```r
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

Generalized Additive Models for Location, Scale and Shape (and Beyond)

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

What is a Regression Model?

- Typical answer is something like

$$\mathrm{E}(y_i|\boldsymbol{x}_i) = h(\eta(\boldsymbol{x}_i))$$

  and much of recent statistical research has focused on flexibility in specifying the regression predictor $\eta(\boldsymbol{x}_i)$, e.g.

  - generalized additive models,
  - random effects,
  - spatial effects,
  - etc.

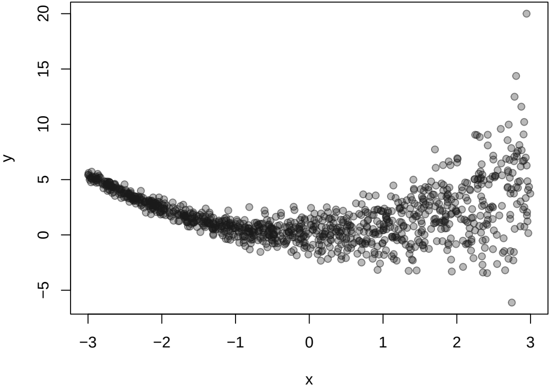
Regression Beyond the Mean

- Classical regression has focused on relating the conditional mean of a response y_i to covariate information x_i for observations $(x_1, y_1), \ldots, (x_n, y_n)$.

- Linear model:

$$y_i = \beta_0 + \beta x_i + \varepsilon_i$$

with ε_i i.i.d. $\mathrm{N}(0, \sigma^2)$



Regression Beyond the Mean

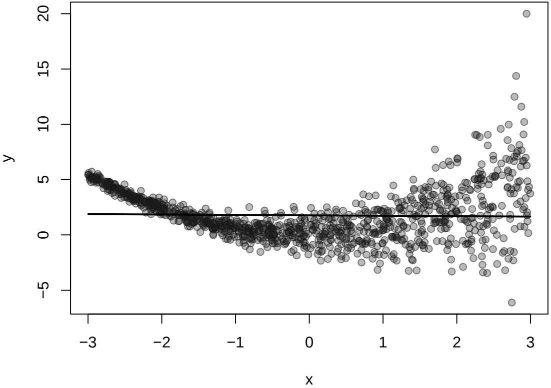
- Classical regression has focused on relating the conditional mean of a response y_i to covariate information x_i for observations $(x_1, y_1), \ldots, (x_n, y_n)$.

- Linear model:

$$y_i = \beta_0 + \beta x_i + \varepsilon_i$$

with ε_i i.i.d. $\mathrm{N}(0, \sigma^2)$

$\Rightarrow\ \mathrm{E}(y_i|x_i) = \mu_i(x_i) = \beta_0 + \beta x_i$



Regression Beyond the Mean

- Classical regression has focused on relating the conditional mean of a response y_i to covariate information x_i for observations $(x_1, y_1), \ldots, (x_n, y_n)$.

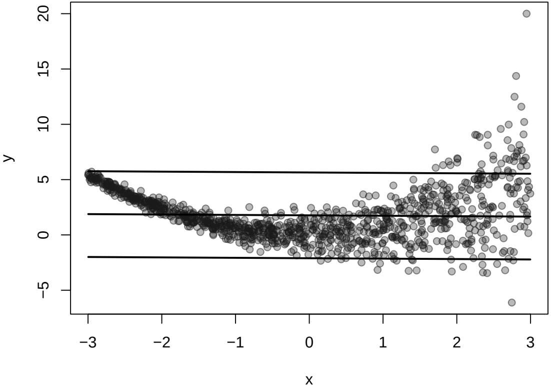
- Linear model:

$$y_i = \beta_0 + \beta x_i + \varepsilon_i$$

with ε_i i.i.d. $\mathrm{N}(0, \sigma^2)$

$\Rightarrow \; \mathrm{E}(y_i|x_i) = \mu_i(x_i) = \beta_0 + \beta x_i$

$\Rightarrow \; \mathrm{Var}(y_i|x_i) = \sigma^2$



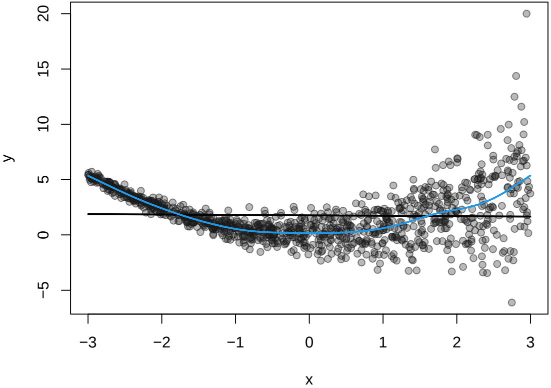
Regression Beyond the Mean

- Classical regression has focused on relating the conditional mean of a response y_i to covariate information x_i for observations $(x_1, y_1), \ldots, (x_n, y_n)$.

- Nonparametric model

$$E(y_i|x_i) = \mu_i(x_i) = \beta_0 + s(x_i)$$

with $y_i|x_i \sim N\left(\mu_i(x_i), \sigma^2\right)$



Regression Beyond the Mean

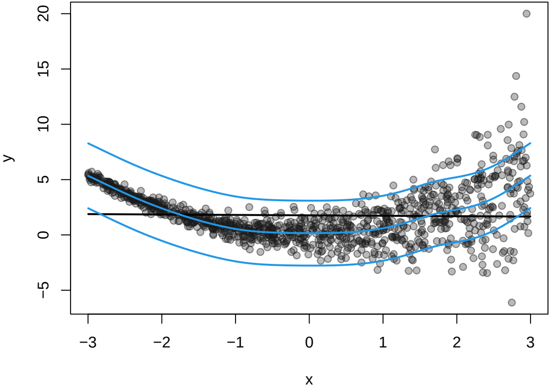
- Classical regression has focused on relating the conditional mean of a response y_i to covariate information x_i for observations $(x_1, y_1), \ldots, (x_n, y_n)$.

- Nonparametric model

$$\mathrm{E}(y_i | x_i) = \mu_i(x_i) = \beta_0 + s(x_i)$$

with $y_i | x_i \sim \mathrm{N}\left(\mu_i(x_i), \sigma^2\right)$

σ^2 fixed.



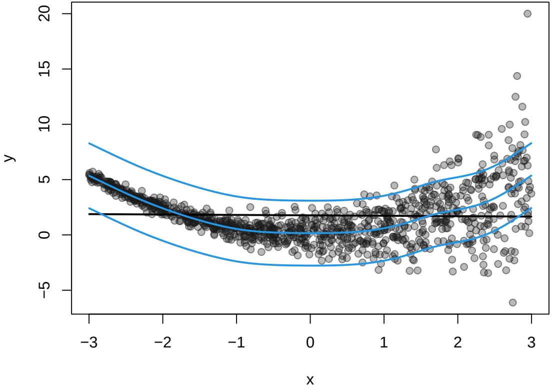
Regression Beyond the Mean

- Nonparametric model for loaction and scale.

- $$E(y_i|x_i) = \mu_i(x_i) = \beta_0^{\mu} + s^{\mu}(x_i)$$

$$\begin{aligned} \text{Var}(y_i|x_i) &= \sigma_i^2(x_i) \\ &= \exp\left(\beta_0^{\sigma^2} + s^{\sigma^2}(x_i)\right) \end{aligned}$$

with $y_i|x_i \sim \text{N}\left(\mu_i(x_i), \sigma_i^2(x_i)\right)$.



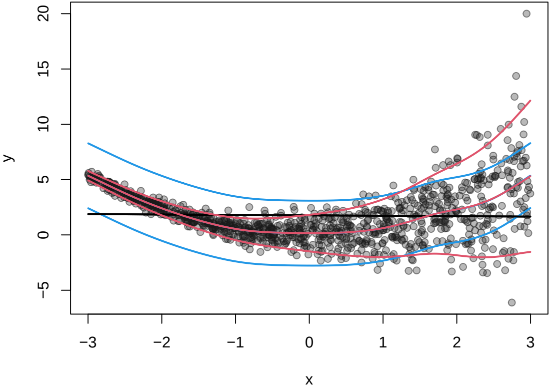
Regression Beyond the Mean

- Nonparametric model for loaction and scale.

- $E(y_i|x_i) = \mu_i(x_i) = \beta_0^{\mu} + s^{\mu}(x_i)$

$$\begin{aligned} \text{Var}(y_i|x_i) &= \sigma_i^2(x_i) \\ &= \exp\left(\beta_0^{\sigma^2} + s^{\sigma^2}(x_i)\right) \end{aligned}$$

with $y_i|x_i \sim \text{N}\left(\mu_i(x_i), \sigma_i^2(x_i)\right)$.



Distributional Regression

- Why should we focus on the mean alone if this gives only such an incomplete picture about the (conditional) distribution of the response y_i?
- Distributional regression is an umbrella term for approaches that model the conditional distribution of the response in terms of covariate information.
- Once you start thinking about it, distributional aspects are relevant in virtually any statistical modelling exercise:
    - Conditional income distributions to analyse income inequality.
    - Economic analyses of firm efficiency.
    - Modelling risk and uncertainty, e.g., weather risks.
    - . . .

Different Frameworks for Distributional Modelling

- There are different frameworks that enable distributional regression modelling:
  - generalized additive models for location, scale and shape (GAMLSS),
  - quantile and expectile regression,
  - conditional transformation models, and
  - various other forms.
- We will focus mostly on GAMLSS-type models but will also briefly review some extensions.

GAMLSS Setup

- Assume a parametric specification for the conditional distribution of the responses y_i given covariates $\boldsymbol{x}_i$ such that

$$p(y_i|\boldsymbol{x}_i) = p(y_i|\boldsymbol{\theta}(\boldsymbol{x}_i)),$$

  where $\boldsymbol{\theta}(\boldsymbol{x}_i) = (\theta_1(\boldsymbol{x}_i), \dots, \theta_K(\boldsymbol{x}_i))^\top$ is a K-dimensional vector of distributional parameters.

- Each parameter $\theta_k(\boldsymbol{x}_i)$ is linked to a regression predictor $\eta_{ik} = \eta_k(\boldsymbol{x}_i)$ based on a response function $h_k(\cdot)$:

$$\theta_k(\boldsymbol{x}_i) = h_k(\eta_k(\boldsymbol{x}_i)) \quad \text{and} \quad \eta_k(\boldsymbol{x}_i) = h_k^{-1}(\theta_k(\boldsymbol{x}_i)).$$

Typical Response Functions

- $\theta_{ik} = \eta_{ik}$ if no restrictions are required,
- $\theta_{ik} = \exp(\eta_{ik})$ for positive parameters such as variances,
- $\theta_{ik} = \exp(\eta_{ik})/(1 + \exp(\eta_{ik}))$ or $\theta_{ik} = \Phi(\eta_{ik})$ for probabilities, or
- $\theta_{ik} = \frac{\eta_{ik}}{\sqrt{1+\eta_{ik}^2}}$ for parameters restricted to $[-1, 1]$.

Types of Distributions

- Zero-inflated and overdispersed count data, i.e. responses with an excess of zeros and / or variances exceeding the expectation.
- Responses with heteroscedastic or skewed distribution.
- Continuous data with a spike in zero.
- Fractional responses restricted to [0,1] (possibly with inflation in 0 and 1).
- Multivariate responses with regression effects on the dependency parameters.

Predictor Components

- Nonlinear effects of continuous covariates.
- Spatial effects based on discrete regional or continuous coordinate information.
- Different types of interactions such as varying coefficients or interaction surfaces.
- Random effects for grouped data.
- etc.

Statistical Inference

- (Penalized) maximum likelihood.
- Bayesian inference with regularisation priors.
- Functional gradient descent boosting.
- Distributional trees and forests.
- Neural networks.
- etc.

Linear Models for Location and Scale

Packages for distributional modelling.

```
R> library("gamlss")
R> library("gamlss2")
R> library("mgcv")
R> library("bamlss")
```

Prices of used VW cars.

```
R> data("Golf", package = "bamlss")
R> print(head(Golf))
```

```
  price age kilometer tia abs sunroof
1  7.30  73        10  12 yes     yes
2  3.85 115        30  20 yes      no
3  2.95 127        43   6  no     yes
4  4.80 104        54  25 yes     yes
5  6.20  86        57  23  no      no
6  5.90  74        57  25 yes      no
```

Linear Models for Location and Scale

Linear model for location and shape.

```
R> b1 <- gamlss(price ~ age + kilometer + tia + abs + sunroof,
+     sigma.formula =~ age + kilometer + tia + abs + sunroof,
+     data = Golf, family = NO)
```

```
GAMLSS-RS iteration 1: Global Deviance = 372.2874
GAMLSS-RS iteration 2: Global Deviance = 368.5893
GAMLSS-RS iteration 3: Global Deviance = 367.6758
GAMLSS-RS iteration 4: Global Deviance = 367.405
GAMLSS-RS iteration 5: Global Deviance = 367.3272
GAMLSS-RS iteration 6: Global Deviance = 367.3052
GAMLSS-RS iteration 7: Global Deviance = 367.2993
GAMLSS-RS iteration 8: Global Deviance = 367.2975
GAMLSS-RS iteration 9: Global Deviance = 367.297
```

Linear Models for Location and Scale

Model summary.

```
R> summary(b1)
```

```
******************************************************************
Family:  c("NO", "Normal")

Call:  gamlss(formula = price ~ age + kilometer + tia + abs +
    sunroof, sigma.formula = ~age + kilometer + tia +
    abs + sunroof, family = NO, data = Golf)

Fitting method: RS()

------------------------------------------------------------------
Mu link function:  identity
Mu Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.277808   0.553530  14.955  < 2e-16 ***
age         -0.033259   0.003811  -8.727 3.26e-15 ***
```

Linear Models for Location and Scale

```
kilometer   -0.008134   0.001353  -6.010 1.21e-08 ***
tia         -0.003752   0.007190  -0.522    0.603
absyes      -0.127191   0.130483  -0.975    0.331
sunroofyes   0.135028   0.112639   1.199    0.232
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

------------------------------------------------------------------
Sigma link function:  log
Sigma Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.860776   0.460441   1.869  0.06339 .
age         -0.006644   0.003604  -1.843  0.06713 .
kilometer   -0.004270   0.001634  -2.613  0.00984 **
tia          0.006237   0.007304   0.854  0.39444
absyes      -0.271020   0.120947  -2.241  0.02641 *
sunroofyes   0.260464   0.146356   1.780  0.07703 .
---
```

Linear Models for Location and Scale

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

------------------------------------------------------------------
No. of observations in the fit:  172
Degrees of Freedom for the fit:  12
      Residual Deg. of Freedom:  160
                      at cycle:  9

Global Deviance:     367.297
            AIC:     391.297
            SBC:     429.0669
******************************************************************
```

Linear Models for Location and Scale

Same model with *gamlss2*.

```
R> f <- price ~ age + kilometer + tia + abs + sunroof |
+    age + kilometer + tia + abs + sunroof
R> b2 <- gamlss2(f, data = Golf, family = NO)
```

```
GAMLSS-RS iteration  1: Global Deviance = 372.2874 eps = 0.338513
GAMLSS-RS iteration  2: Global Deviance = 368.5893 eps = 0.009933
GAMLSS-RS iteration  3: Global Deviance = 367.676 eps = 0.002477
GAMLSS-RS iteration  4: Global Deviance = 367.4059 eps = 0.000734
GAMLSS-RS iteration  5: Global Deviance = 367.3277 eps = 0.000212
GAMLSS-RS iteration  6: Global Deviance = 367.306 eps = 0.000058
GAMLSS-RS iteration  7: Global Deviance = 367.2997 eps = 0.000017
GAMLSS-RS iteration  8: Global Deviance = 367.2976 eps = 0.000005
```

Linear Models for Location and Scale

Summary.

```
R> summary(b2)
```

```
Call:
gamlss2(formula = f, data = Golf, family = NO)
---
Family: NO
Link function: mu = identity, sigma = log
*--------
Parameter: mu
---
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.284338   0.553752  14.960  < 2e-16 ***
age         -0.033294   0.003813  -8.731 3.18e-15 ***
kilometer   -0.008145   0.001353  -6.019 1.16e-08 ***
tia         -0.003744   0.007192  -0.521    0.603
absyes      -0.127937   0.130541  -0.980    0.329
```

Linear Models for Location and Scale

```
sunroofyes   0.134633   0.112612   1.196    0.234
*--------
Parameter: sigma
---
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.857300   0.463734   1.849   0.0663 .
age         -0.006638   0.003650  -1.819   0.0708 .
kilometer   -0.004253   0.001668  -2.550   0.0117 *
tia          0.006237   0.007306   0.854   0.3945
absyes      -0.271197   0.120957  -2.242   0.0263 *
sunroofyes   0.261503   0.146348   1.787   0.0759 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
*--------
n = 172 df =  12 res.df =  160
Deviance = 367.2976 Null Dev. Red. = 34.74%
AIC = 391.2976 elapsed =  0.04sec
```

Linear Models for Location and Scale

Full Bayesian model with *bamlss*.

```
R> f <- price ~ age + kilometer + tia + abs + sunroof |
+     age + kilometer + tia + abs + sunroof
R> b3 <- bamlss(f, data = Golf, family = NO)
```

```
AICc 420.0556 logPost -290.967 logLik -197.046 edf 12.000 eps 4.3611 iteration   1
AICc 397.9170 logPost -279.897 logLik -185.977 edf 12.000 eps 0.5396 iteration   2
AICc 394.2334 logPost -278.055 logLik -184.135 edf 12.000 eps 0.1736 iteration   3
AICc 393.5391 logPost -277.708 logLik -183.788 edf 12.000 eps 0.0927 iteration   4
AICc 393.3447 logPost -277.611 logLik -183.691 edf 12.000 eps 0.0684 iteration   5
AICc 393.2869 logPost -277.582 logLik -183.662 edf 12.000 eps 0.0259 iteration   6
AICc 393.2684 logPost -277.573 logLik -183.653 edf 12.000 eps 0.0218 iteration   7
AICc 393.2622 logPost -277.570 logLik -183.650 edf 12.000 eps 0.0124 iteration   8
AICc 393.2601 logPost -277.569 logLik -183.648 edf 12.000 eps 0.0058 iteration   9
AICc 393.2594 logPost -277.568 logLik -183.648 edf 12.000 eps 0.0032 iteration  10
AICc 393.2591 logPost -277.568 logLik -183.648 edf 12.000 eps 0.0019 iteration  11
AICc 393.2590 logPost -277.568 logLik -183.648 edf 12.000 eps 0.0011 iteration  12
AICc 393.2590 logPost -277.568 logLik -183.648 edf 12.000 eps 0.0006 iteration  13
```

Linear Models for Location and Scale

```
AICc 393.2590 logPost -277.568 logLik -183.648 edf 12.000 eps 0.0003 iteration  14
AICc 393.2590 logPost -277.568 logLik -183.648 edf 12.000 eps 0.0002 iteration  15
AICc 393.2590 logPost -277.568 logLik -183.648 edf 12.000 eps 0.0001 iteration  16
AICc 393.2590 logPost -277.568 logLik -183.648 edf 12.000 eps 0.0000 iteration  17
AICc 393.2590 logPost -277.568 logLik -183.648 edf 12.000 eps 0.0000 iteration  17
elapsed time:  0.03sec
Starting the sampler...

|                    |   0%  1.31sec
|*                   |   5%  0.87sec  0.05sec
|**                  |  10%  0.82sec  0.09sec
|***                 |  15%  0.78sec  0.14sec
|****                |  20%  0.74sec  0.18sec
|*****               |  25%  0.71sec  0.24sec
|******              |  30%  0.67sec  0.29sec
|*******             |  35%  0.62sec  0.34sec
|********            |  40%  0.58sec  0.39sec
|*********           |  45%  0.54sec  0.44sec
```

Linear Models for Location and Scale

```
|**********          |  50%  0.49sec  0.49sec
|***********         |  55%  0.44sec  0.54sec
|************        |  60%  0.39sec  0.59sec
|*************       |  65%  0.34sec  0.64sec
|**************      |  70%  0.30sec  0.69sec
|***************     |  75%  0.25sec  0.74sec
|****************    |  80%  0.20sec  0.79sec
|*****************   |  85%  0.15sec  0.84sec
|******************  |  90%  0.10sec  0.89sec
|******************* |  95%  0.05sec  0.94sec
|********************| 100%  0.00sec  0.99sec
```

Linear Models for Location and Scale

Summary based on MCMC samples.

```
R> summary(b3)
```

```
Call:
bamlss(formula = f, family = NO, data = Golf)
---
Family: NO
Link function: mu = identity, sigma = log
*---
Formula mu:
---
price ~ age + kilometer + tia + abs + sunroof
-
Parametric coefficients:
                 Mean      2.5%       50%     97.5% parameters
(Intercept)  8.305861  7.241521  8.314143  9.430135      8.271
age         -0.033500 -0.040635 -0.033443 -0.025847     -0.033
kilometer   -0.008202 -0.010964 -0.008147 -0.005653     -0.008
```

Linear Models for Location and Scale

```
tia         -0.003499 -0.017816 -0.003382  0.009819     -0.004
absyes      -0.121107 -0.389325 -0.119713  0.142292     -0.126
sunroofyes   0.133672 -0.103824  0.136533  0.358623      0.135
-
Acceptance probability:
      Mean 2.5% 50% 97.5%
alpha    1    1   1     1
---
Formula sigma:
---
~age + kilometer + tia + abs + sunroof
-
Parametric coefficients:
                  Mean       2.5%        50%      97.5% parameters
(Intercept)  0.9075229  0.0118719  0.8960075  1.7911369      0.864
age         -0.0069480 -0.0143901 -0.0069893  0.0001632     -0.007
kilometer   -0.0039761 -0.0072541 -0.0039131 -0.0005404     -0.004
tia          0.0057453 -0.0084837  0.0060793  0.0185759      0.006
```

Linear Models for Location and Scale

```
absyes      -0.2849893 -0.5477280 -0.2807292 -0.0536511     -0.271
sunroofyes   0.2541777 -0.0501468  0.2600088  0.5288864      0.259
-
Acceptance probability:
         Mean    2.5%     50% 97.5%
alpha 0.68255 0.02975 0.76063     1
---
Sampler summary:
-
DIC = 390.9319 logLik = -189.6465 pd = 11.6389
runtime = 0.999
---
Optimizer summary:
-
AICc = 393.259 edf = 12 logLik = -183.6484
logPost = -277.5687 nobs = 172 runtime = 0.025
```

Linear Models for Location and Scale

Model using *mgcv*.

```
R> f <- list(
+    price ~ age + kilometer + tia + abs + sunroof,
+          ~ age + kilometer + tia + abs + sunroof
+  )
R> b4 <- gam(f, data = Golf, family = gaulss)
```

Linear Models for Location and Scale

Model summary.

```
R> summary(b4)
```

```
Family: gaulss 
Link function: identity logb 

Formula:
price ~ age + kilometer + tia + abs + sunroof
~age + kilometer + tia + abs + sunroof

Parametric coefficients:
               Estimate Std. Error z value Pr(>|z|)    
(Intercept)    8.270140   0.552805  14.960  < 2e-16 ***
age           -0.033216   0.003808  -8.723  < 2e-16 ***
kilometer     -0.008124   0.001349  -6.023 1.71e-09 ***
tia           -0.003740   0.007186  -0.520   0.6028    
absyes        -0.126255   0.130400  -0.968   0.3329    
sunroofyes     0.134850   0.112502   1.199   0.2307    
```

Linear Models for Location and Scale

```
(Intercept).1  0.866291   0.471320   1.838   0.0661 .
age.1         -0.006725   0.003738  -1.799   0.0720 .
kilometer.1   -0.004360   0.001724  -2.530   0.0114 *
tia.1          0.006320   0.007421   0.852   0.3944
absyes.1      -0.275462   0.122453  -2.250   0.0245 *
sunroofyes.1   0.263642   0.148391   1.777   0.0756 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Deviance explained = 62.8%
-REML = 221.32  Scale est. = 1         n = 172
```

Linear Models for Location and Scale

Model coefficients.

```
R> coef(b1, what = "mu")
```

```
 (Intercept)          age    kilometer          tia       absyes   sunroofyes 
 8.277807909 -0.033259425 -0.008133827 -0.003751773 -0.127191254  0.135028174 
```

```
R> coef(b1, what = "sigma")
```

```
 (Intercept)          age    kilometer          tia       absyes   sunroofyes 
 0.860776377 -0.006643862 -0.004269571  0.006237025 -0.271019705  0.260464008 
```

```
R> coef(b2)
```

```
   mu.p.(Intercept)            mu.p.age      mu.p.kilometer            mu.p.tia 
        8.284338108        -0.033293975        -0.008144860        -0.003743901 
        mu.p.absyes     mu.p.sunroofyes sigma.p.(Intercept)         sigma.p.age 
       -0.127936780         0.134632970         0.857299923        -0.006638287 
  sigma.p.kilometer         sigma.p.tia      sigma.p.absyes  sigma.p.sunroofyes 
       -0.004253350         0.006237328        -0.271196578         0.261503338 
```

Linear Models for Location and Scale

Model coefficients.

```
R> coef(b3)
                            Mean         2.5%          50%         97.5%
mu.p.(Intercept)     8.305860714  7.241521273  8.314143470  9.4301353228
mu.p.age            -0.033500058 -0.040634582 -0.033443291 -0.0258465288
mu.p.kilometer      -0.008202177 -0.010964208 -0.008146909 -0.0056529716
mu.p.tia            -0.003499014 -0.017815714 -0.003381955  0.0098193863
mu.p.absyes         -0.121107273 -0.389324673 -0.119712764  0.1422915237
mu.p.sunroofyes      0.133671957 -0.103823813  0.136533318  0.3586228726
mu.p.alpha           0.999997967  0.999990111  0.999999960  1.0000000000
sigma.p.(Intercept)  0.907522910  0.011871905  0.896007451  1.7911369233
sigma.p.age         -0.006947958 -0.014390111 -0.006989327  0.0001631742
sigma.p.kilometer   -0.003976115 -0.007254095 -0.003913125 -0.0005404387
sigma.p.tia          0.005745251 -0.008483748  0.006079255  0.0185759491
sigma.p.absyes      -0.284989306 -0.547728007 -0.280729210 -0.0536510908
sigma.p.sunroofyes   0.254177731 -0.050146801  0.260008756  0.5288863980
sigma.p.alpha        0.682552257  0.029746415  0.760631008  1.0000000000
```

Linear Models for Location and Scale

Model coefficients.

```
R> coef(b4)
  (Intercept)           age     kilometer           tia        absyes 
  8.270140470  -0.033216185  -0.008123646  -0.003739547  -0.126255437 
   sunroofyes (Intercept).1         age.1   kilometer.1         tia.1 
  0.134849864   0.866291267  -0.006725209  -0.004360207   0.006320495 
     absyes.1  sunroofyes.1 
 -0.275462193   0.263642254 
```

The *gamlss.dist* package

- The *gamlss.dist* package provides a collection of distributions (families).
- It extends the capabilities of GAMLSS by offering a wide range of probability distributions beyond the default ones available in the GAMLSS framework.
- The package enables users to fit flexible (and customized) distributions to their data, allowing for greater flexibility.
- Each *gamlss.dist* distribution/family has a `d*()`, `p*()`, `q*()` and `r*()` function. E.g, `dNO()`, `pNO()`, `qNO()` and `rNO()` for the normal distribution.

The *gamlss.dist* package

Continuous distributions:

```
R> print(names(bamlss::gamlss_distributions(type = "continuous")))
 [1] "BCCG"     "BCCGo"    "BCCGuntr" "BCPE"     "BCPEo"    "BCPEuntr"
 [7] "BCT"      "BCTo"     "BCTuntr"  "BE"       "BEo"      "EGB2"    
[13] "exGAUS"   "EXP"      "GA"       "GAF"      "GB1"      "GB2"     
[19] "GG"       "GIG"      "GT"       "GU"       "IG"       "IGAMMA"  
[25] "JSU"      "JSUo"     "LNO"      "LO"       "LOGITNO"  "LOGNO"   
[31] "LOGNO2"   "LQNO"     "NET"      "NO"       "NO2"      "NOF"     
[37] "PARETO"   "PARETO1"  "PARETO1o" "PARETO2"  "PARETO2o" "PE"      
[43] "PE2"      "RG"       "RGE"      "SEP"      "SEP1"     "SEP2"    
[49] "SEP3"     "SEP4"     "SHASH"    "SHASHo"   "SHASHo2"  "SIMPLEX" 
[55] "SN1"      "SN2"      "SST"      "ST1"      "ST2"      "ST3"     
[61] "ST3C"     "ST4"      "ST5"      "TF"       "TF2"      "WEI"     
[67] "WEI2"     "WEI3"    
```

The *gamlss.dist* package

Discrete distributions:

```
R> print(names(bamlss::gamlss_distributions(type = "discrete")))
 [1] "BB"       "BI"       "BNB"      "DBI"      "DBURR12"  "DEL"     
 [7] "DPO"      "GEOM"     "GEOMo"    "GPO"      "LG"       "MN3"     
[13] "MN4"      "MN5"      "MULTIN"   "NBF"      "NBI"      "NBII"    
[19] "PIG"      "PIG2"     "PO"       "SI"       "SICHEL"   "WARING"  
[25] "YULE"     "ZABB"     "ZABI"     "ZABNB"    "ZALG"     "ZANBI"   
[31] "ZAP"      "ZAPIG"    "ZASICHEL" "ZAZIPF"   "ZIBB"     "ZIBI"    
[37] "ZIBNB"    "ZINBF"    "ZINBI"    "ZIP"      "ZIP2"     "ZIPF"    
[43] "ZIPIG"    "ZISICHEL"
```

The *gamlss.dist* package

Example: Density of `BCPE()` distribution.

Load the package.

```
R> library("gamlss.dist")
```

Some (reponse) data.

```
R> y <- seq(0, 10, length = 300)
```

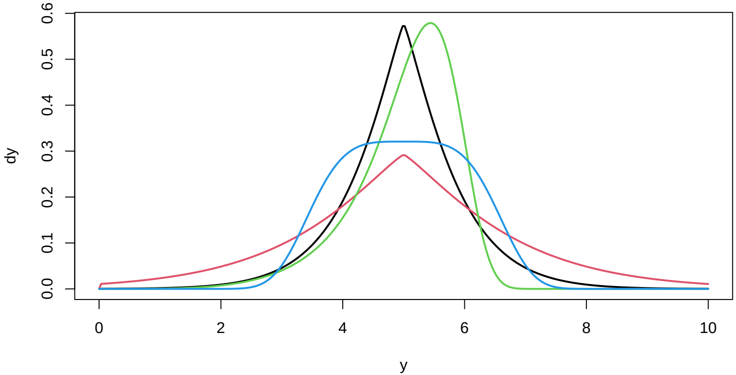
Compute densities.

```
R> dy <- cbind(
+    dBCPE(y, mu = 5, sigma = 0.2, nu = 1, tau = 1.2),
+    dBCPE(y, mu = 5, sigma = 0.4, nu = 1, tau = 1.2),
+    dBCPE(y, mu = 5, sigma = 0.2, nu = 5, tau = 2.0),
+    dBCPE(y, mu = 5, sigma = 0.2, nu = 1, tau = 4.0)
+  )
```

The *gamlss.dist* package

Plot density curves.

```
R> par(mar = c(4, 4, 0.5, 0.5))
R> matplot(y, dy, type = "l", lty = 1, lwd = 2)
```



The *gamlss.dist* package

Example: Log-likelihood esimation of parameters.

Load the `rent` data.

```
R> data("rent", package = "gamlss.data")
```

Define the negative log-likelihood function for the gamma distribution.

```
R> nloglik <- function(theta, y) {
+    ll <- sum(dGA(y, mu = theta[1], sigma = theta[2], log = TRUE))
+    return(-ll)
+  }
```

The *gamlss.dist* package

Optimize parameters using general purpose optimizer `optim()`

```
R> par <- optim(c("mu" = 1, "sigma" = 1), fn = nloglik, y = rent$R,
+    method = "L-BFGS-B", lower = c(0, 0), upper = c(Inf, Inf))
R> print(par[1:2])
```

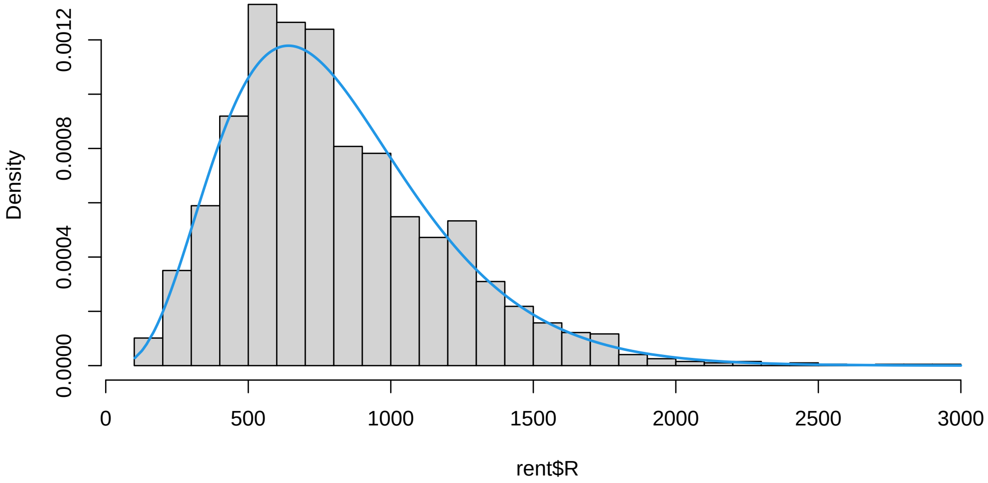
```
$par
         mu       sigma
811.8770220   0.4589693

$value
[1] 14305.79
```

The *gamlss.dist* package

Visualize estimated density.

```
R> dR <- dGA(rent$R, mu = par$par["mu"], sigma = par$par["sigma"])
R> par(mar = c(4, 4, 0.5, 0.5))
R> hist(rent$R, freq = FALSE, breaks = "Scott", main = "")
R> i <- order(rent$R)
R> lines(dR[i] ~ rent$R[i], lwd = 2, col = 4)
```

The *gamlss.dist* package



The *gamlss.dist* package

Estimation with *gamlss*.

```
R> b <- gamlss(R ~ 1, data = rent, family = GA)
```

```
GAMLSS-RS iteration 1: Global Deviance = 28611.58
GAMLSS-RS iteration 2: Global Deviance = 28611.58
```

```
R> cb <- c(
+     "mu" = coef(b, what = "mu"),
+     "sigma" = coef(b, what = "sigma")
+   )
R> print(cb)
```

```
   mu.(Intercept) sigma.(Intercept)
        6.6993530        -0.7787806
```

```
R> print(exp(cb))
```

```
   mu.(Intercept) sigma.(Intercept)
      811.8803454         0.4589653
```

The *gamlss.dist* package

Estimation with *bamlss*.

```
R> b <- bamlss(R ˜ 1, data = rent, family = GA)
```

```
AICc 29220.17 logPost -14623.7 logLik -14608.0 edf 2.0000 eps 0.5134 iteration   1
AICc 28696.89 logPost -14362.0 logLik -14346.4 edf 2.0000 eps 0.2212 iteration   2
AICc 28617.33 logPost -14322.3 logLik -14306.6 edf 2.0000 eps 0.0823 iteration   3
AICc 28615.58 logPost -14321.4 logLik -14305.7 edf 2.0000 eps 0.0128 iteration   4
AICc 28615.58 logPost -14321.4 logLik -14305.7 edf 2.0000 eps 0.0002 iteration   5
AICc 28615.58 logPost -14321.4 logLik -14305.7 edf 2.0000 eps 0.0000 iteration   6
AICc 28615.58 logPost -14321.4 logLik -14305.7 edf 2.0000 eps 0.0000 iteration   6
elapsed time:  0.03sec
Starting the sampler...

|                    |   0%  3.45sec
|*                   |   5%  3.46sec  0.18sec
|**                  |  10%  3.26sec  0.36sec
|***                 |  15%  3.09sec  0.55sec
|****                |  20%  3.38sec  0.85sec
```

The *gamlss.dist* package

```
|*****               |  25%  3.17sec  1.06sec
|******              |  30%  2.95sec  1.27sec
|*******             |  35%  2.75sec  1.48sec
|********            |  40%  2.54sec  1.69sec
|*********           |  45%  2.33sec  1.91sec
|**********          |  50%  2.12sec  2.12sec
|***********         |  55%  1.91sec  2.33sec
|************        |  60%  1.69sec  2.54sec
|*************       |  65%  1.48sec  2.75sec
|**************      |  70%  1.27sec  2.96sec
|***************     |  75%  1.06sec  3.18sec
|****************    |  80%  0.85sec  3.39sec
|*****************   |  85%  0.64sec  3.60sec
|******************  |  90%  0.42sec  3.81sec
|******************* |  95%  0.21sec  4.03sec
|********************| 100%  0.00sec  4.24sec
```

The *gamlss.dist* package

```
R> cb <- c(
+    "mu" = coef(b, model = "mu", FUN = mean, hyper = FALSE),
+    "sigma" = coef(b, model = "sigma", FUN = mean, hyper = FALSE)
+  )
R> print(cb)
      mu.mu.p.(Intercept) sigma.sigma.p.(Intercept)
                6.6990121                -0.7780353
R> print(exp(cb))
      mu.mu.p.(Intercept) sigma.sigma.p.(Intercept)
              811.6036219                 0.4593075
```