Examining the Effects of Censoring Rate on Survival Models: A Simulation Study

Jon Cokisler

21 March 2025

In this simulation study, I am going to examine the Cox Proportional Hazards Model under certain experimental design, and censoring conditions. The story behind the simulations: The simulated data come from an imaginary study examining the effectiveness of Nicotine Replacement Therapy in quitting smoking. The studies are Randomized Control Trials, where the participants are randomly assigned to either the placebo group, or treatment group. The placebo group receives a placebo of the nicotine gum, where as the treatment group receives actual nicotine gum. The participants in this study are smokers, and they are asked to stop smoking as soon as they are assigned their treatment group and given the treatment. The outcome variable in this study is days until relapse (consumption of a cigarette), where the origin event is the start of the experiment phase, where the participants are given the treatment/placebo, and the end event is the first relapse for the participants. There will be two focuses in this study:

1. Effect of censoring rate on Cox PH model, in a Randomized Control Trial setting, with a single factor, treatment group (treatment/placebo).
2. Effect of censoring rate on Cox PH model, in an RCT setting, with two factors:

- Treatment group: Treatment / Placebo
- Number of cigarettes consumed per day

The study is 4 months (120 days) long.

Simulation 1: Placebo-Treatment (Nicotine Replacement Therapy) Simulation - Days to Relapse, One covariate

Data Generation Process

We are looking to generate T values, where T is the days until first relapse. To accomplish this, we are assuming an underlying Weibull data generating process, by sampling from the Weibull distribution.

We want to generate the data with some common sense, so I converted the `rweibull()` parameters into what we used in class for our AFT model. The base R Weibull sampling function `rweibull()` has 3 parameters:

- n: number of observations
- shape: shape -> κ
- scale: scale -> λ

Turns out, these parameters translate into our parameterization of the AFT model in the following way:

- shape: $\kappa = \frac{1}{\tau}$
- scale: $\lambda = \exp(\beta_0 + \beta_1 z_1)$

Therefore, for our data generating process, we will use the following: rbinom(n, shape=τ, scale=$\exp(\beta_0 + \beta_1 z_1)$)

```r
set.seed(11)

n <- 500

n_treatment <- n / 2
n_placebo <- n / 2

shape <- 0.8      # Shape parameter (Weibull) -> 1/scale = Tau
```

```
scale_base <- 3     # Baseline scale parameter -> intercept
beta <- 0.7       # Treatment effect = Beta

# Simulate survival times for placebo group
scale_placebo <- exp(scale_base)
T_placebo <- rweibull(n_placebo, shape = shape, scale = scale_placebo)

# Simulate survival times for treatment group
scale_treatment <- exp(scale_base)* exp(beta)
T_treatment <- rweibull(n_treatment, shape = shape, scale = scale_treatment)


time <- c(T_placebo, T_treatment)
status <- rep(1, n)
group <- rep(c("Placebo", "Treatment"), each = n/2)
```

Verifying The Data Generation

Verifying that the data generation process is correct, and that the parameters we input into the Weibull sampling function, is crucial. Thankfully, it is also relatively easy to check. We can fit a Weibull AFT model using our simulated data, and if the estimates outputted by the model are close to our starting parameters, then we know our data generating process worked correctly.

```
aft <- survreg(Surv(time, status) ~ factor(group), data=tibble(time, status,
                                                               group),
               dist = "weibull")
summary(aft)
```

```
##
## Call:
## survreg(formula = Surv(time, status) ~ factor(group), data = tibble(time,
##     status, group), dist = "weibull")
##                          Value Std. Error     z       p
## (Intercept)            3.0601     0.0748 40.91 < 2e-16
## factor(group)Treatment 0.6404     0.1029  6.22 4.9e-10
## Log(scale)             0.1405     0.0349  4.02 5.8e-05
##
## Scale= 1.15
##
## Weibull distribution
## Loglik(model)= -2216.5   Loglik(intercept only)= -2234.9
##  Chisq= 36.69 on 1 degrees of freedom, p= 1.4e-09
## Number of Newton-Raphson Iterations: 6
## n= 500
```

We can see from the output of the AFT model, $\hat{\beta}_0 = 3.05 \approx 3$, which was our input, represented in the variable `scale_base`. The `scale` value from the AFT model output is 1.16, and when we take its inverse, as we input into the sampling function, $\frac{1}{\hat{\tau}} = \frac{1}{1.16} = 0.86 \approx 0.8$. The $\hat{\beta}_1$ is also close to our original input: $\hat{\beta}_1 = 0.65 \approx 0.7$. Therefore, we can conclude that our data generating process, and our characterizations are correct, since the AFT model estimates align with our original inputs.

Censoring

The censoring being simulated here has two components. 1. Type I Right Censoring - The simulated study takes place in a 120 day period, therefore the subjects that have not relapsed by the 120-day mark, are censored. 2. Random Censoring - Random censoring due to non-study determined factors. We are simulating this type of censoring by sampling from a normal distribution, with various different parameter values to achieve different censoring rates.

```
set.seed(10)
# Censoring Rate = 0.4
c_values <- rtruncnorm(n, a=0, mean=20, sd=30)
```

```r
time <- c(T_placebo, T_treatment)

status <- rep(NA, n)

for(i in 1:n){
  if(time[i] > 120){
    status[i] <- 0
    time[i] <- 120
  }
  else if(c_values[i] < time[i]){
    status[i] <- 0
  }
  else{
    status[i] <- 1
  }
}

df_40_percent <- data.frame(time, status, group)
paste("This censoring rate is:", 100* (1 - (sum(status) /n)), "%")
```

```
## [1] "This censoring rate is: 40.2 %"
```

```r
#-------------------------------------------------------------------
# Censoring Rate = 0.2
c_values <- rtruncnorm(n, a=0, mean=60, sd=30)
time <- c(T_placebo, T_treatment)

status <- rep(NA, n)

for(i in 1:n){
  if(time[i] > 120){
    status[i] <- 0
    time[i] <- 120
  }
  else if(c_values[i] < time[i]){
    status[i] <- 0
  }
  else{
    status[i] <- 1
  }
}

df_20_percent <- data.frame(time, status, group)
paste("This censoring rate is:", 100* (1 - (sum(status) /n)), "%")
```

```
## [1] "This censoring rate is: 20.4 %"
```

```r
#-------------------------------------------------------------------
# Censoring Rate = 0.1
c_values <- rtruncnorm(n, a=0, mean=90, sd=30)
time <- c(T_placebo, T_treatment)

status <- rep(NA, n)

for(i in 1:n){
  if(time[i] > 120){
    status[i] <- 0
    time[i] <- 120
  }
  else if(c_values[i] < time[i]){
    status[i] <- 0
  }
  else{
    status[i] <- 1
  }
}

df_10_percent <- data.frame(time, status, group)
paste("This censoring rate is:", 100* (1 - (sum(status) /n)), "%")
```

```
## [1] "This censoring rate is: 10 %"
```

```
#--------------------------------------------------------------------
# Censoring Rate = 0 (Only Type I Right Censor present)
time <- c(T_placebo, T_treatment)

status <- rep(NA, n)

for(i in 1:n){
  if(time[i] > 120){
    status[i] <- 0
    time[i] <- 120
  }
  else{
    status[i] <- 1
  }
}

df_0_percent <- data.frame(time, status, group)
paste("This censoring rate is:", 100* (1 - (sum(status) /n)), "%")
```

```
## [1] "This censoring rate is: 4.4 %"
```

Cox Proportional Hazards Model Fitting and Evaluation

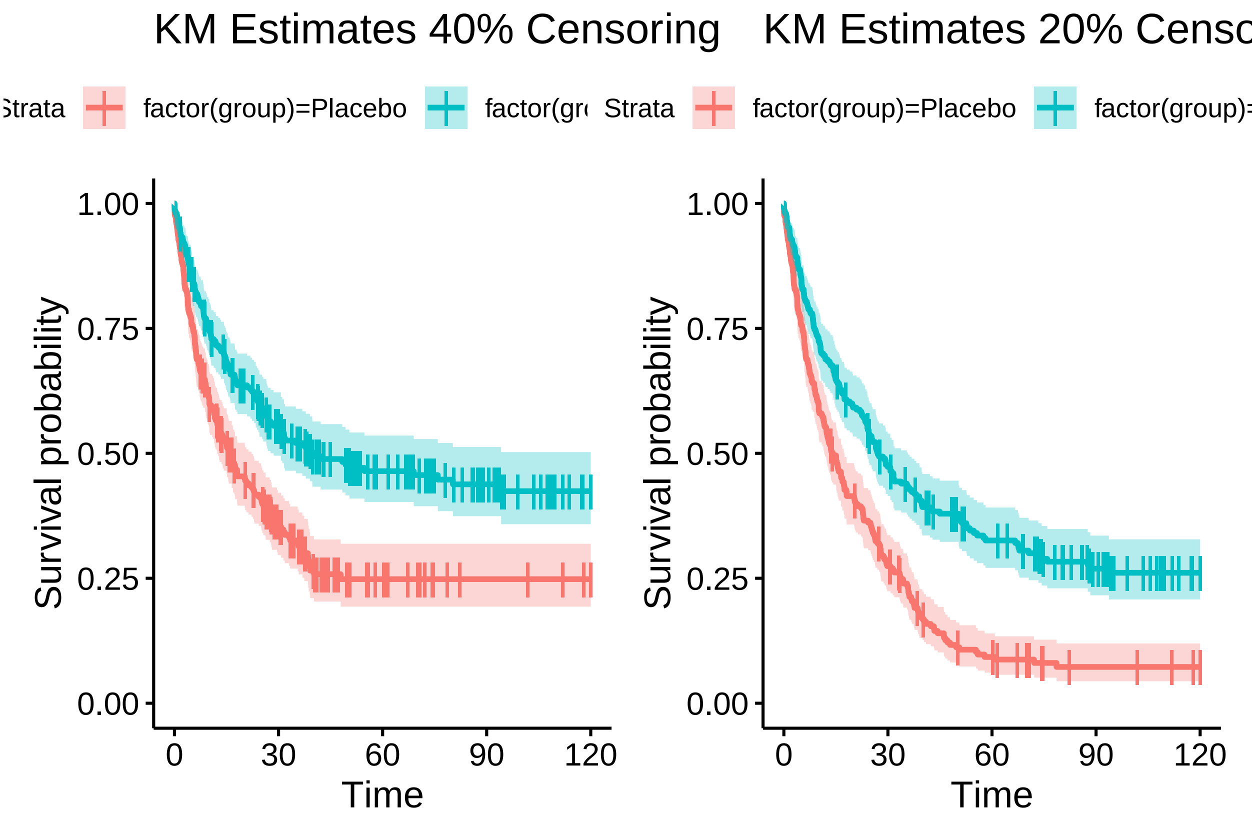
```
km_40_pct_censor <- surv_fit(Surv(time, status) ~ factor(group),
                             data=df_40_percent)
km_20_pct_censor <- surv_fit(Surv(time, status) ~ factor(group),
                             data=df_20_percent)
km_10_pct_censor <- surv_fit(Surv(time, status) ~ factor(group),
                             data=df_10_percent)
km_0_pct_censor <- surv_fit(Surv(time, status) ~ factor(group),
                             data=df_0_percent)
```

Fitting the models

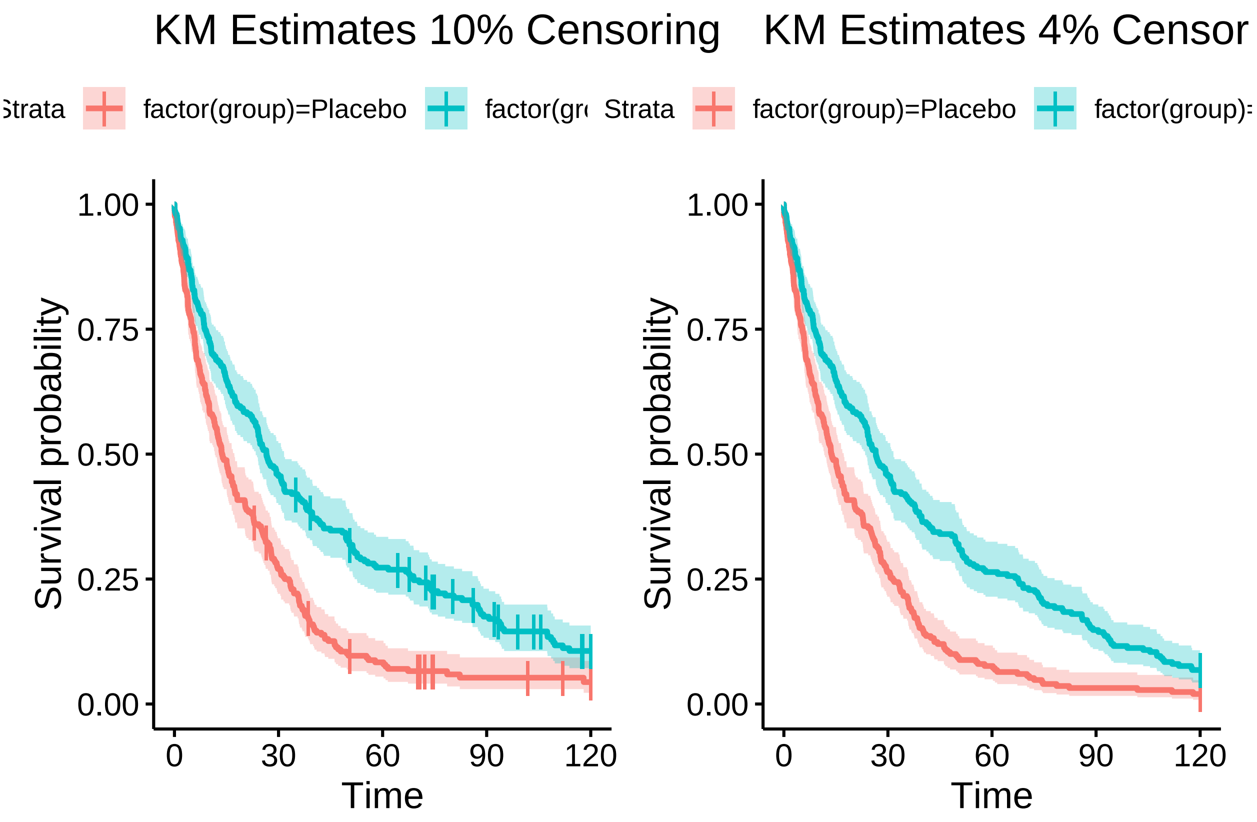
```
pl1 <- ggsurvplot(km_40_pct_censor, title = "KM Estimates 40% Censoring",
                  conf.int = TRUE, censor.shape = "|")

pl2 <- ggsurvplot(km_20_pct_censor, title = "KM Estimates 20% Censoring",
                  conf.int = TRUE, censor.shape = "|")

pl3 <- ggsurvplot(km_10_pct_censor, title = "KM Estimates 10% Censoring",
                  conf.int = TRUE, censor.shape = "|")

pl4 <- ggsurvplot(km_0_pct_censor, title = "KM Estimates 4% Censoring",
                  conf.int = TRUE, censor.shape = "|")
```

```
pl1$plot + pl2$plot
```

Visualizing Survival Probabilities



```
pl3$plot + pl4$plot
```



Even from these plots alone, we can see how much of an effect the censoring rate has on the shape, and confidence intervals of the survival curves. We can clearly see the data with the highest censoring rate (40%) has a flatter right tail, and does not accurately capture the survival probabilities from the data, and it also has very large confidence bands, which means the KM estimates have high variance.

However, as the censoring rate decreases, and when we look at the data with the lowest censoring rates(10% and 4%), we can see that the KM estimates have much lower variance, slimmer confidence bands, and the right tails more accurately capture the actual survival probabilities.

The survival curves suggest that the two treatment groups have considerably different survival probabilities. We can perform a logrank test to see if they are significantly different.

```
logrank <- survdiff(Surv(time, status) ~ factor(group),
                             data=df_40_percent)
logrank
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ factor(group), data = df_40_percent)
##
##                            N Observed Expected (O-E)^2/E (O-E)^2/V
## factor(group)=Placebo   250      171      130      13.1      23.6
## factor(group)=Treatment 250      128      169      10.1      23.6
##
##  Chisq= 23.6  on 1 degrees of freedom, p= 1e-06
```

The logrank test suggests that two treatment groups's survival curves are statistically significantly different, since $p < 0.05$.

```
cox_model1 <- coxph(Surv(time, status) ~ factor(group), data=df_40_percent)
summary(cox_model1)
```

Cox PH models

```
## Call:
```

```
## coxph(formula = Surv(time, status) ~ factor(group), data = df_40_percent)
##
##   n= 500, number of events= 299
##
##                           coef exp(coef) se(coef)      z Pr(>|z|)
## factor(group)Treatment -0.5659    0.5679   0.1180 -4.796 1.62e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                        exp(coef) exp(-coef) lower .95 upper .95
## factor(group)Treatment    0.5679      1.761    0.4506    0.7156
##
## Concordance= 0.57  (se = 0.015 )
## Likelihood ratio test= 23.34  on 1 df,   p=1e-06
## Wald test            = 23  on 1 df,   p=2e-06
## Score (logrank) test = 23.59  on 1 df,   p=1e-06
```

```
cox_model2 <- coxph(Surv(time, status) ~ factor(group), data=df_20_percent)
summary(cox_model2)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ factor(group), data = df_20_percent)
##
##   n= 500, number of events= 398
##
##                           coef exp(coef) se(coef)      z Pr(>|z|)
## factor(group)Treatment -0.6256    0.5349   0.1027 -6.093 1.11e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                        exp(coef) exp(-coef) lower .95 upper .95
## factor(group)Treatment    0.5349      1.869    0.4374    0.6542
##
## Concordance= 0.57  (se = 0.013 )
## Likelihood ratio test= 37.52  on 1 df,   p=9e-10
## Wald test            = 37.13  on 1 df,   p=1e-09
## Score (logrank) test = 38.25  on 1 df,   p=6e-10
```

```
cox_model3 <- coxph(Surv(time, status) ~ factor(group), data=df_10_percent)
summary(cox_model3)
```

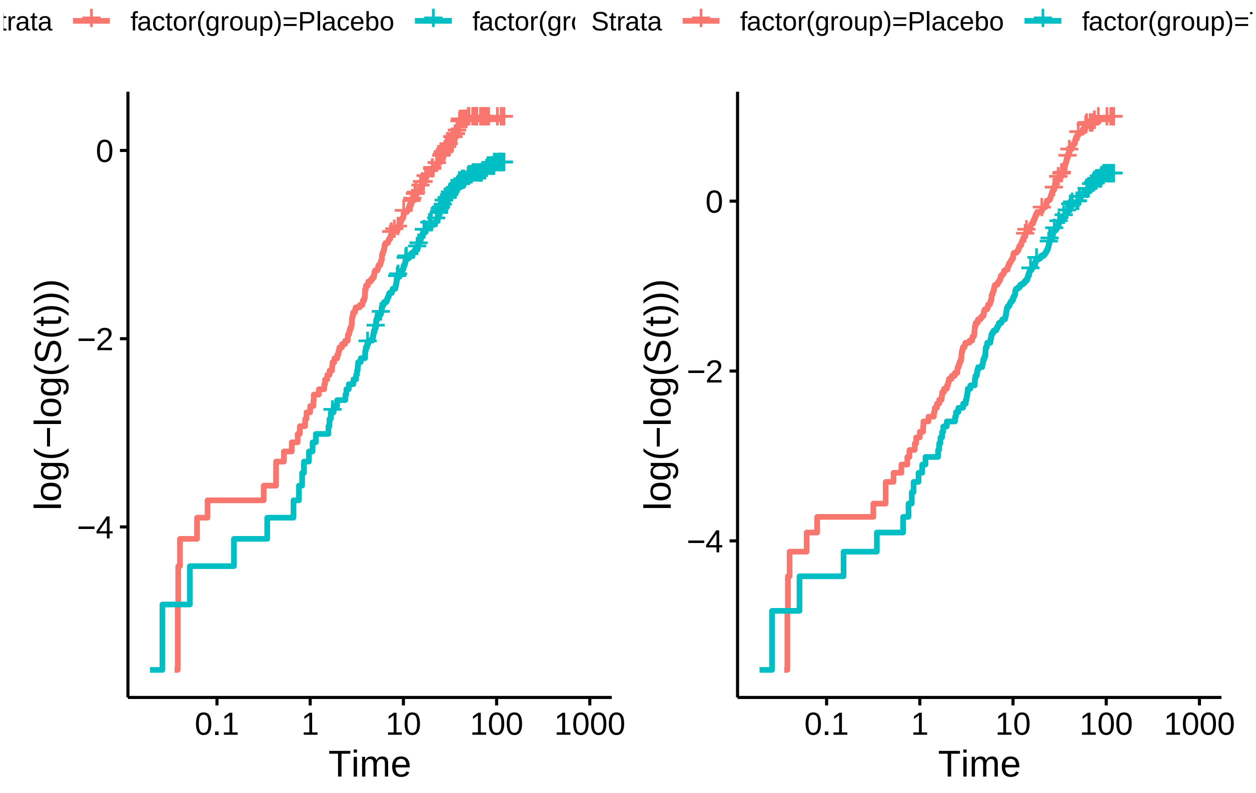
```
## Call:
## coxph(formula = Surv(time, status) ~ factor(group), data = df_10_percent)
##
##   n= 500, number of events= 450
##
##                            coef exp(coef) se(coef)      z Pr(>|z|)
## factor(group)Treatment -0.54206   0.58155  0.09659 -5.612    2e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                        exp(coef) exp(-coef) lower .95 upper .95
## factor(group)Treatment    0.5816       1.72    0.4813    0.7028
##
## Concordance= 0.567  (se = 0.013 )
## Likelihood ratio test= 31.43  on 1 df,   p=2e-08
## Wald test            = 31.5  on 1 df,   p=2e-08
## Score (logrank) test = 32.19  on 1 df,   p=1e-08
```

```
cox_model4 <- coxph(Surv(time, status) ~ factor(group), data=df_0_percent)
summary(cox_model4)
```

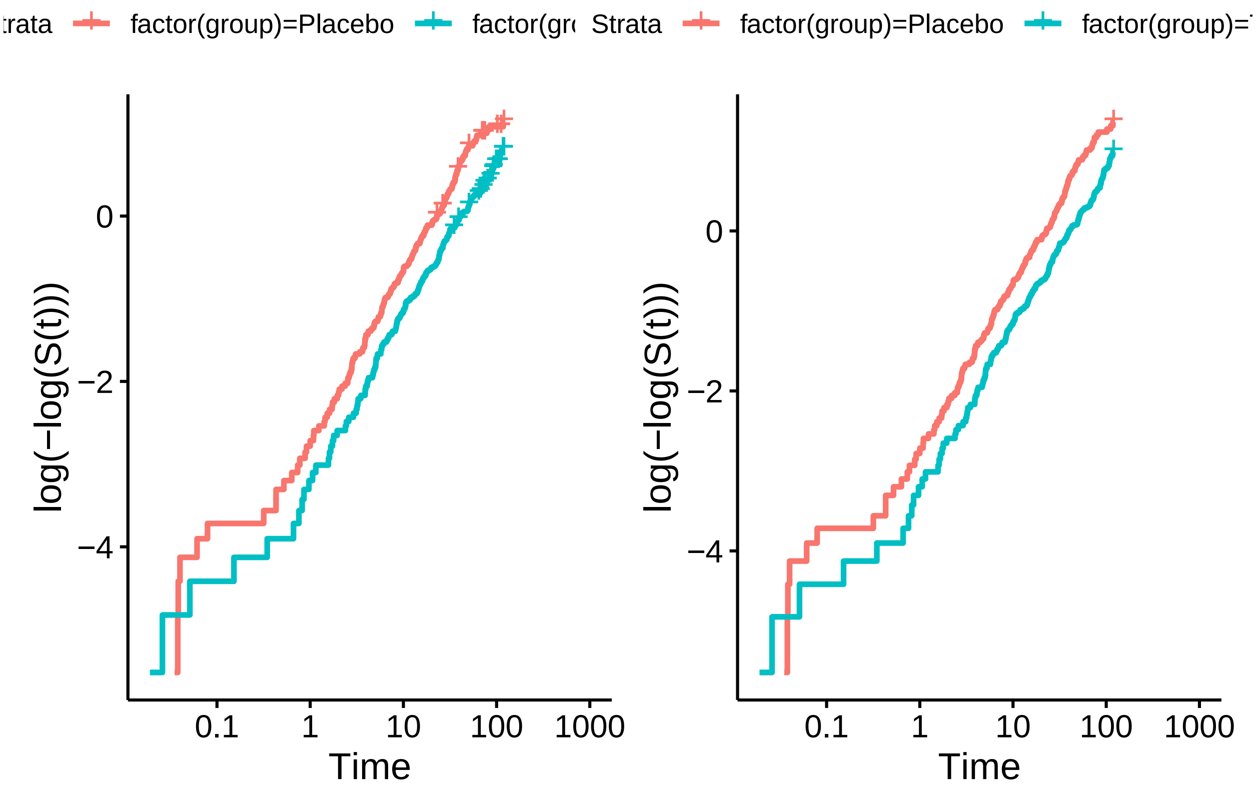
```
## Call:
## coxph(formula = Surv(time, status) ~ factor(group), data = df_0_percent)
##
##   n= 500, number of events= 478
##
##                            coef exp(coef) se(coef)      z Pr(>|z|)
## factor(group)Treatment -0.5575    0.5727   0.0940 -5.931 3.02e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                        exp(coef) exp(-coef) lower .95 upper .95
## factor(group)Treatment    0.5727      1.746    0.4763    0.6885
##
## Concordance= 0.568  (se = 0.012 )
## Likelihood ratio test= 35  on 1 df,   p=3e-09
## Wald test            = 35.17  on 1 df,   p=3e-09
## Score (logrank) test = 35.97  on 1 df,   p=2e-09
```

```
ggsurvplot(km_40_pct_censor, fun = "cloglog")$plot +
  ggsurvplot(km_20_pct_censor, fun = "cloglog")$plot
```

Assessing Proportional Hazards Assumption



```
ggsurvplot(km_10_pct_censor, fun = "cloglog")$plot +
  ggsurvplot(km_0_pct_censor, fun = "cloglog")$plot
```



The visual assessment of the Proportional Hazards assumption seems to hold, since the log(-log) survival probabilities for both groups are parallel. However, there is a single point where they appear to touch, to investigate further, we can perform a `cox.zph` test.

```
cox.zph(cox_model1)
```

```
##                  chisq df p
## factor(group) 2.7e-06  1 1
## GLOBAL        2.7e-06  1 1
```

```
cox.zph(cox_model2)
```

```
##               chisq df    p
## factor(group)  1.76  1 0.18
## GLOBAL         1.76  1 0.18
```

```
cox.zph(cox_model3)
```

```
##                 chisq df    p
## factor(group) 0.00432  1 0.95
## GLOBAL        0.00432  1 0.95
```

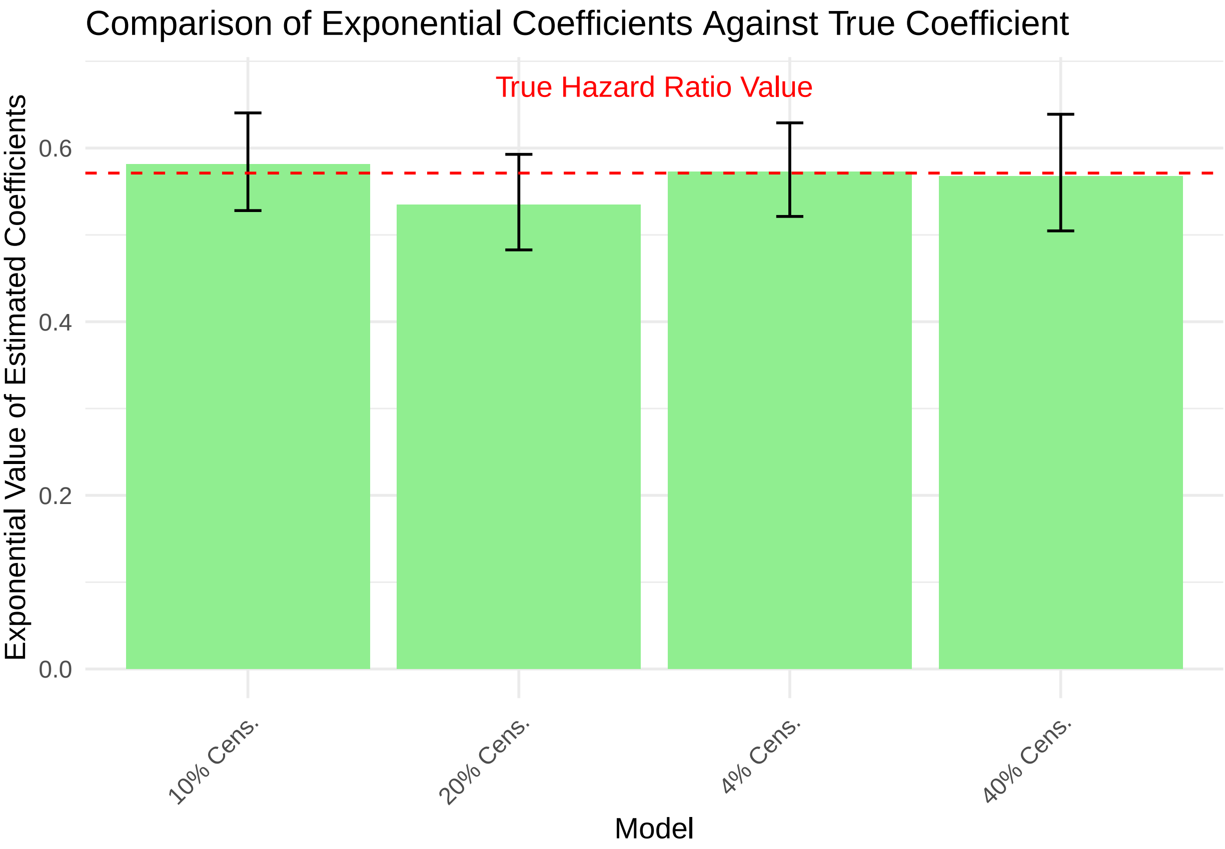
```
cox.zph(cox_model4)
```

```
##               chisq df    p
## factor(group) 7e-04  1 0.98
## GLOBAL        7e-04  1 0.98
```

The null hypothesis of this test is that the PH assumption holds. And since none of the models have a p-value less than 0.05 for this test, we can conclude that the PH assumption holds.

```
true_cox_ph_coeff <- -beta/(1/shape)
model1_coeff <- -0.5659
model2_coeff <- -0.6256
model3_coeff <- -0.54206
```

```r
model4_coeff <- -0.5575

coefficients_df <- tibble(true_cox_ph_coeff, model1_coeff, model2_coeff,
                          model3_coeff, model4_coeff)


coefficients_df <- data.frame(
  coefficient = c("40% Cens.", "20% Cens.", "10% Cens.", "4% Cens."),
  value = c(-0.5659, -0.6256, -0.54206, -0.5575),
  SE = c(0.1180, 0.1027, 0.09659, 0.0940)
)

coefficients_df$exponential <- exp(coefficients_df$value)
coefficients_df$lower <- exp(coefficients_df$value - coefficients_df$SE) # Lower bound
coefficients_df$upper <- exp(coefficients_df$value + coefficients_df$SE) # Upper bound


# Create a bar plot for exponential coefficient values and se's
ggplot(coefficients_df, aes(x = coefficient, y = exponential)) +
  geom_bar(stat = "identity", fill = "lightgreen") +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.1, color = "black") +
  geom_hline(yintercept = exp(true_cox_ph_coeff), color = "red",
             linetype = "dashed", size = 0.5) +
  annotate("text", x = 2.5, y = exp(true_cox_ph_coeff) + 0.1,
           label = "True Hazard Ratio Value",
           color = "red", hjust = 0.5) +
  labs(title = "Comparison of Exponential Coefficients Against True Coefficient",
       x = "Model",
       y = "Exponential Value of Estimated Coefficients") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



To make it easier to see the differences between the coefficient estimates and the true hazard ratio, we can look at this bar plot. From the plot we can observe that the different censoring rates do not have much of an effect on the estimated hazard ratios, and they are all very close to the true hazard ratio.

We can interpret the hazards ratio in the following way, using the hazard ratio from the least censored model:

The hazard of relapse for individuals who are quitting smoking is 1.746 larger for those who get the placebo, than those who get the treatment (Nicotine Replacement Therapy).

Simulation 2: Placebo-Treatment (Nicotine Replacement Therapy) Simulation - Days to Relapse, Two covariates

Now, we replicate this experiment setup with an additional covariate that represents the average number of cigarettes consumed by each participant.

The random variable `cigs_per_day` represents the average number of cigarettes consumed by the participants, and is sampled from a truncated normal distribution with $\mu = 10$, and $\sigma = 5$.

```r
set.seed(11)

n <- 500

n_treatment <- n / 2
n_placebo <- n / 2

shape <- 0.8      # Shape parameter (Weibull) -> 1/scale = Tau
scale_base <- 3     # Baseline scale parameter -> intercept
beta <- 0.7        # Treatment effect = Beta
beta_cigs <- 0.05   # log hazard increase per cigarette

# Simulate continuous covariate (cigarette consumption)
cigs_placebo <- rtruncnorm(a=0, n_placebo, mean = 10, sd = 5)
```

```
cigs_treatment <- rtruncnorm(a=0,n_treatment, mean = 10, sd = 5)
cigs_per_day <- c(cigs_placebo, cigs_treatment)


scale_placebo <- exp(scale_base + beta_cigs * cigs_placebo)
T_placebo <- rweibull(n_placebo, shape = shape, scale = scale_placebo)


scale_treatment <- exp(scale_base + beta + beta_cigs * cigs_treatment)
T_treatment <- rweibull(n_treatment, shape = shape, scale = scale_treatment)

time <- c(T_placebo, T_treatment)
status <- rep(1, n)
group <- rep(c("Placebo", "Treatment"), each = n / 2)

data <- data.frame(
  time = time,
  status = status,
  group = group,
  cigs_per_day = cigs_per_day
)
```

Verifying The Data Generation

```
aft <- survreg(Surv(time, status) ~ factor(group) + cigs_per_day, data=data,
                dist = "weibull")
summary(aft)
```

```
##
## Call:
## survreg(formula = Surv(time, status) ~ factor(group) + cigs_per_day,
##     data = data, dist = "weibull")
##                         Value Std. Error     z       p
## (Intercept)            2.6586     0.1535 17.32 < 2e-16
## factor(group)Treatment 0.6906     0.1132  6.10 1.1e-09
## cigs_per_day           0.0748     0.0128  5.84 5.2e-09
## Log(scale)             0.2353     0.0346  6.79 1.1e-11
##
## Scale= 1.27
##
## Weibull distribution
## Loglik(model)= -2425.1   Loglik(intercept only)= -2459.1
##  Chisq= 68.02 on 2 degrees of freedom, p= 1.7e-15
## Number of Newton-Raphson Iterations: 5
## n= 500
```

We can see from the output of the AFT model that the coefficient estimates, and the estimate of the scale, τ are extremely close to the true values, therefore we can verify that the data generation worked properly.

Censoring

The same censoring mechanism was employed. However, censoring distribution parameters were adjusted to achieve the same level of censoring rates.

```
set.seed(10)
# Censoring Rate = 0.4
c_values <- rtruncnorm(n, a=0, mean=40, sd=30)
time <- c(T_placebo, T_treatment)

status <- rep(NA, n)
```

```r
for(i in 1:n){
  if(time[i] > 120){
    status[i] <- 0
    time[i] <- 120
  }
  else if(c_values[i] < time[i]){
    status[i] <- 0
  }
  else{
    status[i] <- 1
  }
}

df_40_percent <- data.frame(time, status, group)
paste("This censoring rate is:", 100* (1 - (sum(status) /n)), "%")
```

```
## [1] "This censoring rate is: 39.8 %"
```

```r
#-----------------------------------------------------------------
# Censoring Rate = 0.2
c_values <- rtruncnorm(n, a=0, mean=90, sd=30)
time <- c(T_placebo, T_treatment)

status <- rep(NA, n)

for(i in 1:n){
  if(time[i] > 120){
    status[i] <- 0
    time[i] <- 120
  }
  else if(c_values[i] < time[i]){
    status[i] <- 0
  }
  else{
    status[i] <- 1
  }
}

df_20_percent <- data.frame(time, status, group)
paste("This censoring rate is:", 100* (1 - (sum(status) /n)), "%")
```

```
## [1] "This censoring rate is: 20.4 %"
```

```r
#-----------------------------------------------------------------
# Censoring Rate = 0.1
c_values <- rtruncnorm(n, a=0, mean=160, sd=30)
time <- c(T_placebo, T_treatment)

status <- rep(NA, n)

for(i in 1:n){
  if(time[i] > 120){
    status[i] <- 0
    time[i] <- 120
  }
  else if(c_values[i] < time[i]){
    status[i] <- 0
  }
  else{
    status[i] <- 1
  }
}

df_10_percent <- data.frame(time, status, group)
paste("This censoring rate is:", 100* (1 - (sum(status) /n)), "%")
```

```
## [1] "This censoring rate is: 11.6 %"
```

```
#-------------------------------------------------------------------
# Censoring Rate = 0 (Only Type I Right Censor present)
time <- c(T_placebo, T_treatment)

status <- rep(NA, n)

for(i in 1:n){
  if(time[i] > 120){
    status[i] <- 0
    time[i] <- 120
  }
  else{
    status[i] <- 1
  }
}

df_0_percent <- data.frame(time, status, group)
paste("This censoring rate is:", 100* (1 - (sum(status) /n)), "%")
```

```
## [1] "This censoring rate is: 11.2 %"
```

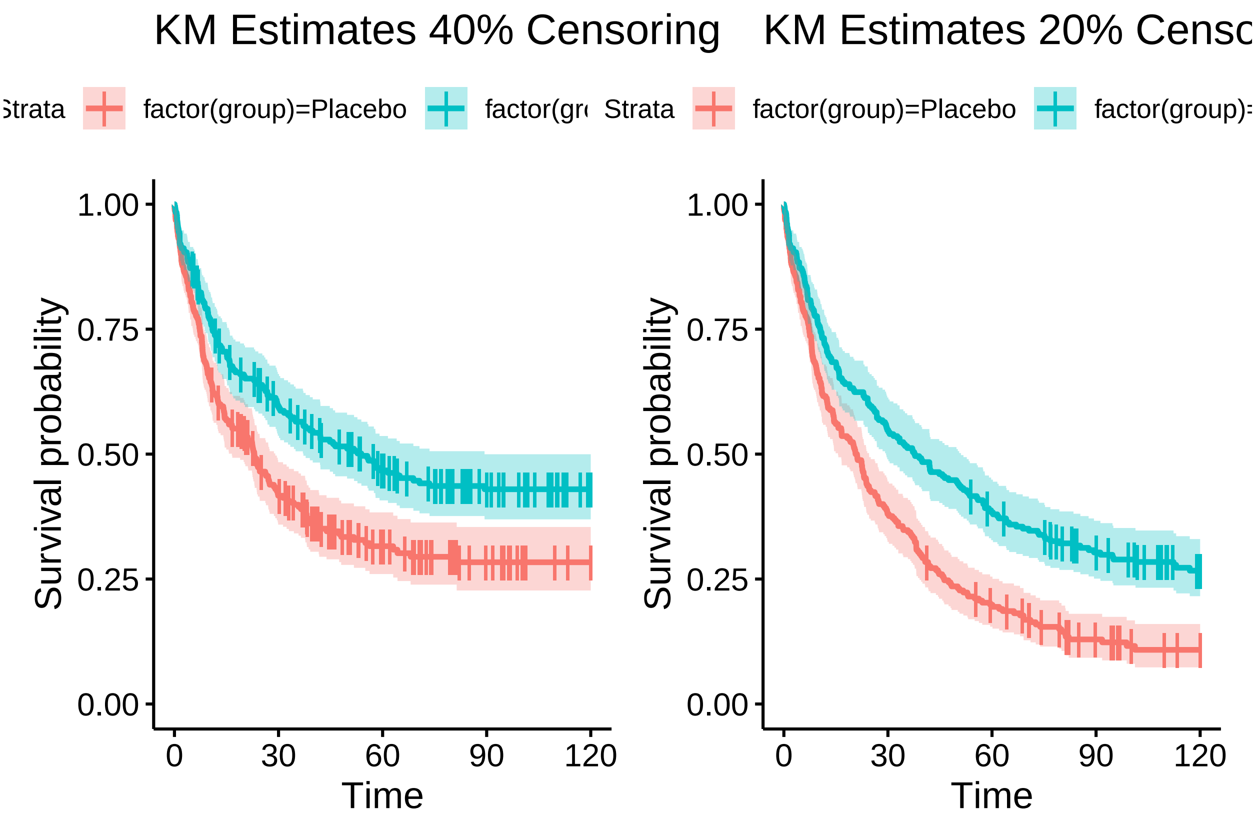
Cox Proportional Hazards Model Fitting and Evaluation

```
km_40_pct_censor <- surv_fit(Surv(time, status) ~ factor(group),
                             data=df_40_percent)
km_20_pct_censor <- surv_fit(Surv(time, status) ~ factor(group),
                             data=df_20_percent)
km_10_pct_censor <- surv_fit(Surv(time, status) ~ factor(group),
                             data=df_10_percent)
km_0_pct_censor <- surv_fit(Surv(time, status) ~ factor(group),
                             data=df_0_percent)
```

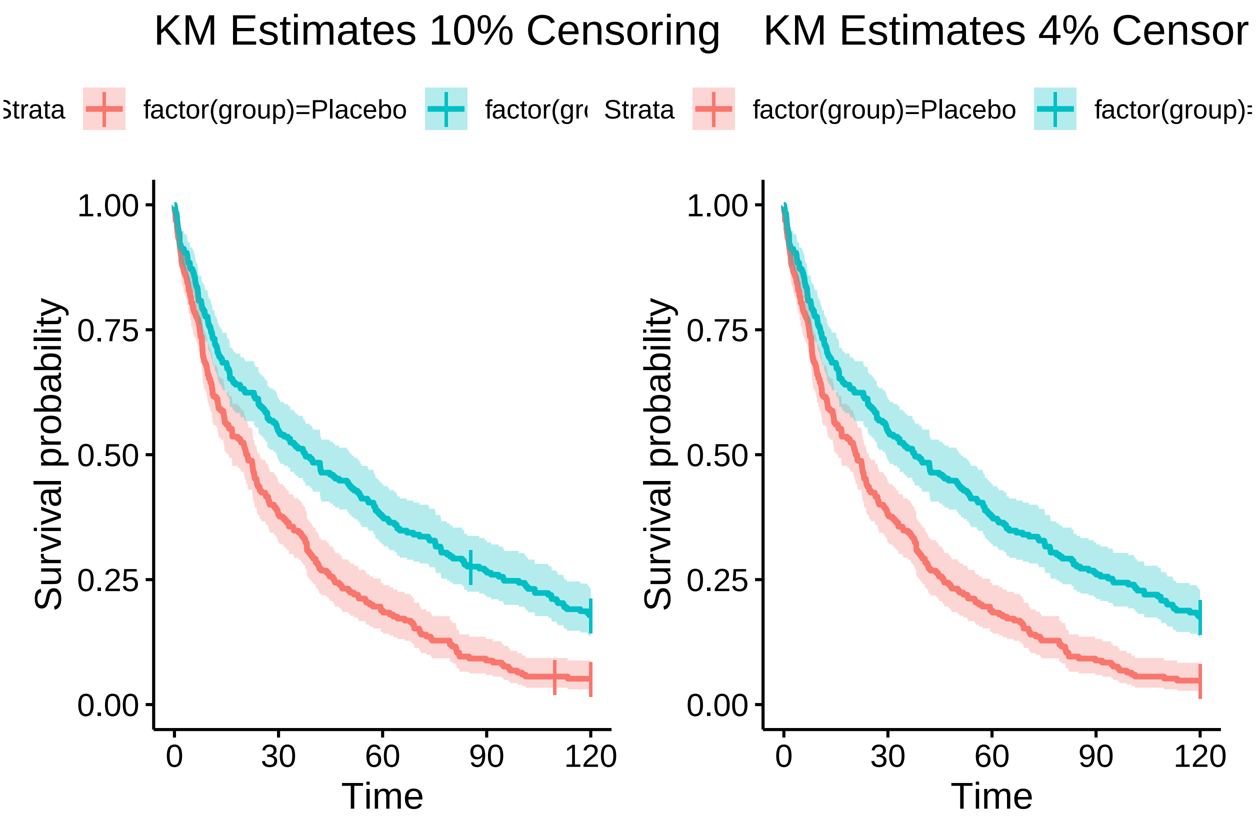
Fitting the models

```
pl1 <- ggsurvplot(km_40_pct_censor, title = "KM Estimates 40% Censoring",
                  conf.int = TRUE, censor.shape = "|")

pl2 <- ggsurvplot(km_20_pct_censor, title = "KM Estimates 20% Censoring",
                  conf.int = TRUE, censor.shape = "|")

pl3 <- ggsurvplot(km_10_pct_censor, title = "KM Estimates 10% Censoring",
                  conf.int = TRUE, censor.shape = "|")

pl4 <- ggsurvplot(km_0_pct_censor, title = "KM Estimates 4% Censoring",
                  conf.int = TRUE, censor.shape = "|")

pl1$plot + pl2$plot
```

Visualizing Survival Probabilities



```r
pl3$plot + pl4$plot
```



A similar observation from Simulation 1 can be made. Higher censoring rates lead to different shaped Survival curves.

```r
cox_model1 <- coxph(Surv(time, status) ~ factor(group) + cigs_per_day,
                    data=df_40_percent)
summary(cox_model1)
```

Cox PH models

```
## Call:
## coxph(formula = Surv(time, status) ~ factor(group) + cigs_per_day,
##     data = df_40_percent)
##
##   n= 500, number of events= 301
##
##                            coef exp(coef) se(coef)      z Pr(>|z|)
## factor(group)Treatment -0.46007   0.63124  0.11673 -3.941 8.11e-05 ***
## cigs_per_day           -0.06370   0.93828  0.01297 -4.911 9.08e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                        exp(coef) exp(-coef) lower .95 upper .95
## factor(group)Treatment    0.6312      1.584    0.5021    0.7935
## cigs_per_day              0.9383      1.066    0.9147    0.9624
##
## Concordance= 0.61  (se = 0.017 )
## Likelihood ratio test= 38.88  on 2 df,   p=4e-09
## Wald test            = 37.91  on 2 df,   p=6e-09
## Score (logrank) test = 38.28  on 2 df,   p=5e-09
```

```r
cox_model2 <- coxph(Surv(time, status) ~ factor(group) + cigs_per_day,
                    data=df_20_percent)
summary(cox_model2)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ factor(group) + cigs_per_day,
##     data = df_20_percent)
##
##   n= 500, number of events= 398
##
##                            coef exp(coef) se(coef)      z Pr(>|z|)
## factor(group)Treatment -0.52070   0.59411  0.10182 -5.114 3.15e-07 ***
## cigs_per_day           -0.05936   0.94236  0.01133 -5.240 1.61e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                        exp(coef) exp(-coef) lower .95 upper .95
## factor(group)Treatment    0.5941      1.683    0.4866    0.7253
## cigs_per_day              0.9424      1.061    0.9217    0.9635
##
## Concordance= 0.609  (se = 0.015 )
## Likelihood ratio test= 52.72  on 2 df,   p=4e-12
## Wald test            = 51.6  on 2 df,   p=6e-12
## Score (logrank) test = 52.26  on 2 df,   p=4e-12
```

```r
cox_model3 <- coxph(Surv(time, status) ~ factor(group) + cigs_per_day,
                    data=df_10_percent)
summary(cox_model3)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ factor(group) + cigs_per_day,
##     data = df_10_percent)
##
```

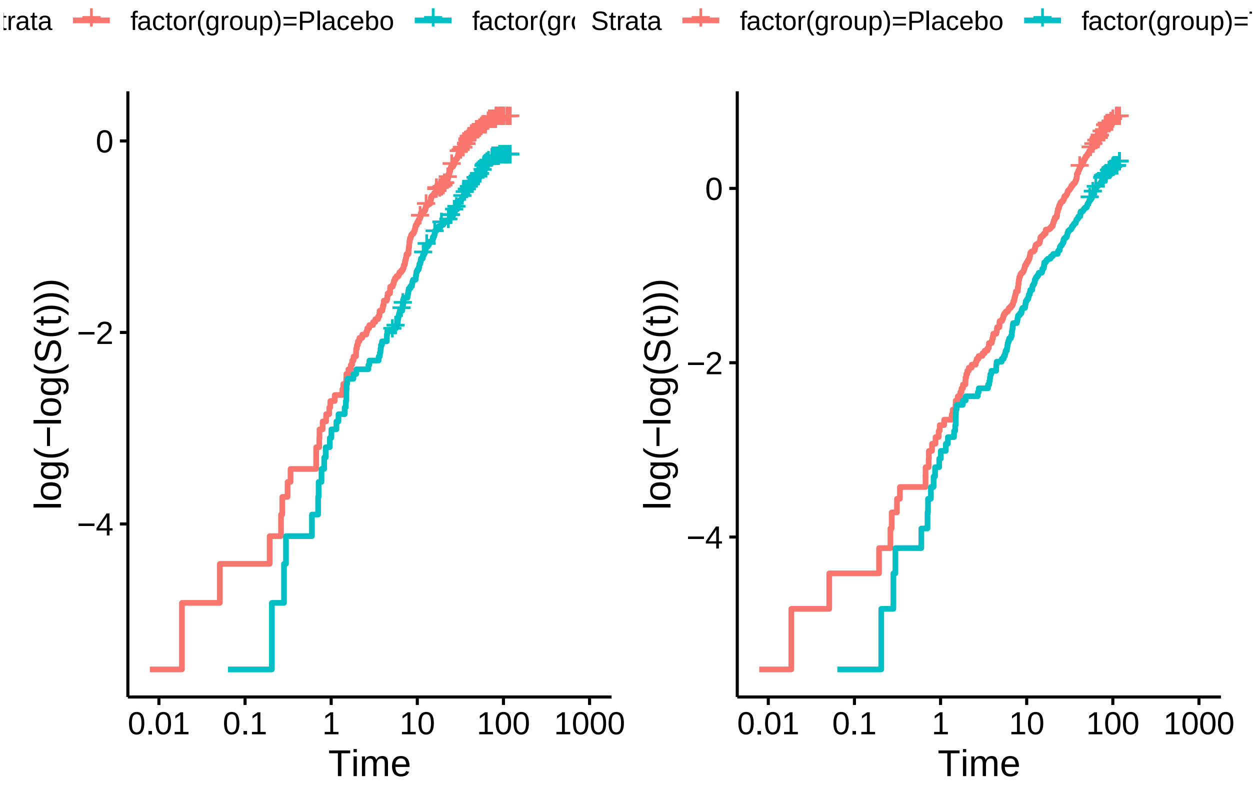
```
##   n= 500, number of events= 442
##
##                             coef exp(coef) se(coef)      z Pr(>|z|)
## factor(group)Treatment -0.54255   0.58126  0.09689 -5.600 2.15e-08 ***
## cigs_per_day           -0.05628   0.94527  0.01079 -5.215 1.83e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                        exp(coef) exp(-coef) lower .95 upper .95
## factor(group)Treatment    0.5813      1.720    0.4807    0.7028
## cigs_per_day              0.9453      1.058    0.9255    0.9655
##
## Concordance= 0.61  (se = 0.015 )
## Likelihood ratio test= 57.88  on 2 df,   p=3e-13
## Wald test            = 56.84  on 2 df,   p=5e-13
## Score (logrank) test = 57.66  on 2 df,   p=3e-13
```

```
cox_model4 <- coxph(Surv(time, status) ~ factor(group) + cigs_per_day,
                    data=df_0_percent)
summary(cox_model4)
```

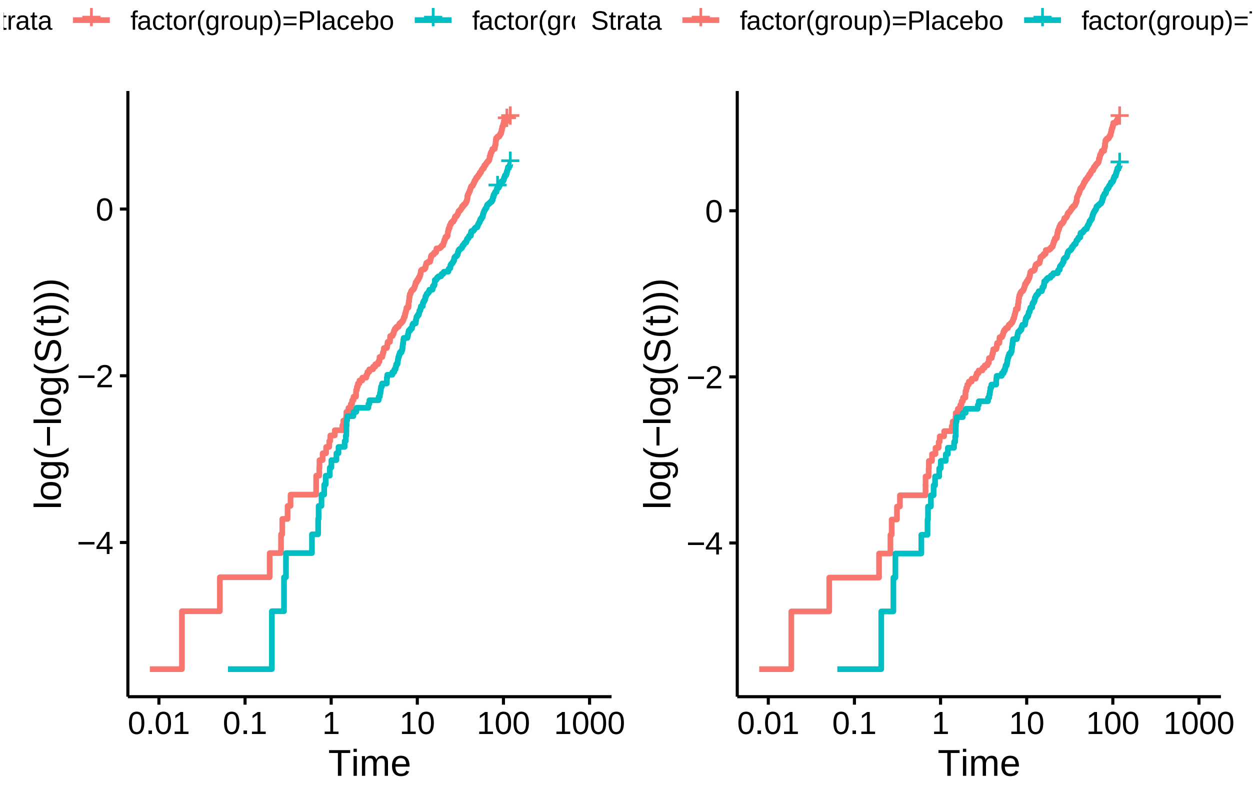
```
## Call:
## coxph(formula = Surv(time, status) ~ factor(group) + cigs_per_day,
##     data = df_0_percent)
##
##   n= 500, number of events= 444
##
##                             coef exp(coef) se(coef)      z Pr(>|z|)
## factor(group)Treatment -0.54575   0.57941  0.09669 -5.644 1.66e-08 ***
## cigs_per_day           -0.05759   0.94403  0.01078 -5.344 9.11e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                        exp(coef) exp(-coef) lower .95 upper .95
## factor(group)Treatment    0.5794      1.726    0.4794    0.7003
## cigs_per_day              0.9440      1.059    0.9243    0.9642
##
## Concordance= 0.61  (se = 0.015 )
## Likelihood ratio test= 59.73  on 2 df,   p=1e-13
## Wald test            = 58.63  on 2 df,   p=2e-13
## Score (logrank) test = 59.48  on 2 df,   p=1e-13
```

```

ggsurvplot(km_40_pct_censor, fun = "cloglog")$plot +
  ggsurvplot(km_20_pct_censor, fun = "cloglog")$plot
```

Checking Proportional Hazards Assumption



```r
ggsurvplot(km_10_pct_censor, fun = "cloglog")$plot +
  ggsurvplot(km_0_pct_censor, fun = "cloglog")$plot
```



The visual assessment shows that the survival curves are parallel, yet there is a single point of intersection, to evaluate further, we can perform a `cox.zph` test.

```
cox.zph(cox_model1)
```

```
##                 chisq df    p
## factor(group) 0.0692  1 0.79
## cigs_per_day  2.4957  1 0.11
## GLOBAL        2.6000  2 0.27
```

```
cox.zph(cox_model2)
```

```
##               chisq df     p
## factor(group) 0.648  1 0.421
## cigs_per_day  4.926  1 0.026
## GLOBAL        5.496  2 0.064
```

```
cox.zph(cox_model3)
```

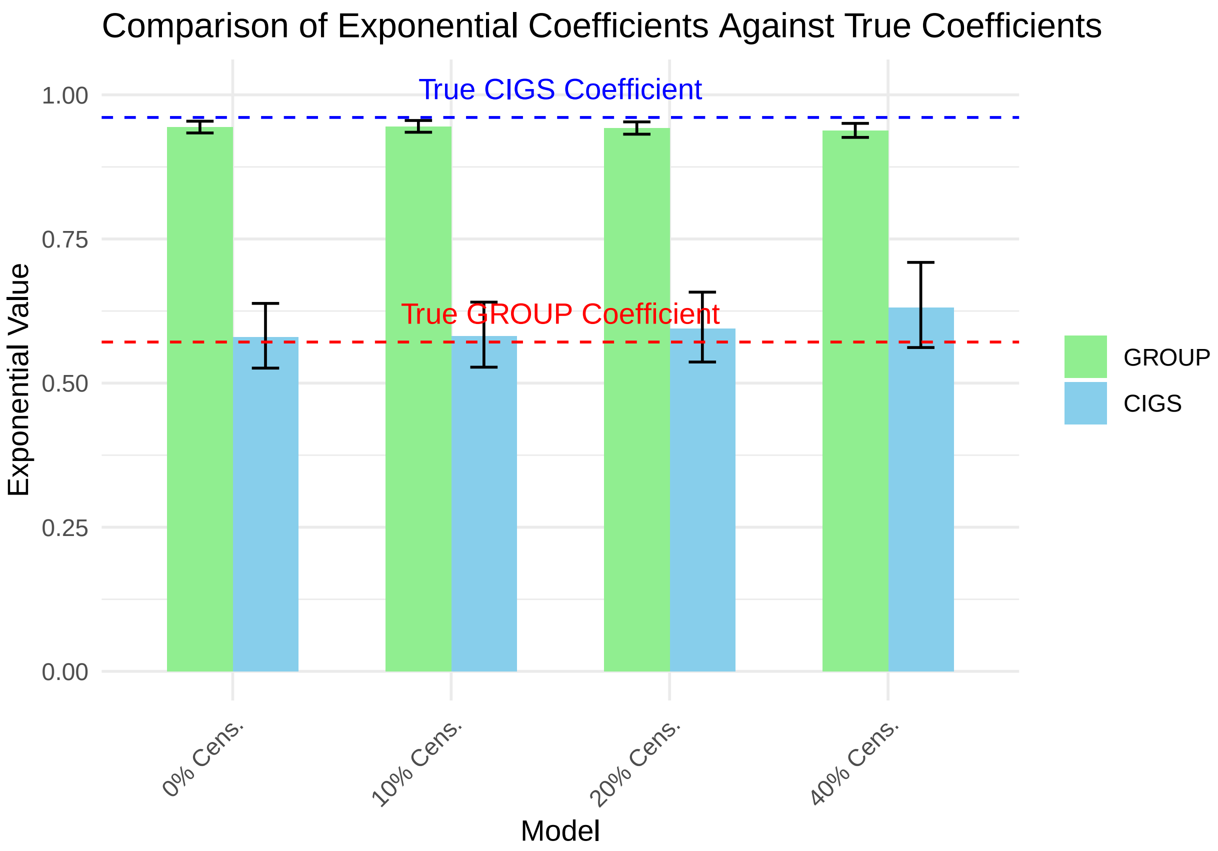
```
##               chisq df     p
## factor(group) 0.852  1 0.356
## cigs_per_day  5.877  1 0.015
## GLOBAL        6.692  2 0.035
```

```
cox.zph(cox_model4)
```

```
##               chisq df     p
## factor(group) 0.939  1 0.333
## cigs_per_day  4.955  1 0.026
## GLOBAL        5.856  2 0.054
```

From the results of the tests, we see that none of the p-values are less than 0.05, so the Proportional Hazards assumption holds for all models.

```
true_cox_ph_coeff_GROUP <- -beta/(1/shape)
model1_coeff_GROUP <- -0.46007
model2_coeff_GROUP <- -0.52070
model3_coeff_GROUP <- -0.54255
model4_coeff_GROUP <- -0.54575

true_cox_ph_coeff_CIGS <- -beta_cigs/(1/shape)
model1_coeff_CIGS <- -0.06370
model2_coeff_CIGS <- -0.05936
model3_coeff_CIGS <- -0.05628
model4_coeff_CIGS <- -0.05759




coefficients_df <- data.frame(
  model = rep(c("40% Cens.", "20% Cens.", "10% Cens.", "0% Cens."), 2),
  covariate = rep(c("GROUP", "CIGS"), each = 4),
  value = c(-0.46007, -0.52070, -0.54255, -0.54575, # GROUP coefficients
            -0.06370, -0.05936, -0.05628, -0.05759), # CIGS coefficients
  SE = c(0.11673, 0.10182, 0.09689, 0.09669, # GROUP standard errors
         0.01297, 0.01133, 0.01079, 0.01078) # CIGS standard errors
)

coefficients_df$exponential <- exp(coefficients_df$value)
coefficients_df$lower <- exp(coefficients_df$value - coefficients_df$SE)
coefficients_df$upper <- exp(coefficients_df$value + coefficients_df$SE)


#  grouped bar plot with error bars
ggplot(coefficients_df, aes(x = model, y = exponential, fill = covariate)) +
  geom_bar(stat = "identity", position = position_dodge(), width = 0.6) +
  geom_errorbar(aes(ymin = lower, ymax = upper), position = position_dodge(0.6),
                width = 0.25, color = "black") +
  geom_hline(yintercept = exp(true_cox_ph_coeff_GROUP), color = "red",
             linetype = "dashed", size = 0.5) +
  geom_hline(yintercept = exp(true_cox_ph_coeff_CIGS), color = "blue",
             linetype = "dashed", size = 0.5) +
  annotate("text", x = 2.5, y = exp(true_cox_ph_coeff_GROUP) + 0.05,
           label = "True GROUP Coefficient", color = "red", hjust = 0.5) +
  annotate("text", x = 2.5, y = exp(true_cox_ph_coeff_CIGS) + 0.05,
           label = "True CIGS Coefficient", color = "blue", hjust = 0.5) +
  labs(title = "Comparison of Exponential Coefficients Against True Coefficients",
       x = "Model",
       y = "Exponential Value") +
  scale_fill_manual(values = c("lightgreen", "skyblue"),
                    labels = c("GROUP", "CIGS")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.title = element_blank())
```



Interestingly, the exponential coefficient estimates for the continuous predictor `cigarettes_per_day` are relatively stable despite the different censoring rates. However, in this simulation the treatment group covariates are impacted more by the higher censoring rates, than the previous simulation. We can see that there is a considerable divergence in the coefficient estimate of treatment group compared to the true coefficient, in the higher censoring rate models; and they also have much larger standard errors.

We can interpret the hazards ratio in the following way, using the hazard ratio from the least censored model:

The hazard of relapse for individuals who are quitting smoking is 1.059 larger when z_2 : Average Cigarettes Per Day is one unit larger, controlling for treatment.

The hazard of relapse for individuals who are quitting smoking is 1.726 larger for those who get the placebo, than those who get the treatment (Nicotine Replacement Therapy), controlling for cigarette consumption.

Conclusion

Contrary to my intuition, the Cox Proportional Hazards model was fairly robust to higher censoring rates, and was able to produce very close estimates of the Hazard ratios. However, coefficient estimates for the treatment group suffered more from the higher censoring rates in the 2nd simulation, which might indicate that including a continuous covariate on top of the treatment group covariate makes the Cox PH model more biased when the data has higher censoring rates.