A simulation study of comparing three survival models

Yijing Tao; Renjie Wei; Jibei Zheng; Haolin Zhong; Anyu Zhu

Objective

In survival analysis, which aims to investigate the efficacy of a treatment X on a survival time T, the three most important models are proportional hazards model of Exponential, Weibull and Cox.

In proportional hazard models, the hazard ratio, which refers to the instantaneous risk of failure at time t giving that a patient has survived until time t, is defined as:

$$h_i(t) = h_0(t)\exp(x_i\theta)$$

The formula suggests that the hazard ratio is dominated by the baseline hazard function $h_0(t)$, a binary treatment indicator x_i which coded 0 for control and 1 for the treatment, and our parameter of interest, θ, which is the log hazard ratio for the treatment effect and measures the relative hazard reduction due to treatment in comparison to the control.

The three proportional hazard models have different assumptions on the baseline hazard function, which makes them differ in flexibility and performance. To examine their accuracy and efficiency in a series of scenarios and their robustness against misspecified distribution, we conducted this simulation study.

Statistical Methods

Structure of Simulated Survival Data

In this study, we simulate right censored survival data with one binary treatment indicator x. Our response variable is a dichotomous variable, coded as 1 when event occurred or 0 when event did not occur during the 5-year observation period.

Follow up time is measured from time zero until the event occurs, the study ends or the participant is lost, whichever comes first.

Methods of Survival Analysis

Suppose $T \in [0, \infty)$ is the time to a event of interest, such as death, disease onset, device failure, etc. To analyze such data, we define a survival function S as

$$S(t) = \Pr(T > t) = \int_t^\infty f(s)ds$$

Design of simulation settings

We conducted simulation studies to assess the performance of three survival models. In total, we created 7 simulation settings by mixing the event time generated from the three specified baseline hazard function: exponential, Weibull, and Gompertz hazard function. Then the generated data were fitted to exponential, Weibull, and Cox proportional hazard models. The parameters we applied in the models are constants: $\alpha = 0.5$, $\gamma = 1.5$, $\beta = -0.5$. Since the shape of Weibull distribution has great difference between $\gamma > 1$ and $\gamma < 1$, thus we simulated in scenarios where $\gamma = 0.5$ and $\gamma = 1.5$,

We simulated 500 data sets in each simulation setting. After running the models, a set of β was extracted and used to calculate the mean bias, variance, and squared error. To evaluate the efficiency performances of models, we simulated data of different sample sizes: 20, 40, 60, 80, 100, 200, 400. Similarly, bias, variance, MSE are calculated.

All the simulation processes were performed in R.

Methods for generating data

The survival dataset contains treatment assignment, status indicator, and observed time. Treatment assignment variable X_i is generated from a Bernoulli distribution with p = 0.5. By utilizing the inverse transformation method, we can obtain event time T:

$T = H_0^{-1}\left(\frac{-\log(u)}{e^{x^T\beta}}\right)$, where $U \sim U(0,1)$

The followings are specific baseline hazard functions we applied:

1. Survival time under Exponential distribution: $T = -\frac{\log(u)}{\lambda e^{x^T\beta}}$
2. Survival time under Weibull distribution: $T = \left(-\frac{\log(u)}{\lambda e^{x^T\beta}}\right)^{\frac{1}{\gamma}}$
3. Survival time under Gompertz distribution: $T = \frac{1}{\alpha}\log\left(1 - \frac{\alpha\log(u)}{\lambda e^{x^T\beta}}\right)$

We simulated survival data by the event time generated from the mixtures of three baseline distributions.

Mixture of Exponential and Weibull distribution:

$$T = p * (-\frac{\log(u)}{\lambda e^{x^T\beta}}) + (1-p) * (-\frac{\log(u)}{\lambda e^{x^T\beta}})^{1/\gamma}$$

We take values of p as 0, 0.5 and 1. When `p = 1`, event time is generated from exponential baseline; when `p = 0`, event time is generated from Weibull distribution.

Mixture of Exponential and Gompertz distribution:

$$T = p * (-\frac{\log(u)}{\lambda e^{x^T\beta}}) + (1-p) * (\frac{1}{\alpha}\log\left(1 - \frac{\alpha\log(u)}{\lambda e^{x^T\beta}}\right))$$

Similary, we take values of p as 0, 0.5 and 1. When `p = 1`, event time is generated from exponential baseline; when `p = 0`, event time is generated from Gompertz distribution. Finally, make event indicator variable by applying administrative censoring at t = 5.

By repeating each of the above simulation process 500 times, we get survival datasets with sample size ranging from 20 to 400.

Selection of performance measure

We decided to use Average deviance, Variance and MSE to measure the performance of the 3 models we have got, which means whether the models can simulate the real hazard model most efficiently.

Bias:
The bias is the average difference between the estimated treatment effects β and the real β. The bias measure deviates the desired prediction of the learning algorithm from the true result, i.e., it portrays the fitting ability of the learning algorithm itself. The larger the bias, the greater the degree of difference between the estimated β and the real β, and the less accuracy the simulation is.
According to the definition, the model which have a smaller MBE might have a better predicting accuracy. In the data we generated, we assume $\beta_1 = 0.5$. So when calculating the average bias, we used the equation

below to get the average bias of each model.

Exponential:

$$MBE = \sum \frac{\beta_{exp} - \beta_1}{n} = \sum \frac{\beta_{exp} - 0.5}{n}$$

Weibull:

$$MBE = \sum \frac{\beta_{weibull} - \beta_1}{n} = \sum \frac{\beta_{weibull} - 0.5}{n}$$

Cox:

$$MBE = \sum \frac{\beta_{cox} - \beta_1}{n} = \sum \frac{\beta_{cox} - 0.5}{n}$$

Variance:
The variance is the average of the sum of the squares of the differences between the estimated β and the real β. Variance measures the change in learning performance due to a change in the same size training set, i.e., it portrays the impact of data perturbations. The variance indicates how much the prediction function constructed by all models differs from the true function.
Efficiency refers to two unbiased estimates of the same overall parameter where the estimate with smaller variance is more valid. According to the definition, the model which have a smaller variance will have a better predicting efficiency. In the data we generated, we assume $\beta_1 = 0.5$. So when calculating the variance, we used the equation below to get the variance of each model.

Exponential:

$$variance = \sum \frac{(\beta_{exp} - \beta_1)^2}{n-1} = \sum \frac{(\beta_{exp} - 0.5)^2}{n-1}$$

Weibull:

$$variance = \sum \frac{(\beta_{weibull} - \beta_1)^2}{n-1} = \sum \frac{(\beta_{weibull} - 0.5)^2}{n-1}$$

Cox:

$$variance = \sum \frac{(\beta_{cox} - \beta_1)^2}{n-1} = \sum \frac{(\beta_{cox} - 0.5)^2}{n-1}$$

Low bias with low variance is the effect we seek, when the predicted values are right on the bull's eye (closest to the true value) and are more concentrated (less variance).

In the case of low bias and high variance, the predicted value basically falls around the true value, but it is very scattered, and the variance is larger, which means the stability of the model is not good enough.

In the case of high bias and low variance, the predicted values are far from the true values, but the values are concentrated and the variance is small; the stability of the model is good, but the prediction accuracy is not

high, and it is in the state of "inaccurate prediction as usual".

When the bias is high and the variance is high, this is the last result we want to see. The model is not only inaccurate, but also unstable, and the predicted values are very different every time.

MSE: The mean squared error (MSE) in parameter estimation is the expected value of the squared difference between the estimated β and the true β. It is defined as

$$MSE(\beta) = var(\beta) + bias^2(\beta)$$

According to the definition, the model which have a smaller MSE will have a better predicting performance in both accuracy and efficiency. In the data we generated, we assume $\beta_1 = 0.5$. So when calculating the MSE, we used the equation below to get the MSE of each model.

Exponential:

$$MSE = \sum \frac{(\beta_{exp} - \beta_1)^2}{n} = \sum \frac{(\beta_{exp} - 0.5)^2}{n}$$

Weibull:

$$MSE = \sum \frac{(\beta_{weibull} - \beta_1)^2}{n} = \sum \frac{(\beta_{weibull} - 0.5)^2}{n}$$

Cox:

$$MSE = \sum \frac{(\beta_{cox} - \beta_1)^2}{n} = \sum \frac{(\beta_{cox} - 0.5)^2}{n}$$

After getting the bias, variance and MSE of the 3 models when the size and the component of the data we generated is different, we made the spaghetti plot where the y value is the value of bias, variance or MSE and the x value is the size of the data.
Based on the plot, we can find out which model is the most suitable model easily by compare the value of bias, variance and MSE.

Results

Figure 1 shows the results of the simulation from mixtures of exponential distribution and Weibull distribution with sample sizes from 20 to 400. The mixing parameter p = 0 represents a full Weibull distribution with λ = 0.5, γ = 1.5; p = 1 represents a full exponential with $\lambda = 0.5(\gamma = 1)$; and p = 0.5 represents a half and half mixture. The true β = -0.5 is lined as reference. We can see obviously from the plot that when sample size is large enough and the true distribution is Weibull, the Cox model and Weibull model outperform the exponential model, which always tends to overestimate the true β. With p increasing, the gaps tend to diminish gradually and when the true distribution is exponential, three models have similar performances.

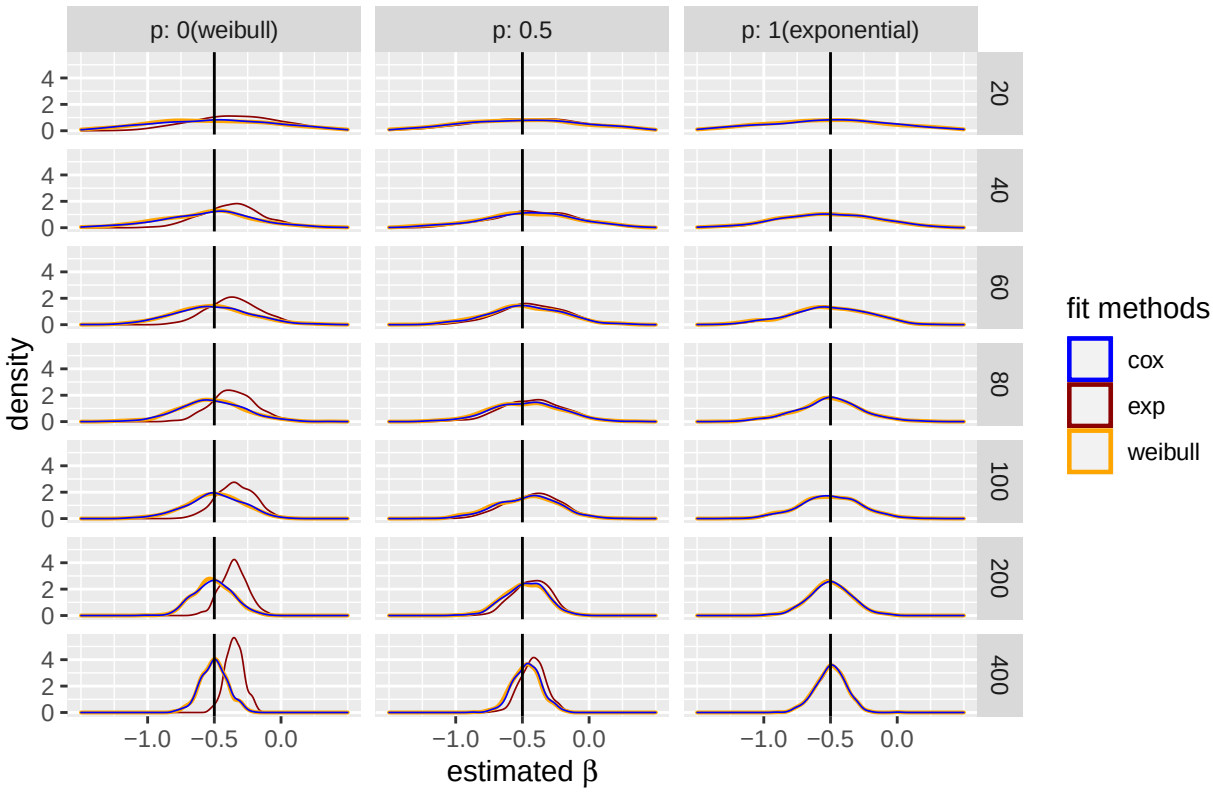

Figure 1

Figure 2-4 show the bias, variance and MSE of each underlying distributions. From the bias plot, we can see that the exponential model has the largest bias, and the only case it has the highest prediction accuracy is when the true distribution close to a purely exponential. The big difference in performances of exponential model in different distributions shows that it is the least robust againt a misspecified distribution. On the other hand, the variance plot shows that the exponential model constantly has the lowest versatility. When looking at MSE, exponential model performs the best when sample sizes are relatively small; but for large sample sizes, the Cox model and Weibull model perform better.

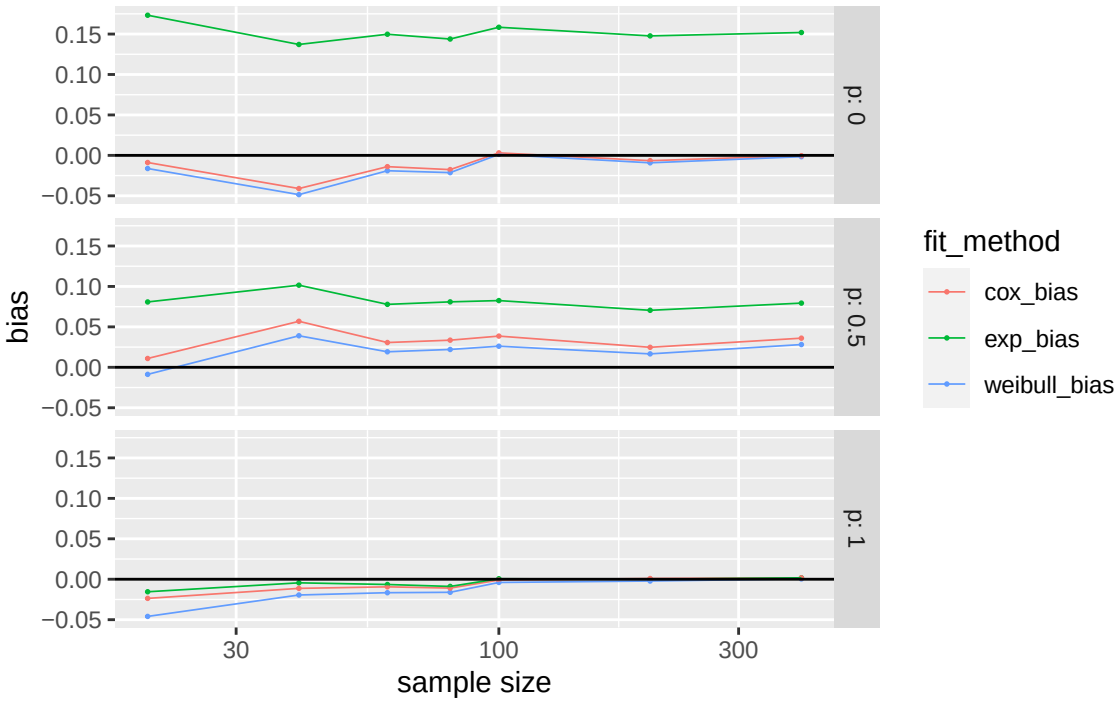

Figure 2

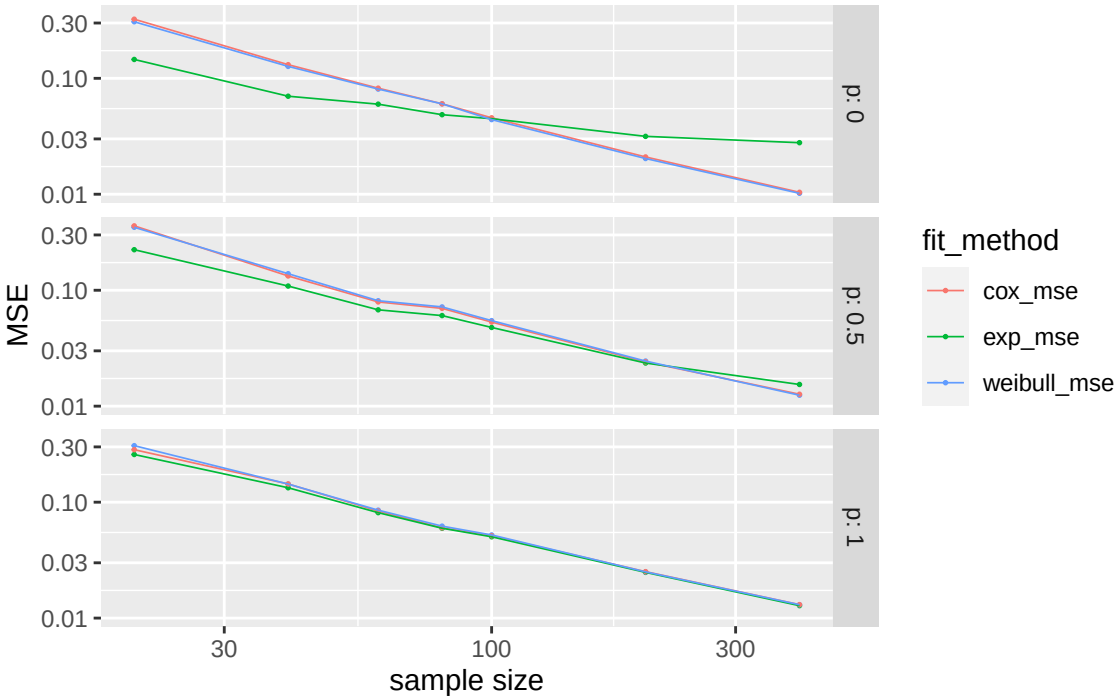

Figure 4

Figure 5 shows the results of the simulation from mixtures of exponential and Weibull distribution with $\gamma = 0.5$. It is the similar case as Figure 1, only this time the exponential model tends to underestimate the true β.

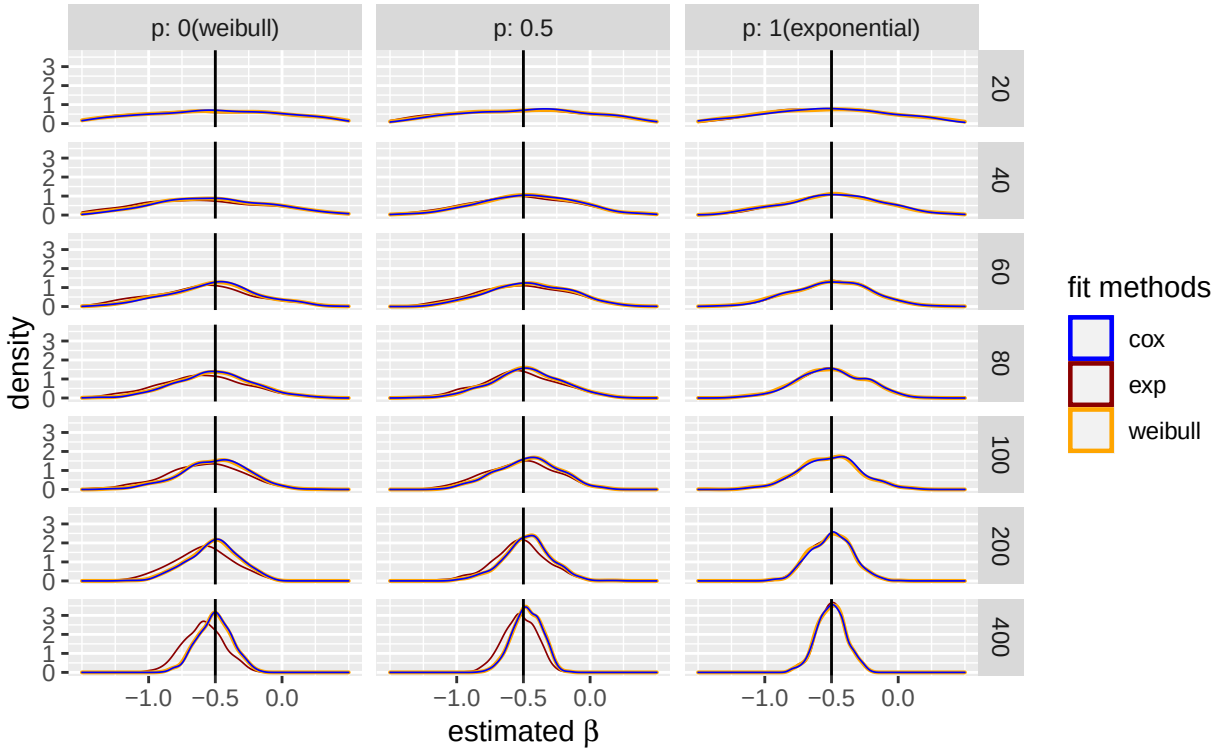

Figure 5

Figure 6-8 show some slightly different results from the previous simulation. In this case the exponential model has the poorest performance in both accuracy and efficiancy, regardless of sample size.

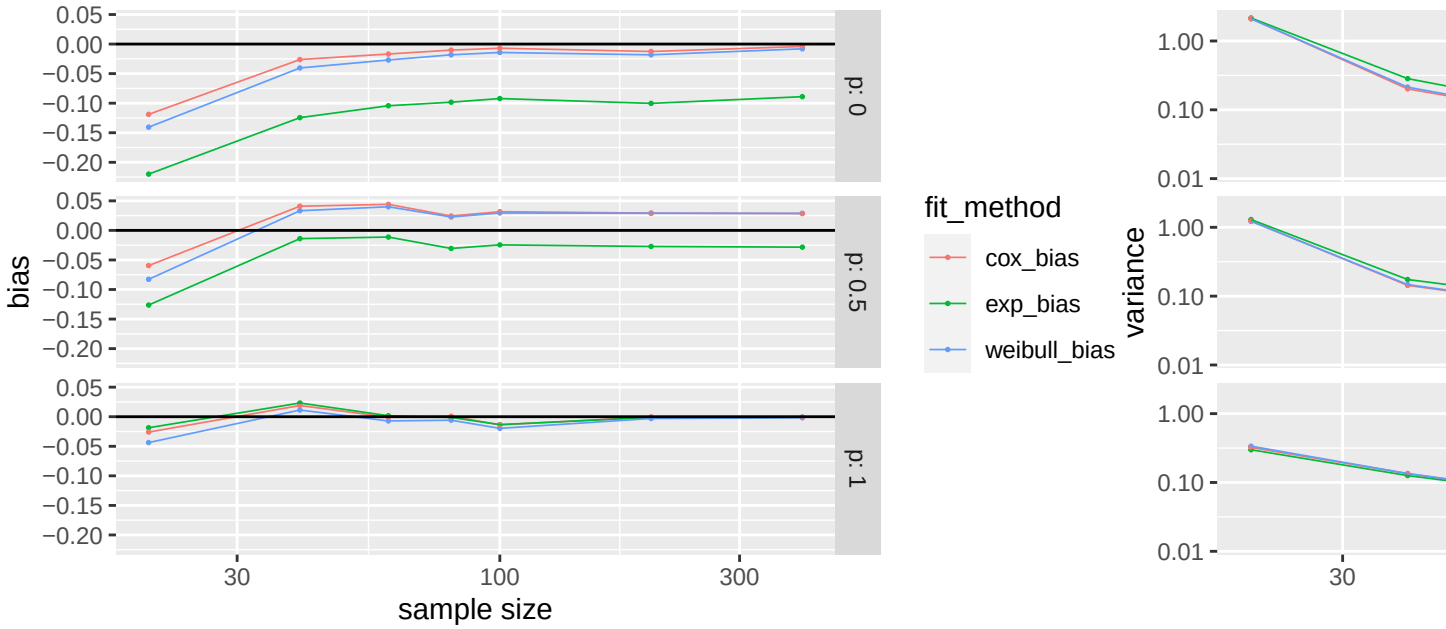

Figure 6

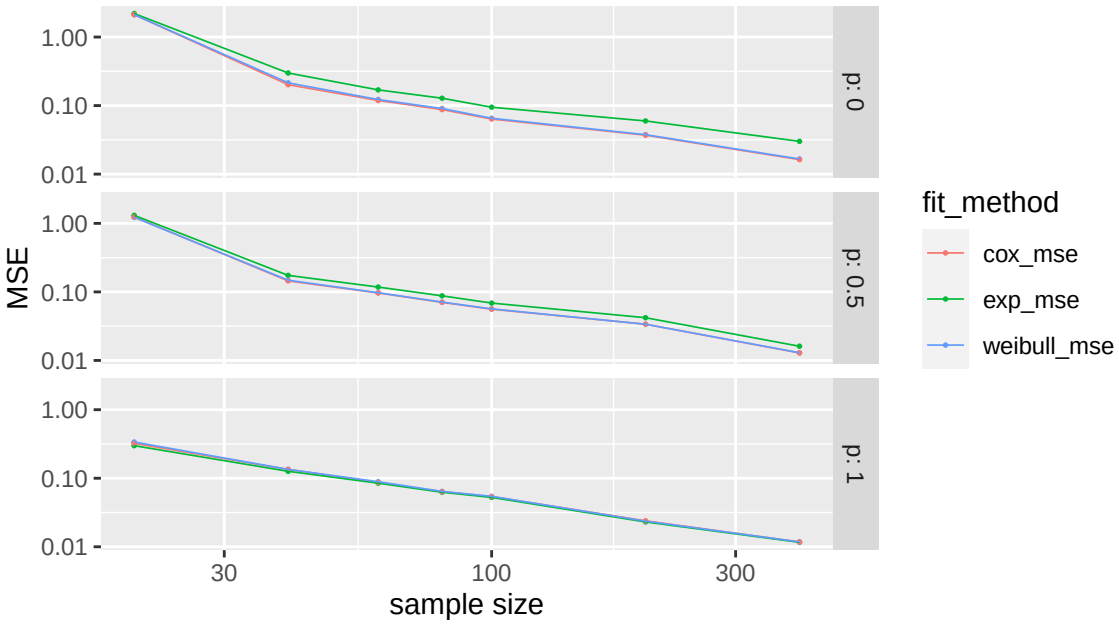

Figure 8

Finally, Figure 9 shows the results of the simulation from a Gompertz distribution, with different proportions of data contaminated by an exponential distribution. At large sample sizes, the Cox model performs the best apparently, because the true distribution is neither exponential nor Weibull. This shows that the Cox model has the highest robustness against misspecified baseline hazard functions, than followed by Weibull, and the most strict exponential model performs poorest when the true distribution does not match.

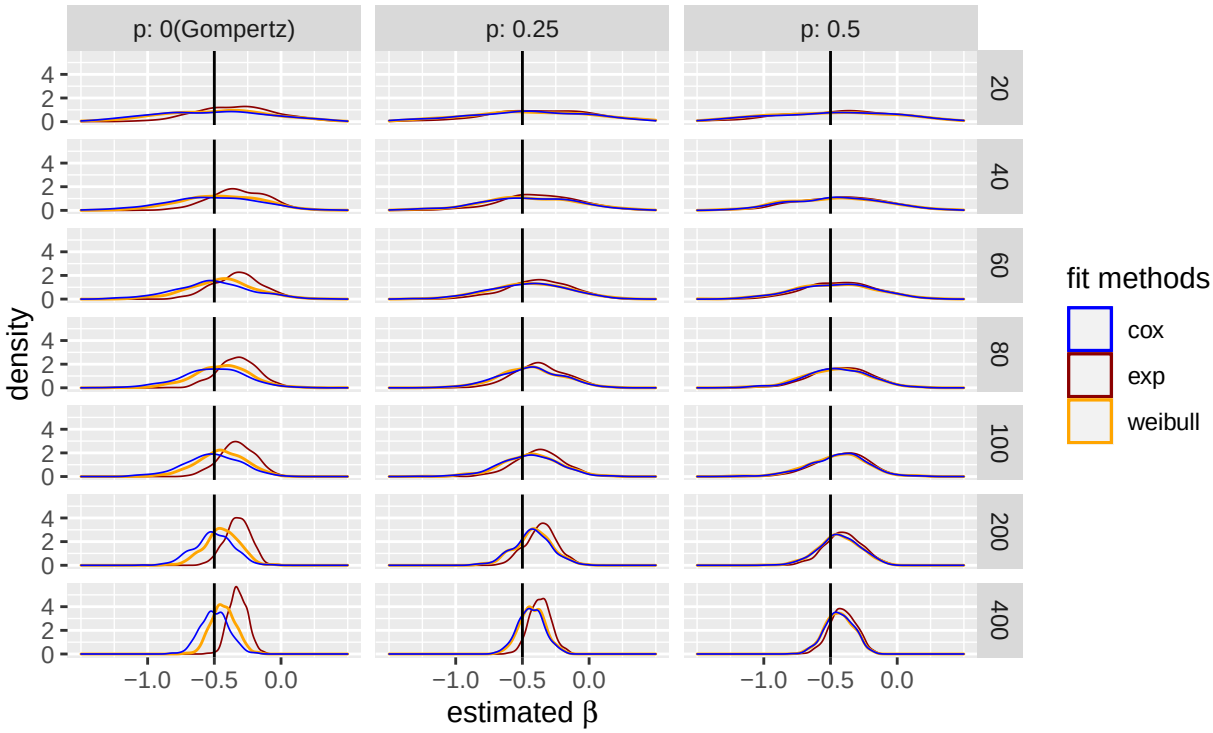

Figure 9

Figure 10-12 show that the exponential is still the most biased one among the three models, and it keeps the highest efficiency with the smallest variance, followed by the Weibull model-but when the true distribution

becomes more mixed up, the Cox model tends to outperform the Weibull model. The MSE plot shows that with small sample sizes, the exponential model has good prediction performance; while with large sample sizes, the Cox model has better prediction performance.

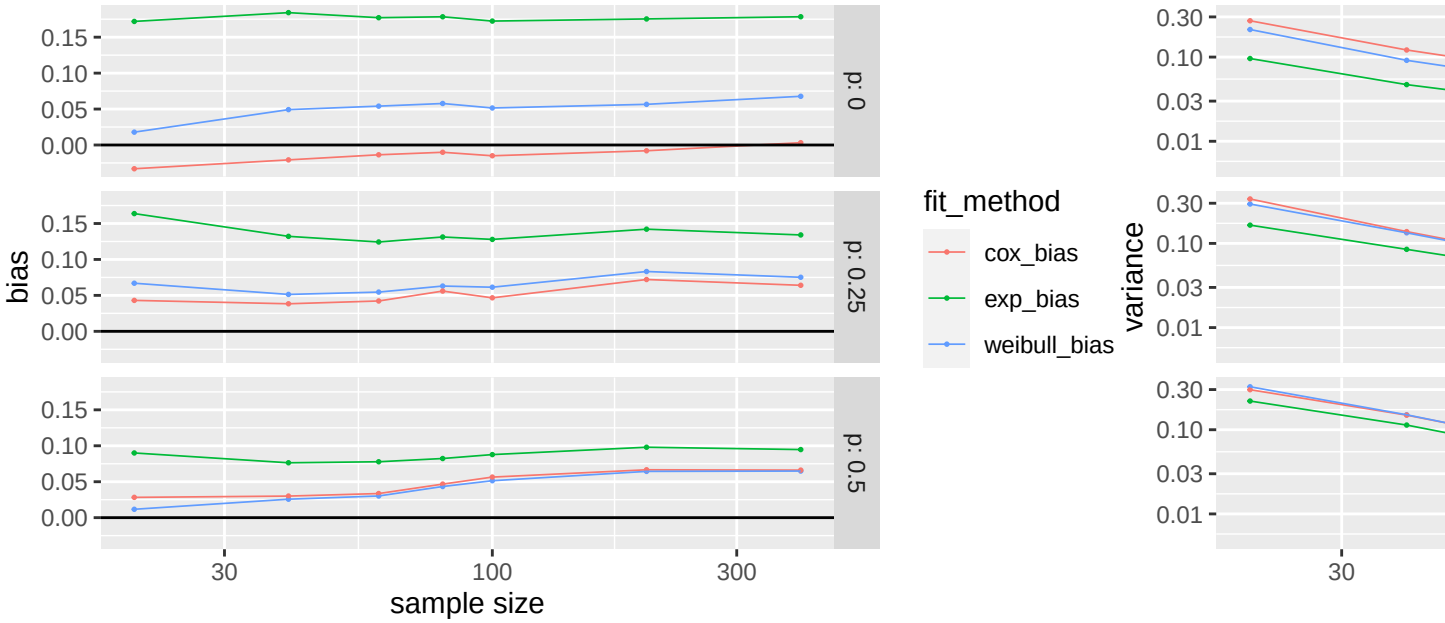

Figure 10

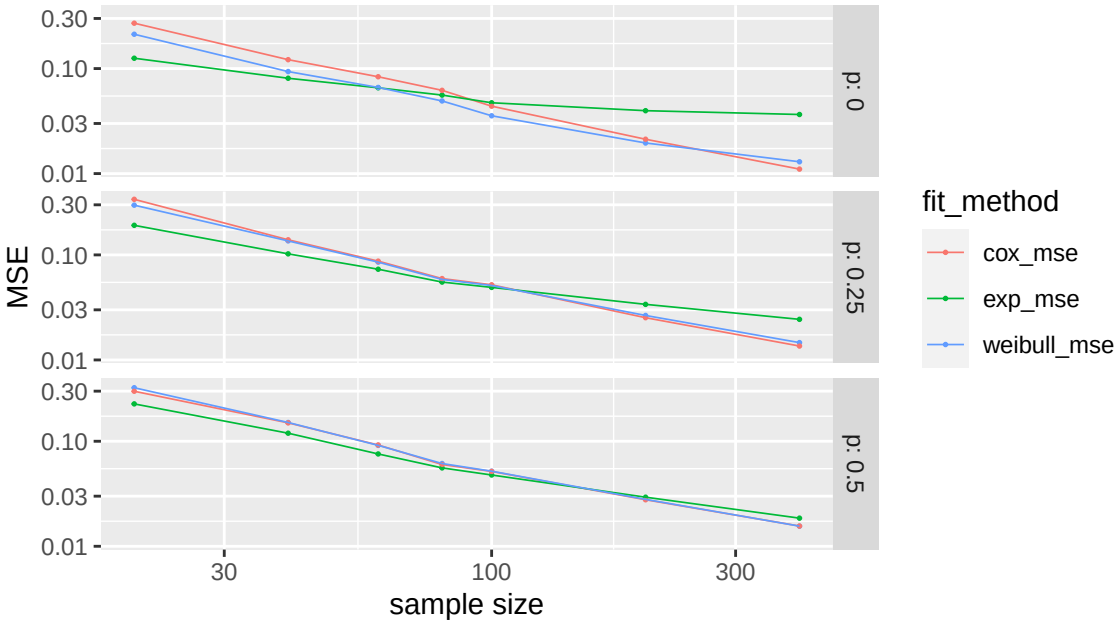

Figure 12

Conclusions and discussion

In conclusion, the exponential model is the most restrictive one, with only one parameter λ, while the Weibull has an additional γ and the Cox does not specify a certain baseline hazard function, being the most flexible and general model. Thus it is natural that the exponential model tends to have higher bias and lower variance, or we can say, lower prediction accuracy and higher efficiency compared to the other two models. Also because of the lack of freedom, the robustness against misspecified baseline hazard functions of exponential

models is the weakest. On the contrary, the Cox model are the most robust, and can fit to any kinds of underlying distribution smoothly, especially when sample size is large.

In reality, it is hard to figure out the true distribution of data. For general users, we recommend to choose a model base on the sample size. When the sample size is relatively small, an exponential model will perform the best because it is the stablest and least likely to get wild - even if you do not have enough observations, it has the largest probability to give you a quite reliable and meaningful estimate. On the other hand, if fortunately you have a large sample size, the Cox model will potentially give the best performances.

Interestingly, we find that in the simulation where $\gamma = 0.5$ in Weibull distribution, the exponential model tends to underestimate the true β, while it tends to overestimate the true β when $\gamma = 1.5$. We know that for $\gamma > 1$, the hazard function increases monotonously and for $\gamma < 1$, the hazard function decreases monotonously. The exponential model is a special case when $\gamma = 1$ and hazard is a constant number. Figure 13-14 show the different Weibull curves for $\gamma = 1.5$ and $\gamma = 0.5$ respectively. This shows that when the baseline function is lower than the true model, the exponential model will have a higher estimate of β to complement for the gap, which also indicates that it is not quite robust against misspecified baseline hazard functions.

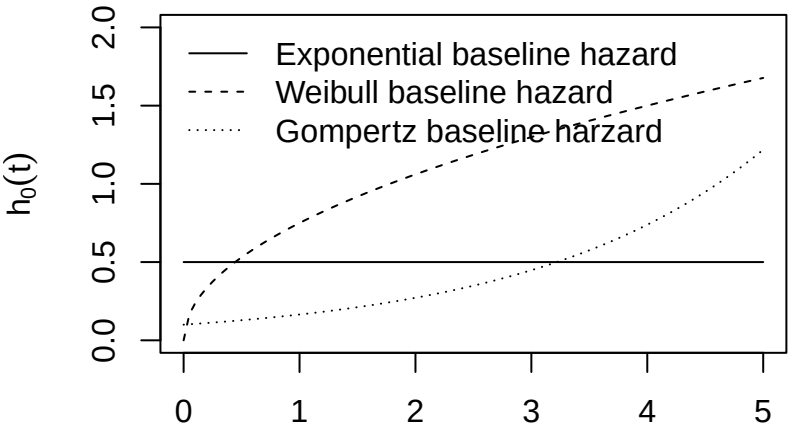

Figure 13

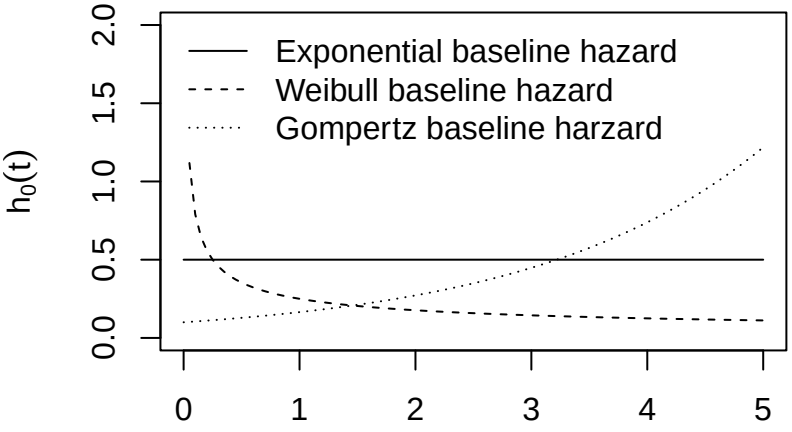

Figure 14

This simulation study has some limitations. We only test some certain values of our parameters so the generalizability of our conclusion is still questioned. For further simulation, we could try more combinations of β, λ, γ, α, and p. In addition, we could take into consideration even more types of other possible baseline

hazard functions, like log-logistic distribution, gamma distribution, log-normal distribution etc. and the mixtures of them. We could test for different time censoring as well.