# tp1-sur

Richard CHEAM

2024-09-26

## Import libraries

```r
library(MASS)
library(asaur)
library(survival)
library(KMsurv)
library(ggplot2)
```

## Read dataframe

```r
df <- read.csv(file = "coalition2.csv", header = TRUE)
head(df)
```

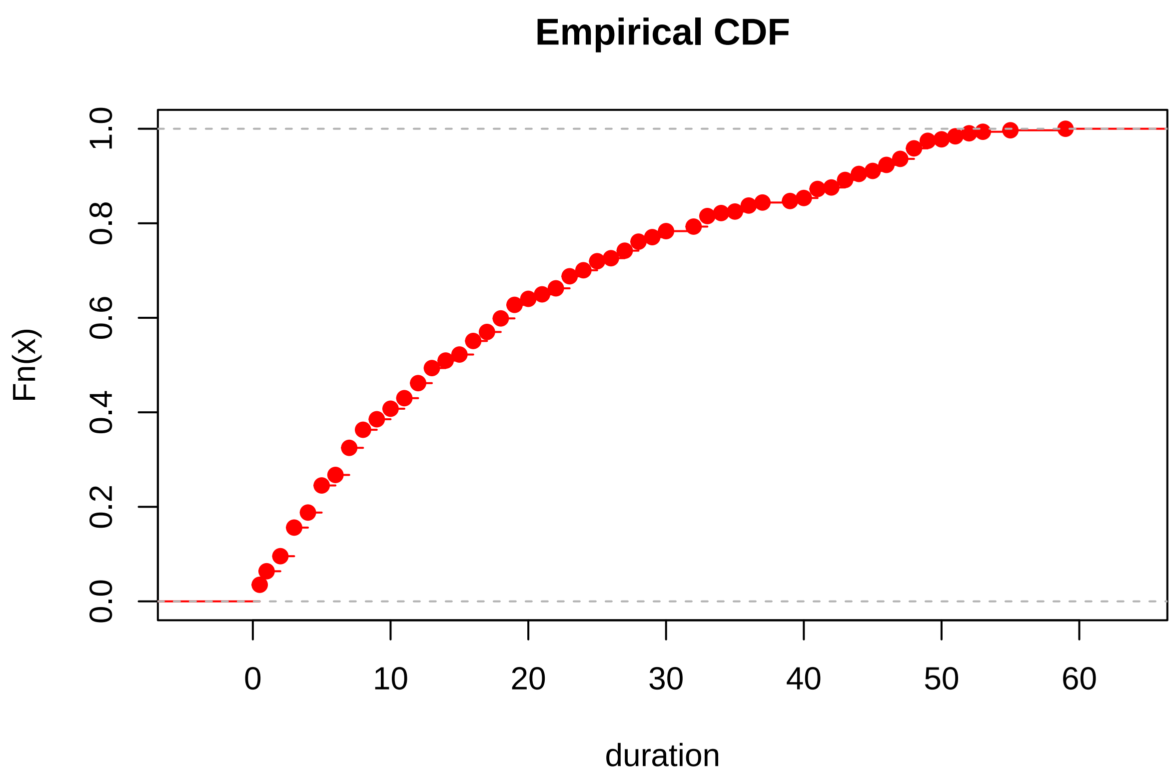
```
##   X duration ciep12 invest fract polar numst2 crisis country
## 1 1      0.5      1      1   656    11      0     24 belgium
## 2 2      3.0      1      1   656    11      1     10 belgium
## 3 3      7.0      1      1   656    11      1     24 belgium
## 4 4     20.0      1      1   656    11      1      7 belgium
## 5 5      6.0      1      1   656    11      1      7 belgium
## 6 6      7.0      1      1   634     6      1     45 belgium
```

```r
cat("Dimension of the dataframe:", dim(df))
```

```
## Dimension of the dataframe: 314 9
```

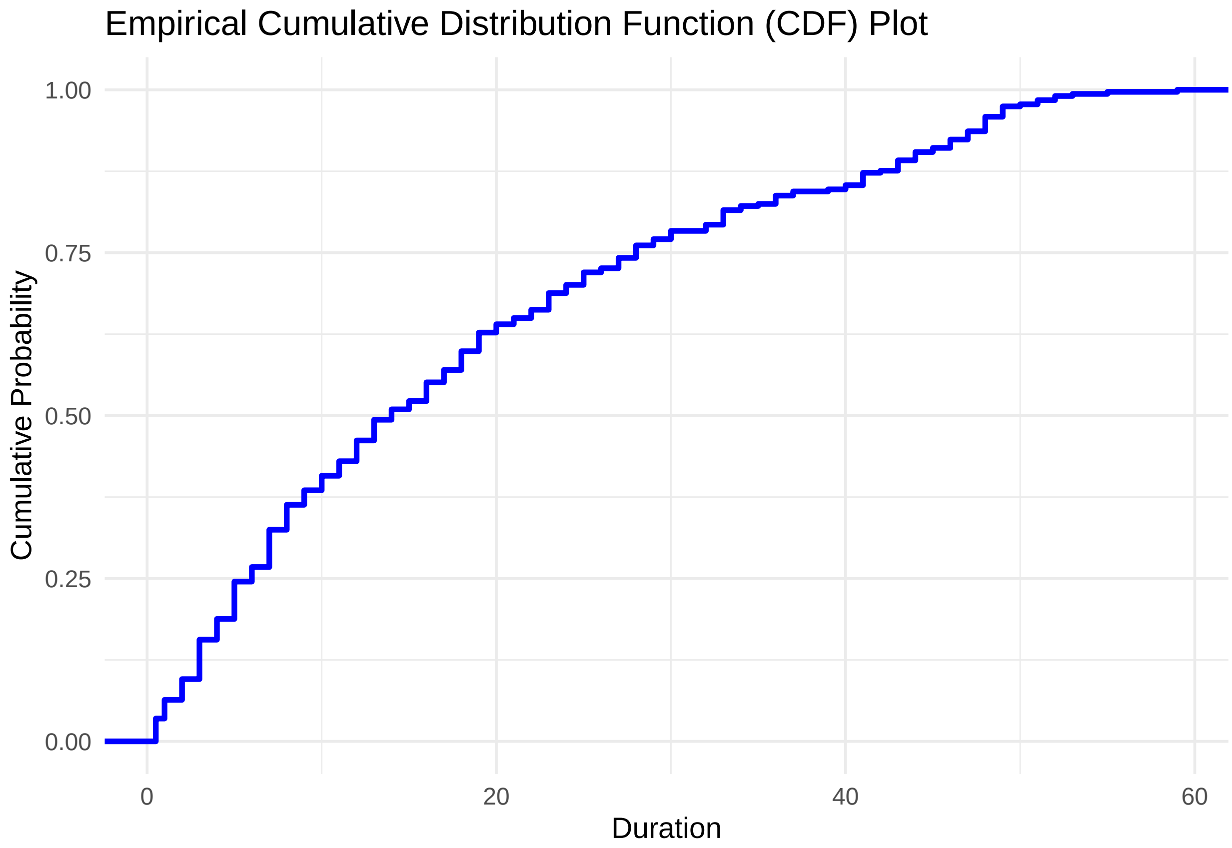
### 1. Compute and draw the empirical c.d.f. for the variable duration.

```r
X = df$duration

emp_cdf = ecdf(X)
plot(emp_cdf, col = 'red', main = 'Empirical CDF', xlab = 'duration')
```




```
ggplot(data = data.frame(duration = X), aes(x = duration)) +
  stat_ecdf(geom = "step", color = "blue", size = 1) +
  theme_minimal() +
  labs(
    title = "Empirical Cumulative Distribution Function (CDF) Plot",
    x = "Duration",
    y = "Cumulative Probability"
  )
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




**2. Compute maximum likelihood estimator for the variable duration for 4 different families.**

Parameters of each family are shown below:

```
fit_normal <- fitdistr(X, "normal")
fit_weibull <- fitdistr(X, "weibull")
fit_lognormal <- fitdistr(X, "lognormal")
fit_gamma <- fitdistr(X, "gamma")
```

```
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced

## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced

## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced

## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced

## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
```

```
fit_normal
```

```
##       mean          sd
##   18.4378981   15.1328027
##  ( 0.8539935) ( 0.6038646)
```

```r
fit_weibull
```

```
##       shape         scale
##    1.13835393   19.28566789
##  ( 0.05190129) ( 1.00563891)
```

```r
fit_lognormal
```

```
##     meanlog       sdlog
##   2.43552795   1.14711279
##  (0.06473532) (0.04577478)
```

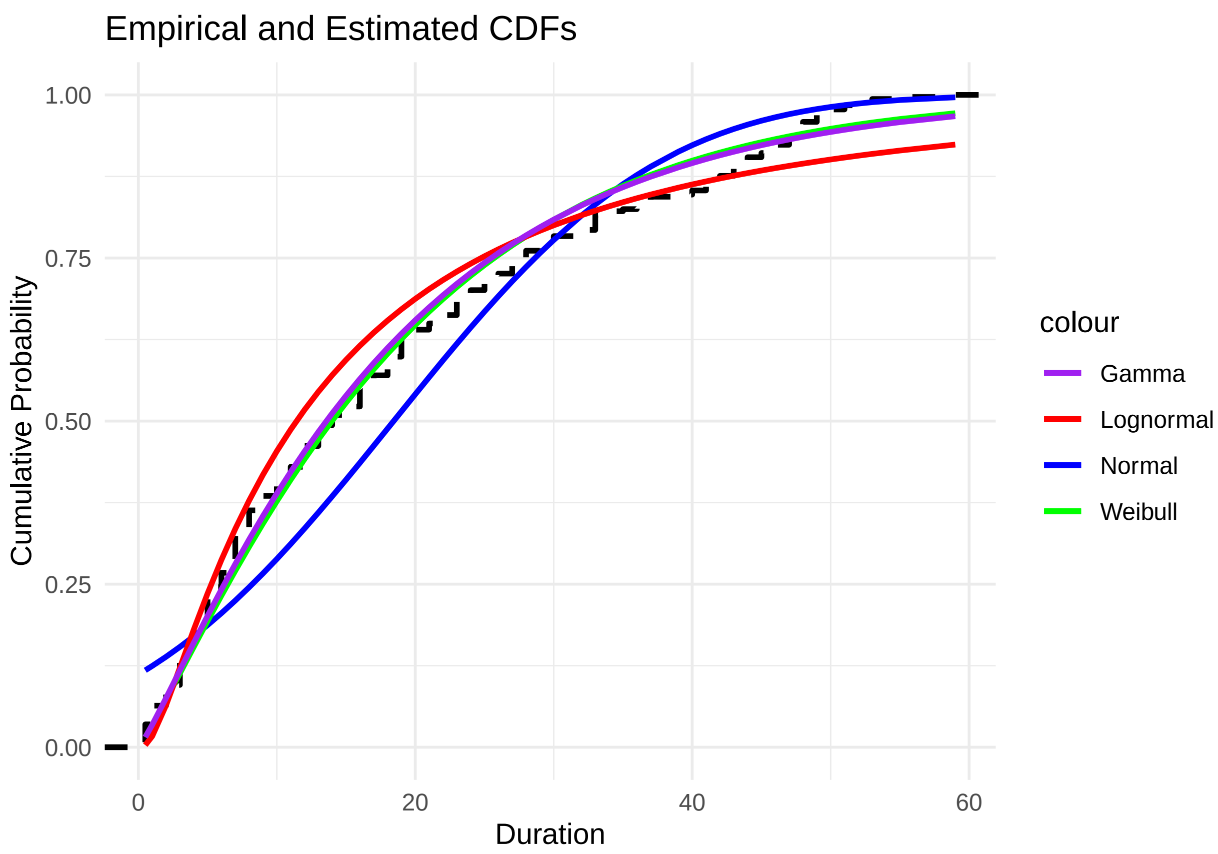
```r
fit_gamma
```

```
##       shape         rate
##   1.182362418   0.064118696
##  (0.084281787) (0.005652631)
```

### 3. Draw the estimated c.d.f. in each family on the same plot as the empirical c.d.f

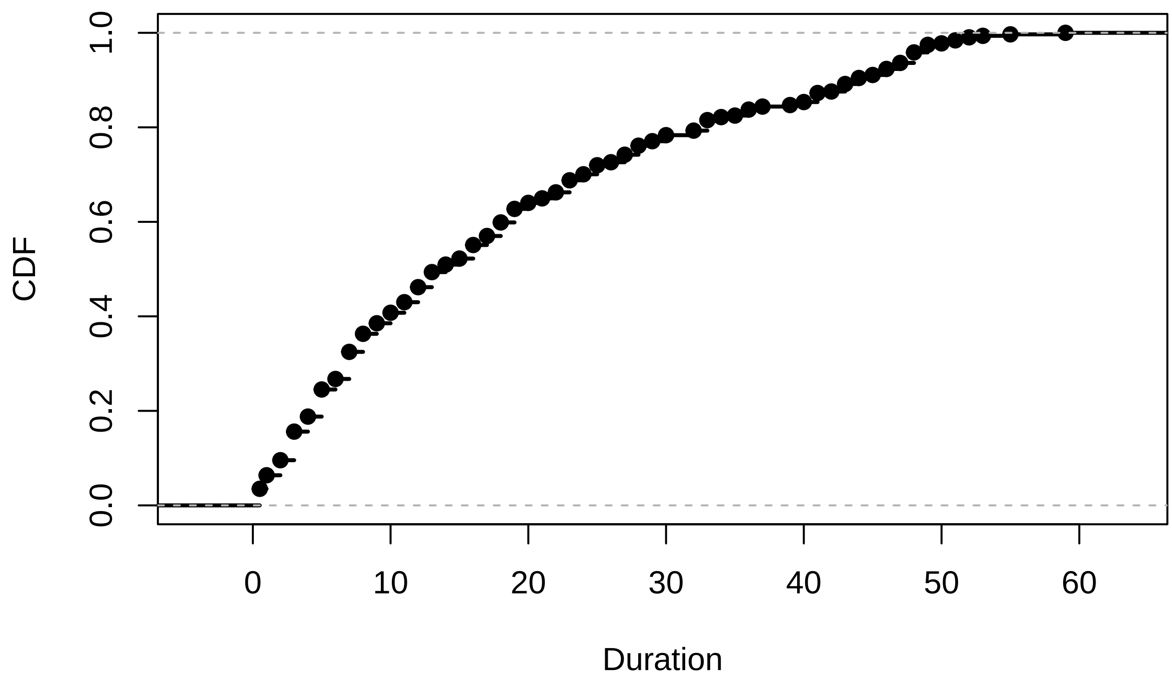
```r
cdf_normal <- pnorm(X, mean = fit_normal$estimate['mean'], sd = fit_normal$estimate['sd'])

cdf_weibull <- pweibull(X, shape = fit_weibull$estimate['shape'], scale = fit_weibull$estimate['scale']

cdf_lognormal <- plnorm(X, meanlog = fit_lognormal$estimate['meanlog'], sdlog = fit_lognormal$estimate[

cdf_gamma <- pgamma(X, shape = fit_gamma$estimate['shape'], rate = fit_gamma$estimate['rate'])
```

```r
# Create a data frame for plotting
cdf_data <- data.frame(
  x = X,
  normal = cdf_normal,
  weibull = cdf_weibull,
  lognormal = cdf_lognormal,
  gamma = cdf_gamma
)

# Plot the empirical CDF and estimated CDFs
ggplot() +
  stat_ecdf(data = df, aes(x = duration), color = 'black', size = 1, linetype = "dashed") +
  geom_line(data = cdf_data, aes(x = x, y = normal, color = 'Normal'), size = 1) +
  geom_line(data = cdf_data, aes(x = x, y = weibull, color = 'Weibull'), size = 1) +
  geom_line(data = cdf_data, aes(x = x, y = lognormal, color = 'Lognormal'), size = 1) +
  geom_line(data = cdf_data, aes(x = x, y = gamma, color = 'Gamma'), size = 1) +
  labs(x = "Duration", y = "Cumulative Probability", title = "Empirical and Estimated CDFs") +
  scale_color_manual(values = c('Empirical' = 'black', 'Normal' = 'blue', 'Weibull' = 'green', 'Lognorma
  theme_minimal()
```

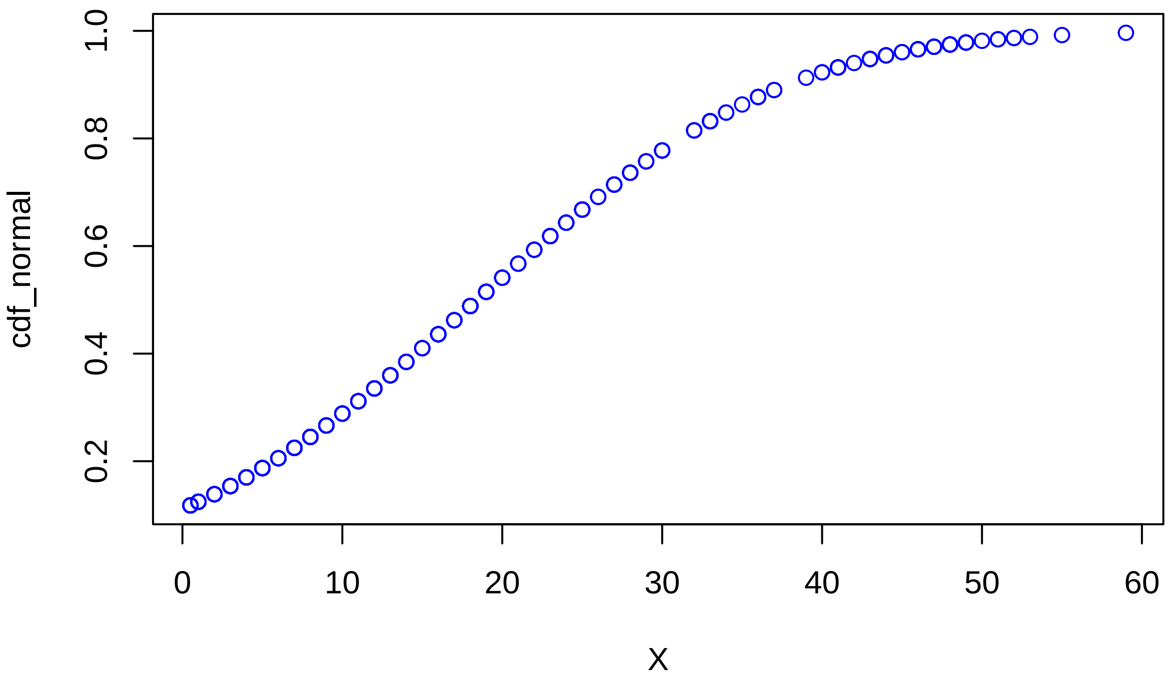



```
# Plot the empirical CDF
plot(emp_cdf, main = "Empirical and Estimated CDFs", xlab = "Duration", ylab = "CDF", col = "black", lw
```

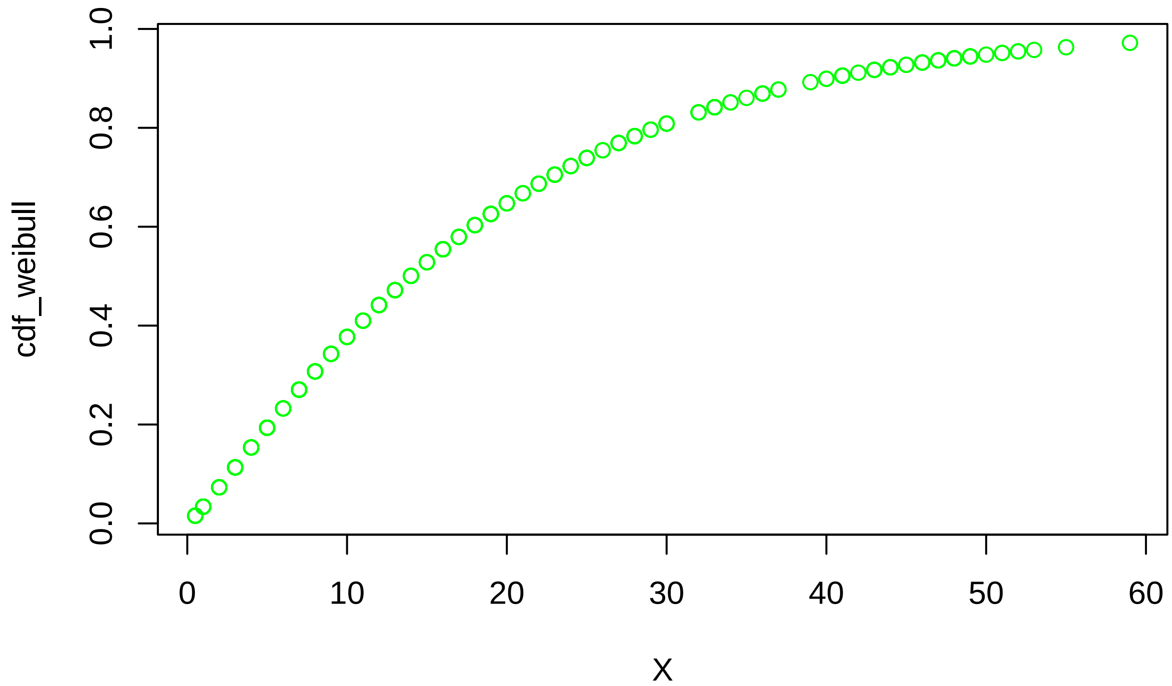
Empirical and Estimated CDFs



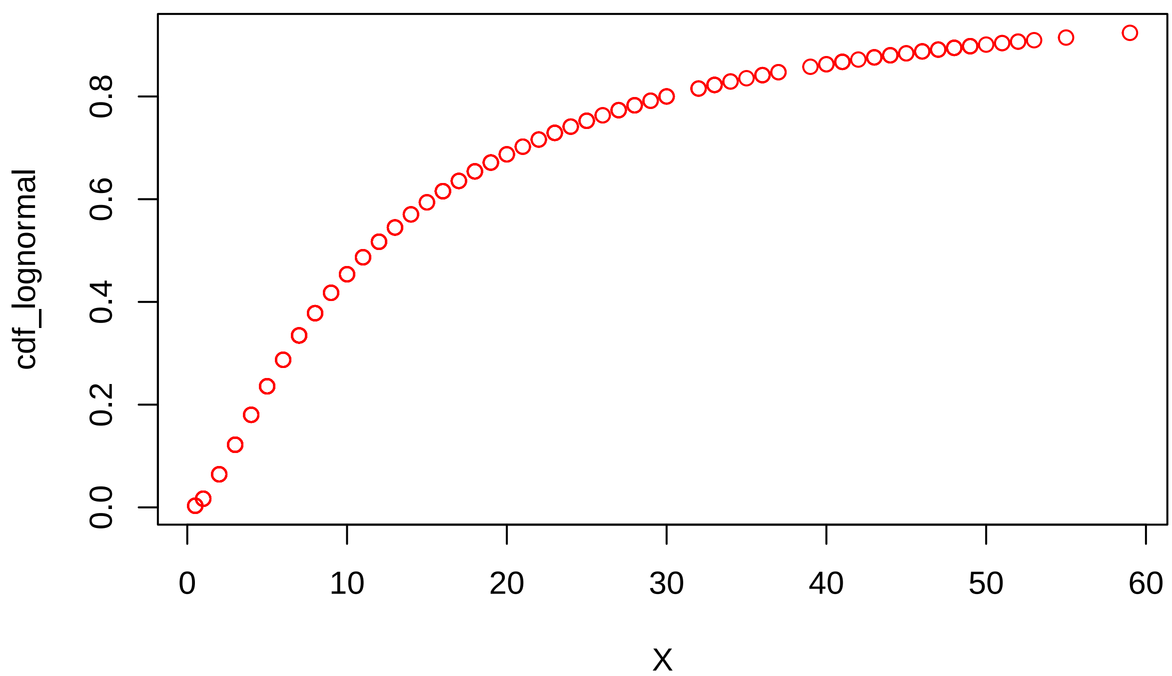

```r
# Add the estimated CDFs using lines()
plot(X, cdf_normal, col = "blue")     # Normal
```



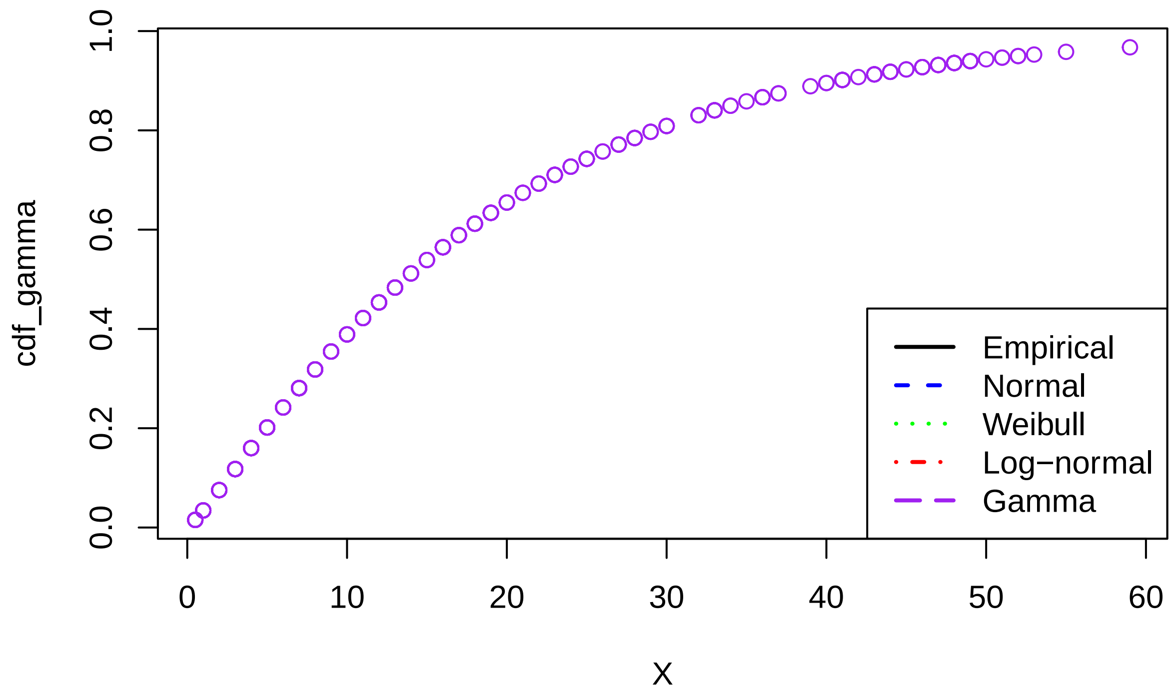

```r
plot(X, cdf_weibull, col = "green")  # Weibull
```




```r
plot(X, cdf_lognormal, col = "red")  # Log-normal
```




```r
plot(X, cdf_gamma, col = "purple")   # Gamma

# Add a legend
legend("bottomright", legend = c("Empirical", "Normal", "Weibull", "Log-normal", "Gamma"),
       col = c("black", "blue", "green", "red", "purple"), lwd = 2, lty = c(1, 2, 3, 4, 5))
```




## 2. Your own code for the Kaplan-Meier estimator

### 1. Develop a function to compute the Kaplan-Meier estimator that takes as inputs

```
KM_est <- function(time, sigma){
  time = order(time)
  n = length(sigma)
  for (indx in 1:n){
    tmp = 1 - sigma[indx]/(n - (indx - 1))
    res = res * tmp
  }
  return (res)
}

KM_est <- function(time, event) {
  # Order the times and get the unique times
  order_index <- order(time)
  time <- time[order_index]
  event <- event[order_index]

  # Initialize variables
  n <- length(time)
```

```r
  res <- numeric(n)  # This will hold the survival probabilities
  res[1] <- 1        # Start with S(0) = 1

  # Compute the survival probabilities
  for (indx in 1:n) {
    if (event[indx] == 1) {  # Event occurred
      # Calculate the survival probability
      tmp <- 1 - sum(event[1:indx]) / (n - (indx - 1))
      res[indx] <- res[indx - 1] * tmp  # Update the survival probability
    } else {
      res[indx] <- res[indx - 1]  # No change in probability if censored
    }
  }

  return(res)
}
```

**2. Consider the pharmocoSmoking data (available in package asaur), compare the results of your code to the one of the function survfit of package survival.**

```r
data("pharmacoSmoking")
df <- pharmacoSmoking
head(df)
```

```
##    id ttr relapse         grp age gender     race employment yearsSmoking
## 1  21 182       0   patchOnly  36   Male    white         ft           26
## 2 113  14       1   patchOnly  41   Male    white      other           27
## 3  39   5       1 combination  25 Female    white      other           12
## 4  80  16       1 combination  54   Male    white         ft           39
## 5  87   0       1 combination  45   Male    white      other           30
## 6  29 182       0 combination  43   Male hispanic         ft           30
##   levelSmoking ageGroup2 ageGroup4 priorAttempts longestNoSmoke
## 1        heavy     21-49     35-49             0              0
## 2        heavy     21-49     35-49             3             90
## 3        heavy     21-49     21-34             3             21
## 4        heavy       50+     50-64             0              0
## 5        heavy     21-49     35-49             0              0
## 6        heavy     21-49     35-49             2           1825
```

```r
dim(df)
```

```
## [1] 125  14
```

```r
KM_fit <- survfit(Surv(df$ttr, df$relapse) ~ 1)
summary(KM_fit)
```

```
## Call: survfit(formula = Surv(df$ttr, df$relapse) ~ 1)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
```

```
##      0     125       12     0.904  0.0263          0.854          0.957
##      1     113        5     0.864  0.0307          0.806          0.926
##      2     108        6     0.816  0.0347          0.751          0.887
##      3     102        1     0.808  0.0352          0.742          0.880
##      4     101        3     0.784  0.0368          0.715          0.860
##      5      98        2     0.768  0.0378          0.697          0.846
##      6      96        1     0.760  0.0382          0.689          0.839
##      7      95        1     0.752  0.0386          0.680          0.832
##      8      94        3     0.728  0.0398          0.654          0.810
##     10      91        1     0.720  0.0402          0.645          0.803
##     12      90        2     0.704  0.0408          0.628          0.789
##     14      88        7     0.648  0.0427          0.569          0.737
##     15      81        4     0.616  0.0435          0.536          0.707
##     16      77        1     0.608  0.0437          0.528          0.700
##     20      76        1     0.600  0.0438          0.520          0.692
##     21      75        2     0.584  0.0441          0.504          0.677
##     25      73        1     0.576  0.0442          0.496          0.669
##     28      72        3     0.552  0.0445          0.471          0.646
##     30      69        3     0.528  0.0447          0.447          0.623
##     40      66        1     0.520  0.0447          0.439          0.615
##     42      65        1     0.512  0.0447          0.431          0.608
##     45      64        1     0.504  0.0447          0.424          0.600
##     49      63        1     0.496  0.0447          0.416          0.592
##     50      62        1     0.488  0.0447          0.408          0.584
##     56      61        5     0.448  0.0445          0.369          0.544
##     60      56        2     0.432  0.0443          0.353          0.528
##     63      54        2     0.416  0.0441          0.338          0.512
##     65      52        1     0.408  0.0440          0.330          0.504
##     75      51        1     0.400  0.0438          0.323          0.496
##     77      50        2     0.384  0.0435          0.308          0.479
##     80      48        1     0.376  0.0433          0.300          0.471
##     84      47        1     0.368  0.0431          0.292          0.463
##    100      46        1     0.360  0.0429          0.285          0.455
##    105      45        1     0.352  0.0427          0.277          0.447
##    110      44        1     0.344  0.0425          0.270          0.438
##    140      43        4     0.312  0.0414          0.240          0.405
##    155      39        1     0.304  0.0411          0.233          0.396
##    170      38        2     0.288  0.0405          0.219          0.379
```

```r
print(KM_fit$surv)
```

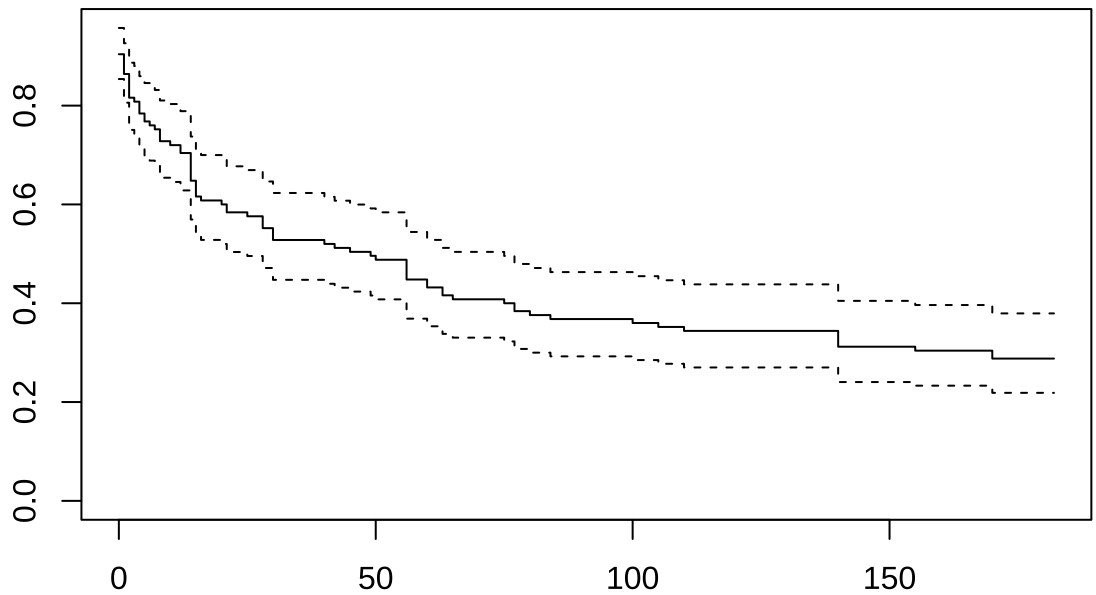
```
##  [1] 0.904 0.864 0.816 0.808 0.784 0.768 0.760 0.752 0.728 0.720 0.704 0.648
## [13] 0.616 0.608 0.600 0.584 0.576 0.552 0.528 0.520 0.512 0.504 0.496 0.488
## [25] 0.448 0.432 0.416 0.408 0.400 0.384 0.376 0.368 0.360 0.352 0.344 0.312
## [37] 0.304 0.288 0.288
```

```r
cat("\n")
```

```r
print(KM_fit$time)
```

```
##  [1]   0   1   2   3   4   5   6   7   8  10  12  14  15  16  20  21  25  28  30
## [20]  40  42  45  49  50  56  60  63  65  75  77  80  84 100 105 110 140 155 170
## [39] 182
```

```r
plot(KM_fit)
```




### 3. Compute the Greenwood estimator of the variance of the Kaplan-Meier estimator

```r
greenwood_est <- function(KM_est, nb_events, nb_at_risk, time){
  num_ = nb_events
  den_ = nb_at_risk * (nb_at_risk - nb_events)
  tmp = sum(num_/den_ < unique(time))
  return (KM_est**2 * tmp)
}

greenwood_est(KM_fit$surv, KM_fit$n.event, KM_fit$n.risk, KM_fit$time)
```

```
##  [1] 31.054208 28.366848 25.302528 24.808832 23.356928 22.413312 21.948800
##  [8] 21.489152 20.139392 19.699200 18.833408 15.956352 14.419328 14.047232
## [15] 13.680000 12.960128 12.607488 11.578752 10.593792 10.275200  9.961472
## [22]  9.652608  9.348608  9.049472  7.626752  7.091712  6.576128  6.325632
## [29]  6.080000  5.603328  5.372288  5.146112  4.924800  4.708352  4.496768
## [36]  3.699072  3.511808  3.151872  3.151872
```

## Ex3. Left-truncated and right-censored data

```r
data("channing")
df <- channing
head(df)
```

```
##   obs death ageentry  age time gender
## 1   1     1     1042 1172  130      2
## 2   2     1      921 1040  119      2
## 3   3     1      885 1003  118      2
## 4   4     1      901 1018  117      2
## 5   5     1      808  932  124      2
## 6   6     1      915 1004   89      2
```

```r
dim(df)
```

```
## [1] 462   6
```

At age 901 how many residents are under observation and still alive