Time-to-Event Analysis: Practical 1 Solutions

Instructor: Graeme L. Hickey

27th October 2014

1 Problem 1

We first read in the dataset and save it as `dialysis`.

```
library(survival)
```

```
## Loading required package: splines
```

```
dialysis <- read.table("dialysis.data", header = TRUE)
```

Preliminary checks suggest the dataset has been read into R correctly.

```
head(dialysis)
```

```
##   id days dead method age
## 1  1 3444    0      1  41
## 2  2 3499    0      1  35
## 3  3 6230    0      1  41
## 4  4 1324    1      1  67
## 5  5 6230    0      1  29
## 6  6  147    1      0  55
```

```
str(dialysis)
```

```
## 'data.frame':    124 obs. of  5 variables:
##  $ id    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ days  : int  3444 3499 6230 1324 6230 147 709 6230 422 5096 ...
##  $ dead  : int  0 0 0 1 0 1 1 0 1 0 ...
##  $ method: int  1 1 1 1 1 0 1 1 0 0 ...
##  $ age   : int  41 35 41 67 29 55 54 42 45 46 ...
```

We now look at the summary statistics for the dataset.

```r
summary(dialysis)
```

```
##        id              days             dead            method
##  Min.   :  1.00   Min.   :  14.0   Min.   :0.0000   Min.   :0.0
##  1st Qu.: 31.75   1st Qu.: 630.8   1st Qu.:0.0000   1st Qu.:0.0
##  Median : 62.50   Median :2684.0   Median :1.0000   Median :0.5
##  Mean   : 62.50   Mean   :2777.6   Mean   :0.5242   Mean   :0.5
##  3rd Qu.: 93.25   3rd Qu.:4635.2   3rd Qu.:1.0000   3rd Qu.:1.0
##  Max.   :124.00   Max.   :6233.0   Max.   :1.0000   Max.   :1.0
##       age
##  Min.   :18.00
##  1st Qu.:38.00
##  Median :52.50
##  Mean   :51.38
##  3rd Qu.:66.25
##  Max.   :85.00
```

```r
fit.dialysis <- survfit(Surv(days, dead) ~ method, data = dialysis)
fit.dialysis
```

```
## Call: survfit(formula = Surv(days, dead) ~ method, data = dialysis)
##
##          records n.max n.start events median 0.95LCL 0.95UCL
## method=0      62    62      62     35    883     551      NA
## method=1      62    62      62     30   4624    2847      NA
```

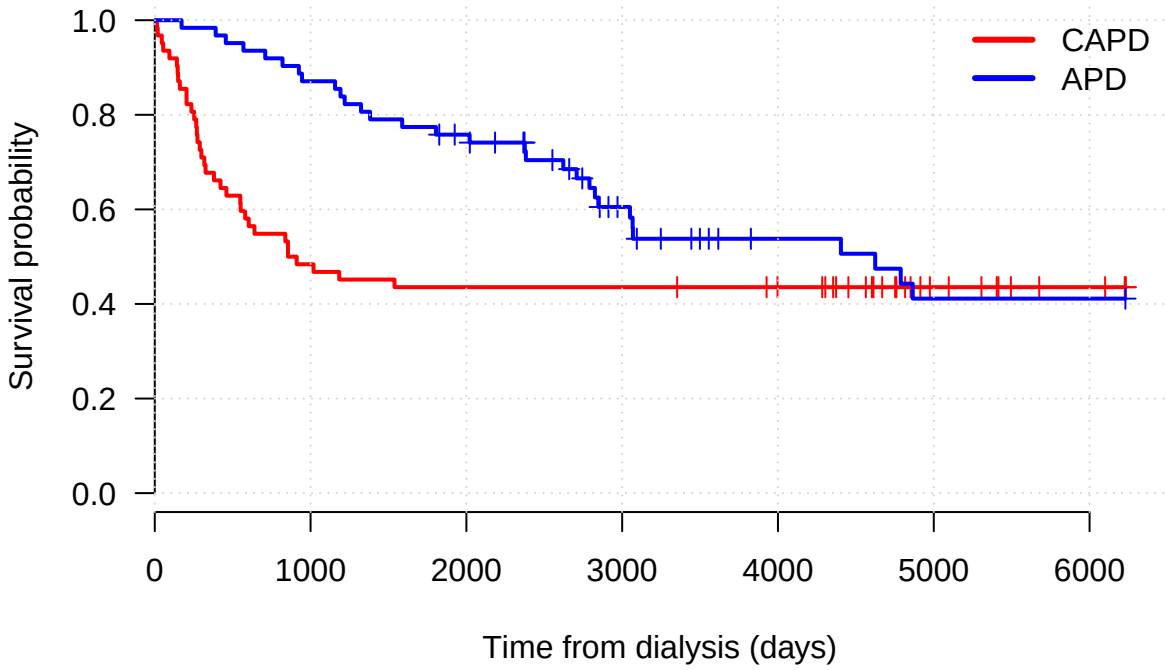
Summary statistics indicate there are 62 patients in each treatment group and that follow-up ranged from 14 to 6233 days. Patients were aged 51 years old on average, varying between 18 and 85 years. 52% of patients died during follow-up: 35 patients in the CAPD group and 30 patients in the APD group. We further find that the median survival times were 883 and 4624 days for CAPD and APD respectively.

The Kaplan-Meier curves are the same as shown in the lecture.

```r
plot(fit.dialysis,
  xlab = "Time from dialysis (days)",
  ylab = "Survival probability",
  bty = "n",
  col = c(2, 4),
  lwd = 2,
  las = 1)

grid()
```

```
legend(
  "topright",
  bty = "n",
  c("CAPD", "APD"),
  col = c(2, 4),
  lty = c(1, 1),
  lwd = 3)
```



In the short term (first 2 years) we find that survival drops much faster in the CAPD group compared to the APD group. However, after $\sim$ 13-years, the survival probabilities are similar.

We use the log-rank test to assess whether there is any significant difference in the survival of patients receiving dialysis by each of these two methods.

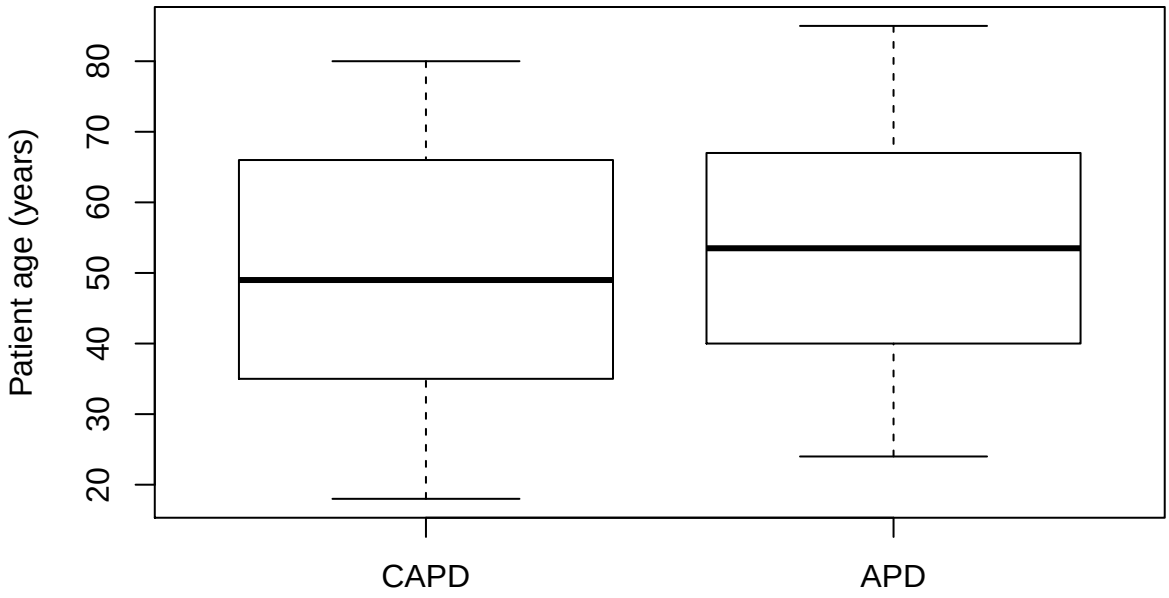
```
survdiff(Surv(days, dead) ~ method, data = dialysis)
```

```
## Call:
## survdiff(formula = Surv(days, dead) ~ method, data = dialysis)
##
##             N Observed Expected (O-E)^2/E (O-E)^2/V
## method=0 62       35     27.4      2.14      3.73
## method=1 62       30     37.6      1.55      3.73
##
##  Chisq= 3.7  on 1 degrees of freedom, p= 0.0534
```

Whilst there is clearly some evidence of a difference, it is marginally above the standard $P = 0.05$ threshold routinely used to classify a result a statistical significant.

An exploratory look at the difference in patient age between the two treatment groups shows no evidence of a treatment selection bias on the basic of age at time of dialysis. The Student's t-test failed to reject a difference between the population means of the two groups ($P = 0.27$).

```
boxplot(age ~ method, data = dialysis,
        names = c("CAPD", "APD"),
        ylab = "Patient age (years)")
```




```
t.test(age ~ method, data = dialysis)
```

```
##
##  Welch Two Sample t-test
##
## data:  age by method
## t = -1.1151, df = 118.302, p-value = 0.2671
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -9.267632  2.590212
## sample estimates:
## mean in group 0 mean in group 1
##        49.70968        53.04839
```

We create a new variable that identifies whether a patient is above or below the age of 50 years, and append this to the dialysis dataset.

```r
dialysis$age50 <- (dialysis$age > 50)
summary(dialysis$age50)
```

```
##    Mode   FALSE    TRUE    NA's
## logical      58      66       0
```

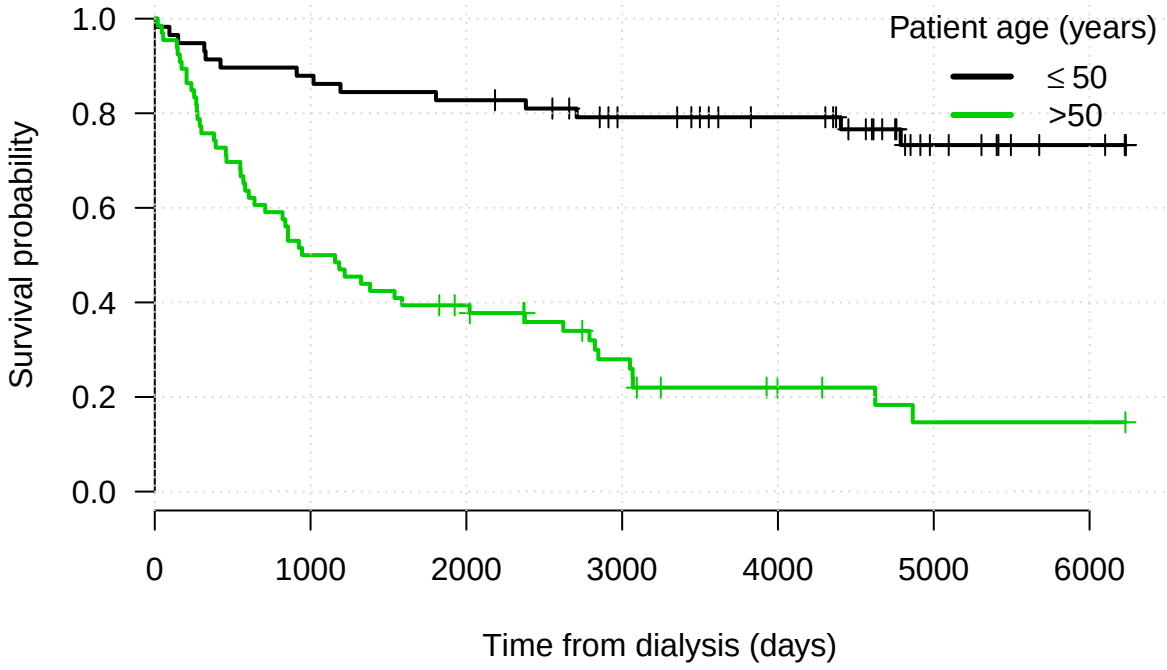
Using this new variable, we can estimate the Kaplan-Meier functions for the two age groups, plot the survival functions and perform a log-rank test.

```r
fit.age50 <- survfit(Surv(days, dead) ~ age50, data = dialysis)
```

```r
plot(fit.age50,
  xlab = "Time from dialysis (days)",
  ylab = "Survival probability",
  bty = "n",
  col = c(1, 3),
  lwd = 2,
  las = 1)

grid()

legend(
  "topright",
  bty = "n",
  title = "Patient age (years)",
  c(expression(""<=50), " >50"),
  col = c(1, 3),
  lty = c(1, 1),
  lwd = 3)
```




```
survdiff(Surv(days, dead) ~ age50, data = dialysis)
```

```
## Call:
## survdiff(formula = Surv(days, dead) ~ age50, data = dialysis)
##
##               N Observed Expected (O-E)^2/E (O-E)^2/V
## age50=FALSE 58       14     38.3      15.4      39.1
## age50=TRUE  66       51     26.7      22.0      39.1
##
##  Chisq= 39.1  on 1 degrees of freedom, p= 4.02e-10
```

We find that there is a very significant difference ($P < 0.001$). Therefore we conclude that patients aged greater than 50-years survive for shorter times than those aged 50-years and younger.

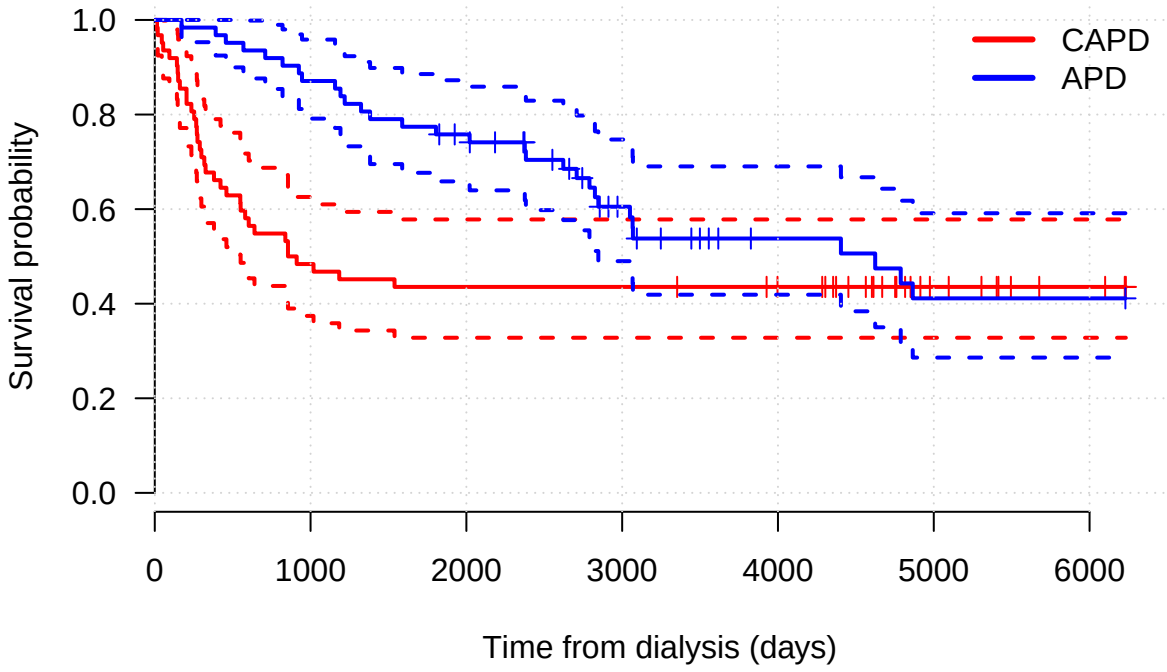
It is worth emphasizing an important limitation in this analysis, namely that we have dichotomized age. This ignores potentially important information. Furthermore, the threshold of 50-years was entirely arbitrary. It is not uncommon to find researchers dichotomising variables such as age in survival analyses despite the known limitations. Whilst the log-rank test requires categorical variables, time-to-event regression modelling does not, which we will cover in the next lecture.

To add confidence limits to the Kaplan-Meier curves stratifying survival times by treatment method, we include the argument `conf.int = TRUE` in the plot function

```r
plot(fit.dialysis,
  xlab = "Time from dialysis (days)",
  ylab = "Survival probability",
  bty = "n",
  conf.int = TRUE,
  col = c(2, 4),
  lwd = 2,
  las = 1)

grid()

legend(
  "topright",
  bty = "n",
  c("CAPD", "APD"),
  col = c(2, 4),
  lty = c(1, 1),
  lwd = 3)
```



The confidence intervals are quite tight to begin with, and non-overlapping, which is consistent with our findings above. The confidence intervals widen as time progresses.

2 Problem 2

We begin by reading in the `calf_pneu` data, saving it as `calf`, and running some basic checks of the data.

```
##       calf           stock          days              pn
##  Min.   : 1.00   Min.   :0.0   Min.   : 27.00   Min.   :0.0
##  1st Qu.: 6.75   1st Qu.:0.0   1st Qu.: 89.75   1st Qu.:0.0
##  Median :12.50   Median :0.5   Median :113.00   Median :0.5
##  Mean   :12.50   Mean   :0.5   Mean   :107.88   Mean   :0.5
##  3rd Qu.:18.25   3rd Qu.:1.0   3rd Qu.:118.50   3rd Qu.:1.0
##  Max.   :24.00   Max.   :1.0   Max.   :150.00   Max.   :1.0
```

At this point we are well versed in estimating and plotting Kaplan-Meier survival function estimates.

```
fit.calf <- survfit(Surv(days, pn) ~ stock, data = calf)
fit.calf
```

```
## Call: survfit(formula = Surv(days, pn) ~ stock, data = calf)
##
##         records n.max n.start events median 0.95LCL 0.95UCL
## stock=0      12    12      12      4     NA     123      NA
## stock=1      12    12      12      8    113      79      NA
```

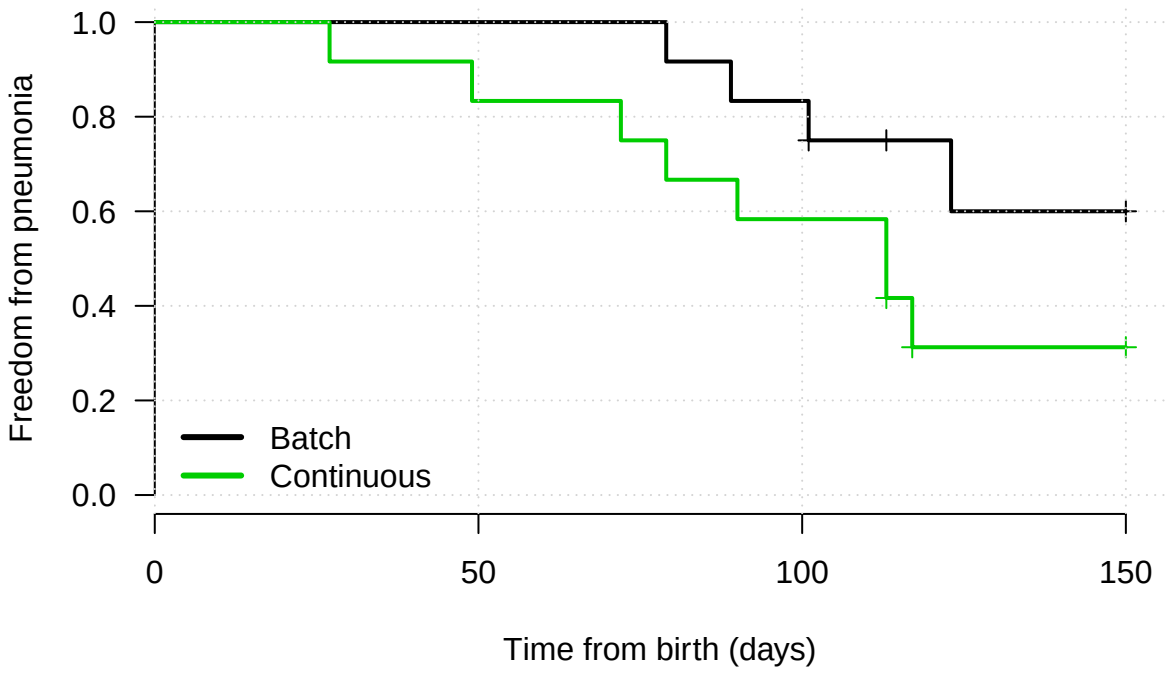
```
summary(fit.calf)
```

```
## Call: survfit(formula = Surv(days, pn) ~ stock, data = calf)
##
##                 stock=0
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    79     12       1    0.917  0.0798        0.773            1
##    89     11       1    0.833  0.1076        0.647            1
##   101     10       1    0.750  0.1250        0.541            1
##   123      5       1    0.600  0.1673        0.347            1
##
##                 stock=1
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    27     12       1    0.917  0.0798        0.773        1.000
##    49     11       1    0.833  0.1076        0.647        1.000
##    72     10       1    0.750  0.1250        0.541        1.000
##    79      9       1    0.667  0.1361        0.447        0.995
##    90      8       1    0.583  0.1423        0.362        0.941
##   113      7       2    0.417  0.1423        0.213        0.814
##   117      4       1    0.312  0.1398        0.130        0.751
```

```
plot(fit.calf,
  xlab = "Time from birth (days)",
  ylab = "Freedom from pneumonia",
  bty = "n",
  col = c(1, 3),
  lwd = 2,
  las = 1)

grid()

legend(
  "bottomleft",
  bty = "n",
  c("Batch", "Continuous"),
  col = c(1, 3),
  lty = c(1, 1),
  lwd = 3)
```



The Kaplan-Meier curves suggests that there is a difference in the time-to-pneumonia, and that calves raised in continuous housing are infected with calf pneumonia sooner than those raised in batch housing. To test this we apply the log-rank test.

```
survdiff(Surv(days, pn) ~ stock, data = calf)
```

```
## Call:
## survdiff(formula = Surv(days, pn) ~ stock, data = calf)
```

```
##
##           N Observed Expected (O-E)^2/E (O-E)^2/V
## stock=0 12        4     6.89      1.21      2.99
## stock=1 12        8     5.11      1.63      2.99
##
##  Chisq= 3  on 1 degrees of freedom, p= 0.084
```

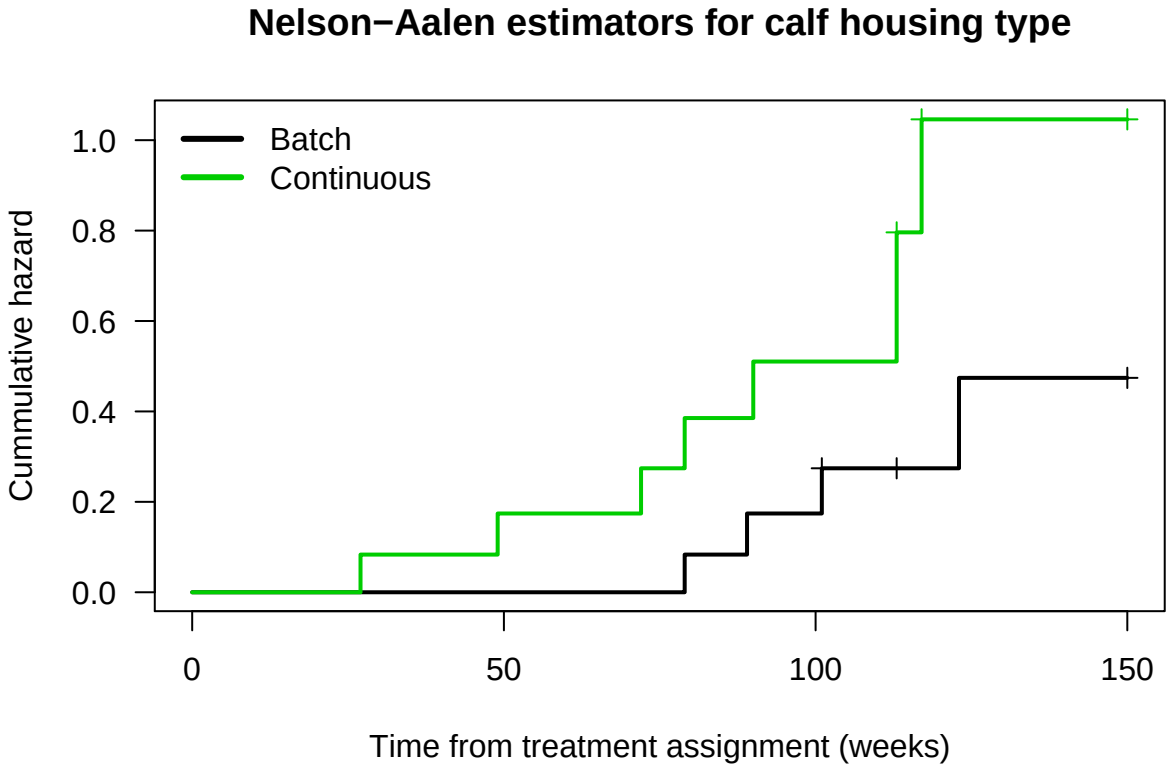
The log-rank test yields a P-value of 0.084, which although not significant at the $P < 0.05$, level, it does suggest that there is some slight evidence of a difference in failure times.

We take advantage of the existing R functions to estimate and plot the cumulative hazard function.

```
fit.calf.flem <- survfit(Surv(days, pn) ~ stock, data = calf,
                     type = "fleming")

plot(fit.calf.flem, fun = "cumhaz",
     col = c(1, 3),
     xlab = "Time from treatment assignment (weeks)",
     ylab = "Cummulative hazard",
     main = "Nelson-Aalen estimators for calf housing type",
     las = 1,
     lwd = 2)

legend(
  "topleft",
  bty = "n",
  c("Batch", "Continuous"),
  col = c(1, 3),
  lty = c(1, 1),
  lwd = 3)
```



Another way to estimate the Kaplan-Meier estimator for a subset of the data is to use the `subset` argument within the `survfit()` function. Namely, to estimate the Kaplan-Meier survival function for continuous-housed calves only we can use

```
fit.calf1 <- survfit(Surv(days, pn) ~ 1,
                     subset = (stock == 1), # Note the double equals sign
                     data = calf)
```

Note:

1. We set the `subset` argument to `subset = (stock == 1)`, which tells `survfit()` to only use rows of data with `stock` equal to 1, i.e. continuous housing. If you want to check what `stock == 1` is doing, type `calf$stock == 1`.
2. We replaced `~ stock` with `~ 1` since there is now only one type of housing stock in the data subset.

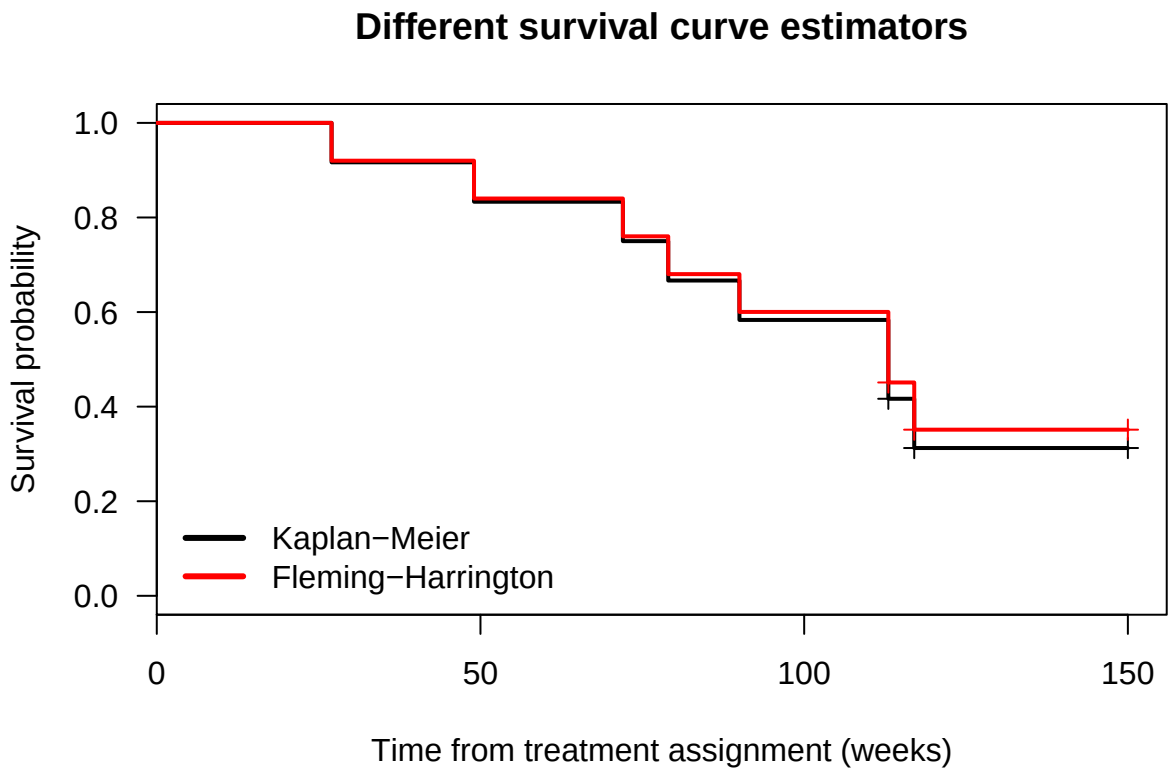
We now calculate a second estimate of the survival curve using the Fleming-Harrington method modifying the `type` argument within the `survfit()` object.

```
fit.calf1.flem <- survfit(Surv(days, pn) ~ 1,
                          subset = (stock == 1),
                          type = "fleming", # Only difference is here
                          data = calf)
```

```r
plot(fit.calf1,
     conf.int = "none",
     xlab = "Time from treatment assignment (weeks)",
     ylab = "Survival probability",
     main = "Different survival curve estimators",
     las = 1,
     lwd = 2)

lines(fit.calf1.flem,
      col = 2,
      lwd = 2,
      conf.int = "none")

legend(
  "bottomleft",
  bty = "n",
  c("Kaplan-Meier", "Fleming-Harrington"),
  col = c(1, 2),
  lty = c(1, 1),
  lwd = 3)
```



We notice that Fleming-Harrington estimator gives a greater estimate of survival [freedom from pneumonia] compared to the Kaplan-Meier estimator, and that this difference is negligible at early times and increases with time.

3 Problem 3

We start with the code from Example 3 that was used to fit and plot the exponential model.

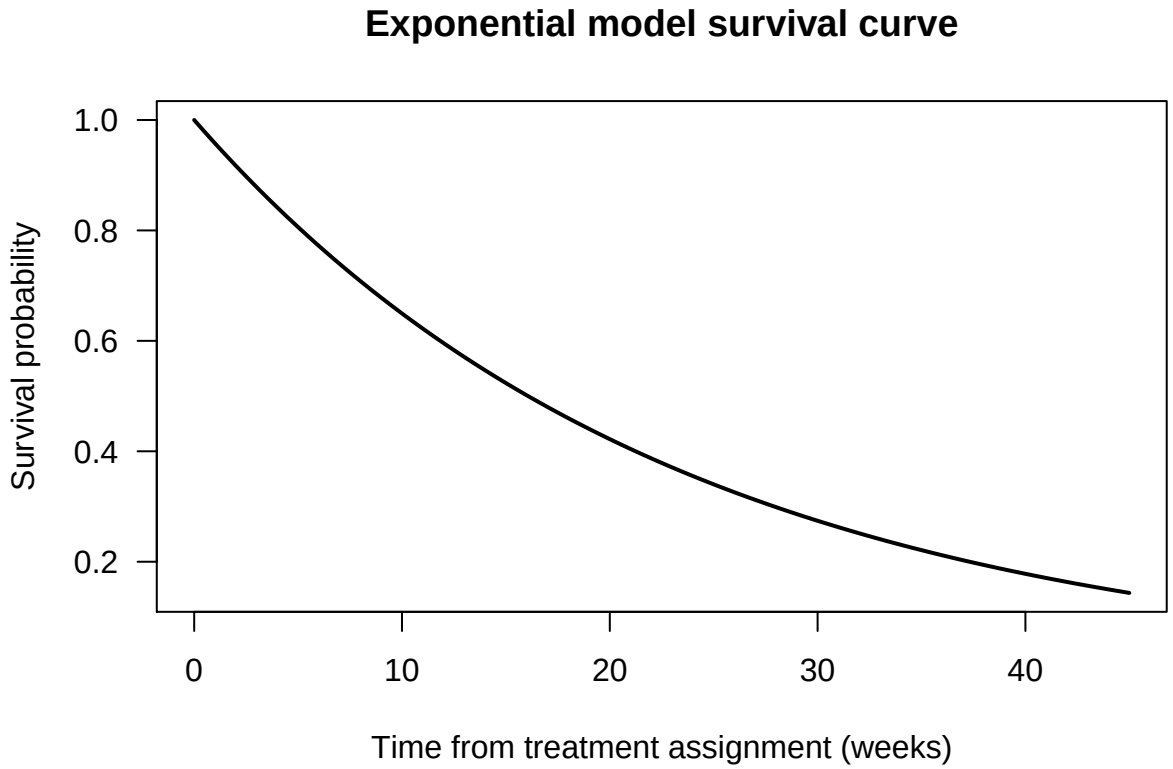
```
aml0 <- aml[aml$x == "Nonmaintained", ]

fit.exp <- survreg(Surv(time, status) ~ 1, data = aml0,
                   dist = "exponential")

intercept <- coef(fit.exp)
lambda.hat <- exp(-intercept)
lambda.hat
```

```
## (Intercept)
##  0.04313725
```

```
# Plot the exponential model survival function
curve(exp(-lambda.hat * x),
      xlim = c(0, max(aml0$time)),
      lwd = 2,
      xlab = "Time from treatment assignment (weeks)",
      ylab = "Survival probability",
      main = "Exponential model survival curve",
      las = 1)
```

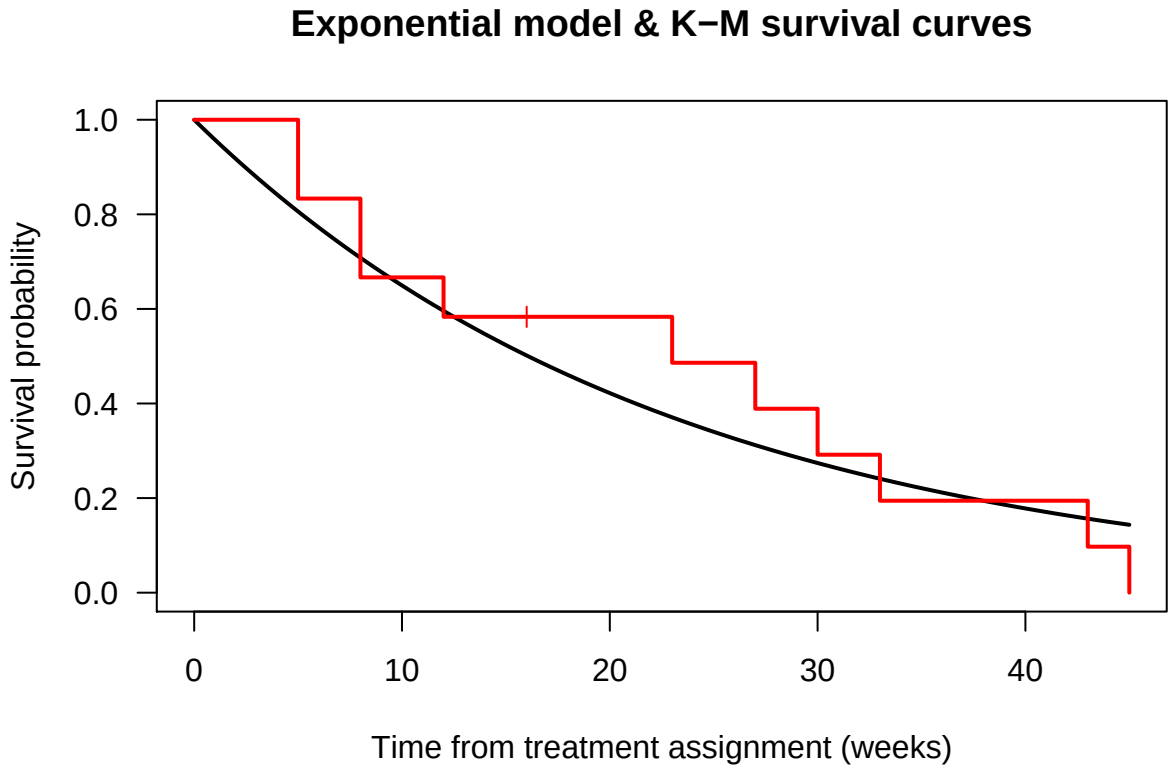


The Kaplan-Meier estimate of the survival function is calculated as

```
fit.km <- survfit(Surv(time, status) ~ 1, data = aml0)
```

We can overlay this plot using the `lines()` function. (**Note:** you need to have the previous plot still displayed, otherwise R has nothing to overlay the lines onto.)

```
lines(fit.km,
      col = 2,
      lwd = 2,
      conf.int = "none")
```



The models align reasonably; however we can see that the exponential model is giving lower failure rates than the Kaplan-Meier estimator.

We fit a more flexible Weibull model instead, and estimate the model parameters p and λ.

```
fit.weib <- survreg(Surv(time, status) ~ 1, data = aml0,
                    dist = "weibull")

intercept <- coef(fit.weib)
scale <- fit.weib$scale

p.hat <- 1 / scale
p.hat
```

```
## [1] 1.573629
```

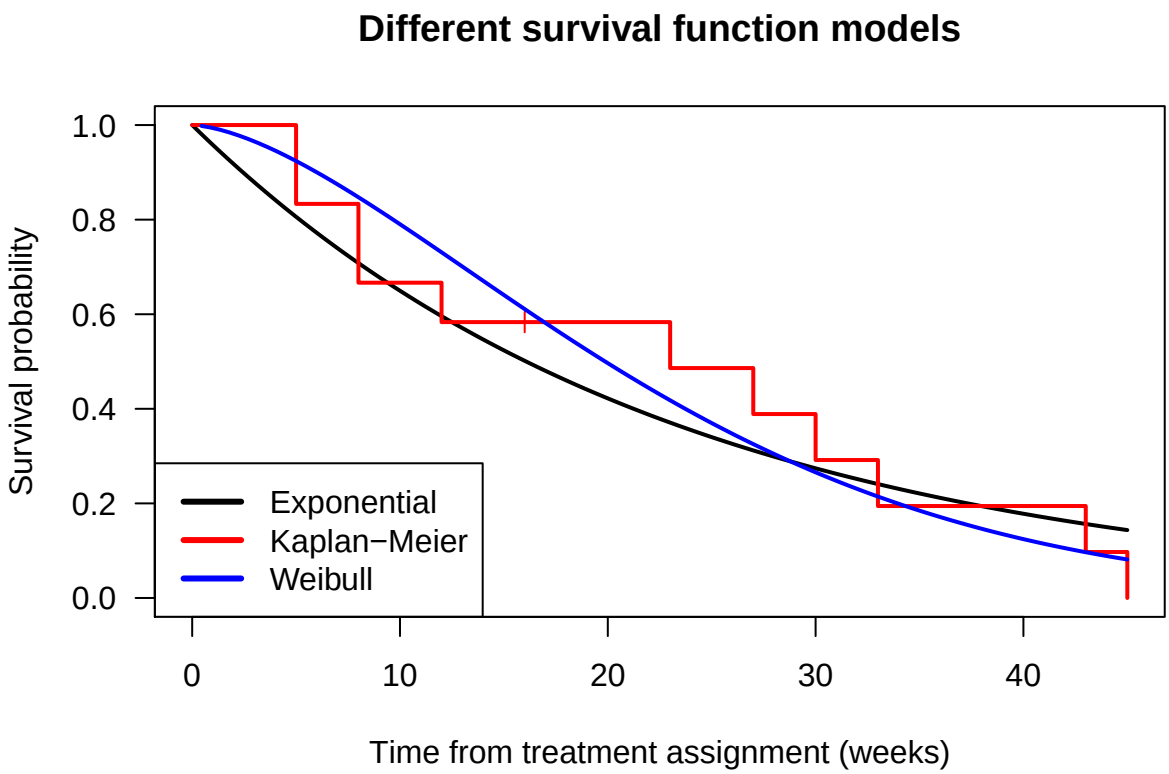
```
lambda.weib.hat <- exp(-p.hat * intercept)
lambda.weib.hat
```

```
## (Intercept)
## 0.006278993
```

We now overlay the Weibull model survival function, recalling from lectures that $S(t) = \exp(-\lambda t^p)$, where we substitute λ and p with their maximum likelihood estimates.

```
curve(exp(-lambda.weib.hat * x^p.hat),
      col = 4,
      lwd = 2,
      add = TRUE)

legend("bottomleft",
       col = c(1, 2, 4),
       c("Exponential", "Kaplan-Meier", "Weibull"),
       lty = rep(1, 3),
       lwd = 3)
```



Inspection of the overlaid curves would suggest the Weibull model survival model gives a closer fit to the Kaplan-Meier fit. In the Weibull model, $\hat{p} = 1.57$, which suggests that an increasing hazard function might be more appropriate than a constant one.

There are a number of methods we can use to compare the exponential and Weibull model fits:

1. Interpret the P-value for the log(scale) coefficient: if it is significantly different from 0 (corresponding to p being significantly different from 1), then it would suggest the exponential model is too simple.
2. We can plot a non-parametric estimate of the log cumulative hazard, $\log\left(\hat{H}(t)\right)$ against log time, $\log(t)$. A straight line would suggest the Weibull function is reasonable, with the special case of the exponential model being appropriate if the slope was 1.
3. We can compare compare models using a likelihood ratio test (as the exponential model is nested within the Weibull model), or use an information criterion approach, e.g. AIC.

Examining the model output for the Weibull model fit

```
summary(fit.weib)
```

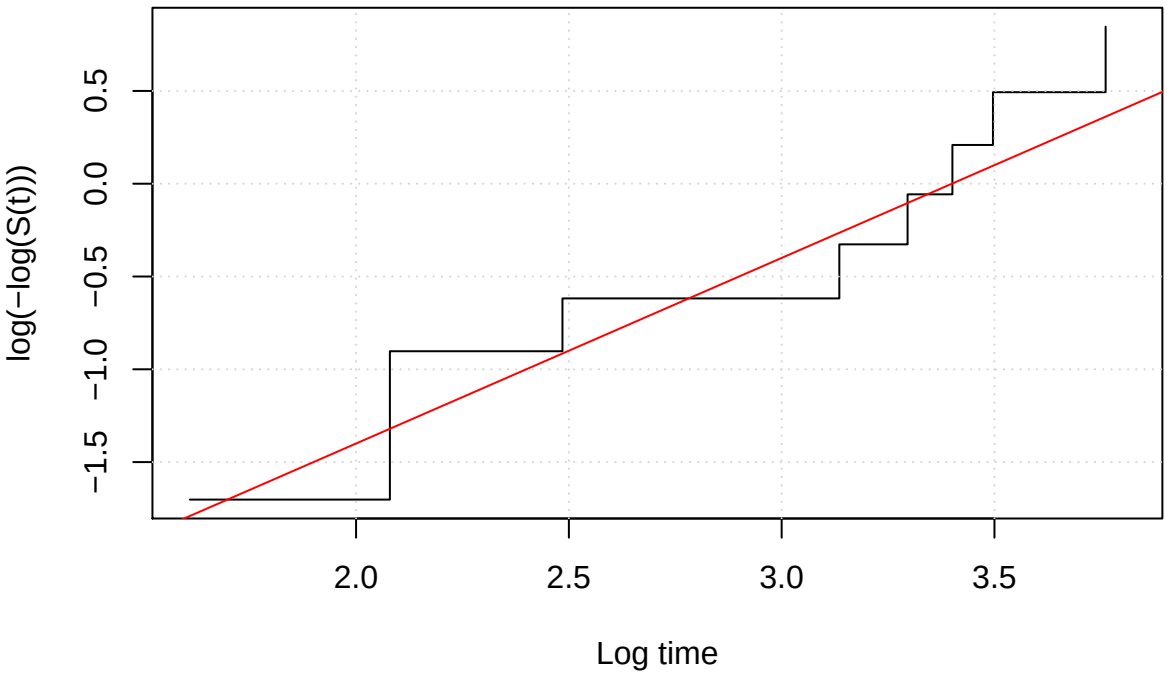
```
##
## Call:
## survreg(formula = Surv(time, status) ~ 1, data = aml0, dist = "weibull")
##               Value Std. Error     z        p
## (Intercept)  3.222      0.198 16.25 2.31e-59
## Log(scale)  -0.453      0.244 -1.86 6.32e-02
##
## Scale= 0.635
##
## Weibull distribution
## Loglik(model)= -44.1   Loglik(intercept only)= -44.1
## Number of Newton-Raphson Iterations: 6
## n= 12
```

we find that the P-value for the log scale is 0.063. As the P-value is not < 0.05, we would not reject the null hypothesis that $p = 1$, i.e. that the simpler exponential model adequately describes the data.

We could use `plot(fit.km, fun = "cloglog")` to plot the log cumulative hazard against log time. We can also do it by hand.

```
plot(log(fit.km$time), log(-log(fit.km$surv)),
     type = "s",
     xlab = "Log time",
     ylab = "log(-log(S(t)))")
grid()

abline(a = -3.4, b = 1, col = "red")
```



The overlaid red line has slope 1. We would conclude that, on the basis of this plot, that a straight-line is a reasonable, and that the data do not grossly deviate from of an exponential model.