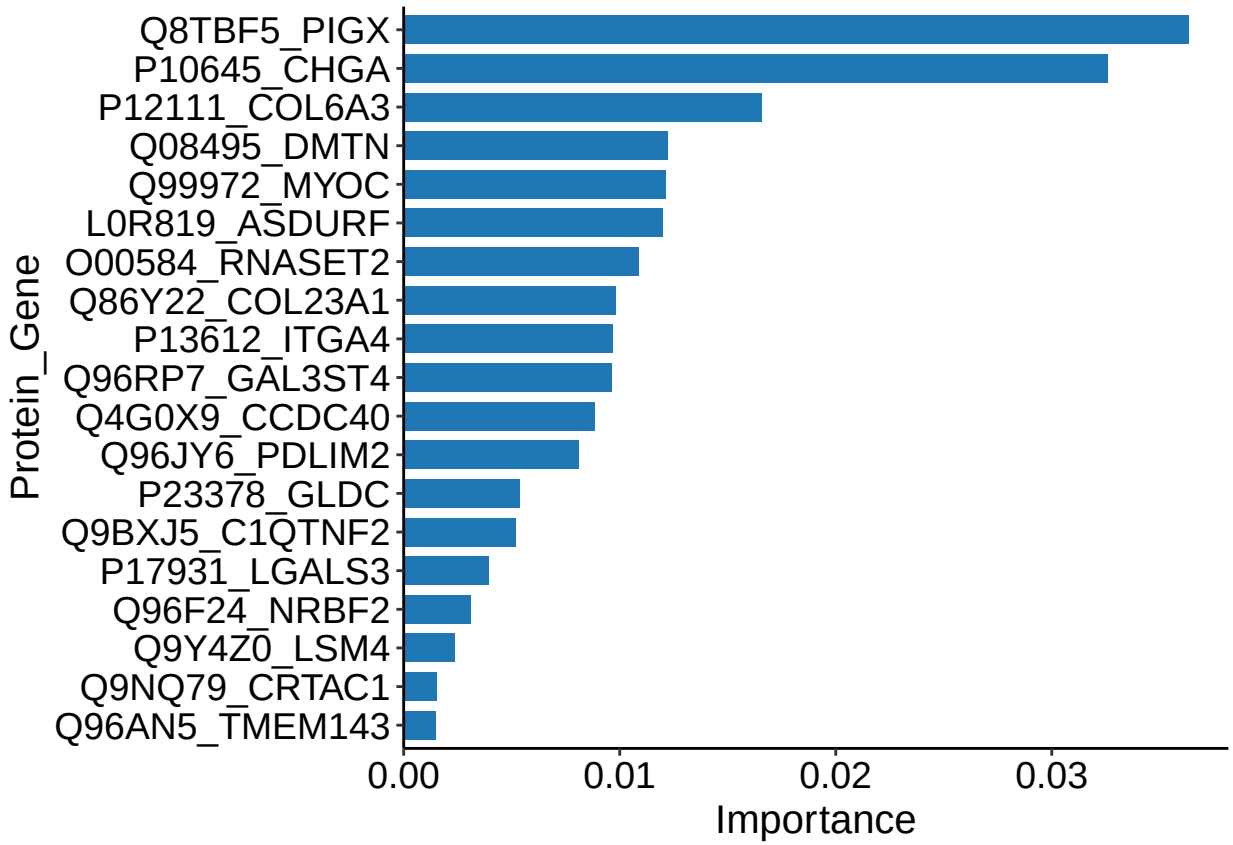
Model_Inference

2024-02-29

```
#Model Inference

library(mlr3)
library(mlr3proba)
library(survival)
library(mlr3extralearners)
library(mlr3learners)
library(mlr3verse)
library(mlr3viz)
library(mlr3tuning)
library(ggplot2)

#Importance
Importance <- as.data.frame(learner_RF_Best$importance())
Importance <- data.frame(rownames(Importance), Importance)
rownames(Importance) <- NULL
colnames(Importance) <- c("Protein_Gene","Importance")
Importance$Protein_Gene<- c("Q8TBF5_PIGX","P10645_CHGA","P12111_COL6A3","Q08495_DMTN","Q99972_MYOC","LO
                            "O00584_RNASET2","Q86Y22_COL23A1","P13612_ITGA4","Q96RP7_GAL3ST4","Q4G0X9_C
                            "P23378_GLDC","Q9BXJ5_C1QTNF2","P17931_LGALS3","Q96F24_NRBF2","Q9Y4Z0_LSM4"
                            "Q96AN5_TMEM143")
library(ggcharts)
bar_chart(Importance, Protein_Gene, Importance) +
  theme_classic() +
  theme(axis.title.x = element_text(size = 15), axis.title.y = element_text(size = 15)) +
  theme(axis.text.x = element_text(size = 14, color = "black"), axis.text.y = element_text(size = 14, c
```




```
#Stratification

#Train
print(learner_RF_Best$predict(task_RF, row_ids = TrainingIndex)$score(msr("surv.cindex")))
```

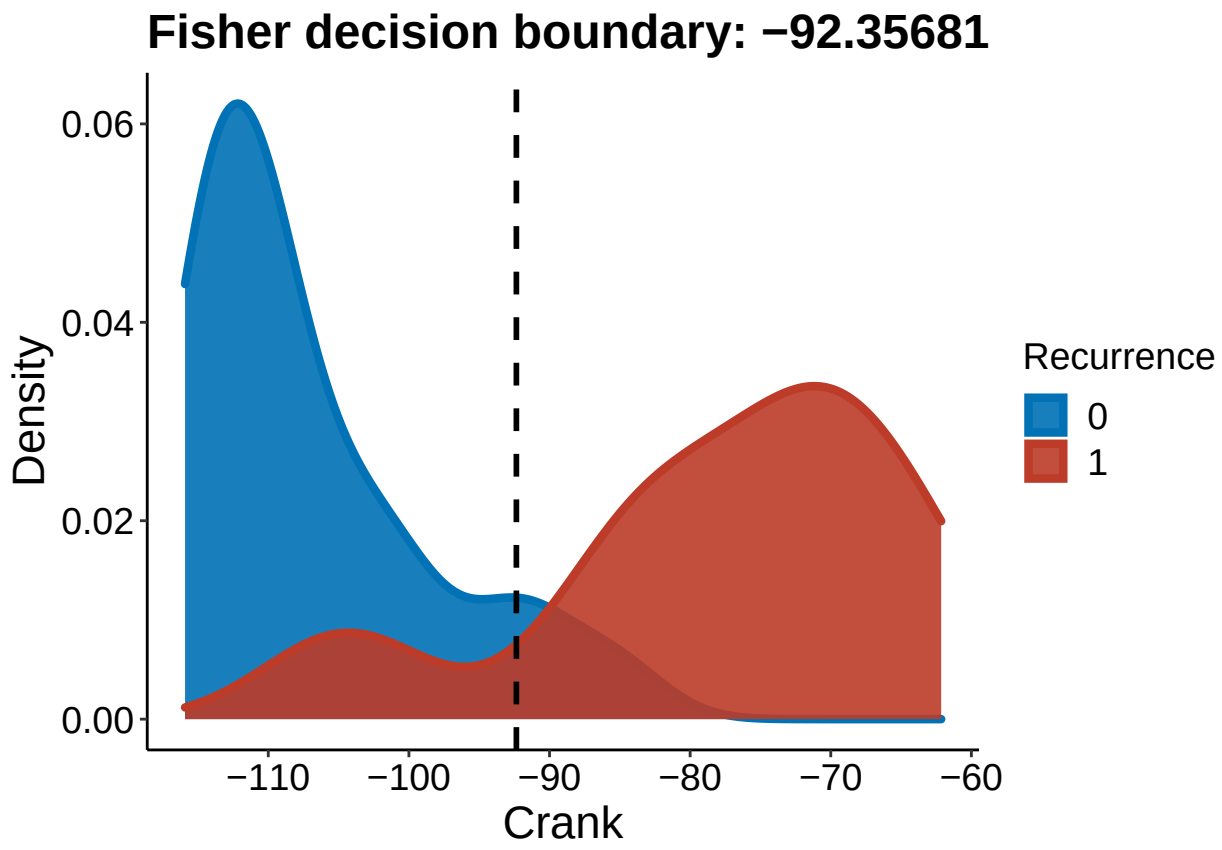
```
## surv.harrell_c
##      0.9961538
```

```
Prediction_RF <- learner_RF_Best$predict(task_RF, row_ids = TrainingIndex)
Crank <- data.frame(Index = TrainingIndex, Recurrence = Label_PM$Recurrence[TrainingIndex], Crank = Pred
(Cutoff <- (mean(Crank[which(Crank$Recurrence == 1), 3]) + mean(Crank[which(Crank$Recurrence == 0), 3]))
```

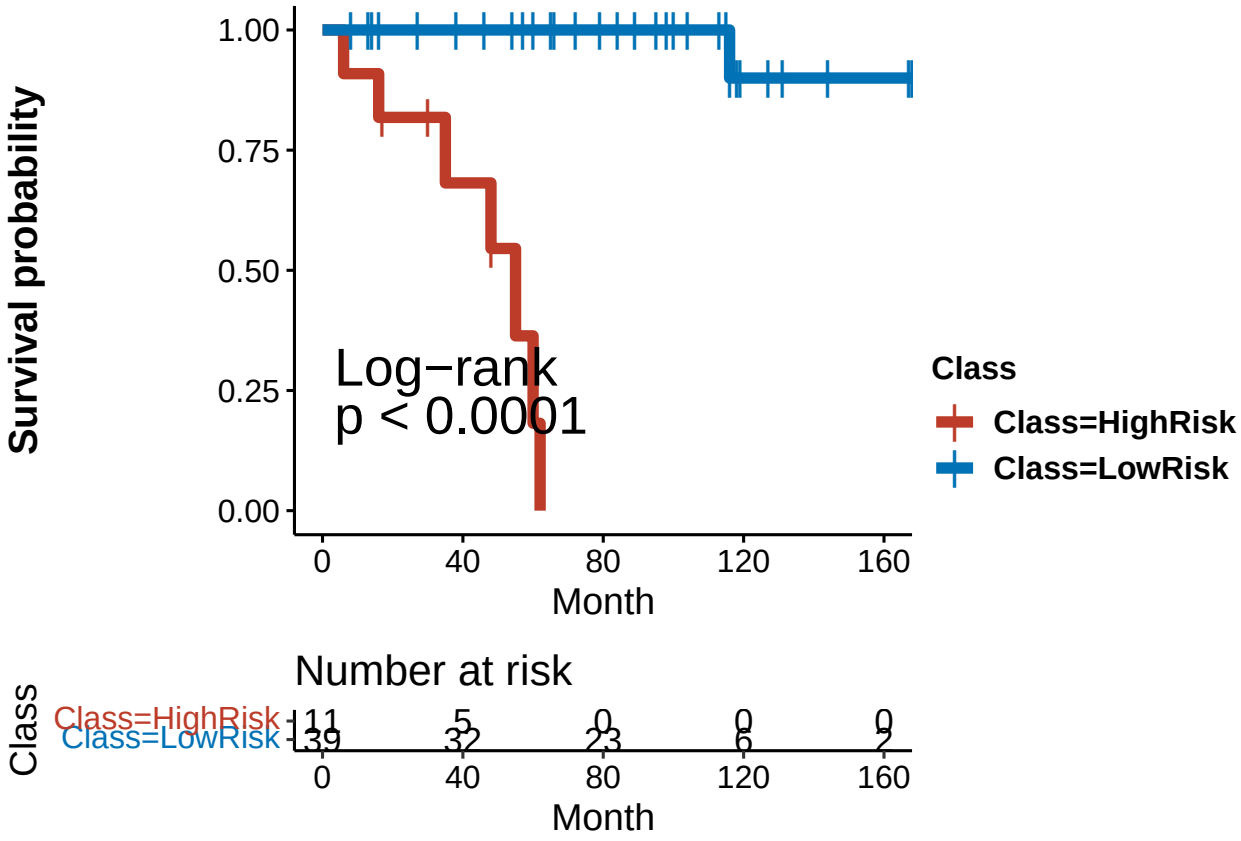
```
## [1] -92.35681
```

```
Crank$Recurrence <- as.factor(Crank$Recurrence)
ggplot(Crank, aes(x = Crank, fill = Recurrence, color = Recurrence)) +
  geom_density(alpha = 0.9, position = "identity", size = 1.5) +
  scale_color_manual(values = c("#0072B5FF","#BC3C29FF")) +
  scale_fill_manual(values = c("#0072B5FF","#BC3C29FF")) +
  theme_classic() +
  theme(axis.title.x = element_text(size = 17), axis.title.y = element_text(size = 17)) +
  theme(legend.text = element_text(size = 14), legend.title = element_text(size = 14)) +
  theme(axis.text.x = element_text(size = 14, color = "black"), axis.text.y = element_text(size = 14, co
  labs(x = "Crank", y = "Density") +
```

```
  ggtitle("Fisher decision boundary: -92.35681") +
  theme(plot.title = element_text(size = 18, face = "bold")) +
  geom_vline(xintercept = (mean(Crank[which(Crank$Recurrence == 1), 3]) + mean(Crank[which(Crank$Recurr
```




```
library(survival)
library(survminer)
Class <- 1:50
Class[which(Crank$Crank >= (mean(Crank[which(Crank$Recurrence == 1), 3]) + mean(Crank[which(Crank$Recur
Class[which(Crank$Crank < (mean(Crank[which(Crank$Recurrence == 1), 3]) + mean(Crank[which(Crank$Recurr
KMdata <- data.frame(Time = Label_PM$FUorRInterval[TrainingIndex], Status = (as.numeric(Label_PM$Recurr
fit <- survfit(Surv(Time,Status) ~ Class, data = KMdata)
ggsurvplot(fit, data = KMdata, palette = c("#BC3C29FF", "#0072B5FF"), pval = TRUE, pval.method = T, lin
```

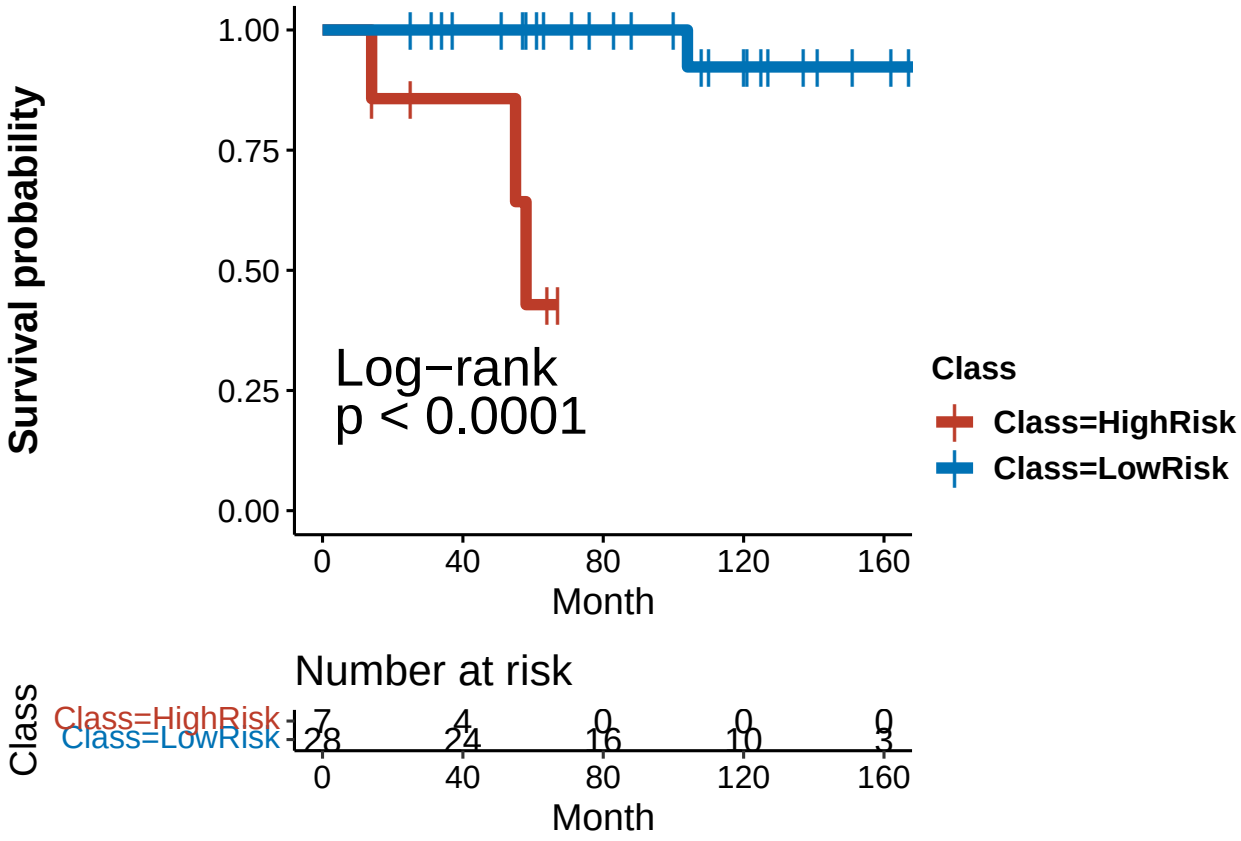



```
HighRiskIndex1 <- TrainingIndex[which(Crank$Crank >= Cutoff)]
LowRiskIndex1 <- TrainingIndex[which(Crank$Crank < Cutoff)]
```

```
#Test
Prediction_RF <- learner_RF_Best$predict(task_RF, row_ids = TestingIndex)
print(learner_RF_Best$predict(task_RF, row_ids = TestingIndex)$score(msr("surv.cindex")))
```

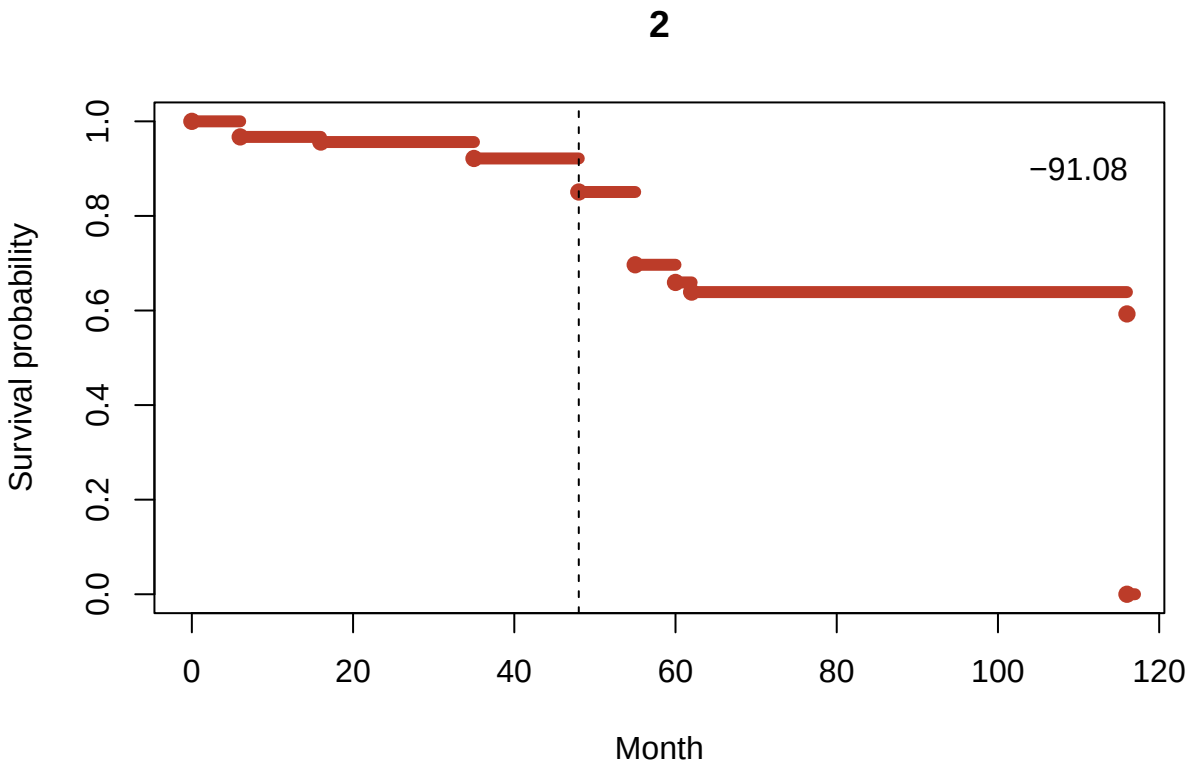
```
## surv.harrell_c
##      0.8494624
```

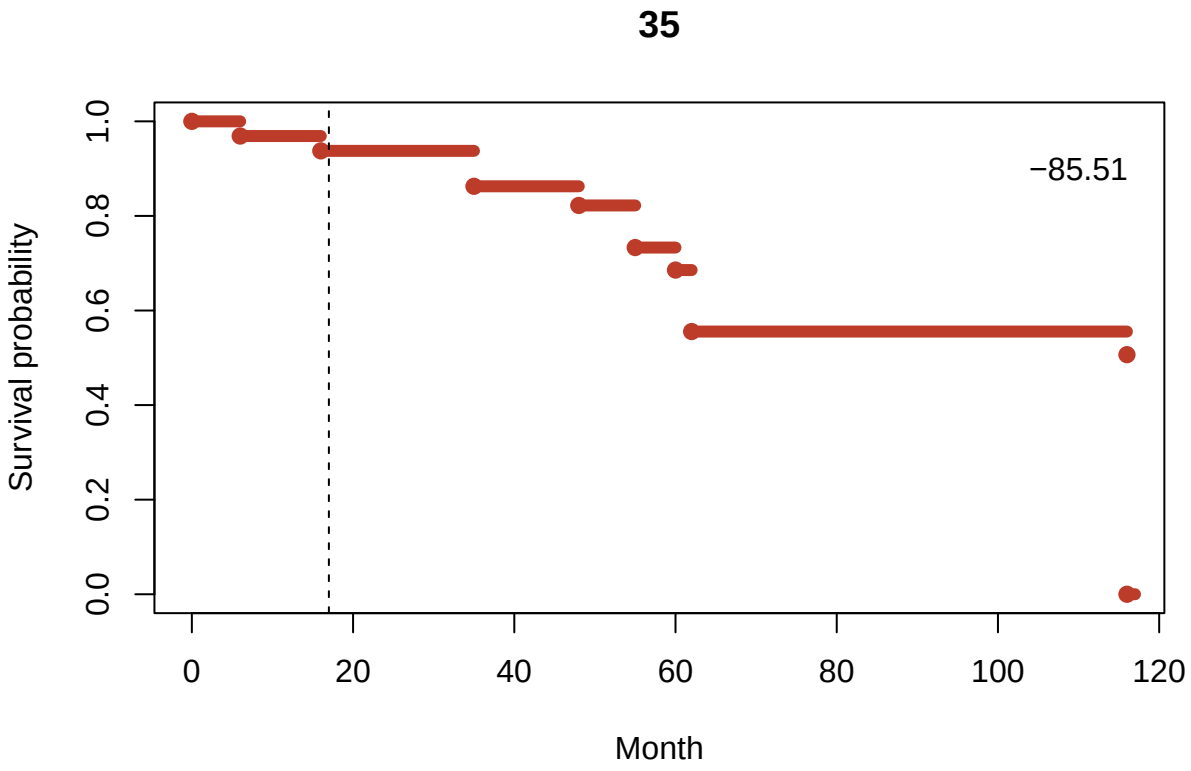
```
Crank <- data.frame(Index = TestingIndex, Recurrence = Label_PM$Recurrence[TestingIndex], Crank = Predi
library(survival)
library(survminer)
Class <- 1:35
Class[which(Crank$Crank >= Cutoff)] <- "HighRisk"
Class[which(Crank$Crank < Cutoff)] <- "LowRisk"
KMdata <- data.frame(Time = Label_PM$FUorRInterval[TestingIndex], Status = (as.numeric(Label_PM$Recurre
fit <- survfit(Surv(Time,Status) ~ Class, data = KMdata)
ggsurvplot(fit, data = KMdata, palette = c("#BC3C29FF", "#0072B5FF"), pval = TRUE, pval.method = T, lin
```

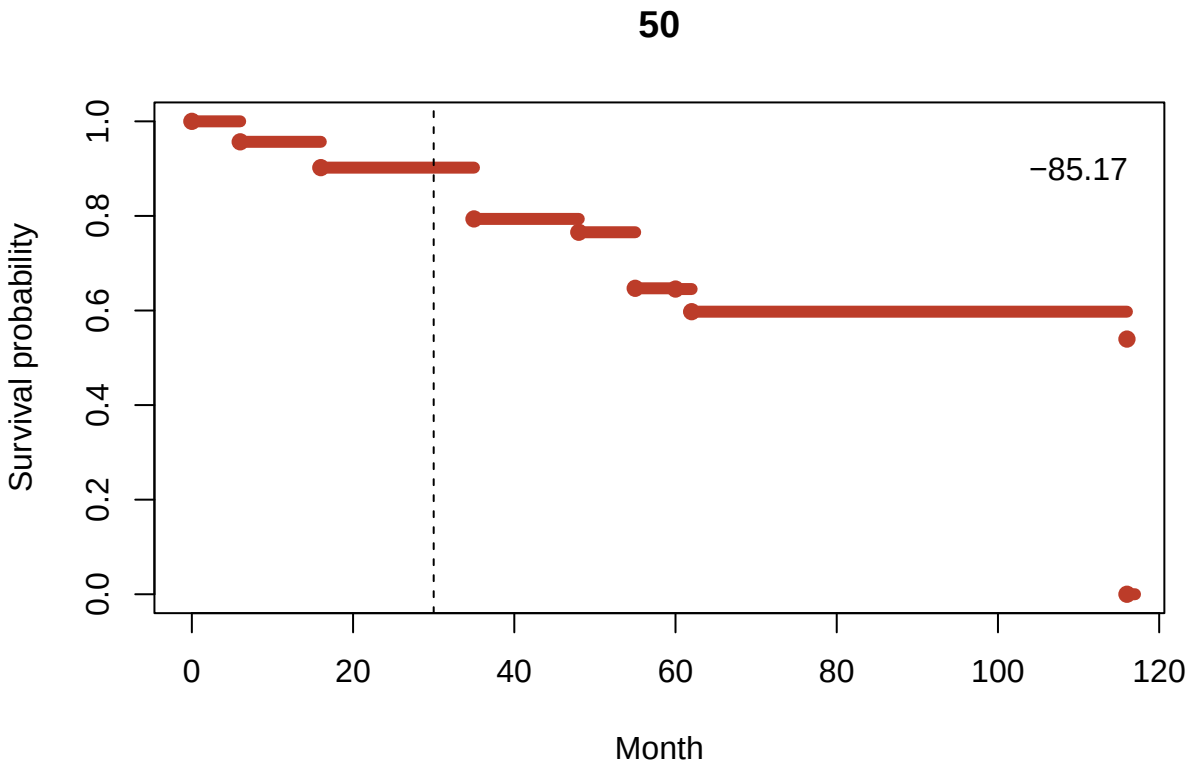


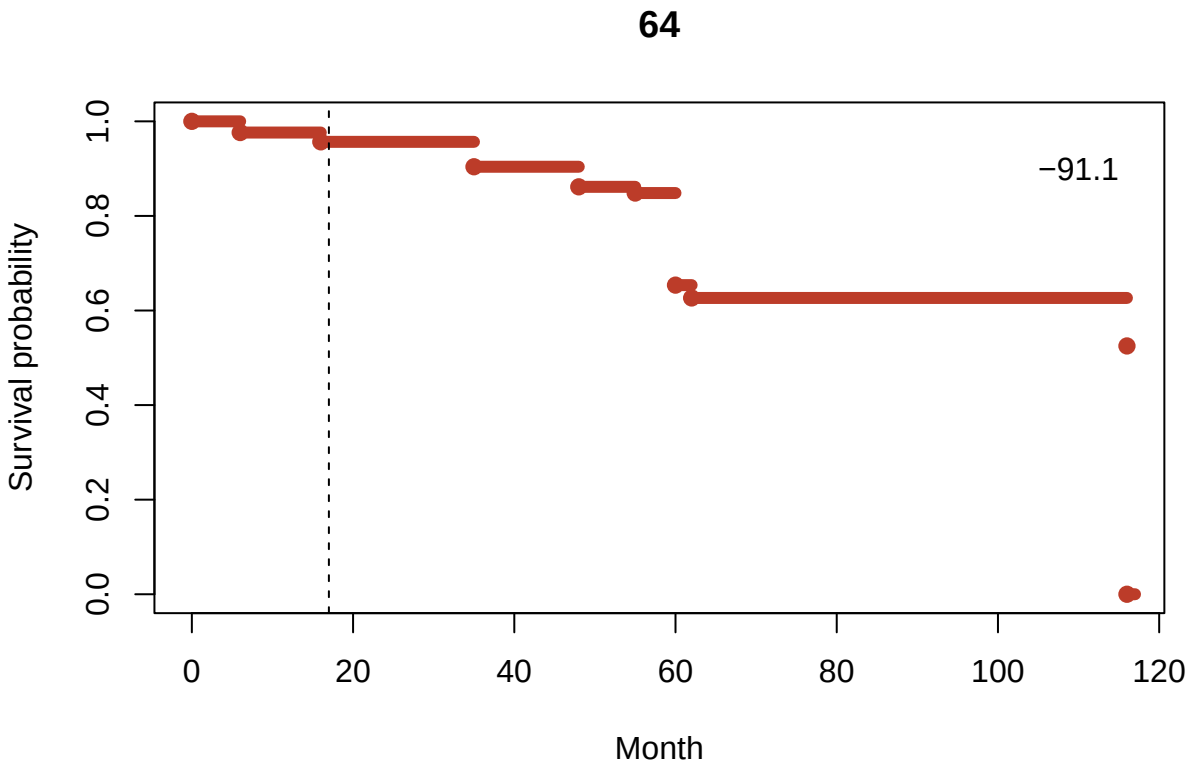

```
HighRiskIndex2 <- TestingIndex[which(Crank$Crank >= Cutoff)]
LowRiskIndex2 <- TestingIndex[which(Crank$Crank < Cutoff)]
```

```
#Survival Curve
HighRiskIndex <- sort(c(HighRiskIndex1, HighRiskIndex2))
LowRiskIndex <- sort(c(LowRiskIndex1, LowRiskIndex2))
library(distr6)
Prediction_RF <- learner_RF_Best$predict(task_RF, row_ids = 1:85)
PredictedSurvCurve <- Prediction_RF$distr
Crank <- Prediction_RF$crank
for (i in c(2,35,50,64)) {
  plot(PredictedSurvCurve, fun = "survival", ind = i, lty = 1, lwd = 6, main = i, xlab = "Month", ylab =
  text(x = 110, y = 0.9, labels = round(Crank[i], 2))
  if(Label_PM$Recurrence[i] == 1){
    abline(v = Label_PM$FUorRInterval[i])
  }
  if(Label_PM$Recurrence[i] == 0){
    abline(v = Label_PM$FUorRInterval[i], lty = 2)
  }
}
```

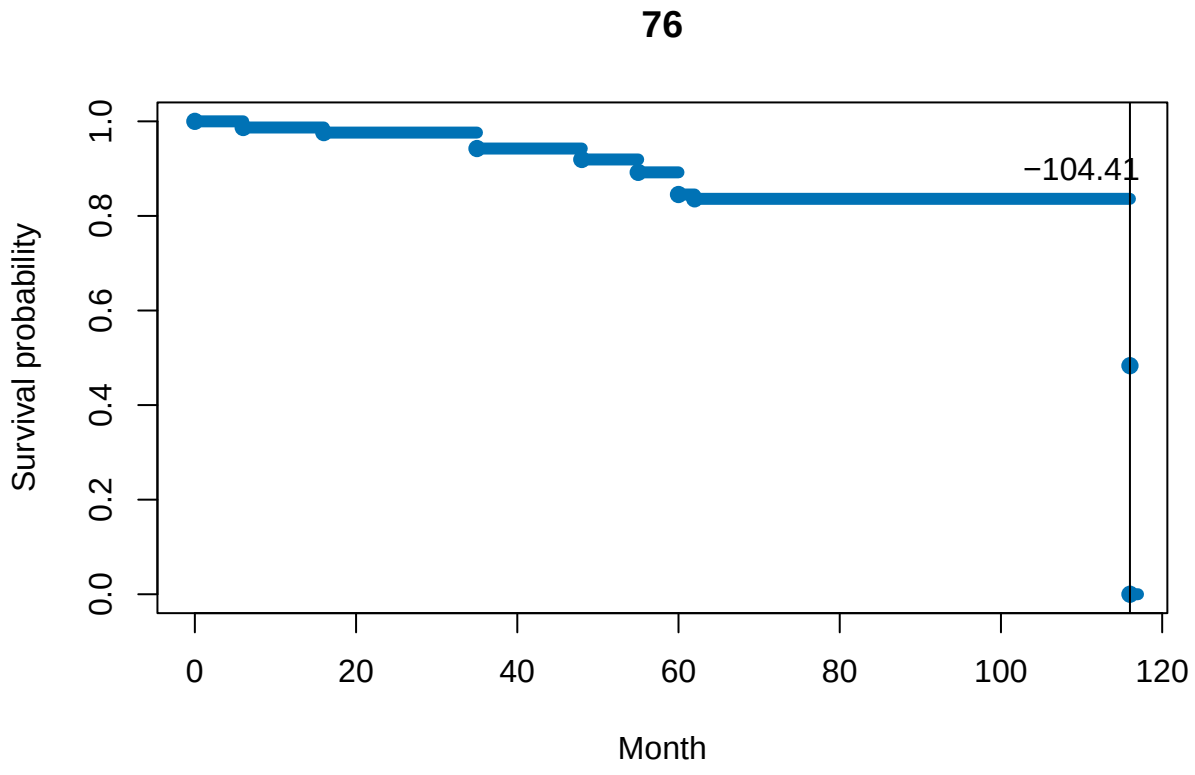

2
Survival probability
0.0
0.2
0.4
0.6
0.8
1.0
−91.08
0
20
40
60
80
100
120
Month


35
Survival probability
0.0
0.2
0.4
0.6
0.8
1.0
−85.51
0
20
40
60
80
100
120
Month

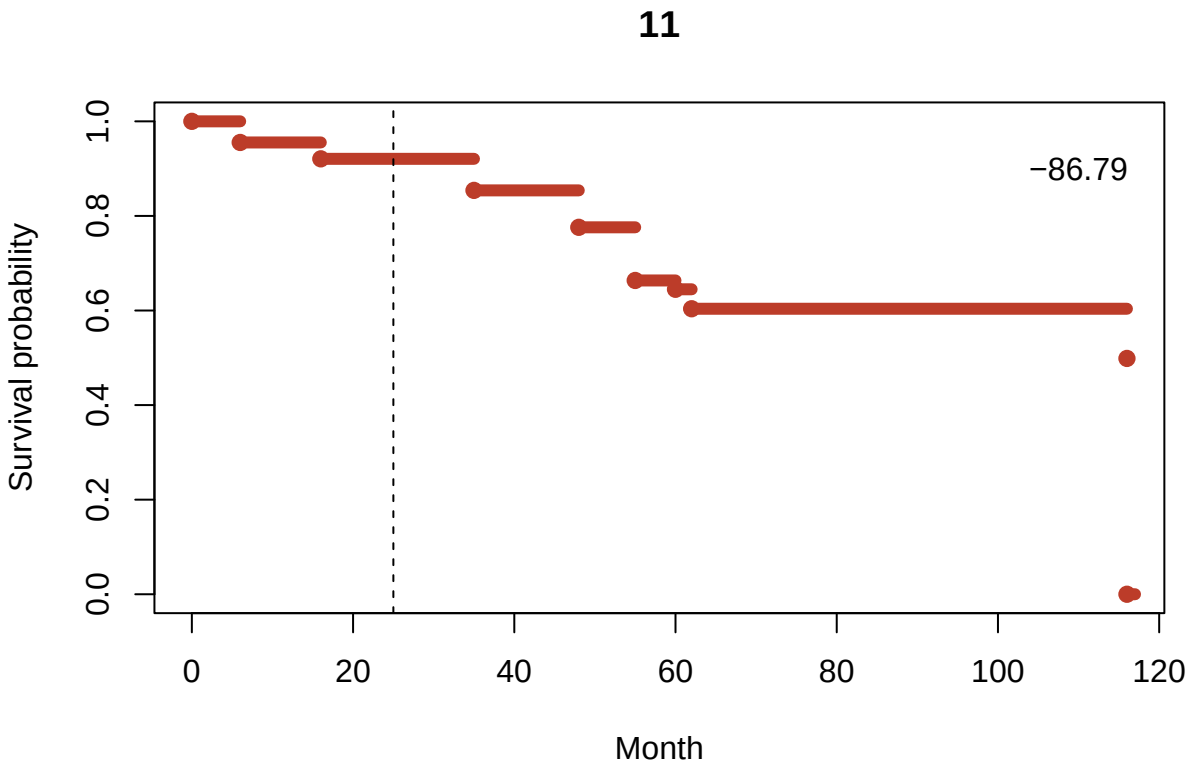

50
Survival probability
0.0
0.2
0.4
0.6
0.8
1.0
−85.17
0
20
40
60
80
100
120
Month

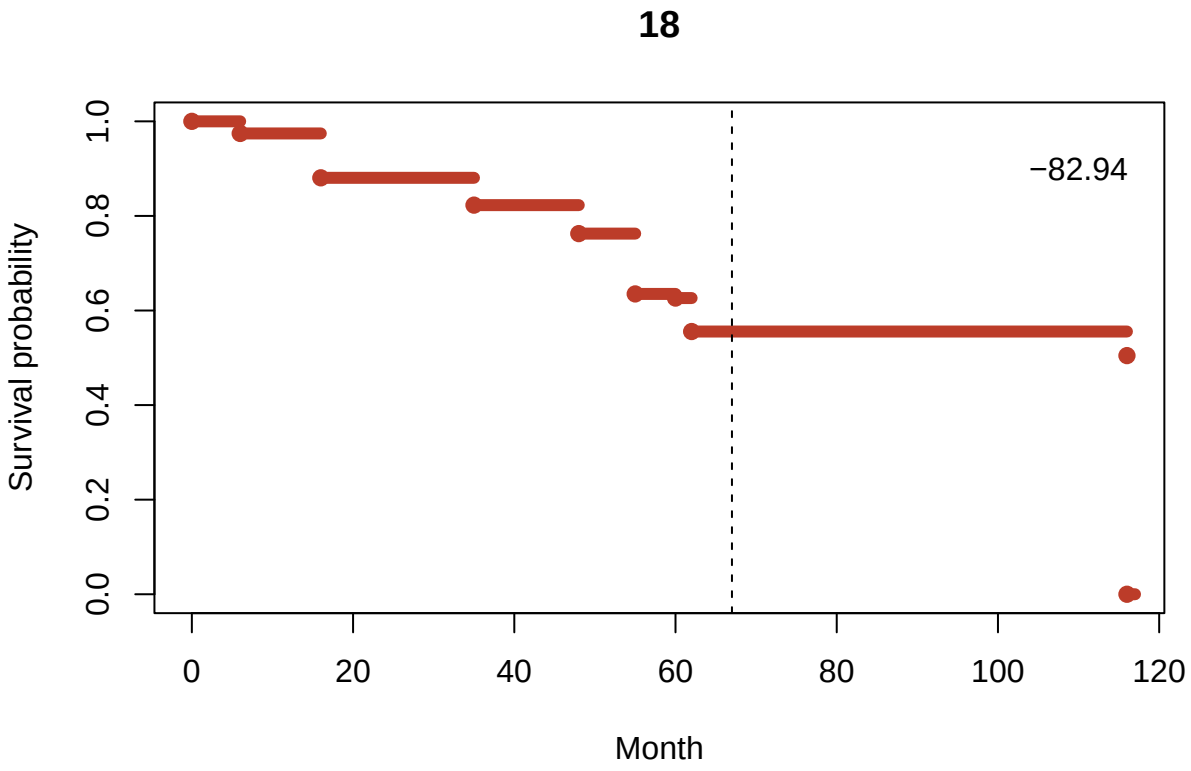


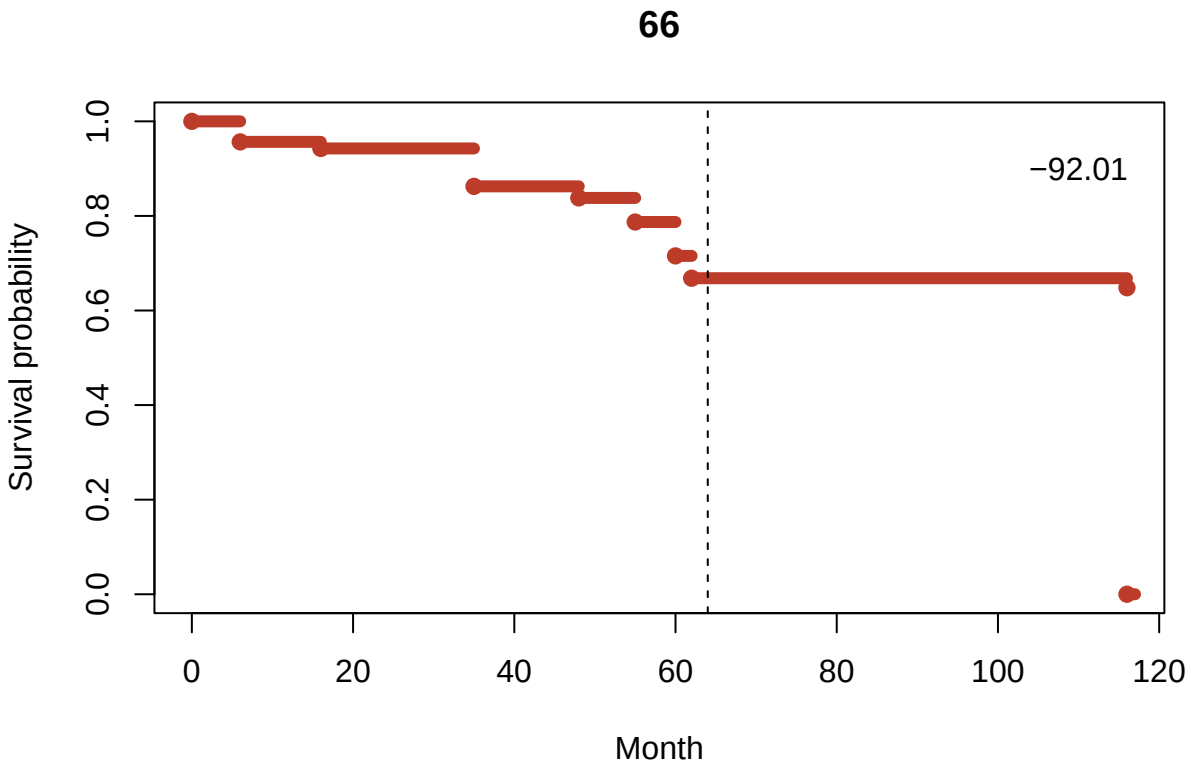

```
for (i in c(76)) {
  plot(PredictedSurvCurve, fun = "survival", ind = i, lty = 1, lwd = 6, main = i, xlab = "Month", ylab =
  text(x = 110, y = 0.9, labels = round(Crank[i], 2))
  if(Label_PM$Recurrence[i] == 1){
    abline(v = Label_PM$FUorRInterval[i])
  }
  if(Label_PM$Recurrence[i] == 0){
    abline(v = Label_PM$FUorRInterval[i], lty = 2)
  }
}
```

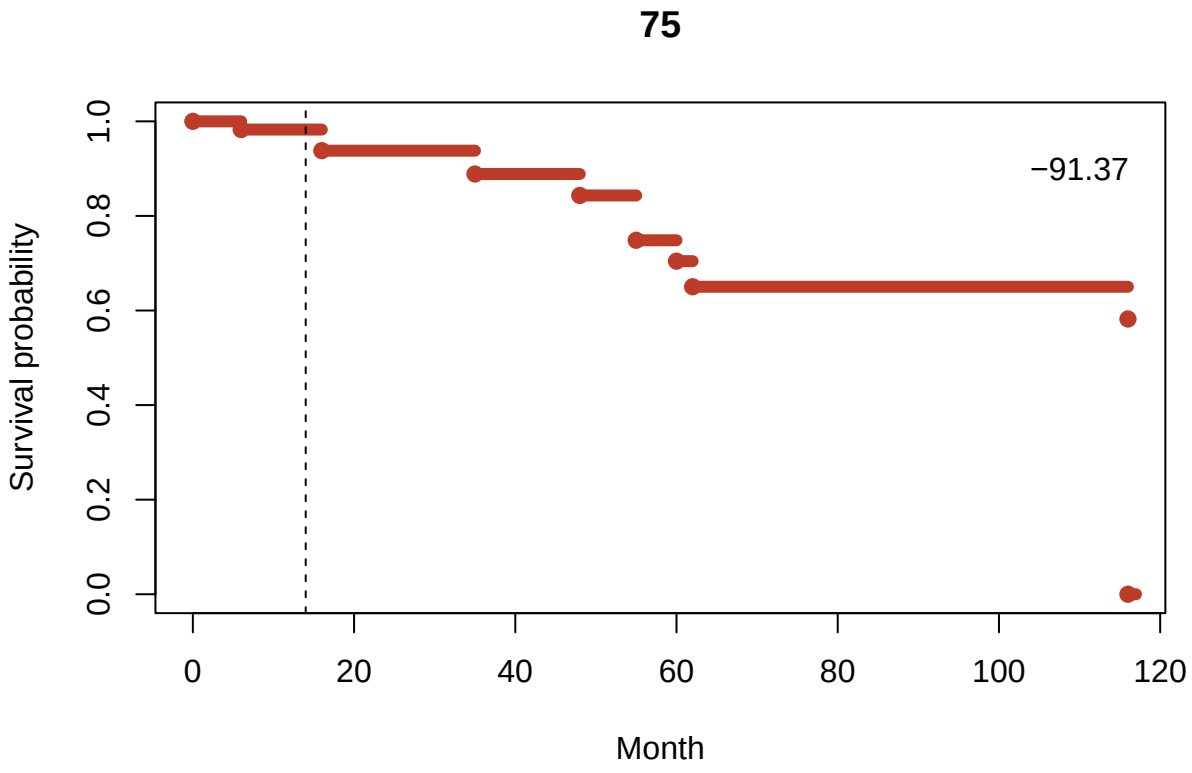



```
for (i in c(11,18,66,75)) {
  plot(PredictedSurvCurve, fun = "survival", ind = i, lty = 1, lwd = 6, main = i, xlab = "Month", ylab =
  text(x = 110, y = 0.9, labels = round(Crank[i], 2))
  if(Label_PM$Recurrence[i] == 1){
    abline(v = Label_PM$FUorRInterval[i])
  }
  if(Label_PM$Recurrence[i] == 0){
    abline(v = Label_PM$FUorRInterval[i], lty = 2)
  }
}
```

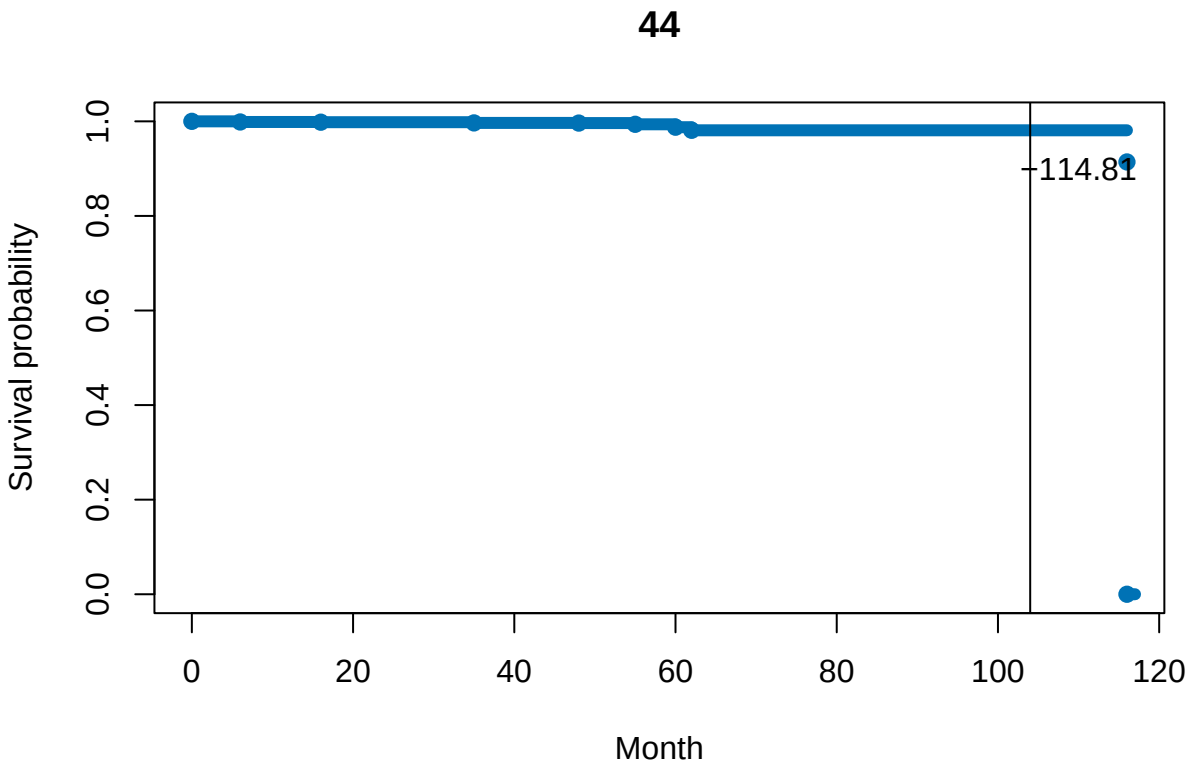

11
Survival probability
0.0
0.2
0.4
0.6
0.8
1.0
−86.79
0
20
40
60
80
100
120
Month


18
Survival probability
0.0
0.2
0.4
0.6
0.8
1.0
−82.94
0
20
40
60
80
100
120
Month


66
Survival probability
0.0
0.2
0.4
0.6
0.8
1.0
−92.01
0
20
40
60
80
100
120
Month




```
for (i in c(44)) {
  plot(PredictedSurvCurve, fun = "survival", ind = i, lty = 1, lwd = 6, main = i,  xlab = "Month", ylab
  text(x = 110, y = 0.9, labels = round(Crank[i], 2))
  if(Label_PM$Recurrence[i] == 1){
    abline(v = Label_PM$FUorRInterval[i])
  }
  if(Label_PM$Recurrence[i] == 0){
    abline(v = Label_PM$FUorRInterval[i], lty = 2)
  }
}
```


44
Survival probability
0.0
0.2
0.4
0.6
0.8
1.0
114.81
0
20
40
60
80
100
120
Month