Simulation_data_generate

1. Choose different cox distributions to generate data
2. Choose different beta
3. Create functions to simplify codes
4. Visualization 4.1 KM-curve vs. fitted curve 4.2 beta ~ MSE with different parameters to generate data 4.3 ...
5. discussion: description, pros and cons ...

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
##
## Attaching package: 'pracma'
## The following object is masked from 'package:purrr':
##
##     cross
## Loading required package: ggpubr
##
## Attaching package: 'survminer'
## The following object is masked from 'package:survival':
##
##     myeloma
```

Generate survival data

```
set.seed(666)

# Generate survival data
genn_dat <- function(n, lambda, beta, gamma, alpha, dist) {
  # Generate key for each observation
  id <- seq(1:n)

  # Predictor X (treatment = 1; control = 0)
  x <- rbinom(n, size = 1, prob = 0.5)

  ## --- Generate Survival Time T ---
  U <- runif(n)
  # Use exponential distribution
  expo <- function(U, lambda, x, beta) {
```

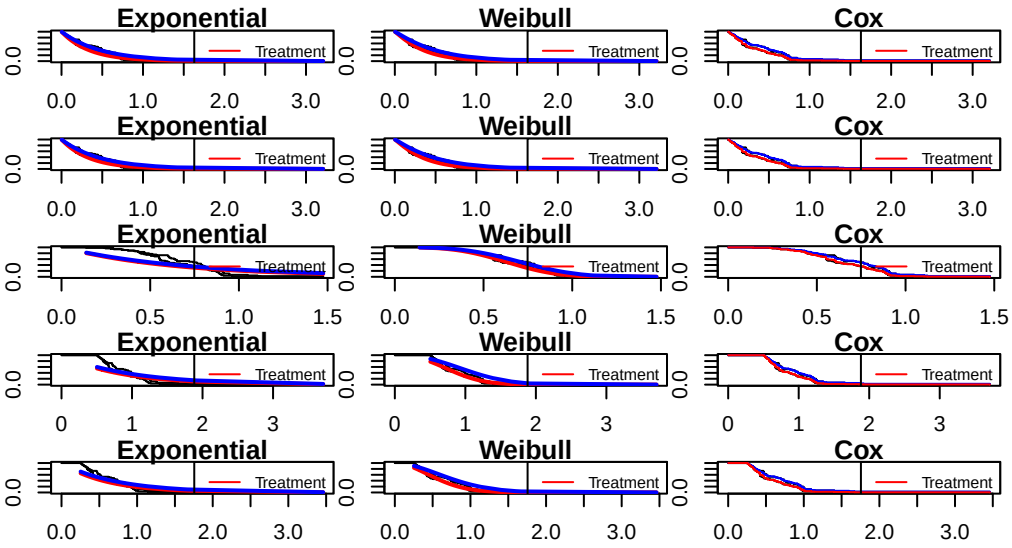
```
    -log(U) / (lambda * exp(x * beta))
  }

  # Use weibull distribution
  weibull <- function(U, lambda, x, gamma, beta) {
    (-log(U) / (lambda * exp(x * beta))) ^ (1 / gamma)
  }

  # --- To be modified (add cox function) ---
  # Use 'Cox' distribution -- gompertz
  gompertz <- function(U, alpha, lambda, x, beta) {
    (1 / alpha) * (1 - alpha * log(U) / (lambda * exp(x * beta)))
  }

  # Select different baseline functions
  if (dist == "expo") {
    surv_t <- expo(U, lambda, x, beta)}
  else if (dist == "weibull") {
    surv_t <- weibull(U, lambda, x, gamma, beta)
  }
  else {
    surv_t <- gompertz(U, alpha, lambda, x, beta)
  }

  return(
    df = data.frame(
      id = id,
      treatment = x,
      time = surv_t
    )
  )
}
```

Visualization(beta = 0.5)

```
par(oma = c(3,3,3,3))
par(mar = c(2,1,1,1))

### 1. data-exponential(lambda = 2)
## Generate Data
set.seed(666)
dat_1 <- genn_dat(n = 100, lambda = 2, beta = 0.5, dist = "expo")
s_1 <- with(dat_1, Surv(time))

par(mfrow=c(5,3))
## Exponential
fit_expo_1 = flexsurvreg(s_1 ~ as.factor(treatment), dist = 'exponential', data = dat_1)
plot(fit_expo_1, col = c('red', 'blue'), main = "Exponential", ylab = "Exponential(2)",cex.lab=1.5, fon
legend("topright", legend = c('Treatment', 'Control'), col = c('red', 'blue'), lty = 1, cex = 0.8)

## Weibull
fit_wei_1 = flexsurvreg(s_1 ~ as.factor(treatment), dist = 'weibull', data = dat_1)
plot(fit_wei_1, col = c('red', 'blue'), main = "Weibull")
```

```r
legend("topright", legend = c('Treatment', 'Control'), col = c('red', 'blue'), lty = 1, cex = 0.8)

## Cox
fit_km_1 = survfit(s_1 ~ treatment, data = dat_1)
plot(fit_km_1, conf.int = F, main = "Cox")
fit_cox_1 = coxph(s_1 ~ as.factor(treatment), data = dat_1)
lines(survfit(fit_cox_1, newdata = data.frame(treatment = 0)), col = "blue", conf.int = F)
lines(survfit(fit_cox_1, newdata = data.frame(treatment = 1)), col = "red", conf.int = F)
legend("topright", legend = c('Treatment', 'Control'), col = c('red', 'blue'), lty = 1, cex = 0.8)

### 2. data-weibull(lambda = 2, gamma = 1)
## Generate Data
set.seed(666)
dat_2 <- genn_dat(n = 100, lambda = 2, beta = 0.5, gamma = 1, dist = "weibull")
s_2 = with(dat_2, Surv(time))

## Exponential
fit_expo_2 = flexsurvreg(s_2 ~ as.factor(treatment), dist = 'exponential', data = dat_2)
plot(fit_expo_2, col = c('red', 'blue'), main = "Exponential", ylab = "Weibull(1)", cex.lab=1.5, font.la
legend("topright", legend = c('Treatment', 'Control'), col = c('red', 'blue'), lty = 1, cex = 0.8)

## Weibull
fit_wei_2 = flexsurvreg(s_2 ~ as.factor(treatment), dist = 'weibull', data = dat_2)
plot(fit_wei_2, col = c('red', 'blue'), main = "Weibull")
legend("topright", legend = c('Treatment', 'Control'), col = c('red', 'blue'), lty = 1, cex = 0.8)

## Cox
fit_km_2 = survfit(s_2 ~ treatment, data = dat_2)
plot(fit_km_2, conf.int = F, main = "Cox")
fit_cox_2 = coxph(s_2 ~ as.factor(treatment), data = dat_2)
lines(survfit(fit_cox_2, newdata = data.frame(treatment = 0)), col = "blue", conf.int = F)
lines(survfit(fit_cox_2, newdata = data.frame(treatment = 1)), col = "red", conf.int = F)
legend("topright", legend = c('Treatment', 'Control'), col = c('red', 'blue'), lty = 1, cex = 0.8)

### 3. data-weibull(lambda = 2, gamma = 3)
## Generate Data
set.seed(666)
dat_4 <- genn_dat(n = 100, lambda = 2, beta = 0.5, gamma = 3, dist = "weibull")
s_4 = with(dat_4, Surv(time))

## Exponential
fit_expo_4 = flexsurvreg(s_4 ~ as.factor(treatment), dist = 'exponential', data = dat_4)
plot(fit_expo_4, col = c('red', 'blue'), main = "Exponential", ylab = "Weibull(3)", cex.lab=1.5, font.la
legend("topright", legend = c('Treatment', 'Control'), col = c('red', 'blue'), lty = 1, cex = 0.8)

## Weibull
fit_wei_4 = flexsurvreg(s_4 ~ as.factor(treatment), dist = 'weibull', data = dat_4)
plot(fit_wei_4, col = c('red', 'blue'), main = "Weibull")
legend("topright", legend = c('Treatment', 'Control'), col = c('red', 'blue'), lty = 1, cex = 0.8)

## Cox
fit_km_4 = survfit(s_4 ~ treatment, data = dat_4)
plot(fit_km_4, conf.int = F, main = "Cox")
```

```r
fit_cox_4 = coxph(s_4 ~ as.factor(treatment), data = dat_4)
lines(survfit(fit_cox_4, newdata = data.frame(treatment = 0)), col = "blue", conf.int = F)
lines(survfit(fit_cox_4, newdata = data.frame(treatment = 1)), col = "red", conf.int = F)
legend("topright", legend = c('Treatment', 'Control'), col = c('red', 'blue'), lty = 1, cex = 0.8)

### 4. data-gompertz(alpha = 2, lambda = 2)
## Generate Data
set.seed(666)
dat_3 <- genn_dat(n = 100, lambda = 2, beta = 0.5, alpha = 2, dist = "gompertz")
s_3 = with(dat_3, Surv(time))

## Exponential
fit_expo_3 = flexsurvreg(s_3 ~ as.factor(treatment), dist = 'exponential', data = dat_3)
plot(fit_expo_3, col = c('red', 'blue'), main = "Exponential", ylab = "Gompertz(2)", cex.lab=1.5, font.l
legend("topright", legend = c('Treatment', 'Control'), col = c('red', 'blue'), lty = 1, cex = 0.8)

## Weibull
fit_wei_3 = flexsurvreg(s_3 ~ as.factor(treatment), dist = 'weibull', data = dat_3)
plot(fit_wei_3, col = c('red', 'blue'), main = "Weibull", cex.lab=1.5)
legend("topright", legend = c('Treatment', 'Control'), col = c('red', 'blue'), lty = 1, cex = 0.8)

## Cox
fit_km = survfit(s_3 ~ treatment, data = dat_3)
plot(fit_km, conf.int = F, main = "Cox", cex.lab=1.5)
fit_cox_3 = coxph(s_3 ~ as.factor(treatment), data = dat_3)
lines(survfit(fit_cox_3, newdata = data.frame(treatment = 0)), col = "blue", conf.int = F)
lines(survfit(fit_cox_3, newdata = data.frame(treatment = 1)), col = "red", conf.int = F)
legend("topright", legend = c('Treatment', 'Control'), col = c('red', 'blue'), lty = 1, cex = 0.8)

### 5. data-gompertz(lambda = 2, alpha = 4)
## Generate Data
set.seed(666)
dat_5 <- genn_dat(n = 100, lambda = 2, beta = 0.5, alpha = 4, dist = "gompertz")
s_5 = with(dat_5, Surv(time))

## Exponential
fit_expo_5 = flexsurvreg(s_5 ~ as.factor(treatment), dist = 'exponential', data = dat_5)
plot(fit_expo_5, col = c('red', 'blue'), main = "Exponential", ylab = "Gompertz(4)", xlab = "t", cex.lab
legend("topright", legend = c('Treatment', 'Control'), col = c('red', 'blue'), lty = 1, cex = 0.8)

## Weibull
fit_wei_5 = flexsurvreg(s_5 ~ as.factor(treatment), dist = 'weibull', data = dat_5)
plot(fit_wei_5, col = c('red', 'blue'), main = "Weibull", xlab = "t", cex.lab=1.5)
legend("topright", legend = c('Treatment', 'Control'), col = c('red', 'blue'), lty = 1, cex = 0.8)

## Cox
fit_km = survfit(s_5 ~ treatment, data = dat_5)
plot(fit_km, conf.int = F, main = "Cox", xlab = "t", cex.lab=1.5)
fit_cox_5 = coxph(s_5 ~ as.factor(treatment), data = dat_5)
lines(survfit(fit_cox_5, newdata = data.frame(treatment = 0)), col = "blue", conf.int = F)
lines(survfit(fit_cox_5, newdata = data.frame(treatment = 1)), col = "red", conf.int = F)
legend("topright", legend = c('Treatment', 'Control'), col = c('red', 'blue'), lty = 1, cex = 0.8)
```



3.MSE vs. beta

```
# Simulation results
sim_result <- function(iteration, n, lambda, beta, gamma, alpha, dist) {
  # Store estimated beta
  result <- tibble(
  expo_beta = rep(NA, iteration),
  weibull_beta = rep(NA, iteration),
  cox_beta = rep(NA, iteration)
  )

  for(i in 1:iteration) {
    # Generate data
    data <- genn_dat(n, lambda, beta, gamma, alpha, dist)

    # Use exponential model
    fit_expo <- survreg(Surv(data$time) ~ data$treatment, dist = "exponential")
    result$expo_beta[i] <- -fit_expo$coefficients[-1]
    # Use weibull model
    fit_weibull <- survreg(Surv(data$time) ~ data$treatment, dist = "weibull")
    result$weibull_beta[i] <- -fit_weibull$coefficients[-1] / fit_weibull$scale
    # Use gompertz model
    fit_cox <- coxph(Surv(data$time) ~ data$treatment)
    result$cox_beta[i] <- fit_cox$coefficients

  }

  # Calculate MSE

  return(result)
}
```

```
# Under 1000 simulations & 100 sample sizes
# True beta = 0.5

# Exponential beta
iteration = 1000
```

```r
n = 100
beta = 0.5
expo_result <- sim_result(iteration, n, lambda = 2, beta,dist = "expo")
expo_beta <- tibble(
  expo_mse = sum((expo_result$expo_beta - beta)^2) / iteration,
  expo_mean = mean(expo_result$expo_beta),
  expo_sd = sd(expo_result$expo_beta),
  weibull_mse = sum((expo_result$weibull_beta - beta)^2) / iteration,
  weibull_mean = mean(expo_result$weibull_beta),
  weibull_sd = sd(expo_result$weibull_beta),
  cox_mse = sum((expo_result$cox_beta - beta)^2) / iteration,
  cox_mean = mean(expo_result$cox_beta),
  cox_sd = sd(expo_result$cox_beta)
)

# Weibull beta (gamma = 1)
weibull_result_gamma1 <- sim_result(iteration, n , beta, lambda = 2, gamma = 1, dist = "weibull")
weibull_beta_gamma1 <- tibble(
  expo_mse = sum((weibull_result_gamma1$expo_beta - beta)^2) / iteration,
  expo_mean = mean(weibull_result_gamma1$expo_beta),
  expo_sd = sd(weibull_result_gamma1$expo_beta),
  weibull_mse = sum((weibull_result_gamma1$weibull_beta - beta)^2) / iteration,
  weibull_mean = mean(weibull_result_gamma1$weibull_beta),
  weibull_sd = sd(weibull_result_gamma1$weibull_beta),
  cox_mse = sum((weibull_result_gamma1$cox_beta - beta)^2) / iteration,
  cox_mean = mean(weibull_result_gamma1$cox_beta),
  cox_sd = sd(weibull_result_gamma1$cox_beta)
)

# Weibull beta (gamma = 3)
weibull_result_gamma3 <- sim_result(iteration, n , beta, lambda = 2, gamma = 3, dist = "weibull")
weibull_beta_gamma3 <- tibble(
  expo_mse = sum((weibull_result_gamma3$expo_beta - beta)^2) / iteration,
  expo_mean = mean(weibull_result_gamma3$expo_beta),
  expo_sd = sd(weibull_result_gamma3$expo_beta),
  weibull_mse = sum((weibull_result_gamma3$weibull_beta - beta)^2) / iteration,
  weibull_mean = mean(weibull_result_gamma3$weibull_beta),
  weibull_sd = sd(weibull_result_gamma3$weibull_beta),
  cox_mse = sum((weibull_result_gamma3$cox_beta - beta)^2) / iteration,
  cox_mean = mean(weibull_result_gamma3$cox_beta),
  cox_sd = sd(weibull_result_gamma3$cox_beta)
)

# Gompertz beta (alpha = 2)
Gompertz_result_alpha2 <- sim_result(iteration, n , beta, lambda = 2, alpha = 2, dist = "Gompertz")
Gompertz_beta_alpha2 <- tibble(
  expo_mse = sum((Gompertz_result_alpha2$expo_beta - beta)^2) / iteration,
  expo_mean = mean(Gompertz_result_alpha2$expo_beta),
  expo_sd = sd(Gompertz_result_alpha2$expo_beta),
  weibull_mse = sum((Gompertz_result_alpha2$weibull_beta - beta)^2) / iteration,
  weibull_mean = mean(Gompertz_result_alpha2$weibull_beta),
  weibull_sd = sd(Gompertz_result_alpha2$weibull_beta),
  cox_mse = sum((Gompertz_result_alpha2$cox_beta - beta)^2) / iteration,
```

```
  cox_mean = mean(Gompertz_result_alpha2$cox_beta),
  cox_sd = sd(Gompertz_result_alpha2$cox_beta)
)

# Gompertz beta (alpha = 4)
Gompertz_result_alpha4 <- sim_result(iteration, n , beta, lambda = 2, alpha = 4, dist = "Gompertz")
Gompertz_beta_alpha4 <- tibble(
  expo_mse = sum((Gompertz_result_alpha4$expo_beta - beta)^2) / iteration,
  expo_mean = mean(Gompertz_result_alpha4$expo_beta),
  expo_sd = sd(Gompertz_result_alpha4$expo_beta),
  weibull_mse = sum((Gompertz_result_alpha4$weibull_beta - beta)^2) / iteration,
  weibull_mean = mean(Gompertz_result_alpha4$weibull_beta),
  weibull_sd = sd(Gompertz_result_alpha4$weibull_beta),
  cox_mse = sum((Gompertz_result_alpha4$cox_beta - beta)^2) / iteration,
  cox_mean = mean(Gompertz_result_alpha4$cox_beta),
  cox_sd = sd(Gompertz_result_alpha4$cox_beta)
)
```