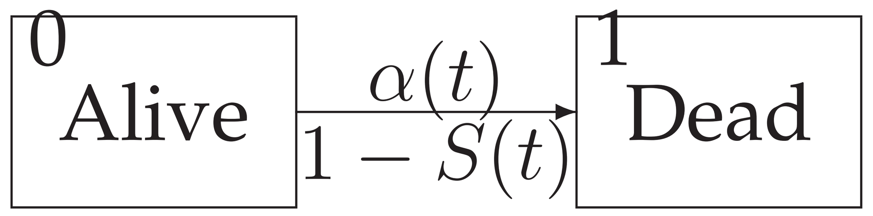
SURVIVAL CHEAT SHEET

http://staff.pubhealth.ku.dk/~ts/survival/survival-cheat.r

SURVIVAL ANALYSIS:

- Survival Outcomes :



- $\alpha(t)$ hazard, instantaneous risk
- $A(t) = \int_0^t \alpha(s)ds$ cumulative hazard
- $S(t) = P(T > t)$ survival probability
- $P(T > t) = \exp(-A(t))$ survival probability

Vocabulary:

- Survival Models: I

R survival	library(survival); library(mets); data(TRACE)
Define Survival outcome	Surv(time,status), Surv(entry,time,status) status is 1 for events and 0 otherwise
	TRACE=transform(TRACE,status9=(status==9),dead=(status!=0))
Kaplan-Meier	$\exp(-\Lambda(t))$
	ss=survfit(Surv(time,dead) ~ vf,data=TRACE); plot(ss); summary(ss); library(timereg); kmplot(ss);
Kaplan-Meier	library(prodlim); ss=prodlim(Surv(time,dead) ~ vf,data=TRACE) plot(ss); summary(ss); summary(ss,time=c(1,5))
Log-rank Stratified LR	survdiff(Surv(time,dead) ~ vf,data=TRACE) coxph(Surv(time,dead) ~ strata(vf)+chf ,data=TRACE)

Nelson-Aalen	library(mets); cc=phreg(Surv(time,dead) ~ strata(vf),data=TRACE); plot(cc);
	ss=survfit(Surv(time,dead) ~ vf,data=TRACE);
	plot(ss,fun="cumhaz"); kmplot(ss,fun="cumhaz");

- Cox predictions: II

Cox regression predictions	out=coxph(Surv(time,dead) ~ vf,data=TRACE) newdata=data.frame(vf=c(0,1)); pred=survfit(out,newdata); plot(pred);

- Cox Goodness of fit: III

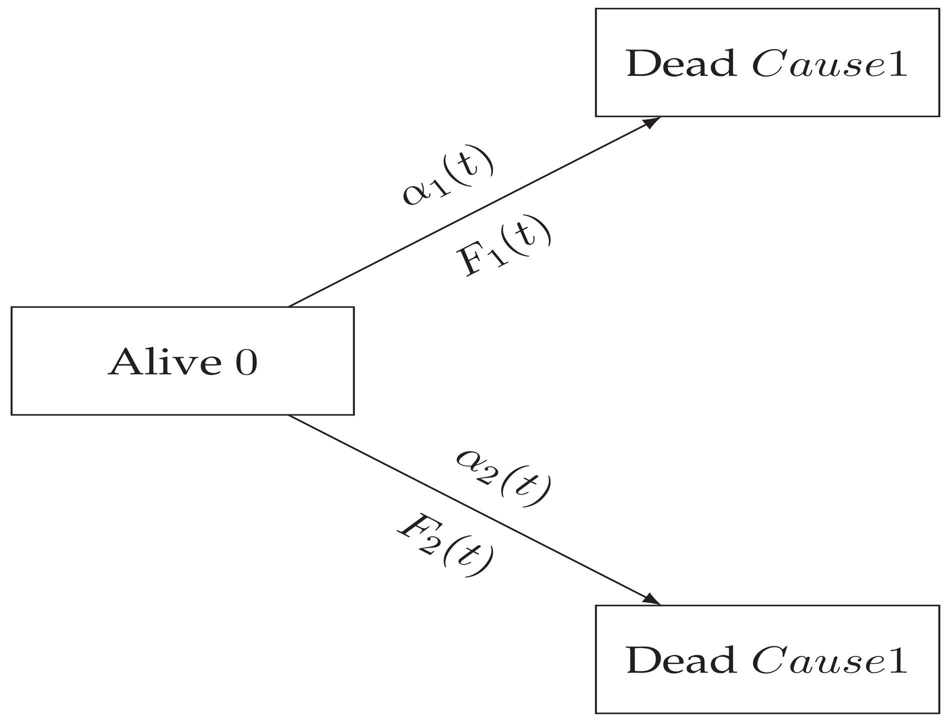
proportionality	library(mets); cc=phreg(Surv(time,dead) ~ chf+vf,data=TRACE); gg=gof(cc); summary(gg); plot(gg);
graphical phreg cloglog-base	cox1=phreg(Surv(time,dead)~ strata(vf)+chf,data=TRACE) plot(cox1); plot(cox1,log="y")
functional form	ca=gofZ.phreg(Surv(time,dead) ~ vf+wmi+age,data=TRACE) summary(ca); plot(ca,type="z")
spline fitting	mf=coxph(Surv(time,dead) ~ vf+pspline(wmi,df=4),data=TRACE) termplot(mf,term=2,se=TRUE)
Interactions	ifit=coxph(Surv(time,dead) ~ vf+chf+wmi+vf*chf,data=TRACE) afit=coxph(Surv(time,dead)~ vf+chf+wmi,data=TRACE) anova(ifit,afit)

- Wald-test, linear combinations: IV

Cox-Wald test	cc=coxph(Surv(time,status==0) ~ vf+chf+vf*chf,data=TRACE) library(mets); estimate(cc,rbind(c(1,1,1))); estimate(cc,as.list(3)) library(timereg); wald.test(cc,contrast=c(1,1,0)); wald.test(cc,coef.null=3)

COMPETING RISKS

- Competing Risks



- $\alpha_1(t)$ cause specific hazard
- $\alpha_2(t)$ cause specific hazard
- $A_1(t) = \int_0^t \alpha_1(s)ds$ cumulative hazard
- $S(t) = P(T > t) = \exp(-A_1(t) - A_2(t))$ survival probability
- $F_1(t) = P(T < t, \epsilon = 1)$ cumulative incidence cause 1
- $F_2(t) = P(T < t, \epsilon = 2)$ cumulative incidence cause 2

Vocabulary:

- Cumulative incidence: V

Define outcome	Surv(time,status), Surv(entry,time,status) library(prodlim); Hist(time,status), Hist(entry=entry,time,status) status is 1,2,.. for events and 0 for censorings
CIF probability	library(prodlim); cifa=prodlim(Hist(time,status) ~ vf,data=TRACE) plot(cifa,cause=9); summary(cifa,cause=9,times=0:8);
Grays Test	library(cmprsk); cifa=with(TRACE,cuminc(time,status,vf)); print(cifa); plot(cifa)

- Cumulative incidence Regression: VI

Regression CIF predictions	library(cmprsk); cifa=with(TRACE,crr(time,status,vf,failcode=9)); summary(cifa); pred=predict(cifa,rbind(c(1,0),c(1,1))); plot(pred);
OR, logit-link predictions FG-link	library(mets); cifa=cifreg(Event(time,status) ~ vf+chf,data=TRACE,cause=9) nd=data.frame(vf=1,chf=c(1,0)); pcifa=predict(cifa,newdata=nd); plot(pcifa) cifa=cifreg(Event(time,status) ~ +vf+chf,data=TRACE,cause=9,propodds=NULL) summary(cifa); pcifa=predict(cifa,newdata=nd); plot(pcifa)
OR, logit-link GOF+Predictions	library(timereg); cifa=prop.odds.subdist(Event(time,status) ~ +vf+chf,data=TRACE,cause=9) summary(cifa); pcifa=predict(cifa,Z=c(1,0)); plot(pcifa)

- Goodness of fit for Cumulative incidence Regression: VII

GOF: FG status=1	library(crskdiag); data(tTRACE); tTRACE=dtransform(tTRACE,status=1,status==9) out=diag_crr(Crsk(time,status) ~ +vf, data=tTRACE,test="prop"); print(out);
GOF: Censoring	out = coxph(Surv(time,status==0) ~ +vf,data=TRACE)

- Binomial Regression Modelling: VIII

OR, logit-link	library(mets); cifat=binreg(Event(time,status) ~ vf+chf,data=TRACE,cause=9,time=5)

- Cause Specific Hazard: IX

Cox for cause Log-rank	data(bmt); out8=coxph(Surv(time,cause==1) ~ tcell,data=bmt) survdiff(Surv(time,cause==1) ~ tcell,data=bmt)
Regression Cause specific Cox's plot via pec	library(riskRegression); data(bmt); nd=data.frame(tcell=c(0,1)) fit=CSC(Hist(time,cause) ~ tcell,data=bmt); tt=times=seq(0,100,by=1) p1=predictRisk(fit,newdata=nd,cause=1,times=tt); matplot(tt,t(p1),type="s") library(pec); plotPredictEventProb(fit,newdata=nd,cause=1,col=1:2)

Survival Cheat Sheet

Survival Analysis:

- Plotting Cox and Kaplan-Meier: X

prodlim +coxph	km1=prodlim(Surv(time,dead)∼ vf ,data=TRACE) cox1=coxph(Surv(time,dead)∼ vf,data=TRACE); newdata=data.frame(vf=c(0,1)); pred=survfit(cox1,newdata); plot(pred); plot(km1,add=TRUE)
survfit+phreg	km1=survfit(Surv(time,dead)∼ vf ,data=TRACE) cox1=phreg(Surv(time,dead)∼ vf,data=TRACE); pred=predict(cox1,newdata,se=0); plotkm(km1); plot(cox1,add=TRUE)

- Splitting time Cox: XI

coxph	cox2=coxph(Surv(time,dead)∼ vf+chf+tt(vf),data=TRACE, tt=function(x,t,...) x*(t >1))
Survsplit	fdat=survSplit(Surv(time,dead)∼ .,data=TRACE,cut=c(1),episode="timeg") fdat=transform(fdat,vflate=vf*(timeg==2)); cox3=coxph(Surv(tstart,time,dead) ∼ vf+chf+vflate,data=fdat)
event-split	TRACE=transform(TRACE,sv=time*0.5); dat=event.split(data=TRACE,cuts="sv")

- Cox-Splines: XII

Simple spline	library(mets); data(TRACE) TRACE=dspline(TRACE,∼ wmi,breaks=c(1,1.3,1.7)) coxs=coxph(Surv(time,status==9) ∼ age+wmi+vf+chf+wmi.spline1+ wmi.spline2+wmi.spline3,data=TRACE) nd=data.frame(age=50,vf=0,chf=0,wmi=seq(0.4,3,by=0.01)) nd=dspline(nd,∼ wmi,breaks=c(1,1.3,1.7)); ps=predict(coxs,newdata=nd); with(nd,plot(wmi,ps,type="l"))

- Multivariate Two-Stage Frailty: XIII

MLE	library(mets); data(diabetes); margph=phreg(Surv(time,status)∼ treat+cluster(id),data=diabetes) fitco1=twostageMLE(margph,data=diabetes); summary(fitco1)
Pairwise	fit1=twostage(margph,data=diabetes); summary(fit1) theta.des=model.matrix(∼ -1+factor(adult),data=diabetes) fit2=twostage(margph,data=diabetes,theta.des=theta.des); summary(fit2)

Recurrent Events :

- Marginal Mean : XIV

Marginal mean	library(mets); data(base1cumhaz); data(drcumhaz); rr=simRecurrent(1000,base1cumhaz,death.cumhaz=drcumhaz) xr=phreg(Surv(start,stop,status) ∼ cluster(id),data=rr) dr=phreg(Surv(start,stop,death) ∼ cluster(id),data=rr) out=recurrentMarginal(xr,dr); bplot(out,se=TRUE,ylab="mean")

Simualting Event History Data

- Competing Risks Regression