

wle: A Package for Robust Statistics using Weighted Likelihood

by *Claudio Agostinelli*

The **wle** is a package for robust statistics using the weighted-likelihood estimating equations approach. This approach is different in many aspects from that presented in [Huber \(1981\)](#) and [Hampel et al. \(1986\)](#). It provides a general framework so that extensions are simpler than in the classical setting. The main feature is to provide first-order efficient (asymptotically) and robust estimators in the sense of breakdown. The current version (0.6-1) of the package implements most of the results presented in the literature.

In the next section we will introduce the weighted likelihood methodology and review the present literature. In the section *Package features* we will give some details about the current release and we will provide an example of some functions.

Weighted likelihood

The definition of Weighted Likelihood Estimating Equations (WLEE) was first proposed by [Markatou et al. \(1997\)](#) for discrete distributions, then in [Markatou et al. \(1998\)](#) the methods were extended to continuous models.

Let x_1, x_2, \dots, x_n be an i.i.d. sample from the random variable X with unknown density $f(\cdot)$ corresponding to the probability measure $F(\cdot)$. We will use the density $m(\cdot; \theta)$ corresponding to the probability measure $M(\cdot; \theta)$ and $\theta \in \Theta$ as a model for the random variable X . Note that in the maximum-likelihood context we assume $f(\cdot) \equiv m(\cdot; \theta_T)$ (almost surely) and $\theta_T \in \Theta$. Let $u(x; \theta) = \frac{\partial}{\partial \theta} \log m(x; \theta)$ be the score function. Under regularity conditions the maximum likelihood estimator of θ is a solution of the likelihood equation $\sum_{i=1}^n u(x_i; \theta) = 0$.

Given any point x in the sample space, [Markatou et al. \(1998\)](#) construct a weight function $w(x; \theta, \hat{F}_n)$ that depends on the chosen model distribution M and the empirical cumulative distribution $\hat{F}_n(t) = \sum_{i=1}^n \mathbf{1}_{x_i < t} / n$. Estimators for the parameter vector θ are obtained as solutions to the set of estimating equations:

$$\sum_{i=1}^n w(x_i; \theta, \hat{F}_n) u(x_i; \theta) = 0 \quad (1)$$

The weight function

$$w(x; \theta, \hat{F}_n) = \min \left\{ 1, \frac{[A(\delta(x; \theta, \hat{F}_n)) + 1]^+}{\delta(x; \theta, \hat{F}_n) + 1} \right\}$$

(where $[\cdot]^+$ indicates the positive part) takes values in the interval $[0, 1]$ by construction.

The quantity $\delta(x; \theta, \hat{F}_n)$ is called the *Pearson residual*, defined as $\delta(x; \theta, \hat{F}_n) = f^*(x) / m^*(x; \theta) - 1$, where $f^*(x) = \int k(x; t, h) d\hat{F}_n(t)$ is a kernel density estimator and $m^*(x; \theta) = \int k(x; t, h) dM(t; \theta)$ is the smoothed model density. Note that sometimes $f^*(x)$ is a function of θ as in the regression case. The Pearson residual expresses the agreement between the data and the assumed probability model. The function $A(\cdot)$ is a *residual adjustment function*, RAF, ([Lindsay, 1994](#)) and it operates on Pearson residuals in the same way as the Huber ψ -function operates on the structural residuals. When $A(\delta) = \delta$ we have $w(x; \theta, \hat{F}_n) \equiv 1$, and this corresponds to maximum likelihood. Generally, the weights w use functions $A(\cdot)$ that correspond to a minimum disparity problem. For example, the function $A(\delta) = 2\{(\delta + 1)^{1/2} - 1\}$ corresponds to Hellinger distance. For an extensive discussion of the concept of RAF see [Lindsay \(1994\)](#).

This weighting scheme provides first-order efficient (asymptotically) and robust estimators in the sense of breakdown, provided that one selects a root by using the parallel disparity measure ([Markatou et al., 1998](#)). However, the inspection of all roots is useful for diagnostics and data analysis.

The estimating equations (1) are solved using a re-weighting scheme. An algorithm based on re-sampling techniques is used to identify the roots and to provide starting values. Sub-samples of fixed dimension and without replication are sampled from the dataset. From each of these sub-samples a maximum likelihood estimator is evaluated and used to start the re-weighted algorithm.

To calculate the Pearson residuals we need to select the smoothing parameter h . [Markatou et al. \(1998\)](#) select $h^2 = g\sigma^2$, where g is a constant independent of the scale of the model which is selected in a way such that it assigns a very small weight to an outlying observation ([Agostinelli and Markatou, 2001](#)).

To illustrate the behaviour of the weight function, let us consider its asymptotic value when the data come from a mixture of two normal distributions, $f(x) = 0.9N(0, 1) + 0.1N(4, 1)$ (Figure 1). We set $g = 0.003$ with a normal kernel and we use a location normal family ($M = \{N(\theta, 1), \theta \in \mathcal{R}\}$) as a model for these data. In Figure 2 we report the Pearson residuals evaluated in the distribution of the majority of the data, that is in $\theta = 0$, while in Figure 3 we report the corresponding weight function based on the Hellinger Residual Adjustment Function.

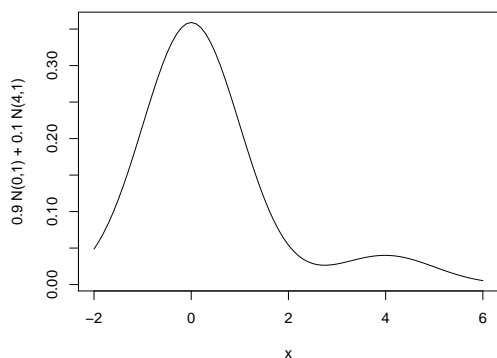


Figure 1: The contaminated normal density distribution

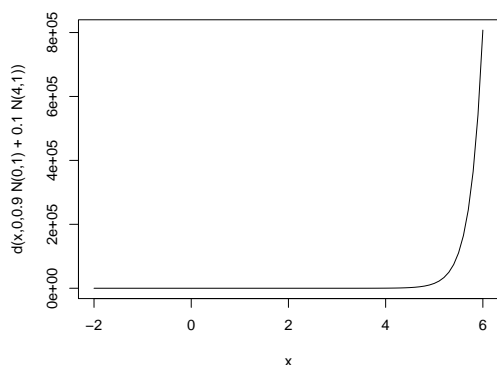


Figure 2: The (asymptotic) Pearson residuals.

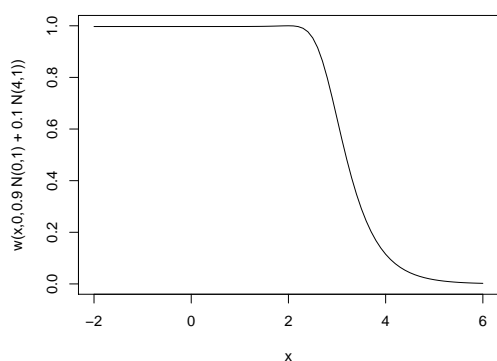


Figure 3: The (asymptotic) weights based on Hellinger Residual Adjustment Function.

Agostinelli (1998a,b) extended the methodology to the regression model while Agostinelli and Markatou (1998) studied the one-step estimator based on high breakdown initial estimator. Agostinelli (2000, 2001a,b) developed robust model selection procedures based on weighted ver-

sions of Akaike Information Criterion, Mallows C_p , Cross-Validation and Stepwise. Agostinelli (1998a, 2001d) and Agostinelli and Markatou (2001) defined weighted versions of the classical likelihood test functions: likelihood ratio, Wald and Rao (or score) tests. Markatou (2000, 2001) studied the estimation problem in a mixture model. Agostinelli (2001c) proposed estimation procedures for ARMA models.

Package features

Version 0.6-1 of package `wle` implements almost all the methods and procedures presented above. In particular, there are functions providing estimates the parameters for the *binomial* and *Poisson* models, and the univariate and multivariate *normal* and *gamma* models. A function is devoted to the *regression* model with normal errors and model-selection procedures are available for this case. The *weighted t-test* is available for one and two samples (paired and unpaired), with a function that works very similarly to the `t.test` function. The *weighted F-test* (Agostinelli, 2001b) may be used for comparison of two variances for objects generated by `wle.normal` and `wle.lm` in the same way as the `var.test` function. Finally, a preliminary version for estimating the parameters of a *univariate normal mixture* models is available.

In the following example we illustrate the functions `wle.lm` and `wle.cv` together with the related methods. We generated a dataset of 70 observations. The first 60 observations follow the $Y = 8 \log(X + 1) + \varepsilon$ regression model with $\varepsilon \sim N(0, 0.6)$ while the last 10 observations are a cluster of observations from the same model but with residuals from $\varepsilon \sim N(-4, 0.6)$. The contamination level is about 14%.

```
> library(wle)
> set.seed(1234)

> x.data <- c(runif(60,20,80), runif(10,73,78))
> e.data <- rnorm(70,0,0.6)
> y.data <- 8*log(x.data+1)+e.data
> y.data[61:70] <- y.data[61:70] - 4
>
> x.model <- function(x) 8*log(x+1)
> x.log.data <- x.model(x.data)/8
```

First, we show how the `wle.lm` works: see Figure 4 on page 35. The function uses formula for describing the regression model structure as in `lm`. The most important parameters are `boot`, `group` and `num.sol`. The first parameter is used to control the number of bootstrap sub-samples, i.e., the number of starting values the function has to use in order to look for different roots of the estimating equation.

The second parameter is the size of the sub-samples; it can not be less than the number of the

unknown parameters since we have to obtain maximum likelihood estimates from those sub-samples. [Markatou et al. \(1998\)](#) empirically found that in most cases, it was sufficient to let `group` equal to the number of parameters to be estimated in order to produce reasonable estimates. On the other hand, in particular cases, this could raise some problems, for instance in the presence of highly correlated explanatory variables. For this reason the default value is the maximum of the number of parameters and one quarter of the sample size. In our example we set `group` equal to the number of parameters.

The third parameter `num.sol` controls the maximum number of roots we expect to find. The algorithm is stopped when it has found `num.sol` roots, regardless of the number of bootstrap replications. Two roots are considered to be distinct if they have at least one component bigger than the `equal` parameter in absolute difference.

The `wle.lm` function has `summary` and `plot` methods. The summary is very similar to that generated for `lm`: one summary is reported for each root. `wle.lm` has found the “robust” root and the MLE-like root. The t-test is performed accordingly with the weighted Wald test for each root ([Agostinelli, 2001d](#); [Agostinelli and Markatou, 2001](#)). Moreover, the weighted residuals are obtained as `weights * residuals` without squaring the weights. In Figure 5 we present the dataset, the true model, the two models suggested by the weighted likelihood and the one found by maximum likelihood.

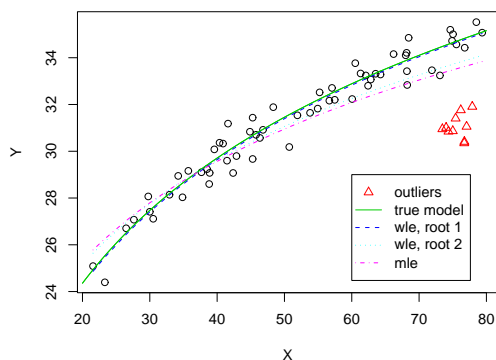


Figure 5: The dataset and the estimated model by `wle.lm` and `lm`.

The `plot` method helps to understand the difference between the roots. The first plot represents in the main diagonal the weights related to each root. The observations with weights close to one are shown in green, while the possible outliers with weights close to zero are displayed in red. The threshold is set by `level.weight` (with default value 0.5). In the lower triangle we compare the weight given by different roots to each observation while in the upper triangle the (unweighted)

residuals are compared; the bisector is reported.

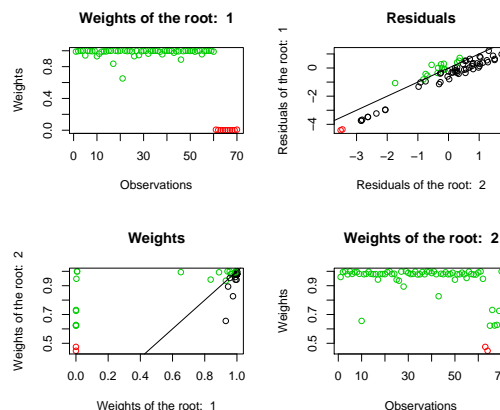


Figure 6: Plot from the `plot` method.

Then, for each root we present the qq-norm of the unweighted and weighted residuals and residuals vs fitted values plots. The observations with weights less than `level.weight` are reported in red.

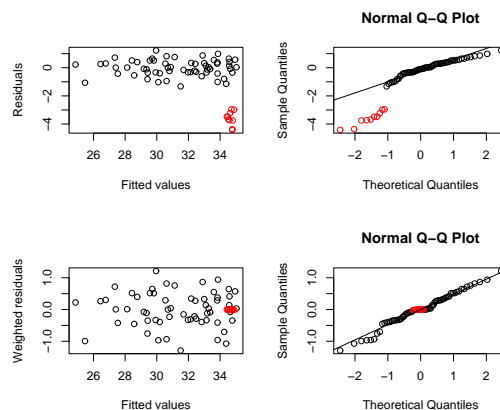


Figure 7: Plot from the `plot` method, root 1.

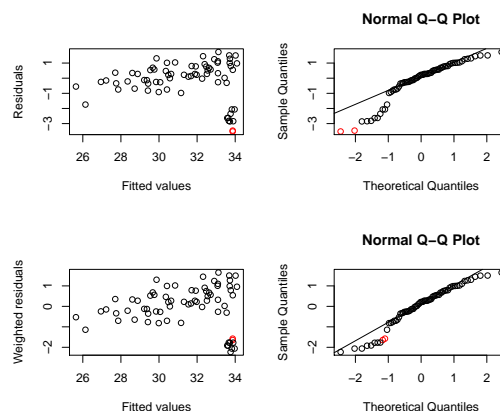


Figure 8: Plot from the `plot` method, root 2.

Now, we try to identify a good model for our dataset given a set of possible explanatory variables that include the “true” ones. Such variables are

```

> wle.lm.result <-
+ wle.lm(y.data~x.log.data, boot=50,
+       group=3, num.sol=3)
> summary(wle.lm.result)

Call:
wle.lm(formula = y.data ~ x.log.data, boot = 50,
       group = 3, num.sol = 3)

Root 1

Weighted Residuals:
      Min       1Q   Median       3Q      Max
-1.30752 -0.32307 -0.04171  0.32204  1.21939

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.09666    0.90864  -0.106   0.916
x.log.data   8.00914    0.23108  34.660 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01
                 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5705 on 56.80948 degrees of freedom
Multiple R-Squared: 0.9548,
Adjusted R-squared: 0.9452
F-statistic: 1201 on 1 and 56.80948
degrees of freedom, p-value: 0

Call:
wle.lm(formula = y.data ~ x.log.data, boot = 50,
       group = 3, num.sol = 3)

Root 2

Weighted Residuals:
      Min       1Q   Median       3Q      Max
-2.3863 -0.4943  0.2165  0.7373  1.6909

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.9007    1.7282   2.836 0.00612 **
x.log.data    6.6546    0.4338  15.340 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01
                 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.151 on 63.58554 degrees of freedom
Multiple R-Squared: 0.7873,
Adjusted R-squared: 0.7692
F-statistic: 235.3 on 1 and 63.58554
degrees of freedom, p-value: 0

```

Figure 4: Using wle.lm

highly correlated, and a model with just one of them could be a good model.

```
> xx <- cbind(x.data,x.data^2,x.data^3,
+           log(x.data+1))
> colnames(xx) <- c("X","X^2","X^3","log(X+1)")
> cor(xx)
```

	X	X ²	X ³	log(X+1)
X	1.0000000	0.9913536	0.9718489	0.9879042
X ²	0.9913536	1.0000000	0.9942109	0.9595103
X ³	0.9718489	0.9942109	1.0000000	0.9253917
log(X+1)	0.9879042	0.9595103	0.9253917	1.0000000

We address the problem by using `wle.cv` which performs Weighted Cross-Validation. For comparison we use `mle.cv` which performs the classical Cross-Validation procedure (Shao, 1993). Since the procedure uses weights based on the full model, a crucial problem arises when multiple roots are present in this model. Currently the package chooses the root with the smallest scale parameter: this should work fine in most cases. In the next release of the package we will give the users the opportunity to choose the root by themselves. As seen in Figure 9 on page 37, while `wle.cv` suggests the “true” model, `mle.cv` chooses models with three explanatory variables.

Next, we estimate the suggested model; only one root is found (see Figure 10 on page 38). In Figure 11 we report the suggested models by the weighted likelihood and the classical procedure.

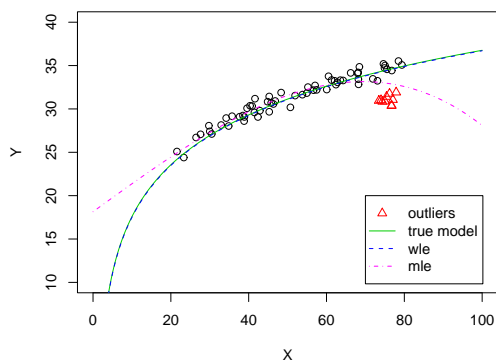


Figure 11: The model suggested by `wle.cv` and `mle.cv`.

Future developments

The next version of the package will probably include functions for autoregressive time-series. In particular, there will be functions for seasonal ARI models, with weighted autocorrelation functions, unit-root tests and perhaps a model selection procedure based on weighted Akaike Information Criterion for the order of the ARI model. Functions will be provided for the logistic regression model. Moreover, we will start to prepare a document to illustrate the use of the functions in applications.

Bibliography

- C. Agostinelli. *Inferenza statistica robusta basata sulla funzione di verosimiglianza pesata: alcuni sviluppi*. PhD thesis, Dipartimento di Scienze Statistiche, Università di Padova, 1998a. 33
- C. Agostinelli. Verosimiglianza pesata nel modello di regressione lineare. In *XXXIX Riunione scientifica della Società Italiana di Statistica*, Sorrento, 1998b. 33
- C. Agostinelli. Robust model selection by cross-validation via weighted likelihood methodology. Submitted to *Australian and New Zealand Journal of Statistics*, 2000. 33
- C. Agostinelli. Robust model selection in regression via weighted likelihood methodology. To appear in *Statistics & Probability Letters*, 2001a. 33
- C. Agostinelli. Robust stepwise regression. To appear in *Journal of Applied Statistics*, 2001b. 33
- C. Agostinelli. Robust time series estimation via weighted likelihood. Presented to the first International Conference on Robust Statistics, (poster session), Vorau, Austria, 2001c. 33
- C. Agostinelli. Un approccio robusto alla verifica d'ipotesi basato sulla funzione di verosimiglianza pesata – robust testing hypotheses via weighted likelihood function. To appear in *Statistica*, 2001d. In Italian. 33, 34
- C. Agostinelli and M. Markatou. A one-step robust estimator for regression based on the weighted likelihood reweighting scheme. *Statistics & Probability Letters*, 37(4):341–350, 1998. 33
- C. Agostinelli and M. Markatou. Test of hypotheses based on the weighted likelihood methodology. *Statistica Sinica*, 11(2):499–514, 2001. 32, 33, 34
- F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach based on Influence Functions*. John Wiley, New York, 1986. 32
- P. J. Huber. *Robust Statistics*. John Wiley, New York, 1981. 32
- B. G. Lindsay. Efficiency versus robustness: The case for minimum hellinger distance and related methods. *Annals of Statistics*, 22:1018–1114, 1994. 32
- M. Markatou. Mixture models, robustness and the weighted likelihood methodology. *Biometrics*, 56: 483–486, 2000. 33

```

> wle.cv.result <- wle.cv(y.data~xx, boot=50,
+                          group=6, num.sol=3)
> summary(wle.cv.result, num.max=10)

Call:
wle.cv(formula = y.data ~ xx, boot = 50,
        group=6, num.sol = 3)
      (Intercept) xxX xxX^2 xxX^3 xxlog(X+1)      wcv
[1,]           0  0    0    0           1 0.3395
[2,]           0  1    0    0           1 0.3631
[3,]           0  0    1    0           1 0.3632
[4,]           0  0    0    1           1 0.3635
[5,]           1  0    0    0           1 0.3639
[6,]           0  0    1    1           1 0.3868
[7,]           0  1    0    1           1 0.3881
[8,]           0  1    1    0           1 0.3896
[9,]           1  0    0    1           1 0.3925
[10,]          1  0    1    0           1 0.3951

Printed the first 10 best models

> mle.cv.result <- mle.cv(y.data~xx)
> summary(mle.cv.result, num.max=10)

Call:
mle.cv(formula = y.data ~ xx)

Cross Validation selection criteria:
      (Intercept) xxX xxX^2 xxX^3 xxlog(X+1)      cv
[1,]           1  1    0    1           0 1.557
[2,]           1  1    1    0           0 1.560
[3,]           0  0    1    1           1 1.579
[4,]           0  1    0    1           1 1.581
[5,]           0  0    0    1           1 1.584
[6,]           0  1    1    0           1 1.589
[7,]           1  0    1    1           0 1.593
[8,]           1  0    0    1           1 1.594
[9,]           1  0    1    0           1 1.617
[10,]          0  0    1    0           1 1.620

Printed the first 10 best models

```

Figure 9: Finding a good model using wle.cv and mle.cv

```

> wle.lm.result.cv <- wle.lm(y.data~x.log.data
+   -1, boot=50, group=3, num.sol=3)
> summary(wle.lm.result.cv)

Call:
wle.lm(formula = y.data ~ x.log.data - 1,
       boot = 50, group = 3, num.sol = 3)

Root 1

Weighted Residuals:
      Min       1Q   Median       3Q      Max
-1.30476 -0.32233 -0.03861  0.32276  1.21646

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
x.log.data  7.98484   0.01874   426.1  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01
                 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5649 on 57.76524 degrees of freedom
Multiple R-Squared:  0.9997,
Adjusted R-squared:  0.9996
F-statistic: 1.815e+05 on 1 and 57.76524
degrees of freedom, p-value: 0

```

Figure 10: Fitting the model suggested by weighted cross-validation.

M. Markatou. A closer look at the weighted likelihood in the context of mixtures. In C. A. Charalambides, M. V. Koutras, and N. Balakrishnan, editors, *Probability and Statistical Models with Applications*, pages 447–467. Chapman and Hall/CRC, 2001. [33](#)

M. Markatou, A. Basu, and B. G. Lindsay. Weighted likelihood estimating equations: The discrete case with applications to logistic regression. *Journal of Statistical Planning and Inference*, 57:215–232, 1997. [32](#)

M. Markatou, A. Basu, and B. G. Lindsay. Weighted likelihood estimating equations with a bootstrap

root search. *Journal of the American Statistical Association*, 93:740–750, 1998. [32](#), [34](#)

J. Shao. Linear model selection by cross-validation. *Journal of the American Statistical Association*, 88: 486–494, 1993. [36](#)

Claudio Agostinelli
 Dipartimento di Statistica
 Università Cà Foscari di Venezia
 30125, Venezia
claudio@unive.it

Changes on CRAN

by Kurt Hornik and Friedrich Leisch

CRAN packages

The following extension packages from ‘src/contrib’ were added since the last newsletter.

AnalyzefMRI Functions for I/O, visualisation and analysis of functional Magnetic Resonance Imaging (fMRI) datasets stored in the ANALYZE format. By J L Marchini.

EMV Estimation of missing values in a matrix by a k -th nearest neighbors algorithm. By Raphael Gottardo.

Rwave Rwave is a collection of R functions which provide an environment for the Time-Frequency analysis of 1-D signals (and especially for the wavelet and Gabor transforms of noisy signals). It is based on the book: ‘Practical Time-Frequency Analysis: Gabor and Wavelet Transforms with an Implementation in S’, by Rene Carmona, Wen L. Hwang and