The Weibull distribution

With i.i.d. Weibull distributed observations $Z_1, \ldots, Z_n$ then $Y_1 = Z_1^k, \ldots, Y_n = Z_n^k$ are i.i.d. from the exponential family.

This is a generalized linear model with $\mathbf{X} = \mathbf{1}$ the $n \times 1$ matrix of ones, and

$$\boldsymbol{\eta} = \mathbf{1}\eta.$$

The score equation is

$$0 = U(\eta) = \sum_{i=1}^{n} \theta'(\eta) y_i - c'(\eta) = \frac{k}{\eta^{k+1}} \left(\sum_{i=1}^{n} y_i \right) - \frac{nk}{\eta}.$$

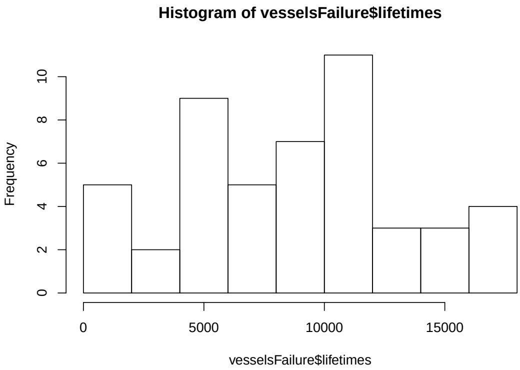
The Fisher information is

$$J(\eta) = n\theta'(\eta)^2 V(Y) = \frac{nk^2}{\eta^2}.$$

Data example

We use a small example data set on the lifetime of presure vessels.

```
vesselsFailure <- read.table(
  "http://www.math.ku.dk/~richard/courses/Regression/vessels.txt",
  header = TRUE)
hist(vesselsFailure$lifetimes)
```



Score function etc.

To implement the Newton algorithm for fitting the Weibull distribution we need to implement the functions that compute the score and the derivative of the score.

```
U <- function(eta, k, y = vesselsFailure$lifetimes)
  k * sum(y^k) / (eta^(k+1)) - length(y) * k / eta

dU <- function(eta, k, y = vesselsFailure$lifetimes)
  length(y) * k / eta^2 - k * (k + 1) * sum(y^k) / (eta^(k + 2))
```

We will also need the Fisher information.

```
J <- function(eta, k, n = length(vesselsFailure$lifetimes))
  n * (k / eta)^2
```

The Newton algorithm

We replace the (nonlinear) score equation by the linearization around η_1

$$U(\eta_1) + U'(\eta_1)(\eta - \eta_1) = 0 \tag{1}$$

relying on $U(\eta) \simeq U(\eta_1) + U'(\eta_1)(\eta - \eta_1)$.

The solution to (1) is

$$\eta = \eta_1 - \frac{U(\eta_1)}{U'(\eta_1)}.$$

The Newton algorithm is obtained by iteration;

$$\eta_{m+1} = \eta_m - \frac{U(\eta_m)}{U'(\eta_m)}$$

with η_1 the initial value or start guess.

Newton and Fisher updates

Replacing $-U'(\eta)$ by its expectation $J(\eta)$ we get the Fisher scoring algorithm

$$\eta_{m+1} = \eta_m + \frac{U(\eta_m)}{J(\eta_m)}.$$

```
Newton <- function(eta, k = 2)
  - U(eta, k) / dU(eta, k)

Fisher <- function(eta, k = 2)
  U(eta, k) / J(eta, k)
```

Newton and Fisher algorithms

Argument `update = Fisher` gives Fisher scoring.

```
iterativeMLE <- function(eta = mean(vesselsFailure$lifetimes),
                         k = 2,
                         nrIter = 6,
                         update = Newton){
  ## Setting up the format of the results
  results <-  matrix(0, ncol = nrIter, nrow = 6)
  row.names(results) <- c("eta", "U", "-dU", "J"," -U/dU", "U/J")
  for(m in 1:nrIter) {
    results[, m] <- c(eta,
                      U(eta, k) * 1e6,
                      -dU(eta, k) * 1e6,
                      J(eta, k) * 1e6,
                      Newton(eta, k),
                      Fisher(eta, k)
                      )
    eta <- eta + update(eta, k)
    }
  results
}
```

Newton and Fisher algorithms

```
iterativeMLE()

##            [,1]   [,2] [,3]    [,4]    [,5]    [,6]
## eta     8805.7 9633.8 9876 9.9e+03 9.9e+03 9.9e+03
## U       2915.8  553.0   33 1.3e-01 2.3e-06 1.7e-12
## -dU        3.5    2.3    2 2.0e+00 2.0e+00 2.0e+00
## J          2.5    2.1    2 2.0e+00 2.0e+00 2.0e+00
##  -U/dU   828.1  242.1   16 6.7e-02 1.1e-06 8.7e-13
## U/J     1153.5  261.9   16 6.7e-02 1.1e-06 8.7e-13
```

```
iterativeMLE(update = Fisher)

##            [,1]   [,2]    [,3]     [,4]    [,5]    [,6]
## eta     8805.7 9959.2 9892.40  9.9e+03 9.9e+03 9.9e+03
## U       2915.8 -132.0   -0.45 -5.2e-06 1.7e-12 1.7e-12
## -dU        3.5    1.9    2.00  2.0e+00 2.0e+00 2.0e+00
## J          2.5    2.0    2.00  2.0e+00 2.0e+00 2.0e+00
##  -U/dU   828.1  -68.2   -0.23 -2.6e-06 8.7e-13 8.7e-13
## U/J     1153.5  -66.8   -0.23 -2.6e-06 8.7e-13 8.7e-13
```

The quadratic approximation

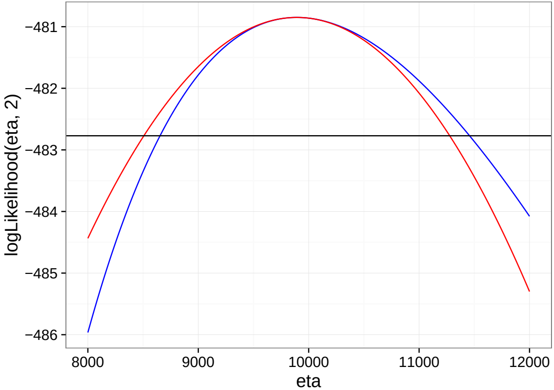
We compare the log-likelihood with a quadratic function with maximum in $\hat{\eta}$.

```
etaHat0 <- iterativeMLE()["eta", 6]
quardApprox <- function(eta) {
  logLikelihood(etaHat0, 2) -
    (eta - etaHat0) * J(etaHat0, 2) * (eta - etaHat0) / 2
}

logLikelihood <- function(eta, k, y = vesselsFailure$lifetimes) {
  length(y)*(log(k) - k*log(eta)) +
    (k-1)*sum(log(y)) - sum(y^k)/eta^k
}
```

The quadratic approximation

```
eta <- 8000:12000
qplot(eta, logLikelihood(eta, 2), geom = "line", color = I("blue")) +
  geom_line(aes(x = eta, y = quardApprox(eta)), color = "red") +
  ## Add a line that indicates estimated 95% confidence intervals.
  geom_abline(slope = 0, intercept =
                logLikelihood(etaHat0, 2) - qchisq(0.95, 1) / 2)
```



Generalized linear models

The p-dimensional score equation for a GLM reads

$$U(\beta) = \mathbf{X}^T U(\boldsymbol{\eta}) = 0.$$

Given β_1 and corresponding $\eta_{1,i} = X_i^T \beta_1$ let

$$\mathbf{W}_1^{\text{obs}} = -\begin{pmatrix} U_1'(\eta_{1,1}) & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & U_n'(\eta_{1,n}) \end{pmatrix}.$$

The linearization of the score equation is

$$\mathbf{X}^T U(\boldsymbol{\eta}_1)^T - \mathbf{X}^T \mathbf{W}_1^{\text{obs}} \mathbf{X}(\beta - \beta_1) = 0,$$

whose solution is

$$\beta = \beta_1 + (\mathbf{X}^T \mathbf{W}_1^{\text{obs}} \mathbf{X})^{-1} \mathbf{X}^T U(\boldsymbol{\eta}_1).$$

Fisher scoring

Replacing $\mathbf{W}_m^{\text{obs}}$ by its expectation

$$\mathbf{W}_m = \frac{1}{\psi}\begin{pmatrix} \mu'(\eta_{m,1})\theta'(\eta_{m,1}) & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \mu'(\eta_{m,n})\theta'(\eta_{m,n}) \end{pmatrix}$$

we get by iteration the Fisher scoring algorithm.

$$\begin{aligned} \beta_{m+1} &= \beta_m + (\mathbf{X}^T\mathbf{W}_m\mathbf{X})^{-1}\mathbf{X}^T U(\boldsymbol{\eta}_m) \\ &= (\mathbf{X}^T\mathbf{W}_m\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}_m\Big(\underbrace{\mathbf{X}\beta_m + \mathbf{W}_m^{-1}U(\boldsymbol{\eta}_m)}_{\mathbf{z}_m}\Big). \end{aligned}$$

Iterative Weighted Least Squares

The vector $\mathbf{Z}_m$ is known as the working response and

$$Z_{m,i} = X_i^T \beta_m + \frac{Y_i - \mu(\eta_{m,i})}{\mu'(\eta_{m,i})} \tag{2}$$

In terms of the working response, the vector β_m is the minimizer of the weighted sum of squares

$$(\mathbf{Z}_m - \mathbf{X}\beta)^T \mathbf{W}_m (\mathbf{Z}_m - \mathbf{X}\beta). \tag{3}$$

This is a standard weighted least squares problem.

Iterative Weighted Least Squares

The Fisher scoring algorithm for GLMs is known as IWLS due to the iterative solution of a weighted least squares problem. Given β_1 we iterate until convergence:

- Compute the working response vector $\mathbf{Z}_m$ based on β_m using (2).
- Compute the weights

$$w_{m,ii} = \frac{(\mu'_{m,i})^2}{\mathcal{V}(\mu_{m,i})} = \frac{\mu'(\eta_{m,i})^2}{\mathcal{V}(\mu(\eta_{m,i}))}.$$

- Minimize the weighted sum of squares (3).

Computations rely only on the mean value function μ, its derivative μ' and the variance function $\mathcal{V}$.

Estimation of the nuisance parameter

Implementation of the profile log-likelihood.

```
etaHat <- function(k = 2,
                   y = vesselsFailure$lifetimes,
                   update = Newton) {
  eta <- mean(y)
  maxIter <- 20
  convCrit <- 1e-6
  m <- 1
  for(m in 2:maxIter) {
    up <- update(eta, k)
    eta <- eta + up
    if(abs(up) < convCrit)
      break
  }
  return(eta)
}
```

Profile log-likelihood

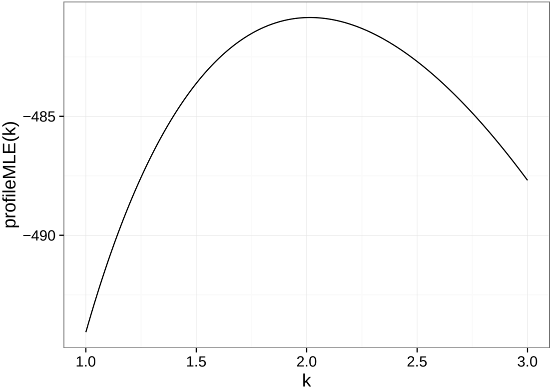
The use of `Vectorize` below makes the function work for vector arguments.

```
profileMLE <- Vectorize(function(k)
                        logLikelihood(etaHat(k), k)
                        )
optimize(profileMLE, c(1, 3), maximum = TRUE)

## $maximum
## [1] 2
##
## $objective
## [1] -481
```

Profile log-likelihood

```
k <- seq(1, 3, 0.01); qplot(k, profileMLE(k), geom = "line")
```



Score function with canonical link

With the canonical link function $\theta'(\eta) = 1$ and the score function is

$$U(\beta) = \sum_{i=1}^{n} (Y_i - \mu(\eta_i))X_i = t - \tau(\beta)$$

whose second derivative equals the negative Fisher information

$$-D_\beta U(\beta) = \mathcal{J}(\beta) = \mathbf{X}^T \mathbf{W} \mathbf{X}$$

with $\mathbf{W}_{ii} = \mu'(\eta_i) = \mathcal{V}(\mu(\eta_i))$. Recall that $\mu'(\eta) > 0$ for all η, thus the weights are always strictly positive for the canonical link.

Define

$$\tau(\beta) = \sum_{i=1}^{n} \mu(\eta_i)X_i \quad \text{and} \quad t = \sum_{i=1}^{n} Y_i X_i$$

The score equation

The score equation is

$$\tau(\beta) = t. \tag{4}$$

Theorem (Thm. 6.9)

If $\mathbf{X}$ has full rank p the map $\tau : \mathbb{R}^p \to \mathbb{R}^p$ is one-to-one. With $C := \tau(\mathbb{R}^p)$ there is a unique solution to (4) if and only if $t \in C$.

Lemma (Lemma 6.10)

If $t_0 = \sum_{i=1}^n \mu_i X_i$ with $\mu_i \in J := \mu(\mathbb{R})$ there is a solution to the equation $\tau(\beta) = t_0$.

Existence of the MLE

Corollary (Cor. 6.11)

The set $C = \tau(\mathbb{R}^p)$ has the representation

$$C = \left\{ \sum_{i=1}^{n} \mu_i X_i \mid \mu_i \in J \right\} \tag{5}$$

and is convex. If $\mathbf{X}$ has full rank p then C is open.

To check if the MLE exists we need to check if $t \in C$. This is trivially the case if

$$P(Y \in J) = 1$$

but less trivial to check if $P(Y \in \partial J) > 0$.

The solution, if it exists, is unique if $\mathbf{X}$ has full rank p.

Poisson example

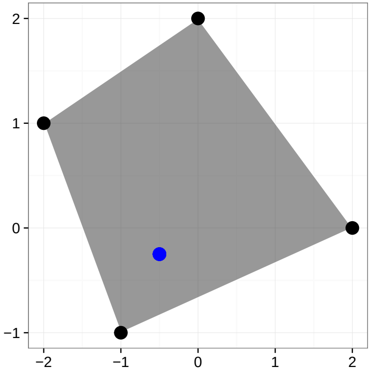
```
X <- data.frame(x1 = c(-2, -1, 2, 0), x2 = c(1, -1, 0, 2))
y <- c(1, 2, 1, 0); Xy <- cbind(y, X)
t <- c(y %*% X$x1, y %*% X$x2) / sum(y)
summary(glm(y ~ x1 + x2, family = poisson, data = Xy))


...
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.0267     0.5461    0.05     0.96
## x1           -0.1237     0.3742   -0.33     0.74
## x2           -0.6550     0.5157   -1.27     0.20
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2.77259  on 3  degrees of freedom
## Residual deviance: 0.75402  on 1  degrees of freedom
## AIC: 13.37
##
## Number of Fisher Scoring iterations: 5
```

Poisson example

In the example the average t was in the interior of the convex hull, and we could fit the Poisson model using `glm`.

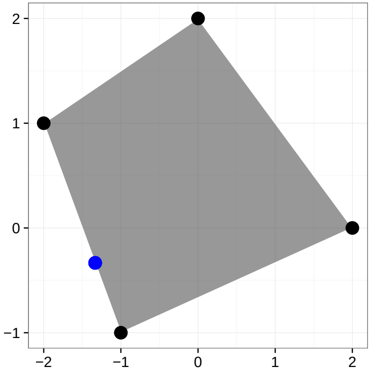
```
p <- qplot(x1, x2, data = X, geom = "polygon", alpha = I(0.5)) +
  geom_point(size = 5, alpha = 1) + xlab("") + ylab("")
p + geom_point(aes(t[1], t[2]), size = 5, color = "blue")
```



Poisson example

Then we consider an example where the average t ends up on the boundary of the convex hull.

```
y <- c(1, 2, 0, 0); Xy <- cbind(y, X)
t <- c(y %*% X$x1, y %*% X$x2) / sum(y)
p + geom_point(aes(t[1], t[2]), size = 5, color = "blue")
```



Poisson example

```
summary(glm(y ~ x1 + x2, family = poisson, data = Xy))
```

```
...
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -12.84   25534.58       0        1
## x1             -8.79   17023.05       0        1
## x2             -4.74    8511.53       0        1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 4.4987e+00  on 3  degrees of freedom
## Residual deviance: 4.0610e-10  on 1  degrees of freedom
## AIC: 10.61
##
## Number of Fisher Scoring iterations: 21
```

Binary response

If the response is binary, $I = \mathbb{R}$, $J = (0,1)$ and the canonical link is the logit function

$$(0,1) \ni p \mapsto \mathrm{logit}(p) = \log\left(\frac{p}{1-p}\right).$$

The response variables all take values on the boundary of $J = (0,1)$!

We have that

$$t = \sum_{i: Y_i = 1} X_i \in \overline{C}$$

and we need to find conditions in terms of the X_i that ensure that $t \in C$.

Separation

The responses $Y_1, \dots, Y_n \in \{0, 1\}$ are binary.

Definition

We say that $X_1, \dots, X_n \in \mathbb{R}^p$ are separated by $Y_1, \dots, Y_n$ if there exists a nonzero vector $\beta \in \mathbb{R}^p$ such that for all $i = 1, \dots, n$

$$X_i^T \beta \geq 0 \quad \text{if } Y_i = 1,$$

and

$$X_i^T \beta \leq 0 \quad \text{if } Y_i = 0.$$

Observe that if $\mathbf{X}$ has full rank p, and the rows are separated according to the definition above, the at least one of the n inequalities above is sharp because β is assumed nonzero. The vector β is called the separating vector.

Existence of the MLE in logistic regression

We consider binary responses $Y_1, \ldots, Y_n \in \{0, 1\}$ and the logistic regression model.

Theorem (Th, 6.16)

Assume that $\mathbf{X}$ has full rank p. The MLE exists if and only if the rows of $\mathbf{X}$ are not separated by $Y_1, \ldots, Y_n$.

Being explicit about the intercept

If the model contains an intercept in addition to the predictors $X_i \in \mathbb{R}^p$, it is

$$\tilde{X}_i = (1, X_i^T)^T$$

for $i = 1, \ldots, n$ that must be checked for separability. This is equivalent to the existence of $\beta \in \mathbb{R}^p$ and $\beta_0 \in \mathbb{R}$ such that for all $i = 1, \ldots, n$

$$X_i^T \beta \geq \beta_0 \quad \text{if } Y_i = 1,$$

and

$$X_i^T \beta \leq \beta_0 \quad \text{if } Y_i = 0.$$

Checking for linear separability

Corollary (Cor. 6.17)

Assume that $\mathbf{X}$ *has full rank* p. *The maximization problem*

$$\begin{array}{ll} \textit{maximize} & \sum_{i=1}^{n} s_i \\ \textit{subject to} & (2Y_i - 1)X_i^T\beta \geq s_i, \; s_i \geq 0, \; i = 1, \ldots, n, \\ & -1 \leq \beta_j \leq 1, \; j = 1, \ldots, p \end{array}$$

in the variables $(\beta^T, s^T)^T \in \mathbb{R}^{n+p}$ *has a solution with* $\sum_{i=1}^{n} s_i > 0$ *if and only if* $X_1, \ldots, X_n$ *are separated by* $Y_1, \ldots, Y_n$.

The constraints on the β_j's force the s_i's to be bounded, and the constraints are fulfilled for $\beta = \mathbf{0}_p$ and $s = \mathbf{0}_n$. Thus we maximize a linear function over a compact set, and there is always a finite solution bounded below by 0.

Poisson responses

For Poisson distributed responses we have $I = \mathbb{R}$, $J = (0, \infty)$ and canonical link

$$(0, \infty) \ni \mu \mapsto \log(\mu).$$

The nonexistence of the MLE is clearly related to observations being 0.

Existence of the MLE in Poisson regression

We consider positive responses $Y_i \geq 0$ and the Poisson regression model with log-link. We let

$$t_0 = \sum_{i=1}^{n} Y_i X_i = \mathbf{X}^T \mathbf{Y}.$$

Corollary (Cor. 6.13)

Assume that $\mathbf{X}$ *has full rank p. The MLE exists if and only if the following linear program*

$$\begin{array}{ll} \textit{maximize} & s \\ \textit{subject to} & \mathbf{X}^T \mu = t_0,\ \mu_i - s \geq 0,\ s \geq 0. \end{array}$$

in the variables $(\mu^T, s)^T \in \mathbb{R}^{n+1}$ *has a feasible point with* $s > 0$.

Note that $(\mathbf{Y}^T, 0)^T$ is a feasible point.

Specifying the linear program in practice

The linear program is specified in practice in terms of a vector $c \in \mathbb{R}^{n+1}$ of objective coefficients and an $(n+p) \times (n+1)$ constraint matrix A. They are given as

$$c = (0, \ldots, 0, 1)^T$$

and

$$A = \left(\begin{array}{cc} \mathbf{I}_n & -\mathbf{1}_n \\ \mathbf{X}^T & \mathbf{0}_p \end{array} \right)$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix, $\mathbf{1}_n$ is the n-dimensional vector of ones and $\mathbf{0}_p$ is the p-dimensional vector of zeroes.

The constraint matrix specifies the left hand side of the $n + p$ constraints in the $n + 1$ variables. The first n are inequality constraints and the last p are equality constraints. The right hand side of the constraints is the $(n + p)$-dimensional vector

$$\left(\begin{array}{c} \mathbf{0}_n \\ t_0 \end{array} \right).$$

Poisson example

```
## Coefficient vector
c <- c(0, 0, 0, 0, 1)
## Constraint matrix
A <- matrix(
  c(1, 0, 0, 0, -1,
    0, 1, 0, 0, -1,
    0, 0, 1, 0, -1,
    0, 0, 0, 1, -1,
    1, 1, 1, 1,  0,
   -2, -1, 2, 0, 0,
    1, -1, 0, 2, 0),
  nrow = 7,
  ncol = 5,
  byrow = TRUE)
## Right hand side
t <- A[5:7, 1:4] %*% c(1, 2, 1, 0)
rhs <- c(0, 0, 0, 0, t)
## Directions of the (in)equalities
dir <- c(rep(">=", 4), rep("=", 3))
```

Poisson example

```
lp(direction = "max",
   objective.in = c,
   const.mat = A,
   const.dir = dir,
   const.rhs = rhs
)

## Success: the objective function is 0.47
```

Poisson example

```r
## Changing the right hand side
t <- A[5:7, 1:4] %*% c(1, 2, 0, 0)
rhs <- c(0, 0, 0, 0, t)
## Solving the linear program
lp(direction = "max",
   objective.in = c,
   const.mat = A,
   const.dir = dir,
   const.rhs = rhs
)

## Success: the objective function is 0
```