Practical session 3: Parameteric survival distributions and regression

September 1 - 25, 2015

1. **Estimation and model checking.** Fit both the exponential and Weibull distributions to describe the overall survival in the veteran data. Compare the fitted survival curves against the Kaplan-Meier estimates of survival. Use a graphical presentation and interpret the results.

   Hint: You can obtain the maximum likelihood estimates of the parameters using the *weibreg* function (in package *eha*). The estimate of the rate parameter ("lambda0") under the exponential model is obtained as follows:

   ```
   veteran.exp0 = weibreg(Surv(time, status) ~ 1,data=veteran,shape=1)
   ```

   Extract the rate estimate as

   ```
    lambda0 = exp(-veteran.exp0$coeff[1]).
   ```

   The shape a and scale b parameters of the Weibull distribution ($S(t) = \exp\{-(t/b)^a\}$) are estimated as follows (note the order of 'coefficients' in the output):

   ```
   veteran.weibull0 = weibreg(Surv(time, status) ~ 1,data=veteran)
   b =  exp(veteran.weibull0$coeff[1]
   a =  exp(veteran.weibull0$coeff[2])
   ```

(a) **Model choice.** Compare the above two models with the likelihood ratio test. You can extract the log-likelihood values from the output objects of function *weibreg*. Use the *pchisq* function to calculate the P value (tail probability).

(b) **Checking the Weibull assumption.** Plot the log-log transformation of the survival function (i.e. the cumulative hazard) against log-time in the veteran data. Stratify by the treatment group. Interpret.

(c) **Checking proportionality.** Plot estimates of the cumulative hazard separatately for the two groups (strata), defined by the prior therapy (variable *prior*). What can you see?

Regression analysis

1. Use veteran data for the following analysis.

```
library(survival)
?veteran # for explanations for the variables in the data set
data(veteran) # load the data
str(veteran) # show records of the data
```

(a) Analyse the data with an exponential regression model using treatment status, performance status, prior therapy status and cell type as the explanatory variables. Use the *weibreg()* function:

```
veteran.exp1 = weibreg(Surv(time,status)~ trt + karno + prior
                       + celltype, shape = 1, data = veteran)
```

(b) Repeat the above analysis using Weibull model:

```
veteran.weib1 = weibreg(Surv(time,status)~ trt + karno + prior
                       + celltype, data = veteran)
```

(c) Next perform a Cox proportional hazards analysis adjusting for treatment status, cell type, performance status, and prior therapy. Compare the results to those from the exponential and Weibull regression models. You can also extract and plot the baseline hazard from the output object using the *basehaz* function.

Use the following commands to do the analysis:

```
veteran.cox1 = coxph(Surv(time,status) ~ trt + karno + prior
                        + celltype, data=veteran);

baseline = basehaz(veteran.cox1,centered=FALSE)
plot(baseline$time,baseline$hazard,xlab='Time',
        ylab='Baseline hazard (no prior therapy)')
```

(d) Interpret the results for the regression parameters. Which category of cell type is used as a reference category?

(e) Fit Cox regression model by using prior as the stratifying variable and other variables as covariates.

(f) Try Cox regression with an interacton between treatment and prior treatment.

2. Use the melanoma data (given in the comma separated file called *melanoma.csv* available at the course website), to split the data according to the calendar period and age using *Lexis* function. Create the grouped data (5-year periods and 5-year age-groups) giving period, age, D, Y and sex. Fit a Poisson regression to these data by using additive random effects for period and age. Further, use also sex in the regression analysis.

```
# Data description of melanoma data
# sex byte sex Sex
# age byte Age at diagnosis
# stage byte stage Clinical stage at diagnosis
# mmdx byte Month of diagnosis
# yydx int Year of diagnosis
# surv_mm float Survival time in months
# surv_yy float Survival time in years
# status byte status Vital status at last contact
# subsite byte colonsub Anatomical subsite of tumour
# year8594 byte year8594 Year of diagnosis 1985-94
# agegrp byte agegrp Age in 4 categories
# dx int Date of diagnosis
# exit int Date of exit
# id float Unique ID
# Melanoma data
library(survival)
library(Epi)
library(eha)

# localised (stage=1) melanoma
melanoma = subset.data.frame(read.table("melanoma.csv",sep=",",header=T),
              stage==1)

# Death due to any cause is the event
melanoma$Status2 = melanoma$status*0
melanoma$Status2[melanoma$status==1 | melanoma$status==2]<-1

melanoma$en = cal.yr(melanoma$dx, format = "%d/%m/%Y")
melanoma$ex = cal.yr(melanoma$exit, format = "%d/%m/%Y")
melanoma$bt = cal.yr(melanoma$bdate, format = "%d/%m/%Y")
melanoma$age.ex =  ((melanoma$ex - melanoma$bt)

melanoma$dur = melanoma$ex - melanoma$en
nobs <- nrow(melanoma)

varname = setdiff(names(melanoma), c("age", "exit"))

M = melanoma[ ,varname]
M$de = melanoma$exit

# Split according to the calendar period and age
mLexis = Lexis( entry = list(per = round(cal.yr(dx, format = "%d/%m/%Y")))
          exit = list(per = round(cal.yr(de, format = "%d/%m/%Y")),
          age = round(age.ex)), exit.status = Status2, data = M )

# spliting follow-up into two time scales
split1 = splitLexis(mLexis, breaks = seq(1975, 2000,5), time.scale="per")
split = splitLexis(split1,breaks = seq(1,96,5), time.scale="age")

# getting the original timeband for each time scale
split$age.cat = timeBand(split, "age", type = "factor")
```

```
split$per.cat = timeBand(split, "per", type = "factor")


# generate attained age and attained year
final <- aggregate(cbind(split$lex.dur, split$lex.Xst),
      by = list(split$per.cat, split$age.cat), FUN = sum)
colnames(final) <- c("Period","Age","Y","D")
final

# Add code to do Poisson regression for one time scale
# and for two time scales - glm
piececonst.ap <- glm( D ~ Age + Period
               + offset( log( Y ) ), family=poisson, data = final )

summary(piececonst.ap)
anova(piececonst.ap)
```

Reference for Lexis and splitLexis functions:
Plummer, M. and Carstensen, B. (2011). Lexis: An R class for Epidemiolgical studies with long-term follow-up. Journal of Statistical Software, 38 (5); 1:12.