CHL5209 HW3
Due March 22

Q1

(a) Specify the fitted model. What is the interpretation of the model parameter(s)? What does the model assume about the hazards over time?

log(scale) = -alpha

- The codes indicates the model assumes event time follows Weibull distribution with shape of 1, which is the exponential model. and it used the PH parametrization. So the model can be specified as:

$$\lambda(t; \lambda_0, \beta) = \lambda_0 \exp(\beta \times arm)$$

or

$$\log(\lambda(t; \lambda_0, \beta)) = \log(\lambda_0) + (\beta \times arm)$$

where λ_0 is the baseline hazard, and $\log(\beta)$ is the treatment effect estimate. Plug in $\lambda_0 = \exp(-intercept) = exp(-\log(scale)) = \exp(-0.642)$ and $\beta = -0.744$, gives us the fitted model as follows:

$$\lambda(t) = \exp(-0.642 - 0.744 \times arm)$$

or

$$\log(\lambda(t)) = -0.642 - 0.744 \times arm$$

- The interpretation of intercept and slope:
  - the intercept corresponds to the baseline hazard $\lambda_0 = \exp(-0.642) = 0.526$, which means the hazard rate in the control arm is 0.526.
  - the slope -0.744 is the log(HR): $\beta = -0.744 = \log(\frac{\lambda_1(t)}{\lambda_0(t)})$, so $HR = \exp(\beta) = \exp(-0.744) = 0.475$, which indicates the hazard of death in the treatment arm is 0.475 times that in the control arm.
- The model assume the hazards are constant over time.

(b)Write the log-likelihood function for the model parameter(s)

Under independent censoring and non-informative censoring assumptions, the likelihood function of the data can be specified as:

$$L(\lambda_0, \beta) \propto \prod_{i=1}^{n} [\lambda_i(t_i; \lambda_0, \beta)^{e_i} S_i(t_i; \lambda_0, \beta)] \quad \text{[note: } i \text{ represents the } i^{th} \text{ subject]}$$

Where e_i is the indicator function for the event and:

$$\lambda_i(t_i; \lambda_0, \beta) = \lambda_0 \exp(\beta \times arm_i)$$

and the survival function can be specified as:

$$\begin{aligned} S_i(t_i; \lambda_0, \beta) &= \exp(-\int_0^{t_i} \lambda_i(u; \lambda_0, \beta) du) \\ &= \exp(-\int_0^{t_i} \lambda_0 \exp(\beta \times arm_i) du) \\ &= \exp(-\lambda_0 \exp(\beta \times arm_i) t_i) \end{aligned}$$

So the likelihood function can be further written as a function of the parameters λ_0 and β:

$$\begin{aligned} L(\lambda_0, \beta) &\propto \prod_{i=1}^{n} [\lambda_i(t_i; \lambda_0, \beta)^{e_i} S_i(t_i; \lambda_0, \beta)] \\ &= \prod_{i=1}^{n} (\lambda_0 \exp(\beta \times arm_i))^{e_i} \exp(-\lambda_0 \exp(\beta \times arm_i) t_i) \end{aligned}$$

So the partial log-likelihood function can be written as:

$$\ell_n(\lambda_0, \beta) = \sum_{i=1}^{n} \{e_i(\log(\lambda_0) + \beta \times arm_i) - \lambda_0 \exp(\beta \times arm_i) t_i\}$$

as $\lambda_0 = exp(-\alpha)$, α is the $\log(scale)$ in the `phreg` output, then the log-likelihood function can be written as:

$$\ell_n(\alpha, \beta) = \sum_{i=1}^{n} \{e_i(-\alpha + \beta \times arm_i) - \exp(-\alpha + \beta \times arm_i) t_i\}$$

(c) Write an R function returning the value of the partial log-likelihood at given parameter value(s) and use the R optim function to verify the above results (maximum likelihood estimates and their standard errors).
The partial log-likelihood at $\alpha = 0.642$ and $\beta = -0.744$ is -20.14074.

```
> observed_t <- c(1,2.5,3,4,4.5,5,0.5,0.75,
+                  1.25,1.5,2,3.5)
> status <- c(1,1,1,1,1,0,1,1,1,0,1,1)
> arm <- c(rep(1,6),rep(0,6))
>
> alpha <- 0.642
> beta <- -0.744
>
> sum((status*(-alpha+beta*arm))- exp(-alpha+beta*arm)*observed_t)
[1] -20.14074
```

To verify whether the numerical results align with the model output, I used the `optim` funtion in R and the results are the same:

```
loglik <- function(par) {
  return(sum((status*(-par[1]+par[2]*arm))-
               exp(-par[1]+par[2]*arm)*observed_t))
}
library(dplyr)
inits <- rep(0.0, 2)
```

```
maxim <- optim(inits, fn=loglik, 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,]  0.642 0.447
[2,] -0.744 0.632
```

Q2

(a) Specify the fitted model. What is the interpretation of the model parameter(s)? What does the model assume about the hazards over time?

- The `coxph` command indicates a cox model, which can be specified as:

  $$\lambda_i(t;\beta) = \lambda_0(t)\exp(\beta \times arm_i)$$

  where $\lambda_0(t)$ is the baseline hazard rate. Plug in $\beta = -1.2760$, the fitted model then can be written as:

  $$\lambda_i(t;\beta) = \lambda_0(t)\exp(-1.276 \times arm_i)$$

- the parameter $\beta = -1.276 = log(\frac{\lambda_1(t;\beta)}{\lambda_0(t;\beta)})$ can be interpreted as the log hazard rate ratio comparing the death between the treatment and control arm over time. so the $HR = exp(-1.276) = 0.2792$, which indicates that the hazard of death in the treatment arm is 0.2792 times that in the control arm.

- the model assumes that the hazard ratio comparing the treatment group and control group is a constant.

(b)Write the Cox partial log-likelihood function for the model parameter(s).

The data shows there is no tied faulure times, so the partial likelihood when there is censoring and no tied faulure times can be expressed as:

$$L(\beta) \propto \prod_{i=1}^{n}\left(\frac{\exp(\beta \times arm_i)}{\sum_{j=1}^{n} Y_j(t_i)\exp(\beta \times arm_j)}\right)^{e_i}$$

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:

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

(c) Write an R function returning the value of the log-likelihood at given parameter value(s) and use the R optim function to verify the above results (maximum likelihood estimates and their standard errors).

The log-likelihood value at $\beta = -1.276$ is -16.432, and the results from the `optim` shows the MSE is the same as the coxph output ($\hat{\beta} = -1.276, se(\hat{\beta}) = 0.758$). the R codes for the computation are shown below:

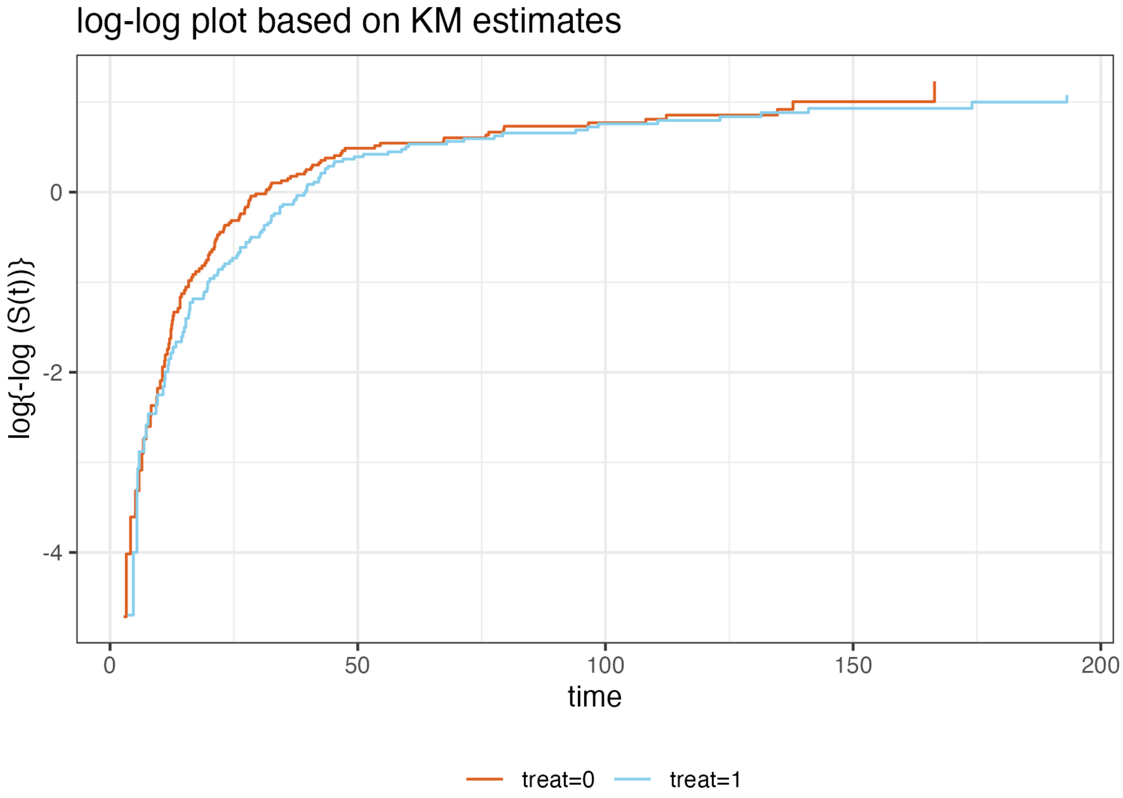
```
> ## y represents the at-risk
> y <- NULL
> for (i in c(1:12)) {
+   y[[i]] <- c(rep(0, i - 1), rep(1, 12 - i + 1))
+ }
>
> Q2_loglik <- NULL
> for (i in c(1:12)) {
+   Q2_loglik[[i]] <- (Q2_dat$status[i] * (beta * Q2_dat$arm[i]) -
Q2_dat$status[i]*log(sum(exp(beta * Q2_dat$arm) * y[[i]])))
+ }
> Q2_loglik %>%
+   unlist() %>%
+   sum()
[1] -16.43216
>
> Q2_loglik1 <- function(par) {
+   for (i in c(1:12)) {
+     Q2_loglik[[i]] <- (Q2_dat$status[i] * (par * Q2_dat$arm[i]) -
+ Q2_dat$status[i]* log(sum(exp(par * Q2_dat$arm) * y[[i]])))
+   }
+   Q2_loglik %>%
+     unlist() %>%
+     sum()
+ }
>
> inits <- 0
> maxim <- optim(inits, fn=Q2_loglik1, 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,] -1.276 0.758
```

difference in log-log plot (time points are not
corresponding to each other between
treat=0 & 1 groups

Q3

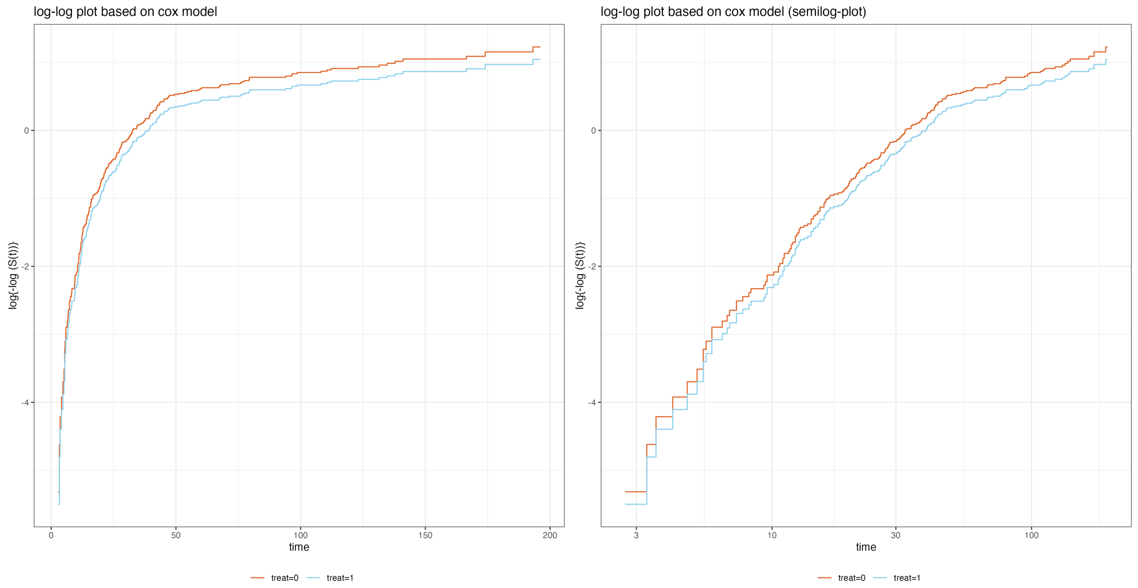
(a)Present a graphical check for proportionality in Cox models can be based on the result $\log[\log S_1(t)] - \log[\log S_0(t)] = \beta$, where the survival function is estimated with KM method.

The log-log plot is presented below (codes can be found in the appendix). From the plot, we see that the differences of the $\log(-\log(S(t))$ values between treat=0 (red) and treat =1 (blue) groups (i.e., β) are not constant over time i.e.lines are not parallel), and we see a slightly increase in the difference from week 13 to week 50 (approximately). For example, the $\beta \approx -0.3153756 - (-0.7627677) = 0.4473921$ in week 24, while for the rest time intervals, the two curves overlap, and β_s are very close to 0.:



(b)When fitting a Cox model to estimate the treatment effect, the survival functions in the two treatment groups may be estimated by $\hat{S}_0(t) = \exp\{-\hat{\Lambda}_0(t)\}$ and $\hat{S}_1(t) = \exp\{-\hat{\Lambda}_0(t)\exp(\hat{\beta})\}$, where $\hat{\Lambda}_0(t)$ is the Breslow estimate for the cumulative baseline hazard, returned by the `basehaz` R function. Present a log-log plot of such survival functions and interpret it. Why is this plot not useful for checking the proportionality assumption?

The log-log plot is presented below (codes are provided in the appendix)
Interpretation: The red curve is $\log(\hat{\Lambda}_0(t))$, and the blue curve is $\log(\hat{\Lambda}_0(t)) + \hat{\beta}$.so the difference of the two curves is $\hat{\beta}$. Overall speaking, the curves are



parallel, This is more obvious when we present time on the log scale (see the right panel the semi-log plot)

This plot is not useful for checking the proportionality assumption as the $\hat{S}_0(t)$ and $\hat{S}_1(t)$ are estiamtions from the Cox model where it already assumed proportionality of the hazard.

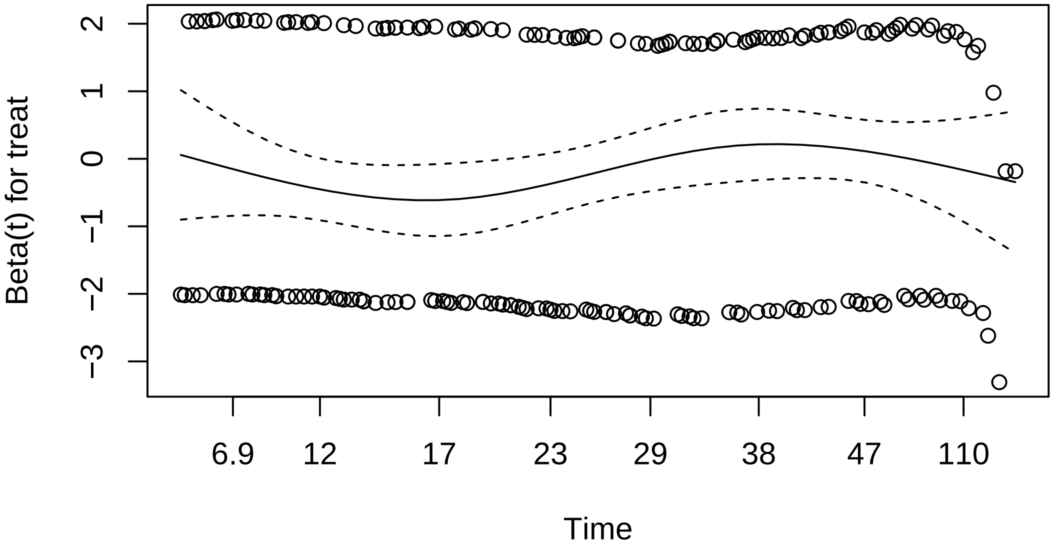
Q4

(a)Present and interpret an appropriate residual check for proportionality of the treatment effect.

The Schoenfeld residual can be used to check the proportionality of the treatment effect. The codes and results are shown below.

1. The residual plot shows the smoothed line is not straight and horizontal, instead, it fluctuates around zero.
2. However, the CIs do include zero (gray area), and the chi-squared test stats gives a P-value of 0.2615 for treat variable, indicating that the fluctuation is not statistically significant.
3. So we conclude that we see fluctuation in the residual plot, but the fluctuation is not statistically significant, we do not have statistical evidence to reject the null hypothesis that the PH assumption is hold.

```
fit_cox <- coxph(formula = Surv(weeks, event) ~ treat,data = brain)
fit.zph1 <- cox.zph(fit_cox)
plot(fit.zph1)
fit.zph1$table
```



```
> fit.zph1$table
          chisq df         p
treat  1.26104  1 0.2614542
GLOBAL 1.26104  1 0.2614542
```

(b)Proportionality can be tested also by adding an interaction term with the treatment arm indicator and time into the model using the `tt` argument of the coxph. Add this to the model and interpret the result.

The codes and model output are presented as follows. If the treatment effect does not change over time, the interactive effect should be zero. so to test the PH assumption, we can test whether the interactive effect is zero.

1. From the model output we see the interactive effect is 0.0014, with Wald test statistics of 0.280 and corresponding P-value of 0.779, indicating the interactive term is not statistically different from zero. Therefore, this is no evidence to reject the null hypothesis that the interactive effect between treat and time is different from zero. That is, we do not have statistical evidence to reject the null hypothesis that PH assumption is met.
2. the treatment effect is $\exp(-0.23) = 0.79$, but the effect is not of statistical significant(P-value:0.29).

```
> coxph(Surv(weeks, event) ~  tt(treat),
+       tt=function(x,t, ...) cbind(x, x * t),
+       data=brain)
Call:
coxph(formula = Surv(weeks, event) ~ tt(treat), data = brain,
    tt = function(x, t, ...) cbind(x, x * t))
                coef exp(coef)
tt(treat)x -0.230714  0.793966
tt(treat)   0.001429  1.001430
            se(coef)      z     p
tt(treat)x  0.217851 -1.059 0.290
tt(treat)   0.005094  0.280 0.779
Likelihood ratio test=1.8  on 2 df, p=0.4063
n= 222, number of events= 207
```

Q5

(a)Present and interpret a log-log plot of the baseline survival functions to check the proportionality assumption of the treatment effect in a model adjusted for age at the start of the treatment.

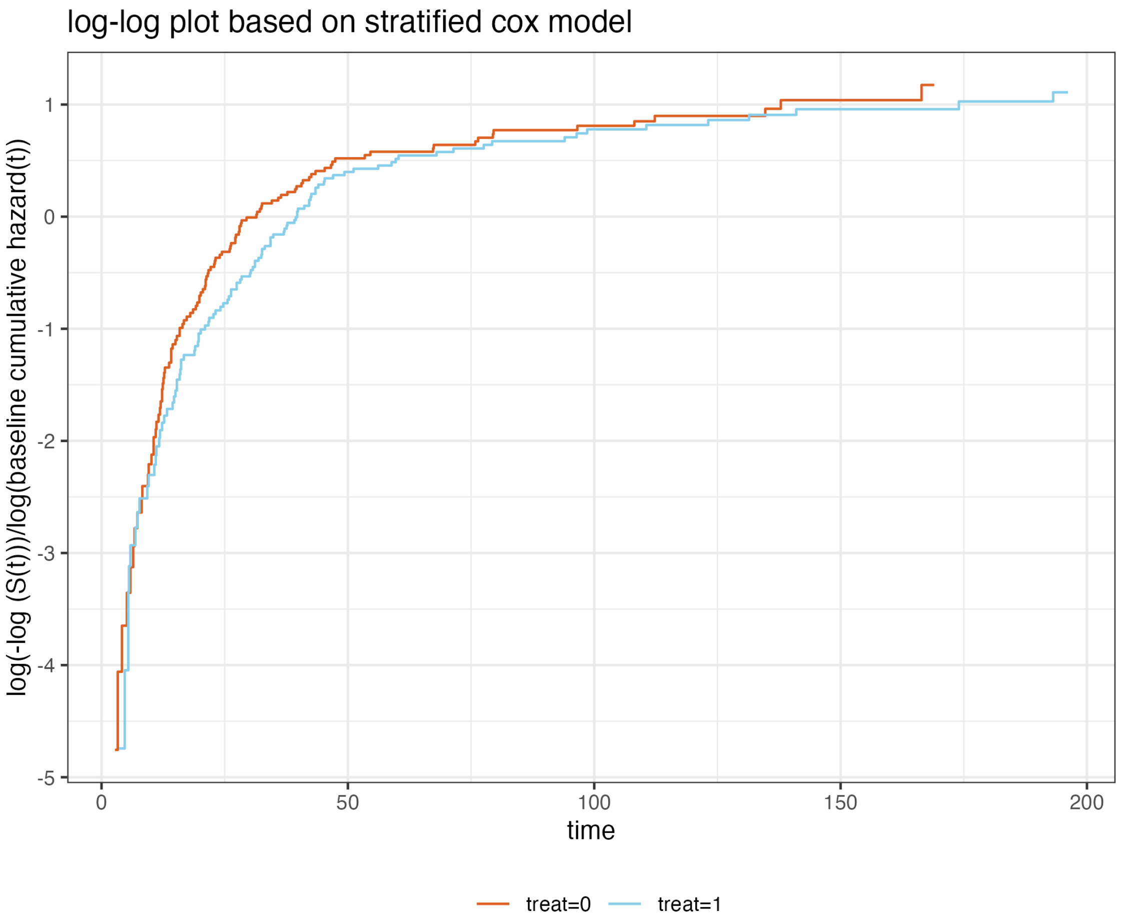
The stratified Cox models were fit (strata = treat) using the codes shown below.

```
fit_cox_Q5 <- coxph(formula = Surv(weeks, event)
~ strata(treat)+age,data = brain)

cum_basehaz_val_Q5 <- basehaz(fit_cox_Q5) %>%
  mutate(lnHt = log(hazard))
  ## s = exp(-Ht) log(-log(s)) = lnHt
```

The log-log plot based on the cumulative baseline hazard is presented below (codes are in the appendix). If the PH assumption holds, the baseline cumulative hazard rates in the two strata should be a constant multiple, that is, the two curves should be parallel.

1. From the plot, we see that the curves are not parallel in some time intervals (around 13 weeks to 50 weeks), and in the remaining time intervals, the two curves are almost overlap.
2. So we conclude that the PH assumption is met in some of the time intervals, but hazard is not always a constant over all time intervals.



(b)Fit a Cox model to estimate the treatment effect adjusting for the age at the start of the treatment. Present and interpret an appropriate residual check for log-linearity of the age effect.

Two models are fitted, which are:

$$\lambda_i(t) = \lambda_0(t)\exp(\beta_1 * treat + \beta_2 * age)$$

and

$$\lambda_i(t) = \lambda_0(t)\exp(\beta_1 * treat + \beta_2 * log(age))$$

The model fit results are shown as follows:

```
>  coxph(formula = Surv(weeks, event) ~ treat+age,data = brain)
Call:
coxph(formula = Surv(weeks, event) ~ treat + age, data = brain)

           coef  exp(coef)
treat -0.216585  0.805264
age    0.022154  1.022401
       se(coef)      z       p
```

```
treat  0.140220 -1.545    0.122
age    0.005583  3.968 7.24e-05

Likelihood ratio test=17.44  on 2 df, p=0.0001636
n= 222, number of events= 207

## log(age)
fit_cox_Q5_b_logage <- coxph(formula = Surv(weeks, event) ~ treat + log(age),
                          data = brain)
fit_cox_Q5_b_age
Call:
coxph(formula = Surv(weeks, event) ~ treat + log(age), data = brain)

            coef exp(coef)
treat    -0.2212    0.8016
log(age)  0.9617    2.6162
         se(coef)      z
treat      0.1402 -1.577
log(age)   0.2581  3.726
                p
treat    0.114801
log(age) 0.000194

Likelihood ratio test=16.16  on 2 df, p=0.0003102
n= 222, number of events= 207
```

To check the log-linearity of the age effect, we can use martingale residual, which is used to check the function form of continuous covariates. The codes and residual plots are shown below.

1. The first column shows the residual plots for age, we see that in the original form, the smooth fitted curve is not linear, indicating the relationship between age and the outcome is not linear. Some transformation of age should be applied. Additionally, we see most of the residuals are randomly scattered around zero, and the confidence intervals also include zero.
2. The second column presents the martingale residuals for log(age).The plots in the second column show the martingale residual plots after the log transformation, the smooth fitted line shows non-linearity of the log(age) as it is not straight, the residuals are also randomly scattered around zero. the two function forms of age have similar results regarding the martingale residual.
3. Though the smoothed curves in two models are not strictly straight, I would argue that the two curves are quite flat (as this is the data from

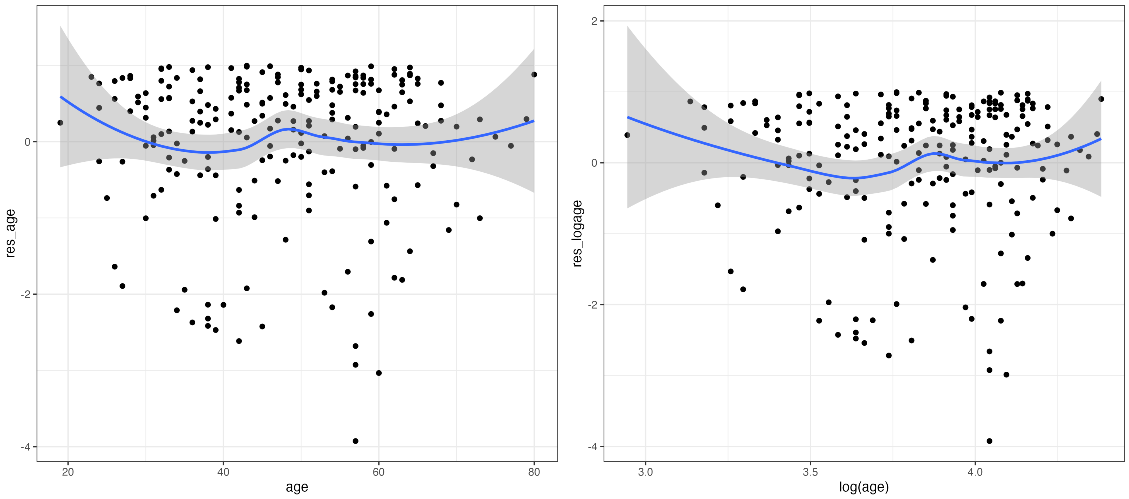
real world, it is acceptable to have some fluctuations), so I would say the function form of age or log(age) are both appropriate.

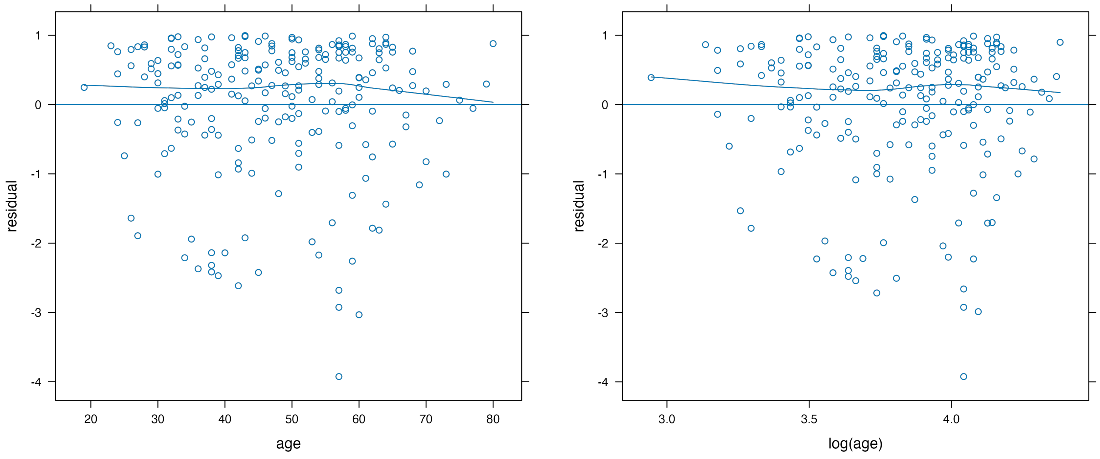
```
fit_cox_Q5_b_age <- coxph(formula = Surv(weeks, event) ~ treat+age,
data = brain)


brain$res_age <- resid(fit_cox_Q5_b_age)

fit_cox_Q5_b_logage <- coxph(formula = Surv(weeks, event) ~ treat + log(age),
                          data = brain)

brain$res_logage <- resid(fit_cox_Q5_b_logage)
```

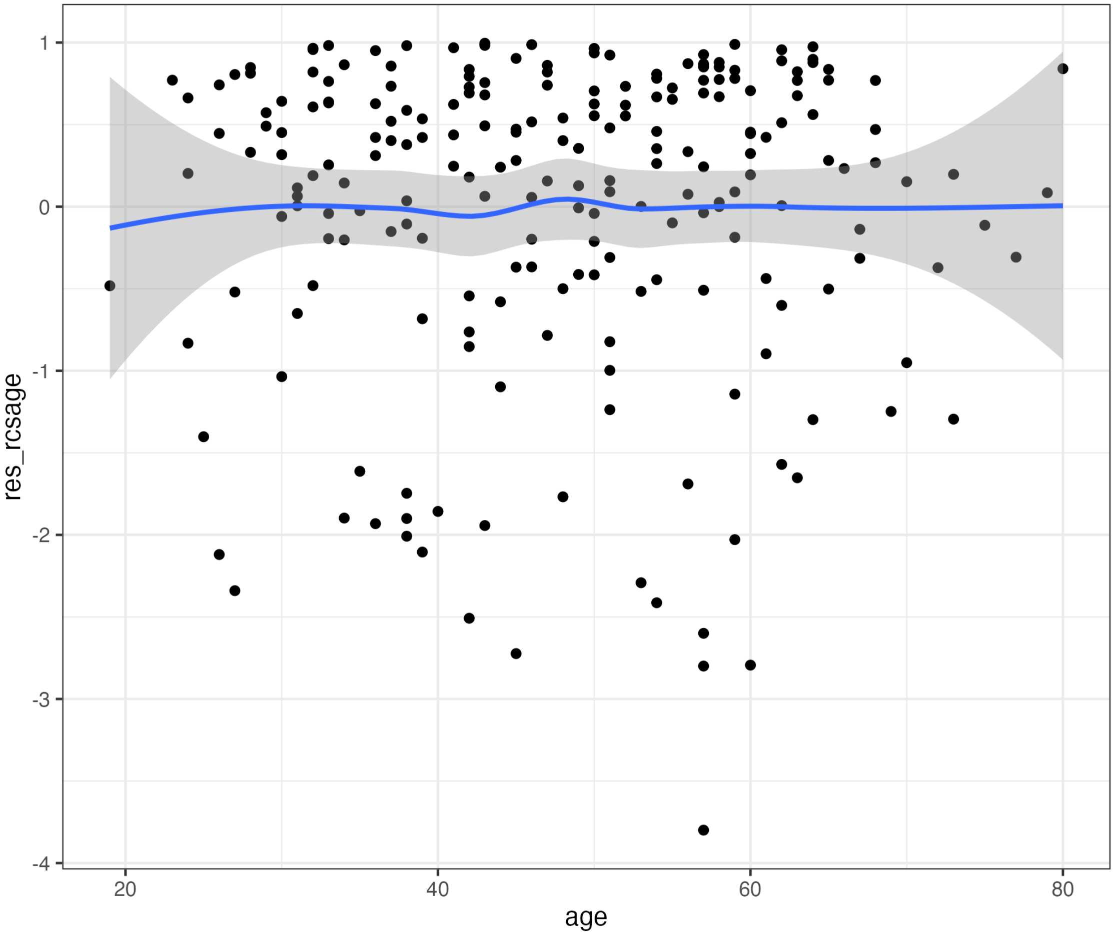




Additionally, we can try to use restricted cubic spline (rcs) to capture the non-linearity of age effect. The martingale residual plot using this rcs transformation is as follows, and we see the smoothed line is straight and horizontal, and is around zero. so the rcs transformation with 5 knot is an appropriate form for the age variable. (but we need to be cautious as it may lead to overfitting)

```
library(rms)
fit_cox_Q5_b_rcs <- cph(Surv(weeks, event) ~ treat+ rcs(age, 5) ,
data = brain, x = TRUE, y = TRUE)
brain$res_rcsage <- resid(fit_cox_Q5_b_rcs)

ggplot(data = brain, aes(x=age, y=res_rcsage))+
  geom_point() + stat_smooth()+theme_bw()
```



(c)Age could be added to the model also as a time-dependent covariate, to take into account that the patients are aging during the follow-up. Do this using the tt argument of the coxph function (please show your function call). What happened to the age effect estimate compared to the model with the fixed baseline age variable? Explain why.

Since age changes over the follow-up period, we can define it as a time-dependent variable using the `tt` function call. The `x` argument in `tt` is a placeholder for the covariate variable, in our case, x is age (in years), the `t` represents the follow-up time, as the time unit of the original follow-up time is in weeks, we then divide `t` by (365/7) to get age (in years) at each time point during the follow-up. The codes are presented below:

```
>  coxph(formula = Surv(weeks, event) ~ treat + tt(age),
+                         tt=function(x,t, ...) (x+t/(365/7)),
+                             data = brain)
Call:
coxph(formula = Surv(weeks, event) ~ treat + tt(age), data = brain,
    tt = function(x, t, ...) (x + t/(365/7)))
```

365.25/7

```
             coef exp(coef)
treat   -0.216585  0.805264
tt(age)  0.022154  1.022401
         se(coef)      z
treat    0.140220 -1.545
tt(age)  0.005583  3.968
                p
treat       0.122
tt(age) 7.24e-05

Likelihood ratio test=17.44  on 2 df, p=0.0001636
n= 222, number of events= 207
```

The age effect estimate is the same compared to the model with the fixed baseline age variable ($\beta = 0.022$, $HR(t) = \exp(\beta) = 1.022$). This is because cox model still assumes the covariate effects to be constant over time, even though the covariates may change/be time-dependent. This is also reflected on the specification of cox model with time-dependent variables and the partial likelihood. The cox model with time-dependent variable is specified as:

$$\lambda_i(t) = \lambda_0(t)\exp(\beta Z_i(t))$$

where β is a constant for Z at different ts. The partial likelihood is specified as:

$$L(\beta) \propto \prod_{i=1}^{n}\left(\frac{\exp(\beta \times treat_i + \gamma(age_i + t/(365/7)))}{\sum_{j=1}^{n} Y_j(t_i)\exp(\beta \times treat_j + \gamma(age_j + t/(365/7)))}\right)^{e_i}$$

which can be simplified as:

$$
\begin{aligned}
L(\beta) &\propto \prod_{i=1}^{n}\left(\frac{\exp(\beta \times treat_i + \gamma age_i)\exp(\gamma(t/(365/7)))}{\exp(\gamma(t/(365/7)))\{\sum_{j=1}^{n} Y_j(t_i)\exp(\beta \times treat_j + \gamma age_j)\}}\right)^{e_i} \\
&= \prod_{i=1}^{n}\left(\frac{\exp(\beta \times treat_i + \gamma age_i)}{\sum_{j=1}^{n} Y_j(t_i)\exp(\beta \times treat_j + \gamma age_j)}\right)^{e_i}
\end{aligned}
$$

We see that t does not contribute to the partial likelihood, and the likelihood function treating age as a time-dependent variable is the same as that when we model age with the baseline value.

Appendix codes

```
## Q1
library(eha)
library(survival)

data(mort)
fit <- phreg(Surv(enter, exit, event) ~ ses, data = mort)
fit
plot(fit)

# Poisson model (PH parametrization):

model0ph <- glm(status ~ trt, offset=log(time),
                family=poisson(link=log), data=veteran)
summary(model0ph)


# Weibull model (PH parametrization, package eha):

library(eha)
model1ph <- phreg(Surv(time, status) ~ trt, dist = "weibull", data=veteran,shape
summary(model1ph)
c <- model1ph$coefficients

# Weibull model (AFT parametrization):

model1aft <- survreg(Surv(time, status) ~ trt, dist="weibull", data=veteran)
summary(model1aft)

exp(-0.642)
exp(-0.744)

exp(-0.642 - 0.744)/exp(-0.642)
```

```
###
observed_t <- c(1,2.5,3,4,4.5,5,0.5,0.75,
                1.25,1.5,2,3.5)
status <- c(1,1,1,1,1,0,1,1,1,0,1,1)
arm <- c(rep(1,6),rep(0,6))

alpha <- 0.642
beta <- -0.744

sum((status*(-alpha+beta*arm))-
      exp(-alpha+beta*arm)*observed_t)


loglik <- function(par) {
  return(sum((status*(-par[1]+par[2]*arm))-
               exp(-par[1]+par[2]*arm)*observed_t))
}
library(dplyr)
inits <- rep(0.0, 2)
maxim <- optim(inits, fn=loglik, control=list(fnscale=-1), method=’BFGS’, hessia
mles <- maxim$par %>% round(3)
ses <- sqrt(diag(solve(-(maxim$hessian)))) %>% round(3)
cbind(mles, ses)

############Q2################

observed_t <- c(
  1, 2.5, 3, 4, 4.5, 5, 0.5, 0.75,
  1.25, 1.5, 2, 3.5
)
status <- c(1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1)
arm <- c(rep(1, 6), rep(0, 6))

beta <- -1.2760
Q2_dat <- data.frame(
  observed_t = observed_t,
  status = status,
  arm = arm
) %>%
  arrange(observed_t) %>%
  mutate(
    seq_id = c(1:12),
    z = beta * arm,
    first_part = status * z,
    exp_z = exp(z)
  )

beta <- -1.2760

exp(beta * arm)

## y represents the at-risk
y <- NULL
for (i in c(1:12)) {
  y[[i]] <- c(rep(0, i - 1), rep(1, 12 - i + 1))
}

Q2_loglik <- NULL
for (i in c(1:12)) {
  Q2_loglik[[i]] <- (Q2_dat$status[i] * (beta * Q2_dat$arm[i]) - Q2_dat$status[i
  log(sum(exp(beta * Q2_dat$arm) * y[[i]])))
}
Q2_loglik %>%
  unlist() %>%
  sum()

#[1] -16.43216

Q2_loglik1 <- function(par) {
  for (i in c(1:12)) {
    Q2_loglik[[i]] <- (Q2_dat$status[i] * (par * Q2_dat$arm[i]) - Q2_dat$status[
                         log(sum(exp(par * Q2_dat$arm) * y[[i]])))
  }
  Q2_loglik %>%
    unlist() %>%
    sum()
}

inits <- 0
maxim <- optim(inits, fn=Q2_loglik1, control=list(fnscale=-1), method='BFGS', he
mles <- maxim$par %>% round(3)
ses <- sqrt(diag(solve(-(maxim$hessian)))) %>% round(3)
cbind(mles, ses)

###################Q3########################
brain <- read.csv("~/DLSPH/CHL5209/Assignments/HW3/data/brain.csv")
## a----
library(survival)
library(dplyr)
```

```
fit_treat <- survfit(Surv(weeks,event)~treat,
                     data = brain)

Q3_plot_a_dat<- data.frame(time = summary(fit_treat)$time,
                       treatment = as.factor(summary(fit_treat)$strata),
           surv_p = summary(fit_treat)$surv
           ) %>%
  mutate(loglog_val = log(-log(surv_p)),
         logtime = log(time))



library(ggplot2)
Q3_plota <- Q3_plot_a_dat %>%
  ggplot(aes(x = time, y = loglog_val,color = treatment)) +
  geom_step()+
  scale_color_manual(values = c("#DD5F20", "skyblue"))+
  theme_bw()+
  theme(legend.position = "bottom",
        legend.title = element_blank())+
  ylab("log{-log (S(t))}")+
  ggtitle('log-log plot based on KM estimates')

Q3_plota_logtime <- Q3_plot_a_dat %>%
  ggplot(aes(x = logtime, y = loglog_val,color = treatment)) +
  geom_step()+
  scale_color_manual(values = c("#DD5F20", "skyblue"))+
  theme_bw()+
  xlab('log(time)')+
  theme(legend.position = "bottom",
        legend.title = element_blank())+
  ylab("log{-log (S(t))}")+
  ggtitle('log-log plot based on KM estimates')


ggsave(Q3_plota,width = 5.96,
       height=4.4,
       file='Q3_plota.png')

## b-----

fit_cox <- coxph(formula = Surv(weeks, event) ~ treat,data = brain)

fit_cox$coefficients[1]
cum_basehaz_val <- basehaz(fit_cox)
```

```
time_weeks <-  rep(cum_basehaz_val$time,2)

Q3_plot_b_dat <- data.frame(time = time_weeks,
                            treatment = c(rep("treat=0",length(cum_basehaz_val$ti
                            surv_p = c(exp(-cum_basehaz_val$hazard),
                                       exp(-cum_basehaz_val$hazard*exp(fit_cox$co
                            ) %>%
  mutate(loglog_cox = log(-log(surv_p)))

Q3_plotb_semilog <- Q3_plot_b_dat %>%
  ggplot(aes(x = time, y = loglog_cox,color = treatment)) +
  geom_step()+
  scale_color_manual(values = c("#DD5F20", "skyblue"))+
  theme_bw()+
  theme(legend.position = "bottom",
        legend.title = element_blank())+
  scale_x_continuous(trans = "log10")+
  ylab("log{-log (S(t))}")+
ggtitle(’log-log plot based on cox model (semilog-plot)’)

Q3_plotb <- Q3_plot_b_dat %>%
  ggplot(aes(x = time, y = loglog_cox,color = treatment)) +
  geom_step()+
  scale_color_manual(values = c("#DD5F20", "skyblue"))+
  theme_bw()+
  theme(legend.position = "bottom",
        legend.title = element_blank())+
  ylab("log{-log (S(t))}")+
  ggtitle(’log-log plot based on cox model’)

ggsave(Q3_plotb_semilog,width = 5.96,
       height=4.4,
       file=’Q3_plotb_semilog.png’)

library(ggpubr)
Q3_b_plots <- ggarrange(Q3_plotb,Q3_plotb_semilog)

ggsave(Q3_b_plots,width = 16,
       height=8.4,
       file=’Q3_b_plots.png’)
```

```
#################Q4####################
fit_cox <- coxph(formula = Surv(weeks, event) ~ treat,
                 data = brain)

fit.zph1 <- cox.zph(fit_cox)

fit.zph1$table
summary(fit.zph1)
plot(fit.zph1)

### b------
coxph(Surv(weeks, event) ~  tt(treat),
      tt=function(x,t, ...) cbind(x, x * t),
      data=brain)

## see week 7 .r file
#############Q5#################
fit_cox_Q5 <- coxph(formula = Surv(weeks, event) ~ strata(treat)+age,
                 data = brain)


cum_basehaz_val_Q5 <- basehaz(fit_cox_Q5) %>%
  mutate(lnHt = log(hazard)) ## s = exp(-Ht) log(-log(s)) = lnHt

Q5_plota <- cum_basehaz_val_Q5 %>%
  ggplot(aes(x = time, y = lnHt,color = strata)) +
  geom_step()+
  scale_color_manual(values = c("#DD5F20", "skyblue"))+
  theme_bw()+
  theme(legend.position = "bottom",
        legend.title = element_blank())+
  ylab("log(-log (S(t)))/log(baseline cumulative hazard(t))")+
  ggtitle(’log-log plot based on stratified cox model’)

ggsave(Q5_plota,width = 6.69,
       height=5.62,
       file=’Q5_plota.png’)

### b --------

fit_cox_Q5_b_age <- coxph(formula = Surv(weeks, event) ~ treat+age,data = brain)

brain$res_age <- resid(fit_cox_Q5_b_age)

Q5_b_plot_age <- ggplot(data = brain, aes(x=age, y=res_age)) +
  geom_point() +
  stat_smooth() +# shows non-linearity
  theme_bw()
library(lattice)


ggsave(Q5_c_plot_age,width = 6.69,
       height=5.62,
       file='Q5_c_plot_age.png')


fit_cox_Q5_b_logage <- coxph(formula = Surv(weeks, event) ~ treat + log(age),
                          data = brain)

brain$res_logage <- resid(fit_cox_Q5_b_logage)

Q5_b_plot_logage <- ggplot(data = brain, aes(x=log(age), y=res_logage))+
geom_point() + stat_smooth()+theme_bw() # shows non-linearity

fit_cox_Q5_b_rcs <- cph(Surv(weeks, event) ~ treat+ (rcs(age, 5) ),
                        data = brain, x = TRUE, y = TRUE)
brain$res_rcsage <- resid(fit_cox_Q5_b_rcs)

Q5_b_plot_rcsage <- ggplot(data = brain, aes(x=age, y=res_rcsage))+
  geom_point() + stat_smooth()+theme_bw()

ggsave(Q5_b_plot_rcsage,width = 6.69,
       height=5.62,
       file='Q5_b_plot_rcsage.png')

residual_rcsage_plot <- xyplot(resid(fit_cox_Q5_b_rcs)~brain$age,
                            type=c("p","r","smooth"),
                            xlab='age',
                            ylab='residual')

residual_age_plot <- xyplot(resid(fit_cox_Q5_b_age)~brain$age,
       type=c("p","r","smooth"),
       xlab='age',
       ylab='residual') # shows departure from linearity

residual_logage_plot <- xyplot(resid(fit_cox_Q5_b_logage)~ log(brain$age),
       type=c("p","r","smooth"),
       xlab="log(age)",
       ylab='residual') # shows departure from linearity
```

```
par(mfrow= c(1,1))
ggsave(Q5_b_plot_logage,width = 6.69,
       height=5.62,
       file='Q5_c_plot_logage.png')

library(ggpubr)
Q5_b_plot <- ggarrange(Q5_b_plot_age,
                       Q5_b_plot_logage
                       )

Q5_b_plot_1 <- ggarrange( residual_age_plot,
                       residual_logage_plot
)

ggsave(Q5_b_plot,width = 12.69,
       height=5.62,
       file='Q5_b_plot.png')

ggsave(Q5_b_plot_1,width = 12.69,
       height=5.62,
       file='Q5_b_plot_1.png')

#### c-------
lung
fit_cox_Q5_c <- coxph(formula = Surv(weeks, event) ~ treat + tt(age),
                      tt=function(x,t, ...) (x+t/(365/7)),
                          data = brain)

fit_cox_Q5_c
coxph(formula = Surv(weeks, event) ~ treat + age,
      data = brain)

coxph(formula = Surv(weeks, event) ~ treat + tt(age),
      tt=function(x,t, ...) x+t,
      data = brain)

coxph(formula = Surv(weeks, event) ~ treat + tt(age),
      data = brain)
```