CHL5209 HW3
Due March 22

Question 1

(a) Specify the fitted models 1 and 2. Which model would you prefer and why? Why did they give different results?

- Model 1 where `transplant` is treated as fixed value can be specified as:

$$\lambda_i(t) = \lambda_0(t)\exp(\beta \times \text{transplant}_i)$$

Based on the model output, the fitted model 1 can be written as:

$$\lambda_i(t) = \lambda_0(t)\exp(-0.7447 \times \text{transplant}_i)$$

- Model 2 treated `transplant` as a time-dependent variable, and the cox model for time-dependent variable is specified as:

$$\lambda_i(t) = \lambda_0(t)\exp(\beta \times 1_{\{v_i<t\}})$$

where v_i is the time that patient i received transplant. Based on the model output, the fitted model 2 can be written as:

$$\lambda_i(t) = \lambda_0(t)\exp(0.3246 \times 1_{\{v_i<t\}})$$

- I would prefer Model 2 (cox model with time-dependent variable) because:
  - The state of `transplant` variable changes over time (i.e., `transplant` is a time-dependent variable)
  - Model 1 treats patients as either received transplant or did not receive transplant. However, as the state of `transplant` variable changes over time, we can not classify patients into a specific group (received or did not receive transplant). Patients who received transplant after some wait time contribute to both transplant and not transplant, if we classify these patients into the transplant group, we would introduce immortal time bias, as patients who received transplant need to be alive to receive transplant. So model 1 is not appropriate.
  - Model 2 treats `transplant` as a time-dependent variable, and $\lambda_i(t)$ is modeled using the state of the `transplant` at time t. The follow up time before the transplantation happens for those who received transplant would contribute to the never receive transplant group.

(b)Create a long format dataset that could be used to fit model 2. Show your dataset. Note that two patients (IDs 3 and 6) received the transplant on the same day as they entered the study; don't modify the times, instead you can remove the rows with zero follow-up duration from your dataset. Verify the above results by fitting model 2 to your long format dataset using `coxph`.

Below is the dataset in long format.(code can be found in the appendix.)

```
> trans_dat_long
   id sex time1 time2 transplant status
1   1   1     0     3          0      1
2   2   1     0     5          0      0
3   3   1     0     5          1      1
4   4   1     0     6          0      1
5   5   1     0     6          0      0
6   5   1     6     8          1      0
7   6   0     0     4          1      0
8   7   0     0     5          0      0
9   7   0     5     7          1      1
10  8   0     0     8          0      0
11  9   0     0     5          0      0
12  9   0     5     9          1      1
13 10   0     0     3          0      0
14 10   0     3    10          1      0
```

cox model was fitted using this long format data. The results are the same as the `coxph` function with `tt` argument. Results are shown as follows:

```
> library(survival)
> # fit cox model with time-dependent variable
> Q1_model <- coxph(Surv(time1,time2,status)~transplant,
+                   data=trans_dat_long)
> summary(Q1_model)
Call:
coxph(formula = Surv(time1, time2, status) ~ transplant, data = trans_dat_long)

  n= 14, number of events= 5

             coef exp(coef) se(coef)     z Pr(>|z|)
transplant 0.3246    1.3835   1.1322 0.287    0.774

           exp(coef) exp(-coef) lower .95 upper .95
transplant     1.383     0.7228    0.1504     12.73

Concordance= 0.538  (se = 0.126 )
Likelihood ratio test= 0.08  on 1 df,   p=0.8
Wald test            = 0.08  on 1 df,   p=0.8
Score (logrank) test = 0.08  on 1 df,   p=0.8
```

(c)Write the Cox partial likelihood function for the parameters in model 2 and an R function returning the value of the partial log-likelihood at given parameter value(s). Use the R `optim` function to verify the above results (maximum likeli- hood estimates and their standard errors).

- **The partial log-likelihood** The dataset has no tied failure times, and thus the partial likelihood can be written as follows:

$$L(\beta) = \prod_{i=1}^{n} (\frac{\exp(\beta \times transplant_i(t_i))}{\sum_{j=1}^{n} Y_j(t_i) \exp(\beta \times transplant_j(t_i))})^{e_i}$$

  where $Y_j(t_i)$ is the indicator for individual i being at risk at time t, e^i is the indicator for the outcome status of individual i. The log-likelihood function is:

$$\ell_n(\beta) = \sum_{i}^{n} \{e_i(\beta \times transplant_i(t_i)) - e_i \log(\sum_{j=1}^{n} Y_j(t_i) \exp(\beta \times transplant_j(t_i)))\}$$

- **The value of partial log-likelihood** The value of partial log-likelihood is -8.436. The following R codes are used for calculation.

```
beta <- 0.3246
time_seq <- c(unique(trans_dat_long$time2)) ### unique time2 sequence
i <- 0
y <- NULL
loglikeout <- NULL
for(event_time in time_seq){
  i <- i +1
  ## risk set at each event time point
  risk_set_dat <- trans_dat_long %>% filter(time2 >=event_time
  & time1 <event_time)
  y[i] = nrow(risk_set_dat)
  ## denominator in the likelihood
  denominator <- sum(exp(beta*risk_set_dat$transplant))
  ## event set in that point
  event_set_dat <- risk_set_dat %>% filter(time2 == event_time) %>%
    mutate(numerator = exp(beta*transplant),
           loglike_i = log((numerator/denominator)^status ))
  ## sum over log likelihood output
  loglikeout[i] = sum( na.omit(event_set_dat$loglike_i))
}
loglikeout %>%na.omit() %>%  sum()
[1] -8.435596
```

- **Verify MLE estimates** The mle estimates are verified using the `optim` function. the function returns mle of 0.325 and se of 1.132, which are the same as the output from model 2.

```
> Q1_loglike_f <- function(beta) {
+ time_seq <- c(unique(trans_dat_long$time2)) ### unique time2 sequence
+ i <- 0
+ y <- NULL
+ loglikeout <- NULL
+ for(event_time in time_seq){
+   i <- i +1
+   ## risk set at each event time point
+   risk_set_dat <- trans_dat_long %>% filter(time2 >=event_time
& time1 <event_time)
+   y[i] = nrow(risk_set_dat)
+   ## denominator in the likelihood
+   denominator <- sum(exp(beta*risk_set_dat$transplant))
+   ## event set in that point
+   event_set_dat <- risk_set_dat %>% filter(time2 == event_time) %>%
+     mutate(numerator = exp(beta*transplant),
+            loglike_i = log((numerator/denominator)^status ))
+   ## sum over log likelihood output
+   loglikeout[i] = sum( na.omit(event_set_dat$loglike_i))
+ }
+ loglikeout %>%na.omit() %>%  sum()}
>
> inits <- 0
> maxim <- optim(inits, fn=Q1_loglike_f, control=list(fnscale=-1),
method='BFGS', hessian=TRUE)
> mles <- maxim$par %>% round(3)
> ses <- sqrt(diag(solve(-(maxim$hessian)))) %>% round(3)
> cbind(mles, ses)
      mles   ses
[1,] 0.325 1.132
```

Question 2

(a) Fit a Cox model for mortality in the brain dataset including the treatment arm indicator and all the prognostic factors. Report the results in such a form that someone could use the model for prognostic purposes, to calculate the six month absolute risk of death for a new patient. Based on the results, calculate the model-based six month survival probability and risk of death for a 70-year old white male receiving standard of care with local radiation, with less than 75% resection in the new operation one year after the previous operation, having Karnofsky score of more than 70, no previous exposure to nitrosoureas, glioblastoma pathology and inactive grade.

- **Specify the prognostic model** For individual i, the risk is a probability that is defined as the probability of event happening before time t (i.e., $\pi_i(t) = P(T <= t)$). As there are no competing causes (outcome is all-cause death) the risk then can be rewritten w.r.t the survival probability as:

$$\pi_i(t) = P(T <= t) = 1 - P(T > t) = 1 - S(t)$$

where S(t) can be estimated from the cox model. So the prognostic model can be written as:

$$\begin{aligned}\pi_i(t) &= 1 - S_i(t)\\ &= 1 - \exp\{-\Lambda_i(t)\}\\ &= 1 - \exp\{-\Lambda_0(t)\exp(\beta^T X)\} \quad [\text{note:}\lambda_i(t) = \lambda_0(t)\exp(\beta^T X)]\end{aligned}$$

the prognostic model that can be used to calculate the 6-month absolute risk of death can be specified if the following two components are known:

- the coefficient estimates of all covariates (from cox model)
- the cumulative baseline hazard at t = 6 month, which can be estimated using Breslow estimator (`basehaz` function in R)

- **The Cox model output** The following presents the cox model output

```
> Q2_fit_cox <- coxph(formula = Surv(weeks,event)~.,data = brain)
> Q2_fit_cox
Call:
coxph(formula = Surv(weeks, event) ~ ., data = brain)

              coef exp(coef)  se(coef)      z        p
treat    -0.396796  0.672471  0.144512 -2.746 0.006037
resect75 -0.443613  0.641714  0.164990 -2.689 0.007172
age       0.017294  1.017445  0.006029  2.868 0.004127
interval -0.139448  0.869838  0.047318 -2.947 0.003208
karn     -0.376704  0.686119  0.160759 -2.343 0.019115
race      0.592330  1.808197  0.270284  2.192 0.028415
local    -0.466002  0.627506  0.176343 -2.643 0.008228
male     -0.239337  0.787150  0.153227 -1.562 0.118294
nitro     0.480875  1.617490  0.154996  3.102 0.001919
path2    -0.640822  0.526859  0.217740 -2.943 0.003250
path3    -0.804242  0.447427  0.223189 -3.603 0.000314
path4    -0.623663  0.535977  0.429548 -1.452 0.146528
grade    -0.857132  0.424377  0.293056 -2.925 0.003447

Likelihood ratio test=104.3  on 13 df, p=2.383e-16
n= 221, number of events= 206
```

- **The cumulative baseline hazard at t = 6 month** The cumulative baseline hazard at t = 6 month is calculated using `basehaz` in R, based on the output, we know that λ_0(t = 6 month) = 0.649

```
> cum_basehazard_dat <- basehaz(Q2_fit_cox, centered=FALSE)
> time_6mon <- 365/2/7
> cum_base_hazard <- cum_basehazard_dat$hazard[findInterval(time_6mon,
cum_basehazard_dat$time)]
> cum_base_hazard
```

```
[1] 0.6486341
```

- **The fitted prognostic model** Given the coefficients estiamtes and the cumulative baseline hazard at time = 6 month, the fitted prognostic model ca be written as:

$$
\begin{aligned}
\hat{\pi}_i(t = 6month) = 1 - \exp\{-0.649\exp((-0.397) * treat \\
&+ (-0.444) * resect75 + (0.017) * age \\
&+ (-0.139) * interval + (-0.377) * karn + (0.592) * race + \\
&(-0.466) * local + (-0.239) * male \\
&+ (0.481) * nitro + (-0.641) * path2 \\
&+ (-0.804) * path3 + (-0.624) * path4 + (-0.857) * grade)\}
\end{aligned}
$$

- **Risk prediction for a new patient** The covariates of the new patients: treat=0; interval = 1; karn = 1; race = 1; local = 1; male = 1; nitro = 0; path1 = 1; path2 = 0; path3 = 0; path4 = 0; grade =0. The survival probability is 0.313, the risk of death at 6 month is 0.687. The R codes for the calculation are given below:

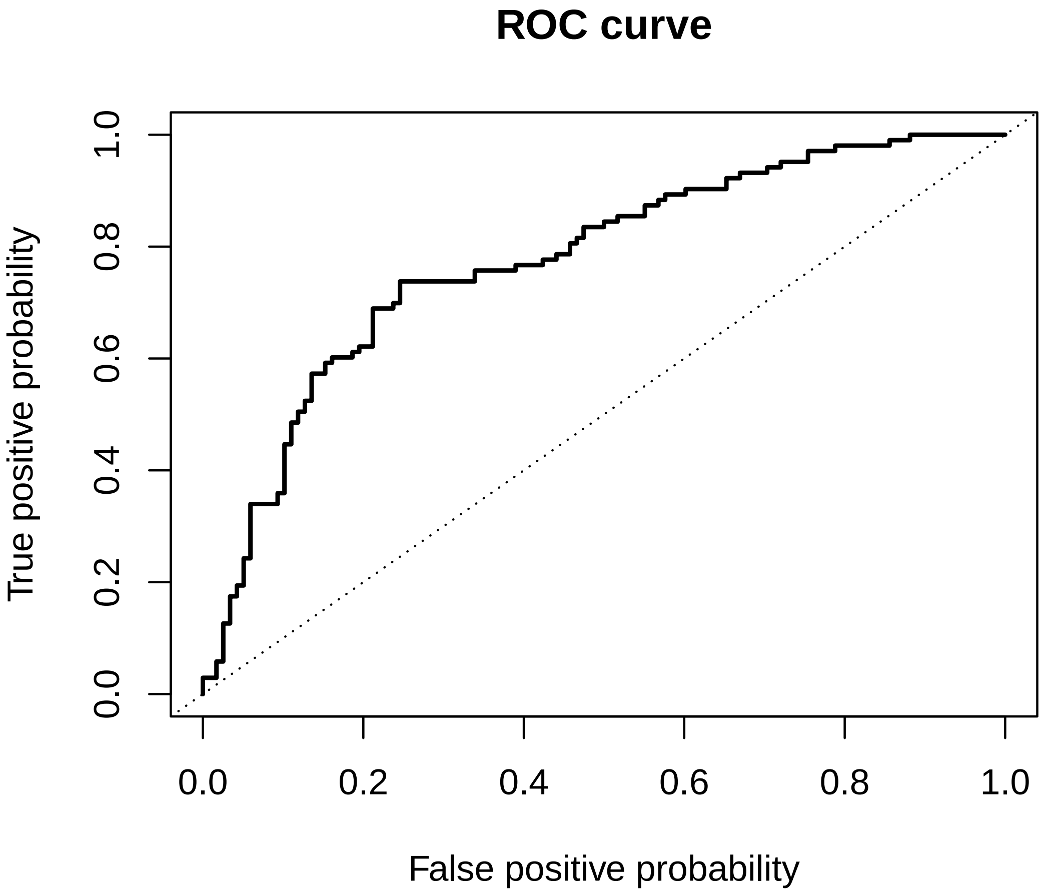
```
> interval = 1;karn = 1;
> race = 1; local = 1;male = 1
> nitro = 0; path1 = 1;path2 = 0;
> path3 = 0; path4 = 0; grade =0
> cum_base_hazard
[1] 0.6486341
>
> # new <- c(treat = 0)
> ## coef %*% new
>
> surv <- exp(-cum_base_hazard*
+    exp((-0.396795745732561)*treat+(-0.443612613402307)*resect75+
+       (0.0172942093041816)*age+(-0.139448299513788)*interval+
+       (-0.376704447251808)*karn+(0.592330167341483)*race+
+       (-0.466002307985353)*local+(-0.239336792392096)*male+
+       (0.480875474089499)*nitro+(-0.640822151976394)*path2+
+       (-0.804241648472914)*path3+(-0.62366328303592)*path4+
+       (-0.857131918548911)*grade))
>
> surv
[1] 0.3134402
>
> 1- surv
[1] 0.6865598
```

Question 3 Check the discrimination and calibration of the model you fitted in Q2 (choose one statistic and one graphical presentation to describe each).

- **Discrimination** can be evaluated using AUC and can be visualized using ROC curve.
  - The AUC of this model is 0.775.

    ```
    > roc_list <- survivalROC(Stime=brain$weeks,
    +                         status=brain$event,
    +                         marker=risk_prob,
    +                         predict.time=time_6mon,
    +                         method="KM")
    >
    > roc_list$AUC
    [1] 0.775218
    ```

  - The ROC curve is shown below. The ROC curve is above the diagonal line (model without prognostic ability), indicating good discrimination.



- **Calibration** can be evaluated using Hosmer-Lemeshow statistics and can be visualized using calibration plot.

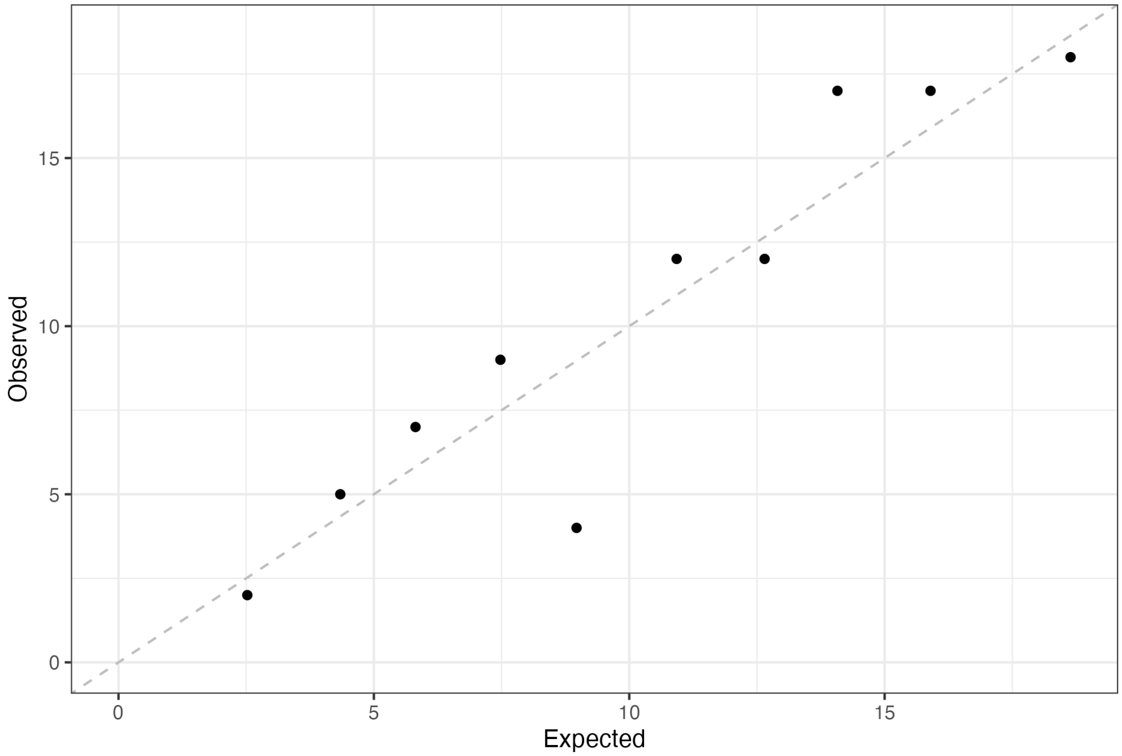
- The Hosmer-Lemeshow statistics compares the observed event and expected event counts using the individual follow up data where the data is split into K groups by the estimated risk. The observed expected counts in each groups are compared. The Hosmer-Lemeshow test statistics is defined as follows:

$$\sum_{k=1}^{K} \frac{(O_k - E_k)^2}{n_k \bar{\pi}_k (1 - \bar{\pi}_k)} \sim \chi^2(K-2)$$

  n_k: number of subjects in the k^{th} risk group
  O_k: observed counts in the k^{th} risk group $O_k = n_k \times (1 - \hat{S}(t = 6month)$, $\hat{S}(t = 6month)$ can be estimated from KM. (censoring and no competing causes)
  E_k: expected counts in the k^{th} risk group. $E_k = n_k \times \bar{\pi}_k$
  $\bar{\pi}_k$: the average of the predicted risk in the k^{th} risk group The following codes are used to calculate the expected and observed counts in each risk group. The Hosmer-Lemeshow test statistics is 8.083 with p-value of 0.425.

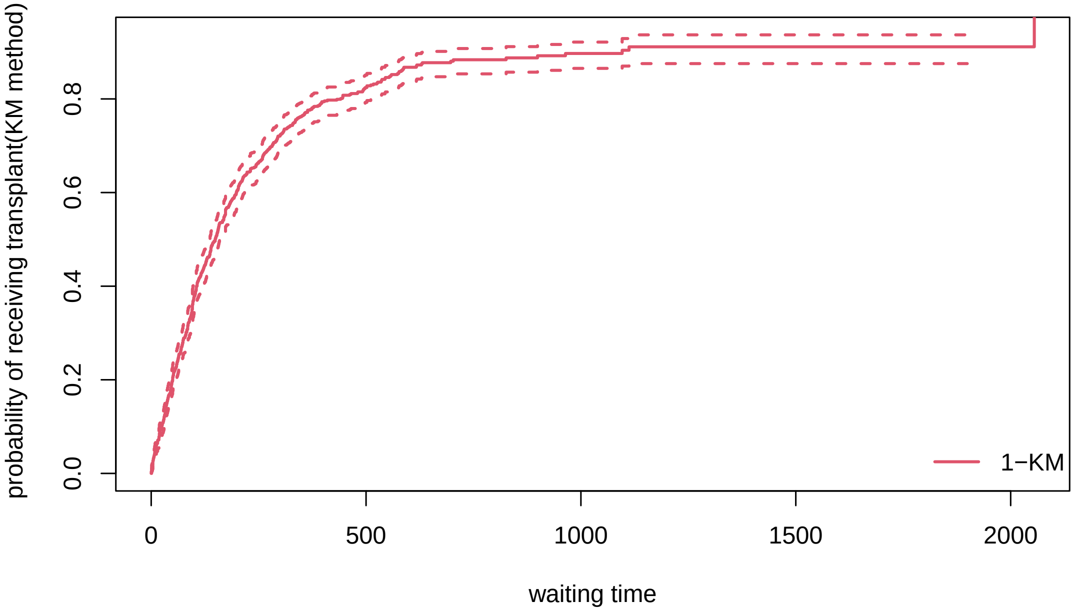
```
> deciles <- quantile(risk_prob,probs = seq(0.1,0.9,0.1))
> ## 10 chunks
> cat_k <- cut(risk_prob,c(0,deciles,1),labels = 1:10)
>
> n <- table(cat_k)
> pred_pi_k <- km_risk <- rep(NA, 10)
> for (k in 1:10) {
+   fit <- summary(survfit(Surv(weeks,event)~1,
data=brain[which(cat_k==k),]))
+   pred_pi_k[k] <- mean(risk_prob[cat_k==k])
+   km_risk[k] <- (1 - fit$surv[findInterval(time_6mon, fit$time)])
+ }
>
> Ek = n*pred_pi_k
> Ok = n*km_risk
>
> chisq_HL <- sum((Ok - Ek)^2/(n * pred_pi_k* (1 - pred_pi_k)))
> chisq_HL
[1] 8.083051
> pchisq(chisq_HL, df=length(unique(cat_k))-2, lower.tail=FALSE)
[1] 0.4253998
```

- The corresponding calibration plot is shown below(codes are in the appendix) we see most of the dots are scattered around the diagonal line (y=x), indicating good calibration.

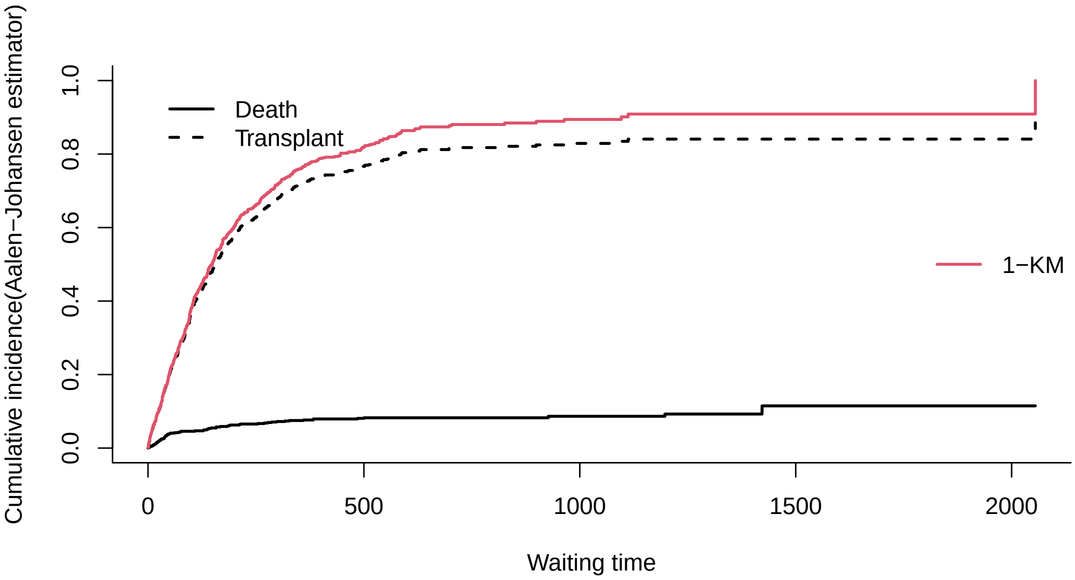


Question 4 From the transplant data, estimate (i) the probability of receiving transplant using the Kaplan-Meier method and (ii) the cumulative incidence of receiving transplant using the non-parametric cumulative incidence estimator. Present the results as curves over time. Which method would you prefer and why?

- (i) **probability of receiving transplant using the Kaplan-Meier method** is presented in the following figure:



- (ii) **Cumulative incidence(Aalen-Johansen estimator)** is presented below. The black dashed and solid lines are the Aalen-Johansen estimator results of cumulative incidence for transplant and death respectively, and the red curve is the KM estimator result.

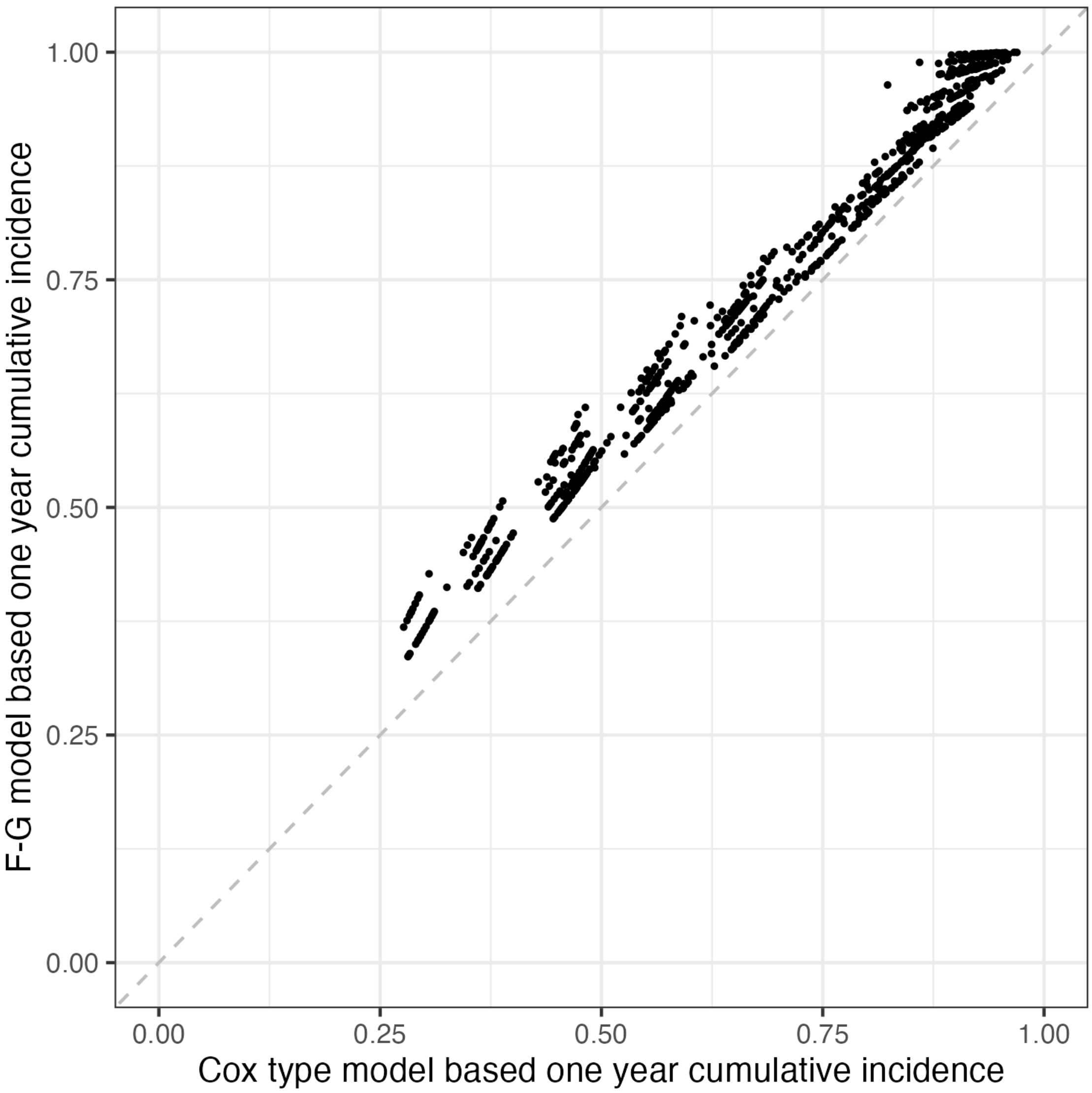


- **I prefer the Cumulative incidence(Aalen-Johansen estimator) method** because the KM estimator calculates the probability of receiving transplant assuming the absence of death (i.e., death is treated as censoring). However, death is a competing risk and people who died would never receive transplant. Therefore, in the KM method, it is inappropriate to assume people do not die.

Question 5 Use the fitted Cox-type (cause-specific hazard) and Fine Gray (subdistribution hazard) models adusting for age, sex, ABO blood group and year to calculate individual-level one-year cumulative incidences of receiving the transplant. Present these in a scatterplot. How do the results compare between the models? (You don't have to validate the models, just comment on whether the cause-specific hazard and subdistribution hazard model results are "similar".)

The age variable has 18 missing values, for the purpose of this assignment, I only used the complete cases for model building. The one-year cumulative incidences of receiving the transplant is presented in the following. And we see that:

- Overall, the results are similar, the dots lie around the diagonal line.
- However, Fine and Gray model always gives higher one-year cumulative incidence compared to cox-type model.



- **Cox type model**

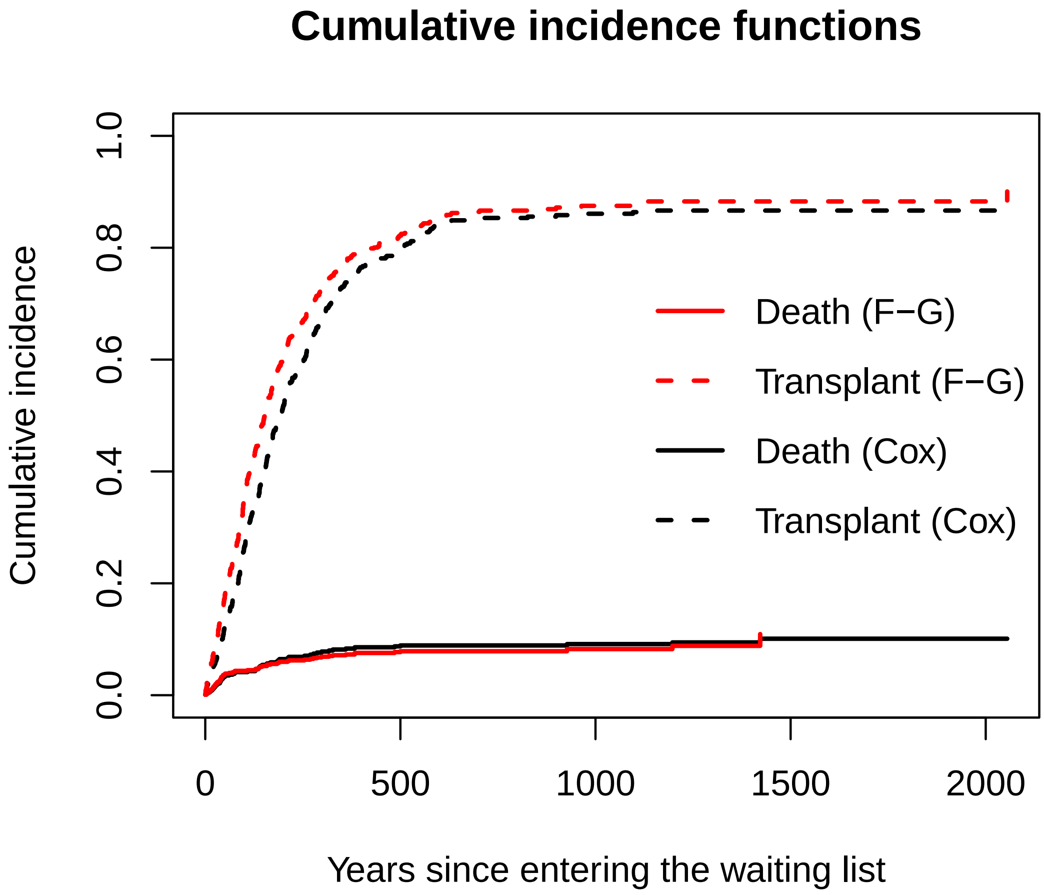
```
> model12 <- coxph(Surv(transplant2$futime,
+                         as.factor(transplant2$event1))~
+                      age +sex1 + aboA +aboB+aboAB +year, id=id,
+                    data=transplant2)
> summary(model12)
Call:
coxph(formula = Surv(transplant2$futime, as.factor(transplant2$event1)) ~
    age + sex1 + aboA + aboB + aboAB + year, data = transplant2,
    id = id)

  n= 797, number of events= 684

               coef exp(coef)  se(coef) robust se      z Pr(>|z|)
```

```
age_1:2    0.019871  1.020069  0.012939  0.011892  1.671 0.094740 .
sex1_1:2   0.453554  1.573896  0.257779  0.252546  1.796 0.072506 .
aboA_1:2  -0.011232  0.988831  0.286165  0.286130 -0.039 0.968687
aboB_1:2   0.154845  1.167477  0.366768  0.370834  0.418 0.676270
aboAB_1:2  0.233344  1.262816  0.605478  0.574928  0.406 0.684841
year_1:2  -0.183108  0.832679  0.052462  0.048535 -3.773 0.000162 ***
age_1:3   -0.003038  0.996967  0.004029  0.004628 -0.656 0.511553
sex1_1:3   0.059804  1.061628  0.081380  0.108058  0.553 0.579961
aboA_1:3   0.586539  1.797755  0.090500  0.115218  5.091 3.57e-07 ***
aboB_1:3   0.266730  1.305688  0.131536  0.148926  1.791 0.073290 .
aboAB_1:3  0.675516  1.965047  0.188234  0.270489  2.497 0.012511 *
year_1:3  -0.304127  0.737767  0.016314  0.034363 -8.850  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

          exp(coef) exp(-coef) lower .95 upper .95
age_1:2      1.0201     0.9803    0.9966    1.0441
sex1_1:2     1.5739     0.6354    0.9594    2.5819
aboA_1:2     0.9888     1.0113    0.5644    1.7325
aboB_1:2     1.1675     0.8565    0.5644    2.4149
aboAB_1:2    1.2628     0.7919    0.4092    3.8969
year_1:2     0.8327     1.2009    0.7571    0.9158
age_1:3      0.9970     1.0030    0.9880    1.0061
sex1_1:3     1.0616     0.9419    0.8590    1.3121
aboA_1:3     1.7978     0.5562    1.4344    2.2532
aboB_1:3     1.3057     0.7659    0.9752    1.7483
aboAB_1:3    1.9650     0.5089    1.1565    3.3390
year_1:3     0.7378     1.3554    0.6897    0.7892

Concordance= 0.743  (se = 0.01 )
Likelihood ratio test= 381  on 12 df,   p=<2e-16
Wald test            = 127.5  on 12 df,   p=<2e-16
Score (logrank) test = 447.7  on 12 df,   p=<2e-16,   Robust = 289.4
p=<2e-16

  (Note: the likelihood ratio and score tests assume independence of
     observations within a cluster, the Wald and robust score tests do not)
```

- **Fine-Gray model**

```
> mm <- cbind(transplant2$age-mean(transplant2$age),
+             transplant2$sex1,
+             transplant2$aboA,
+             transplant2$aboB,
+             transplant2$aboAB,
+             transplant2$year-mean( transplant2$year)
+ ) %>% data.frame()
> colnames(mm) <- c('age','sex1','aboA','aboB','aboAB','year')
> fgmodel2 <- crr(transplant2$futime,
+                 transplant2$event1,
+                 cov1=mm,
+                 failcode=2,
+                 cencode=0)
> summary(fgmodel2)
Competing Risks Regression

Call:
crr(ftime = transplant2$futime, fstatus = transplant2$event1,
    cov1 = mm, failcode = 2, cencode = 0)

         coef exp(coef) se(coef)     z p-value
age   -0.0062     0.994  0.00453 -1.37 1.7e-01
sex1  -0.1028     0.902  0.09653 -1.07 2.9e-01
aboA   0.5121     1.669  0.10504  4.88 1.1e-06
aboB   0.2518     1.286  0.13520  1.86 6.2e-02
aboAB  0.5504     1.734  0.25785  2.13 3.3e-02
year  -0.2326     0.792  0.02387 -9.74 0.0e+00

      exp(coef) exp(-coef)  2.5% 97.5%
age       0.994      1.006 0.985  1.00
sex1      0.902      1.108 0.747  1.09
aboA      1.669      0.599 1.358  2.05
aboB      1.286      0.777 0.987  1.68
aboAB     1.734      0.577 1.046  2.87
year      0.792      1.262 0.756  0.83

Num. cases = 797
Pseudo Log-likelihood = -3593
Pseudo likelihood ratio test = 245  on 6 df,
```

- **The cumulative incidence plots**



Appendix

```
## Question 1 ##
###Q1c
library(dplyr)
Q1_dat <- data.frame( id = c(1:10),
            sex = c(rep(1,5),rep(0,5)),
            time = c(3,5,5,6,8,4,7,8,9,10),
            status = c(1,0,1,1,0,0,1,0,1,0),
            transplant = c(0,0,1,0,1,1,1,0,1,1),
            wait = c(NA,NA,0,NA,6,0,5,NA,5,3)
)

Q1_trans <- Q1_dat %>% filter(transplant==1 & wait>0)

Q1_before_trans <-  Q1_trans %>%
  select(id, sex)
```

```
Q1_before_trans$time1 = 0

Q1_before_trans$time2 = Q1_trans$wait

Q1_before_trans$transplant = 0

Q1_before_trans$status =  as.numeric( (Q1_trans$time <= Q1_trans$wait) &
                                        Q1_trans$status==1)

Q1_after_trans <- Q1_trans %>% select(id, sex)

Q1_after_trans$time1  = Q1_trans$wait

Q1_after_trans$time2  = Q1_trans$time

Q1_after_trans$transplant  = 1


Q1_after_trans$status =  as.numeric( (Q1_trans$time >
Q1_trans$wait) & Q1_trans$status==1)

nontrans_dat <-  Q1_dat %>% filter( !(transplant==1 & wait>0) )

nontrans_dat_long <- nontrans_dat %>%
  mutate(time1 = 0,time2 = time,transplant = transplant, status = status) %>%
  select(id, sex, time1, time2, transplant, status)

trans_dat_long <- rbind(Q1_before_trans,
      Q1_after_trans) %>%
  bind_rows(nontrans_dat_long) %>%
  arrange(time2)
```

```
library(survival)
# fit cox model with time-dependent variable
Q1_model <- coxph(Surv(time1,time2,status)~transplant,
                  data=trans_dat_long)
summary(Q1_model)
```

```
beta <- 0.3246
time_seq <- c(unique(trans_dat_long$time2)) ### unique time2 sequence
```

```
i <- 0
y <- NULL
loglikeout <- NULL
for(event_time in time_seq){
  i <- i +1
  risk_set_dat <- trans_dat_long %>% filter(time2 >=event_time & time1 <event_ti
 y[i] = nrow(risk_set_dat)
denominator <- sum(exp(beta*risk_set_dat$transplant))
 event_set_dat <- risk_set_dat %>% filter(time2 == event_time) %>%
   mutate(numerator = exp(beta*transplant),
          loglike_i = log((numerator/denominator)^status ))

 loglikeout[i] = sum( na.omit(event_set_dat$loglike_i))
}
loglikeout %>%na.omit()%>%  sum()


Q1_loglike_f <- function(beta) {
time_seq <- c(unique(trans_dat_long$time2)) ### unique time2 sequence
i <- 0
y <- NULL
loglikeout <- NULL
for(event_time in time_seq){
  i <- i +1
  ## risk set at each event time point
  risk_set_dat <- trans_dat_long %>%
    filter(time2 >=event_time & time1 <event_time)
  y[i] = nrow(risk_set_dat)
  ## denominator in the likelihood
  denominator <- sum(exp(beta*risk_set_dat$transplant))
  ## event set in that time point
  event_set_dat <- risk_set_dat %>% filter(time2 == event_time) %>%
    mutate(numerator = exp(beta*transplant),
           loglike_i = log((numerator/denominator)^status ))
  ## sum over log likelihood output
  loglikeout[i] = sum( na.omit(event_set_dat$loglike_i))
}
loglikeout %>%na.omit() %>%  sum()}

inits <- 0
maxim <- optim(inits, fn=Q1_loglike_f, control=list(fnscale=-1), method=’BFGS’,
mles <- maxim$par %>% round(3)
ses <- sqrt(diag(solve(-(maxim$hessian)))) %>% round(3)
cbind(mles, ses)
```

```
######################
# nrow(Q1_dat)
# Q1_dat2 <- NULL
# for (i in 1:nrow(Q1_dat)) {
#   if (Q1_dat$transplant[i] == 0) {
#     Q1_dat2 <- rbind(Q1_dat2, data.frame(Q1_dat[i,], id=i,
# start=0, stop=Q1_dat$time[i], event=Q1_dat$status[i], tr=0))
#   }
#   else if (Q1_dat$transplant[i] == 1) {
#     Q1_dat2 <- rbind(Q1_dat2, data.frame(Q1_dat[i,], id=i,
# start=0, stop=Q1_dat$wait[i], event=0, tr=0))
#     Q1_dat2 <- rbind(Q1_dat2, data.frame(Q1_dat[i,], id=i,
# start=Q1_dat$wait[i], stop=Q1_dat$time[i],
# event=Q1_dat$status[i], tr=1))
#   }
# }
#
# Q1_dat2 <- Q1_dat2 %>% select(id,sex,start,stop,tr,event)

## Question 2 ##
library(readr)
library(survival)
brain <- read_csv("data/brain.csv")

brain <- brain[complete.cases(brain),] %>%
  mutate(path = as.factor(path))



Q2_fit_cox <- coxph(formula = Surv(weeks,event)~.,data = brain)
Q2_fit_cox
lp <- model.matrix(Q2_fit_cox) %*% coef(Q2_fit_cox)

cum_basehazard_dat <- basehaz(Q2_fit_cox, centered=FALSE)
time_6mon <- 365/2/7
cum_base_hazard <- cum_basehazard_dat$hazard[findInterval(time_6mon,
cum_basehazard_dat$time)]
cum_base_hazard

surv_prob  =  exp(-cum_base_hazard*exp(lp))
risk_prob = 1- surv_prob

## use the code to verify the result
# fit <- survfit(Q2_fit_cox, newdata=brain)
# risk2 <- 1.0 - fit$surv[findInterval(time_6mon, cum_basehazard_dat$time),]
#
# plot(risk_prob,risk2)

Q2_fit_cox$formula

c(treat , resect75 , age , interval , karn ,
  race , local , male , nitro , path , grade)

cox_output <- Q2_fit_cox$coefficients %>% data.frame()

cox_output$coef = Q2_fit_cox$coefficients

cox_output$var_name <-  rownames(cox_output)

cox_output$lp <- paste0(paste0("(",cox_output$coef,")*"),cox_output$var_name)

paste0(cox_output$lp,collapse="+")

treat = 0; resect75 = 0; age = 70;
interval = 1;karn = 1;
race = 1; local = 1;male = 1
nitro = 0; path1 = 1;path2 = 0;
path3 = 0; path4 = 0; grade =0
cum_base_hazard

# new <- c(treat = 0)
## coef %*% new

surv <- exp(-cum_base_hazard*
      exp((-0.396795745732561)*treat+(-0.443612613402307)*resect75+
            (0.0172942093041816)*age+(-0.139448299513788)*interval+
            (-0.376704447251808)*karn+(0.592330167341483)*race+
            (-0.466002307985353)*local+(-0.239336792392096)*male+
            (0.480875474089499)*nitro+(-0.640822151976394)*path2+
            (-0.804241648472914)*path3+(-0.62366328303592)*path4+
            (-0.857131918548911)*grade))

surv

1- surv
## Question 3 ##
## discrimination

# divide into small intervals based on the estiamted risk (10 quantiles)
```

```
library(readr)
brain <- read_csv("data/brain.csv")

brain <- brain[complete.cases(brain),] %>%
  mutate(path = as.factor(path))

Q2_fit_cox <- coxph(formula = Surv(weeks,event)˜.,data = brain)
Q2_fit_cox
lp <- model.matrix(Q2_fit_cox) %*% coef(Q2_fit_cox)

lp %>% data.frame()

cum_basehazard_dat <- basehaz(Q2_fit_cox, centered=FALSE)
time_6mon <- 365/2/7
cum_base_hazard <- cum_basehazard_dat$hazard[findInterval(time_6mon,
cum_basehazard_dat$time)]
cum_base_hazard

surv_prob  =  exp(-cum_base_hazard*exp(lp))
risk_prob = 1- surv_prob

deciles <- quantile(risk_prob,probs = seq(0.1,0.9,0.1))
## 10 chunks
cat_k <- cut(risk_prob,c(0,deciles,1),labels = 1:10)

n <- table(cat_k)
pred_pi_k <- km_risk <- rep(NA, 10)
for (k in 1:10) {
  fit <- summary(survfit(Surv(weeks,event)˜1, data=brain[which(cat_k==k),]))
  pred_pi_k[k] <- mean(risk_prob[cat_k==k])
  km_risk[k] <- (1 - fit$surv[findInterval(time_6mon, fit$time)])
}

Ek = n*pred_pi_k
Ok = n*km_risk

chisq_HL <- sum((Ok - Ek)^2/(n * pred_pi_k* (1 - pred_pi_k)))
pchisq(chisq_HL, df=length(unique(cat_k))-2, lower.tail=FALSE)
library(ggplot2)
cali_plot <- ggplot(data = data.frame(Ok = Ok, Ek = Ek),aes(y = Ok,x = Ek))+
  geom_point()+
  theme_bw()+
  xlab(’Expected’)+
  ylab(’Observed’) +
  geom_abline(slope = 1,intercept = 0,lty=’dashed’,color=’gray’)+
  xlim(c(0,max(Ok,Ek)))+
  ylim(c(0,max(Ok,Ek)))

ggsave(cali_plot,file = "cali_plot.png",dpi=300)



### ROC curve
library(survivalROC)

roc_list <- survivalROC(Stime=brain$weeks,
                   status=brain$event,
                   marker=risk_prob,
                   predict.time=time_6mon,
                   method="KM")

roc_list$AUC
roc_dat <- data.frame(TP = roc_list$TP,FP = roc_list$FP)



 plot(roc_dat$FP, roc_dat$TP, col=’black’,
          lwd=2, type=’s’, xlab=’False positive probability’,
          ylab=’True positive probability’, main=’ROC curve’)
# abline(h=c(0,1),lty=’dotted’)
# abline(v=c(0,1),lty=’dotted’)
abline(a=0, b=1,lty=’dotted’)


######

# roc_dat1 <- roc_dat %>% arrange(TP,FP)
#
#
# ggplot(data = roc_dat1, aes(x = FP, y = TP)) +
#   geom_step()
# geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
#   labs(title = "ROC Curve",
#        x = "False Positive Rate (1 - Specificity)",
#        y = "True Positive Rate (Sensitivity)")
# plot(as.numeric(Ek), as.numeric(Ok), xlim=c(0,max(Ek)),
ylim=c(0,max(Ok)), xlab=’Expected’, ylab=’Observed’)
# abline(a=0, b=1, lty=’dashed’)
#
```

```
#
# cali_dat <- data.frame(k = k) %>% ## the number of obs in each chunk
#   group_by(k) %>%
#   summarise(Nk = n()) %>%
#   ## right join the averaged estiamted risk
#   right_join(., data.frame(risk_prob,
#                             k = k ) %>% group_by(k) %>%
#                 summarise(mean_pi_k = mean(risk_prob))) %>%
#   right_join(.,data.frame(surv_prob, k = k ) %>% group_by(k) %>%
#                 summarise(mean_pi_k = mean(risk_prob)))
#
#
# E_k <- Nk*pi_hat_k
#

## Question 4 ##
## load dataset
library(survival)
library(dplyr)
#data("transplant")

transplant0 <- transplant[complete.cases(transplant),]

transplant1 <- transplant0 %>%
  mutate(event1 = event %>% as.numeric(),
         event1 = ifelse(event1 ==4, 1,event1),
         event1 = case_when(event1==1~0,
                            event1 == 2 ~1,
                            event1 ==3 ~2
                            ))
## event1 = 0 censoring =1 death =2 transplant
fit_km <- survfit(Surv(futime,event1==2)~1, data=transplant1)
summary(fit)



plot(fit_km, col=2, xlab="waiting time",
     ylab=" probability of receiving transplant(KM method)",
     lwd=2, conf.int=T,
     # mark.time=TRUE,
     fun="event")
legend("bottomright", col=2, legend=c("1-KM"), lwd=2, bty=’n’)
abline(h=seq(0,1,0.2), lty=’dotted’)

# ###
```

```
cifit <- survfit(Surv(futime,as.factor(event1))~1, data=transplant1)
summary(cifit)
plot(cifit,
     xlab="Years since entering the waiting list",
    ylab="Cumulative incidence",
    lwd=2, conf.int=T, mark.time=TRUE,
     main="Waiting time for transplantation", fun="event",
     lty=c(’solid’,’dashed’))
abline(h=seq(0,1,0.2), lty=’dotted’)
legend("topleft", legend=c(’Transplant’,’Death’), lwd=2,
lty=c(’solid’,’dashed’), bty=’n’)

# Using cmprsk package:

library(cmprsk)
cifit <- cuminc(transplant1$futime, transplant1$event1, cencode=0)
cifit

plot(cifit, curvlab=c(’Death’,’Transplant’),conf.int=T,
     xlab="Years since entering the waiting list",
     ylab="Cumulative incidence(Aalen-Johansen estimator)", lwd=2)
#abline(h=seq(0,1,0.2), lty=’dotted’)
lines(fit_km, lwd=2, col=2, conf.int=F, fun="event")
legend("right", col=2, legend=c("1-KM"), lwd=2, bty=’n’)
#### age, sex, ABO blood group and year

## Question 5 ##
## load dataset
library(survival)
library(dplyr)
library(cmprsk)
#data("transplant")

transplant0 <- transplant[complete.cases(transplant),]

transplant1 <- transplant0 %>%
  mutate(event1 = event %>% as.numeric(),
         event1 = ifelse(event1 ==4, 1,event1),
         event1 = case_when(event1==1~0,
                            event1 == 2 ~1,
                            event1 ==3 ~2
         ))
## event1 = 0 censoring; =1 death ;=2 transplant
transplant2 <- transplant1 %>% select(-event) %>%
  mutate(id = 1:nrow(transplant2),
         sex1 =ifelse(sex==’m’,1,0)
  )
# transplant2$abo1 = as.numeric(transplant2$abo)

transplant2$aboA = ifelse(transplant2$abo=="A",1,0)
transplant2$aboB = ifelse(transplant2$abo=="B",1,0)
transplant2$aboAB = ifelse(transplant2$abo=="AB",1,0)


model12 <- coxph(Surv(transplant2$futime,
                      as.factor(transplant2$event1))˜
                   age +sex1 + aboA +aboB+aboAB +year, id=id,
                 data=transplant2)
summary(model12)
mm <- model.matrix(model12)

# The cumulative incidence estimates are given by the survfit function:

cifit <- survfit(model12, newdata=as.data.frame(t(colMeans(mm))))
summary(cifit)
str(cifit)
dim(cifit$pstate)
plot(cifit)

####



mm <- cbind(transplant2$age-mean(transplant2$age),
            transplant2$sex1,
            transplant2$aboA,
            transplant2$aboB,
            transplant2$aboAB,
            transplant2$year-mean( transplant2$year)
) %>% data.frame()
colnames(mm) <- c(’age’,’sex1’,’aboA’,’aboB’,’aboAB’,’year’)

fgmodel1 <- crr(transplant2$futime,
                transplant2$event1,
                cov1=mm,
                failcode=1,
                cencode=0)

summary(fgmodel1)
```

```
fgmodel2 <- crr(transplant2$futime,
                transplant2$event1,
                cov1=mm,
                failcode=2,
                cencode=0)

summary(fgmodel2)

# Cumulative incidence at average covariate values:

fgci1 <- predict(fgmodel1, cov1=t(colMeans(mm)))
fgci2 <- predict(fgmodel2, cov1=t(colMeans(mm)))

plot(cifit, ylim=c(0,1), xlim=c(0,max(transplant2$futime)), lwd=2,
xlab="Years since entering the waiting list",
 ylab="Cumulative incidence", main=’Cumulative incidence functions’,
 lty=c(’solid’,’dashed’))
lines(fgci1[,1], fgci1[,2], type=’s’, lwd=2, lty=c(’solid’), col=’red’)
lines(fgci2[,1], fgci2[,2], type=’s’, lwd=2, lty=c(’dashed’), col=’red’)
legend(’right’, legend=c(’Death (F-G)’,’
Transplant (F-G)’,’Death (Cox)’,’Transplant (Cox)’), lwd=2,
lty=c(’solid’,’dashed’,’solid’,’dashed’),
col=c(’red’,’red’,’black’,’black’), bty=’n’)
#abline(h=seq(0,1,0.2), lty=’dotted’)

# Comparison of predictions: calculate 1 year cumulative incidences of
# receiving transplant for everyone in the dataset:

ciall <- rep(NA, nrow(transplant2))
fgciall <- rep(NA, nrow(transplant2))
s <- 365
mm <- model.matrix(model12)
cifit <- survfit(model12, newdata=as.data.frame(mm))
str(cifit)
dim(cifit$pstate)
idx <- findInterval(s, cifit$time)

ciall <- NULL
for (i in 1:nrow(transplant2)) {
  ciall <- c(ciall, cifit$pstate[idx,i,3])
}
sum(ciall)

mm <- cbind(transplant2$age-mean(transplant2$age),
            transplant2$sex1,
            transplant2$aboA,
            transplant2$aboB,
            transplant2$aboAB,
            transplant2$year-mean(transplant2$year)
)
colnames(mm) <- c(’age’,’sex1’,’aboA’,’aboA’,’aboAB’,’year’)

fgci <- predict(fgmodel2, cov1=mm)
fgciall <- fgci[findInterval(s, fgci[,1]),2:ncol(fgci)]
sum(fgciall)

prediction_dat <- data.frame(ciall = ciall,
           fgciall = fgciall)

library(ggplot2)

Q5_pred_plot <- prediction_dat %>% ggplot(aes(x=ciall,y=fgciall))+
  geom_point(size=0.5)+
  xlim(0,1)+
  ylim(0,1)+
  geom_abline(slope = 1,
              intercept = 0,
              lty=’dashed’,color=’gray’)+
  theme_bw()+
  xlab("Cox type model based one year cumulative incidence")+
  ylab("F-G model based one year cumulative incidence")

ggsave(Q5_pred_plot,file=’Q5_pred_plot.png’,
       width = 5,height = 5)
plot(ciall, fgciall,xlim=c(0,1),
     ylim=c(0,1),
     xlab=’Cox model based CI’,
     ylab=’F-G model based CI’,
     main=’Comparison of predictions from the two models’)
abline(a=0, b=1, lty=’dotted’, col=’blue’)
```