Homework 4 Solutions, 3 Apr

1.

The data set at `http://lib.stat.cmu.edu/datasets/csb/ch15.dat` from contains results compiled by the Cooperative Early Lung Cancer Detection Program. A full description of the data set can be found at `http://lib.stat.cmu.edu/datasets/csb/ch15.txt` . SAS code to read the data can be found at `http://lib.stat.cmu.edu/datasets/csb/ch15.sas` . Focus attention on three variables: cancer cell type (variable 5), stage (variable 6), survival time in days (variable 11), and status (variable 12). Status is coded as 0 if alive, 1 if dead from lung cancer, and 2 if dead from other causes. Consider individuals with status either 0 or 2 as censored.
a. Using the proportional hazards regression model, perform a likelihood ratio test of the hypothesis of no difference in survival among individuals with the various cancer cell types.

R commands are

```
fle<-read.table("ch15.dat",na.strings=".",
 stringsAsFactors=FALSE,
 col.names=c("number","inst","gr","det",
 "ct","st","op","a", "b","c","surv","status"))
# Five subjects have events at time zero and
# three subjects have missing ct. Delete these.
fle<-fle[!is.na(fle$ct),]
fle<-fle[fle$surv>0,]
# Count anyone censored or with cause of death other than
# lung cancer as censored.
fle$newstat<-fle$status==1
library(survival)
coxph(Surv(surv,newstat)~as.factor(ct),data=fle)
```

The important part of the output is:

```
Likelihood ratio test=100.7  on 4 df, p=< 2.2e-16
n= 1029, number of events= 682
 (3 observations deleted due to missingness)
```

Hence reject the null hypothesis of no cell type effect.

b. Test the hypothesis that the “adenocarcinoma” cell type has a hazard rate proportional to those of the other four cell types.

Here's an approach in R that is inefficient:

```
#Less efficient approach
begin<-Sys.time()
bigfle<-NULL; previous<-0
for(tt in sort(unique(fle$surv))){
 cat("tt=",tt," ")
 temp<-fle[fle$surv>=tt,]
 temp$start<-previous; previous<-tt
 temp$newstat<-(temp$surv==tt)*temp$newstat
 temp$surv<-tt
 bigfle<-rbind(bigfle,temp)
}
print(Sys.time()-begin)
```

This took 3 minutes. Do this more quickly in R via

```
#More efficient approach
begin<-Sys.time()
cnt<-0
for(tt in sort(unique(fle$surv))){
 cnt<-cnt+sum(fle$surv>=tt)
}
fle$start<-0
bigfle<-as.data.frame(array(NA,c(cnt,dim(fle)[2])))
names(bigfle)<-names(fle)
previous<-0 filled<-0
```

```
for(tt in sort(unique(fle$surv))){
 cat("tt",tt)
 use<-fle$surv>=tt
 new<-sum(use)
 temp<-fle[use,]
 temp$start<-previous; previous<-tt
 temp$newstat<-(temp$surv==tt)*temp$newstat
 temp$surv<-tt
 bigfle[filled+seq(new),]<-temp
 filled<-filled+new
}
print(Sys.time()-begin)
```

This took 34 seconds. After either, do

```
bigfle$diag1<-(bigfle$ct==1)*bigfle$surv
coxph(Surv(start,surv,newstat)~as.factor(ct)+diag1,
 data=bigfle)
```

Results are:

```
              coef exp(coef) se(coef)    z     p
as.factor(ct)1 0.43695 1.54796 0.147500 2.962 0.00305
as.factor(ct)2 0.72340 2.06143 0.120133 6.022 1.73e-09
as.factor(ct)3 1.03253 2.80816 0.110589 9.337 < 2e-16
as.factor(ct)4 0.19695 1.21768 0.343180 0.574 0.56604
diag1     -0.00026 0.99974 0.000115 -2.286 0.02224
```

Alternatively, R now has a built-in time-dependent covariate handler:

```
cat("Exhibition of tt() syntax")
coxph(Surv(surv,newstat)~as.factor(ct)+tt(ct),
 data=fle, tt=function(x,t,...) t*(x==1))
# Argument tt seems to be a a function with first argument
# referring to a data frame item, the time variable, and
three
# dots. R uses the three dot argument for arguments passed
to a
# function inside of tt. From this I conjecture that coxph
# modifies tt before evaluating it; this modification
must
# include adding additional commands using the .... If
you add
# the tt= argument, you'll get something, but it likely
won't be
# what you want.
coxph(Surv(surv,newstat)~as.factor(ct)+tt(ct),data=fle)
```

c. Estimate the ratio of hazards for stage 1 and stage 2 cancers, adjusting for cell type two ways: by adding cell types as coviariates, and stratifying. Compare the precision of your results.

Do this in R via

```
coxph(Surv(surv,newstat)~as.factor(st)+strata(ct),
 data=fle)
coxph(Surv(surv,newstat)~as.factor(st)+as.factor(ct),
 data=fle)
```

Results for stage using strata and factor for cell type are, respectively,

```
                 coef exp(coef) se(coef)      z        p
as.factor(st)2 0.8559    2.3535   0.1514  5.654 1.56e-08
as.factor(st)3 1.3834    3.9883   0.1087 12.727  < 2e-16
```

and

```
                  coef exp(coef) se(coef)      z        p
as.factor(st)2 0.87056   2.38825  0.15092  5.768 8.01e-09
as.factor(st)3 1.41513   4.11702  0.10889 12.996  < 2e-16
```

Standard errors are comparable when including cell type as a covariate rather and as a stratifier.

2.
Consider the Stanford Heart Transplant Data from `http://lib.stat.cmu.edu/datasets/stanford`. (Note that there are two versions of this data set in the file; choose the top part). Fit the Cox proportional hazards model for time to death, taking the fixed covariate prior surgery, and the time-dependent covariate transplant status.

These R commands do the job:

```
stanford<-read.table("stanford",skip=25,nrows=103,
 fill=TRUE,
 col.names=c("id","start","age","status","time",
  "prior","transp","wait","alleles","antigen",
  "score"))
# Some subjects died very shortly after receiving
# transplant. Life is measured on a scale too coarse
# to capture this. For these individuals, adjust time
# to reflect this.
change<-stanford$time==stanford$wait
change[is.na(change)]<-FALSE
stanford$time[change]<-stanford$time[change]+.1
s1<-stanford
s1$start<-0; s1$transi<-0
s1[s1$transp==1,"status"]<-0
s1[s1$transp==1,"time"]<-s1[s1$transp==1,"wait"]
s2<-stanford[stanford$transp==1,]
s2$start<-s2$wait
s2$transi<-1
library(survival)
fit<-coxph(Surv(start,time,status)~prior+transi,data=rbind(s1
```

The result is

```
       coef exp(coef) se(coef)      z     p
prior -1.05293  0.34891 0.43011 -2.448 0.0144
transi 0.08831  1.09233 0.29317 0.301 0.7632
```

Here transplant is not significance. Prior surgery is significant, and protective.

Alternatively, one could do

```
# Use tt() to generate time-dependent covariates. Make
some
# fixed-time variables to make time-dependent
covariates.
# In this case, change missing waiting times to end time
plus 1.
stanford$temp<-stanford$wait
stanford$temp[is.na(stanford$wait)]<-
stanford$time[is.na(stanford$wait)]+1
# Build a function of the variable, time, and other stuff
# (in that order):
(fit<-coxph(Surv(time,status)~prior+tt(temp),
 data=stanford, tt=function(x,t,...) (x<t)+0))
```

giving

```
        coef exp(coef) se(coef)      z     p
prior   -1.0671   0.3440  0.4298 -2.483 0.013
tt(temp) 0.1608   1.1744  0.2936 0.548 0.584
```

On the other hand, this :

```
(fit<-coxph(Surv(time,status)~prior+tt(wait),
 data=stanford, tt=function(x,t,...){out<-(x<t);
  out[is.na(out)]<-0;return(out)}))
```

gives the wrong answer:

```
      coef exp(coef) se(coef)   z   p
prior  -7.537e-01 4.706e-01 4.447e-01 -1.695 0.0901
tt(wait) 2.006e+01 5.154e+08 5.557e+03 0.004 0.9971
```

3.
The Center for Analysis and Management of Multicenter AIDS Cohort Study (2002) reported on a the health history of a cohort of individuals. A subset of these individuals were followed up for as many as 31 visits, approximately six months apart. All individuals were HIV-positive at the beginning of the study. The visit at which they were last found to be without AIDS (or in one case, the visit at which an individual was last followed) was recorded, along with an indicator of educational level. Individuals who did not report an educational level, or who missed visits during the interval in which they developed AIDS, were omitted. Furthermore, some individuals had additional screenings between scheduled screenings; results from these visits were ignored. The entries are educational level , last visit before seroconversion or AIDS, and status indicator (1 for lost to followup prior to AIDS, 4 for seroconversion, 6 for AIDS prevalence; treat 1 as censored and the others as having the event), and the count of individuals with this pattern. Treat these individuals as though the

times to AIDS have a continuous distribution with the hazards for the various educational groups proportional, and fit a model measuring the effect of education.

Here is the R code:

```
sero<-read.table('seroconvert.dat',
 col.names=c("ed","last","status","count"))
sero$event<-1
sero$event[sero$status==1]<-0
sero<-sero[rep(seq(length(sero$count)),sero$count),]
library(survival)
coxph(Surv(last,event)~as.factor(ed),data=sero)
```

to observe

```
as.factor(ed)2 -0.62628  0.53458 1.05604 -0.593 0.553
as.factor(ed)3 -0.15602  0.85554 1.00951 -0.155 0.877
as.factor(ed)4 -0.09030  0.91366 1.00329 -0.090 0.928
as.factor(ed)5 -0.20430  0.81522 1.00495 -0.203 0.839
as.factor(ed)6 -0.05358  0.94783 1.00760 -0.053 0.958
as.factor(ed)7  0.20917  1.23265 1.00667  0.208 0.835
```

This shows that hazard is highest for the highest educated group, and next highest for the lowest educated group. Otherwise, there's no particular ordering. P-vaues are all non-significant. We'll come back to this data set and treat it more appropriately as interval censored.