Análise de sobrevivência em pacientes com mieloma múltiplo

SME0821 - Análise de Sobrevivência - Atividade I

Francisco Rosa Dias de Miranda - 4402962 Heitor Carvalho Pinheiro - 11833351
Lua Nardi Quito - 11371270 Vitor Pinho Iecks Ponce - 10785968
Gusthavo Henrique Parra da Silva - 7086506 Felipe Tadaki T. Ida - 11027629

abril 2022

Contents

1) Introdução 2
2) Metodologia 4
3) Análise de dados 5
K-M Algoritmo 5
Algoritmo Nelson-Aalen 15
Tábua Atuarial 16
4) Conclusão 17
5) Bibliografia 17

1) Introdução

O mieloma múltiplo é o câncer que afeta aos plasmócitos, células da medula óssea responsáveis pela produção de anticorpos. Nos indivíduos acometidos, os plasmócitos são anormais e se multiplicam rapidamente, comprometendo a produção das outras células do sangue.

Foram obtidas medidas de expressão gênica em indivíduos com mieloma múltiplo, a partir de bases disponíveis no GEO (Id: GSE4581), um repositório de dados genômicos públicos do NCBI (National Center for Biotechnology Information). Nesse estudo, foram coletados dados de uma amostra de 256 pacientes, consistindo nas 11 colunas descritas abaixo:

Variável	Descrição
`molecular_group`	Subgrupos moleculares dos pacientes
`chr1q21_status`	Status de amplificação do cromossomo 1q21
`treatment`	Todos os pacientes receberam o tratamento TT2
`event`	Status de sobrevivência, 0 = vivo, 1 = morto
`time`	Tempo de sobrevivência, em meses
`CCND1`, `CRIM1`, `DEPDC1`, `IRF4` `TP53`, `WHSC1`	Nível de expressão dos respectivos genes

O conjunto de dados também encontra-se disponível no R através do comando `survminer::myeloma`.

```
library(tidyverse)
library(survival)
library(survminer)
library(pander)
library(biostat3)
```

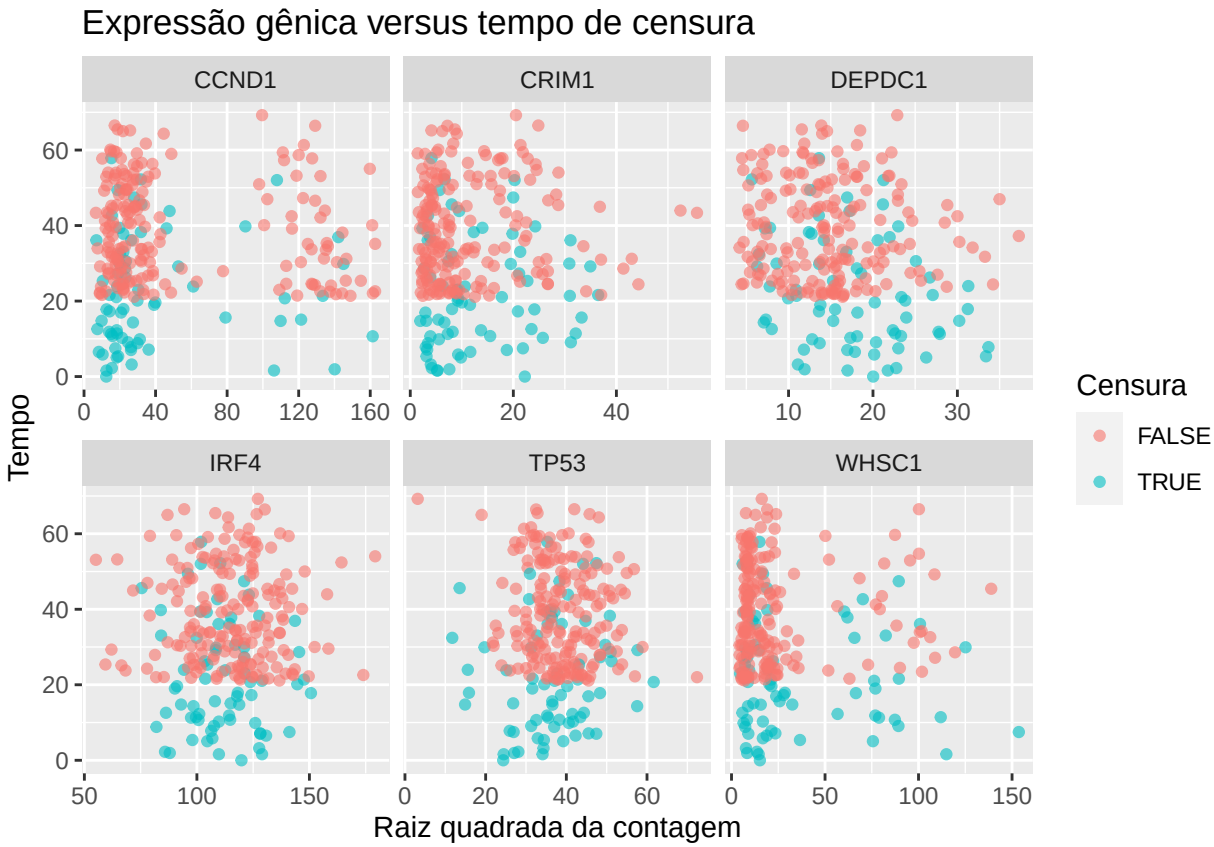
```
# Conjunto de dados utilizado
df <- survminer::myeloma %>% rownames_to_column %>% tibble
# Substituindo o indicador de censura por um tipo booleano
df$event <- as.logical(df$event)
head(df)
```

```
## # A tibble: 6 x 12
##   rowname  molecular_group chr1q21_status treatment event  time  CCND1 CRIM1
##   <chr>    <fct>           <fct>          <fct>     <lgl> <dbl>  <dbl> <dbl>
## 1 GSM50986 Cyclin D-1      3 copies       TT2       FALSE  69.2  9908. 421.
## 2 GSM50988 Cyclin D-2      2 copies       TT2       FALSE  66.4 16699.  52
## 3 GSM50989 MMSET           2 copies       TT2       FALSE  66.5   294. 618.
## 4 GSM50990 MMSET           3 copies       TT2       TRUE   42.7   242.  11.9
## 5 GSM50991 MAF             <NA>           TT2       FALSE  65     473.  38.8
## 6 GSM50992 Hyperdiploid    2 copies       TT2       FALSE  65.2   664.  16.9
## # ... with 4 more variables: DEPDC1 <dbl>, IRF4 <dbl>, TP53 <dbl>, WHSC1 <dbl>
```

```
# tabela descritivas da variável resposta
pander(summary(df$time))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0	23.7	33.05	34.43	45.47	69.24

```
# expressoes genicas que iremos analisar
gex_cols <- c("CCND1", "CRIM1", "DEPDC1","IRF4","TP53","WHSC1")
# grafico de tempo versus raiz quadrada da contagem
df %>% pivot_longer(cols = all_of(gex_cols)) %>%
ggplot(aes(y=time, x=sqrt(value))) +
  geom_point( aes(color=event), alpha=0.6) +
  facet_wrap(~name, scales = "free_x") +
  labs(x = "Raiz quadrada da contagem",
       y = "Tempo",
       color = "Censura",
       title = "Expressão gênica versus tempo de censura")
```

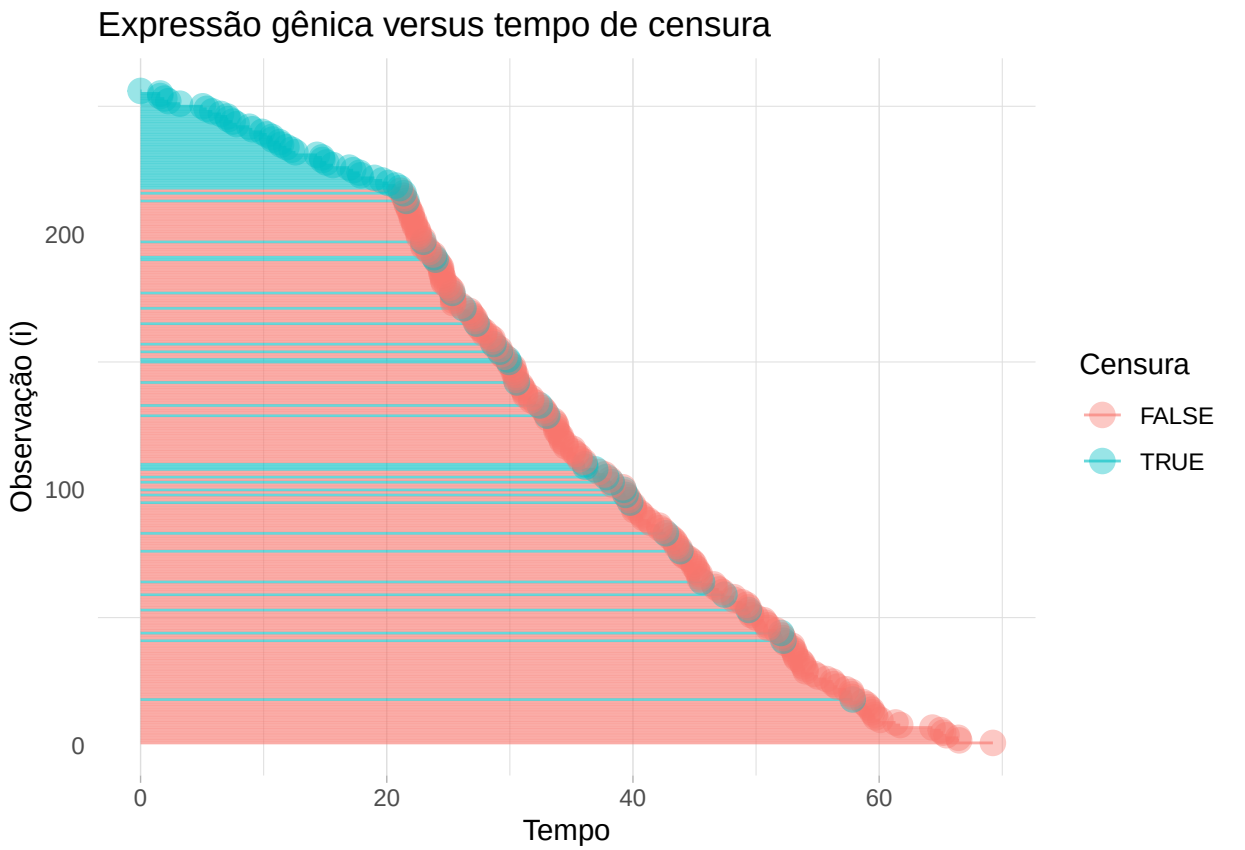


Foram plotadas em cada gráfico, as covariáveis de expressões gênicas em relação ao tempo de censura e a sua respectiva expressão gênica, no eixo X temos a raiz quadrada da contagem de pacientes e no eixo Y é o tempo (em meses).

Observa-se que não há aparente relação linear entre o tempo até o evento e a contagem do evento de interesse

```
## trocar por tempo vs raiz quadrada da contagem
  df %>% arrange(desc(time)) %>%
ggplot(aes(x=1:nrow(df), y=time)) +
  geom_segment( aes(x=1:nrow(df), xend=1:nrow(df), y=0, yend=time, color = event), alpha = 0.6) +
  geom_point( aes(color=event), size=4, alpha=0.4) +
  theme_light() +
  coord_flip() +
  theme(
    panel.grid.major.y = element_blank(),
    panel.border = element_blank(),
```

```
    axis.ticks.y = element_blank(),
  ) +
    labs(x = "Observação (i)",
       y = "Tempo",
       color = "Censura",
       title = "Expressão gênica versus tempo de censura")
```



Podemos observar que as censuras presentes nesse conjunto de dados são do tipo aleatória e a direita. Além disso, o tratamento utilizado nesses pacientes foi sempre o mesmo (TT2), dessa forma, não havendo um grupo de controle.

2) Metodologia

Nesse trabalho, nosso objetivo é a análise de dados de sobrevivência com censura a direita a partir de uma abordagem não-paramétrica, em que o interesse é identificar fatores de prognóstico para o mioma múltiplo a partir da amostra coletada.

Vamos usar os metodos K-M, tabela atuarial e Nelson-Aalen para analisar os dados nesse estudo.

Kaplan-Meier: O estimador de Kaplan–Meier, também conhecido como estimador do limite de produto, é uma estatística não paramétrica usada para estimar a função de sobrevivência a partir de dados de sobrevivência. Na pesquisa médica, muitas vezes é usado para medir a fração de pacientes que vivem por um determinado período de tempo após o tratamento. Em outros campos, os estimadores de Kaplan-Meier podem ser usados para medir o tempo que as pessoas permanecem desempregadas após uma perda de emprego ou o tempo até a falha de peças de máquinas.

Tabela Atuarial:

TODO: descrever os metodos utilizados

3) Análise de dados

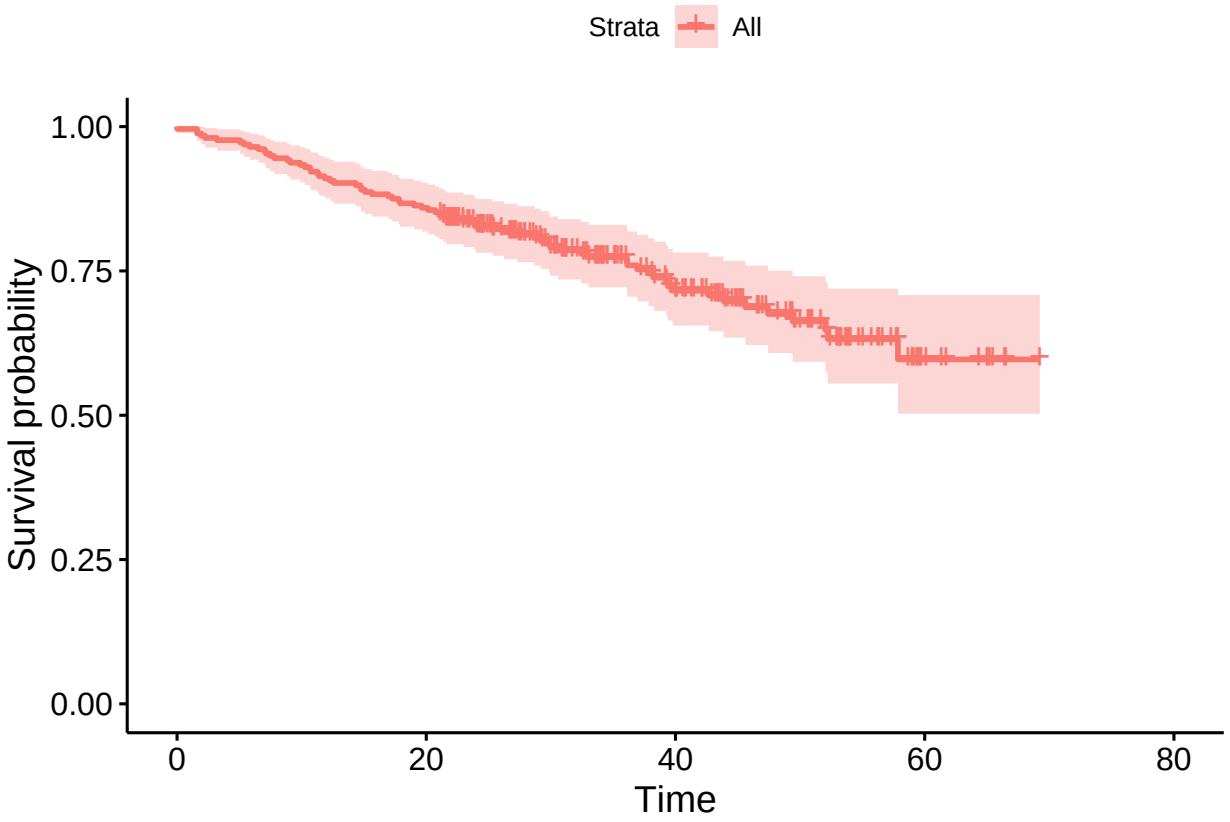
Inicialmente, realizaremos a estimativa da Curva de Sobrevivência dos pacientes com Mieloma, utilizando o estimador de Kaplan-Meier.

K-M Algoritmo

```
km_fit <- survfit(Surv(time, event) ~ 1, data = df)
```

Definida a função de sobrevivência, podemos verificar a curva de sobrevivência considerando todas as covariáveis.

```
ggsurvplot(km_fit, data = df, risk.table = FALSE)
```



Podemos verificar os resultados da estimação completa, usando `summary()`

```
summary(km_fit, times = seq(0,70,5))
```

```
## Call: survfit(formula = Surv(time, event) ~ 1, data = df)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     0    256       1    0.996 0.00390        0.988        1.000
##     5    250       5    0.977 0.00946        0.958        0.995
##    10    239      11    0.934 0.01556        0.904        0.965
##    15    228      11    0.891 0.01951        0.853        0.930
##    20    220       8    0.859 0.02173        0.818        0.903
##    25    179       8    0.827 0.02378        0.781        0.875
##    30    149       7    0.791 0.02628        0.741        0.845
##    35    116       3    0.774 0.02758        0.722        0.830
##    40     94       8    0.716 0.03225        0.655        0.782
```

```
##     45     69       2     0.698 0.03385        0.635        0.767
##     50     50       3     0.663 0.03780        0.592        0.741
##     55     27       2     0.632 0.04187        0.555        0.719
##     60     10       1     0.597 0.05222        0.503        0.708
##     65      6       0     0.597 0.05222        0.503        0.708
```

De início, percebemos que não há tempo mediano, uma vez que as observações se encerram antes de serem obtidas probabilidades de sobrevivência de 50%.

Sendo assim, os pacientes em estudo apresentariam uma probabilidade de sobrevivência de 60% após cinco anos.

Determinando o *Cutpoint* para cada expressão gênica Temos diferentes níveis de expressão para os genes CRIM1", "DEPDC1", "WHSC1", "CCND1", "IRF4" e "TP53". Entretanto, a categorização desses valores nos auxilia na comparação entre as variáveis. O R nos permite estimar os *cutpoints* (Pontos de corte) ideais, para cada variável numérica, permitindo reduzir os diversos valores a duas categorias: "high" e "low".

O teste utilizado é o ***Maximally Selected Rank statistics***, que assume que um valor desconhecido de X, determina dois grupos distintos em Y. No nosso, caso, o teste busca encontrar o valor númerico do nível da expressão gênica que melhor separa os valores em dois grupos distintos.

```
res.cut <- surv_cutpoint(df, time = "time", event = "event",
                         variables = gex_cols,
                         progressbar = FALSE)
pander(summary(res.cut))
```

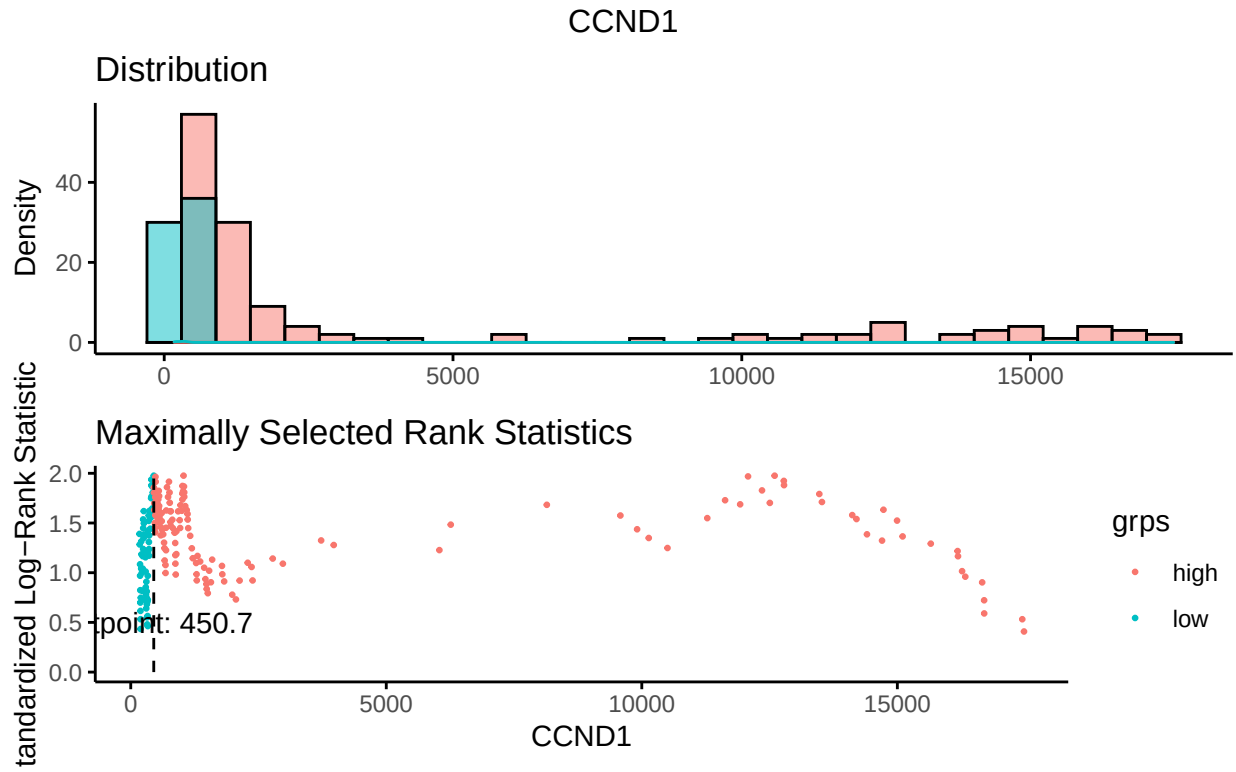
Cutpoint ótimo de cada expressão

	cutpoint	statistic
CCND1	450.7	1.976
CRIM1	82.3	1.968
DEPDC1	279.8	4.275
IRF4	12053	2.178
TP53	748.3	2.929
WHSC1	3206	3.361

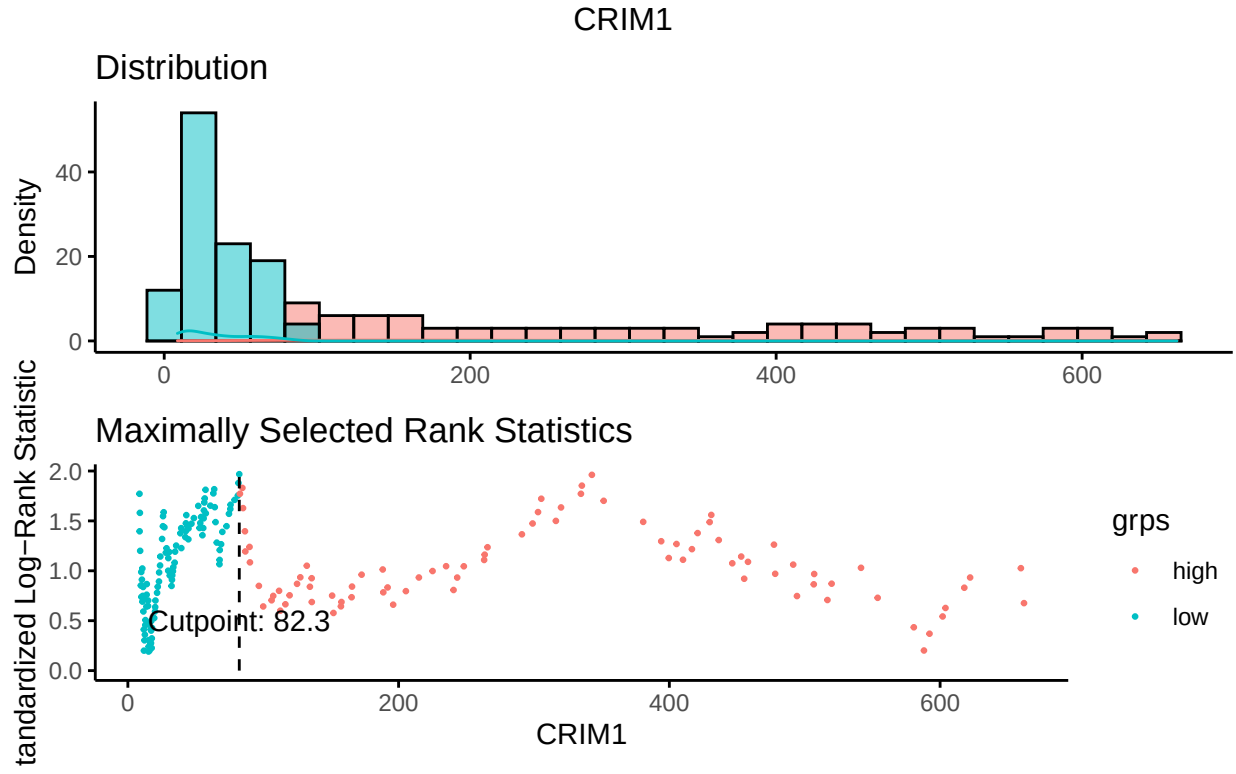
```
plot(res.cut, gex_cols, pallete = "npg")
```

Gráfico para cada "Cutpoint"

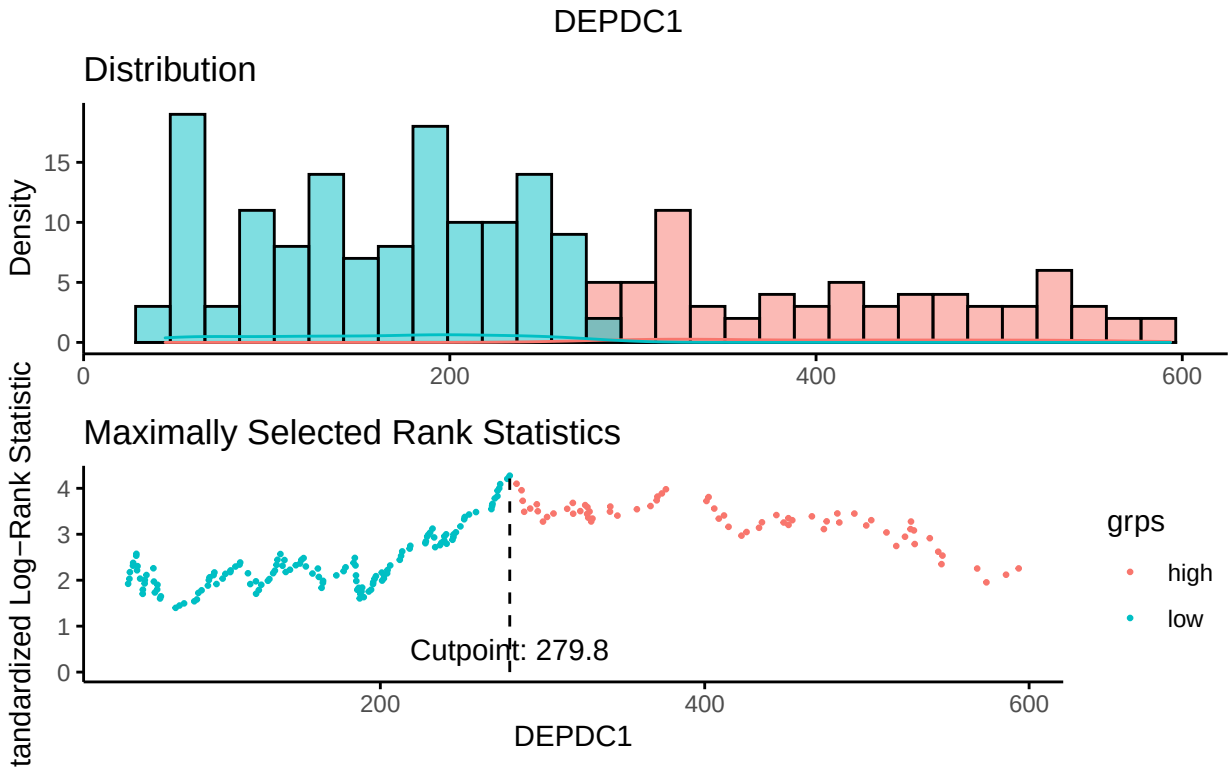
```
## $CCND1
```



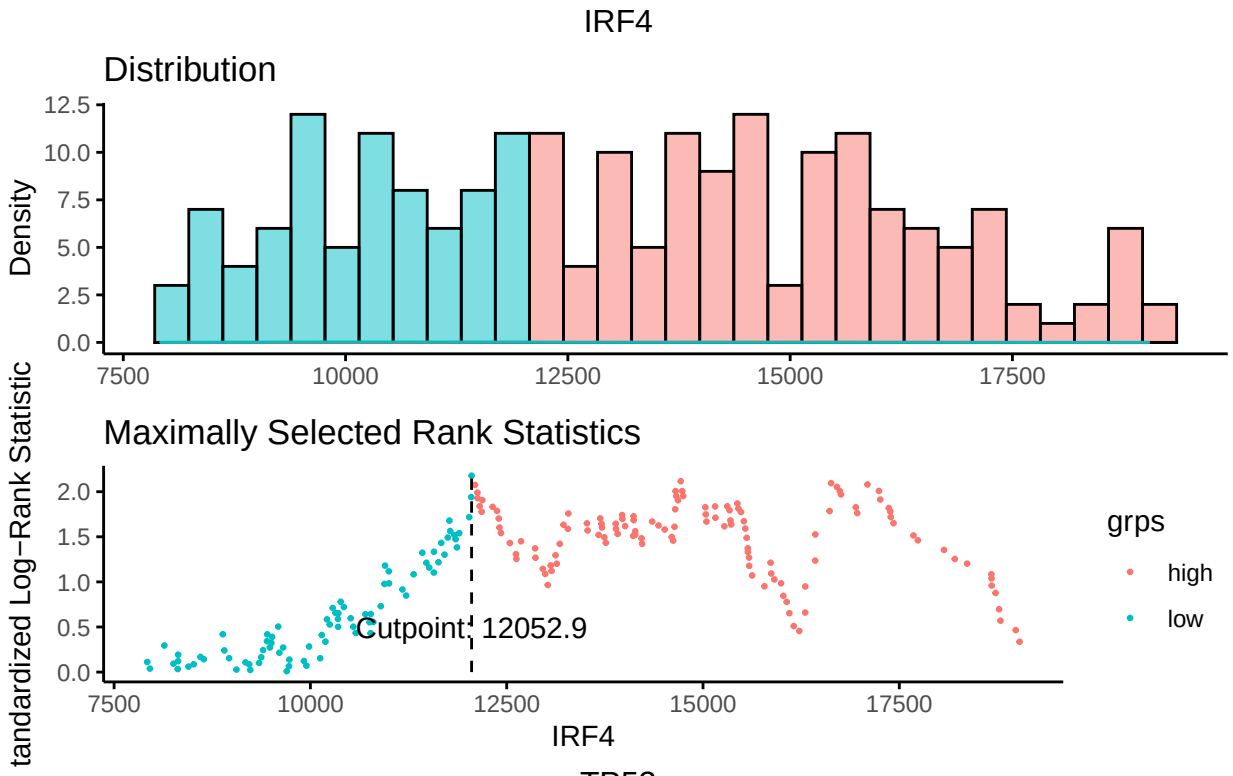

```
##
## $CRIM1
```

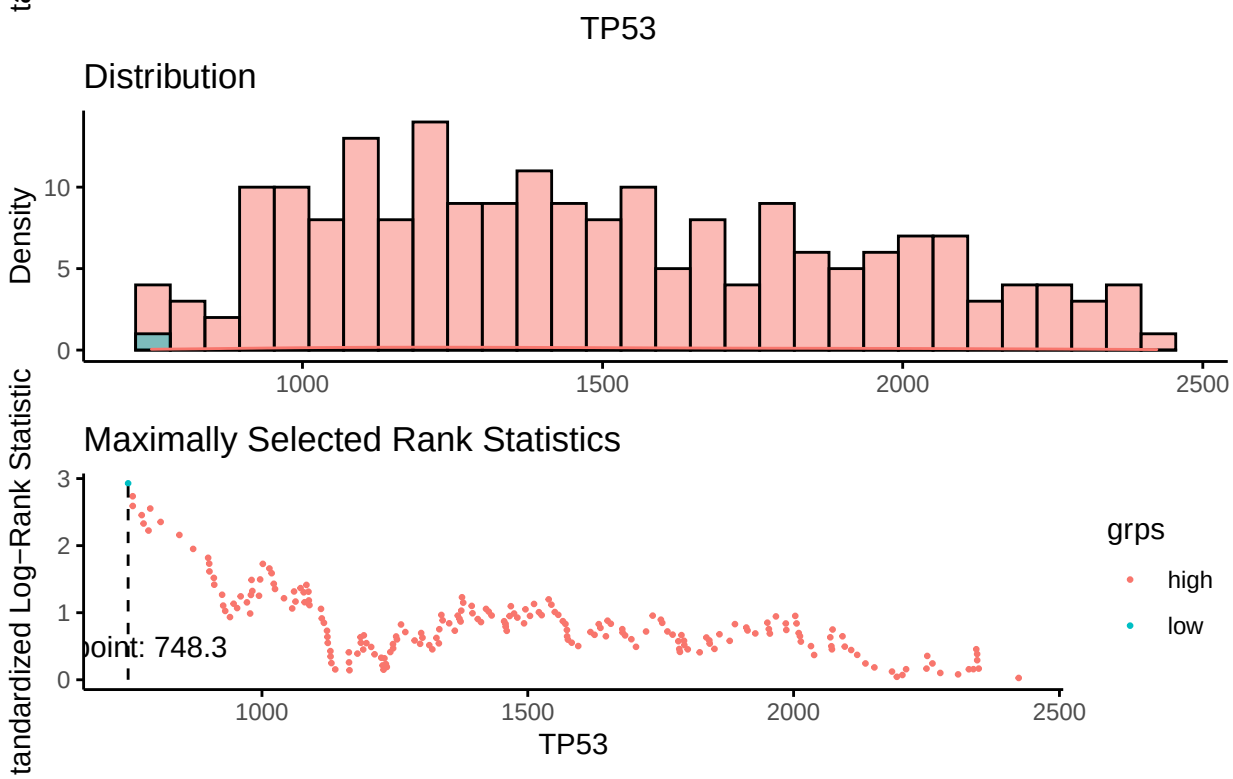



```
##
## $DEPDC1
```

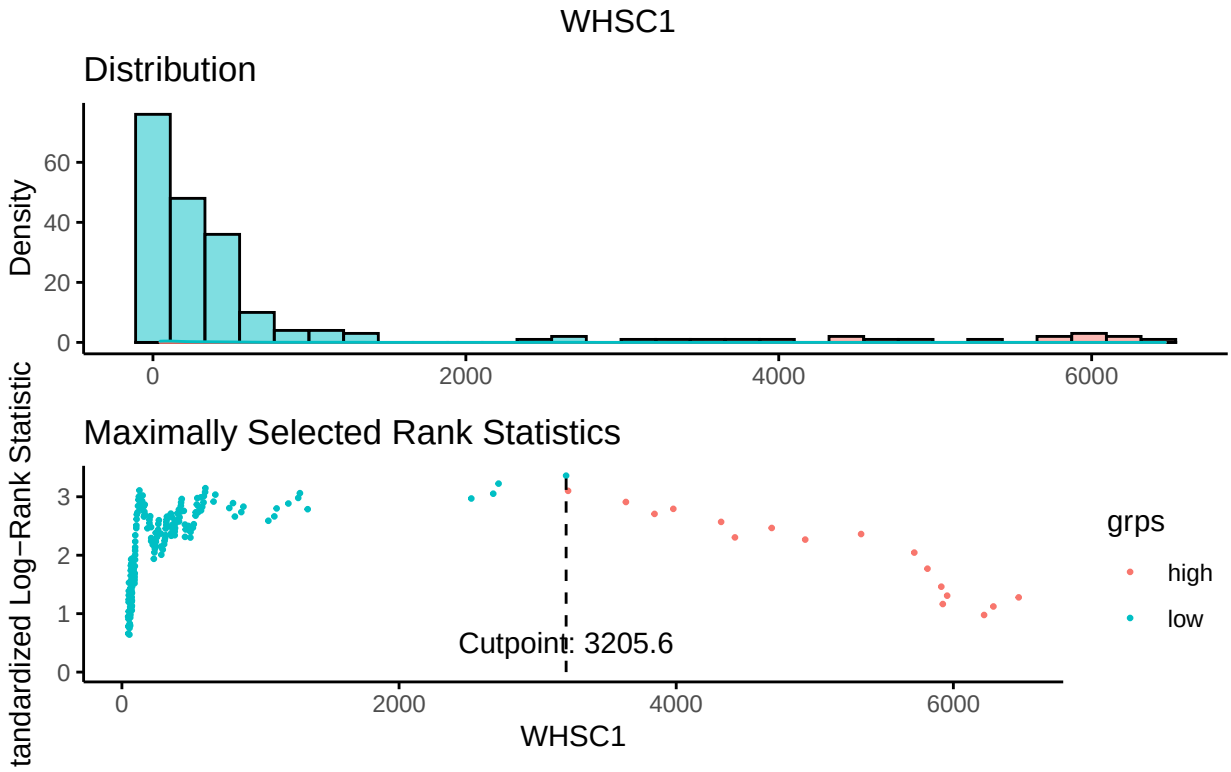



```
##
## $IRF4

##
## $TP53

## Warning: Groups with fewer than two data points have been dropped.

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
```


IRF4
Distribution
Density
12.5
10.0
7.5
5.0
2.5
0.0
7500
10000
12500
15000
17500
Maximally Selected Rank Statistics
tandardized Log−Rank Statistic
2.0
1.5
1.0
0.5
0.0
Cutpoint: 12052.9
grps
high
low
IRF4


TP53
Distribution
Density
10
5
0
1000
1500
2000
2500
Maximally Selected Rank Statistics
tandardized Log−Rank Statistic
3
2
1
0
oint: 748.3
grps
high
low
TP53

```
##
## $WHSC1
```



Definidos os melhores pontos que dividem os valores das expressões gênicas em dois grupos distintos, podemos enfim, categorizar nossas variáveis em dois grupos: ***high*** e ***low***.

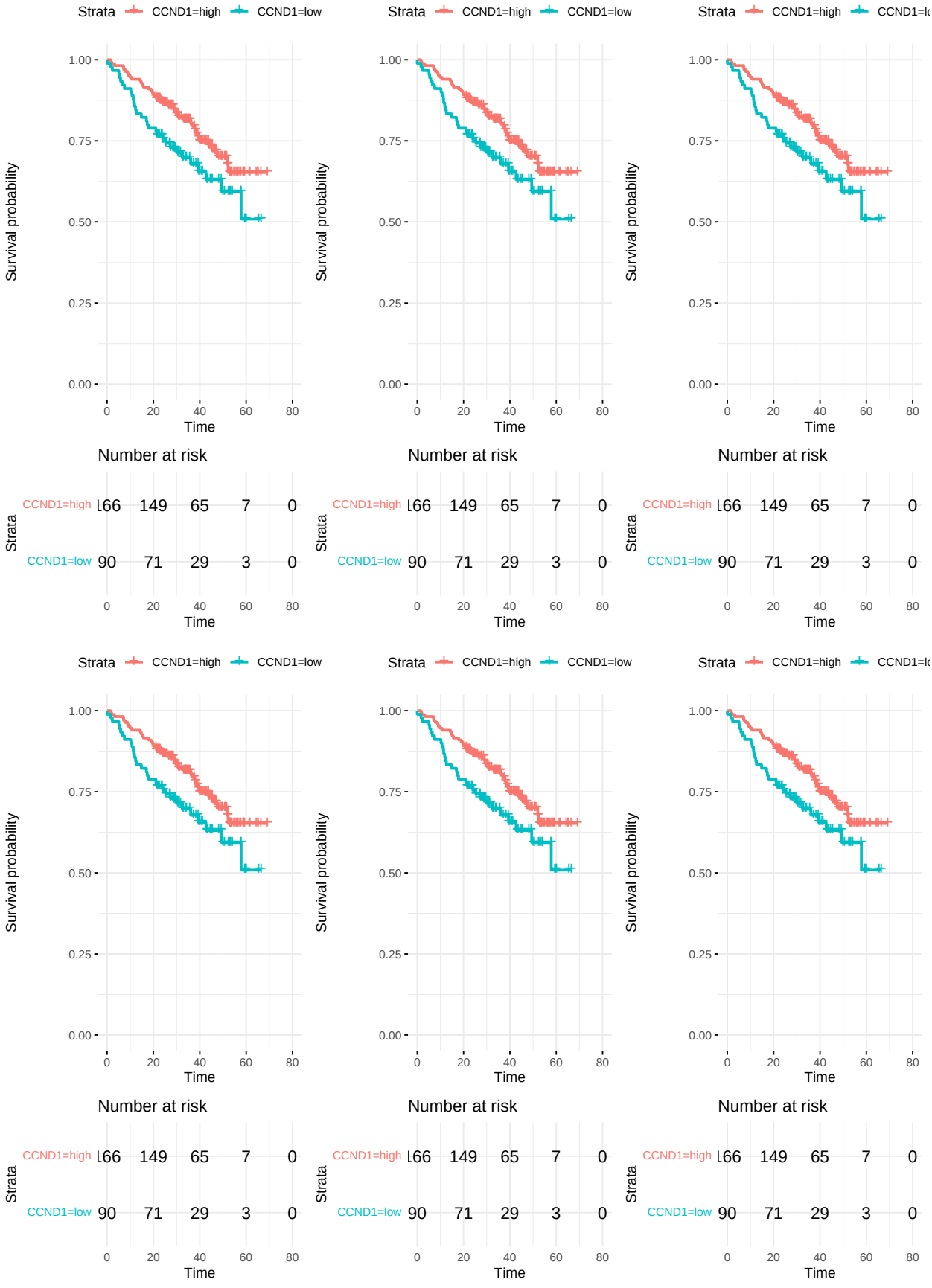
```
res.cat <- surv_categorize(res.cut)
pander(head(res.cat))
```

time	event	CCND1	CRIM1	DEPDC1	IRF4	TP53	WHSC1
69.24	FALSE	high	high	high	high	low	low
66.43	FALSE	high	low	low	high	high	low
66.5	FALSE	low	high	low	low	high	high
42.67	TRUE	low	low	low	low	high	high
65	FALSE	high	low	low	low	low	low
65.2	FALSE	high	low	high	high	high	low

Curvas de sobrevivência para cada expressão gênica Curvas de sobrevivência para cada expressão gênica, considerando os níveis *low* e *high*.

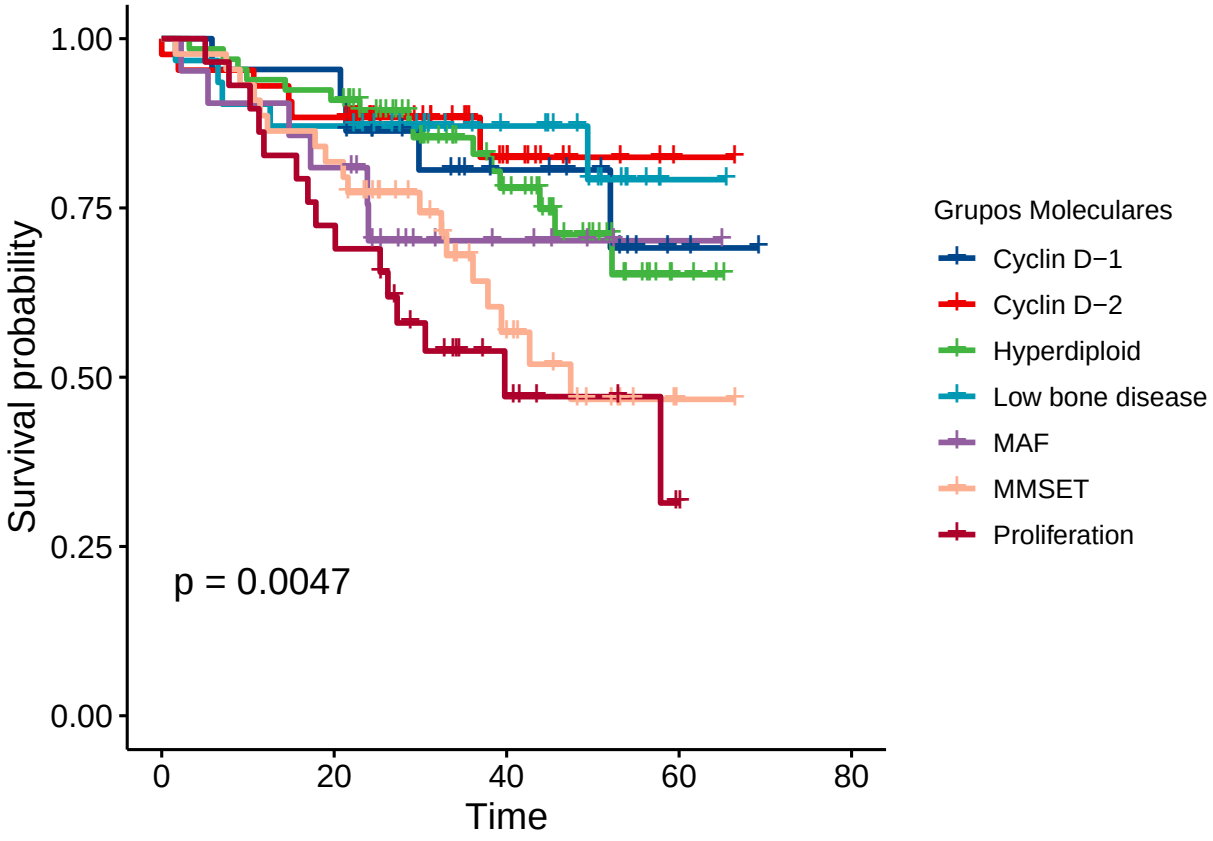
```
#defyning each fit for each gene
fit1 <- survfit(Surv(time, event) ~ CCND1, data = res.cat)
fit2 <- survfit(Surv(time, event) ~ CRIM1, data = res.cat)
fit3 <- survfit(Surv(time, event) ~ DEPDC1, data = res.cat)
fit4 <- survfit(Surv(time, event) ~ IRF4, data = res.cat)
fit5 <- survfit(Surv(time, event) ~ TP53, data = res.cat)
fit6 <- survfit(Surv(time, event) ~ WHSC1, data = res.cat)

#List of ggsurvplots

fit_list <- list(fit1,fit2,fit3,fit4,fit5,fit6)
```

```
splots <- fit_list %>% map(~ggsurvplot(fit1, data = df,
                                       risk.table = TRUE,
                                       risk.table.height = 0.3,
                                       ggtheme = theme_minimal()))

#arrange multiple ggsurvplots
arrange_ggsurvplots(splots, print = TRUE,
                    ncol = 3, nrow = 2)
```


Strata
CCND1=high
CCND1=low
Survival probability
1.00
0.75
0.50
0.25
0.00
0
20
40
60
80
Time
Number at risk
CCND1=high 166 149 65 7 0
CCND1=low 90 71 29 3 0

```r
# Survival curves with global p-value
fit2 <- survfit(Surv(time, event) ~ molecular_group, data = df)
ggsurvplot(fit2, data = myeloma,
           legend.title = "Grupos Moleculares",
           legend.labs = levels(myeloma$molecular_group),
           legend = "right",
           pval = TRUE, palette = "lancet")
```

Curvas de sobrevivência para os diferentes grupos moleculares




```r
# summary(fit2)
```

O gráfico acima expõe todas as curvas de sobrevivência para os diferentes grupos moleculares.

Em nosso teste de hipótese temos duas hipóteses possíveis:

H_0 : Não há diferença entre as curvas de sobrevivência para os diferentes grupos moleculares. H_1 Há diferença entre as curvas de sobrevivência para os diferentes grupos moleculares.

Em nosso teste, utilizamos por padrão um $\alpha = 0.05$, ou seja, nosso Intervalo de Confiança é de 95%.

O valor-p resposta é um valor global que apenas nos indica se há alguma diferença entre as curvas de sobrevivência. Como $p = 0.047 < 0.05$, podemos rejeitar H_0 e concluir que existe uma diferença entre os grupos moleculares.

Podemos realizar um Log-rank teste pareado entre os diferentes grupos moleculares, a fim de identificar quais grupos apresentam diferenças significativas de risco de morte.

```r
  fit <- survfit(Surv(time, event) ~ chr1q21_status, data = df)
# Pairwise survdiff
```

```
res <- pairwise_survdiff(Surv(time, event) ~ molecular_group,
     data = myeloma)
res
```

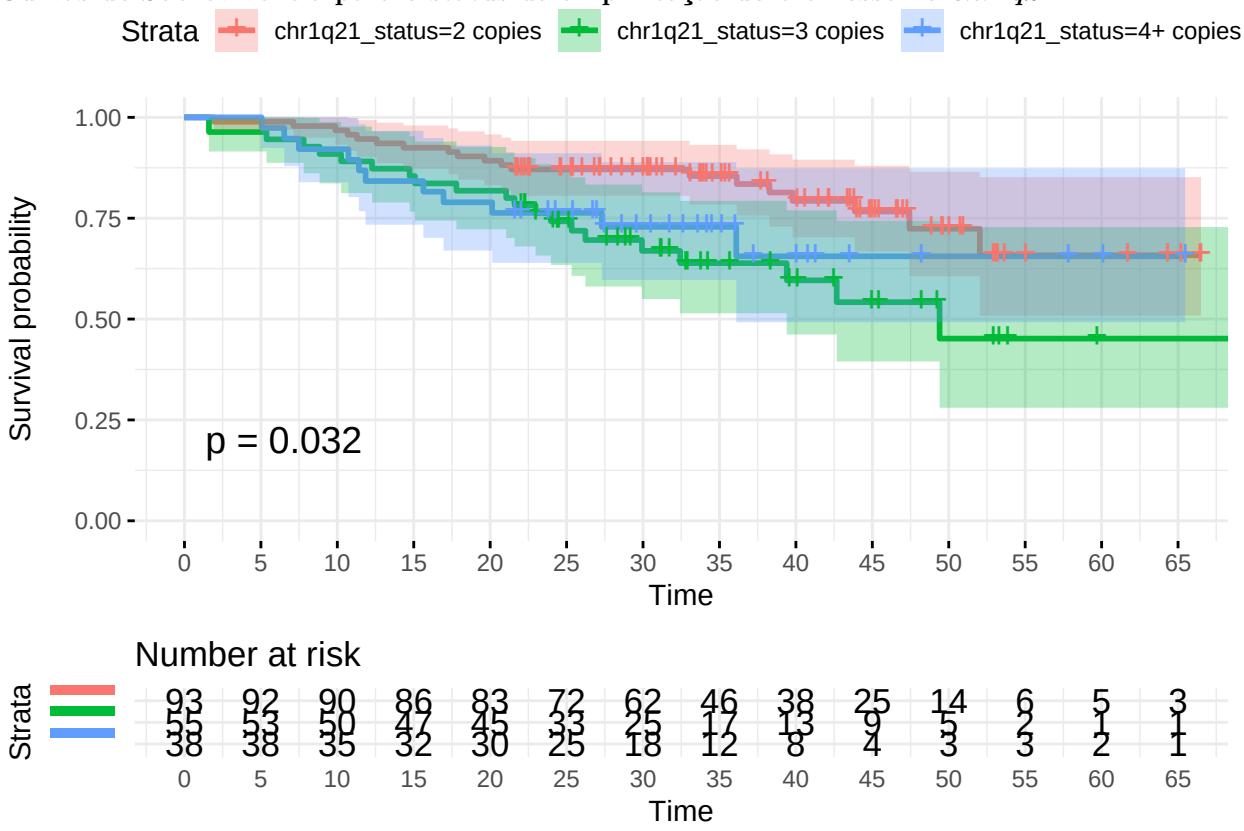
```
##
##  Pairwise comparisons using Log-Rank test
##
## data:  myeloma and molecular_group
##
##                  Cyclin D-1 Cyclin D-2 Hyperdiploid Low bone disease MAF
## Cyclin D-2       0.723      -          -            -                -
## Hyperdiploid     0.943      0.723      -            -                -
## Low bone disease 0.723      0.988      0.644        -                -
## MAF              0.644      0.447      0.523        0.485            -
## MMSET            0.328      0.103      0.103        0.103            0.723
## Proliferation    0.103      0.038      0.038        0.062            0.485
##                  MMSET
## Cyclin D-2       -
## Hyperdiploid     -
## Low bone disease -
## MAF              -
## MMSET            -
## Proliferation    0.527
##
## P value adjustment method: BH
```

De acordo com o teste Log-Rank entre os grupos moleculares, podemos concluir que existe diferença significativa entre os seguintes grupos moleculares:

- Proliferation e Cyclin D-2
- Proliferation e Hyperdiploid

```
ggsurvplot(
   fit,                     # survfit object with calculated statistics.
   data = df,  # data used to fit survival curves.
   risk.table = TRUE,       # show risk table.
   pval = TRUE,             # show p-value of log-rank test.
   conf.int = TRUE,         # show confidence intervals for
                            # point estimaes of survival curves.
   xlim = c(0,65),        # present narrower X axis, but not affect
                            # survival estimates.
   break.time.by = 5,     # break X axis in time intervals by 500.
   ggtheme = theme_minimal(), # customize plot and risk table with a theme.
 risk.table.y.text.col = T, # colour risk table text annotations.
  risk.table.y.text = FALSE # show bars instead of names in text annotations
                            # in legend of risk table
)
```

Curvas de Sobrevivencia para o status da amplificação do cromossomo *chr1q21*



O p-valor do grafico acima é do teste log-rank, e como o valor deu menor que 0.05, podemos concluir que há evidência estatística de que as curvas de sobrevivência são diferentes para a amplificação do cromossomo *chr1q21. Em seguida temos que fazer o teste 2 a 2 para essas curvas

```
# log rank 2 a 2
abc <- pairwise_survdiff(Surv(time, event) ~ chr1q21_status,
     data = myeloma)
abc
```

```
##
##  Pairwise comparisons using Log-Rank test
##
## data:  myeloma and chr1q21_status
##
##           2 copies 3 copies
## 3 copies  0.025    -
## 4+ copies 0.193    0.508
##
## P value adjustment method: BH
```

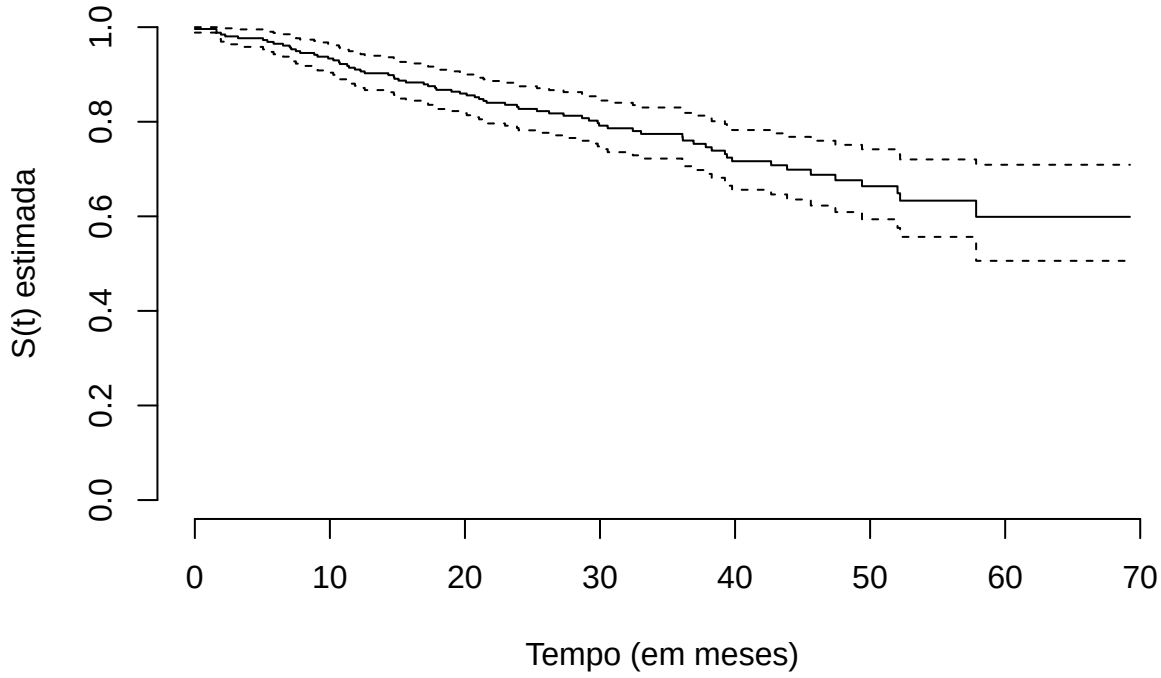
Podemos concluir que existe diferença significativa entre os seguintes status do cromossomo: 2 copias e 3 copias (p-valor abaixo de 0.05)

Algoritmo Nelson-Aalen

```
na_fit <- survfit(coxph(Surv(time, event) ~ 1, data = df))
summary(na_fit, times = seq(0,70,5))
```

```
## Call: survfit(formula = coxph(Surv(time, event) ~ 1, data = df))
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     0    256       1    0.996 0.00389        0.989        1.000
##     5    250       5    0.977 0.00944        0.958        0.995
##    10    239      11    0.934 0.01553        0.904        0.965
##    15    228      11    0.891 0.01947        0.853        0.930
##    20    220       8    0.860 0.02169        0.818        0.903
##    25    179       8    0.827 0.02373        0.782        0.875
##    30    149       7    0.792 0.02623        0.742        0.845
##    35    116       3    0.774 0.02752        0.722        0.830
##    40     94       8    0.717 0.03217        0.656        0.782
##    45     69       2    0.699 0.03376        0.636        0.768
##    50     50       3    0.664 0.03767        0.594        0.742
##    55     27       2    0.633 0.04168        0.556        0.720
##    60     10       1    0.599 0.05159        0.506        0.709
##    65      6       0    0.599 0.05159        0.506        0.709
```

```
plot(na_fit, conf.int=T,  xlab="Tempo (em meses)", ylab="S(t) estimada", bty="n")
```



Tábua Atuarial

```
lifetab2(Surv(time, event) ~ 1, data = df, breaks = seq(0,70,5))
```

```
##         tstart tstop nsubs nlost nrisk nevent      surv         pdf      hazard
## 0-5          0     5   256     0 256.0      6 1.0000000 0.004687500 0.004743083
## 5-10         5    10   250     0 250.0     11 0.9765625 0.008593750 0.008997955
## 10-15       10    15   239     0 239.0     11 0.9335938 0.008593750 0.009421842
## 15-20       15    20   228     0 228.0      8 0.8906250 0.006250000 0.007142857
## 20-25       20    25   220    33 203.5      8 0.8593750 0.006756757 0.008020050
## 25-30       25    30   179    23 167.5      7 0.8255912 0.006900464 0.008536585
## 30-35       30    35   149    30 134.0      3 0.7910889 0.003542189 0.004528302
## 35-40       35    40   116    14 109.0      8 0.7733780 0.011352337 0.015238095
```

```
## 40-45      40    45    94    23  82.5      2 0.7166163 0.003474503 0.004907975
## 45-50      45    50    69    16  61.0      3 0.6992438 0.006877807 0.010084034
## 50-55      50    55    50    21  39.5      2 0.6648547 0.006732706 0.010389610
## 55-60      55    60    27    16  19.0      1 0.6311912 0.006644118 0.010810811
## 60-65      60    65    10     4   8.0      0 0.5979706 0.000000000 0.000000000
## 65-70      65    70     6     6   3.0      0 0.5979706 0.000000000 0.000000000
## 70-Inf     70   Inf     0     0   0.0      0 0.5979706          NA          NA
##            se.surv      se.pdf   se.hazard
## 0-5    0.000000000 0.001891105 0.001936219
## 5-10   0.009455526 0.002534833 0.002712299
## 10-15  0.015561930 0.002534833 0.002840004
## 15-20  0.019506821 0.002174908 0.002524979
## 20-25  0.021727144 0.002347671 0.002834946
## 25-30  0.023932025 0.002560874 0.003225791
## 30-35  0.026245434 0.002025474 0.002614249
## 35-40  0.027577954 0.003884710 0.005383570
## 40-45  0.032034021 0.002431847 0.003470201
## 45-50  0.033530156 0.003886048 0.005820169
## 50-55  0.037299082 0.004654004 0.007344085
## 55-60  0.042330039 0.006482242 0.010806862
## 60-65  0.051514121         NaN         NaN
## 65-70  0.051514121         NaN         NaN
## 70-Inf 0.051514121          NA          NA
```

4) Conclusão

TODO: escrever a conclusão

5) Bibliografia

```
citation("survminer")
```

```
##
## To cite package 'survminer' in publications use:
##
##   Kassambara A, Kosinski M, Biecek P (2021). _survminer: Drawing
##   Survival Curves using 'ggplot2'_. R package version 0.4.9,
##   <https://CRAN.R-project.org/package=survminer>.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {survminer: Drawing Survival Curves using 'ggplot2'},
##     author = {Alboukadel Kassambara and Marcin Kosinski and Przemyslaw Biecek},
##     year = {2021},
##     note = {R package version 0.4.9},
##     url = {https://CRAN.R-project.org/package=survminer},
##   }
```