# Manual of functions contained in `Functions-for-far-WeibullEstimation.R`

*Julien Worms (UVSQ,* `julien.worms@uvsq.fr`*)*

The R code file `Functions-for-far-WeibullEstimation.R` contains material useful for applying the methods detailed in the paper *Record events attribution for climate studies* (P.Naveau & J.Worms). This documents aims at describing the main functions and providing an example of use.

## 1. A single standard function for an easy treatment of one setting

If the user only has a single framework to study, *i.e.* has at disposal a single counterfactual dataset $X = (X_1, \dots, X_m)$ and a single factual dataset $Z = (Z_1, \dots, Z_n)$, then he/she can consider to use the following "all-in-one" function `farrecord()`, which directly produces the estimates, standard errors, and confidence intervals, for $p_{1,r}$ and $far(r)$ (for a chosen series of values of $r$). It also provides a $p$-value of the WLK weibullity test according to the procedure described in the paper.

However, if the user's data is of larger size, we advise to proceed step by step by using the functions described in part 2.

More precisely, the inputs of function `farrecord()` are `X` and `Z`, numerical vectors containing the datasets, and `r.vec` a vector of values of $r$. Optional inputs are the risk chosen for the confidence intervals, and an option indicating is illustrative plots are requested. The output is a list containing, for each input value of $r$, the different estimates and results (with the notations of the paper) :

$\widehat{p}_{1,r}^{(W)}$, $\widehat{p}_{1,r}$, $\widehat{far}^{(W)}(r)$, $\widehat{far}(r)$, $\widehat{\sigma}_r^{(W)}$, $\widehat{\sigma}_r$, lower and upper confidence bounds (95% coverage) for $p_{1,r}$ and $far(r)$.

in the respective fields `$p1rhat`, `$p1rhatnp`, `$farhat`, `$farhatnp`, `$stdp1rW`, `$stdp1rnp`, `$conintp1rW`, `$conintp1rnp`, `$conintfarW`, `$conintpfarnp` (the last four fields are $nr \times 2$ matrices where $nr$ is the number of input $r$ values, while the other fields are vectors of size $nr$).

*Example*

Using the function `simulWclass()` described later in this document, we simulate two GEV datasets satisfying the $W$-class assumption, with respective shape parameters $\xi_X = -0.2$ and $\xi_Z = -0.3$, and scale parameters $\sigma_X = 1$ and $\sigma_Z = 1$ (the location parameters are automatically set by the function so that the supports are identical). The datasets sizes are $m = 150$ and $n = 30$ (counterfactual and factual). If one is interested in the values $r = 5, 10, 20, 30, 50$, here is how the function `farrecord()` can be used :

```
> data <- simulWclass(m=150,n=30,N=1,ksiX=-0.2,ksiZ=-0.25,sigX=1,sigZ=1)
> farrecord(data$matX,data$matZ,r=seq(5,50,by=5))
```

Below a second example, with positive shape parameters $\xi_X = 0.1$, $\xi_Z = 0.2$, and the code now requests the evaluation of the $p$-value of the WLK Weibullity test , for assessing whether the $W$-class assumption holds (which is the case here, by construction). The output now includes another field, `$weibtestpvalue`.

```
> data <- simulWclass(m=150,n=30,N=1,ksiX=0.1,ksiZ=0.2,sigX=1,sigZ=2)
> farrecord(data$matX,data$matZ,r=seq(5,50,by=5),weibtest=T,plot=T)
```

(the default values for the parameters `weibtest=` and `plot=` are respectively false and true).

## 2. The main functions for more flexibility and many settings

- `GZestimation(matX,matZ,...)` : this function computes all the values $\widehat{G}_m(Z_i)$ from the data, its output will be used for estimating the Weibull coefficients. These values are, after the transformation $x \mapsto -\log x$, the pseudo-Weibull variables.

  More precisely, the inputs are the matrices `matX` et `matZ` containing (say) $N$ samples of sizes $m$ and $n$, of the counterfactual $X$ and factual $Z$ variables under study (each column of these matrices typically represents a framework or a gridpoint of the area under study) :

$$\texttt{matX} = \begin{pmatrix} X_{1,1} & \cdots & X_{1,j} & \cdots & X_{1,N} \\ \vdots & & \vdots & \vdots & \\ X_{i,1} & \cdots & X_{i,j} & \cdots & X_{i,N} \\ \vdots & & \vdots & \vdots & \\ X_{m,1} & \cdots & X_{m,j} & \cdots & X_{m,N} \end{pmatrix} \text{ and } \texttt{matZ} = \begin{pmatrix} Z_{1,1} & \cdots & Z_{1,j} & \cdots & Z_{1,N} \\ \vdots & & \vdots & \vdots & \\ Z_{i,1} & \cdots & Z_{i,j} & \cdots & Z_{i,N} \\ \vdots & & \vdots & \vdots & \\ Z_{n,1} & \cdots & Z_{n,j} & \cdots & Z_{n,N} \end{pmatrix}$$

  The output is a $n \times N$ matrix `matGm` which elements are the $\widehat{G}_{m,j}(Z_{i,j})$, where $\widehat{G}_{m,j}$ is some estimator of the cdf $G$ of $X$, computed from the $j$-th sample of $X$ (the $j$-th column of `matX`) :

$$\hookrightarrow \quad \texttt{matGn} = \begin{pmatrix} \widehat{G}_{m,1}(Z_{1,1}) & \cdots & \widehat{G}_{m,j}(Z_{1,j}) & \cdots & \widehat{G}_{m,N}(Z_{1,N}) \\ \vdots & & \vdots & \vdots & \\ \widehat{G}_{m,1}(Z_{i,1}) & \cdots & \widehat{G}_{m,j}(Z_{i,j}) & \cdots & \widehat{G}_{m,N}(Z_{i,N}) \\ \vdots & & \vdots & \vdots & \\ \widehat{G}_{m,1}(Z_{n,1}) & \cdots & \widehat{G}_{m,j}(Z_{n,j}) & \cdots & \widehat{G}_{m,N}(Z_{n,N}) \end{pmatrix}$$

  By default, $\widehat{G}_m$ is the modified empirical cdf defined in equation (4) of the paper, with the tuning parameter $b = 0.05$. The function proposes to change the value of $b$, or to choose the option `methodGhatm="npudist"`, which uses instead a kernel method for estimating $G$ (from the R package `np`).

- `weibullGMMestim(matGm,truevalues=NULL)` : this function takes as input the output `matGm` of the function `GZestimation()`, and provides as output the GMM estimates $(\hat{\lambda}, \hat{k})$ of the supposedly underlying Weibull distribution, associated with each data sample, according to the method described in section 3.3 of the paper. The output of `weibullGMMestim()` will be the input of most of the other functions described below.

  More precisely, the output is a list of 2 elements : the first and second elements (the `$lambdahat` and `$khat` fields) are vectors of size $N$ containing respectively the values $(\hat{\lambda}_1, \dots, \hat{\lambda}_N)$ and $(\hat{k}_1, \dots, \hat{k}_N)$, where $(\hat{\lambda}_j, \hat{k}_j)$ is the couple of solutions of the estimating equations

$$\frac{1}{n}\sum_{i=1}^{n} \begin{pmatrix} \widehat{G}_{m,j}(Z_{i,j}) & - & p_{1,2}(\lambda, k) \\ \widehat{G}_{m,j}(Z_{i,j}) & - & p_{1,3}(\lambda, k) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

  (see the paper for the definition of functions $p_{1,2}(\lambda, k)$ and $p_{1,3}(\lambda, k)$, which are computed with the help of the function `laplaceWeibull()`). The code relies on the function `gmm()` of the `gmm` package.

  The option `truevalues=` can provide starting values for the GMM estimation process (in the form of a $2 \times N$ matrix, the first row for the starting values for $\lambda$, and the second for those for $k$).

- `p1rfarW(lam.vec,k.vec,r.vec,...)` : this function takes as input values of $\lambda$, $k$ and $r$, and computes the corresponding values of

  $$p_{1,r}(\lambda,k) = \mathbb{E}(G(Z)^{r-1}) = \mathbb{E}\left(e^{-(r-1).W}\right) \quad \text{where} \quad W \sim Weibull(\lambda,k)$$

  and $far(r,\lambda,k) = 1 - 1/(rp_{1,r}(\lambda,k))$. In practice though, the provided values of $\lambda$ and $k$ are the *estimates* of $\lambda$ and $k$ issued from function `weibullGMMestim()`, and therefore the output of `p1rfarW()` is a list of the parametric estimators $\widehat{p}_{1,r}^{(W)}$ and $\widehat{far}^{(W)}(r)$ of the underlying $p_{1,r}$ and $far(r)$ for every considered setting.

  The input vectors `lam.vec`, `k.vec` and `r.vec` must have the same size (say $N$), and the output is a list of 2 elements, the field `$p1r` containing the $N$ corresponding values of $p_{1,r}(\lambda,k)$, and the `$far` containing the associated $far(r)$ values. This function computes no associated variances, see the function `p1rfarW_var()` for this purpose.

- `p1rfarW_var(lam.vec,k.vec,r.vec,a=1,...)` : if the inputs `lam.vec`, `k.vec` are estimates of $\lambda$ and $k$ issued from function `weibullGMMestim()`, this function mainly computes the estimate $(\widehat{\sigma}_r^{(W)})^2$ of the asymptotic variance of the parametric estimator $\hat{p}_{1,r}^{(W)}$ of $p_{1,r}$ (up to the factor $1/\sqrt{n}$). It also computes that of $\widehat{far}^{(W)}(r)$. This function involves the numerical evaluation of a number of integrals detailed in the appendix of the paper.

  If the inputs `lam.vec`, `k.vec` are true values of $\lambda$ and $k$, then the outputs are the true asymptotic variances.

  Apart from the usual input vectors `lam.vec`, `k.vec`, `r.vec` (as in previous function), the value of $a = \sqrt{n/m}$ is also required (by default , it is 1, which may cause a fair misevaluation of the variances), and the optional parameter `p1rvec=` (`NULL` by default) can provide the values of $\widehat{p}_{1,r}^{(W)}$ (which have generally been already computed) to avoir them from being re-computed.

The rest of this section deals with the implementation of the *nonparametric* material of the paper (its Section 2), which is not the main contribution of the paper, and will thus be covered more briefly here.

- `p1rfar_NPestim(matGm,r.vec,a=1)` : this function performs the non-parametric statistical analysis of the data, as detailed in Section 2 of the paper. In particular, for every value $r$ in `r.vec`, it computes the non-parametric estimator $\widehat{p}_{1,r}$ of $p_{1,r}$, as well as the non-parametric estimate of its asymptotic variance (up to $1/\sqrt{n}$) $\widehat{\sigma}_r^2$, which is defined after the statement of Proposition 1 in the paper.

  Note that the asymptotic variance of $\widehat{far}(r) = 1 - 1/(r\widehat{p}_{1,r})$ is (up to $1/\sqrt{n}$) $\sigma_r^2/(rp_{1,r}^2)^2$, and can thus be estimated by $\widehat{\sigma}_r^2/(r\widehat{p}_{1,r}^2)^2$.

  The inputs are simply the output `matGm` of function `GZestimation()`, a vector `r.vec` of values of $r$, and the value $a = \sqrt{n/m}$ appearing in the formula for $\widehat{\sigma}_r^2$. The output is a list of vectors, each of the same size as `r.vec`, containing $\widehat{p}_{1,r}$ (`$p1rhat` field), $\widehat{\sigma}_r^2$ (`$sigma2rhat` field), as well as the three fields `$p12rm1hat`, `$p1rhat`, and `$Mrhat` containing the non-parametric estimates of $p_{1,2r-1}$, $\tau_r^2$ and $M_r$ (see the paper for the definition of these notations).

- `p1rfar_NPparamestim(lam.vec,k.vec,r.vec,a=1,...)` : this function has a purpose similar to the previous function, except that now the estimates of the asymptotic variances (of the nonparametric estimates of $p_{1,r}$ and $far(r)$) are of *parametric* nature, *i.e.* it uses the W-class assumption for estimating $\sigma_r^2$.

The inputs are now the vectors `lam.vec`, `k.vec` (instead of `matGm`), `r.vec`, and the value of $a = \sqrt{n/m}$. The output is also a list, which fields have slightly different names as in the previous function : see the code file directly for more information.

- `simulWclass(m,n,N,ksiX,ksiZ,sigX,supportauto=TRUE,...)` : this function produces GEV samples in a form adapted to the present framework, and by default satisfying automatically the W-class assumption. In particular, the output is a list composed of four elements : the first two (`$matX` and `$matZ`) are data matrices that can be provided as inputs of function `GZestimation()`, and the other two are the values of $\lambda$ and $k$ which correspond to the formula provided by Lemma 1 of the paper. This function is useful for testing the different functions described in this document.

  The mandatory inputs are the sample sizes `m` (of the counterfactual samples) and `n` (of the factual samples), the number `N` of desired samples (which will constitute the number of columns of the output of `GZZestimation()`), the shape parameters `ksiX` and `ksiZ` of the GEV distributions, and the scale parameter `sigX` of the counterfactual GEV distribution. By default, the function sets $\sigma_Z = \sigma_X$ and sets the location parameters of the GEV so that the W-class assumption holds (*i.e.* equal supports). And it awaits values of `ksiX` and `ksiZ` which have the same sign.

  The optional parameters are the specification of the factual scale parameter $\sigma_Z$ (`sigZ`), and possible free choice for the location parameters $\mu_X, \mu_Z$ (`muX` and `muZ`) so that the supports are not necessarily equal ; in that case, the user must set the value `F` to the option `supportauto=`.

  The output is a list which has been described a few lines above.

  The function also provides a plot of the kernel density estimates of the $X$ and $Z$ distributions (respectively in black and red), based on the first columns of each matrix `matX` and `matZ`.

*We let the interested user read the code file for further details, and description of other functions which may be of interest (for instance those computing the estimates of the different covariance and Jacobian matrices which appear in the statement of Proposition 4 in the paper).*

## 3. Example

```
> data <- simulWclass(m=150,n=30,N=100,ksiX=-0.2,ksiZ=-0.25,sigX=1,sigZ=1)
      # data generation, 100 (couples of) samples
> matGm <- GZestimation(data$matX,data$matZ)
      # much more computation time with option methodGhatm="npudist" than with the default "ecdfmodif"
> thetahat <- weibullGMMestim(matGm)
      # computes the estimates of lambda and k for each of the 100 samples
> rvalues <- seq(5,30,by=5)
> p1rfarestim <- p1rfarW(thetahat$lambdahat,thetahat$khat,10)
      # computes the parametric estimates of p1r and far(r) for r=10 for each of the 100 samples
> p1rfarestimbis <- p1rfarW(rep(thetahat$lambdahat[1],6),rep(thetahat$khat[1],6),rvalues)
      # computes the parametric estimates for r=5,10,15,20,25,30 for the first sample only
> p1rfarestim2$p1rhat
      # displays the parametric estimates of p1r for r=5,10,15,20,25,30 for the first sample only
> results <- p1rfarW_var(thetahat$lambdahat,thetahat$khat,rep(10,100),a=sqrt(30/150))
      # computes the variance estimates of the parametric estimators of p1r and far(r)
> p1rfarestim$p1r[1] * exp( c(-1,1)*qnorm(0.975)*sqrt(results$varp1r[1]/30) / p1rfarestim$p1r[1] )
      # 95% confidence bounds for p1r for r=10 and the first sample (formula (9) in the paper)
```