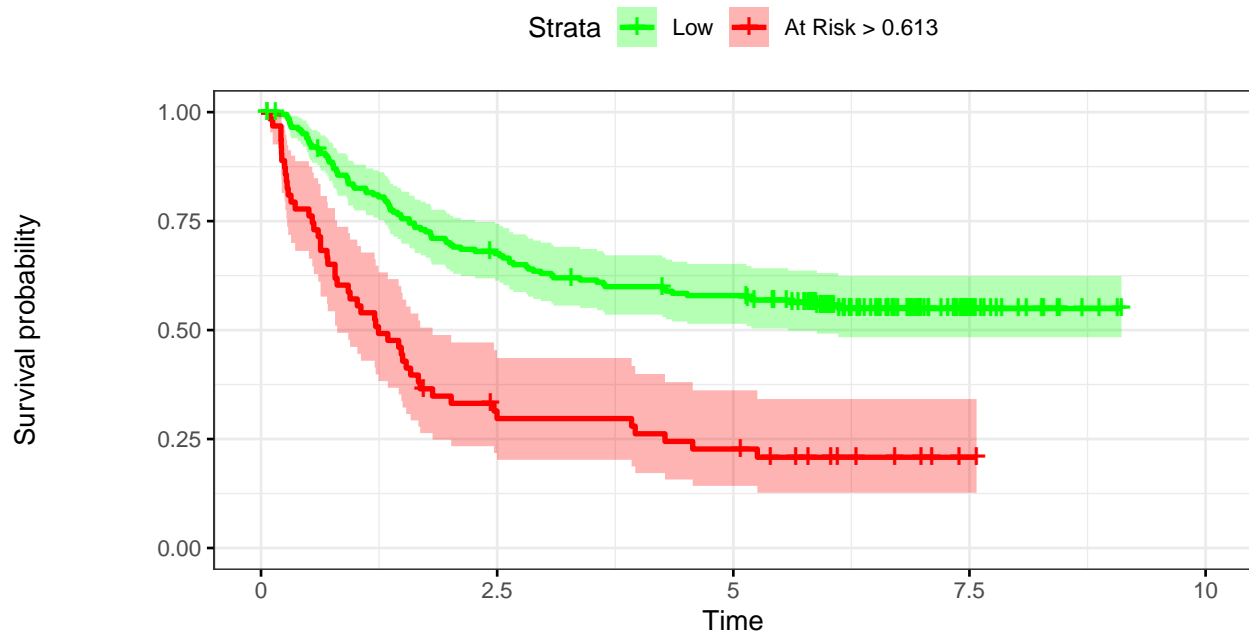


ROC: Test: Colon Cancer

Time vs. Events: Test: Colon Cancer



Kaplan–Meier: Test: Colon Cancer



Number at risk

Low	204	134	113	20	0
At Risk > 0.613	63	17	13	1	0

Test Performance

```
pander::pander(t(rrAnalysisTest$OERatio),caption="O/E Ratio")
```

Table 27: O/E Ratio

est	lower	upper
0.8021	0.6739	0.9476

```
pander::pander(t(rrAnalysisTest$OE95ci),caption="O/E Ratio")
```

Table 28: O/E Ratio

mean	50%	2.5%	97.5%
0.9212	0.921	0.9002	0.941

```
pander::pander(t(rrAnalysisTest$OAcum95ci),caption="O/Acum Ratio")
```

Table 29: O/Acum Ratio

mean	50%	2.5%	97.5%
1.046	1.046	1.039	1.053

```
pander::pander(rrAnalysisTest$c.index$cstatCI,caption="C. Index")
```

mean.C Index	median	lower	upper
0.6525	0.6526	0.6058	0.6979

```
pander::pander(t(rrAnalysisTest$ROCAalysis$aucs),caption="ROC AUC")
```

Table 31: ROC AUC

est	lower	upper
0.7033	0.6405	0.766

```
pander::pander((rrAnalysisTest$ROCAalysis$sensitivity),caption="Sensitivity")
```

Table 32: Sensitivity

est	lower	upper
0.3551	0.2755	0.441

```
pander::pander((rrAnalysisTest$ROCAalysis$specificity),caption="Specificity")
```

Table 33: Specificity

est	lower	upper
0.8915	0.8246	0.9394

```
pander::pander(t(rrAnalysisTest$thr_atP),caption="Probability Thresholds")
```

Table 34: Probability Thresholds

90%
0.6125

```
pander::pander(t(rrAnalysisTest$RR_atP),caption="Risk Ratio")
```

Table 35: Risk Ratio

est	lower	upper
1.783	1.453	2.187

```
pander::pander(rrAnalysisTest$sufdif,caption="Logrank test")
```

Table 36: Logrank test Chisq = 36.794054 on 1 degrees of freedom,
p = 0.000000

	N	Observed	Expected	(O-E)^2/E	(O-E)^2/V
class=0	204	89	115.3	5.988	36.79
class=1	63	49	22.73	30.37	36.79

Cross-Validation

Here we will cross validate the training set and evaluate also on the testing set. The h0 and the timeinterval are the ones estimated on the calibration process

```
rcv <- randomCV(theData=dataColonTrain,
  theOutcome = Surv(time,status)~1,
  fittingFunction=BSWiMS.model,
  trainFraction = 0.75,
  repetitions=50,
  classSamplingType = "Pro",
  testingSet=dataColonTest
)
```

```
## .[+++++] .[++++] .[+++] .[+++-.] .[+++++] .[+++-.] .[++++] .[+++++] .[+++++] .[+++]10 Tested: 848 Avg. Selected: 8.
## .[++++-] .[++++] .[+++++] .[++++-] .[+++] .[++++] .[++] .[+++] .[+++] .[++]20 Tested: 886 Avg. Selected: 8.
## .[+++-.] .[+++-.] .[+++++] .[++++] .[+++] .[++++-] .[++] .[+++++] .[++++-] .[++++-]30 Tested: 888 Avg. Selected: 8.
## .[+++-.] .[++++] .[+++-.] .[+++] .[+++] .[++++] .[+++] .[++-] .[+++++] .[+++-.]40 Tested: 888 Avg. Selected: 8.
## .[++++] .[++++-] .[+++++] .[++++-] .[+++] .[++++-] .[++++] .[+++] .[++++-] .[++++-]50 Tested: 888 Avg. Selected: 8.
##
```

```

stp <- rcv$urvTestPredictions
stp <- stp[!is.na(stp[,4]),]

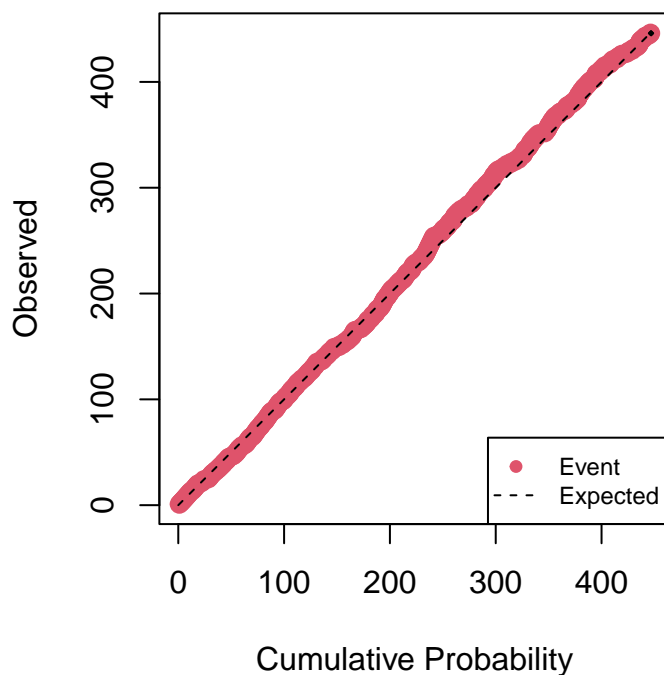
bbx <- boxplot(unlist(stp[,1])~rownames(stp),plot=FALSE)
times <- bbx$stats[3,]
status <- boxplot(unlist(stp[,2])~rownames(stp),plot=FALSE)$stats[3,]
prob <- ppoisGzero(boxplot(unlist(stp[,4])~rownames(stp),plot=FALSE)$stats[3,],h0)

rdatacv <- cbind(status,prob)
rownames(rdatacv) <- bbx$names
names(times) <- bbx$names

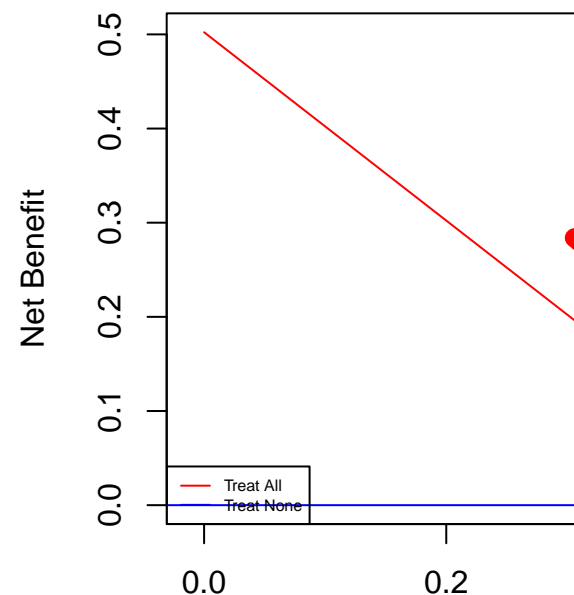
rrAnalysisCVTest <- RRPlot(rdatacv,atThr = rrAnalysisTrain$thr_atP,
                           timetoEvent=times,
                           title="CV Test: Colon Cancer",
                           ysurvlim=c(0.00,1.0),
                           riskTimeInterval=timeinterval)

```

Cumulative vs. Observed: CV Test: Colon Cancer



Decision Curve Analysis



```

## Warning in plot.xy(xy.coords(x, y), type = type, ...): font width unknown for
## character 0x1

## Warning in plot.xy(xy.coords(x, y), type = type, ...): font metrics unknown for
## character 0x1

## Warning in plot.xy(xy.coords(x, y), type = type, ...): font width unknown for
## character 0x1

## Warning in plot.xy(xy.coords(x, y), type = type, ...): font metrics unknown for
## character 0x1

## Warning in plot.xy(xy.coords(x, y), type = type, ...): font width unknown for

```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): font metrics unknown for
## character 0x1

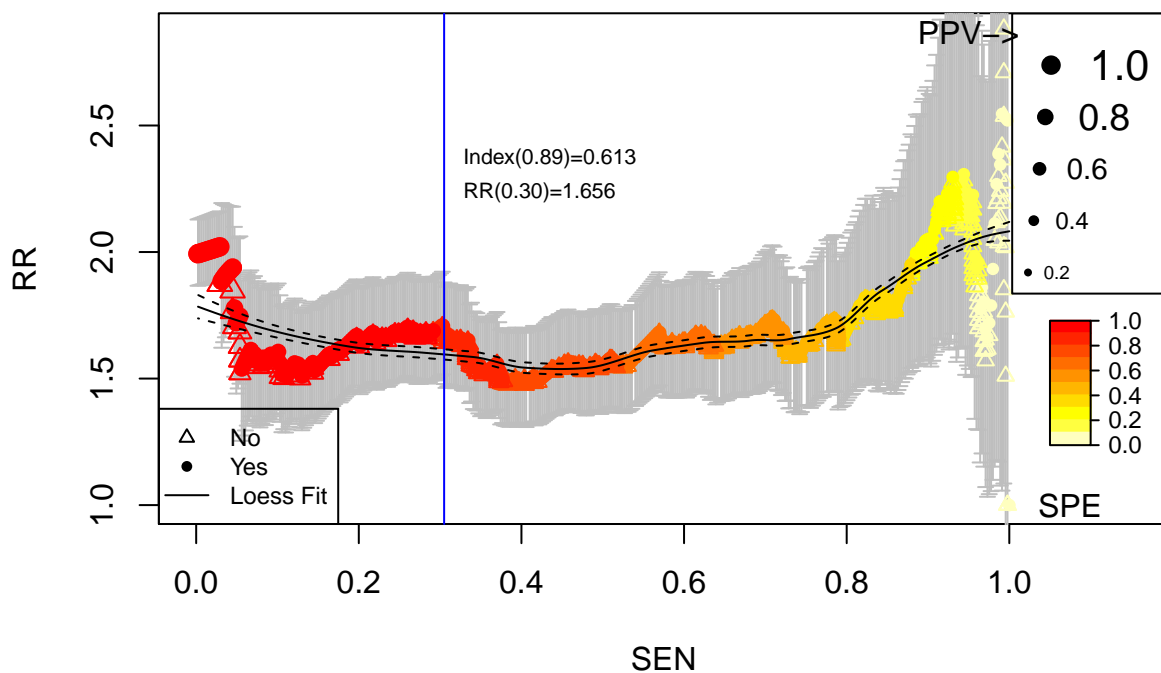
## Warning in plot.xy(xy.coords(x, y), type = type, ...): font width unknown for
## character 0x1

## Warning in plot.xy(xy.coords(x, y), type = type, ...): font metrics unknown for
## character 0x1

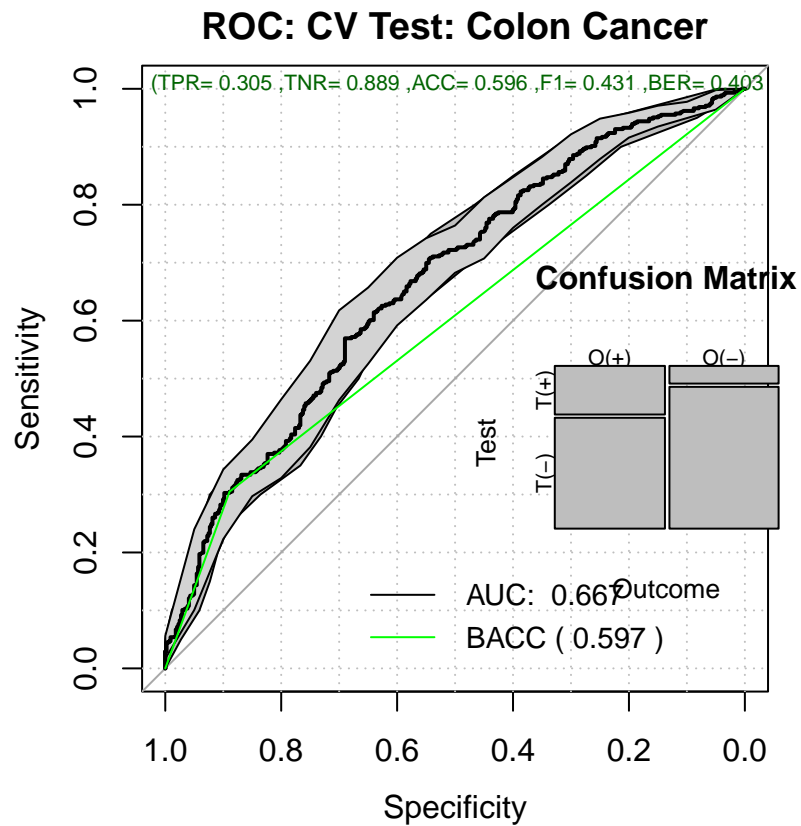
## Warning in plot.xy(xy.coords(x, y), type = type, ...): font width unknown for
## character 0x1

## Warning in plot.xy(xy.coords(x, y), type = type, ...): font metrics unknown for
## character 0x1
```

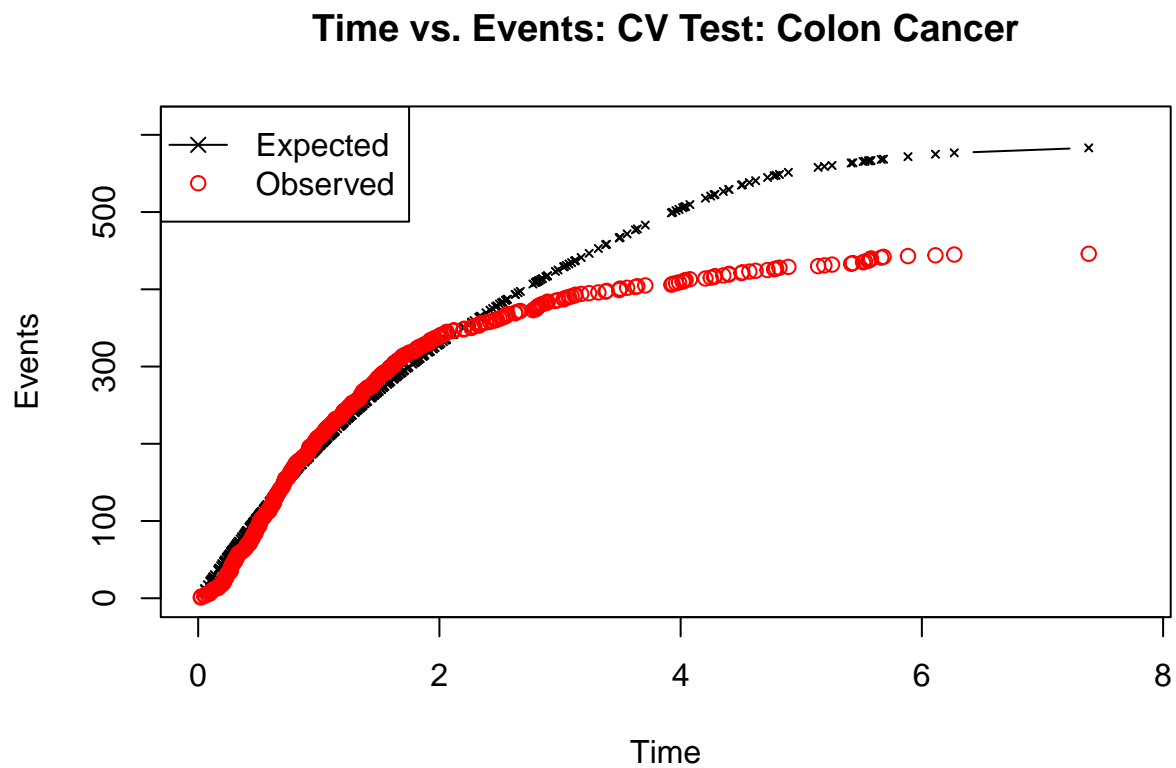
Relative Risk: CV Test: Colon Cancer



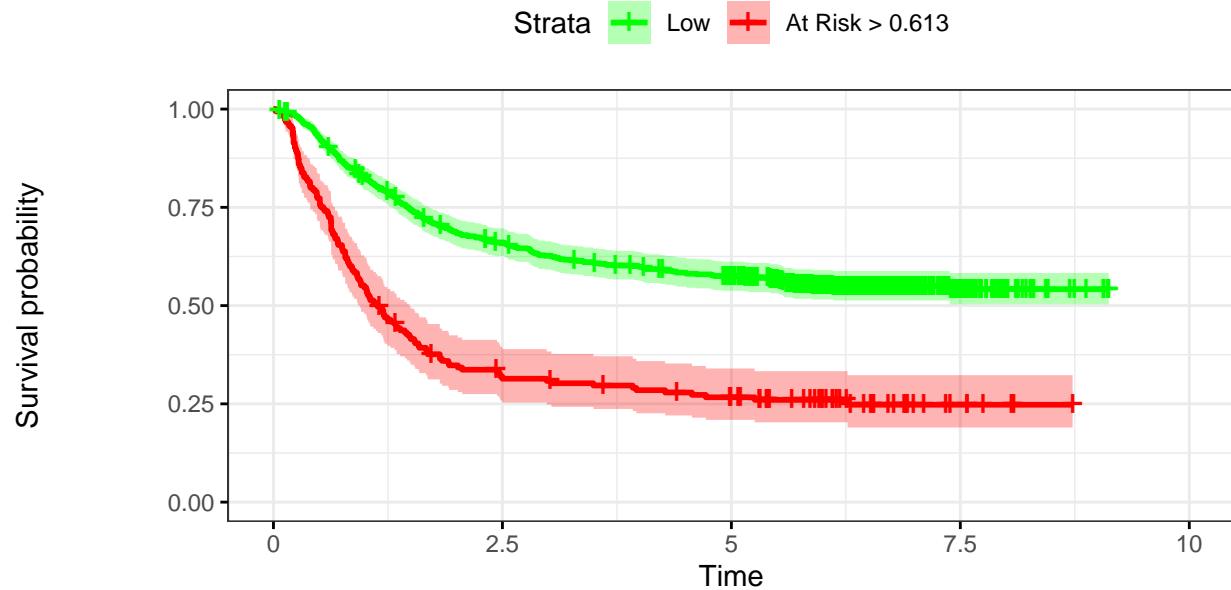
```
## ROC: CV Test: Colon Cancer
```



```
## Warning in cstat$cstatCI <- cstatCI: Coercing LHS to a list
```



Kaplan–Meier: CV Test: Colon Cancer



Number at risk

Low	703	452	377	56	0
At Risk > 0.613	185	55	42	6	0

CV Test Performance

```
pander::pander(t(rrAnalysisCVTest$OERatio),caption="O/E Ratio")
```

Table 37: O/E Ratio

est	lower	upper
0.7648	0.6955	0.8392

```
pander::pander(t(rrAnalysisCVTest$OE95ci),caption="O/E Ratio")
```

Table 38: O/E Ratio

mean	50%	2.5%	97.5%
0.9408	0.9413	0.9274	0.9535

```
pander::pander(t(rrAnalysisCVTest$OAcum95ci),caption="O/Acum Ratio")
```

Table 39: O/Acum Ratio

mean	50%	2.5%	97.5%
1.008	1.008	1.005	1.01

```
pander::pander(rrAnalysisCVTest$c.index$cstatCI,caption="C. Index")
```

mean.C Index	median	lower	upper
0.6437	0.6435	0.6162	0.6689

```
pander::pander(t(rrAnalysisCVTest$ROCAalysis$aucs),caption="ROC AUC")
```

Table 41: ROC AUC

est	lower	upper
0.6669	0.6316	0.7021

```
pander::pander((rrAnalysisCVTest$ROCAalysis$sensitivity),caption="Sensitivity")
```

Table 42: Sensitivity

est	lower	upper
0.3049	0.2625	0.35

```
pander::pander((rrAnalysisCVTest$ROCAalysis$specificity),caption="Specificity")
```

Table 43: Specificity

est	lower	upper
0.8891	0.8561	0.9169

```
pander::pander(t(rrAnalysisCVTest$thr_atP),caption="Probability Thresholds")
```

Table 44: Probability Thresholds

90%
0.6125

```
pander::pander(t(rrAnalysisCVTest$RR_atP),caption="Risk Ratio")
```

Table 45: Risk Ratio

est	lower	upper
1.656	1.468	1.868

```
pander::pander(rrAnalysisCVTest$sufdif,caption="Logrank test")
```

Table 46: Logrank test Chisq = 89.573729 on 1 degrees of freedom,
p = 0.000000

	N	Observed	Expected	(O-E)^2/E	(O-E)^2/V
class=0	703	310	380.5	13.05	89.57
class=1	185	136	65.55	75.73	89.57