```
### ---------------------------------------------------------------------------
### ---------------------------------------------------------------------------
### SURVIVAL ANALYSIS
### ---------------------------------------------------------------------------
### ---------------------------------------------------------------------------
library(survival)
library(survminer)
```

```
library(dplyr)
```

```
# The core functions we'll use out of the survival package include:
#   Surv()    : Creates a survival object. (can be used as a response of a model)
#   survfit() : Fits a survival curve using a formula.
#   survdiff(): Log-rank test for differences in survival between two or more groups.

# The daset lung of survival package is related to patients with advanced lung cancer
# from the North Central Cancer Treatment Group. Performance scores rate how well
# the patient can perform usual daily activities.
help(lung)
```

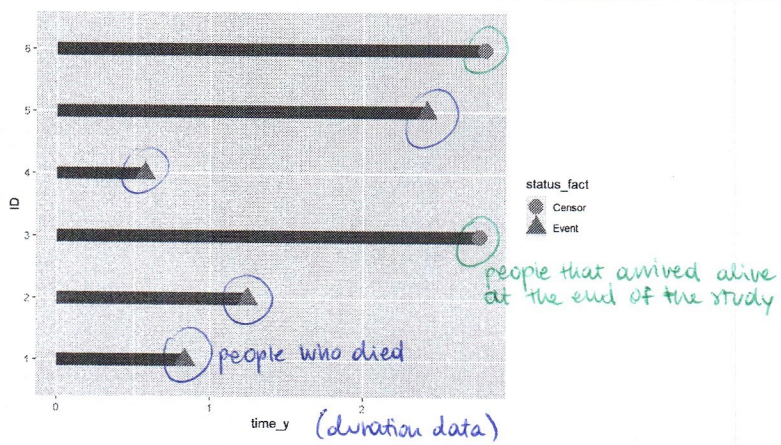
```
# Load the data
data("lung")
dim(lung)
```

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

```
head(lung)
```

```
##   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
## 4    5  210      2  57   1       1       90        60     1150      11
## 5    1  883      2  60   1       0      100        90       NA       0
## 6   12 1022      1  74   1       1       50        80      513       0
```

```
# 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 score (0=good 5=dead)
# ph.karno : Karnofsky performance score (bad=0-good=100) rated by physician
# pat.karno: Karnofsky performance score as rated by patient
# meal.cal : Calories consumed at meals
# wt.loss  : Weight loss in last six months
lung$ID          = factor(seq(1:nrow(lung)))
lung$time_y      = lung$time/365
lung$status_fact = factor(lung$status,labels=(c('Censor','Event')))
lung_subs        = head(lung)

ggplot(data=lung_subs,aes(x=ID,y=time_y)) +
  geom_bar(stat='identity',width=0.2) +
  geom_point(aes(color=status_fact,shape=status_fact),size=6) +
  coord_flip()
```




```
# Investigate the survival probability for all the subjects, by gender and by age.

# ---------------------------------------------------------------------------
# Survival Object
# ---------------------------------------------------------------------------
# The function Surv(time, event) of survival package allows to create a survival object,
# usually used as a response variable in a model formula.
help(Surv)
Surv(lung$time, lung$status==2)
```

we're considering people that are dead

```
##   [1]  306   455  1010+  210   883  1022+  310   361   218   166   170   654
##  [13]  728    71   567   144   613   707    61    88   301    81   624   371
##  [25]  394   520   574   118   390    12   473    26   533   107    53   122
##  [37]  814   965+   93   731   460   153   433   145   583    95   303   519
##  [49]  643   765   735   189    53   246   689    65     5   132   687   345
##  [61]  444   223   175    60   163    65   208  821+   428   230  840+   305
##  [73]   11   132   226   426   705   363    11   176   791    95  196+   167
##  [85]  806+  284   641   147  740+   163   655   239    88   245  588+    30
##  [97]  179   310   477   166  559+   450   364   107   177   156  529+    11
## [109]  429   351    15   181   283   201   524    13   212   524   288   363
## [121]  442   199   550    54   558   207    92    60  551+  543+   293   202
## [133]  353  511+   267  511+   371   387   457   337   201  404+   222    62
## [145]  458+  356+   353   163    31   340   229  444+  315+   182   156   329
## [157]  364+  291   179  376+  384+   268  292+   142  413+  266+   194   320
## [169]  181   285   301+  348   197  382+  303+  296+   180   186   145  269+
## [181]  300+  284+   350  272+  292+  332+   285  259+   110   286   270    81
## [193]  131  225+   269  225+  243+  279+  276+   135    79    59  240+  202+
## [205]  235+  105   224+  239   237+  173+  252+  221+  185+   92+    13  222+
## [217]  192+  183   211+  175+  197+  203+   116  188+  191+  105+  174+  177+
```

```
# ------------------------------------------------------------------------------
# Kaplan-Meier estimator for survival curve
# ------------------------------------------------------------------------------
# The Kaplan-Meier estimator of a survival curve can be computed using the survfit function():
fit = survfit(Surv(time, status==2) ~ 1, data = lung)

# The function survfit() returns a list of variables
names(fit)
```

```
##  [1] "n"         "time"      "n.risk"    "n.event"   "n.censor"  "surv"
##  [7] "std.err"   "cumhaz"    "std.chaz"  "type"      "logse"     "conf.int"
## [13] "conf.type" "lower"     "upper"     "call"
```

```
# including the following components:
# n           : total number of subjects
# time        : the event time points on the curve (t=t*_j)
# n.risk      : the number of subjects at risk at time t
# n.event     : the number of events that occurred at time t
# n.censor    : the number of censored subjects, who exit the risk set at time t
# surv        : the kaplan-meier estimator for survival S(t)
# std.err     : the standard error for S(t)
# lower, upper: Lower and upper confidence limits for the survival curve S(t), respectively
# cumhaz      : the cumulative hazard curve H(t) = - log(S(t))
# std.err     : the standard error for H(t)

# Complete table for Kaplan-Meier estimator
summary(fit)
```

```
## Call: survfit(formula = Surv(time, status == 2) ~ 1, data = lung)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     5    228       1   0.9956 0.00438       0.9871        1.000
##    11    227       3   0.9825 0.00869       0.9656        1.000
##    12    224       1   0.9781 0.00970       0.9592        0.997
##    13    223       2   0.9693 0.01142       0.9472        0.992
##    15    221       1   0.9649 0.01219       0.9413        0.989
##    ..
```

```
# Median Survival time
# The median survival times represents the time at which the survival probability, S(t), is 0.5
median_St = fit$time[fit$surv<=0.5][1]
median_St
```

```
## [1] 310
```

```
#or
surv_median(fit)
```

```
##   strata median lower upper
## 1    All    310   285   363
```

```
#or
print(fit)
```

```
## Call: survfit(formula = Surv(time, status == 2) ~ 1, data = lung)
##
##       n  events  median 0.95LCL 0.95UCL
##     228     165     310     285     363
```

```
# Access to the sort summary table
summary(fit)$table
```

```
##    records      n.max    n.start     events      *rmean *se(rmean)     median
##  228.00000  228.00000  228.00000  165.00000  376.27475   19.70779  310.00000
##    0.95LCL    0.95UCL
##  285.00000  363.00000
```

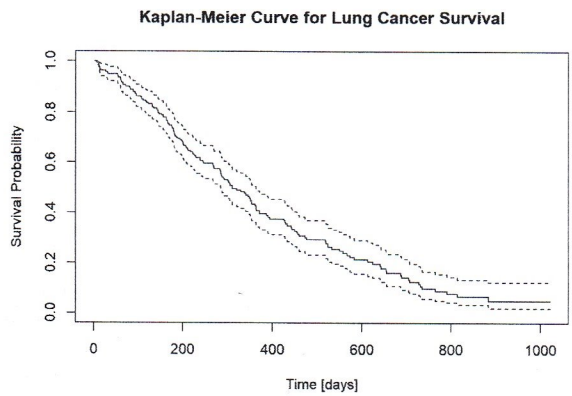
```
# What happens if I try to estimate in a naive fashion the median survival time?
lung %>% filter(status == 2) %>% summarize(median_surv = median(time))
```

```
##   median_surv
## 1         226
```

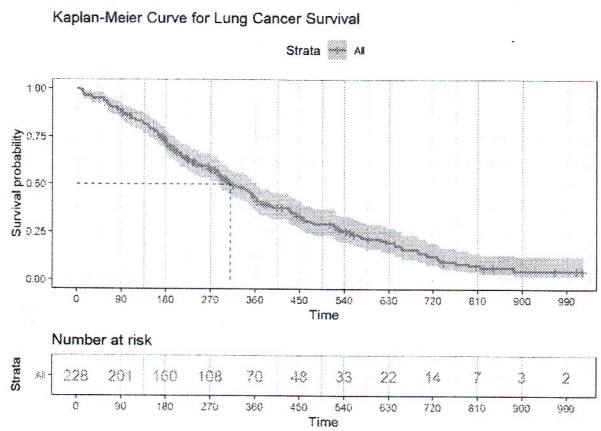
```
# By default, the function print() shows a short summary of the survival curves.
# It prints the number of observations, number of events, the median survival and
# the confidence limits for the median.
print(fit)
```

```
## Call: survfit(formula = Surv(time, status == 2) ~ 1, data = lung)
##
##       n  events  median 0.95LCL 0.95UCL
##     228     165     310     285     363
```

```
# ------------------------------------------------------------------------------
# Kaplan-Meier curve plot
# ------------------------------------------------------------------------------
# To plot the KM estimator you can use the function plot
plot(fit, conf.int = T, xlab='Time [days]', ylab = 'Survival Probability', col='red',
     main="Kaplan-Meier Curve for Lung Cancer Survival")
```



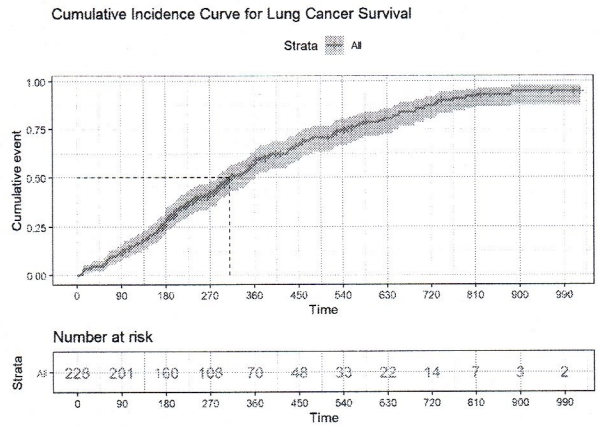

```
# For a better visualization use ggsurvplot() function [package survminer]:
ggsurvplot(fit,
           risk.table = TRUE,         # Add risk table
           risk.table.col = "strata", # Change risk table color by groups
           surv.median.line = "hv",   # Specify median survival
           ggtheme = theme_bw(),      # Change ggplot2 theme
           break.time.by=90,
           title="Kaplan-Meier Curve for Lung Cancer Survival")
```



```
# At time zero, the survival probability is 1.0 (or 100% of the participants are alive).
# At time 180 (after 6 months), the probability of survival is approximately 0.75 (or 75%).
# The median survival is approximately 310 days.
# After 540 days (1 year and a half), the survival probability is below 0.25 (25%).

# ------------------------------------------------------------------------------------
# Cumulative incidence function
# ------------------------------------------------------------------------------------
# The cumulative incidence, or cumulative failure probability (CFP), shows the cumulative
# probabilities of experiencing the event of interest and it is computed as CFP(t) = P(T<t)
# so can be estimated as 1-S(t):
cumulative_incidence = 1 - fit$surv
head(cumulative_incidence)
```

```
## [1] 0.004385965 0.017543860 0.021929825 0.030701754 0.035087719 0.039473684
```

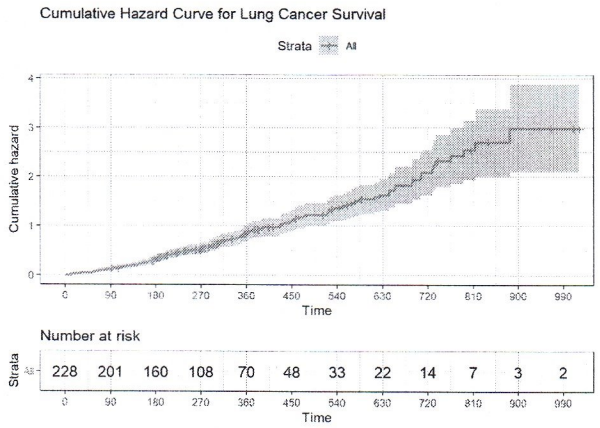
```
# CFP can be visualized using the ggsurvplot() function [package survminer],
# specifying the option fun='event':
ggsurvplot(fit,
           risk.table = TRUE,         # Add risk table
           risk.table.col = "strata", # Change risk table color by groups
           surv.median.line = "hv",   # Specify median survival
           ggtheme = theme_bw(),      # Change ggplot2 theme
           break.time.by=90,
           fun='event',
           title="Cumulative Incidence Curve for Lung Cancer Survival")
```




```
# ------------------------------------------------------------------------------------
# Cumulative hazard function
# ------------------------------------------------------------------------------------
# The cummulative hazard is commonly used to estimate the hazard probability.
# It's defined as H(t) = ???log(S(t)). The cumulative hazard (H(t)) can be interpreted
# as the cumulative force of mortality. In other words, it corresponds to the number
# of events that would be expected for each individual by time t if the event were a
# repeatable process.

# The cumulative hazard H(t) = -log(S(t)) is computed by function survdiff() using the
# Nelson-Aalen cumulative hazard rate estimator and it is given by:
H = fit$cumhaz
head(H)
```

```
## [1] 0.004385965 0.017601824 0.022066110 0.031034720 0.035559606 0.040105061
```

```
# H(t) can be visualized using the ggsurvplot() function [package survminer],
# specifying the option fun='cumhaz':
ggsurvplot(fit,
           risk.table = TRUE,    # Add risk table
           ggtheme = theme_bw(), # Change ggplot2 theme
           break.time.by=90,
           fun='cumhaz',
           title="Cumulative Hazard Curve for Lung Cancer Survival")
```




```
curves = data.frame('time' = fit$time,
                    'Survival' = fit$surv,
                    'Cum_incidence' = 1 - fit$surv,
                    'Cum_hazard' = fit$cumhaz)
head(curves)
```

```
##   time  Survival Cum_incidence  Cum_hazard
## 1    5 0.9956140   0.004385965 0.004385965
## 2   11 0.9824561   0.017543860 0.017601824
## 3   12 0.9780702   0.021929825 0.022066110
## 4   13 0.9692982   0.030701754 0.031034720
## 5   15 0.9649123   0.035087719 0.035559606
## 6   26 0.9605263   0.039473684 0.040105061
```

```
# ------------------------------------------------------------------------------
# Kaplan-Meier Curves by gender
# ------------------------------------------------------------------------------
# We want to consider now the gender groups and investigate if there is a difference
# in terms of survival among the two groups.
fit.sex = survfit(Surv(time, status) ~ sex, data = lung)

# By default, the function print() shows a short summary of the survival curves.
# It prints the number of observations, number of events, the median survival and
# the confidence limits for the median for both groups:
print(fit.sex)
```

```
## Call: survfit(formula = Surv(time, status) ~ sex, data = lung)
##
##         n events median 0.95LCL 0.95UCL
## sex=1 138    112    270     212     310
## sex=2  90     53    426     348     550
```

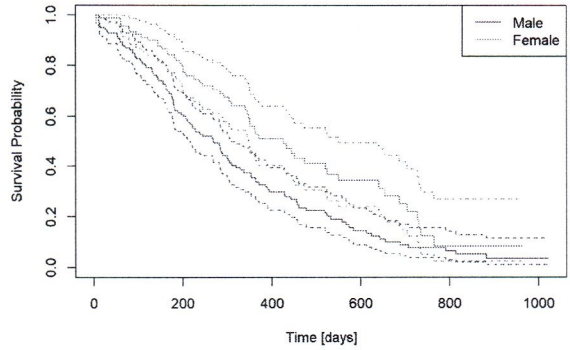
```
# Summary of survival curves
summary(fit.sex)$table
```

```
##       records n.max n.start events   *rmean *se(rmean) median 0.95LCL 0.95UCL
## sex=1     138   138     138    112 326.0841   22.91156    270     212     310
## sex=2      90    90      90     53 460.6473   34.68985    426     348     550
```

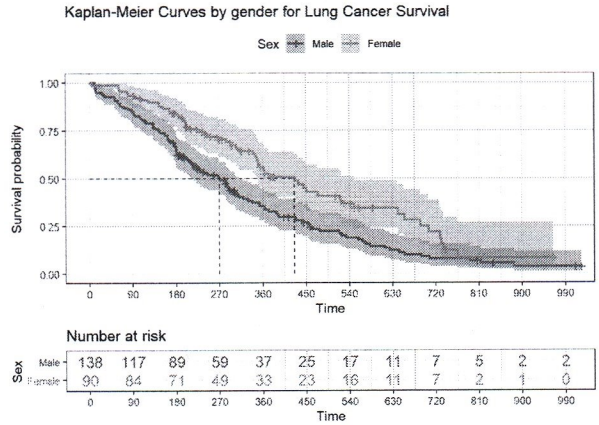
```
# Complete KM estimation tables
summary(fit.sex)
```

```
## Call: survfit(formula = Surv(time, status) ~ sex, data = lung)
##
##                 sex=1
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    11    138       3   0.9783  0.0124       0.9542        1.000
##    12    135       1   0.9710  0.0143       0.9434        0.999
##    13    134       2   0.9565  0.0174       0.9231        0.991
##    15    132       1   0.9493  0.0187       0.9134        0.987
##    26    131       1   0.9420  0.0199       0.9038        0.982
##    ..
```

```
# ------------------------------------------------------------------------------
# Kaplan-Meier plot by gender
# ------------------------------------------------------------------------------
plot(fit.sex, conf.int = T, xlab='Time [days]', ylab = 'Survival Probability',
     col=c("dodgerblue2", "orchid2"))
legend('topright', legend=c('Male','Female'), lty=c(1,1), col=c("dodgerblue2", "orchid2"))
```




```
ggsurvplot(fit.sex, conf.int = T,
           risk.table = TRUE,          # Add risk table
           risk.table.col = "strata", # Change risk table color by groups
           surv.median.line = "hv",   # Specify median survival
           ggtheme = theme_bw(),      # Change ggplot2 theme
           break.time.by=90,
           legend.labs=c("Male", "Female"), legend.title="Sex",
           palette=c("dodgerblue2", "orchid2"),
           title="Kaplan-Meier Curves by gender for Lung Cancer Survival")
```



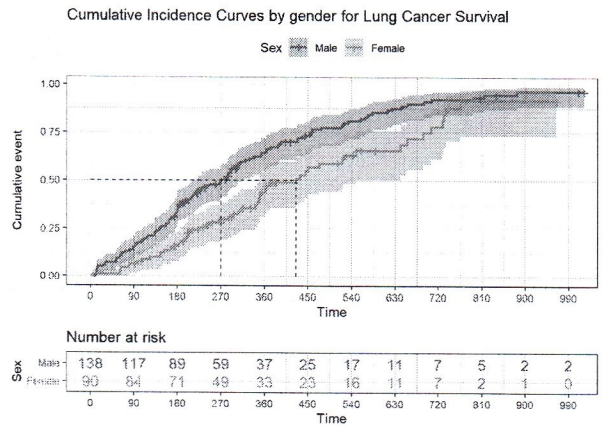

```
# At time zero, the survival probability is 1.0 (or 100% of the participants are alive).
# At time 180 days (6 months), the probability of survival is approximately 0.645 (or 64.5%)
# for males (sex=1) and 0.842 (or 84.2%) for females (sex=2).
# The median survival is approximately 270 days for males and 426 days for females,
# suggesting a good survival for females compared to males.

# Survival probability at time 180 days
summary(fit.sex, times=180)
```

```
## Call: survfit(formula = Surv(time, status) ~ sex, data = lung)
##
##                sex=1
##         time       n.risk      n.event     survival      std.err lower 95% CI
##     180.0000      89.0000      49.0000       0.6445       0.0408       0.5693
## upper 95% CI
##       0.7296
##
##                sex=2
##         time       n.risk      n.event     survival      std.err lower 95% CI
##     180.0000      71.0000      14.0000       0.8424       0.0387       0.7699
## upper 95% CI
##       0.9217
```

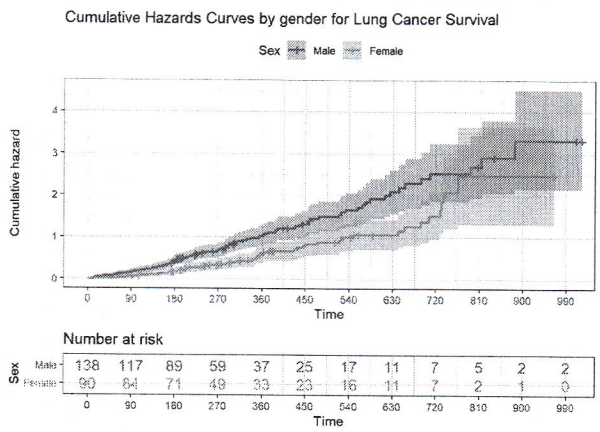
```
# Survival probabilities every six months
summary(fit.sex, times=seq(0,365*3,182.5))
```

```
## Call: survfit(formula = Surv(time, status) ~ sex, data = lung)
##
##                sex=1
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     0    138       0   1.0000  0.0000       1.0000        1.000
##   182     86      51   0.6298  0.0412       0.5541        0.716
##   365     35      34   0.3361  0.0434       0.2609        0.433
##   548     17      14   0.1897  0.0385       0.1275        0.282
##   730      7      10   0.0781  0.0276       0.0390        0.156
##   912      2       3   0.0357  0.0216       0.0109        0.117
##
##                sex=2
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     0     90       0   1.0000  0.0000       1.0000        1.000
##   182     70      15   0.8305  0.0399       0.7559        0.913
##   365     30      21   0.5265  0.0597       0.4215        0.658
##   548     15       8   0.3678  0.0630       0.2628        0.515
##   730      6       6   0.1872  0.0621       0.0978        0.359
##   912      1       3   0.0832  0.0499       0.0257        0.270
```

```
# ------------------------------------------------------------------------------
# Cumulative incidence function by gender
# ------------------------------------------------------------------------------
ggsurvplot(fit.sex, conf.int = T,
           risk.table = TRUE,          # Add risk table
           risk.table.col = "strata", # Change risk table color by groups
           surv.median.line = "hv",    # Specify median survival
           ggtheme = theme_bw(),       # Change ggplot2 theme
           break.time.by=90,
           legend.labs=c("Male", "Female"), legend.title="Sex",
           palette=c("dodgerblue2", "orchid2"),
           title="Cumulative Incidence Curves by gender for Lung Cancer Survival",
           fun='event')
```




```
# ------------------------------------------------------------------------------
# Cumulative hazard function by gender
# ------------------------------------------------------------------------------
ggsurvplot(fit.sex, conf.int = T,
           risk.table = TRUE,          # Add risk table
           risk.table.col = "strata", # Change risk table color by groups
           ggtheme = theme_bw(),       # Change ggplot2 theme
           break.time.by=90,
           legend.labs=c("Male", "Female"), legend.title="Sex",
           palette=c("dodgerblue2", "orchid2"),
           title="Cumulative Hazards Curves by gender for Lung Cancer Survival",
           fun='cumhaz')
```



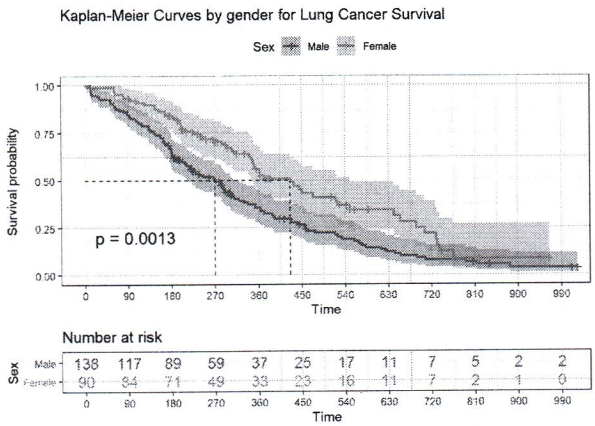
```
# There appears to be a survival advantage for female with lung cancer compare to male.
# However, to evaluate whether this difference is statistically significant requires a
# formal statistical test --> log-rank test.

# ------------------------------------------------------------------------------
# Log-rank test for gender groups
# ------------------------------------------------------------------------------
# The log-rank test is the most widely used method of comparing two or more survival curves.
# The null hypothesis is that there is no difference in survival between the two groups.
# The log rank test is a non-parametric test, which makes no assumptions about the survival
# distributions. Essentially, the log rank test compares the observed number of events in
# each group to what would be expected if the null hypothesis were true
# (i.e., if the survival curves were identical).
# The log rank statistic is approximately distributed as a chi-square test statistic.

# The function survdiff() [in survival package] can be used to compute log-rank test comparing
# two or more survival curves.
# survdiff() can be used as follow:
survdiff(Surv(time, status) ~ sex, data = lung)
```

How to perform tests?

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

Small p-value ⟹ the two curves are different
high p-value ⟹ the two curves are the same

```
# In the ggsurvplot() function we can specify the factor pval=T
# which return the p-values of the log-rank test
ggsurvplot(fit.sex, conf.int = T,
           risk.table = TRUE,          # Add risk table
           risk.table.col = "strata", # Change risk table color by groups
           surv.median.line = "hv",   # Specify median survival
           ggtheme = theme_bw(),      # Change ggplot2 theme
           break.time.by=90,
           legend.labs=c("Male", "Female"), legend.title="Sex",
           palette=c("dodgerblue2", "orchid2"),
           title="Kaplan-Meier Curves by gender for Lung Cancer Survival",
           pval=T)
```



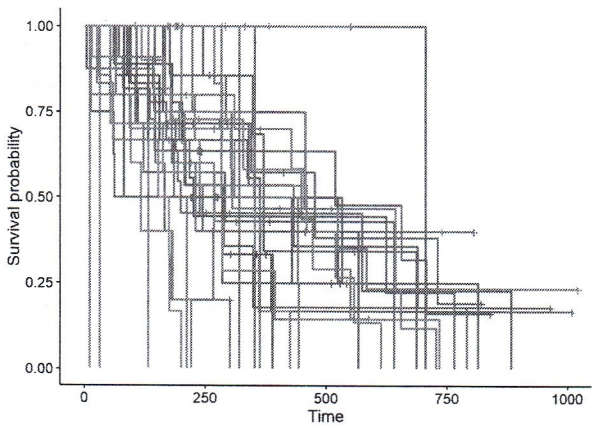

```
# The log rank test for difference in survival gives a p-value of p = 0.0013,
# indicating that the gender groups differ significantly in survival.

# ------------------------------------------------------------------------------
# Hazard Ratio
# ------------------------------------------------------------------------------
# To quantify the difference in the survivals we can compute the hazard ratio, i.e.
# the ratio between the death hazard of the first group vs the other one.

# From the output of the Log-rank test  we can extract the number of observed and expected
# deaths in male and female groups:
#     - observed deaths in males: 112
#     - expected deaths in males: 91.6
#     - observed deaths in males: 53
#     - expected deaths in males: 73.4
# Therefore, the death hazard ratio of males vs females is:
hazard_ratio = (112/91.6)/(53/73.4)
hazard_ratio
```

```
## [1] 1.693334
```

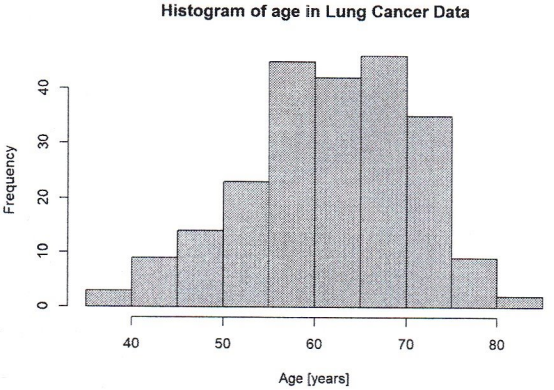
```
# HR = 1.693 > 1 indicating that the risk of deaths in males is 1.693 times
#               times the risk of death in females.
# Males have lower survival probability than males.
# Being a female is a protective factor.

# ------------------------------------------------------------------------------
# Kaplan-Meier plot by age
# ------------------------------------------------------------------------------
# We want to consider now patient's age and analyse the impact on survival.
# If we use a continuous variable in a group KM survival estimation, each value
# of the variable is considered as a different group:
fit.age = survfit(Surv(time, status) ~ age, data=lung)
print(fit.age)
```

```
## Call: survfit(formula = Surv(time, status) ~ age, data = lung)
##
##         n events median 0.95LCL 0.95UCL
## age=39  2      0     NA      NA      NA
## age=40  1      1    132      NA      NA
## age=41  1      0     NA      NA      NA
## age=42  1      0     NA      NA      NA
## age=43  1      0     NA      NA      NA
## ..
## age=76  5      5    116      95      NA
## age=77  2      0     NA      NA      NA
## age=80  2      2    323     283      NA
## age=81  1      1     11      NA      NA
## age=82  1      1     31      NA      NA
```

```
ggsurvplot(fit.age, conf.int = F, risk.table.col = "strata", legend='none')
```




```
# We can not just do this, because we'll get a separate curve for every unique value of age!
# We have to categorize data in some way!

# ------------------------------------------------------------------------------
# Categorizing for KM plots
# ------------------------------------------------------------------------------
# One thing we might do is an attempt to categorize a continuous variable into different groups.
# But, how we make that cut is meaningful!
# Let's get the median age in the dataset, and plot a histogram showing the distribution of age.
hist(lung$age, xlab='Age [years]', main='Histogram of age in Lung Cancer Data')
```




```
summary(lung$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   39.00   56.00   63.00   62.45   69.00   82.00
```

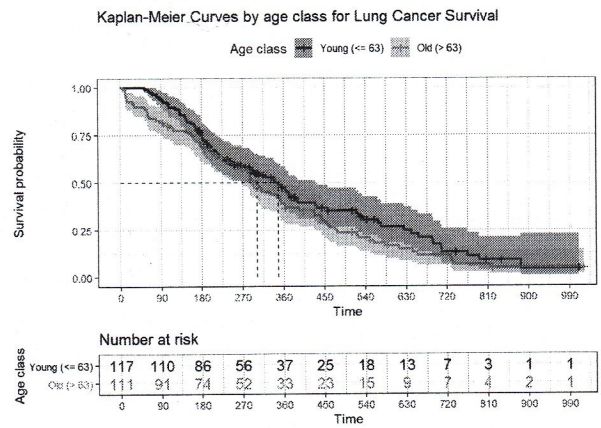
```
# ------------------------------------------------------------------------------
# Age cut at 63 years old
# ------------------------------------------------------------------------------
# Create a categorical variable on lung$age with cut pounts at 0, 63 (the median),
# and +Infinity (no upper limit), labelling as 'young' subjects with age less or equal
# than 63 and as 'old' subject aged more than 63.
lung$agecat63 = cut(lung$age, breaks=c(0, 63, Inf), labels=c("young", "old"))
head(lung)
```

```
##   inst time status age sex ph.ecog ph.karno pat.karno meal.cal wt.loss ID
## 1    3  306      2  74   1       1       90       100     1175      NA  1
## 2    3  455      2  68   1       0       90        90     1225      15  2
## 3    3 1010      1  56   1       0       90        90       NA      15  3
## 4    5  210      2  57   1       1       90        60     1150      11  4
## 5    1  883      2  60   1       0      100        90       NA       0  5
## 6   12 1022      1  74   1       1       50        80      513       0  6
##      time_y status_fact agecat63
## 1 0.8383562       Event      old
## 2 1.2465753       Event      old
## 3 2.7671233      Censor    young
## 4 0.5753425       Event    young
## 5 2.4191781       Event    young
## 6 2.8000000      Censor      old
```

```
# What happens when we make a KM plot with this new categorization?
fit.age = survfit(Surv(time, status) ~ agecat63, data=lung)
print(fit.age)
```

```
## Call: survfit(formula = Surv(time, status) ~ agecat63, data = lung)
##
##                  n events median 0.95LCL 0.95UCL
## agecat63=young 117     80    348     268     429
## agecat63=old   111     85    301     269     361
```

```
ggsurvplot(fit.age, conf.int = T,
           risk.table = TRUE,         # Add risk table
           risk.table.col = "strata", # Change risk table color by groups
           surv.median.line = "hv",   # Specify median survival
           ggtheme = theme_bw(),      # Change ggplot2 theme
           break.time.by=90,
           legend.labs=c("Young (<= 63)","Old (> 63)"), legend.title="Age class",
           palette=c("darkblue","cyan3"),
           title="Kaplan-Meier Curves by age class for Lung Cancer Survival")
```



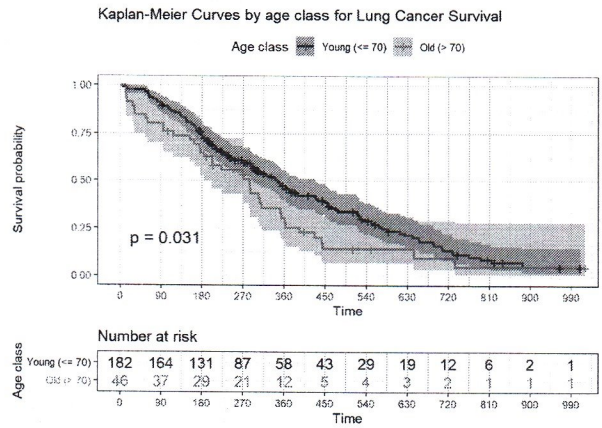

```
# It looks like there's some differences in the curves between "old" and "young" patients,
# with older patients having slightly worse survival odds.
# Is there statistical evidence for that difference?
survdiff(Surv(time, status) ~ agecat63, data=lung)
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ agecat63, data = lung)
##
##                  N Observed Expected (O-E)^2/E (O-E)^2/V
## agecat63=young 117       80     88.8     0.865      1.88
## agecat63=old   111       85     76.2     1.007      1.88
##
##  Chisq= 1.9  on 1 degrees of freedom, p= 0.2
```

```
# p=0.2, the difference in survival between those younger than 63 and older than 63 is NOT significant.

# ------------------------------------------------------------------------------
# Age cut at 70 years old
# ------------------------------------------------------------------------------
# What happens if we chose a different cut point?
# Let's try with 70 years old, which is roughly the cutoff for the upper quartile of
# the age distribution.
lung$agecat70 = cut(lung$age, breaks=c(0, 70, Inf), labels=c("young", "old"))
fit.age       = survfit(Surv(time, status) ~ agecat70, data=lung)
print(fit.age)
```

```
## Call: survfit(formula = Surv(time, status) ~ agecat70, data = lung)
##
##                  n events median 0.95LCL 0.95UCL
## agecat70=young 182    127    345     291     429
## agecat70=old    46     38    283     201     353
```

```
ggsurvplot(fit.age, conf.int = T,
           risk.table = TRUE,         # Add risk table
           risk.table.col = "strata", # Change risk table color by groups
           ggtheme = theme_bw(),      # Change ggplot2 theme
           break.time.by=90,
           legend.labs=c("Young (<= 70)","Old (> 70)"), legend.title="Age class",
           palette=c("darkblue","cyan3"),
           title="Kaplan-Meier Curves by age class for Lung Cancer Survival",
           pval=T)
```




```
survdiff(Surv(time, status) ~ agecat70, data=lung)
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ agecat70, data = lung)
##
##                  N Observed Expected (O-E)^2/E (O-E)^2/V
## agecat70=young 182      127    137.3     0.773      4.64
## agecat70=old    46       38     27.7     3.829      4.64
##
##  Chisq= 4.6  on 1 degrees of freedom, p= 0.03
```

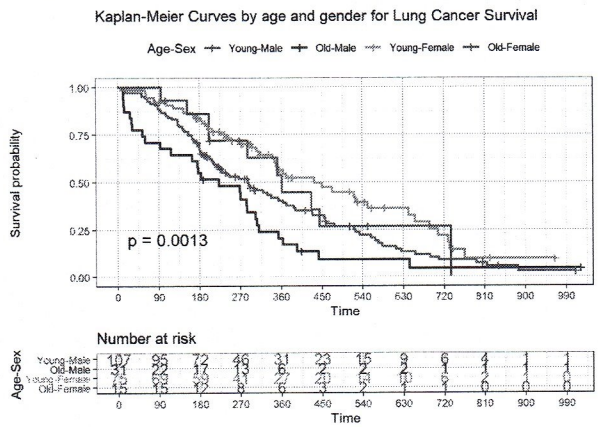
```
# p=0.03, the difference in survival between those younger than 70 and older
# than 70 is significant.

# ------------------------------------------------------------------------------
# Hazard Ratio
# ------------------------------------------------------------------------------
# From the output of the log-rank test  we can extract the number of observed and expected
# deaths in the groups of younger than 70 and older than 70:
#     - observed deaths in young: 127
#     - expected deaths in young: 137.3
#     - observed deaths in old: 38
#     - expected deaths in old: 27.7
# Therefore, the death hazard ratio of young vs old is:
hazard_ratio = (127/137.3)/(38/27.7)
hazard_ratio
```

```
## [1] 0.674263
```

```
# HR = 0.674 < 1 indicating that the risk of deaths in younger than 70 years old is
# to be 0.674 times the risk in old than 70.
# Being young is a protective factor.

# The log-rank test on the Kaplan-Meier plot can change depending on how you categorize
# your continuous variable. Indeed, with the log-rank test we are asking: "Are there
# differences in survival between those less than 70 and those greater than 70 years old?"

# If we want to investigate the effect of continuous age on survival, without depending on how
# we categorize the variable, we have to use a survival model, for example a Cox regression
# model, which analyzes the continuous variable over the whole range of its distribution.
# A survival regression model is asking: "What is the effect of the variable on survival?".

# ------------------------------------------------------------------------------
# Kaplan-Meier curves by gender and age
# ------------------------------------------------------------------------------
# We now consider both gender and age categorized with cut point 70 years old.
fit.sex.age = survfit(Surv(time, status) ~ sex + agecat70, data=lung)
print(fit.sex.age)
```

```
## Call: survfit(formula = Surv(time, status) ~ sex + agecat70, data = lung)
##
##                          n events median 0.95LCL 0.95UCL
## sex=1, agecat70=young 107     84    286     218     371
## sex=1, agecat70=old    31     28    222     116     306
## sex=2, agecat70=young  75     43    433     345     654
## sex=2, agecat70=old    15     10    361     285      NA
```

```
ggsurvplot(fit.sex.age, conf.int = F,
           risk.table = TRUE,         # Add risk table
           risk.table.col = "strata", # Change risk table color by groups
           ggtheme = theme_bw(),      # Change ggplot2 theme
           break.time.by=90,
           legend.labs=c("Young-Male","Old-Male", "Young-Female","Old-Female"),
           legend.title="Age-Sex",
           palette=c("dodgerblue2","dodgerblue4","orchid2","orchid4"),
           title="Kaplan-Meier Curves by age and gender for Lung Cancer Survival",
           pval=T)
```




```
survdiff(Surv(time, status) ~ sex + agecat70, data=lung)
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ sex + agecat70, data = lung)
##
##                          N Observed Expected (O-E)^2/E (O-E)^2/V
## sex=1, agecat70=young 107       84     75.4     0.993     1.839
## sex=1, agecat70=old    31       28     16.2     8.536     9.561
## sex=2, agecat70=young  75       43     61.9     5.795     9.351
## sex=2, agecat70=old    15       10     11.5     0.189     0.204
##
##  Chisq= 15.7  on 3 degrees of freedom, p= 0.001
```

```
# p=0.001 -> evidence for difference in survival curves

# Is there evidence for difference in males survival curves of younger/older than 70?
survdiff(Surv(time, status) ~ agecat70, data=lung[lung$sex==1,])
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ agecat70, data = lung[lung$sex ==
##     1, ])
##
##                  N Observed Expected (O-E)^2/E (O-E)^2/V
## agecat70=young 107       84     91.9     0.673      3.79
## agecat70=old    31       28     20.1     3.068      3.79
##
##  Chisq= 3.8  on 1 degrees of freedom, p= 0.05
```

```
# p = 0.05 -> no evidence for a survival difference  between young-males and old-males

# Is there evidence for difference in females survival curves of younger/older than 70?
survdiff(Surv(time, status) ~ agecat70, data=lung[lung$sex==2,])
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ agecat70, data = lung[lung$sex ==
##     2, ])
##
##                 N Observed Expected (O-E)^2/E (O-E)^2/V
## agecat70=young 75       43    44.61    0.0584     0.372
## agecat70=old   15       10     8.39    0.3105     0.372
##
##  Chisq= 0.4  on 1 degrees of freedom, p= 0.5
```

```
# p = 0.5 -> no evidence for a survival difference  between young-females and old-females

# Is there evidence for difference in survival curves of males and females for younger than 70?
survdiff(Surv(time, status) ~ sex, data=lung[lung$agecat70=='young',])
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ sex, data = lung[lung$agecat70 ==
##     "young", ])
##
##         N Observed Expected (O-E)^2/E (O-E)^2/V
## sex=1 107       84     69.6      2.97      6.64
## sex=2  75       43     57.4      3.61      6.64
##
##  Chisq= 6.6  on 1 degrees of freedom, p= 0.01
```

```
# p = 0.01 -> evidence for a survival difference  between young-males and young-females

# Is there evidence for differencein survival curves of males and females for older than 70?
survdiff(Surv(time, status) ~ sex, data=lung[lung$agecat70=='old',])
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ sex, data = lung[lung$agecat70 ==
##     "old", ])
##
##        N Observed Expected (O-E)^2/E (O-E)^2/V
## sex=1 31       28     21.9      1.71      4.15
## sex=2 15       10     16.1      2.33      4.15
##
##  Chisq= 4.1  on 1 degrees of freedom, p= 0.04
```

```
# p = 0.04 -> evidence for a survival difference  between old-males and old-females

# ------------------------------------------------------------------------------
# COX PH MODEL
# ------------------------------------------------------------------------------
# The Cox proportional-hazards model (Cox, 1972) is essentially a regression model commonly
# used statistical in medical research for investigating the association between the survival
# time of patients and one or more predictor variables.

# In the previous practical session, we described the basic concepts of
# survival analyses and methods for analyzing and summarizing survival data, including:
# - the definition of hazard and survival functions,
# - the construction of Kaplan-Meier survival curves for different patient groups
# - the logrank test for comparing two or more survival curves

# The Kaplan-Meier curves and logrank tests are examples of univariate analysis.
# They describe the survival according to one factor under investigation, but ignore the impact
# of any others. Additionally, Kaplan-Meier curves and logrank tests are useful only when the
# predictor variable is categorical (e.g.: treatment A vs treatment B; males vs females).
# They don't work easily for quantitative predictors such as gene expression, weight, or age.

# An alternative method is the Cox proportional hazards regression analysis, which works for
# both quantitative predictor variables and for categorical variables. Furthermore, the Cox
# regression model extends survival analysis methods to assess simultaneously the effect of
# several risk factors on survival time.

# The function coxph()[in survival package] can be used to compute the Cox proportional-hazards
# regression model in R.
help(coxph)

# The simplified format is as follow:
# coxph(formula, data, method)

# formula : is linear model with a survival object as the response variable.
#           [Survival object is created using the function Surv(time, event)]
# data    : a data frame containing the variables
# method  : is used to specify how to handle ties.
```

```
### ---------------------------------------------------------------------------
### ---------------------------------------------------------------------------
### Exercise 1
### ---------------------------------------------------------------------------
### ---------------------------------------------------------------------------
# The daset lung of survival package is related to patients with advanced lung cancer
# from the North Central Cancer Treatment Group. Performance scores rate how well
# the patient can perform usual daily activities.
help(lung)

# Load the data
data("lung")
dim(lung)
```

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

```
head(lung)
```

```
##   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
## 4    5  210      2  57   1       1       90        60     1150      11
## 5    1  883      2  60   1       0      100        90       NA       0
## 6   12 1022      1  74   1       1       50        80      513       0
```

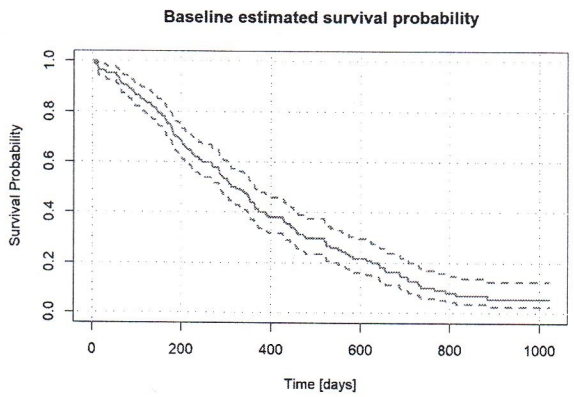
```
# inst     : Institution code
# time     : Survival time in days
# status   : censoring status 1=censored, 2=dead
# age      : Age in years
# sex      : Male=1 Female=2
# ph.ecog  : ECOG performance score (0=good 5=dead)
# ph.karno : Karnofsky performance score (bad=0-good=100) rated by physician
# pat.karno: Karnofsky performance score as rated by patient
# meal.cal : Calories consumed at meals
# wt.loss  : Weight loss in last six months

# ---------------------------------------------------------------------------
# Univariate Cox regression
# ---------------------------------------------------------------------------
# Considr the continuous variable age and fit a Cox regression model.
cox.age = coxph(Surv(time, status) ~ age, data = lung)
cox.age
```

```
## Call:
## coxph(formula = Surv(time, status) ~ age, data = lung)
##
##         coef exp(coef) se(coef)     z      p
## age 0.018720  1.018897 0.009199 2.035 0.0419
##
## Likelihood ratio test=4.24  on 1 df, p=0.03946
## n= 228, number of events= 165
```

```
# The function summary() for Cox models produces a more complete report:
summary(cox.age)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ age, data = lung)
##
##   n= 228, number of events= 165
##
##         coef exp(coef) se(coef)     z Pr(>|z|)
## age 0.018720  1.018897 0.009199 2.035   0.0419 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##     exp(coef) exp(-coef) lower .95 upper .95
## age     1.019     0.9815     1.001     1.037
##
## Concordance= 0.55  (se = 0.025 )
## Likelihood ratio test= 4.24  on 1 df,   p=0.04
## Wald test            = 4.14  on 1 df,   p=0.04
## Score (logrank) test = 4.15  on 1 df,   p=0.04
```

```
# The Cox regression results can be interpreted as follow:
#
# 1) STATISTICAL SIGNIFICANCE
#     The column marked "z" gives the Wald statistic value. It corresponds to the
#     ratio of each regression coefficient to its standard error (z = coef/se(coef)).
#     The wald statistic evaluates, whether the beta coefficient of a given
#     variable is statistically significantly different from 0. From the output above,
#     we can conclude that the variable age is statistically significant at 5%.
#
#
# 2) THE REGRESSION COEFFICIENTS
#     The second feature to note in the Cox model results is the the sign of the
#     regression coefficients (coef).
#     A positive sign means that the hazard (risk of death) is higher,
#     and thus the prognosis is worse, for subjects with higher values of that variable.
#     The beta coefficient for sex = +0.0187 indicates that yonger patients have lower
#     risk of death (higher survival rates) than elder ones, in these data.
#
# 3) HAZARD RATIO & CONFIDENCE INTERVAL
#     The exponentiated coefficients (exp(coef) = exp(0.0187) = 1.019), also known as hazard
#     ratios, give the effect size of covariates.
#     For example, the increase of 1 unit (1 year) in the age increase the hazard of 1.9%.
#     The summary output also gives upper and lower 95% confidence intervals for the hazard
#     ratio (exp(coef)), lower 95% bound = 1.001, upper 95% bound = 1.037.
#     Being younger is associated with good prognostic.
#     Similarly, the increase of 10 units (10 years) in the age increase the hazard of a
#     factor exp(0.0187*10)=1.2056, or 20.5%.
#
# 4) GLOBAL STATISTICAL SIGNIFICANCE OF THE MODEL
#     Finally, the output gives p-values for three alternative tests for overall significance
#     of the model: The likelihood-ratio test, Wald test, and score logrank statistics.
#     These three methods are asymptotically equivalent. For large enough N, they will give
#     similar results. For small N, they may differ somewhat. The Likelihood ratio test has
#     better behavior for small sample sizes, so it is generally preferred.

# ---------------------------------------------------------------------------------
# Visualizing the estimated distribution of survival times
# ---------------------------------------------------------------------------------
# Having fit a Cox model to the data, it's possible to visualize the predicted survival
# proportion at any given point in time for a particular risk group. The function survfit()
# estimates the survival proportion, by default at the mean values of covariates.

# Plot the baseline survival function S_0(t)
plot(survfit(cox.age, data=lung),
     col="darkorange2", lwd=2, lty=1,
     xlab='Time [days]', ylab='Survival Probability',
     main='Baseline estimated survival probability')
grid()
```




```
# We may wish to display how estimated survival depends upon the value of the covariate
# of interest. Consider that, we want to assess the impact of the age on the estimated
# survival probability. In this case, we construct a new data frame with M rows, one for
# each different value of age we are interested in (usually 2 or 3).

# Suppose we want to consider ages equal to 50, 65 and 80. We create the new data:
summary(lung$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   39.00   56.00   63.00   62.45   69.00   82.00
```

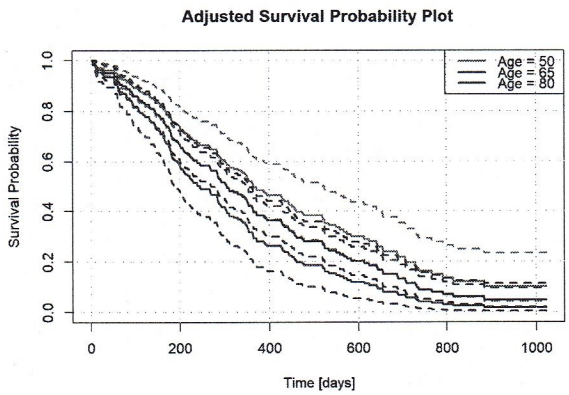
```
age_df = with(lung, data.frame(age = c(50,65,80)))
age_df
```

```
##   age
## 1  50
## 2  65
## 3  80
```

```
# This data frame is passed to survfit() via the newdata argument to estimate survival:
fit.age = survfit(cox.age, newdata = age_df)
fit.age
```

```
## Call: survfit(formula = cox.age, newdata = age_df)
##
##     n events median 0.95LCL 0.95UCL
## 1 228    165    364     306     524
## 2 228    165    305     270     353
## 3 228    165    245     189     350
```

```
# Estimated survival curves plot:
x11()
plot(fit.age, conf.int=T,
     col=c("dodgerblue2","navy","darkmagenta"), lwd=2, lty=1,
     xlab='Time [days]', ylab='Survival Probability',
     main='Adjusted Survival Probability Plot')
grid()
legend('topright', c("Age = 50", "Age = 65", "Age = 80"),
       lty=c(1,1,1), lwd=c(2,2,2), col=c("dodgerblue2","navy","darkmagenta"))
```




```
# ------------------------------------------------------------------------------
# Multivariate Cox regression
# ------------------------------------------------------------------------------
# Now, we want to describe how different factors jointly impact on survival.
graphics.off()
help(lung)
# To answer to this question, we'll perform a multivariate Cox regression analysis
# with covariates: age, sex, Karnofsky performance score rated by physician and weight loss.

# Check if you categorical covariates are considered factors:
summary(lung)
```

```
##       inst             time            status           age
##  Min.   : 1.00   Min.   :   5.0   Min.   :1.000   Min.   :39.00
##  1st Qu.: 3.00   1st Qu.: 166.8   1st Qu.:1.000   1st Qu.:56.00
##  Median :11.00   Median : 255.5   Median :2.000   Median :63.00
##  Mean   :11.09   Mean   : 305.2   Mean   :1.724   Mean   :62.45
##  3rd Qu.:16.00   3rd Qu.: 396.5   3rd Qu.:2.000   3rd Qu.:69.00
##  Max.   :33.00   Max.   :1022.0   Max.   :2.000   Max.   :82.00
##  NA's   :1
##       sex           ph.ecog          ph.karno        pat.karno
##  Min.   :1.000   Min.   :0.0000   Min.   : 50.00   Min.   : 30.00
##  1st Qu.:1.000   1st Qu.:0.0000   1st Qu.: 75.00   1st Qu.: 70.00
##  Median :1.000   Median :1.0000   Median : 80.00   Median : 80.00
##  Mean   :1.395   Mean   :0.9515   Mean   : 81.94   Mean   : 79.96
##  3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.: 90.00   3rd Qu.: 90.00
##  Max.   :2.000   Max.   :3.0000   Max.   :100.00   Max.   :100.00
##                  NA's   :1        NA's   :1        NA's   :3
##     meal.cal        wt.loss
##  Min.   :  96.0   Min.   :-24.000
##  1st Qu.: 635.0   1st Qu.:  0.000
##  Median : 975.0   Median :  7.000
##  Mean   : 928.8   Mean   :  9.832
##  3rd Qu.:1150.0   3rd Qu.: 15.750
##  Max.   :2600.0   Max.   : 68.000
##  NA's   :47       NA's   :14
```

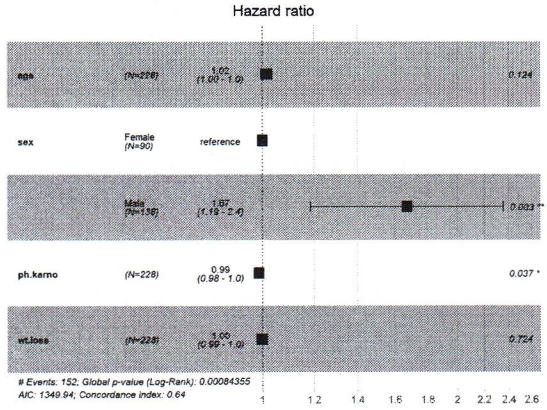
```
lung$sex = ifelse(lung$sex==1,'Male','Female')
lung$sex = as.factor(lung$sex)
summary(lung)
```

```
##       inst            time            status           age            sex
##  Min.   : 1.00   Min.   :   5.0   Min.   :1.000   Min.   :39.00   Female: 90
##  1st Qu.: 3.00   1st Qu.: 166.8   1st Qu.:1.000   1st Qu.:56.00   Male  :138
##  Median :11.00   Median : 255.5   Median :2.000   Median :63.00
##  Mean   :11.09   Mean   : 305.2   Mean   :1.724   Mean   :62.45
##  3rd Qu.:16.00   3rd Qu.: 396.5   3rd Qu.:2.000   3rd Qu.:69.00
##  Max.   :33.00   Max.   :1022.0   Max.   :2.000   Max.   :82.00
##  NA's   :1
##     ph.ecog          ph.karno        pat.karno         meal.cal
##  Min.   :0.0000   Min.   : 50.00   Min.   : 30.00   Min.   :  96.0
##  1st Qu.:0.0000   1st Qu.: 75.00   1st Qu.: 70.00   1st Qu.: 635.0
##  Median :1.0000   Median : 80.00   Median : 80.00   Median : 975.0
##  Mean   :0.9515   Mean   : 81.94   Mean   : 79.96   Mean   : 928.8
##  3rd Qu.:1.0000   3rd Qu.: 90.00   3rd Qu.: 90.00   3rd Qu.:1150.0
##  Max.   :3.0000   Max.   :100.00   Max.   :100.00   Max.   :2600.0
##  NA's   :1        NA's   :1        NA's   :3        NA's   :47
##     wt.loss
##  Min.   :-24.000
##  1st Qu.:  0.000
##  Median :  7.000
##  Mean   :  9.832
##  3rd Qu.: 15.750
##  Max.   : 68.000
##  NA's   :14
```

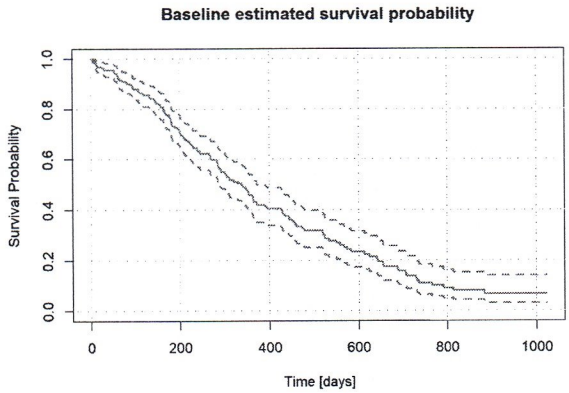
```
# Fit the Cox's regression model:
mod.cox = coxph(Surv(time, status) ~ age + sex + ph.karno + wt.loss, data =  lung)
summary(mod.cox)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ age + sex + ph.karno + wt.loss,
##     data = lung)
##
##   n= 214, number of events= 152
##    (14 observations deleted due to missingness)
##
##                coef exp(coef)  se(coef)      z Pr(>|z|)
## age       0.015140  1.015255  0.009837  1.539  0.12379
## sexMale   0.513955  1.671891  0.174410  2.947  0.00321 **
## ph.karno -0.012871  0.987211  0.006184 -2.081  0.03741 *
## wt.loss  -0.002246  0.997757  0.006357 -0.353  0.72389
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          exp(coef) exp(-coef) lower .95 upper .95
## age         1.0153     0.9850    0.9959    1.0350
## sexMale     1.6719     0.5981    1.1878    2.3532
## ph.karno    0.9872     1.0130    0.9753    0.9993
## wt.loss     0.9978     1.0022    0.9854    1.0103
##
## Concordance= 0.643  (se = 0.027 )
## Likelihood ratio test= 18.84  on 4 df,   p=8e-04
## Wald test            = 18.68  on 4 df,   p=9e-04
## Score (logrank) test = 18.99  on 4 df,   p=8e-04
```

```
## Warning: 14 observations deleted due to missingness
##          -> n= 214, number of events= 152

# The p-value for all three overall tests (likelihood, Wald, and score) are significant,
# indicating that the model is significant. These tests evaluate the omnibus null hypothesis
# that all of the betas are 0. In the above example, the test statistics are in close
# agreement, and the omnibus null hypothesis is soundly rejected.
#
# In the multivariate Cox analysis, the covariates sex and ph.karno are significant (p < 0.05).
# However, the covariates age and wt.loss fail to be significant.
#
# The HR for sex is exp(coef) = exp(0.514) = 1.67 with 95% CI = [1.19; 2.35].
# The hazard ratios of covariates are interpretable as multiplicative effects on the hazard.
# For example, holding the other covariates constant, being a male increases the hazard by a
# factor of 1.67, or 67%. We conclude that, being male is associated with bad prognostic.
#
# The HR for ph.karno is exp(coef) = exp(-0.013) = 0.987 with 95% CI = [0.975;0.999], indicating
# a strong relationship between the ph.karno value and decreased risk of death. Holding the
# other covariates constant, a higher value of ph.karno is associated with a good survival.
#
# The hazard ratio HR  of age is exp(coef) = 1.01, with a 95% CI = [0.996;1.035].
# Because the confidence interval for HR includes 1, these results indicate that age
# makes a smaller contribution to the difference in the HR after adjusting for other covariates.
#
# Similarly, the hazard ratio HR  of wt.loss is exp(coef) = 0.998, with a 95% CI = [0.985;1.010].
# Because the confidence interval for HR includes 1, these results indicate that wt.loss
# makes a smaller contribution to the difference in the HR after adjusting for other covariates.

# ------------------------------------------------------------------------------
# Visualizing Hazard ratios
# ------------------------------------------------------------------------------
# You can visualize Hr and its CIs using the ggforest() function of package survminer:
x11()
ggforest(mod.cox, data=lung)
```




```
# ------------------------------------------------------------------------------
# Adjusted Survival Curves
# ------------------------------------------------------------------------------

# Plot the baseline survival function S_0(t)
x11()
plot(survfit(mod.cox, data=lung),
     col="darkorange2", lwd=2, lty=1,
     xlab='Time [days]', ylab='Survival Probability',
     main='Baseline estimated survival probability')
grid()
```




```
# To assess the impact of the covariates on the estimated survival probability, we consider
# each covariate one at a time and we construct a new data frame with:
#   - as many rows as the categories, for categorical variables
#   - as many rows as the values of interest, for numerical variable (usually 2/3 values).
# The other covariates are fixed to their average/median values (if numerical) or to
# the most frequent class (if categorical).

# ------------------------------------------------------------------------------
# AGE
# ------------------------------------------------------------------------------
# Create the new data :
summary(lung$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   39.00   56.00   63.00   62.45   69.00   82.00
```

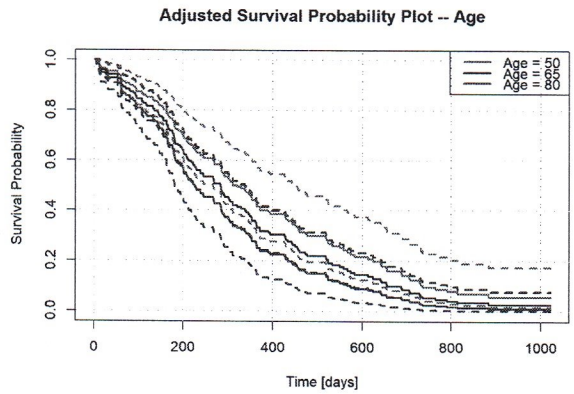
```
age_df = with(lung,
              data.frame(age = c(50,65,80),
                         sex = rep('Male',3),
                         ph.karno = rep(median(lung$pat.karno, na.rm=T),3),
                         wt.loss = rep(median(lung$wt.loss, na.rm=T),3)))
age_df
```

```
##   age  sex ph.karno wt.loss
## 1  50 Male       80       7
## 2  65 Male       80       7
## 3  80 Male       80       7
```

```
# This data frame is passed to survfit() via the newdata argument:
fit.age = survfit(mod.cox, newdata = age_df)
fit.age
```

```
## Call: survfit(formula = mod.cox, newdata = age_df)
##
##     n events median 0.95LCL 0.95UCL
## 1 214    152    320     267     450
## 2 214    152    284     226     337
## 3 214    152    226     181     337
```

```
# Estimated survival curves:
x11()
plot(fit.age, conf.int=T,
     col=c("dodgerblue2","navy","darkmagenta"), lwd=2, lty=1,
     xlab='Time [days]', ylab='Survival Probability',
     main='Adjusted Survival Probability Plot -- Age')
grid()
legend('topright', c("Age = 50", "Age = 65", "Age = 80"),
       lty=c(1,1,1), lwd=c(2,2,2), col=c("dodgerblue2","navy","darkmagenta"))
```




```
# ------------------------------------------------------------------------------
# SEX
# ------------------------------------------------------------------------------
table(lung$sex)
```

```
##
## Female   Male
##     90    138
```

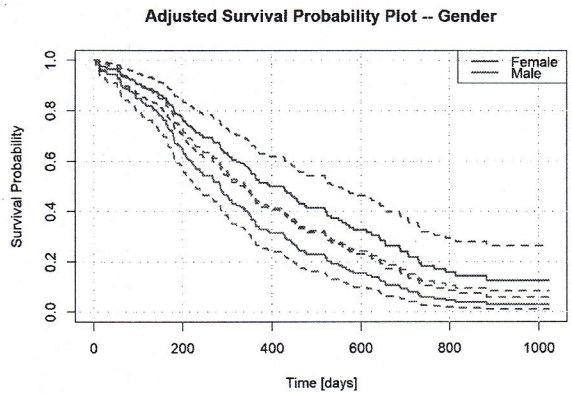
```
# Create the new data :
sex_df = with(lung,
              data.frame(age = rep(median(lung$age, na.rm=T),2),
                         sex = c('Female','Male'),
                         ph.karno = rep(median(lung$pat.karno, na.rm=T),2),
                         wt.loss = rep(median(lung$wt.loss, na.rm=T),2)))
sex_df
```

```
##   age    sex ph.karno wt.loss
## 1  63 Female       80       7
## 2  63   Male       80       7
```

```
# This data frame is passed to survfit() via the newdata argument:
fit.sex = survfit(mod.cox, newdata = sex_df)
fit.sex
```

```
## Call: survfit(formula = mod.cox, newdata = sex_df)
##
##     n events median 0.95LCL 0.95UCL
## 1 214    152    426     337     550
## 2 214    152    285     229     345
```

```
# Estimated survival curves:
x11()
plot(fit.sex, conf.int=T,
     col=c("deeppink2","dodgerblue2"), lwd=2, lty=1,
     xlab='Time [days]', ylab='Survival Probability',
     main='Adjusted Survival Probability Plot -- Gender')
grid()
legend('topright', c("Female", "Male"),
       lty=c(1,1), lwd=c(2,2), col=c("deeppink2","dodgerblue2"))
```




```
# ------------------------------------------------------------------------------
# PH.KARNO
# ------------------------------------------------------------------------------
# Create the new data :
summary(lung$ph.karno)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   50.00   75.00   80.00   81.94   90.00  100.00       1
```

```
table(lung$ph.karno)
```

```
##
##  50  60  70  80  90 100
##   6  19  32  67  74  29
```

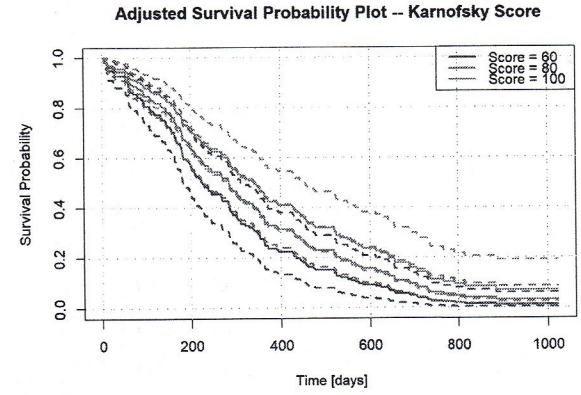
```
ph_df = with(lung,
             data.frame(age = rep(median(lung$age, na.rm=T),3),
                        sex = rep('Male',3),
                        ph.karno = c(60,80,100),
                        wt.loss = rep(median(lung$wt.loss, na.rm=T),3)))
ph_df
```

```
##   age  sex ph.karno wt.loss
## 1  63 Male       60       7
## 2  63 Male       80       7
## 3  63 Male      100       7
```

```
# This data frame is passed to survfit() via the newdata argument:
fit.ph = survfit(mod.cox, newdata = ph_df)
fit.ph
```

```
## Call: survfit(formula = mod.cox, newdata = ph_df)
##
##     n events median 0.95LCL 0.95UCL
## 1 214    152    226     182     310
## 2 214    152    285     229     345
## 3 214    152    340     284     450
```

```
# Estimated survival curves:
x11()
plot(fit.ph, conf.int=T,
     col=c("saddlebrown","orange3","salmon"), lwd=2, lty=1,
     xlab='Time [days]', ylab='Survival Probability',
     main='Adjusted Survival Probability Plot -- Karnofsky Score')
grid()
legend('topright', c("Score = 60", "Score = 80", "Score = 100"),
       lty=c(1,1,1), lwd=c(2,2,2), col=c("saddlebrown","orange3","salmon"))
```



```
# ------------------------------------------------------------------------------
# WT.LOSS
# ------------------------------------------------------------------------------
# Create the new data :
summary(lung$wt.loss)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## -24.000   0.000   7.000   9.832  15.750  68.000      14
```

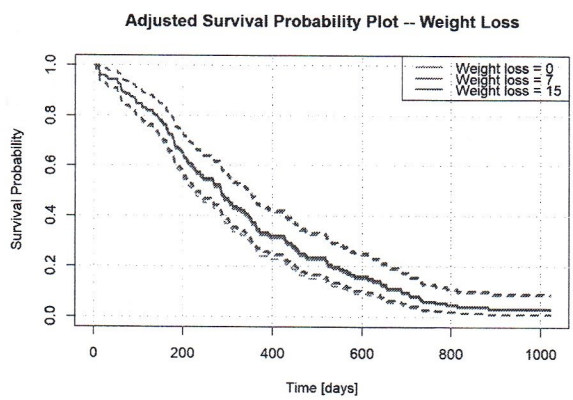
```
wt_df = with(lung,
             data.frame(age = rep(median(lung$age, na.rm=T),3),
                        sex = rep('Male',3),
                        ph.karno = rep(median(lung$ph.karno, na.rm=T),3),
                        wt.loss = c(0,7,15)))
wt_df
```

```
##   age  sex ph.karno wt.loss
## 1  63 Male       80       0
## 2  63 Male       80       7
## 3  63 Male       80      15
```

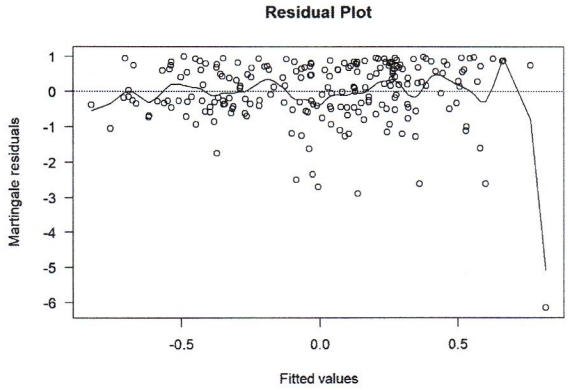
```
# This data frame is passed to survfit() via the newdata argument:
fit.wt = survfit(mod.cox, newdata = wt_df)
fit.wt
```

```
## Call: survfit(formula = mod.cox, newdata = wt_df)
##
##     n events median 0.95LCL 0.95UCL
## 1 214    152    285     223     348
## 2 214    152    285     229     345
## 3 214    152    285     230     348
```

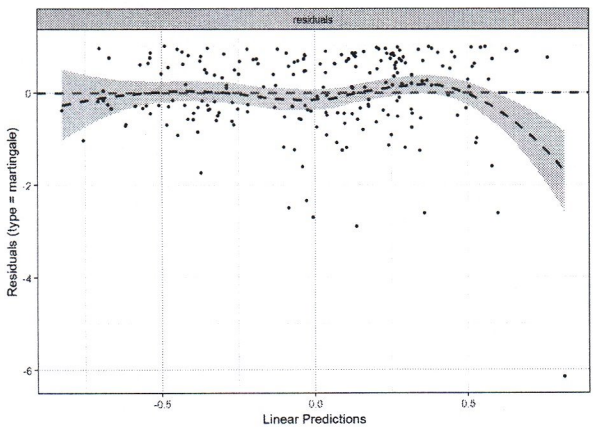
```
# Estimated survival curves:
x11()
plot(fit.wt, conf.int=T,
     col=c("springgreen","springgreen3","springgreen4"), lwd=2, lty=1,
     xlab='Time [days]', ylab='Survival Probability',
     main='Adjusted Survival Probability Plot -- Weight Loss')
grid()
legend('topright', c("Weight loss = 0", "Weight loss = 7", "Weight loss = 15"),
       lty=c(1,1,1), lwd=c(2,2,2), col=c("springgreen","springgreen3","springgreen4"))
```



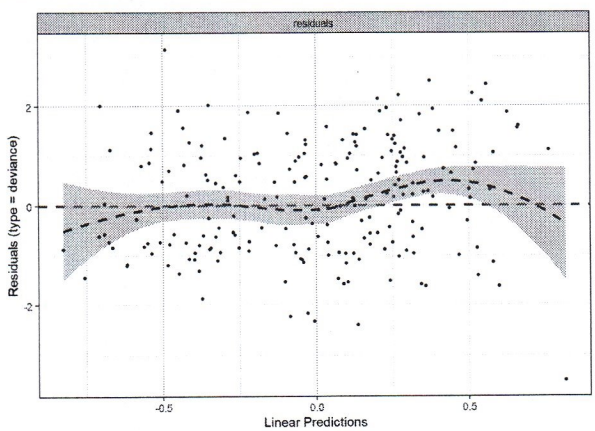

```
# ------------------------------------------------------------------------------
# Cox Model Assumptions
# Goodness of fit & PH assumption
# ------------------------------------------------------------------------------
# When used inappropriately, statistical models may give rise to misleading conclusions.
# Therefore, it's important to check that a given model is an appropriate representation
# of the data.
graphics.off()

# ------------------------------------------------------------------------------
# MARTINGALE & DEVIANCE RESIDUALS
# ------------------------------------------------------------------------------
# A first graphical option to check for goodness of fit is to check if
# the Martingale Residuals have 0 mean along time.

# Plot martingale residuals
x11()
plot(predict(mod.cox), residuals(mod.cox, type='martingale'),
     xlab='Fitted values', ylab='Martingale residuals', main='Residual Plot', las=1)
# Add a line for residual=0
abline(h=0, col='red')
# Fit a smoother for the points
lines(smooth.spline(predict(mod.cox), residuals(mod.cox, type='martingale')), col='blue')
```



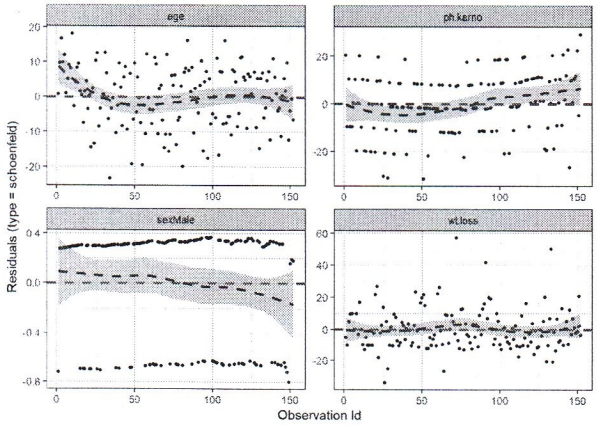

```
# Alternatively
x11()
ggcoxdiagnostics(mod.cox, type = "martingale")
```



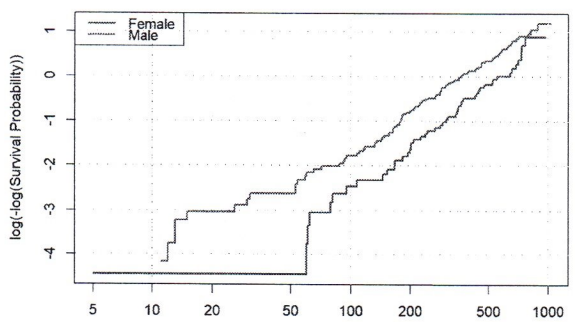

```
# Sometimes, martingale residuals are difficult to be interpreted.
# The deviance residual is a normalized transform of the martingale residual. These residuals
# should be roughly symmetrically distributed about zero with a standard deviation of 1.
#      - Positive values correspond to individuals that "died too soon" compared to expected
#        survival times.
#      - Negative values correspond to individual that "lived too long".
#      - Very large or small values are outliers, which are poorly predicted by the model.
# It's also possible to check outliers by visualizing the deviance residuals.
# Example of deviance residuals:
x11()
ggcoxdiagnostics(mod.cox, type = "deviance")
```




```
# The pattern looks fairly symmetric around 0

# ------------------------------------------------------------------------------
# SCHOENFELD RESIDUALS
# ------------------------------------------------------------------------------
# A second graphical option could be to use the Schoenfeld residuals to examine model
# fit and detect outlying covariate values. Shoenfeld residuals represent the difference
# between the observed covariate and the expected given the risk set at that time.
# They should be flat, centred about zero.
# In principle, the Schoenfeld residuals are independent of time.
# A plot showing a non-random pattern against time is evidence of violation of the PH assumption.
x11()
ggcoxdiagnostics(mod.cox, type = "schoenfeld")
```



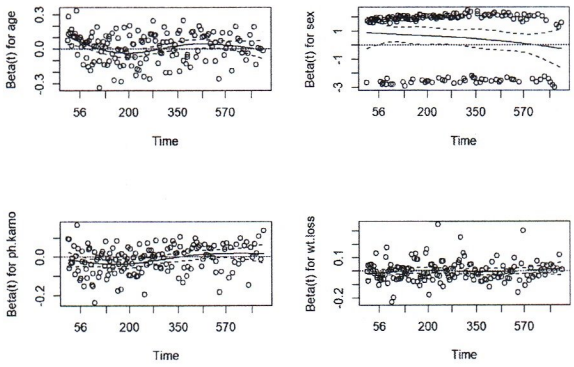

```
# ------------------------------------------------------------------------------
# LOG(-LOG(KM))
# ------------------------------------------------------------------------------
# A first graphical methods for checking proportional hazards is to plot log(-log(KM(t)))
# vs. t or log(t) and look for parallelism. This can be done only for categorical covariates.

# We consider the KM estimators for sex variable:
sex.km = survfit(Surv(time, status) ~ sex, data = lung[!is.na(lung$wt.loss) & !is.na(lung$ph.karno),])

# We plot log(-log(KM(t))) using option fun='cloglog' in plot.survfit()
help(plot.survfit)

x11()
plot(sex.km, fun='cloglog',
     col=c("deeppink2","dodgerblue2"), lwd=2, lty=1,
     ylab="log(-log(Survival Probability))")
grid()
legend('topleft', c("Female", "Male"),
       lty=c(1,1), lwd=c(2,2), col=c("deeppink2","dodgerblue2"))
```



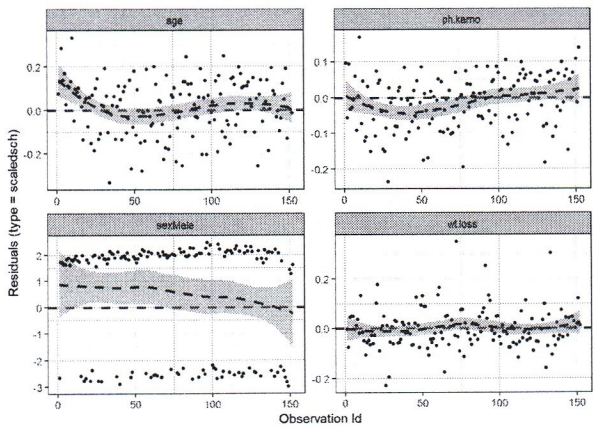

```
# Curves seem to be parallel -> PH assumption seems to be satisfied for gender.

# ------------------------------------------------------------------------------
# COX.ZPH TEST
# ------------------------------------------------------------------------------
# The function cox.zph() [in the survival package] provides a convenient solution to test
# the proportional hazards assumption for each covariate included in a Cox refression model fit.
#
# For each covariate, the function cox.zph() correlates the corresponding set of scaled
# Schoenfeld  residuals with time, to test for independence between residuals and time.
# Additionally, it performs a global test for the model as a whole.
#
# The proportional hazard assumption is supported by a non-significant relationship between
# residuals and time, and refuted by a significant relationship.
help("cox.zph")

# Test for PH using scaled Schoenfeld test for PH
# H0: Hazards are proportional
# H1: Hazards are NOT proportional
# cox.zph() return tests for each X and for the global model
test.ph = cox.zph(mod.cox)
test.ph
```

```
##             chisq df      p
## age       0.90304  1 0.3420
## sex       3.24299  1 0.0717
## ph.karno  7.98433  1 0.0047
## wt.loss   0.00382  1 0.9507
## GLOBAL   10.73439  4 0.0297
```

```
# From the output above, the global test is statistically significant.
# Therefore, we can not assume the proportional hazards.
# In particular, the test for ph.karno is statistically significant.

# Plot the scaled schoenfeld residuals:
x11()
par(mfrow=c(2,2))
for(i in 1:4){
  plot(test.ph[i])
  abline(h=0, col='red')
}
```




```
# alternatively:
x11()
ggcoxdiagnostics(mod.cox, type = "scaledsch")
```




```
graphics.off()
```