```
logLik.bamlss <- function(object, ..., optimizer = FALSE, samples = FALSE)
{
  Call <- match.call()
  Call <- Call[!(names(Call) %in% c("optimizer", "samples"))]
  mn <- as.character(Call)[-1L]
  object <- list(object, ...)
  mstop <- object$mstop
  if(any(names(object) != "")) {
    i <- names(object) == ""
    object <- object[i]
    mn <- mn[i]
  }
  object <- object[mn != "mstop"]
```

Advanced Bayesian Methods: Theory and Applications in R

Creating Families

Nikolaus Umlauf
https://nikum.org/abm.html

`"family.bamlss"`, `"gamlss2.family"`, `"gamlss.family"`

- Objects of class `"family.bamlss"`, `"gamlss2.family"` and `"gamlss.family"` are used to specify the information required for model fitting in the *gamlss2* package.
- These objects include the parameter names, link functions, density function, log-likelihood function, and the derivatives of the log-likelihood with respect to predictors.
- Family objects also assist with post-modeling tasks, such as residual diagnostics and random number generation.

Details of Family Objects

Family objects in *bamlss* must satisfy the following minimum requirements:

- Return a list of class `"family.bamlss"` or `"gamlss2.family"`.
- Contain the family name as a character string.
- Contain the names of parameters and corresponding link functions.
- Contain a density function `d()` for evaluating (log-)density.

The density function must take arguments:

```
d(y, par, log = FALSE, ...)
```

with `y` as the response and `par` as a named list of distribution parameters.

Optional Derivative Functions

Optionally, family objects may include functions to compute:

- First derivatives of the log-likelihood w.r.t predictors.
- Negative second derivatives.

These functions should have the following structure:

```
function(y, par, ...)
```

If not provided, numerical approximation of derivatives will be used.

Additional Functions in Family Objects

Family objects may also include:

- Cumulative distribution function, `p()`.
- Quantile function, `q()`.
- Random number generation, `r()`.

These functions follow the structure of the density function.

Example: Normal Distribution Family Object

A simple example for the normal distribution family:

- Two parameters: `mu` (mean) and `sigma` (standard deviation).
- Link functions: identity for `mu`, log for `sigma`.
- Missing score functions.
- Missing Hessian functions.
- Density function for normal distribution.

Normal Family Object Code

```
R> Normal <- function(...) {
+    fam <- list(
+      "family" = "Normal",
+      "names" = c("mu", "sigma"),
+      "links" = c("mu" = "identity", "sigma" = "log"),
+      "d" = function(y, par, log = FALSE, ...) {
+        dnorm(y, par$mu, par$sigma, log = log)
+      }
+    )
+    class(fam) <- "family.bamlss"
+    return(fam)
+  }
```

Normal Family Object Code

```
R> library("bamlss")
R> data("rent", package = "gamlss.data")
R> print(dim(rent))

[1] 1969    9

R> f <- list(
+    R ~ s(Fl) + s(A),
+      ~ s(Fl) + s(A)
+  )

R> set.seed(123)
R> b <- bamlss(f, data = rent, family = Normal)
```

Normal Family Object Code

```
AICc 87934513 logPost -4410029 logLik -4396722 edf 34.852 eps 1.0000 iteration   1
AICc 30613243 logPost -1533240 logLik -1530658 edf 36.998 eps 0.7936 iteration   2
AICc 11197622 logPost -5603084 logLik -5598772 edf 37.587 eps 0.2116 iteration   3
AICc 4125293. logPost -2063192 logLik -2062608 edf 37.892 eps 0.1427 iteration   4
AICc 1527177. logPost -763794. logLik -763550. edf 37.969 eps 0.1093 iteration   5
AICc 572684.4 logPost -286529. logLik -286303. edf 37.953 eps 0.0890 iteration   6
AICc 222786.6 logPost -111583. logLik -111354. edf 37.954 eps 0.0752 iteration   7
AICc 95296.98 logPost -47839.1 logLik -47609.7 edf 37.954 eps 0.0649 iteration   8
AICc 49602.98 logPost -24992.1 logLik -24762.7 edf 37.954 eps 0.0566 iteration   9
AICc 33937.97 logPost -17159.6 logLik -16930.2 edf 37.954 eps 0.0488 iteration  10
AICc 29169.43 logPost -14775.3 logLik -14545.9 edf 37.954 eps 0.0398 iteration  11
AICc 28103.62 logPost -14242.4 logLik -14013.0 edf 37.954 eps 0.0275 iteration  12
AICc 27991.44 logPost -14186.3 logLik -13957.0 edf 37.954 eps 0.0122 iteration  13
AICc 27989.25 logPost -14185.2 logLik -13955.9 edf 37.954 eps 0.0020 iteration  14
AICc 27989.25 logPost -14185.2 logLik -13955.9 edf 37.954 eps 0.0000 iteration  15
AICc 27989.25 logPost -14185.2 logLik -13955.9 edf 37.954 eps 0.0000 iteration  15
elapsed time:  0.76sec
Starting the sampler...
```

Normal Family Object Code

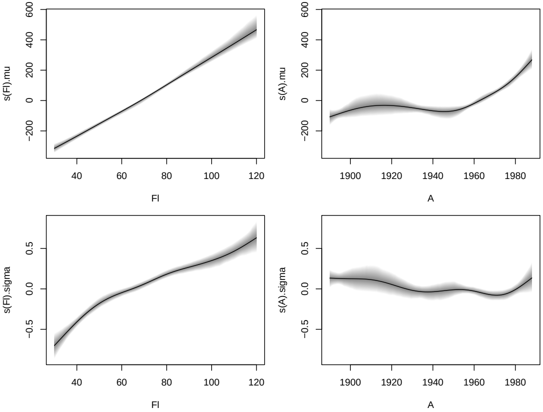
```
|                    |   0%  9.28sec
|*                   |   5%  8.91sec  0.47sec
|**                  |  10%  8.42sec  0.94sec
|***                 |  15%  7.93sec  1.40sec
|****                |  20%  7.91sec  1.98sec
|*****               |  25%  7.41sec  2.47sec
|******              |  30%  6.93sec  2.97sec
|*******             |  35%  6.44sec  3.47sec
|********            |  40%  5.96sec  3.97sec
|*********           |  45%  5.47sec  4.47sec
|**********          |  50%  4.97sec  4.97sec
|***********         |  55%  4.48sec  5.47sec
|************        |  60%  4.03sec  6.04sec
|*************       |  65%  3.52sec  6.54sec
|**************      |  70%  3.02sec  7.04sec
|***************     |  75%  2.51sec  7.54sec
|****************    |  80%  2.01sec  8.04sec
```

Normal Family Object Code

```
|*****************   |  85%  1.51sec  8.53sec
|******************  |  90%  1.00sec  9.04sec
|******************* |  95%  0.50sec  9.54sec
|********************| 100%  0.00sec 10.04sec
```

Normal Family Object Code

```
R> par(mfrow = c(2, 2), mar = c(4, 4, 1, 1))
R> plot(b, spar = FALSE, pages = 1)
```



Conclusion

- Family objects in *bamlss* (and *gamlss2*) play a crucial role in defining the model fitting process.
- They specify the distribution, parameter link functions, density, and derivative functions required for model estimation and diagnostics.