Homework 1, skade2

Anna Sailegtim, Jens Fischer, Peter Garde

2024-05-07

Library

R exercise 1

We calculate the log likelihood where deductables is taken care of.

```
log_likelihood_lognormal = function(theta){
  return( -sum( log(dlnorm(claim$Clr, theta[1], theta[2]))
              - log(plnorm(claim$Dedr, theta[1], theta[2], lower.tail = F)) ) )
}

log_likelihood_weibull = function(theta){
  return( -sum( log(dweibull(claim$Clr, theta[1], theta[2]))
            - log(pweibull(claim$Dedr, theta[1], theta[2], lower.tail = F)) ) )
}
```

Pareto parametrisation Non-Life insurance mathematics, Mikosch 2009.

```
f_pareto = function(x, kappa, alpha){
  return( (alpha / (kappa + x))*(kappa / (kappa + x))^alpha )
}

F_bar_pareto = function(x, kappa, alpha){
  return( (kappa / (kappa + x))^alpha )
}

log_likelihood_pareto = function(theta){
  return( -sum( log(f_pareto(claim$Clr, theta[1], theta[2]))
                - log(F_bar_pareto(claim$Dedr, theta[1], theta[2])) ) )
}
```

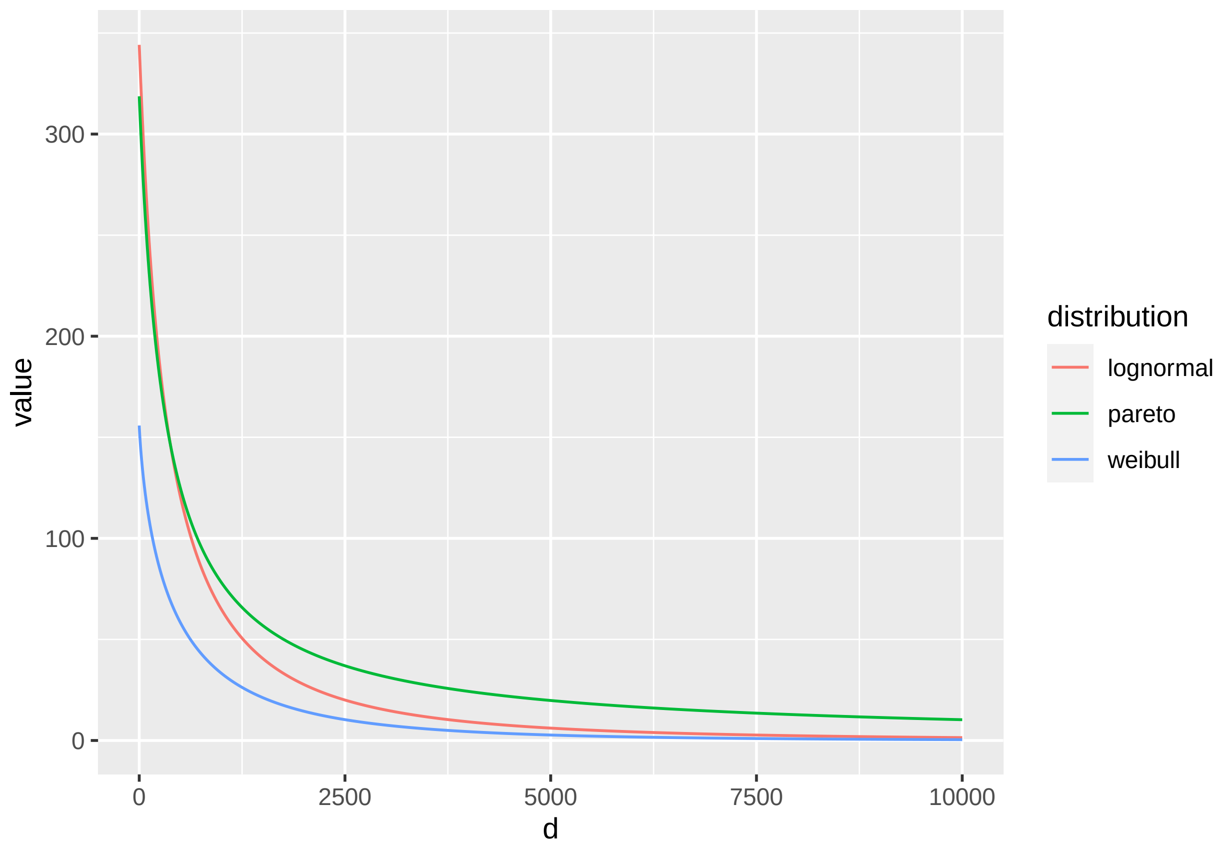
```
params_lognormal = optim(c(mean(claim$Clr),sd( claim$Clr) ),
                         log_likelihood_lognormal)
params_weibull = optim(c(0.7342342, 364.9229076 ), log_likelihood_weibull)
params_pareto = optim(c(2.005486, 0.1 ), log_likelihood_pareto)

tibble(log_normal_param = params_lognormal$par,
       weibull_param = params_weibull$par,
       pareto_param = params_pareto$par)
```

```
## # A tibble: 2 x 3
##   log_normal_param weibull_param pareto_param
##              <dbl>         <dbl>        <dbl>
## 1             5.12         0.417       312.
## 2             1.20        52.5           1.98
```

We apply eq. (1.22) for the expectation $E((X^* - d)_+)$.

```
F_bar_lognormal_fitted = function(x){
  return( plnorm(x, params_lognormal$par[1], params_lognormal$par[2],
                 lower.tail = F) )
}

F_bar_weibull_fitted = function(x){
  return( pweibull(x, params_weibull$par[1], params_weibull$par[2],
                   lower.tail = F))
}

F_bar_pareto_fitted = function(x){
  return(F_bar_pareto(x, params_pareto$par[1], params_pareto$par[2]))
}

numerical_int_lognormal = function(d, N){
  return( integrate(F_bar_lognormal_fitted, d, N)$val )
}

numerical_int_weibull = function(d, N){
  return( integrate(F_bar_weibull_fitted, d, N)$val )
}

numerical_int_pareto = function(d, N){
  return( integrate(F_bar_pareto_fitted, d, N)$val )
}
```

```
N = 1000000

plot = tibble(d = seq(0, 10000)) %>%
  group_by_all() %>%
  mutate(lognormal = numerical_int_lognormal(d, N),
         weibull = numerical_int_weibull(d, N),
         pareto = numerical_int_pareto(d, N))

plot_edited = plot %>%
  pivot_longer(!d, names_to = "distribution", values_to = "value")

ggplot(plot_edited, aes(d, value, color = distribution)) +
  geom_line()
```



We can interpert the expectation $E((X^* - d)_+)$ as the risk premium at a deductable level d without taking frequency into account. It seems that the Pareto and log normal dist is more heavy tailed than the weibull. The most heavy tailed distribution is the Pareto distribution. For deductables under 500 we see that the Pareto and Log Normal dist. implies approx. the same risk premium. Beyond deductable 500 we notice that the risk premium for Pareto distbution is noticeable higer for Pareto than both Log normal and Weibull.

R exercise 2

We use the parameters of θ from exercise 1 as the first pseudo MLE estimator and now calculate the second pseduo MLE estimator.

```
frequency_pre = read_csv("frequency.csv")
```

```
## Rows: 18353 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (10): Time, Dedr, Age, Brt, Dwt, Value, HP, Stroke, Code1, Claims
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

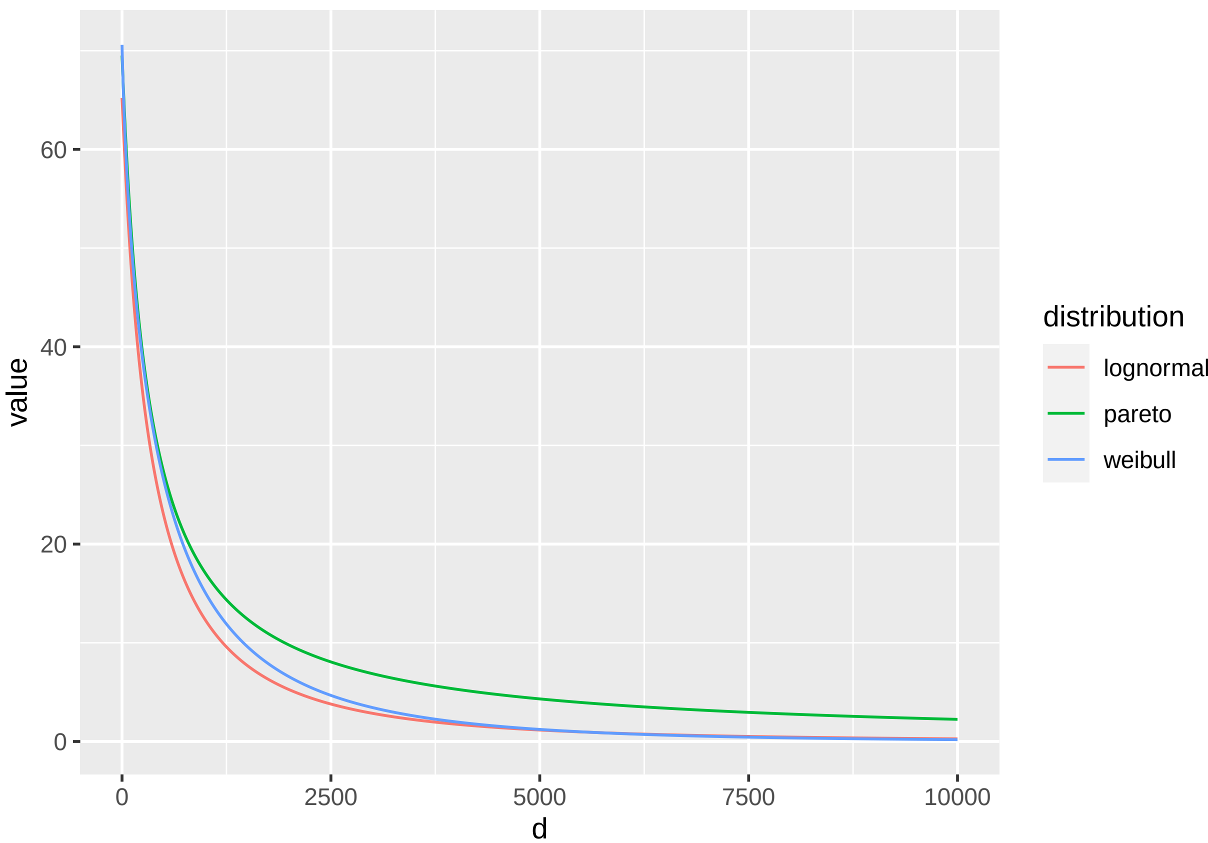
```
frequency = frequency_pre
frequency$Dedr = frequency$Dedr / 1000
```

```r
log_likelihood_lognormal_N = function(lambda){
  F_bar = plnorm(frequency$Dedr, params_lognormal$par[1],
                 params_lognormal$par[2], lower.tail = F)
  return( -sum( frequency$Claims * log( lambda * frequency$Time *  F_bar)
        - lambda * frequency$Time *  F_bar - log(factorial(frequency$Claims)) ))
}

log_likelihood_weibull_N = function(lambda){
  F_bar = pweibull(frequency$Dedr, params_weibull$par[1], params_weibull$par[2],
                   lower.tail = F)
  return( -sum( frequency$Claims * log( lambda * frequency$Time *  F_bar)
        - lambda * frequency$Time *  F_bar - log(factorial(frequency$Claims)) ))
}

log_likelihood_pareto_N = function(lambda){
  F_bar = F_bar_pareto(frequency$Dedr, params_pareto$par[1], params_pareto$par[2])
  return( -sum( frequency$Claims * log( lambda * frequency$Time *  F_bar)
        - lambda * frequency$Time *  F_bar - log(factorial(frequency$Claims)) ))
}

params_lognormal_N = optim(1, log_likelihood_lognormal_N, method = "Brent",
                           lower = 0, upper = 1000000)
params_weibull_N = optim(1, log_likelihood_weibull_N, method = "Brent",
                         lower = 0, upper = 1000000)
params_pareto_N = optim(1, log_likelihood_pareto_N, method = "Brent",
                        lower = 0, upper = 1000000)

lambda_params = tibble(lambda_poisson_lognormal = params_lognormal_N$par,
       lambda_poisson_weibull = params_weibull_N$par,
       lambda_poisson_pareto = params_pareto_N$par)
lambda_params
```

```
## # A tibble: 1 x 3
##   lambda_poisson_lognormal lambda_poisson_weibull lambda_poisson_pareto
##                      <dbl>                  <dbl>                 <dbl>
## 1                    0.189                  0.453                 0.218
```

We plot the risk premiums and again see that the risk premium of the Pareto-Poisson dist. is the higest.

```r
plot_edited = plot %>%
  mutate(
    lognormal = lognormal * lambda_params$lambda_poisson_lognormal,
    weibull = weibull * lambda_params$lambda_poisson_weibull,
    pareto = pareto * lambda_params$lambda_poisson_pareto
  ) %>%
  pivot_longer(!d, names_to = "distribution", values_to = "value")

ggplot(plot_edited, aes(d, value, color = distribution)) +
  geom_line()
```


value
60
40
20
0
0
2500
5000
7500
10000
d
distribution
lognormal
pareto
weibull