basic equations for survival analysis and cox-regresion

More details can be found in my notebook https://zjuwhw.github.io/notebook-surivialR/

- Survival function, the probability of surviving up to a point t: $S(t) = pr(T > t), 0 < t < \infty$
- Hazard function is the instantaneous failure rate: $h(t) = \lim_{\delta \to \infty} \frac{pr(t<T<t+\delta|T>t)}{\delta}$
- Cumulative distribution function (CDF): $F(t) = pr(T \leq t) = 1 - S(t), 0 < t < \infty$
- Probability density function (PDF):
  - $f(t) = \frac{d}{dt}F(t) = \frac{d}{dt}[1 - S(t)] = -\frac{d}{dt}S(t)$
  - $f(t) = h(t)S(t)$
- cumulative hazard function, the area under the hazard function up to time t:
  - $H(t) = \int_0^t h(u)du$
  - $H(t) = -log[S(t)]$

Cox-regression

$$h_1(t) = \psi h_0(t); \psi = e^{z\beta}$$

- β: the effect size for the covariate
- $h_0(t)$: baseline hazard function; "baseline" means z=0
- z is the covariate
- $z\beta$ is called "linear predictor" (log-hazard score)
- $\psi = e^{z\beta}$ is hazard ratio
- baseline cumulative hazard function: $H_0(t) = \int_0^t h_0(t)dt$
- baseline survival function: $S_0(t) = exp[-H_0(t)]$
- survival function for a particular individual with covariate value z: $S(t|z) = [S_0(t)]^{exp(z\hat{\beta})}$
- absolute risk at time t with covariate value z: $1 - S(t|z)$

R code

```
### simple example with a dummy variable
library(survival)
```

toy example data

```
## Warning: package 'survival' was built under R version 4.4.1
```

```
tt = c(6, 7, 10, 15, 19, 25)
delta = c(1, 0, 1, 1, 0, 1)  ### 0 means censored; 1 means observed
summary(survfit(Surv(tt,delta)~1))
```

```
## Call: survfit(formula = Surv(tt, delta) ~ 1)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     6      6       1    0.833   0.152        0.583            1
##    10      4       1    0.625   0.213        0.320            1
##    15      3       1    0.417   0.222        0.147            1
##    25      1       1    0.000     NaN           NA           NA
```

```r
trt = c(0, 0, 1, 0, 1, 1)  ## group 0 or 1
#### cox regression
result.cox = coxph(Surv(tt, delta)~trt, x=T, y=T)
summary(result.cox) ## beta is -1.3261
```

binary variable: baseline hazard function will be defined on the variable is 0

```
## Call:
## coxph(formula = Surv(tt, delta) ~ trt, x = T, y = T)
##
##   n= 6, number of events= 4
##
##        coef exp(coef) se(coef)     z Pr(>|z|)
## trt -1.3261    0.2655   1.2509 -1.06    0.289
##
##     exp(coef) exp(-coef) lower .95 upper .95
## trt    0.2655      3.766   0.02287     3.082
##
## Concordance= 0.7  (se = 0.116 )
## Likelihood ratio test= 1.21  on 1 df,   p=0.3
## Wald test            = 1.12  on 1 df,   p=0.3
## Score (logrank) test = 1.27  on 1 df,   p=0.3
```

```r
### no mean correction for binary variable
result.cox$means
```

```
## trt
##   0
```

```r
#### linear predictor: beta*z
result.cox$linear.predictor
```

```
## [1]  0.000000  0.000000 -1.326129  0.000000 -1.326129 -1.326129
```

```r
result.cox$coefficients*c(0,1)
```

```
## [1]  0.000000 -1.326129
```

```r
predict(result.cox, type="lp")
```

```
## [1]  0.000000  0.000000 -1.326129  0.000000 -1.326129 -1.326129
```

```r
predict(result.cox, newdata=data.frame(trt=c(0,1)), type="lp")  ## use the predict function with type="
```

```
##         1         2
##  0.000000 -1.326129
```

```r
#### risk in the predict() function: exp(beta*z)
predict(result.cox, newdata=data.frame(trt=c(0,1)), type="risk")
```

```
##        1        2
## 1.000000 0.265503
```

```r
exp(result.cox$linear.predictor)
```

```
## [1] 1.000000 1.000000 0.265503 1.000000 0.265503 0.265503
```

```r
### hazard ratio
exp(result.cox$coefficients)  ### HR<1 means decrease the risk
```

```
##      trt
## 0.265503
```

```
### baseline cumulative hazard function
#Be sure to use the option "centered = F" to cause it to estimate the cumulative hazard at
#beta= 0. The default is to estimate it at the mean of the covariates. This will often not
#make sense, particularly for categorical covariates such as treatment indicator, sex, or race
basehaz(result.cox, centered=F)
```

```
##      hazard time
## 1 0.2633999    6
## 2 0.2633999    7
## 3 0.8200350   10
## 4 1.4732003   15
## 5 1.4732003   19
## 6 5.2396357   25
```

```
basehaz(result.cox, centered=T)
```

```
##      hazard time
## 1 0.2633999    6
## 2 0.2633999    7
## 3 0.8200350   10
## 4 1.4732003   15
## 5 1.4732003   19
## 6 5.2396357   25
```

```
### no difference between centered=T and F return results in basehaz

### use the function survfit
base_surv = survfit(result.cox)
summary(base_surv)
```

```
## Call: survfit(formula = result.cox)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     6      6       1   0.7684  0.2093     4.51e-01            1
##    10      4       1   0.4404  0.3182     1.07e-01            1
##    15      3       1   0.2292  0.2555     2.58e-02            1
##    25      1       1   0.0053  0.0351     1.22e-08            1
```

```
base_hazard <- data.frame(
  time = base_surv$time,
  cumhaz = base_surv$cumhaz,
  surv = base_surv$surv
)
base_hazard
```

```
##   time    cumhaz        surv
## 1    6 0.2633999 0.768434561
## 2    7 0.2633999 0.768434561
## 3   10 0.8200350 0.440416253
## 4   15 1.4732003 0.229190841
## 5   19 1.4732003 0.229190841
## 6   25 5.2396357 0.005302188
```

```
### absolute risk at a time point: 1-S(t|z)
library(pec)
```

```
## Loading required package: prodlim
```

```
1-pec::predictSurvProb(result.cox, newdata=data.frame(trt=c(0,1)), time=25)
```

```
##           [,1]
## [1,] 0.9946978
## [2,] 0.7512083
```

```
1 - base_hazard$surv[base_hazard$time==25]^(predict(result.cox, newdata=data.frame(trt=c(0,1)), type="r
```

```
##         1         2
## 0.9946978 0.7512083
```

```
age = c(50,30,50,34,65,90)
result.cox2 = coxph(Surv(tt, delta)~age,x=T, y=T)
summary(result.cox2)   ## coef: -0.03785
```

continuous variable: baseline hazard function will be defined on the variable is at mean value

```
## Call:
## coxph(formula = Surv(tt, delta) ~ age, x = T, y = T)
##
##   n= 6, number of events= 4
##
##         coef exp(coef) se(coef)      z Pr(>|z|)
## age -0.03785   0.96286  0.03550 -1.066    0.286
##
##     exp(coef) exp(-coef) lower .95 upper .95
## age    0.9629      1.039    0.8981     1.032
##
## Concordance= 0.65  (se = 0.189 )
## Likelihood ratio test= 1.45  on 1 df,   p=0.2
## Wald test            = 1.14  on 1 df,   p=0.3
## Score (logrank) test = 1.3  on 1 df,   p=0.3
```

```
### mean correction
result.cox2$means
```

```
##      age
## 53.16667
```

```
mean(age)
```

```
## [1] 53.16667
```

```
### linear predictor
result.cox2$linear.predictor
```

```
## [1]  0.1198498  0.8767962  0.1198498  0.7254069 -0.4478599 -1.3940429
```

```
result.cox2$coefficients*(age-result.cox2$means)
```

```
## [1]  0.1198498  0.8767962  0.1198498  0.7254069 -0.4478599 -1.3940429
```

```
predict(result.cox2, type="lp")
```

```
## [1]  0.1198498  0.8767962  0.1198498  0.7254069 -0.4478599 -1.3940429
```

```r
predict(result.cox2, newdata = data.frame(age-mean(age)), type="lp")
```

```
##          1          2          3          4          5          6
##  0.1198498  0.8767962  0.1198498  0.7254069 -0.4478599 -1.3940429
```

```r
#### risk in the predict() function: exp(beta*z)
predict(result.cox2, type="risk")
```

```
## [1] 1.1273276 2.4031880 1.1273276 2.0655715 0.6389942 0.2480704
```

```r
exp(predict(result.cox2, type="lp"))
```

```
## [1] 1.1273276 2.4031880 1.1273276 2.0655715 0.6389942 0.2480704
```

```r
predict(result.cox2, newdata = data.frame(age-mean(age)), type="lp")
```

```
##          1          2          3          4          5          6
##  0.1198498  0.8767962  0.1198498  0.7254069 -0.4478599 -1.3940429
```

```r
### hazard ratio
exp(result.cox2$coefficients)
```

```
##       age
## 0.9628599
```

```r
### baseline cumulative hazard function
basehaz(result.cox2, centered=F) # age at 0
```

```
##       hazard time
## 1  0.9828386    6
## 2  0.9828386    7
## 3  2.8161570   10
## 4  5.3494434   15
## 5  5.3494434   19
## 6 35.5016657   25
```

```r
basehaz(result.cox2, centered=T) # age at the mean
```

```
##      hazard time
## 1 0.1313978    6
## 2 0.1313978    7
## 3 0.3764980   10
## 4 0.7151784   15
## 5 0.7151784   19
## 6 4.7462929   25
```

```r
basehaz(result.cox2, centered=F)$hazard * exp(mean(age)*result.cox2$coefficients)
```

```
## [1] 0.1313978 0.1313978 0.3764980 0.7151784 0.7151784 4.7462929
```

```r
### use the function survfit: same withe basehaz() with centered=T, default
base_surv2 = survfit(result.cox2)
summary(base_surv2)
```

```
## Call: survfit(formula = result.cox2)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     6      6       1  0.87687  0.1241     6.64e-01            1
##    10      4       1  0.68626  0.2050     3.82e-01            1
##    15      3       1  0.48910  0.2360     1.90e-01            1
```

```
##    25      1       1  0.00868  0.0564     2.59e-08            1
```

```
base_hazard2 <- data.frame(
  time = base_surv2$time,
  cumhaz = base_surv2$cumhaz,
  surv = base_surv2$surv
)
base_hazard2
```

```
##   time    cumhaz        surv
## 1    6 0.1313978 0.876868912
## 2    7 0.1313978 0.876868912
## 3   10 0.3764980 0.686260491
## 4   15 0.7151784 0.489104839
## 5   19 0.7151784 0.489104839
## 6   25 4.7462929 0.008683827
```

```
### absolute risk at a time point: 1-S(t|z)
library(pec)

1-pec::predictSurvProb(result.cox2, newdata=data.frame(age=mean(age)), time=6)
```

```
##           [,1]
## [1,] 0.1231311
```

```
1 - base_hazard2$surv[base_hazard2$time==6]^(predict(result.cox2, newdata=data.frame(age=mean(age)), ty
```

```
##         1
## 0.1231311
```

```
1-pec::predictSurvProb(result.cox2, newdata=data.frame(age=50), time=6)
```

```
##           [,1]
## [1,] 0.1376796
```

```
1 - base_hazard2$surv[base_hazard2$time==6]^(predict(result.cox2, newdata=data.frame(age=50), type="risk
```

```
##         1
## 0.1376796
```