Parametric analysis (II part)

Data description: The "ovarian" dataset contains information on survival in a randomized trial comparing two treatments group for ovarian cancer

1. Working on the pooled sample:

```
cd "working directory path"
use ovarian.dta,clear
des
recode rx (2=0)
```

2. **Preliminary analysis** Describe data looking at frequency distributions of variables and at their cross-tabulations

```
tab rx
tab residds
tab ecogps
sum age,d
ttest age,by(rx)
tab residds rx,col row chi2
tab ecogps rx, col row chi2
tab residds ecogps, col row chi2
ttest age,by(residds)
ttest age,by(ecogps)
```

3. Declare data to be survival-data in order to take into account the follow-up

```
stset futime,fail(fustat)id(id)
```

4. By using non-parametric methods represent the total survival curve and the survival curves by variables by Kaplan-Meier estimator and test their difference

```
sts graph, xtitle("Days") ytitle("Survivor function")
sts graph,by(rx) xtitle("Days") ytitle("Survivor function")
sts test rx
sts graph,by(residds) xtitle("Days") ytitle("Survivor function")
sts test residds
sts graph,by(ecogps) xtitle("Days") ytitle("Survivor function")
sts test ecogps
egen age_cat=cut(age),at(0,50(10)60,100)
sts graph,by(age_cat) xtitle("Days") ytitle("Survivor function")
sts test age_cat
```

5. Before fitting the regression models, check for the assumptions underlying the exponential, weibull, log-logistic and log-normal models respectively by graphical methods

```
*survival curve
sts gen km=s,by(rx)
gen logkm_exp=log(km)
twoway (scatter logkm_exp _t if rx==0,sort)(scatter logkm_exp _t if rx==1,sort) ///
(lfit logkm_exp _t if rx==0,sort) (lfit logkm_exp _t if rx==1,sort), ///
legend(lab(1 "Untreated") lab(2 "Treated")) ytitle("log(S(t)") xtitle("t")
```

```
*cumulative hazard
sts gen cumhaz_exp=na,by(rx)
gen cumhaz_exp1=cumhaz_exp
gen cumhaz_exp0=cumhaz_exp
gen logcumhaz_exp=log(cumhaz_exp)
gen logtime=log(_t)
twoway (scatter logcumhaz_exp logtime if rx==0,connect(J)sort) ///
(scatter logcumhaz_exp logtime if rx==1,connect(J)sort) ///
(lfit logcumhaz_exp logtime if rx==0) (lfit logcumhaz_exp logtime if rx==1), ///
legend(lab(1 "Untreated") lab(2 "Treated")) ytitle("log(H(t)") xtitle("log(t)")

*observed vs predicted
xi:streg rx,dist(e) nohr
predict surv_exp,surv
twoway (line km _t if rx==0,sort)(line km _t if rx==1,sort) ///
(scatter surv_exp _t if rx==0,sort) (scatter surv_exp _t if rx==1,sort), ///
legend(lab(1 "Untreated observed") lab(2 "Treated observed") lab(3 "Untreated predicted") //
lab(4 "Treated predicted")) ytitle("log(S(t)") xtitle("t")

*alternatively
gen s1_exp=exp(-exp(_b[_cons]+_b[rx])*_t)
gen s0_exp=exp(-exp(_b[_cons])*_t)
sort _t
twoway (scatter s1_exp _t) (scatter s0_exp _t)
stcurve, surv at(rx=0) at1(rx=1)
twoway (line km _t if rx==0,sort)(line km _t if rx==1,sort) ///
(scatter s0_exp _t,sort) (scatter s1_exp _t,sort), ///
legend(lab(1 "Untreated observed") lab(2 "Treated observed") lab(3 "Untreated predicted") //
lab(4 "Treated predicted")) ytitle("log(S(t)") xtitle("t")

*weibull model (proportional hazard)
gen logkm_w=log(-log(km))
gen logt=log(_t)
twoway (line logkm_w logt if rx==0,sort)(line logkm_w logt if rx==1,sort) ///
(lfit logkm_w logt if rx==0,sort) (lfit logkm_w logt if rx==1,sort), ///
legend(lab(1 "Untreated") lab(2 "Treated")) ytitle("log(-log(S(t))") xtitle("log(t)")

*observed vs predicted
xi:streg rx,dist(w) nohr
predict surv_w,surv
twoway (line km _t if rx==0,sort)(line km _t if rx==1,sort) ///
(scatter surv_w _t if rx==0,sort) (scatter surv_w _t if rx==1,sort), ///
legend(lab(1 "Untreated observed") lab(2 "Treated observed") ///
lab(3 "Untreated predicted") lab(4 "Treated predicted"))

*alternatively
gen s_w0=exp(-exp(_b[_cons])*_t^1.129104)
gen s_w1=exp(-exp(_b[_cons]+_b[rx])*_t^1.129104)
twoway (scatter s_w0 _t) (scatter s_w1 _t)
```

```
stcurve, surv at(rx=0) at1(rx=1)
twoway (line km _t if rx==0,sort)(line km _t if rx==1,sort) ///
(scatter s_w0 _t,sort) (scatter s_w1 _t,sort), ///
legend(lab(1 "Untreated observed") lab(2 "Treated observed") ///
lab(3 "Untreated predicted") lab(4 "Treated predicted"))

*log-logistic model
gen odds_km=(1-km)/km
gen logodds=log(odds_km)
*gen logt=log(_t)
twoway (line logodds logt if rx==0,sort)(line logodds logt if rx==1,sort) ///
(lfit logodds logt if rx==0,sort) (lfit logodds logt if rx==1,sort), ///
legend(lab(1 "Untreated") lab(2 "Treated")) ytitle("log(odds)") xtitle("log t")

*observed vs predicted
xi:streg rx,dist(loglog)
predict surv_loglog,surv
twoway (line km _t if rx==0,sort)(line km _t if rx==1,sort) ///
(scatter surv_loglog _t if rx==0,sort) (scatter surv_loglog _t if rx==1,sort), ///
legend(lab(1 "Untreated observed") lab(2 "Treated observed") lab(3 "Untreated predicted") //
lab(4 "Treated predicted"))

*log-normal model
gen inv_km=invnormal(1-km)
*gen logt=log(_t)
twoway (line inv_km logt if rx==0,sort)(line inv_km logt if rx==1,sort) ///
(lfit inv_km logt if rx==0,sort) (lfit inv_km logt if rx==1,sort), ///
legend(lab(1 "Untreated") lab(2 "Treated")) ytitle("log(odds)") xtitle("log t")

*observed vs predicted
xi:streg rx,dist(ln)
predict surv_ln,surv
twoway (line km _t if rx==0,sort)(line km _t if rx==1,sort) ///
(scatter surv_ln _t if rx==0,sort) (scatter surv_ln _t if rx==1,sort), ///
legend(lab(1 "Untreated observed") lab(2 "Treated observed") lab(3 "Untreated predicted") //
lab(4 "Treated predicted"))

drop   km logkm_exp cumhaz_exp cumhaz_exp1 cumhaz_exp0 logcumhaz_exp ///
logtime s1_exp s0_exp logkm_w logt s_w0 s_w1 odds_km odds_km surv_loglog inv_km surv_ln
```

6. By using the exponential model evaluate the treatment effect (crude and adjusted estimates) on mortality

```
xi:streg i.rx,dist(e)
xi:streg i.rx age,dist(e)
xi:streg i.rx i.residds age  i.ecogps,dist(e)
xi:streg i.rx*i.ecogps i.residds age,dist(e)
xi:streg i.rx*i.residds age  i.ecogps,dist(e)
xi:streg i.rx i.residds age  i.ecogps,dist(e)
```

```
est store a
xi:streg i.rx i.residds i.age_cat  i.ecogps,dist(e)
est store b
lrtest a b
xi:streg i.rx i.residds age  i.ecogps,dist(e)
```

7. Calculate Cox-Snell residuals for the exponential model (post-estimation command)

```
xi:streg i.rx i.residds age  i.ecogps,dist(e)
predict residual, csnell
stset residual, failure(fustat)id(id)
sts gen km = s
generate cumhaz = -ln(km)
line cumhaz residual residual, sort name("exponential, replace") ///
legend(off) t1(Exponential) xtitle("")
drop cumhaz residual k
```

8. Calculate standardized residuals for the exponential model (post-estimation command)

```
stset futime,fail(fustat)id(id)
xi:streg i.rx i.residds age  i.ecogps,dist(e)
predict pred, xb
gen res_stand=(log(_t)-pred)
replace res_stand=exp(res_stand)
stset res_stand, failure(fustat)id(id)
sts gen km = s
gen logkm=log(km)
line logkm _t,sort legend(off) t1(Exponential) xtitle("t") ytitle("log(S(t))")
drop km logkm res_stand pred
```

9. Calculate deviance residuals for the exponential model (post-estimation command)

```
stset futime,fail(fustat)id(id)
xi:streg i.rx i.residds age  i.ecogps,dist(e)
predict deviance, deviance
graph twoway scatter deviance id, legend(off) t1(Exponential) xtitle("id")
drop deviance
```

10. After fitting the Weibull model, calculate Cox-Snell residuals (post-estimation command)

```
stset futime,fail(fustat)id(id)
xi:streg i.rx i.residds age  i.ecogps,dist(w)
predict residual, csnell
stset residual, failure(fustat)id(id)
sts gen km = s
generate cumhaz = -ln(km)
line cumhaz residual residual, sort name("Weibull, replace") ///
legend(off) t1(Weibull) xtitle("")
drop cumhaz residual k
```

11. Calculate standardized residuals for the weibull model (post-estimation)

```
stset futime,fail(fustat)id(id)
xi:streg i.rx i.residds age  i.ecogps,dist(w)
predict pred, xb
gen sigma= 0.5195874
gen res_stand=(log(_t)-pred)/sigma
replace res_stand=exp(res_stand)
stset res_stand, failure(fustat)id(id)
sts gen km = s
gen tras_km=log(-log(km))
gen log_time=log(_t)
line tras_km log_time ,sort legend(off) t1(Weibull) xtitle("log(t)") ///
ytitle("log(-log(S(t)))")
drop km tras_km res_stand sigma pred log_time
```

12. Calculate deviance residuals for the weibull model (post-estimation)

```
stset futime,fail(fustat)id(id)
xi:streg i.rx i.residds age  i.ecogps,dist(w)
predict deviance, deviance
graph twoway scatter deviance id, legend(off) t1(Weibull) xtitle("id")
drop deviance
```

13. After fitting the log-logistic model, calculate Cox-Snell residuals (post-estimation)

```
stset futime,fail(fustat)id(id)
xi:streg i.rx i.residds age  i.ecogps,dist(loglog)
predict residual, csnell
stset residual, failure(fustat)id(id)
sts gen km = s
generate cumhaz = -ln(km)
line cumhaz residual residual, sort name("loglogistica, replace") ///
legend(off) t1(log-logistica) xtitle("")
drop cumhaz residual km
```

14. Calculate standardized residuals for the log-logistic model (post-estimation)

```
stset futime,fail(fustat)id(id)
xi:streg i.rx i.residds age  i.ecogps,dist(loglog)
predict pred, xb
gen sigma=.3727281
gen res_stand=(log(_t)-pred)/sigma
replace res_stand=exp(res_stand)
stset res_stand, failure(fustat)id(id)
sts gen km = s
gen tras_km=log((1-km)/km)
gen log_time=log(_t)
line tras_km log_time,sort legend(off) t1(Log-logistica) xtitle("log(t)") ///
ytitle("log(1-S(t)/S(t))")
drop km tras_km res_stand sigma pred log_time
```

15. Calculate deviance residuals for the log-logistic model (post-estimation)

```
stset futime,fail(fustat)id(id)
xi:streg i.rx i.residds age  i.ecogps,dist(loglog)
predict deviance, deviance
graph twoway scatter deviance id, legend(off) t1(Log-logistica) xtitle("id")
drop deviance
```

16. After fitting the log-normal model, calculate Cox-Snell residuals (post-estimation)

```
stset futime,fail(fustat)id(id)
xi:streg i.rx i.residds age  i.ecogps,dist(ln)
predict residual, csnell
stset residual, fail(fustat)id(id)
sts gen km = s
generate cumhaz = -ln(km)
line cumhaz residual residual, sort name("lognormale, replace") ///
legend(off) t1(log-normal) xtitle("")
drop cumhaz residual km
```

17. Calculate standardized residuals for the log-normal model (post-estimation)

```
stset futime,fail(fustat)id(id)
xi:streg i.rx i.residds age  i.ecogps,dist(ln)
predict pred, xb
gen sigma=.6319339
gen res_stand=(log(_t)-pred)/sigma
replace res_stand=exp(res_stand)
stset res_stand, fail(fustat)id(id)
sts gen km = s
gen tras_km=invnormal(1-km)
gen log_time=log(_t)
line tras_km log_time,sort legend(off) t1(Log-normal) xtitle("log(t)") ///
ytitle("")
drop km tras_km res_stand sigma pred log_time
```

18. Calculate deviance residuals for the log-normal model (post-estimation)

```
stset futime,fail(fustat)id(id)
xi:streg i.rx i.residds age  i.ecogps,dist(ln)
predict deviance, deviance
graph twoway scatter deviance id, legend(off) t1(Log-normal) xtitle("id")
drop deviance
```

19. Compare the models by AIC (Akaike Criterion Information=2k-2ln(L))

```
stset futime,fail(fustat)id(id)
xi:streg i.rx i.residds age  i.ecogps,dist(e)
estat ic
xi:streg i.rx i.residds age  i.ecogps,dist(w)
estat ic
xi:streg i.rx i.residds age  i.ecogps,dist(loglog)
estat ic
xi:streg i.rx i.residds age  i.ecogps,dist(ln)
estat ic
```

20. After selecting the best model fitting data, interpret the estimates obtained by that model

```
xi:streg i.rx i.residds age  i.ecogps,dist(ln)
xi:streg i.rx i.residds age  i.ecogps,dist(loglog)
```