# Medical Statistics Using R: Part 2

Short version. Draft updated August 13, 2018. Not for sale :-)

*Wan Nor Arifin and Kamarul Imran Musa*

# Contents

**1 Survival analysis: Kaplan-Meier Survival Curve 5**

1.1 Introduction . . . 5

1.2 Kaplan-Meier survival curve in a group . . . 5

1.3 Comparing Kaplan-Meier curves between groups . . . 8

1.4 Exercises . . . 12

**2 Survival analysis: Cox Proportional Hazards Model 13**

2.1 Introduction . . . 13

2.2 Analysis . . . 14

2.3 Exercises . . . 34

**3 References 35**

# Chapter 1

# Survival analysis: Kaplan-Meier Survival Curve

## 1.1 Introduction

1. A statistical method to analyze:
   - outcome: time to event (e.g. death, recurrence etc).
   - (comparison) predictors/independent variables: categorical variables.
2. It is concerned with survival probability at specific time points over a time interval (follow-up period), e.g. five year survival etc.
3. Basically, the *interval survival* at time $t$ is as follows,

   $$Interval\ survival, p_t = \frac{Survivors,\ n_t - Deaths,\ e_t}{Survivor,\ n_t}$$

   The survival is usually represented by *cumulative survival* until time $t$,

   $$Cumulative\ survival, s_t = p_0 p_1 p_2 ... p_{t-1}$$

4. In follow-up study over a period of time, not all subjects will experience event (e.g. not everyone die in 5 years). The subjects are called *censored* observations. In survival analysis, this censored observations are taken into account.

## 1.2 Kaplan-Meier survival curve in a group

Table 1.1: Acute myeloid leukemia data (Miller, 1997).

| time | status | x |
|---:|---:|:---|
| 9 | 1 | Maintained |
| 13 | 1 | Maintained |
| 13 | 0 | Maintained |
| 18 | 1 | Maintained |
| 23 | 1 | Maintained |
| 28 | 0 | Maintained |

| time | status | x |
|---:|---:|:---|
| 31 | 1 | Maintained |
| 34 | 1 | Maintained |
| 45 | 0 | Maintained |
| 48 | 1 | Maintained |
| 161 | 0 | Maintained |
| 5 | 1 | Nonmaintained |
| 5 | 1 | Nonmaintained |
| 8 | 1 | Nonmaintained |
| 8 | 1 | Nonmaintained |
| 12 | 1 | Nonmaintained |
| 16 | 0 | Nonmaintained |
| 23 | 1 | Nonmaintained |
| 27 | 1 | Nonmaintained |
| 30 | 1 | Nonmaintained |
| 33 | 1 | Nonmaintained |
| 43 | 1 | Nonmaintained |
| 45 | 1 | Nonmaintained |

Load the required packages. Make sure you have all these in your computer.

```
# library
library(survival)
library(epiDisplay)
library(coin)
```

We are going to use a built-in dataset in `survival` package, namely `aml`. We assign it to `acute` data object to avoid confusion with the built-in dataset name.

```
# data
?aml  # about the dataset
acute = aml
acute
```

```
##    time status             x
## 1     9      1    Maintained
## 2    13      1    Maintained
## 3    13      0    Maintained
## 4    18      1    Maintained
## 5    23      1    Maintained
## 6    28      0    Maintained
## 7    31      1    Maintained
## 8    34      1    Maintained
## 9    45      0    Maintained
## 10   48      1    Maintained
## 11  161      0    Maintained
## 12    5      1 Nonmaintained
## 13    5      1 Nonmaintained
## 14    8      1 Nonmaintained
## 15    8      1 Nonmaintained
## 16   12      1 Nonmaintained
## 17   16      0 Nonmaintained
## 18   23      1 Nonmaintained
## 19   27      1 Nonmaintained
## 20   30      1 Nonmaintained
## 21   33      1 Nonmaintained
## 22   43      1 Nonmaintained
## 23   45      1 Nonmaintained
```

Explore the data,

```
codebook(acute)
```

```
##
##
##
## time      :
##  obs. mean   median  s.d.   min.   max.
##  23   29.478 23      31.72  5      161
##
##  ==================
## status    :
##  obs. mean   median  s.d.   min.   max.
##  23   0.783  1       0.422  0      1
##
##  ==================
## x     :
##              Frequency Percent
## Maintained           11    47.8
## Nonmaintained        12    52.2
##
##  ==================
```

```
table(acute$status)  # number of events
```

```
##
##  0  1
##  5 18
```

Generate survival curve data for plotting by `survfit()`.

```
sur_aml = survfit(Surv(time, status) ~ 1, data = acute)
```

View median survival time and its 95% confidence interval from `sur_aml`,

```
sur_aml  # median survival time = 27 (95%CI: 18, 45)
```

```
## Call: survfit(formula = Surv(time, status) ~ 1, data = acute)
##
##       n  events  median 0.95LCL 0.95UCL
##      23       18      27      18      45
```

and the details of the survival curve data,

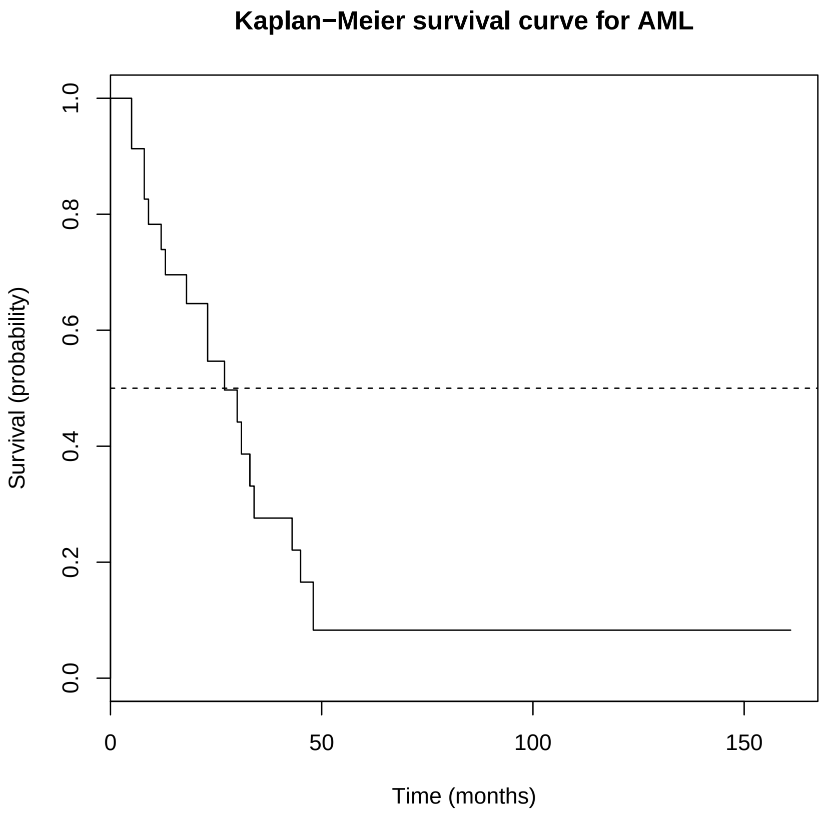
```
summary(sur_aml)
```

```
## Call: survfit(formula = Surv(time, status) ~ 1, data = acute)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     5     23       2   0.9130  0.0588       0.8049        1.000
##     8     21       2   0.8261  0.0790       0.6848        0.996
##     9     19       1   0.7826  0.0860       0.6310        0.971
##    12     18       1   0.7391  0.0916       0.5798        0.942
##    13     17       1   0.6957  0.0959       0.5309       0.912
##    18     14       1   0.6460  0.1011       0.4753       0.878
##    23     13       2   0.5466  0.1073       0.3721       0.803
##    27     11       1   0.4969  0.1084       0.3240       0.762
##    30      9       1   0.4417  0.1095       0.2717       0.718
##    31      8       1   0.3865  0.1089       0.2225       0.671
##    33      7       1   0.3313  0.1064       0.1765       0.622
##    34      6       1   0.2761  0.1020       0.1338       0.569
##    43      5       1   0.2208  0.0954       0.0947       0.515
##    45      4       1   0.1656  0.0860       0.0598       0.458
##    48      2       1   0.0828  0.0727       0.0148       0.462
```

Note the `survival` column for $s_t$.

Now plot Kaplan-Meier survival curve,

```
plot(sur_aml, conf.int = F,
     main = "Kaplan-Meier survival curve for AML",
     xlab = "Time (months)", ylab = "Survival (probability)")
# `conf.int = F` to supress 95% CI line, can remove the argument
abline(0.5, 0, lty = 2)
```



## 1.3 Comparing Kaplan-Meier curves between groups

Table 1.2: Glioma data for histology grade 4 patients (Grana et al., 2002).

| | time | event | group |
|---|---|---|---|
| 12 | 43 | FALSE | RIT |
| 13 | 20 | TRUE | RIT |
| 14 | 14 | TRUE | RIT |
| 15 | 36 | FALSE | RIT |
| 16 | 59 | FALSE | RIT |
| 17 | 31 | TRUE | RIT |
| 18 | 14 | TRUE | RIT |
| 19 | 36 | TRUE | RIT |
| 26 | 8 | TRUE | Control |
| 27 | 8 | TRUE | Control |
| 28 | 11 | TRUE | Control |
| 29 | 12 | TRUE | Control |
| 30 | 15 | TRUE | Control |
| 31 | 5 | TRUE | Control |
| 32 | 8 | TRUE | Control |
| 33 | 8 | TRUE | Control |
| 34 | 6 | TRUE | Control |
| 35 | 14 | TRUE | Control |
| 36 | 13 | TRUE | Control |
| 37 | 25 | TRUE | Control |

### 1.3.1 KM for two groups

Now we use the built-in data in `coin` package, namely `glioma`. We assign it to `gli` data object to avoid confusion with the built-in dataset name.

```
# data
?glioma  # about the dataset
gli = glioma
str(gli)
```

```
## 'data.frame':    37 obs. of  7 variables:
##  $ no.      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ age      : int  41 45 48 54 40 31 53 49 36 52 ...
##  $ sex      : Factor w/ 2 levels "Female","Male": 1 1 2 2 1 2 2 2 2 2 ...
##  $ histology: Factor w/ 2 levels "GBM","Grade3": 2 2 2 2 2 2 2 2 2 2 ...
##  $ group    : Factor w/ 2 levels "Control","RIT": 2 2 2 2 2 2 2 2 2 2 ...
##  $ event    : logi  TRUE FALSE FALSE FALSE FALSE TRUE ...
##  $ time     : int  53 28 69 58 54 25 51 61 57 57 ...
```

For this analysis, we only need the data for `GBM` subgroup under `histology`,

```
gli4 = subset(gli, histology == "GBM")  # grade 4 glioma
str(gli4)
```

```
## 'data.frame':    20 obs. of  7 variables:
##  $ no.      : int  12 13 14 15 16 17 18 19 7 8 ...
##  $ age      : int  55 70 39 40 47 58 40 36 32 70 ...
##  $ sex      : Factor w/ 2 levels "Female","Male": 1 2 1 1 1 2 1 2 1 2 ...
##  $ histology: Factor w/ 2 levels "GBM","Grade3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ group    : Factor w/ 2 levels "Control","RIT": 2 2 2 2 2 2 2 2 1 1 ...
##  $ event    : logi  FALSE TRUE TRUE FALSE FALSE TRUE ...
##  $ time     : int  43 20 14 36 59 31 14 36 8 8 ...
```

Expore the data,

```
codebook(gli4)
```

```
##
##
##
## no.   :
##  obs. mean   median  s.d.   min.   max.
##  20   13.7   14      3.466  7      19
##
##  ==================
## age   :
##  obs. mean   median  s.d.   min.   max.
##  20   53.9   52.5    14.436 32     83
##
##  ==================
## sex   :
##        Frequency Percent
## Female        10      50
## Male          10      50
##
##  ==================
## histology     :
##        Frequency Percent
## GBM           20     100
## Grade3         0       0
##
##  ==================
## group     :
##         Frequency Percent
## Control        12      60
## RIT             8      40
##
##  ==================
## event     :
##       Frequency Percent
## FALSE         3      15
## TRUE         17      85
##
##  ==================
## time      :
##  obs. mean   median  s.d.   min.   max.
##  20   19.3   14      14.55  5      59
##
##  ==================
```

Now, we generate survival curve data,

```
sur_gli4 = survfit(Surv(time, event) ~ group, data = gli4)
```

then view the median (95% CI),

```
sur_gli4  # median survival times/group, 0.95UCL cannot be estimated
```

```
## Call: survfit(formula = Surv(time, event) ~ group, data = gli4)
##
##                n events median 0.95LCL 0.95UCL
## group=Control 12     12    9.5       8      NA
## group=RIT      8      5   33.5      20      NA
```

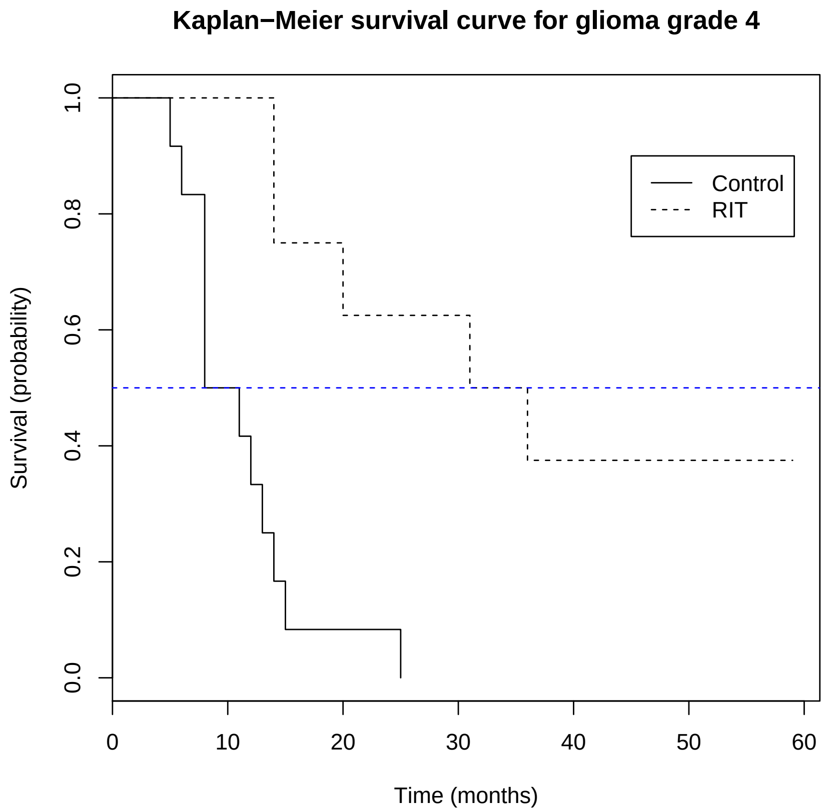
and the details of the survival curve data,

```
summary(sur_gli4)
```

```
## Call: survfit(formula = Surv(time, event) ~ group, data = gli4)
##
##                 group=Control
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     5     12       1   0.9167  0.0798       0.7729        1.000
##     6     11       1   0.8333  0.1076       0.6470        1.000
##     8     10       4   0.5000  0.1443       0.2840        0.880
##    11      6       1   0.4167  0.1423       0.2133        0.814
##    12      5       1   0.3333  0.1361       0.1498        0.742
##    13      4       1   0.2500  0.1250       0.0938        0.666
##    14      3       1   0.1667  0.1076       0.0470        0.591
##    15      2       1   0.0833  0.0798       0.0128        0.544
##    25      1       1   0.0000     NaN           NA           NA
##
##                 group=RIT
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    14      8       2    0.750   0.153        0.503        1.000
##    20      6       1    0.625   0.171        0.365        1.000
##    31      5       1    0.500   0.177        0.250        1.000
##    36      4       1    0.375   0.171        0.153        0.917
```

Now we plot the `sur_gli4` data,

```
plot(sur_gli4, main = "Kaplan-Meier survival curve for glioma grade 4",
     xlab = "Time (months)", ylab = "Survival (probability)", lty = c(1, 2))
abline(0.5, 0, lty = 2, col = "blue")
legend(45, 0.9, c("Control", "RIT"), lty = c(1, 2))
```



### 1.3.2 Log-rank test comparing two groups

Now, we compare statistically the survival curves of the groups by log-rank test,

```
survdiff(Surv(time, event) ~ group, data = gli4)
```

```
## Call:
## survdiff(formula = Surv(time, event) ~ group, data = gli4)
##
##                N Observed Expected (O-E)^2/E (O-E)^2/V
## group=Control 12       12     5.93      6.23      12.6
## group=RIT      8        5    11.07      3.33      12.6
##
##  Chisq= 12.6  on 1 degrees of freedom, p= 4e-04
```

## 1.4 Exercises

1. Analyze gli data for histology == “Grade3”.

```
gli3 = subset(gli, histology == "Grade3")
```

2. Again, analyze using `aml` data from `survival` package. This time compare the groups (`x` variable in the dataset).

# Chapter 2

# Survival analysis: Cox Proportional Hazards Model

## 2.1 Introduction

1. A statistical method to model:
   - outcome: time to event (e.g. death, recurrence etc).
   - predictors/independent variables: numerical, categorical variables.
2. In contrast to KM approach, it is concerned with hazard of event.
3. Basically, the (interval) *hazard* at time $t$ is as follows,

$$Hazard,\ h_t = \frac{Deaths,\ e_t}{Survivors,\ n_t \times Interval,\ u_t}$$

   where interval $u_t$ is the time interval from present time $t$ until the next event time $t+1$.

Nelson-Aalen's *cumulative hazard* estimate is given sum of $e_i/n_1$ until time $t$,

$$Nelson - Aalen\ cumulative\ hazard,\ NA = \sum_{i \leq t} \frac{e_i}{n_i}$$

In R, *cumulative hazard function*, $H(t)$ is calculated from the *estimated cumulative survival function*, $S(t)$ as follows,

$$H(t) = -log_e S(t)$$

4. The formula for Cox proportional hazards (PH) model,

$$\begin{aligned} log_e\left(\frac{hazard\ at\ time,\ t}{baseline\ hazard\ at\ time,\ t}\right) &= \\ log_e(hazard\ ratio,\ HR) &= coefficients \times numerical\ predictors \\ &\quad + coefficients \times categorical\ predictors \end{aligned}$$

   or in notational form,

$$\begin{aligned} log_e\left(\frac{h(t)}{h_0(t)}\right) &= \\ log_e HR &= \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k \end{aligned}$$

   where we have $k$ predictors. Notice there is something missing in the equations above, which is the intercept ($\beta_0$). It is because the intercept = 0 at time = 0, i.e. nobody experiences the event at the start of the followup period, everyone is still alive!

Whenever the predictor is a categorical variable with more than two levels, remember to consider dummy (binary) variable(s).

5. Hazard ratio (HR) is the ratio of hazards of two levels. HR for a predictor is easily calculated from a Cox PH model,

$$HR_i = e^{\beta_i}$$

## 2.2 Analysis

Load the required packages. Make sure you have all these in your computer.

```
# library
library(survival)
library(epiDisplay)
library(coin)
library(TH.data)
library(car)
```

### 2.2.1 A detour: obtaining the cumulative survival and hazard function

Before we start with Cox PH model, we want to obtain the cumulative survival and hazard until time $t$ using `coxph()` function on one group. Following our previous one-group survival analysis `sur_aml` on `aml` dataset,

```
acute = aml
cox_aml = coxph(Surv(time, status) ~ 1, data = acute)
sur_cox_aml = survfit(cox_aml)
summary(sur_cox_aml)
```

```
## Call: survfit(formula = cox_aml)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     5     23       2    0.915  0.0575       0.8088        1.000
##     8     21       2    0.830  0.0775       0.6910        0.997
##     9     19       1    0.787  0.0844       0.6381        0.971
##    12     18       1    0.745  0.0899       0.5878        0.944
##    13     17       1    0.702  0.0943       0.5397        0.914
##    18     14       1    0.654  0.0995       0.4852        0.881
##    23     13       2    0.557  0.1057       0.3840        0.808
##    27     11       1    0.509  0.1070       0.3367        0.768
##    30      9       1    0.455  0.1083       0.2855        0.725
##    31      8       1    0.402  0.1079       0.2372        0.680
##    33      7       1    0.348  0.1060       0.1917        0.632
##    34      6       1    0.295  0.1023       0.1493        0.582
##    43      5       1    0.241  0.0966       0.1101        0.529
##    45      4       1    0.188  0.0887       0.0745        0.474
##    48      2       1    0.114  0.0784       0.0296        0.439
```

```
basehaz(cox_aml)  # also try `-log(sur_cox_aml$surv)` to obtain H_t
```

```
##        hazard time
## 1  0.08893281    5
## 2  0.18655185    8
## 3  0.23918343    9
## 4  0.29473899   12
## 5  0.35356252   13
## 6  0.35356252   16
## 7  0.42499109   18
## 8  0.58524750   23
## 9  0.67615659   27
## 10 0.67615659   28
## 11 0.78726770   30
## 12 0.91226770   31
## 13 1.05512484   33
## 14 1.22179151   34
## 15 1.42179151   43
## 16 1.67179151   45
## 17 2.17179151   48
## 18 2.17179151  161
```

### 2.2.2 Data

Now, back to our main business, we are going to use a built-in dataset in `survival` package, namely `lung`. We assign it to `lca` data object.

```
# data
?lung  # about the dataset
lca = na.omit(lung)  # omit subjects with missing data
str(lca)
```

```
## 'data.frame':    167 obs. of  10 variables:
##  $ inst     : num  3 5 12 7 11 1 7 6 12 22 ...
##  $ time     : num  455 210 1022 310 361 ...
##  $ status   : num  2 2 1 2 2 2 2 2 2 2 ...
##  $ age      : num  68 57 74 68 71 53 61 57 57 70 ...
##  $ sex      : num  1 1 1 2 2 1 1 1 1 1 ...
##  $ ph.ecog  : num  0 1 1 2 2 1 2 1 1 1 ...
##  $ ph.karno : num  90 90 50 70 60 70 70 80 80 90 ...
##  $ pat.karno: num  90 60 80 60 80 80 70 80 70 100 ...
##  $ meal.cal : num  1225 1150 513 384 538 ...
##  $ wt.loss  : num  15 11 0 10 1 16 34 27 60 -5 ...
##  - attr(*, "na.action")= 'omit' Named int  1 3 5 12 13 14 16 20 23 25 ...
##   ..- attr(*, "names")= chr  "1" "3" "5" "12" ...
```

The dataset needs some modifications and preparations for the purpose of our analysis, specifically `status`, `sex` and `ph.ecog`. Again, use `?lung` to look at the description of the dataset.

Give proper coding for `status` variable as `0/1` for `censored/dead`

```
table(lca$status)  # status: 1=censored, 2=dead
```

```
## 
##   1   2 
##  47 120
```

```
lca$status = lca$status - 1  # turn to our usual 0=censored, 1=dead
```

Factor `sex` variable properly as male/female,

```
lca$sex = factor(lca$sex, levels = 1:2, labels = c("male", "female"))
str(lca$sex)
```

```
##  Factor w/ 2 levels "male","female": 1 1 1 2 2 1 1 1 1 1 ...
```

Although `ph.ecog` variable is supposed to be a numerical variable, it has narrow scale (only 0 to 3 in our data). Thus we turn it into a categorical variable,

```
table(lca$ph.ecog)  # only one obs. = 3, set to 2
```

```
##
##  0  1  2  3
## 47 81 38  1
```

```
lca[lca$ph.ecog == 3, ]$ph.ecog = 2
lca$ph.ecog = factor(lca$ph.ecog)
str(lca$ph.ecog)
```

```
##  Factor w/ 3 levels "0","1","2": 1 2 2 3 3 2 3 2 2 2 ...
```

### 2.2.3 Data exploration

```
# explore
codebook(lca)
```

```
##
##
##
## inst      :
##  obs. mean   median  s.d.   min.   max.
##  167  10.707 11      8.168  1      32
##
##  ==================
## time      :
##  obs. mean    median  s.d.    min.   max.
##  167  309.934 268     209.436 5      1022
##
##  ==================
## status    :
##  obs. mean   median  s.d.   min.   max.
##  167  0.719  1       0.451  0      1
##
##  ==================
## age   :
##  obs. mean   median  s.d.   min.   max.
##  167  62.569 64      9.211  39     82
##
##  ==================
## sex   :
##        Frequency Percent
## male         103    61.7
## female        64    38.3
##
##  ==================
## ph.ecog   :
```

```
##   Frequency Percent
## 0        47    28.1
## 1        81    48.5
## 2        39    23.4
##
##  ==================
## ph.karno      :
##  obs. mean   median  s.d.   min.   max.
##  167  82.036 80      12.779 50     100
##
##  ==================
## pat.karno     :
##  obs. mean   median  s.d.   min.   max.
##  167  79.581 80      15.104 30     100
##
##  ==================
## meal.cal      :
##  obs. mean    median  s.d.   min.   max.
##  167  929.126 975     413.49 96     2600
##
##  ==================
## wt.loss   :
##  obs. mean   median  s.d.   min.   max.
##  167  9.719  7       13.379 -24    68
##
##  ==================
```

```
table(lca$status)  # number of events
```

```
##
##   0   1
##  47 120
```

### 2.2.4 Univariable

Out task now is to decide on the predictors to include in the multivariable model. As usual, to use `add1()` function, we start with an empty model,

```
cox_lca0 = coxph(Surv(time, status) ~ 1, data = lca)  # empty model
summary(cox_lca0)  # remember, no intercept in Cox PH, so there's nothing here
```

```
## Call:  coxph(formula = Surv(time, status) ~ 1, data = lca)
##
## Null model
##   log likelihood= -508.1168
##   n= 167
```

We list the variable names in the dataset, and apply some R trick to include " + " in between the variable names,

```
names(lca)
```

```
##  [1] "inst"      "time"      "status"    "age"       "sex"
##  [6] "ph.ecog"   "ph.karno"  "pat.karno" "meal.cal"  "wt.loss"
```

```r
cat(names(lca), sep = " + ")
```

```
## inst + time + status + age + sex + ph.ecog + ph.karno + pat.karno + meal.cal + wt.loss
```

This makes our life easier to copy all the variable names for regression analysis.

For the current analysis, we skip `ph.karno` and `pat.carno`, we only include `age + sex + ph.ecog + meal.cal + wt.loss` in the predictor list,

```r
add1(cox_lca0, scope = ~ age + sex + ph.ecog + meal.cal + wt.loss,
     test = "Chisq")  # LR test, note the argument's value is different from glm
```

```
## Single term additions
##
## Model:
## Surv(time, status) ~ 1
##          Df    AIC     LRT Pr(>Chi)
## <none>      1016.2
## age       1 1014.7  3.5236 0.060503 .
## sex       1 1012.0  6.2468 0.012442 *
## ph.ecog   2 1007.6 12.5861 0.001849 **
## meal.cal  1 1018.0  0.2486 0.618081
## wt.loss   1 1018.2  0.0005 0.981746
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
                      # which uses test = "LRT"
```

Since `meal.cal` and `wt.loss`'s P-values $< 0.25$, we exclude these variables from multivariable model. We proceed with three variables, `age, sex` and `ph.ecog`.

### 2.2.5 Multivariable model

We include the selected variables into our multivariable model

```r
cox_lca = coxph(Surv(time, status) ~ age + sex + ph.ecog, data = lca)
cox_lca  # basic results
```

```
## Call:
## coxph(formula = Surv(time, status) ~ age + sex + ph.ecog, data = lca)
##
##               coef exp(coef) se(coef)     z       p
## age        0.00722   1.00725  0.01123  0.64 0.52006
## sexfemale -0.50167   0.60552  0.19737 -2.54 0.01103
## ph.ecog1   0.31399   1.36888  0.23333  1.35 0.17839
## ph.ecog2   0.88975   2.43453  0.26921  3.31 0.00095
##
## Likelihood ratio test=20  on 4 df, p=5e-04
## n= 167, number of events= 120
```

Focus on:

- Coefficients, $\beta$s.
- HRs, `exp(coef)`, which are given here.
- $P$-values.

It also gives the `likelihood ratio test`, i.e. LR test of the present model vs empty model.

Using `summary()` gives more details,

```
summary(cox_lca)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ age + sex + ph.ecog, data = lca)
##
##   n= 167, number of events= 120
##
##                coef exp(coef)  se(coef)      z Pr(>|z|)
## age        0.007223  1.007249  0.011229  0.643  0.52006
## sexfemale -0.501674  0.605516  0.197374 -2.542  0.01103 *
## ph.ecog1   0.313994  1.368882  0.233325  1.346  0.17839
## ph.ecog2   0.889753  2.434527  0.269210  3.305  0.00095 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           exp(coef) exp(-coef) lower .95 upper .95
## age          1.0072     0.9928    0.9853    1.0297
## sexfemale    0.6055     1.6515    0.4113    0.8915
## ph.ecog1     1.3689     0.7305    0.8665    2.1626
## ph.ecog2     2.4345     0.4108    1.4364    4.1264
##
## Concordance= 0.642  (se = 0.031 )
## Rsquare= 0.113   (max possible= 0.998 )
## Likelihood ratio test= 20  on 4 df,   p=5e-04
## Wald test            = 20.32  on 4 df,   p=4e-04
## Score (logrank) test = 21.16  on 4 df,   p=3e-04
```

which now includes 95% CI of the HRs.

### 2.2.6 Stepwise

```
# both
cox_lca_stepboth = step(cox_lca, direction = "both")
```

```
## Start:  AIC=1004.24
## Surv(time, status) ~ age + sex + ph.ecog
##
##           Df    AIC
## - age      1 1002.6
## <none>       1004.2
## - sex      1 1009.0
## - ph.ecog  2 1011.4
##
## Step:  AIC=1002.65
## Surv(time, status) ~ sex + ph.ecog
##
##           Df    AIC
## <none>       1002.6
## + age      1 1004.2
## - sex      1 1007.6
## - ph.ecog  2 1012.0
```

```
cox_lca_stepboth
```

```
## Call:
## coxph(formula = Surv(time, status) ~ sex + ph.ecog, data = lca)
##
##              coef exp(coef) se(coef)     z      p
## sexfemale -0.508     0.602    0.197 -2.58 0.0100
## ph.ecog1   0.320     1.378    0.233  1.37 0.1693
## ph.ecog2   0.937     2.552    0.259  3.61 0.0003
##
## Likelihood ratio test=19.58  on 3 df, p=2e-04
## n= 167, number of events= 120
```

```
# forward
cox_lca_stepforward = step(cox_lca0, scope = ~ age + sex + ph.ecog, direction = "forward")
```

```
## Start:  AIC=1016.23
## Surv(time, status) ~ 1
##
##             Df    AIC
## + ph.ecog  2 1007.6
## + sex      1 1012.0
## + age      1 1014.7
## <none>       1016.2
##
## Step:  AIC=1007.65
## Surv(time, status) ~ ph.ecog
##
##         Df    AIC
## + sex    1 1002.6
## <none>     1007.6
## + age    1 1009.0
##
## Step:  AIC=1002.65
## Surv(time, status) ~ ph.ecog + sex
##
##         Df    AIC
## <none>     1002.6
## + age    1 1004.2
```

```
cox_lca_stepforward
```

```
## Call:
## coxph(formula = Surv(time, status) ~ ph.ecog + sex, data = lca)
##
##              coef exp(coef) se(coef)     z      p
## ph.ecog1   0.320     1.378    0.233  1.37 0.1693
## ph.ecog2   0.937     2.552    0.259  3.61 0.0003
## sexfemale -0.508     0.602    0.197 -2.58 0.0100
##
## Likelihood ratio test=19.58  on 3 df, p=2e-04
## n= 167, number of events= 120
```

```
# backward
cox_lca_stepback = step(cox_lca, direction = "backward")
```

```
## Start:  AIC=1004.24
## Surv(time, status) ~ age + sex + ph.ecog
## 
##           Df    AIC
## - age      1 1002.6
## <none>       1004.2
## - sex      1 1009.0
## - ph.ecog  2 1011.4
## 
## Step:  AIC=1002.65
## Surv(time, status) ~ sex + ph.ecog
## 
##           Df    AIC
## <none>       1002.6
## - sex      1 1007.6
## - ph.ecog  2 1012.0
```

```
cox_lca_stepback
```

```
## Call:
## coxph(formula = Surv(time, status) ~ sex + ph.ecog, data = lca)
## 
##             coef exp(coef) se(coef)     z      p
## sexfemale -0.508     0.602    0.197 -2.58 0.0100
## ph.ecog1   0.320     1.378    0.233  1.37 0.1693
## ph.ecog2   0.937     2.552    0.259  3.61 0.0003
## 
## Likelihood ratio test=19.58  on 3 df, p=2e-04
## n= 167, number of events= 120
```

All stepwise methods give the same set of variables, which is `sex + ph.ecog`. We name it as `cox_lca1`,

```
cox_lca1 = cox_lca_stepboth
summary(cox_lca1)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ sex + ph.ecog, data = lca)
## 
##   n= 167, number of events= 120 
## 
##              coef exp(coef) se(coef)      z Pr(>|z|)    
## sexfemale -0.5076    0.6019   0.1970 -2.576 0.009983 ** 
## ph.ecog1   0.3204    1.3776   0.2331  1.374 0.169289    
## ph.ecog2   0.9368    2.5518   0.2592  3.614 0.000301 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
##           exp(coef) exp(-coef) lower .95 upper .95
## sexfemale    0.6019     1.6614    0.4091    0.8856
## ph.ecog1     1.3776     0.7259    0.8724    2.1753
## ph.ecog2     2.5518     0.3919    1.5354    4.2410
## 
## Concordance= 0.646  (se = 0.03 )
## Rsquare= 0.111   (max possible= 0.998 )
## Likelihood ratio test= 19.58  on 3 df,   p=2e-04
## Wald test            = 20.21  on 3 df,   p=2e-04
## Score (logrank) test = 20.97  on 3 df,   p=1e-04
```

### 2.2.7 Confounder

We skip confounder checking this time, you may do this step as an exercise.

### 2.2.8 Model comparison

Compare `cox_lca1` model with the no-variable model and all-variable model by LR test and AIC comparison.

```
# LR test
anova(cox_lca0, cox_lca1, cox_lca, test = "Chisq")
```

```
## Analysis of Deviance Table
##  Cox model: response is  Surv(time, status)
##  Model 1: ~ 1
##  Model 2: ~ sex + ph.ecog
##  Model 3: ~ age + sex + ph.ecog
##    loglik   Chisq Df P(>|Chi|)
## 1 -508.12
## 2 -498.33 19.5796  3 0.0002074 ***
## 3 -498.12  0.4173  1 0.5183043
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There is no difference of 2-variable model (`cox_lca1`) to 3-variable full model.

```
# AIC
AIC(cox_lca0, cox_lca1, cox_lca)
```

```
##          df      AIC
## cox_lca0  0       NA
## cox_lca1  3 1002.654
## cox_lca   4 1004.237
```

`cox_lca1` has lower AIC than 3-variable full model. Note that there is no AIC for empty model in Cox PH because there's no intercept.

### 2.2.9 Multicollinearity, MC

We check for the variables are redundant (multicollinear) by looking at the magnitude of SE to its coefficient (same approach to that of logistic regression),

```
cox_lca1  # small SEs < coefficients
```

```
## Call:
## coxph(formula = Surv(time, status) ~ sex + ph.ecog, data = lca)
##
##             coef exp(coef) se(coef)     z      p
## sexfemale -0.508     0.602    0.197 -2.58 0.0100
## ph.ecog1   0.320     1.378    0.233  1.37 0.1693
## ph.ecog2   0.937     2.552    0.259  3.61 0.0003
##
## Likelihood ratio test=19.58  on 3 df, p=2e-04
## n= 167, number of events= 120
```

There are no large SEs for these two variables.

### 2.2.10 Interaction, *

Now we add *sex* × *ph.ecog* interaction term to the model,

```
add1(cox_lca1, scope = ~ . + sex * ph.ecog, test = "Chisq")  # insig. *
```

```
## Single term additions
##
## Model:
## Surv(time, status) ~ sex + ph.ecog
##              Df    AIC    LRT Pr(>Chi)
## <none>          1002.6
## sex:ph.ecog  2 1005.2 1.4089   0.4944
```

There was no significant interaction to be included in out model.

### 2.2.11 Proportional hazards assumption

This is a very important assumption in Cox PH model. As the name of the model itself suggests, *proportional hazards* asssumption is central to the model.

#### 2.2.11.1 Statistical test

We start with a formal statistical test of the PH assumption by predictors and overall (global test) using `cox.zph()` function. It tests for constant coefficients over the time.

```
cox.zph(cox_lca1)
```

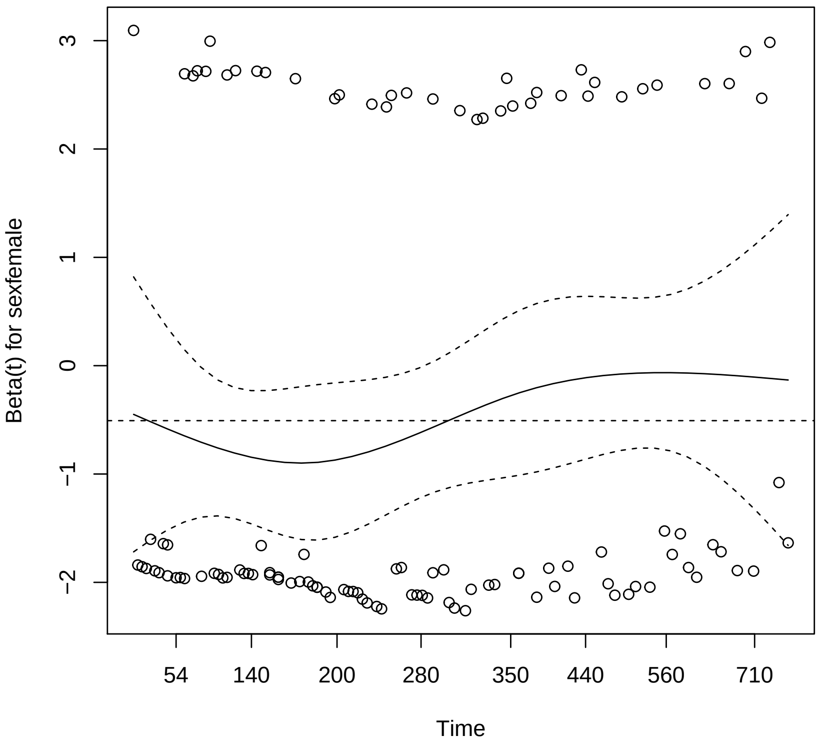
```
##                rho chisq      p
## sexfemale  0.1192 1.639 0.2005
## ph.ecog1  -0.0558 0.367 0.5444
## ph.ecog2  -0.2055 4.859 0.0275
## GLOBAL          NA 7.237 0.0647
```

We notice that the $P$-value $< 0.05$ for `pg.ecog2` hazards i.e. for pg.ecog = 2 vs pg.ecog = 0 levels, which indicates the the hazards are not proportionate. But as we turn to the global test, the $P$-value is $> 0.05$. We may conclude it is proportionate if we consider the model as a whole.

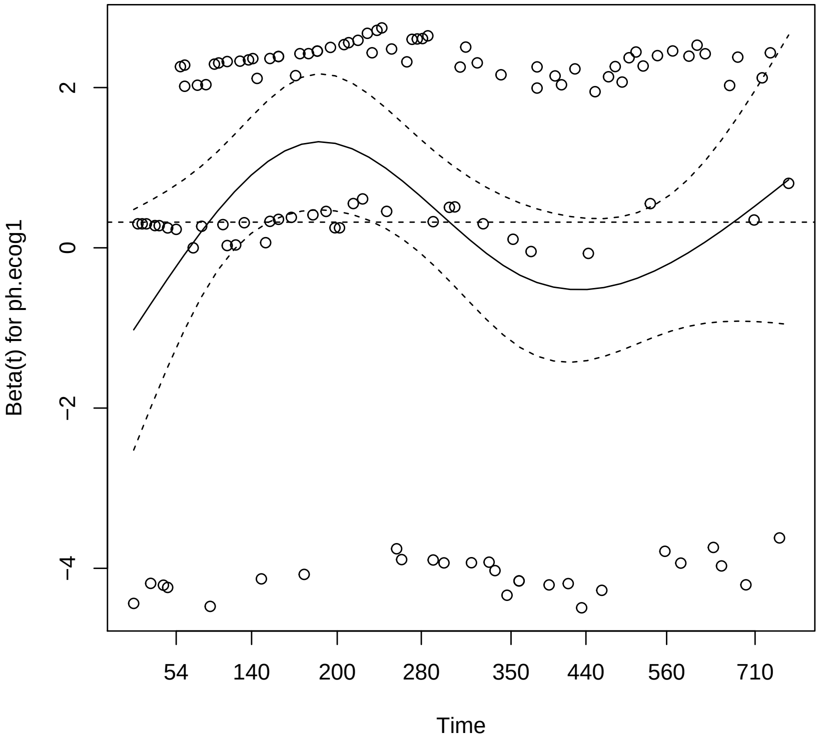
#### 2.2.11.2 Plots

##### 2.2.11.2.1 Plots of coefficients over time

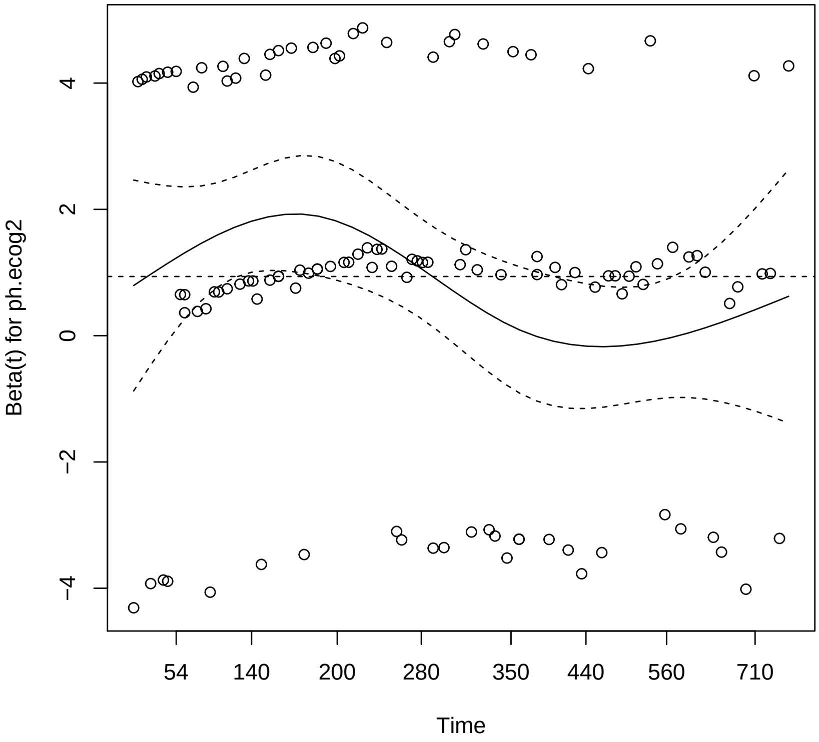
```
# sex=female
plot(cox.zph(cox_lca1), var = 1)
abline(coef(cox_lca1)[1], 0, lty = 2)
```



```
# ph.ecog=1
plot(cox.zph(cox_lca1), var = 2)
abline(coef(cox_lca1)[2], 0, lty = 2)
```



```r
# ph.ecog=2
plot(cox.zph(cox_lca1), var = 3)
abline(coef(cox_lca1)[3], 0, lty = 2)
```



The points scattered fairly equally above and below the estimated coefficient lines over time. The points jumping above and below maybe because of categorical predictors, may look better if we have numerical predictors.

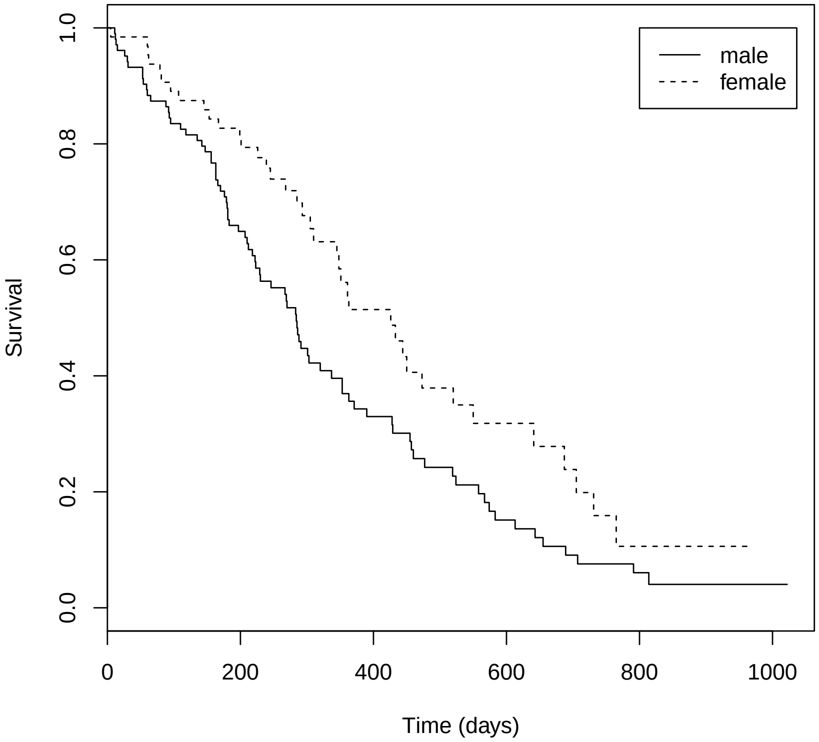
#### 2.2.11.2.2 KM survival curves

KM survival curves for male and female,

```
sur_lca_sex = survfit(Surv(time, status) ~ sex, data = lca)
sur_lca_sex
```

```
## Call: survfit(formula = Surv(time, status) ~ sex, data = lca)
##
##              n events median 0.95LCL 0.95UCL
## sex=male   103     82    284     223     353
## sex=female  64     38    426     345     641
```

```
plot(sur_lca_sex, xlab = "Time (days)",
     ylab = "Survival", lty = 1:2)
legend(800, 1, c("male", "female"), lty = 1:2)
```



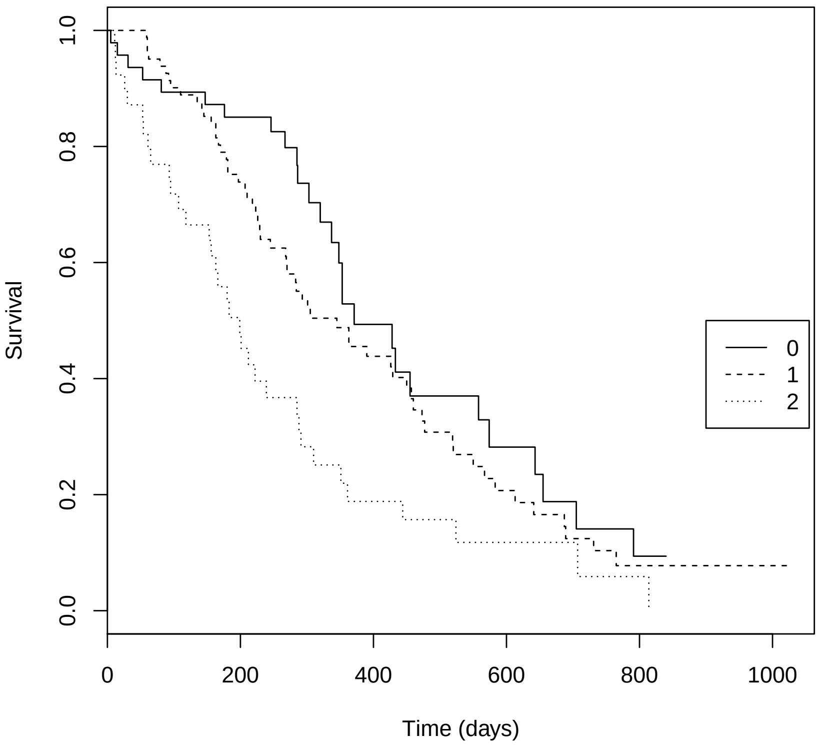
The lines are clearly parallel to each other, indicating the hazards are proportionate over time.

KM survival curves for ph.ecog of 0, 1 and 2,

```
sur_lca_ph.ecog = survfit(Surv(time, status) ~ ph.ecog, data = lca)
sur_lca_ph.ecog
```

```
## Call: survfit(formula = Surv(time, status) ~ ph.ecog, data = lca)
##
##             n events median 0.95LCL 0.95UCL
## ph.ecog=0 47     27    371     337     643
## ph.ecog=1 81     59    345     269     460
## ph.ecog=2 39     34    199     156     291
```

```
plot(sur_lca_ph.ecog, xlab = "Time (days)",
     ylab = "Survival", lty = 1:3)
legend(900, 0.5, c("0", "1", "2"), lty = 1:3)
```



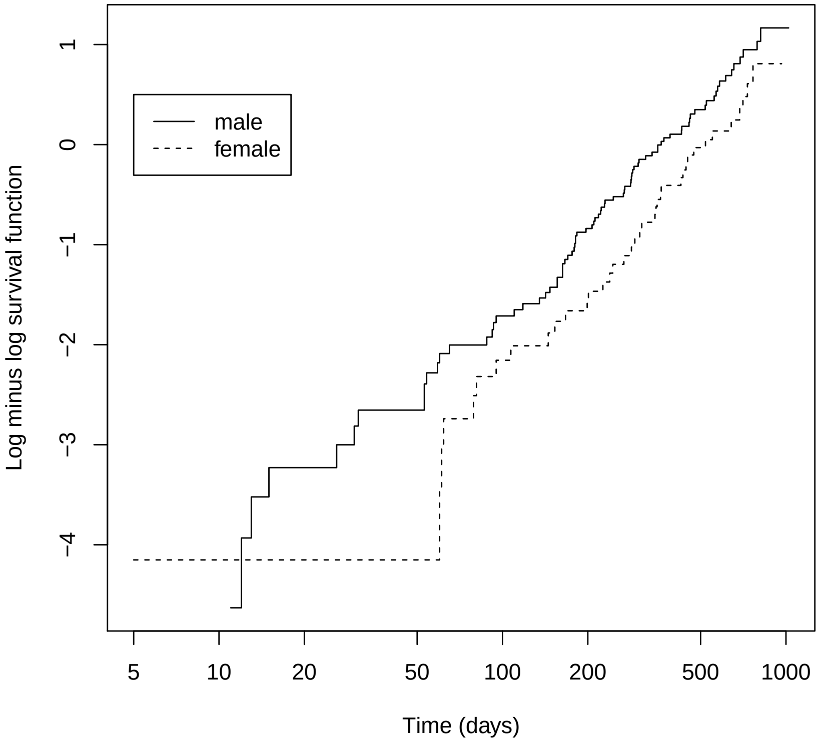
`ph.ecog` $= 2$ looks less parallel to the other two curves. It could be a violation of the PH assumption.

#### 2.2.11.2.3 Log minus log (LML) survival function plot

LML is also called *log cumulative hazard* plot. Remember that we can obtain cumulative hazard function $H(t)$ from cumulative survival function $S(t)$, because $H_t = -log_e S(t)$. LML is just $log_e(H_t)$.

LML plot for male and female,

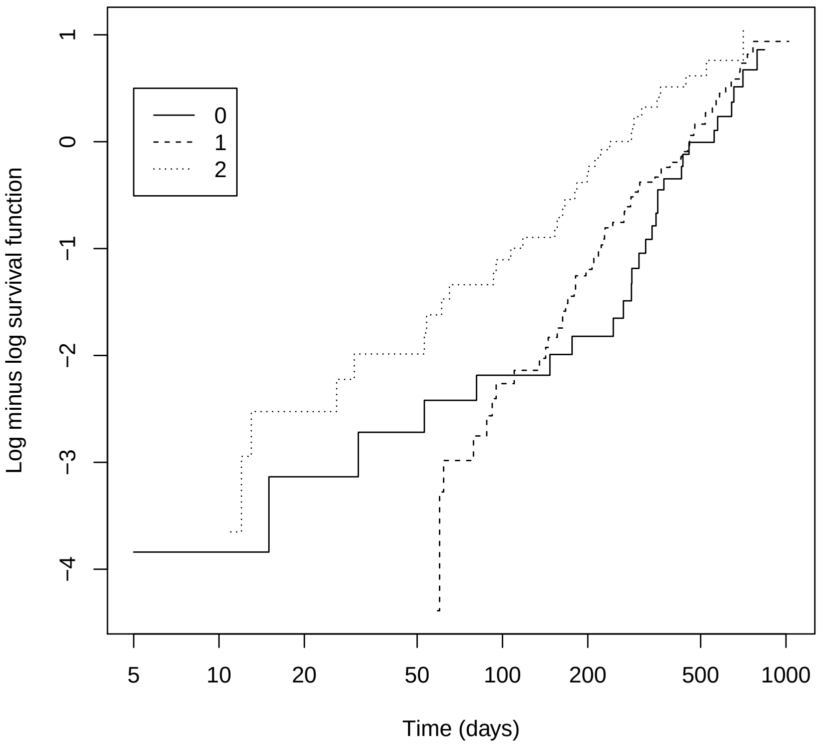
```
plot(sur_lca_sex, fun = "cloglog", xlab = "Time (days)",
     ylab = "Log minus log survival function", lty = 1:2)
legend(5, 0.5, c("male", "female"), lty = 1:2)
```



The lines are parallel, especially prominent for time period of > 50 days.

LML plot for ph.ecog of 0, 1 and 2,

```
plot(sur_lca_ph.ecog, fun = "cloglog", xlab = "Time (days)",
     ylab = "Log minus log survival function", lty = 1:3)
legend(5, 0.5, c("0", "1", "2"), lty = 1:3)
```



At longer time period (> 100 days), the lines look more parallel.

Judging from KM curve and LML plots, there is probably a violation of PH assumption. But we should keep in mind that KM curves and LML plots show us the survivals/hazards by group from basic survival curve data (univariable), thus just serve as rough PH assumption check.

### 2.2.12 Interpretation

At this point we have decided on the final model,

```
cox_lca_final = cox_lca1
summary(cox_lca_final)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ sex + ph.ecog, data = lca)
##
##   n= 167, number of events= 120
##
##              coef exp(coef) se(coef)      z Pr(>|z|)
## sexfemale -0.5076    0.6019   0.1970 -2.576 0.009983 **
## ph.ecog1   0.3204    1.3776   0.2331  1.374 0.169289
## ph.ecog2   0.9368    2.5518   0.2592  3.614 0.000301 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           exp(coef) exp(-coef) lower .95 upper .95
## sexfemale    0.6019     1.6614    0.4091    0.8856
## ph.ecog1     1.3776     0.7259    0.8724    2.1753
## ph.ecog2     2.5518     0.3919    1.5354    4.2410
##
## Concordance= 0.646  (se = 0.03 )
## Rsquare= 0.111   (max possible= 0.998 )
## Likelihood ratio test= 19.58  on 3 df,   p=2e-04
## Wald test            = 20.21  on 3 df,   p=2e-04
## Score (logrank) test = 20.97  on 3 df,   p=1e-04
```

- Female has lower hazard with HR = 0.60 (40% lower) than male, controlling for other predictors.
- ECOG score 1 has higher hazard with HR = 1.38 (38% higher) than ECOG score 0, controlling for other predictors.
- ECOG score 2 has higher hazard with HR = 2.55 (155% higher) than ECOG score 0, controlling for other predictors.

### 2.2.13 Model equations

Log$_e$ hazard ratio is given by,

$$log_e\left(\frac{h(t)}{h_0(t)}\right) = log_e HR = -0.51 * sex\ (female) + 0.32 \times ph.ecog\ (1) + 0.94 \times ph.ecog\ (2)$$

Hazard ratio is easily obtained by,

$$\frac{h(t)}{h_0(t)} = HR = e^{-0.51*sex\ (female)+0.32\times ph.ecog\ (1)+0.94\times ph.ecog\ (2)}$$

Lastly, hazard (or risk in R) can be obtained from this equation,

$$h(t) = h_0(t) \times e^{-0.51*sex\ (female)+0.32\times ph.ecog\ (1)+0.94\times ph.ecog\ (2)}$$

Hazard/risk is given by `predict(..., type = "risk")` in R. $h_0(t)$ is found by setting all values of the predictors to baseline values. In our case, by setting `sex = 0` (“male”) and `ph.ecog = 0` (i.e. dummy variables ph.ecog1 = 0, ph.ecog2 = 0).

### 2.2.14 Prediction

#### 2.2.14.1 HR and hazard

We start by adding predicted hazard to our sample,

```
lca$hazard = predict(cox_lca_final, type = "risk")
```

To add HR, we need to find $h_0(t)$, the baseline hazard by predicting for, `sex = 0` (“male”) and `ph.ecog = 0` (i.e. dummy variables ph.ecog1 = 0, ph.ecog2 = 0),

```
h0_t = predict(cox_lca_final, list(sex = "male", ph.ecog = "0"), type = "risk")
h0_t  # h0(t)
```

```
##         1
## 0.8355948
```

followed by $HR = h(t)/h_0(t)$,

```
# HR = h(t)/h0(t)
lca$hr = lca$hazard/h0_t
```

To see whether we did it right, view the first 20 observations in the sample,

```
head(lca[c("sex", "ph.ecog", "time", "status", "hazard", "hr")], 20)
```

```
##       sex ph.ecog time status    hazard        hr
## 2    male       0  455      1 0.8355948 1.0000000
## 4    male       1  210      1 1.1511334 1.3776214
## 6    male       1 1022      0 1.1511334 1.3776214
## 7  female       2  310      1 1.2834381 1.5359575
## 8  female       2  361      1 1.2834381 1.5359575
## 9    male       1  218      1 1.1511334 1.3776214
## 10   male       2  166      1 2.1322541 2.5517798
## 11   male       1  170      1 1.1511334 1.3776214
## 15   male       1  567      1 1.1511334 1.3776214
## 17   male       1  613      1 1.1511334 1.3776214
## 18   male       2  707      1 2.1322541 2.5517798
## 19 female       2   61      1 1.2834381 1.5359575
## 21   male       1  301      1 1.1511334 1.3776214
## 22 female       0   81      1 0.5029580 0.6019161
## 24   male       0  371      1 0.8355948 1.0000000
## 26 female       1  520      1 0.6928858 0.8292126
## 27   male       0  574      1 0.8355948 1.0000000
## 28   male       2  118      1 2.1322541 2.5517798
## 29   male       1  390      1 1.1511334 1.3776214
## 30   male       2   12      1 2.1322541 2.5517798
```

Now, we proceed to obtain the hazard and HR for a subject (`sex = "female"`, `ph.ecog = "2"`). Hazard,

```
predict(cox_lca_final, list(sex = "female", ph.ecog = "2"), type = "risk")  # hazard
```

```
##        1
## 1.283438
```

There are two ways to obtain HR. First, our equation for HR above,

```
exp(coef(cox_lca_final)[1]*1 + coef(cox_lca_final)[2]*0 + coef(cox_lca_final)[3]*1)  # HR
```

```
## sexfemale
##  1.535957
```

We can also obtain HR by dividing hazard by the baseline hazard `h0_t` that we had before,

```
predict(cox_lca_final, list(sex = "female", ph.ecog = "2"), type = "risk")/h0_t  # HR
```

```
##        1
## 1.535957
```

```
# we utilize the baseline hazard h0(t)
```

Then we obtain the hazard and HR for a data frame,

```
new_data = data.frame(sex = c("male", "male", "male", "female", "female", "female"),
                      ph.ecog = c("0", "1", "2", "0", "1", "2"))
new_data
```

```
##      sex ph.ecog
## 1   male       0
## 2   male       1
## 3   male       2
## 4 female       0
## 5 female       1
## 6 female       2
```

```
new_hazard = hazard = predict(cox_lca_final, new_data, type = "risk")
new_hr = new_hazard/h0_t
data.frame(new_data, hazard = round(new_hazard, 3), hr = round(new_hr, 3))
```

```
##      sex ph.ecog hazard    hr
## 1   male       0  0.836 1.000
## 2   male       1  1.151 1.378
## 3   male       2  2.132 2.552
## 4 female       0  0.503 0.602
## 5 female       1  0.693 0.829
## 6 female       2  1.283 1.536
```

#### 2.2.14.2 Median survival times and survival probabilities

To obtain survival probabilities, Rizopoulos (2017) gives a good guide here:

> http://www.drizopoulos.com/courses/emc/ep03_%20survival%20analysis%20in%20r%20companion

Here we obtain the median survival time and probabilities for a subject,

```
# simple, sex = "female", ph.ecog = "2"
new_cox1 = survfit(cox_lca_final, newdata = list(sex = "female", ph.ecog = "2"))
new_cox1  # median survival time
```

```
## Call: survfit(formula = cox_lca_final, newdata = list(sex = "female",
##     ph.ecog = "2"))
##
##       n  events  median 0.95LCL 0.95UCL
##     167     120     285     218     429
```

```
summary(new_cox1, times = 100)  # survival at 100 days
```

```
## Call: survfit(formula = cox_lca_final, newdata = list(sex = "female",
##     ph.ecog = "2"))
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##   100    143      24     0.83  0.0434        0.749        0.919
```

```
summary(new_cox1, times = c(100, 200, 300))  # survival at 100, 200 and 300 days
```

```
## Call: survfit(formula = cox_lca_final, newdata = list(sex = "female",
##     ph.ecog = "2"))
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##   100    143      24    0.830  0.0434        0.749        0.919
##   200    111      24    0.656  0.0672        0.536        0.802
##   300     67      25    0.449  0.0827        0.313        0.644
```

Now for a data frame,

```
new_data = data.frame(sex = c("male", "male", "male", "female", "female", "female"),
                      ph.ecog = c("0", "1", "2", "0", "1", "2"))
new_data
```

```
##      sex ph.ecog
## 1   male       0
## 2   male       1
## 3   male       2
## 4 female       0
## 5 female       1
## 6 female       2
```

```
new_cox2 = survfit(cox_lca_final, newdata = new_data)
new_cox2  # median survival times
```

```
## Call: survfit(formula = cox_lca_final, newdata = new_data)
##
##     n events median 0.95LCL 0.95UCL
## 1 167    120    361     288     558
## 2 167    120    291     245     371
## 3 167    120    199     163     284
## 4 167    120    550     426      NA
## 5 167    120    429     337     641
## 6 167    120    285     218     429
```

```
summary(new_cox2, times = 100)  # survival at 100 days
```

```
## Call: survfit(formula = cox_lca_final, newdata = new_data)
##
##  time n.risk n.event survival1 survival2 survival3 survival4 survival5
##   100    143      24     0.886     0.846     0.733     0.929     0.904
##  survival6
##       0.83
```

```
summary(new_cox2, times = c(100, 200, 300))  # survival at 100, 200 and 300 days
```

```
## Call: survfit(formula = cox_lca_final, newdata = new_data)
##
##  time n.risk n.event survival1 survival2 survival3 survival4 survival5
##   100    143      24     0.886     0.846     0.733     0.929     0.904
##   200    111      24     0.760     0.685     0.496     0.848     0.796
##   300     67      25     0.594     0.488     0.264     0.731     0.649
##  survival6
##      0.830
##      0.656
##      0.449
```

## 2.3 Exercises

1. Present the results in a table (follow Arifin et al. (2016)). You may follow the way of presentation for logistic regression.
2. Now, include `ph.kano` and `pat.karno` in the multivariable analysis. What do you get? *Hint: Interaction?
3. Perform Cox PH on builtin `GBSG2` dataset (in `PH.data` package).

# Chapter 3

# References

Arifin, W. N., Sarimah, A., Norsa'adah, B., Majdi, Y. N., Siti-Azrin, A. H., Imran, M. K., … Naing, L. (2016). Reporting statistical results in medical journals. *The Malaysian Journal of Medical Sciences: MJMS*, *23*(5), 1.

Chongsuvivatwong, V. (2018). *EpiDisplay: Epidemiological data display package.* Retrieved from https://CRAN.R-project.org/package=epiDisplay

Grana, C., Chinol, M., Robertson, C., Mazzetta, C., Bartolomei, M., De Cicco, C., … Paganelli, G. (2002). Pretargeted adjuvant radioimmunotherapy with yttrium-90-biotin in malignant glioma patients: A pilot study. *British Journal of Cancer*, *86*(2), 207.

Hosmer, D., Lemeshow, S., & Sturdivant, R. (2013). *Applied logistic regression.* Wiley. Retrieved from https://books.google.com.my/books?id=bRoxQBIZRd4C

Hothorn, T., Hornik, K., van de Wiel, M. A., Winell, H., & Zeileis, A. (2017). *Coin: Conditional inference procedures in a permutation test framework.* Retrieved from https://CRAN.R-project.org/package=coin

Miller, R. (1997). *Survival analysis.* John Wiley & Sons.

Rizopoulos, D. (2017). EP03: Survival analysis in R companion. Retrieved September 12, 2018, from http://www.drizopoulos.com/courses/emc/ep03_%20survival%20analysis%20in%20r%20companion#survival-probabilities-from-cox-models

RStudio Team. (2018). *RStudio: Integrated development environment for R.* Boston, MA: RStudio, Inc. Retrieved from http://www.rstudio.com/

Therneau, T. M. (2018). *Survival: Survival analysis.* Retrieved from https://CRAN.R-project.org/package=survival

Woodward, M. (2013). *Epidemiology: Study design and data analysis.* Boston, FL, USA: CRC Press.