

Statistical Computing



Survival Analysis - ML | January , 2022
Emmanuelle Rodrigues Nunes

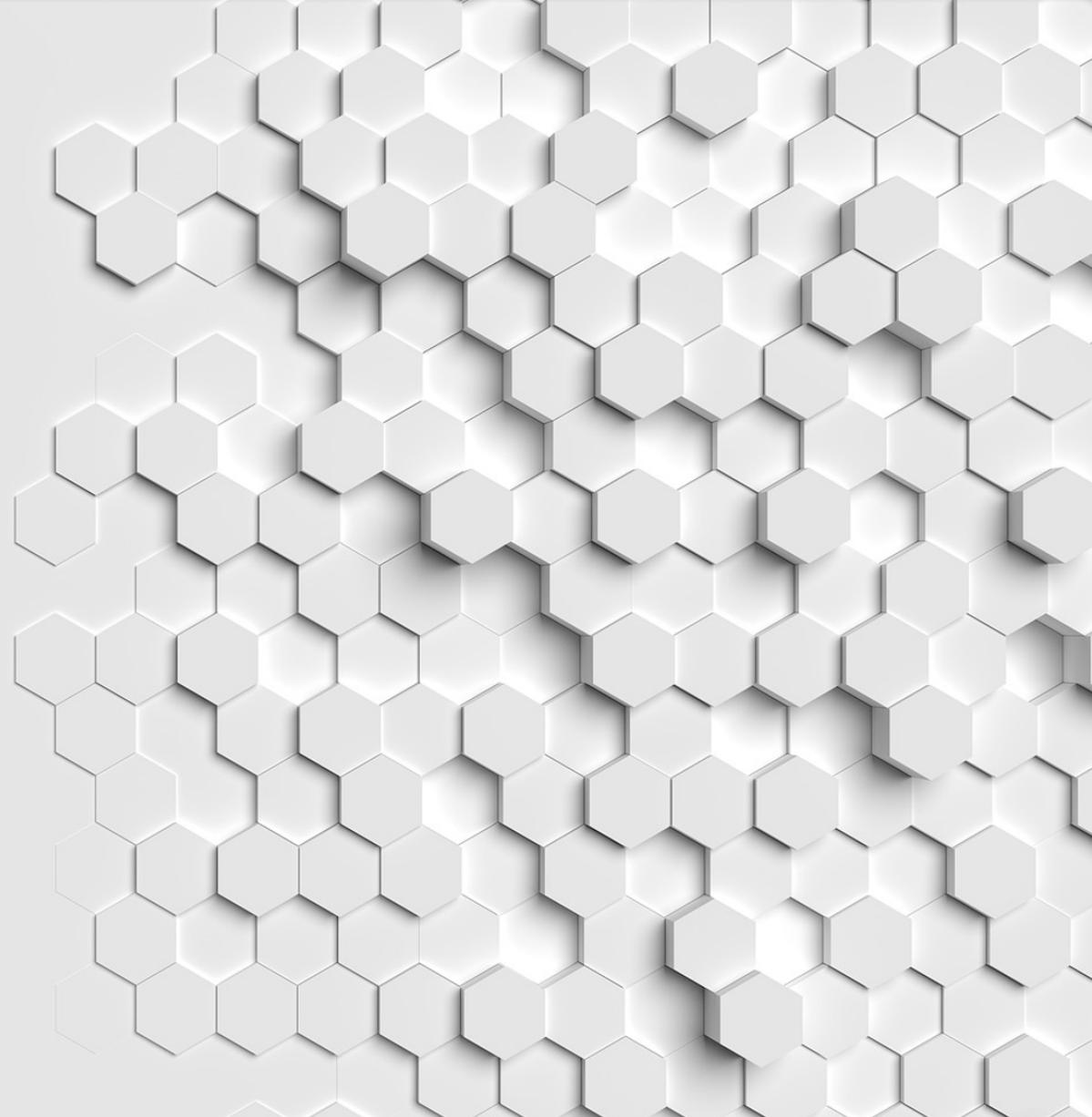


CIDRA-AIPD
African Institute
for Professional
Development



STATISTICS WITHOUT
BORDERS

@SWBprobono



StatisticsWithoutBorders.org 1

Survival Analysis

Survival Analysis - Recapping

Survival analysis utilises Time-To-Event (TTE) data where the event can be any *binary* outcome, not necessarily death, without assuming the rates are constant. It lets you model:

- Time until an event occurs or has not yet occurred due to censoring
- Compare the time-to-event between different groups
- Assess how time-to-event correlates with quantitative or categorical variables

For example:

- Well defined time
 - Time from surgery to death
 - Time from start of treatment to cancer progression
 - Time from HIV infection to development of AIDS
- Unsure time
 - Time to heart attack
 - Time to onset of substance abuse
 - Time to initiation of sexual activity
 - Time to machine malfunction

Important feature: Survival analysis can handle missing data due to censoring

Definitions

The **hazard** is the instantaneous event (death) rate at a particular time point t , conditional on having survived to t .

- Survival analysis **doesn't** assume the hazard is constant over time.
- The cumulative hazard is the total hazard experienced up to time t .

The **survival function**, is the probability an individual will survive (or, the probability that the event of interest does not occur) beyond time t .

- It's the probability that the event hasn't occurred yet.

$$\mathbb{S}(t) = \mathbb{P}(T > t),$$

where T is the time of death

Definitions - Kaplan-Meier

The **Kaplan-Meier** curve illustrates the survival function.

- It estimates $\hat{S}(t)$
- It's a step function illustrating the cumulative survival probability over time.
- It's the product of probabilities that the subject did not experience the event in any interval up to time t

$$\hat{S}(t) = \prod_{t_i < t} \left(1 - \frac{d_i}{n_i}\right),$$

where d_i is the number of subjects that died (or experienced the event of interest) at time t_i , and n_i is the number of subjects at risk at time t_i .

At time 0, the survival probability is 1, i.e., $S(t_0) = 1$

Definitions - Censoring

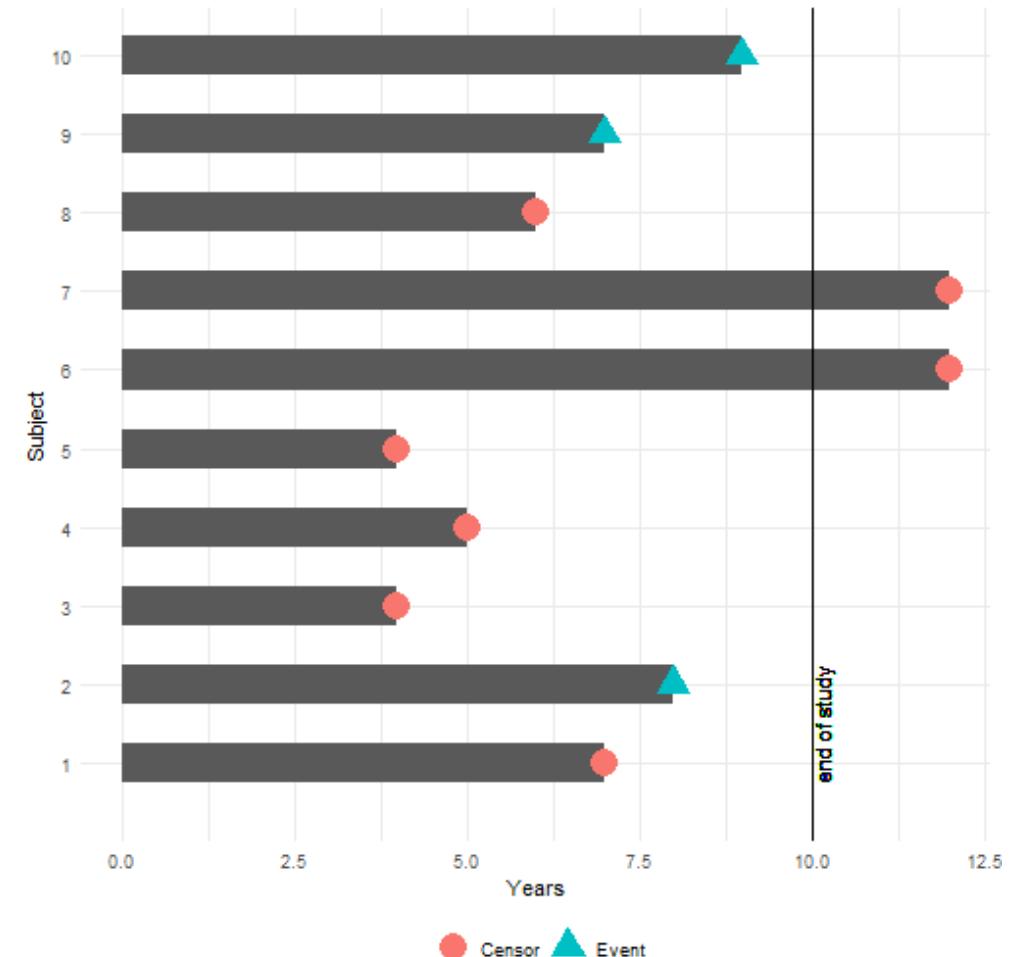
Censoring is a type of missing data problem unique to survival analysis.

- It occurs when you track the sample/subject through the end of the study and the event never occurs, i.e., no event by end of fixed study period
- It could also happen due to the sample/subject dropping out of the study for reasons other than death, for example, stopping treatment due to harmful side effects
- It could also happen due to loss to followup,

If the sample is censored, you only know that the individual survived up to the loss to followup, but you do not know anything about survival after that.

Specifically these are examples of **right censoring**.

Left censoring and interval censoring are also possible.



Survival Analysis in R

Packages we will use for this example:

- **survival**
 - The core functions for this type of analysis are available in this library
 - It is a part of the standard R packages, so you don't need to `install.packages()` before using it
- **dplyr**
 - Data manipulation
- **survminer**
 - Better visualization for Kaplan-Meier plots

For the example we will use the dataset `lung` that is available within the `survival` package.

```
library(survival)
library(dplyr)
library(survminer)

head(lung, 3)
```

```
##   inst time status age sex ph.ecog ph.karno pat.karno meal.cal wt.loss
## 1    3  306      2  74   1       1      90       100     1175      NA
## 2    3   455      2  68   1       0      90       90     1225      15
## 3    3 1010      1  56   1       0      90       90       NA      15
```

Example - Lung

We can see the dimensions of the dataset by using the function `dim()` from base R. The first value represent the rows, in this example 228 and the second represents the columns, 10 in the example.

```
dim(lung)
```

```
## [1] 228 10
```

The 10 columns represent:

- **inst:** Institution code
- **time:** Survival time in days
- **status:** censoring status 1 = censored, 2 = dead
- **age:** Age in years
- **sex:** Male = 1, Female = 2
- **ph.ecog:** ECOG performance (0 = good, 5 = dead)
- **ph.karno:** Karnofsky performance rated by physician
- **pat.karno:** Karnofsky performance rated by patient
- **meal.cal:** Calories consumed at meals
- **wt.loss:** Weight loss in last six months

```
glimpse(lung)
```

```
## #> Rows: 228
## #> Columns: 10
## #> $ inst      <dbl> 3, 3, 3, 5, 1, 12, 7, 11, 1, 7, 6, 16,
## #> $ time       <dbl> 306, 455, 1010, 210, 883, 1022, 310, 36
## #> $ status     <dbl> 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2,
## #> $ age        <dbl> 74, 68, 56, 57, 60, 74, 68, 71, 53, 61,
## #> $ sex         <dbl> 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 2, 2,
## #> $ ph.ecog    <dbl> 1, 0, 0, 1, 0, 1, 2, 2, 1, 2, 1, 2, 1,
## #> $ ph.karno   <dbl> 90, 90, 90, 90, 100, 50, 70, 60, 70, 70
## #> $ pat.karno  <dbl> 100, 90, 90, 60, 90, 80, 60, 80, 80, 70
## #> $ meal.cal   <dbl> 1175, 1225, NA, 1150, NA, 513, 384, 538
## #> $ wt.loss    <dbl> NA, 15, 15, 11, 0, 0, 10, 1, 16, 34, 27
```

Lung dataset

As all the columns of the dataset are numerical, we need to transform the categorical ones into categorical.

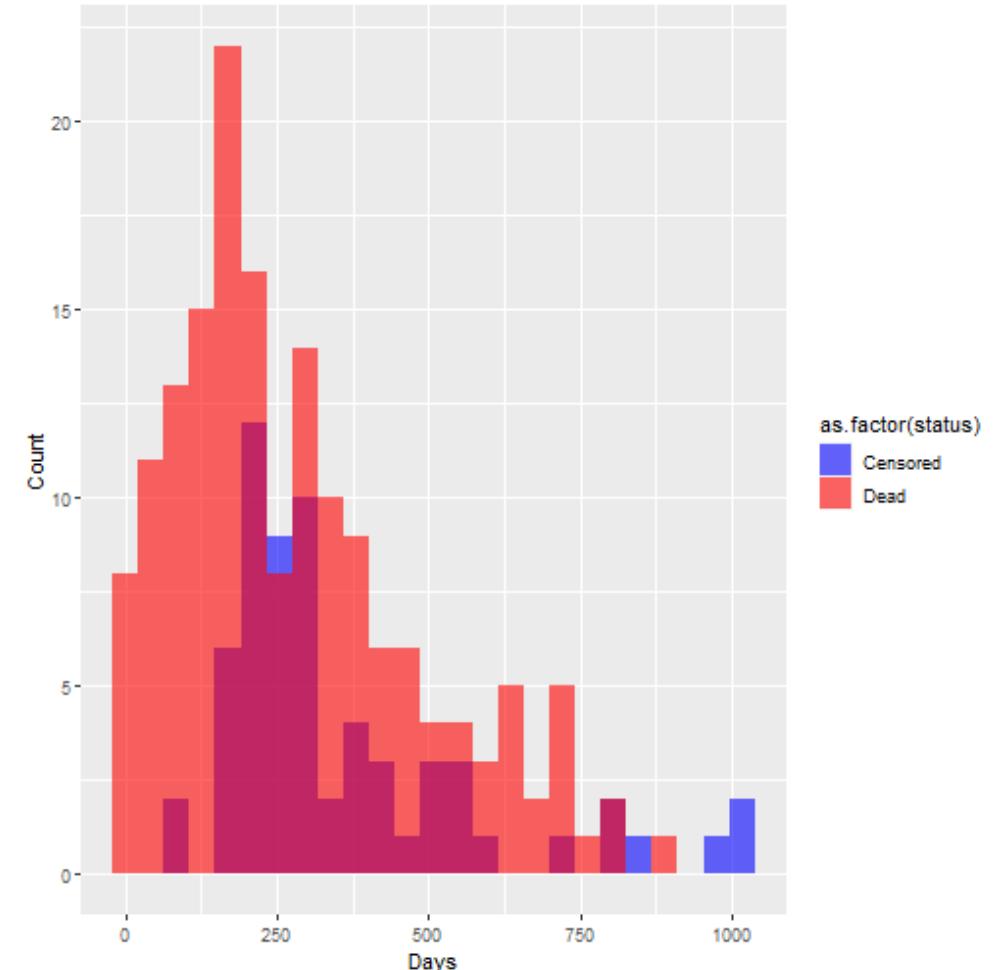
- Use function `mutate_at` from `dplyr` library
 - This function mutates the desired variables to a function
 - The function is `as.factor` to convert the variable to a factor
- We will also convert the `sex` variable to `Male` and `Female` for visualization purposes

```
lung ← lung %>%  
  mutate_at(vars("sex", "ph.ecog"), as.factor) %>%  
  mutate(sex = forcats::fct_recode(sex, "Male" = "1", "Female" = "2"))
```

Distribution of follow-up time

- Censored subjects still provide information so must be appropriately included in the analysis
- Distribution of follow-up times is skewed, and may differ between censored patients and those with events
- Follow-up times are always positive

```
ggplot(lung, aes(x = time,
                  fill = as.factor(status))) +
  geom_histogram(bins = 25,
                 alpha = 0.6,
                 position = "identity") +
  scale_fill_manual(values = c("blue", "red"),
                    labels = c("Censored",
                              "Dead")) +
  labs(x = "Days",
       y = "Count")
```



Kaplan-Meier Analysis

The first thing to do is to use `Surv()` from `survival` to build the standard survival object.

The most important parameters are:

- `time`: Follow-up time for right censored data;
 - For interval data it represents the start time of the interval.
 -
- `event`: Normally 0 = alive, 1 = dead
- `time2`: Ending time of the interval for interval censored
- `type`: Type of censoring
 - Options: "right", "left", "counting", "interval", "interval2" or "mstate"
 - Default: "right"

Note that a "+" after the time in the print out of `km` indicates censoring. It is represented by "+" because the true survival time is greater than the value as the event has not yet occurred.

```
km ← Surv(lung$time, lung$status)
head(km, 10)
```

```
## [1] 306 455 1010+ 210 883 1022+ 310 361 218 166
```

Kaplan-Meier Analysis

Now, let's fit a survival curve with the `survfit()` function. We will create a survival curve that doesn't consider any different variables, or groupings.

- We'll need to specify an intercept (represented by `~1` in the formula that `survfit` expects).
- We can fit the curve in one step by nesting the `Surv()` function within `survfit()`
- Similarly to how we specify data for linear models with `lm()`, we'll use the `data =` argument to specify which data we're using.

```
km_fit ← survfit(Surv(time, status) ~ 1, data = lung)
summary(km_fit, times = c(1, 30, 60, 90))
```

```
## Call: survfit(formula = Surv(time, status) ~ 1, data = lung)
##
##    time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    1     228      0    1.000  0.0000    1.000    1.000
##   30     219     10    0.956  0.0136    0.930    0.983
##   60     213      7    0.925  0.0174    0.892    0.960
##   90     201     10    0.882  0.0214    0.841    0.925
```

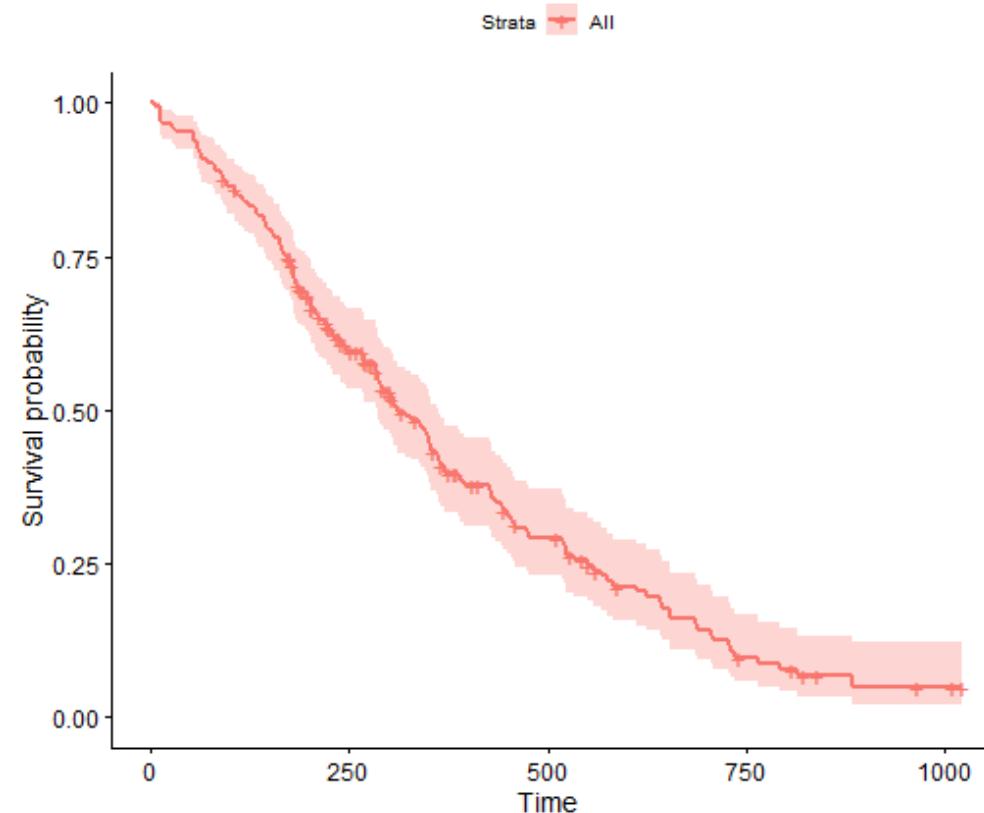
Kaplan-Meier Plot

- We can plot the `survfit` object using the function `plot` from `base`

```
plot(km_fit)
```

- The `ggsurvplot` function from the `survminer` package is built on `ggplot2`
- The default plot shows the step function (solid line) with associated confidence bands (shaded area).
- The tick marks representing censored patients are shown by default
 - Can be suppressed using the option `censor = FALSE`

```
ggsurvplot(km_fit)
```



Kaplan-Meier plot

We can also plot the KM curve upside down; this shows the probability of the event.

- It shows death (or the event of interest) instead of survival
- We can do this by adding the option `fun = 'event'` in both `plot` and `ggsurvplot`

```
plot(km_fit, fun = 'event')
```

```
ggsurvplot(km_fit, fun = 'event')
```

Estimating x-year survival

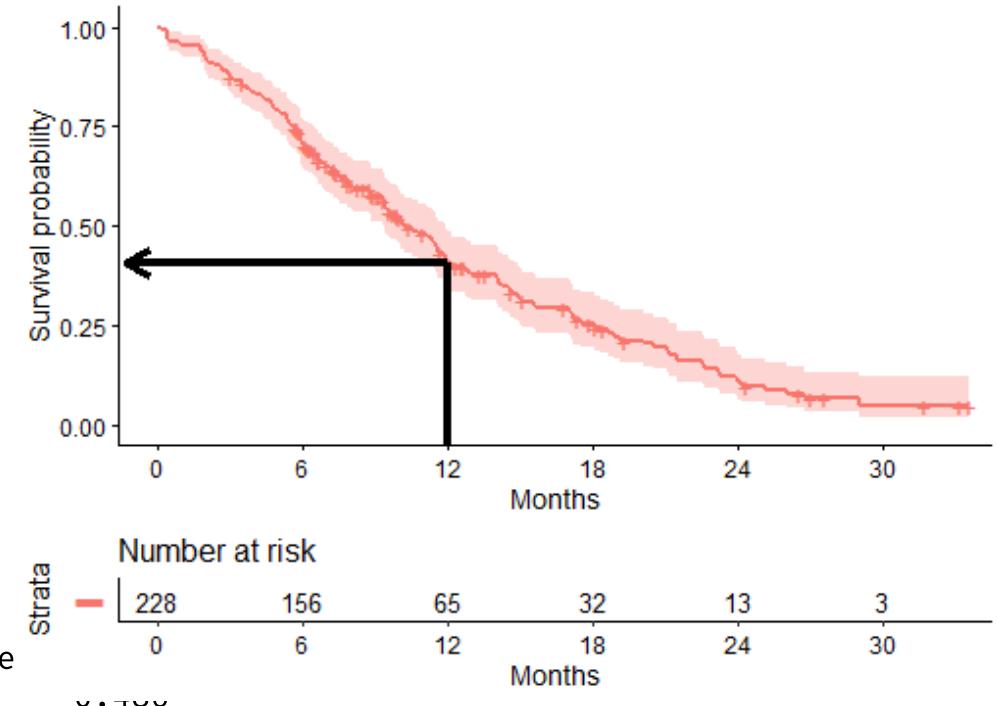
Often of interest is the probability of surviving beyond a certain number x of years.

For example, to estimate the probability of surviving to 1 year:

- Use `summary` with the `times` argument
 - Note the time variable in the `lung` data is actually in days

```
summary(km_fit, times = 365)
```

```
## Call: survfit(formula = Surv(time, status) ~ 1, data =
## 
##   time n.risk n.event survival std.err lower 95% CI upper
##   365     65     121     0.409    0.0358      0.345
```



The 1-year probability of survival in this study is 40.92%.

Estimating median survival time

Another quantity often of interest is the **median survival time**.

- We can obtain this directly from the `survfit` object

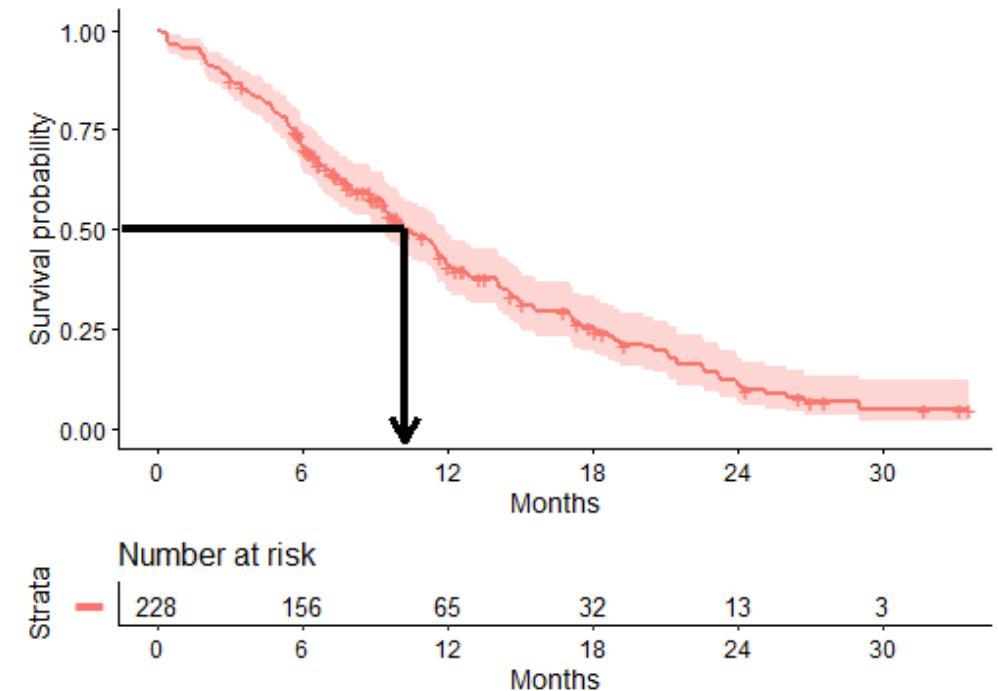
```
km_fit
```

```
## Call: survfit(formula = Surv(time, status) ~ 1, data =
## 
##      n events median 0.95LCL 0.95UCL
## [1,] 228     165     310     285     363
```

We see the median survival time is 310 days

- The 95% confidence interval is also displayed.

Median survival is the time corresponding to a survival probability of 0.5:



Comparing survival times between groups

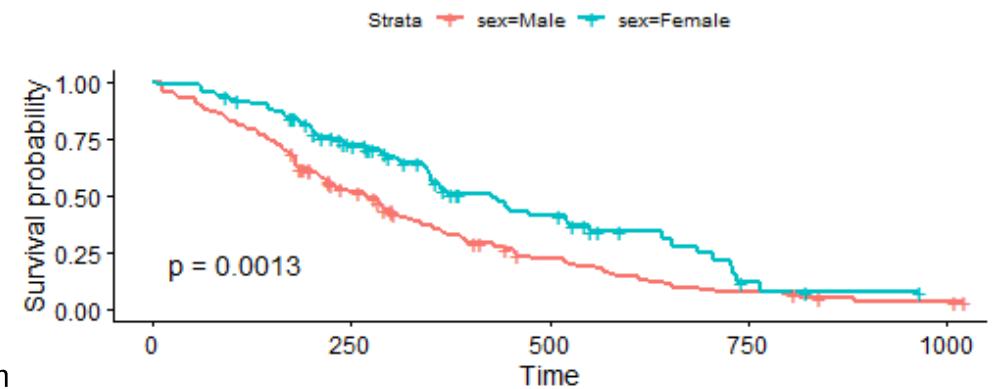
- The **log-rank test** equally weights observations over the entire follow-up time
 - Most common way to compare survival times between groups

We get the log-rank p-value using the `survdiff` function. For example, we can test whether there was a difference in survival time according to sex.

```
survdiff(Surv(time, status) ~ sex, data = lung)
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ sex, data = lung)
##
##          N Observed Expected (0-E)^2/E (0-E)^2/V
## sex=Male   138      112     91.6      4.55     10.3
## sex=Female  90       53     73.4      5.68     10.3
##
##  Chisq= 10.3 on 1 degrees of freedom, p= 0.001
```

```
ggsurvplot(survfit(Surv(time, status) ~ sex,
                     data = lung),
            pval = TRUE)
```



Cox-Regression

Kaplan-Meier curves are good for visualizing differences in survival between two categorical groups.

- Does not generalize well for assessing the effect of **quantitative** variables.

We may want to quantify an effect size for a single variable, or include more than one variable into a regression model to account for the effects of multiple variables.

The Cox regression model can be used to fit univariable and multivariable regression models that have survival outcomes.

$$h(t|X_i) = h_0(t)e^{(\beta_1 X_{i1} + \dots + \beta_p X_{ip})},$$

where $h(t)$ is the hazard and $h_0(t)$ is the baseline hazard

The Cox model relies on the assumption of proportional hazards (PH) across different covariates, i.e., the hazard ratio (HR) is assumed to be constant over time, even though the hazards vary.

Cox Regression - R

Continuing with the `lung` dataset. We can fit regression models for survival data using the `coxph` function from `survival` package.

- It requires a `Surv` object.

```
cox_fit <- coxph(Surv(time, status) ~ sex, data = lung)
cox_fit %>% broom::tidy()
```

```
## # A tibble: 1 x 5
##   term      estimate std.error statistic p.value
##   <chr>      <dbl>     <dbl>     <dbl>    <dbl>
## 1 sexFemale -0.531     0.167    -3.18  0.00149
```

The quantity of interest from a Cox regression model is the **hazard ratio (HR)**

- Represents the ratio of hazards between two groups at any particular point in time
- Interpreted as the instantaneous rate of occurrence of the event of interest in those who are still at risk for the event
- A $\text{HR} < 1$ indicates reduced hazard of death whereas a $\text{HR} > 1$ indicates an increased hazard of death.

From R output, HR is represented by $\exp(\text{estimate}) = e^{-0.5310} = 0.588$

Check Cox assumption

To test for the proportional-hazards (PH) assumption we are going to use the function `cox.zph` from `survival` package.

```
cox.zph(cox_fit)  
  
##          chisq df      p  
## sex      2.86  1 0.091  
## GLOBAL   2.86  1 0.091
```

From the output above, the test is not statistically significant using an α of 5% for the covariate, and the global test is also not statistically significant.

- Therefore, we **can** assume the proportional hazards.