Introduction to survival analysis with R

Statistics M1

Raphaël Bonnet (PhD student C3M-U1065, Inserm, UCA)

13 novembre 2020

Introduction notes

In this course you will learn to work with survival data, and to handle gene expression and annotations matrix.

Access & explore the data

Loading packages

```
suppressMessages(library(limma))
suppressMessages(library(survival))
suppressMessages(library(survminer))
suppressMessages(library(genefilter))
```

Loading data

```
#load up data

data=readRDS("data_intro.RData")
clinical=data[[1]] #clinical data
dat.m=data[[2]] #data matrix
annot=data[[3]] #probe/gene correspondance
```

Basic functions R - Exploring the data

- Data matrix are in RNA-DNA seq are constituted of:
  - n rows corresponding to genes called features
  - p columns corresponding to patients called samples

- Before starting to work on a object its always necessary to
  - know the type of object (character, numeric, boolean, matrix, data.frame, list, S4, etc.)
  - know the dimensions of the object

- Using class(), dim(), colnames(), rownames() and unique() on expression data (dat.m & annot)

```
#dims and class of dat.m
dim(dat.m)
```

```
## [1] 54696   114
```

```
class(dat.m)
```

```
## [1] "matrix" "array"
```

```
# head(), colnames(), rownames()
print('first 5 sample names')
```

```
## [1] "first 5 sample names"
```

```
head(colnames(dat.m))
```

```
## [1] "411409" "411408" "411407" "411406" "411405" "411404"
```

```
print('first 5 feature names')
```

```
## [1] "first 5 feature names"
```

```
head(rownames(dat.m))
```

```
## [1] "1" "2" "3" "4" "5" "6"
```

```
#Bien penser à utliser head() pour n'obtenir qu'un apercu des données
```

We must include gene names into the data matrix

```
# Have a look at the annotation table
head(annot)
```

```
##       probe gene_symbol
## 1 1007_s_at        DDR1
## 2   1053_at        RFC2
## 3    117_at       HSPA6
## 4    121_at        PAX8
## 5 1255_g_at      GUCA1A
## 6   1294_at        UBA7
```

```
# ! let's include gene names into the matrix
rownames(dat.m)=annot[,2]
# let's check rownames
head(rownames(dat.m))
```

```
## [1] "DDR1"   "RFC2"   "HSPA6"  "PAX8"   "GUCA1A" "UBA7"
```

```
#extract gene names
genes=rownames(dat.m)
```

Not all probes have a corresponding annotated genes, this is why we have many genes called NA

- Using sum() and is.na() on genes (dat.m)

How many genes are NA?

How many genes have been annotated?

```
print('na genes')
```

```
## [1] "na genes"
```

```
sum(is.na(genes))
```

```
## [1] 15811
```

```
print('annotated genes')
```

```
## [1] "annotated genes"
```

```
sum(!is.na(genes))
```

```
## [1] 38885
```

```
#ou nrow(dat.m)-sum(is.na(genes))
```

- Most of the time, and depending on the sequencing library we should find

- ~20K protein-coding genes
- and ~10K unannnotated pseudogenes (NA)

- Using class() name() and $ (accessor) on clinical data (clinical)

- Clinical tables in survival studies are constituted of:
  - n rows corresponding to samples
  - p columns corresponding to clinical features

```
class(clinical)
```

```
## [1] "data.frame"
```

```
dim(clinical)
```

```
## [1] 114   2
```

```
#we can have a look at clinical features
names(clinical)
```

```
## [1] "clinical.OS"
## [2] "clinical.OS_months_or_MIN_months_of_OS"
```

```
#two types of clinical data required
unique(clinical$clinical.OS)  #boolean
```

```
## [1] 1 0
```

```
class(clinical$clinical.OS_months_or_MIN_months_of_OS) #numeric
```

```
## [1] "numeric"
```

What are the practical differences between data.frame and matrix objects ?

Check if dimensions matches in dat.m, clinical and annot

What are the number of features, clinical features and samples ?

```
#features (annotated)
sum(!is.na(genes))
```

```
## [1] 38885
```

```
#samples
nrow(clinical)
```

```
## [1] 114
```

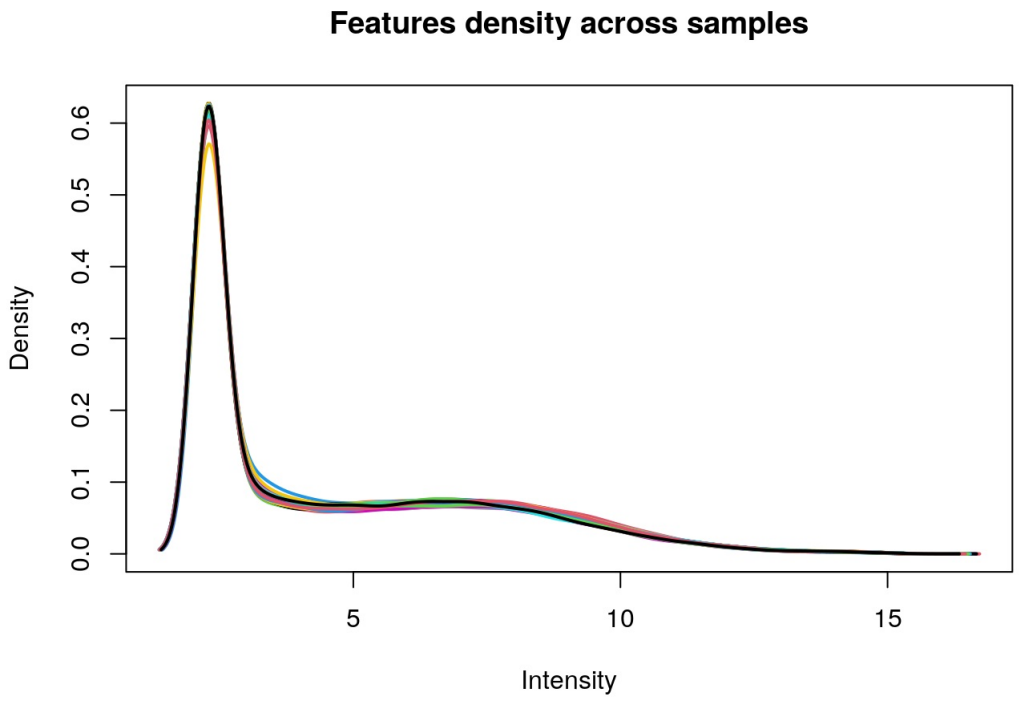
```
#clinical features
ncol(clinical)
```

```
## [1] 2
```

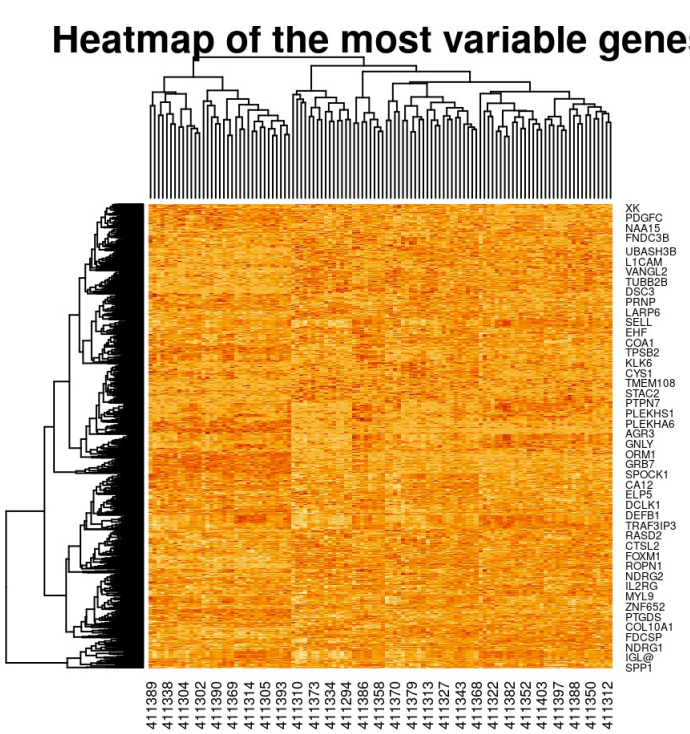
It’s very rare to have access to clinical data, most of the time, researcher must provide the omic data but the clinical infromations remain confidential.

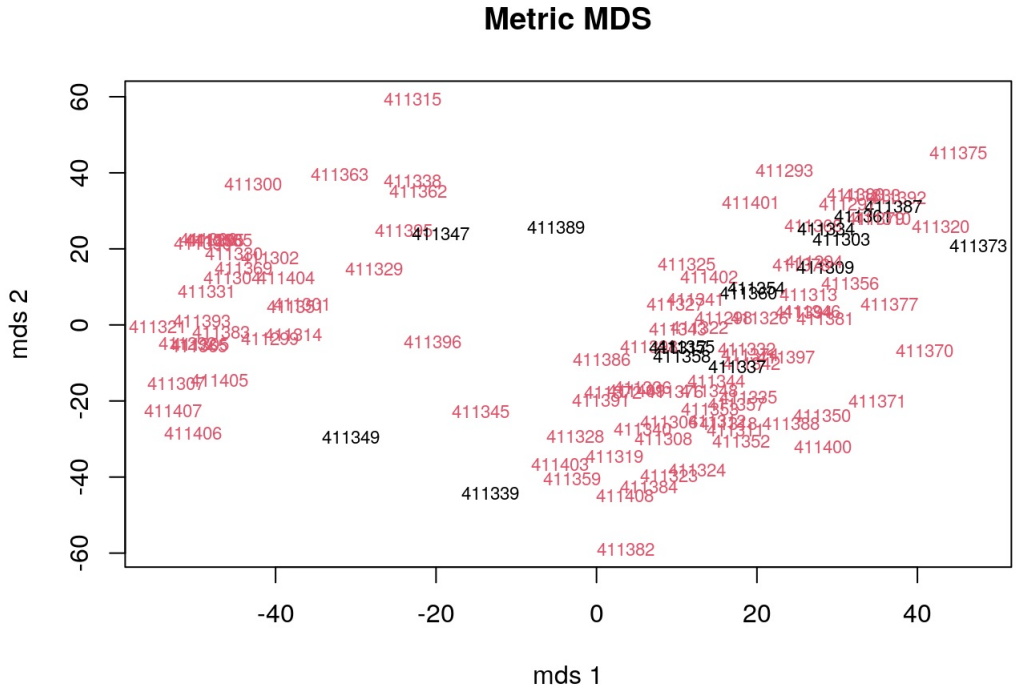
To access them, one must contact the holder of the clinical data, ask for access and wait for authorization and afterwards data. It is best not to rely on that.

Overview of the gene expression: & heatmap & MDS plots


Features density across samples
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
5
10
15
Intensity

```
## [1] 54696   114
```





Survival of the patient is overlayed in color (black: remission, red: relapse)

These vizualisation gives us some insights on the quality of the expression set we are analysing.

- We can have a look to the distribution of the feature (genes) expressions.
- We can overlook the expression of the top variable genes in a heatmap to see wether or not some patterns already emerge.
- We can see here that the most variable genes fail to explain the survival of the patients.

Cox Proportional Hazards Regression Analysis

Basic concepts

- **Survival time**: (how long one survive)
  - Event-free survival time: (how long one survive without any event)

- **Events**
  - Relapse
  - Death

- **Censoring**: end of patient following or incomplete data
  - End of treatment (5 years in general)
  - Loss of consent

- **Survival probability**

function S(t): survival probability of an individual from t0 to t

S(t) is a **step** function that **changes value only at the time of each event**.

- **Kaplan-Meier plot**

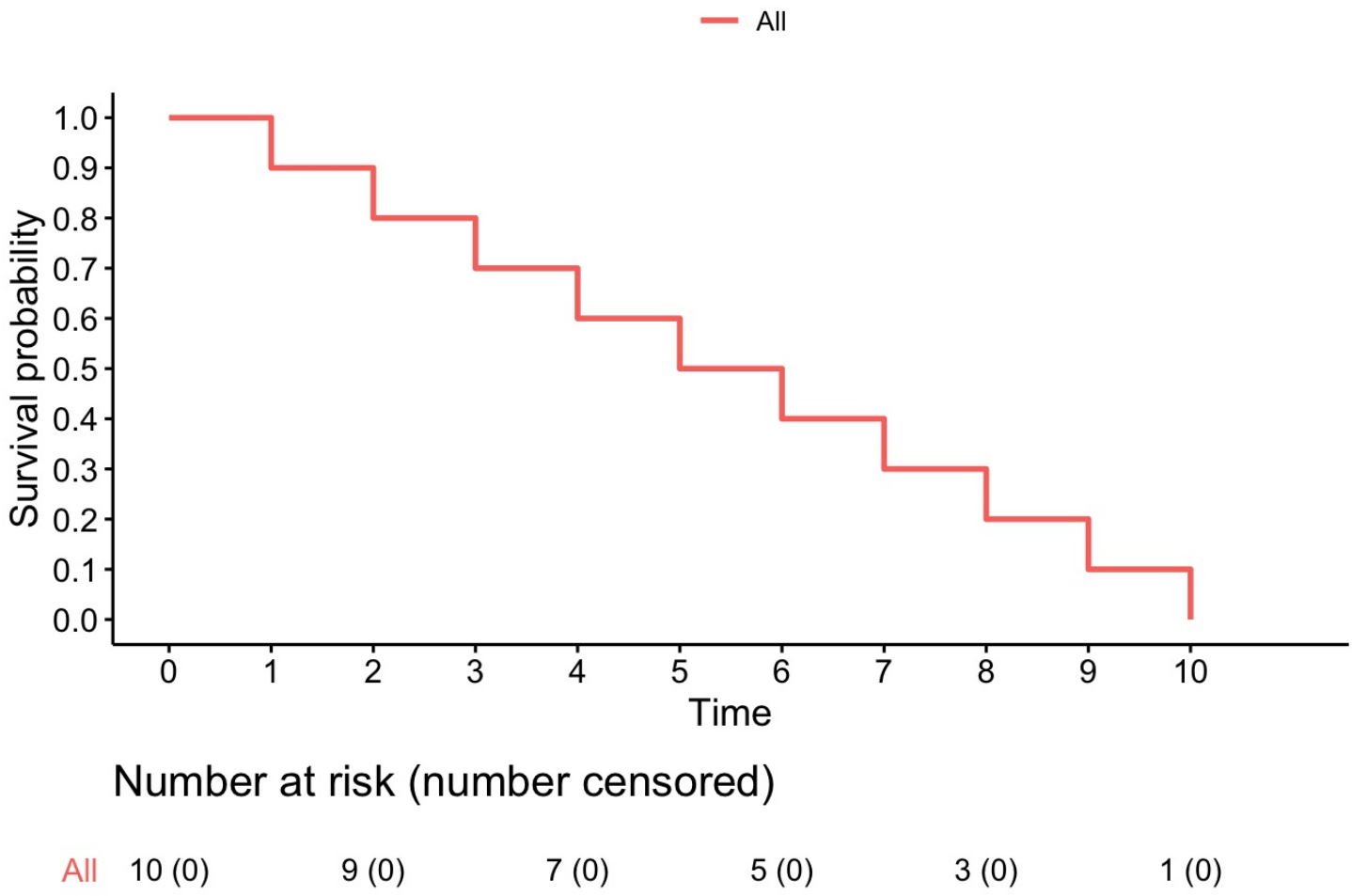
Represent the estimation of the survival probability based on true survival times. *(Kaplan and Meier, J Am Stat Assoc, 1958).*

Provides a useful summary of the data that can be used to estimate measures such as **median survival time**.

Without cencoring in the data

$$S(t)=(N - n)/N$$

- *Where*
  - *N = the total number of patients*
  - *n = the number of events from t0 to t*



Calculate the survival probability at t=7

```
print('survival time t=7')
```

```
## [1] "survival time t=7"
```

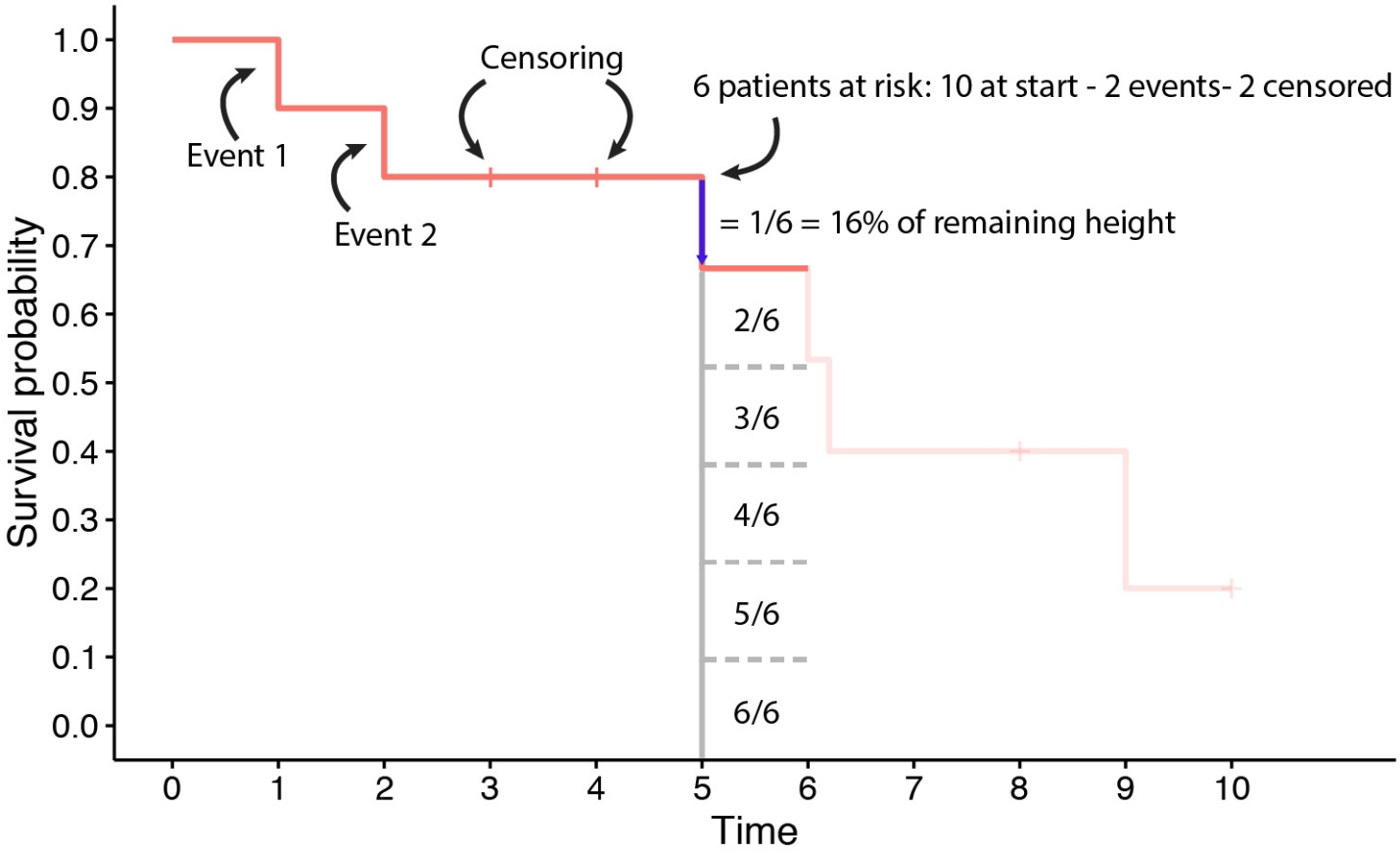
```
St7=(10-7)/10
print(St7)
```

```
## [1] 0.3
```

With cencoring in the data

$$S(t)=S(t-1)(1-d/n)$$

- *Where*
  - *S(t−1) = the probability of being alive at t−1*
  - *n = the number of patients left just before t*
  - *d = the number of events at t*
  - *dt0 = 0*
  - *S(0) = 1*



Calculate the survival probability at t=5

```
print('survival time t=5')
```

```
## [1] "survival time t=5"
```

```
St5=0.8*(1-(1/6))
#St4=0.8 d=1 et n=6 car 2 event et 2 censor avant t=5
print(St5)
```

```
## [1] 0.6666667
```

What are the median survival times for the 2 previous examples?

Median survival times are respectively 5 and 6 (here arbitrary unit).

Implementing the model in R

To fit a cox model to data, we use the following code:

fit=survfit(Surv(os_time,os) ~ covariate)

os_time is a numeric vector associated to every patients containing the time from diagnosis (t0) to event (death, relapse, cencoring)

os is a boolean resulting from the question 'does an event occured?'

covariate here set to 1 (no covariate)

- **Hazard ratio probalility**

function h(t): event probability at time t

exp(coefficient) gives you the survival ratio between two groups

(i.e: exp(coefficient)=0.2 means that the risk group has 20% more chance of an occuring event)

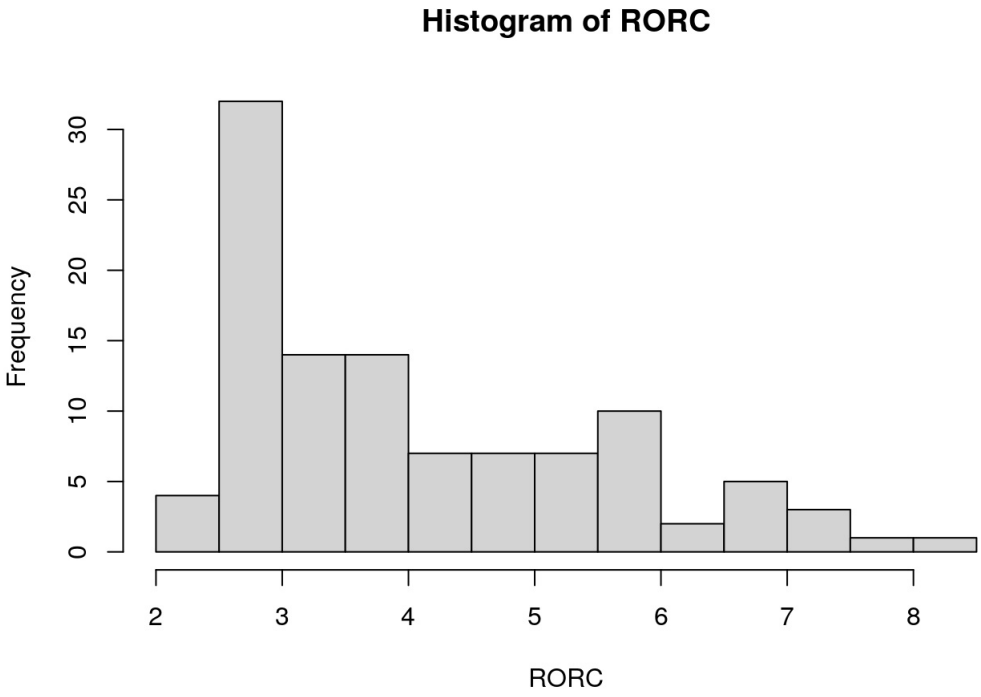
- Using the mean of gene expression of all patients as a cut-off, it is possible to discretize (0: low expression, 1: high expression) the gene expression to create two groups of patients.

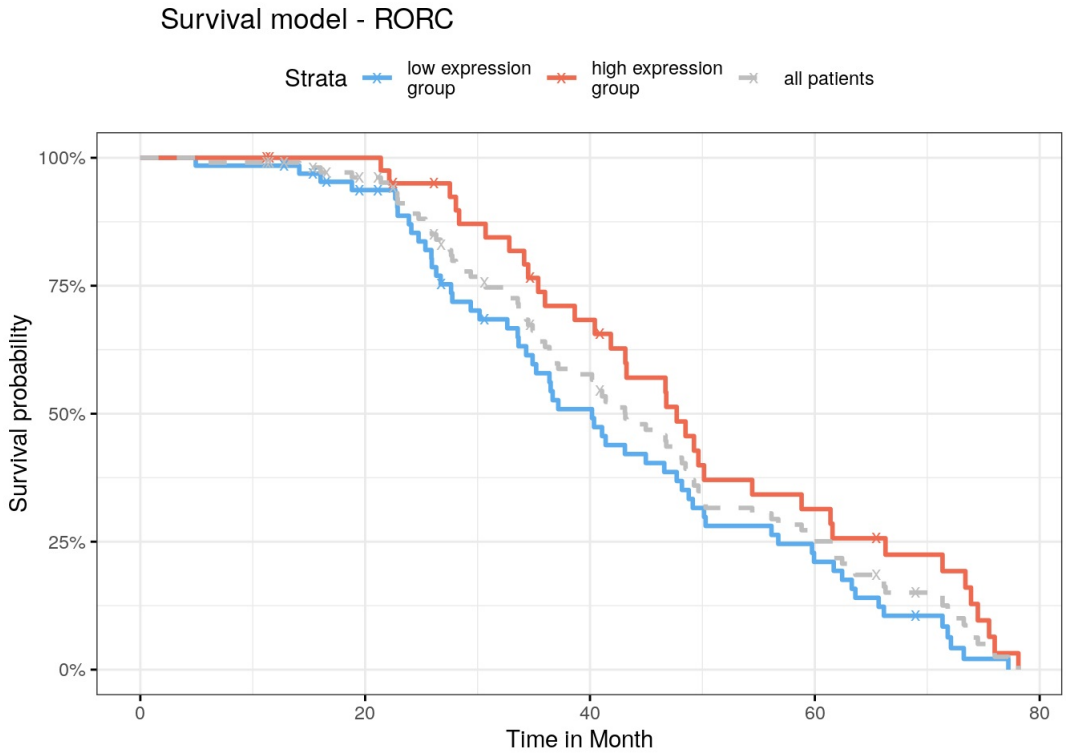
- RORC - regulatory role in thymopoiesis

```
dat.m=dat.m[,-which(is.na(clinical$clinical.OS_months_or_MIN_months_of_OS))]
clinical=clinical[-which(is.na(clinical$clinical.OS_months_or_MIN_months_of_OS)),]

g=which(rownames(dat.m)=="RORC")
  if (length(g)>1){subset=apply(dat.m[g,],MARGIN = 2,FUN = max)} else {subset=dat.m[g,]}
gene=ifelse(subset>mean(subset),1,0)
RORC=as.numeric(subset)
hist(RORC,breaks = 10)

fit <- survfit(Surv(clinical$clinical.OS_months_or_MIN_months_of_OS,clinical$clinical.OS) ~ gene ,

            data = data.frame(clinical))
fit_table <- coxph(Surv(clinical$clinical.OS_months_or_MIN_months_of_OS,clinical$clinical.OS) ~ gene)
info=summary(fit_table)
pval=info$sctest[3]
HR=info$coefficients[2]
if (pval<0.05){print(pval);print(HR)}
```

```
##     pvalue
## 0.04053539
## [1] 0.6380224
```

```
fit0 <- survfit(Surv(clinical$clinical.OS_months_or_MIN_months_of_OS,clinical$clinical.OS) ~ 1,
              data = data.frame(clinical))
suppressWarnings(ggsurvplot(list('risk'=fit,'null.model'=fit0),data=clinical,
           pval = TRUE,palette =c("steelblue2", "coral2",'Grey'),legend.labs =
               c("low expression\ngroup","high expression\ngroup","all patients"),linetype = c(1,1,2),censor.shap
e="x", censor.size = 3,
           title = "\tSurvival model - RORC",xlab = "Time in Month",pval.method =   TRUE,
           ggtheme = theme_bw(), ncensor.plot = F, combine = T,surv.scale=c("percent")))
```





- To what correspond the fit0 variable ?

fit0 corresponds to the survival of the whole cohort, without covariate that splits data in groups.

- Display the 'fit' variable, what are the proportions of events for each group ?

```
print(fit)
```

```
## Call: survfit(formula = Surv(clinical$clinical.OS_months_or_MIN_months_of_OS,
##     clinical$clinical.OS) ~ gene, data = data.frame(clinical))
##
##         n events median 0.95LCL 0.95UCL
## gene=0 65     57   40.2    34.9    48.2
## gene=1 42     35   47.7    41.9    61.4
```

There are 65 patients in first group (gene=0) with 57 events. ~ 87% of events. Mean surv time = 40.2 months

There are 42 patients in first group (gene=1) with 35 events. ~ 87% of events. Mean surv time = 47.7 months

- Display the 'fit_table' variable, have a look at the output

```
print(fit_table)
```

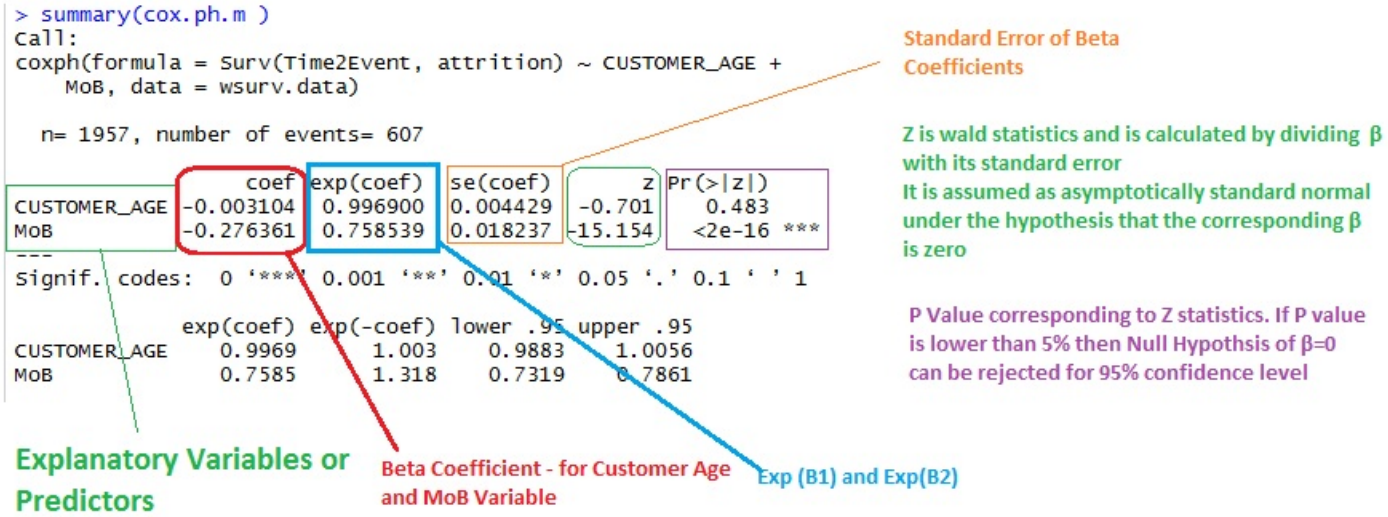
```
## Call:
## coxph(formula = Surv(clinical$clinical.OS_months_or_MIN_months_of_OS,
##     clinical$clinical.OS) ~ gene)
##
##         coef exp(coef) se(coef)      z      p
## gene -0.4494    0.6380   0.2211 -2.033 0.0421
##
## Likelihood ratio test=4.26  on 1 df, p=0.03898
## n= 107, number of events= 92
```

Better survival for low expression group for RORC expression is consistent as it is a prognostic factor.

Be very careful when analysing expression data as many other confounding factors can be responsible for the heterogeneity from one cohort to another.

In silico hypothesis must be confronted to a biological experiment

Statistics:



- What is the pvalue for this model ?

Likelihood ratio test p=0.03898

- What is the hazard ratio between the two groups ? What does it mean ?

HR = 0.6380.

As HR < 1 the effect of the covariate is favorable on survival.

Explore another dataset:

Go to https://stat.ethz.ch/R-manual/R-devel/library/survival/html/lung.html (https://stat.ethz.ch/R-manual/R-devel/library/survival/html/lung.html)

- How many patients are there ?

```
#load the data
data(lung)
dim(lung)[1]
```

```
## [1] 228
```

- How many men - women?

```
#Number of male, female
male=sum(lung$sex==1)
female=sum(lung$sex==2)
```

- To what correspond the column named status ?

The status columns corresponds to the survival status of the patients. 1 alive (or censored), 2 dead

- How many number of events for this variable status ?

```
#number of events
events=sum(lung$status==2)
```

- How many other covariate are there and to what do they correspond ?

They are age, sex, ph.ecog, ph.karno, pat.karno, meal.cal wt.loss See documentation https://stat.ethz.ch/R-manual/R-devel/library/survival/html/lung.html (https://stat.ethz.ch/R-manual/R-devel/library/survival/html/lung.html) for more info.

- Fit and represent a cox model with no covariate, what is the median survival time?

```
#fit
 fit0 <- survfit(Surv(time,status) ~ 1 ,
            data = data.frame(lung))
```

Mean survival time between 306 and 310. Use summary(fit0) or print(fit0)

- What is the value of S(t=558) ?

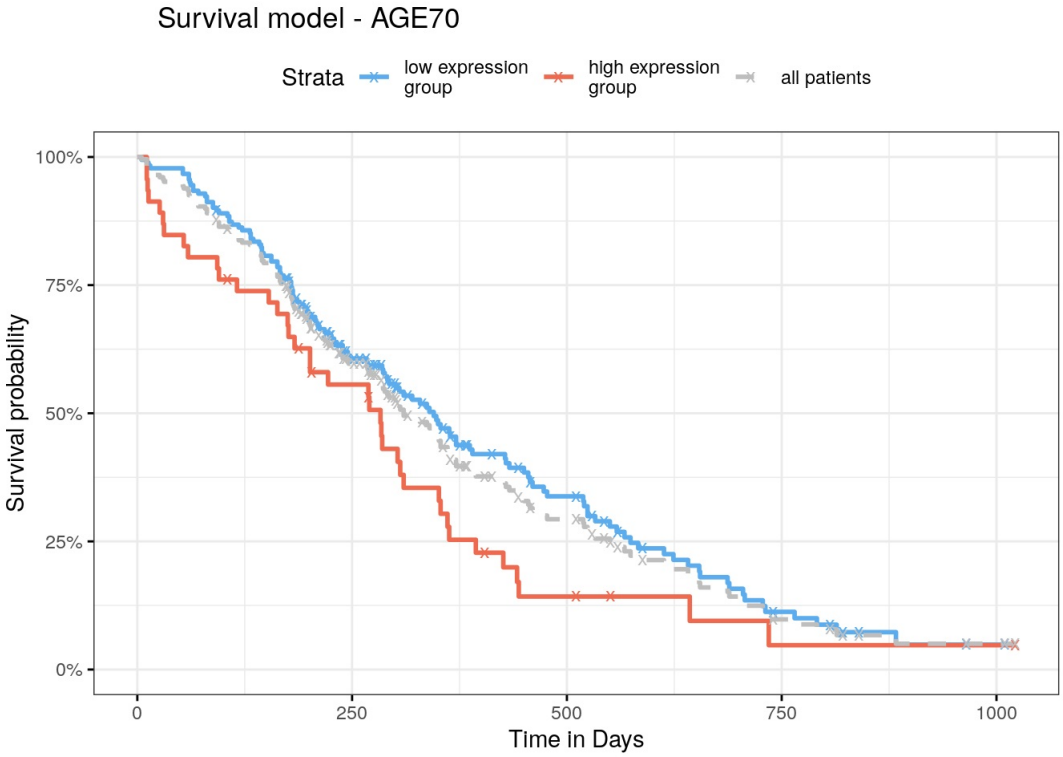
S(t=558)=23.92%

- Use discrete age variable (cut-off>70) to fit a new model, what do we see ?

```
#descretize
lung$age=ifelse(lung$age>70,1,0)

fit <- survfit(Surv(time,status) ~ age ,
            data = data.frame(lung))
fit_table <- coxph(Surv(clinical$clinical.OS_months_or_MIN_months_of_OS,clinical$clinical.OS) ~ gene)
info=summary(fit_table)
pval=info$sctest[3]
HR=info$coefficients[2]
if (pval<0.05){print(pval);print(HR)}
```

```
##     pvalue
## 0.04053539
## [1] 0.6380224
```

```
suppressWarnings(ggsurvplot(list('risk'=fit,'null.model'=fit0),data=lung,
           pval = TRUE,palette =c("steelblue2", "coral2",'Grey'),legend.labs =
               c("low expression\ngroup","high expression\ngroup","all patients"),linetype = c(1,1,2),censor.shap
e="x", censor.size = 3,
           title = "\tSurvival model - AGE70",xlab = "Time in Days",pval.method =   TRUE,
           ggtheme = theme_bw(), ncensor.plot = F, combine = T,surv.scale=c("percent")))
```


Survival model - AGE70
Strata
low expression group
high expression group
all patients
100%
75%
50%
25%
0%
Survival probability
0
250
500
750
1000
Time in Days