Import des fonctions.

```
rm(list=ls())
set.seed(133)
require(reshape2)
```

```
## Le chargement a nécessité le package : reshape2
```

```
## Warning: le package 'reshape2' a été compilé avec la version R 4.3.3
```

```
require(ggplot2)
```

```
## Le chargement a nécessité le package : ggplot2
```

```
## Warning: le package 'ggplot2' a été compilé avec la version R 4.3.3
```

```
require(gridExtra)
```

```
## Le chargement a nécessité le package : gridExtra
```

```
## Warning: le package 'gridExtra' a été compilé avec la version R 4.3.3
```

```
require(grid)
```

```
## Le chargement a nécessité le package : grid
```

```
require(dplyr)
```

```
## Le chargement a nécessité le package : dplyr
```

```
## Warning: le package 'dplyr' a été compilé avec la version R 4.3.3
```

```
##
## Attachement du package : 'dplyr'
```

```
## L'objet suivant est masqué depuis 'package:gridExtra':
##
##     combine
```

```
## Les objets suivants sont masqués depuis 'package:stats':
##
##     filter, lag
```

```
## Les objets suivants sont masqués depuis 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
require(threshr)
```

```
## Le chargement a nécessité le package : threshr
```

```
## Warning: le package 'threshr' a été compilé avec la version R 4.3.3
```

```
require(extRemes)
```

```
## Le chargement a nécessité le package : extRemes
```

```
## Warning: le package 'extRemes' a été compilé avec la version R 4.3.3
```

```
## Le chargement a nécessité le package : Lmoments
```

```
## Warning: le package 'Lmoments' a été compilé avec la version R 4.3.3
```

```
## Le chargement a nécessité le package : distillery
```

```
##
## Attachement du package : 'extRemes'
## Les objets suivants sont masqués depuis 'package:stats':
##
##     qqnorm, qqplot
require(ggside)
## Le chargement a nécessité le package : ggside
## Warning: le package 'ggside' a été compilé avec la version R 4.3.3
## Registered S3 method overwritten by 'ggside':
##   method from
##   +.gg   ggplot2
require(FactoMineR)
## Le chargement a nécessité le package : FactoMineR
## Warning: le package 'FactoMineR' a été compilé avec la version R 4.3.3
require(POT)
## Le chargement a nécessité le package : POT
##
## Attachement du package : 'POT'
## L'objet suivant est masqué depuis 'package:extRemes':
##
##     mrlplot
require(VineCopula)
## Le chargement a nécessité le package : VineCopula
## Warning: le package 'VineCopula' a été compilé avec la version R 4.3.3
require(goftest)
## Le chargement a nécessité le package : goftest
require(threshr)
```

Setup

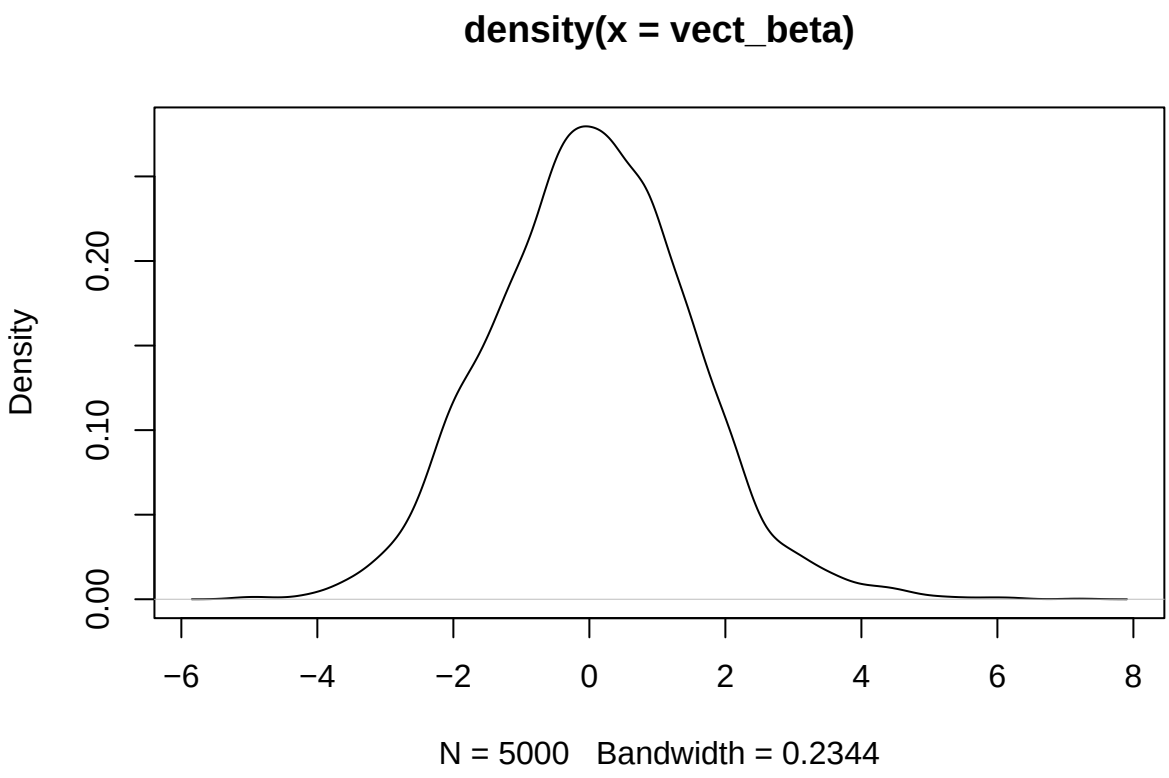
```
### Import of the functions
source("functions_toy.R")

# Parameters chosen.
######################

##### Number of generated time series.
M<-5000

##### Weight put on the distribution tail
qtile_ext<-0.95

##### Mean and standard deviation of beta
```

```r
mu<-0
sigma_N<-2**(1/2)

##### Parameters of the excesses
params_LAW<-list(mu,sigma_N)
ech_ev<-1
gam_ev<--0.10

seuil<-qnorm(p = qtile_ext,mean = mu,sd = sigma_N)
seuil
```

```
## [1] 2.326174
```

```r
Qweibull<-function(u,k,lambda){
  return((log(1/u))^(1/k)/lambda)
}
Generateur_beta<-function(u,quant,params_lois,seuil_ev,ech_ev,gam_ev){
  #return(qnorm(p = u,mean = params_lois[[1]],
  #               sd=params_lois[[2]]))
  if(u<quant){
    return(qnorm(p = u,mean = params_lois[[1]],
                 sd=params_lois[[2]]))
  }
  else{
    qprime<-(u-quant)/(1-quant)
    return(extRemes::qevd(p=qprime,
                          threshold =seuil_ev,shape = gam_ev,scale=ech_ev,type="GP"))
  }

}
```

```r
vecteur_u<-runif(M)
vect_beta<-sapply(vecteur_u,Generateur_beta,quant=qtile_ext,params_lois=params_LAW,
       gam_ev=gam_ev,ech_ev=ech_ev,seuil_ev=seuil)
plot(density(vect_beta))
```



Generate the residuals.

```
# Generation curves ----------------------------------------------------
c<-0
L_courbe<-37
begg<-1
end<-2

# change the peak position ---------------------------------------------
Time_step<-seq.int(from = begg,to = end,length.out = L_courbe)
DPHASE<-TRUE

# choose the toy dataset -----------------------------------------------
CHOICE<-"GAUSS"
```

```
Generation_curves_poly<-function(beta_,c,L_courbe,begg,end,dephase){
  B<-beta_
  vect_time<-seq.int(from = begg,to = end,length.out = L_courbe)
  Mid_<-(begg+end)/2
  sig<-(end-begg)/6
  if(dephase==TRUE){
    delta<-rnorm(n = 1,sd = sig-1/10)
    Mid_<-Mid_+delta
  }
  alpha_b<-2*B*Mid_
  return(-B*(vect_time)^2+alpha_b*vect_time+c)
```

```
}
Generation_curves_normal_LAW<-function(beta_,dephase,begg,end,L_courbe){
  delta<-0
  if(dephase==TRUE){
    delta<-(rnorm(1)/L_courbe)*6
  }
  mu<-(end+begg)/2+delta
  vect_time<-seq.int(from = begg,to = end,length.out = L_courbe)
  return(sign(beta_)*(abs(beta_)/sqrt(2*pi))*exp(-((beta_)^2/2)*(vect_time-mu)^2))
}

Generation_curves_sin<-function(beta_,dephase,begg,end,L_courbe){
  delta<-0
  if(dephase==TRUE){
    delta<-(rnorm(1,sd=2)/L_courbe)*pi
  }
  mid<-(pi/2)+delta
  seq_temps<-pi/2*(seq.int(from = begg,to = end,length.out = L_courbe))/beta_+mid
  return(sapply(X =seq_temps,FUN = function(x){return(sin(x))}))
}
```

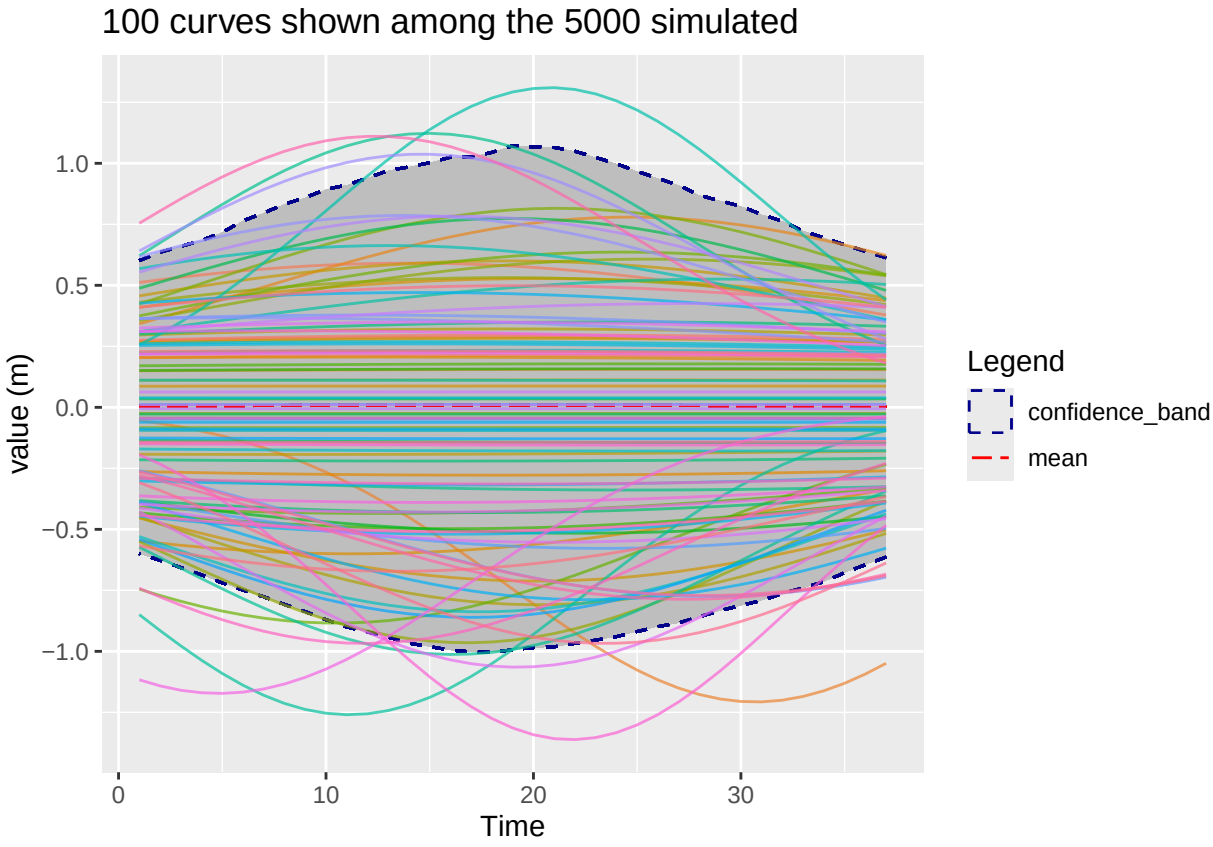
```
if(CHOICE=="POLY"){
  vect_courbes<-t(sapply(vect_beta,FUN = Generation_curves_poly,
       L_courbe=L_courbe,begg=begg,end=end,dephase=DPHASE,c=c))
}
if(CHOICE=="GAUSS"){
  vect_courbes<-t(sapply(vect_beta,Generation_curves_normal_LAW,
      L_courbe=L_courbe,begg=begg,end=end,dephase=DPHASE))
}
# vect_courbes<-t(sapply(vect_beta,Generation_curves_sin,
#        L_courbe=L_courbe,begg=-1,end=end,dephase=DPHASE))
```

Illustration

```
Nshown<-100
percent_studied<-0.05
Q1<-as.numeric(apply(X = vect_courbes,MARGIN = 2,
          FUN = function(x){return(quantile(x,percent_studied/2))}))
Q2<-as.numeric(apply(X = vect_courbes,MARGIN = 2,
          FUN = function(x){return(quantile(x,1-percent_studied/2))}))

Mean<-colMeans(vect_courbes)
inds_taken<-sample(c(1:M),size = Nshown)
Curves_seen<-vect_courbes[inds_taken,]
```

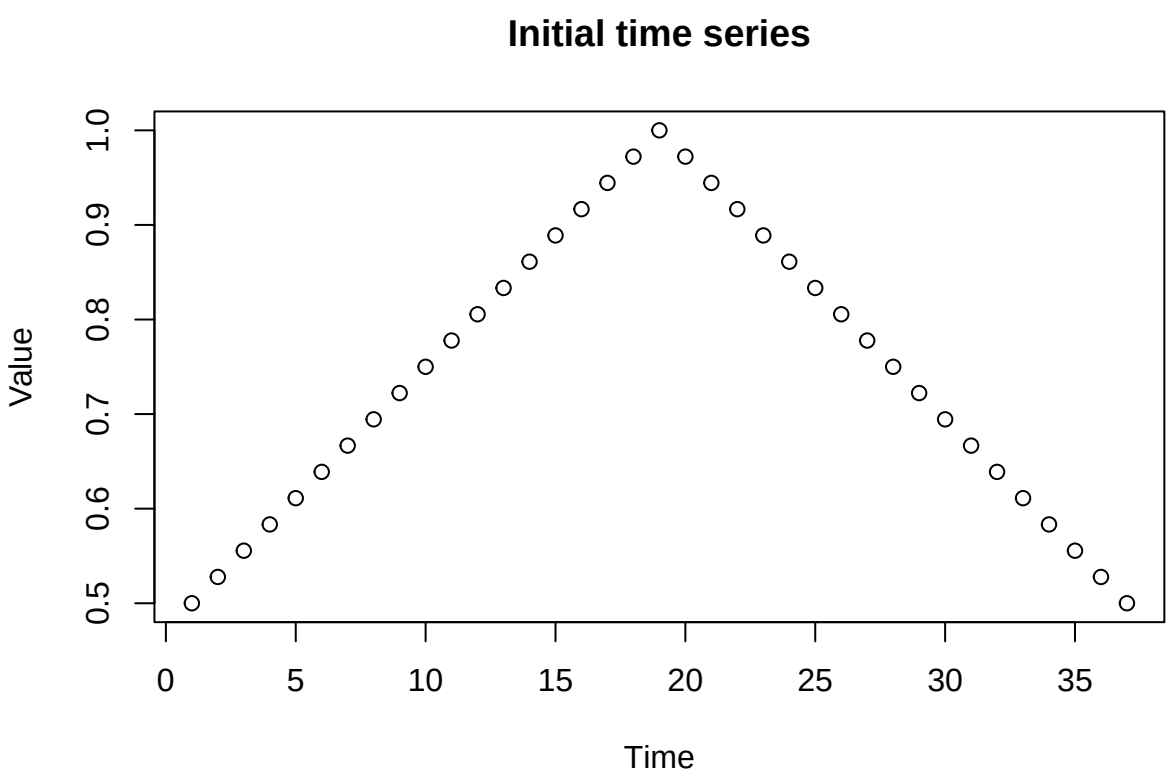
```
Melteur<-melt(t(Curves_seen))
colnames(Melteur)<-c("Time","number","value")
Melteur$number<-as.character(Melteur$number)
Melteur$Q1<-Q1[Melteur$Time]
Melteur$Q2<-Q2[Melteur$Time]
Melteur$mean_curve<-Mean[Melteur$Time]
ggplot(data=Melteur,aes(x=Time,y=value,col=number,
                        group=interaction(number),
```

```
                        ))+
  geom_ribbon(mapping=aes(ymin=Q1,ymax=Q2,linetype="confidence_band"),
              alpha=0.02,
              fill="grey",col="darkblue")+
  ylab("value (m)")+
  geom_line(aes(y=mean_curve,linetype="mean"),col="red")+
  geom_line(alpha=0.7)+
  guides(col="none")+
  scale_linetype_manual("Legend",values=c("confidence_band"=2,
                                          "mean"=5))+
  ggtitle(paste0(Nshown," curves shown among the ",M," simulated"))
```



1) Initialisation

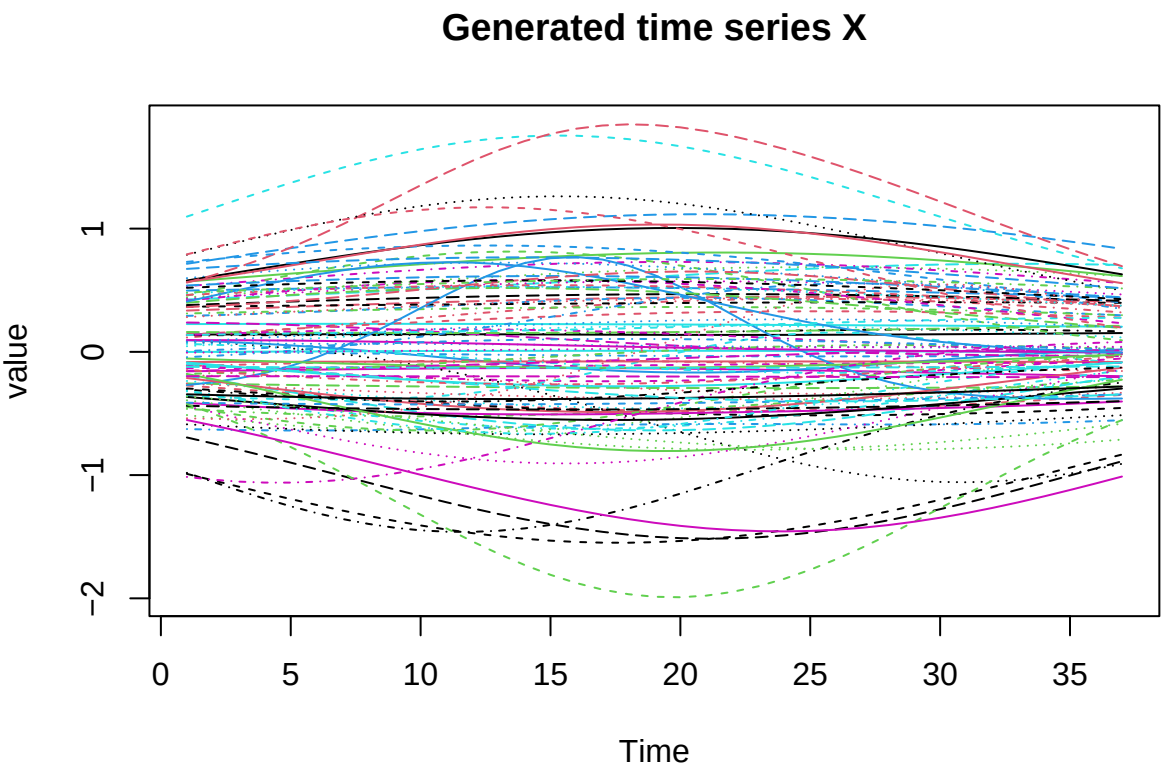
```
mid<-(end+begg)/2
vect_time<-seq.int(from = begg,to = end,length.out = L_courbe)
D<--abs(mid-vect_time)
X0<-D-min(D)+0.5
plot(c(1:37),X0,main="Initial time series",
     xlab="Time",ylab="Value")
```



AR(1) model at each time

```
ALPHA<-0.5
X_variable<-matrix(NA,nrow = M,ncol = L_courbe)
X_variable[1,]<-X0
for(j in c(2:M)){
  #remarque de Jeremy, rajouter un bruit.
  X_variable[j,]<-X_variable[j-1,]*ALPHA+vect_courbes[j,]
}
```

```
matplot(t(X_variable[inds_taken,]),type="l",xlab="Time",ylab="value",
        main="Generated time series X")
```



(I) Construction of the polar coordinates representation

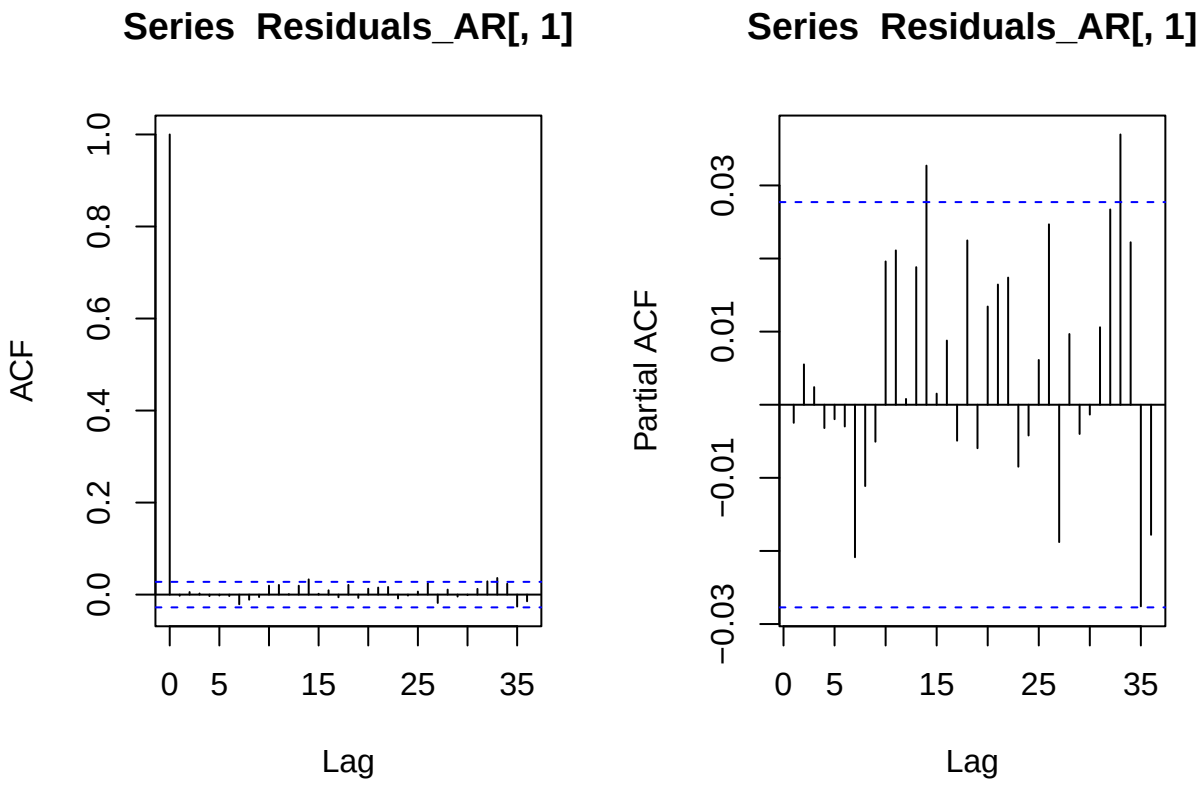
1) Applying an auto-regressive model at each time

```
# degree of the ar model.
P<-1
Residuals_AR<-matrix(NA,nrow = M,ncol = L_courbe)
Df_reg_ar<-matrix(NA,nrow=L_courbe,ncol=P+1)
for(j in c(1:L_courbe)){
  series<-X_variable[,j]
  MODEL_AR<-arima(series,order=c(P,0,0),include.mean = TRUE)
  Df_reg_ar[j,]<-unlist(MODEL_AR$coef)
  Residuals_AR[,j]<-MODEL_AR$residuals
}
```

```
P<-1
Df_reg_ar<-as.data.frame(Df_reg_ar)
colnames(Df_reg_ar)<-c("Intercept")
Name_cols_modele_lin<-c()
for(j in c(1:P)){
    Name_cols_modele_lin<-c(Name_cols_modele_lin,paste0("coeff_ar",j))
}
colnames(Df_reg_ar)<-c(Name_cols_modele_lin,"Intercept")
```

As we apply our model at each time, we can observe the residuals obtained.

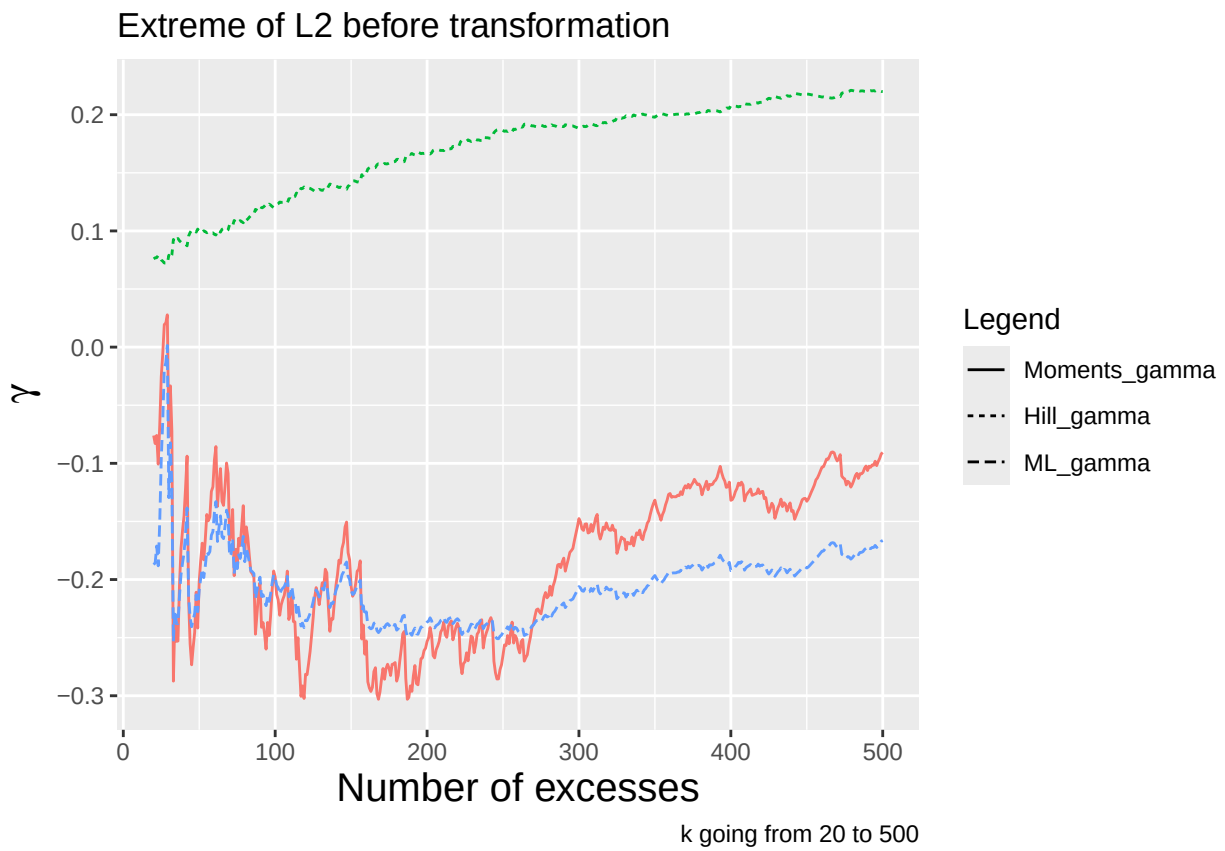
```
par(mfrow=c(1,2))
acf(Residuals_AR[,1])
pacf(Residuals_AR[,1])
```




```
par(mfrow=c(1,1))
```

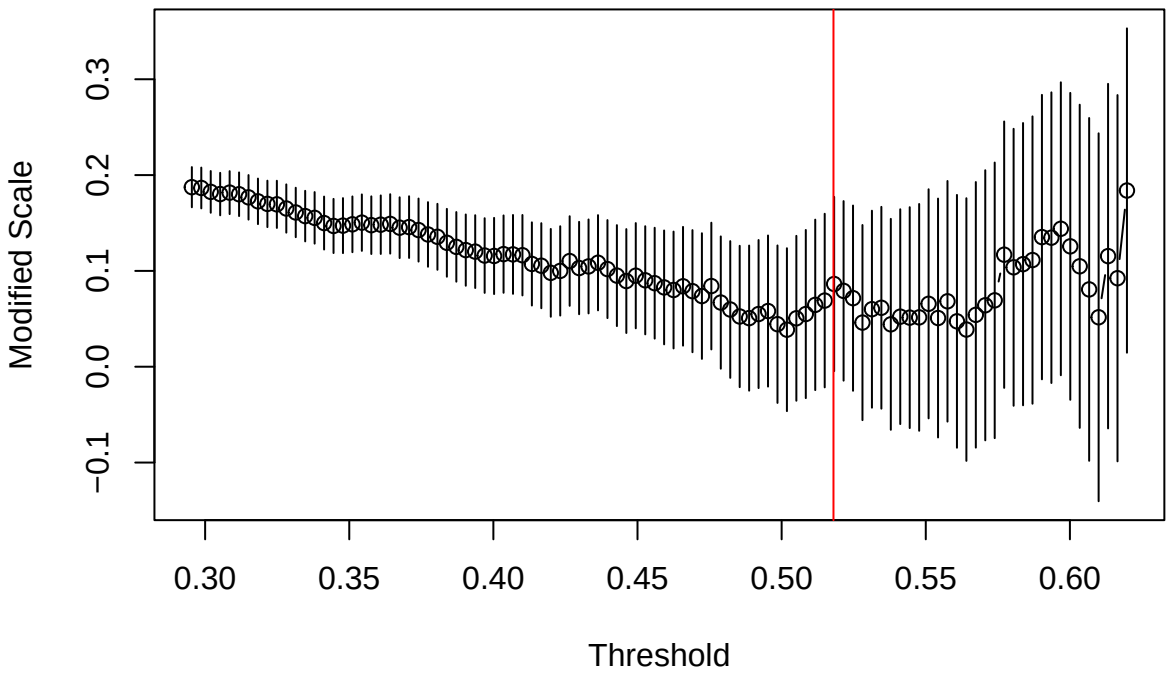
We reduce the correlation level. We can analyse the type of extremes found in our data. #### Extreme value diagnostic

```
### Type of extremes
L2<-apply(X = Residuals_AR,MARGIN = 1,FUN = calcul_norme_L2)
L<-length(L2)/10
vect_k<-c(20:L)

Graphics_estimators_gamma(series = L2,vect_k = vect_k,
                          Title_graphic  = "Extreme of L2 before transformation")
```

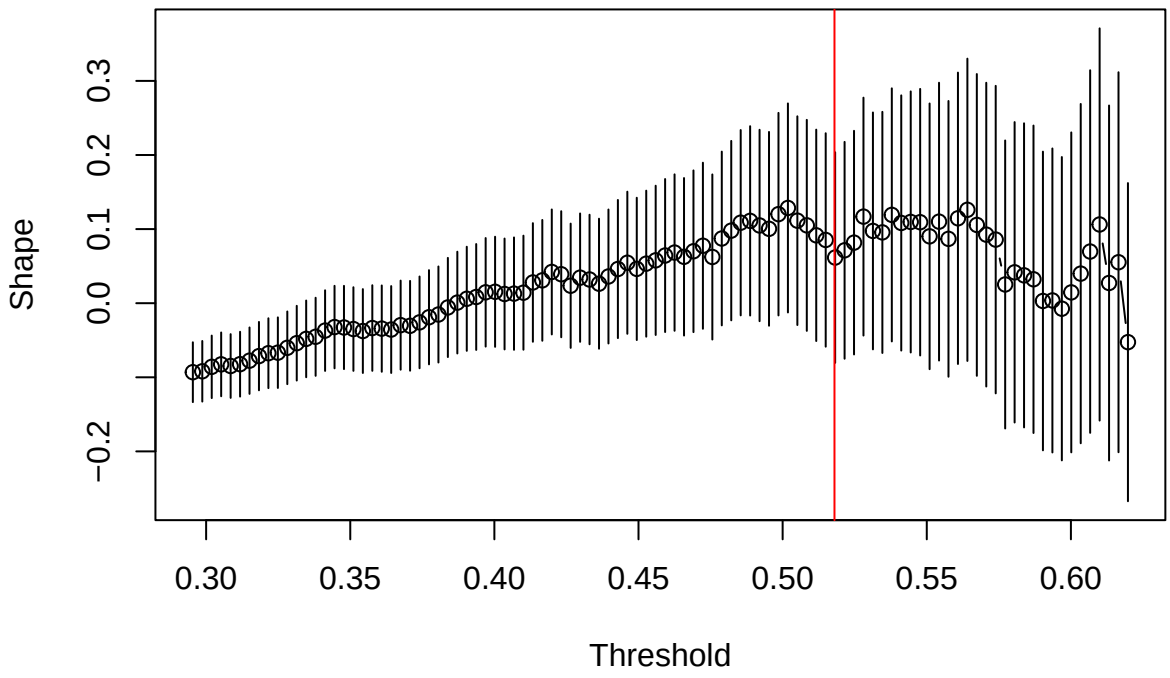


As we see that the type of extreme is not convenient, we need to apply a marginal transformation. The latter uses as parameter a threshold u_t defined at each time t. We can use several tools to identify relevant thresholds such as the mrlplot and the tcplot.

```
j<-1
Quantile_level<-0.95
Chosen_leven<-quantile(series,Quantile_level)
series<-Residuals_AR[,j]
Min_u<-quantile(series,0.75)
Max_u<-quantile(series,0.98)
NB_TH_for_graphs<-100
POT::tcplot(series,ask = FALSE,u.range = c(Min_u,Max_u),
            which = 1,nt =NB_TH_for_graphs)
abline(v=quantile(series,qtile_ext),col="red")
```

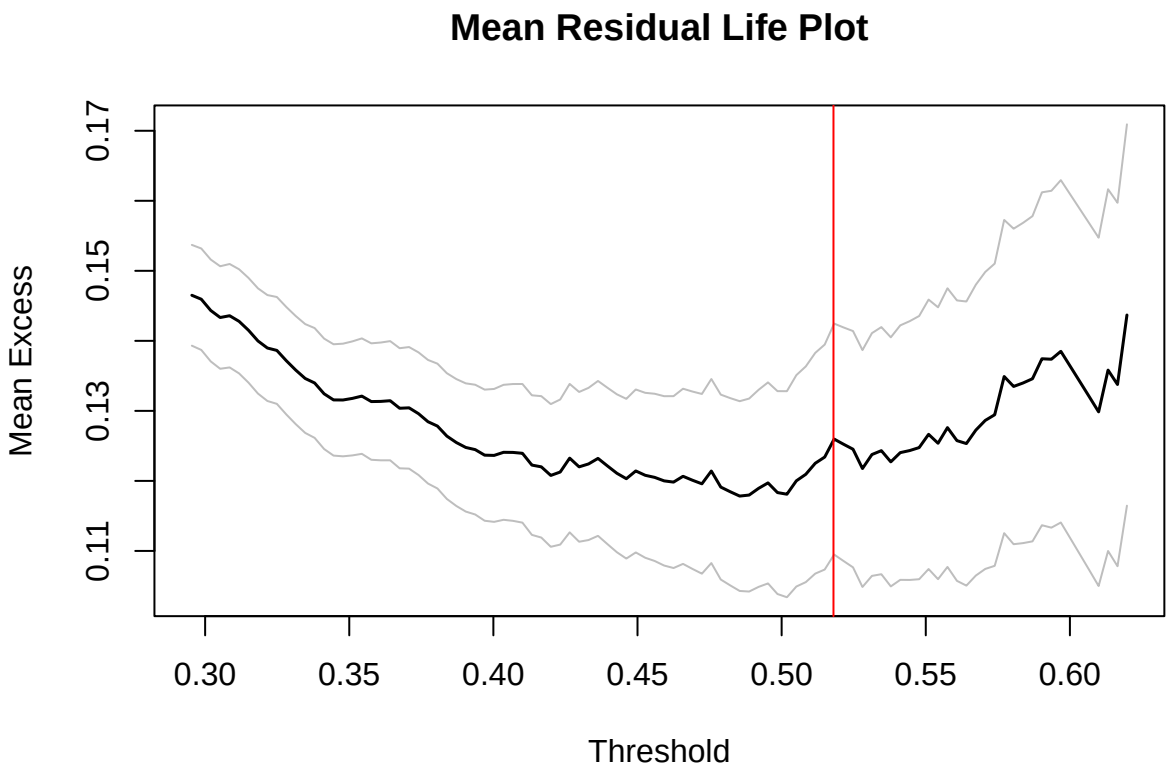



```
POT::tcplot(series,ask = FALSE,u.range = c(Min_u,Max_u),
            which = 2,nt = NB_TH_for_graphs)
abline(v=quantile(series,qtile_ext),col="red")
```



If the parameters obtained on the tcplot are relatively stable for the selected threshold, we can use it in our transformation. We can also use the mean residual life plot to go further.

```
POT::mrlplot(data = series,ask = FALSE,u.range = c(Min_u,Max_u),
             nt = NB_TH_for_graphs)
abline(v=quantile(series,qtile_ext),col="red")
```



If the function becomes linear near the threshold we selected, it means that we made a good choice. We can now use the quantile level as a parameter of our function. ### 3) Applying the marginal transformation

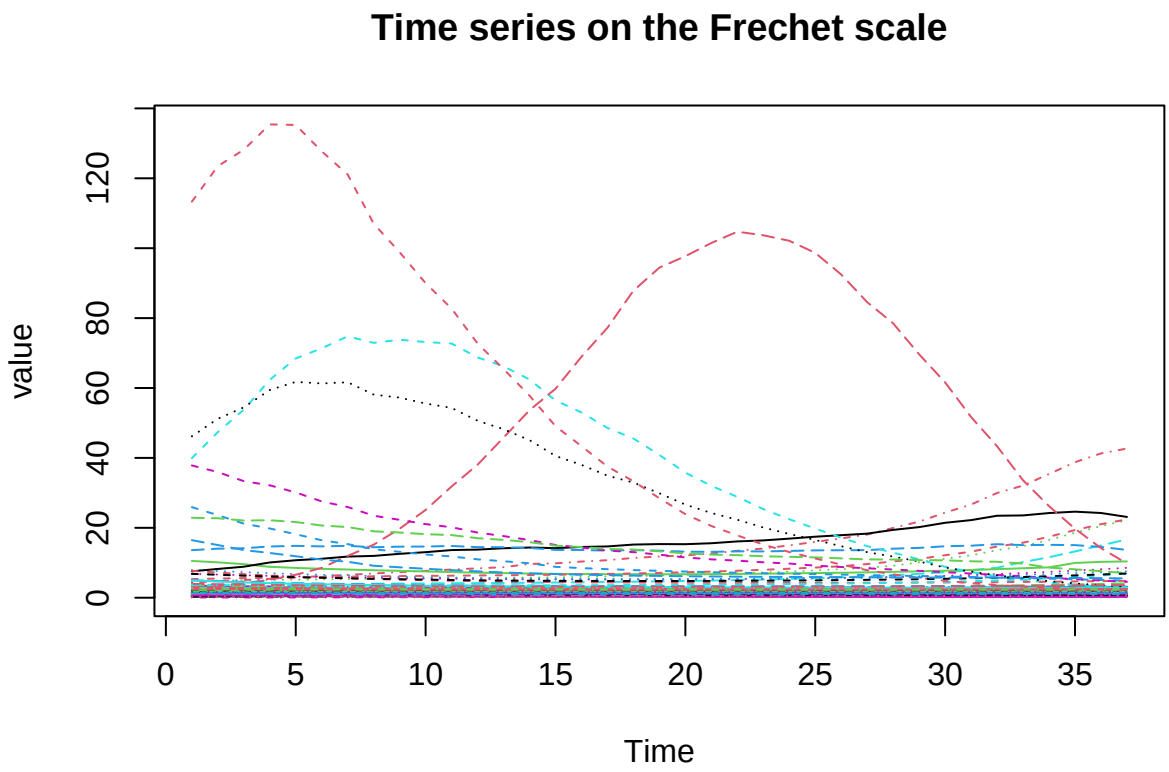
```
### List of quantile
p_U<-rep(0.10,37)

### Density parameter for the estimation of the bulk.
n.dens<-2^(14)

Vecteur_DIM<-c(1:ncol(Residuals_AR))
Margi_transformation<-as.data.frame(t(sapply(Vecteur_DIM,f_marginales_all_Pareto,
                                             data_to_tf=Residuals_AR,p_u=p_U,
                                      n.dens=n.dens)))
```

```
K_dens<-Margi_transformation$kernel_dens
Pareto<-do.call(cbind.data.frame,Margi_transformation$obs)
Unif<-1-(Pareto)^(-1)
Frechet<--1/log(Unif)
colnames(Frechet)<-c(1:ncol(Frechet))
```

```
L_EVT<-list("gamma"=Margi_transformation$gamma,
            "scale"=Margi_transformation$scale,
            "threshold"=Margi_transformation$threshold,
            "p_u"=Margi_transformation$p_u)
```

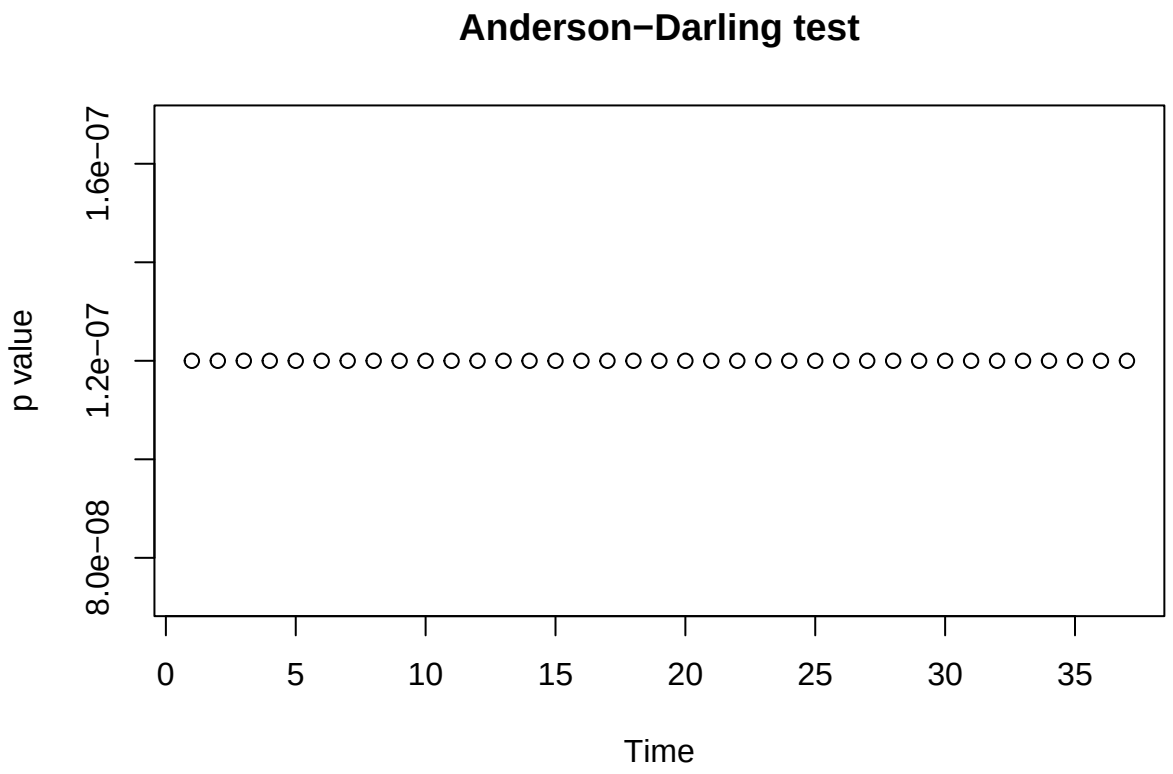
We obtain very large values as we have now Frechet marginals.

```r
### Illustration
matplot(t(Frechet[inds_taken,]),type="l",ylab="value",xlab="Time",
        main="Time series on the Frechet scale")
```



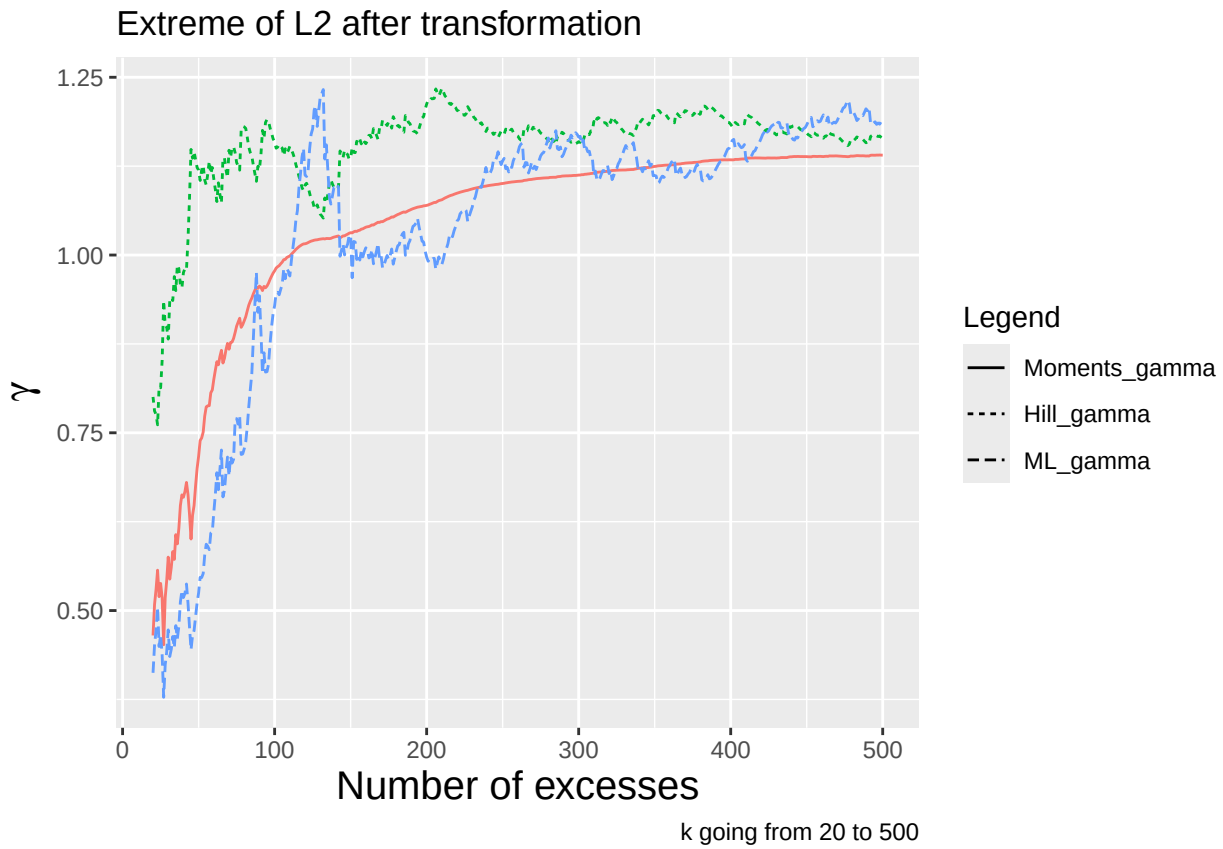
We can verify that we obtained Frechet margins with our transformations.

```r
###
test_ad<-rep(NA,ncol(Frechet))
for(t in c(1:ncol(Frechet))){
  test_ad[t]<-goftest::ad.test(x = Frechet[,t],null = extRemes::"pevd",
                               threshold=1,shape=1,scale=1,type="GEV")$p.value
}
plot(c(1:ncol(Frechet)),test_ad
     ,xlab="Time",ylab="p value",main="Anderson-Darling test")
```



We use again the L2 norm to analyse the type of extremes in our data.

```
L2<-apply(X = Frechet,MARGIN = 1,FUN = calcul_norme_L2)
Graphics_estimators_gamma(series = L2,vect_k = vect_k,
                          Title_graphic  = "Extreme of L2 after transformation")
```

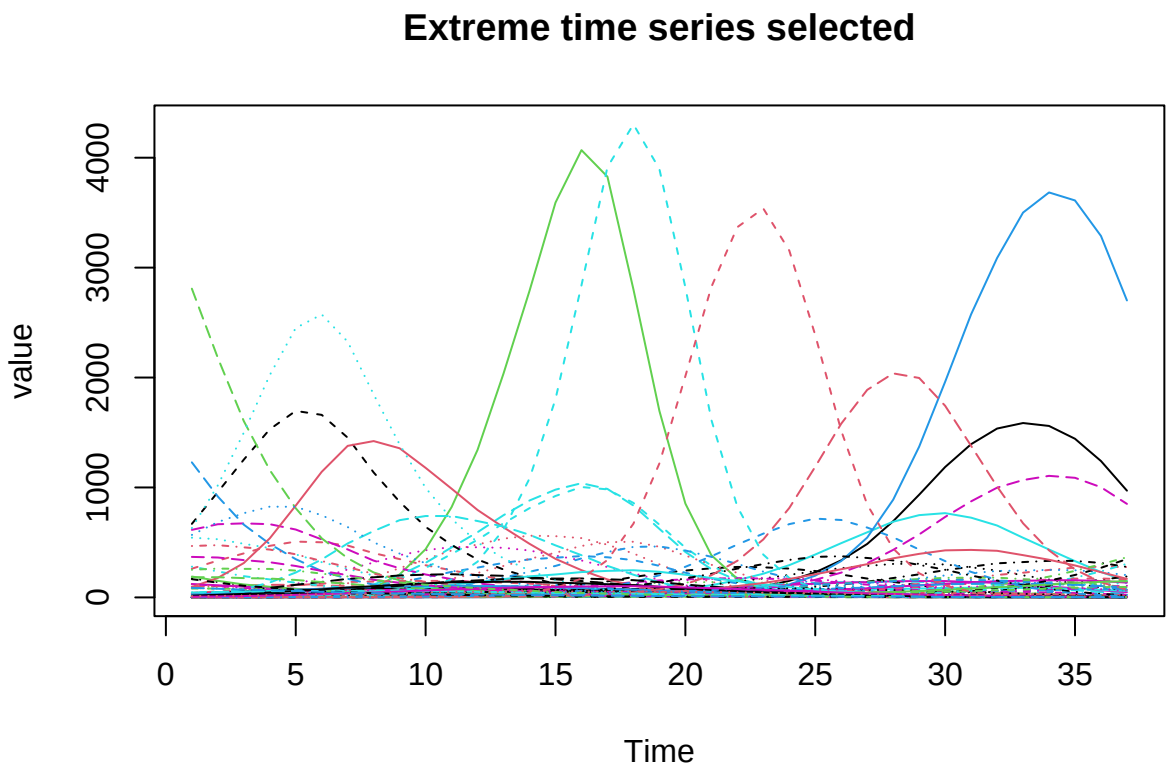


We obtain a heavy-tail variable. We can now apply the results in regular variations to simulate new extreme time series. ### 4) Selection of extreme time series

```
L2_norm<-apply(X = Frechet,MARGIN = 1,FUN = calcul_norme_L2)
Threshold_1<-quantile(L2_norm,0.95)
Index_extremes<-which(L2_norm>Threshold_1)
Exts_Pto<-Frechet[Index_extremes,]
Inds_drawn<-sample(c(1:nrow(Exts_Pto)),size = Nshown)
```

We represent the time series we obtain.

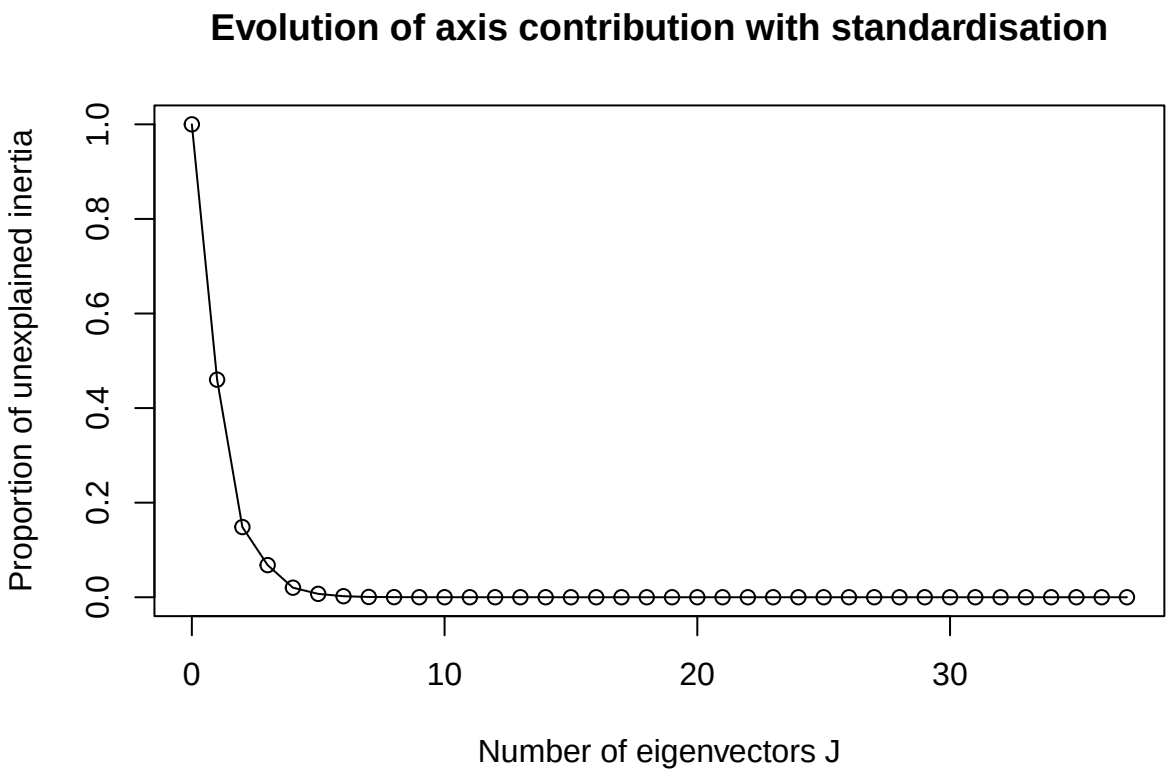
```
matplot(t(Exts_Pto[Inds_drawn,]),type="l",ylab="value",main="Extreme time series selected",xlab="Time")
```




```
Theta<-t(t(Exts_Pto)%*%diag(L2_norm[Index_extremes]^(-1)))
Poly<-log(Theta+0.1)
```

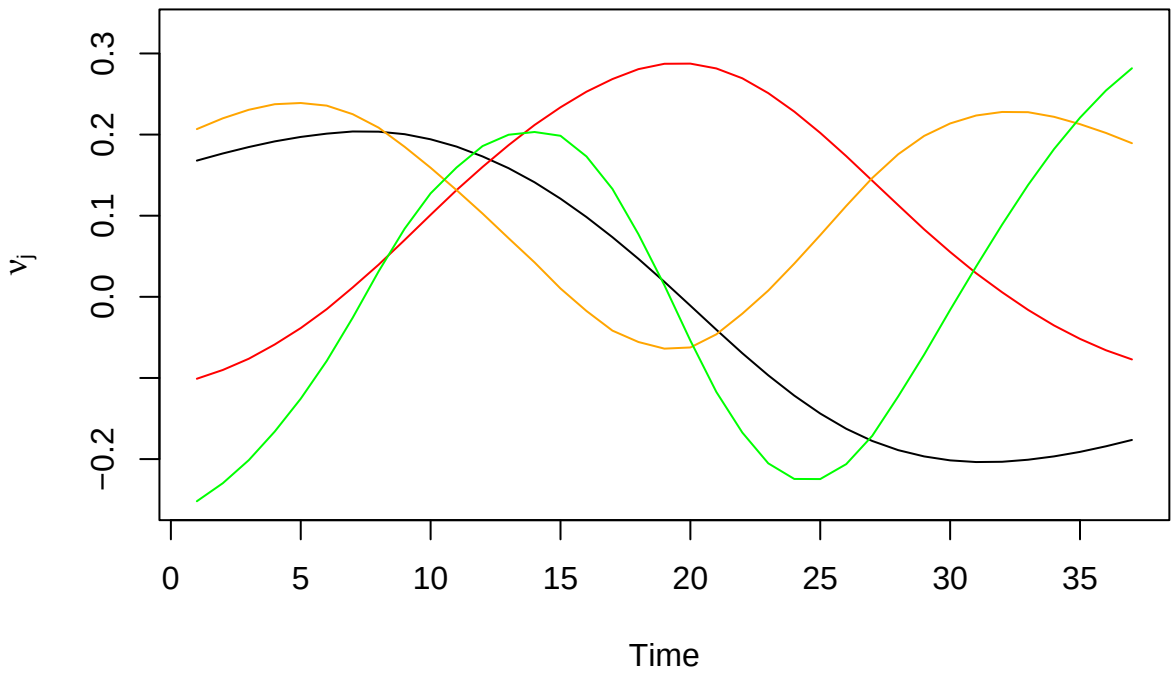
5) Approach the angle of the transformed time series

```
STD_PCA<-TRUE
### As we standardise the time series, we save its mean and its standard deviation
### at each time.
esp_time<-colMeans(Poly)
sd_time<-apply(X = Poly,MARGIN = 2,FUN = sd)

### PCA decomposition
PCA_analysis<-FactoMineR::PCA(X = Poly,scale.unit =STD_PCA ,graph = FALSE)

### Eigenvectors.
F_eigen<-PCA_analysis$svd$V
val_lambda<-PCA_analysis$eig[,1]
vect_diff<-cumsum(c(0,val_lambda))/sum(val_lambda)
vecteur_propvarexp<-1-vect_diff

### We choose the number of dimensions to keep with the proporion of unexplained inertia.
L<-length(vecteur_propvarexp)-1
plot(c(0:L),vecteur_propvarexp,
       main="Evolution of axis contribution with standardisation",
       type="o",ylab="Proportion of unexplained inertia",
       xlab="Number of eigenvectors J")
```



We keep the first J eigenvectors. Thus, we reduce the problem dimension as we only approach the law of the first two coordinates.

```
### Eigenvectors obtained.
Min_f<-min(apply(X = F_eigen,MARGIN = 1,FUN = min))
Max_f<-max(apply(X = F_eigen,MARGIN = 1,FUN = max))
plot(c(1:37),F_eigen[,1],type="l",ylim=c(Min_f,Max_f),xlab="Time",
     ylab=expression(nu[j]))
lines(c(1:37),F_eigen[,2],col="red")
lines(c(1:37),F_eigen[,3],col="orange")
lines(c(1:37),F_eigen[,4],col="green")
```



Copula theory to approach the law of coordinates

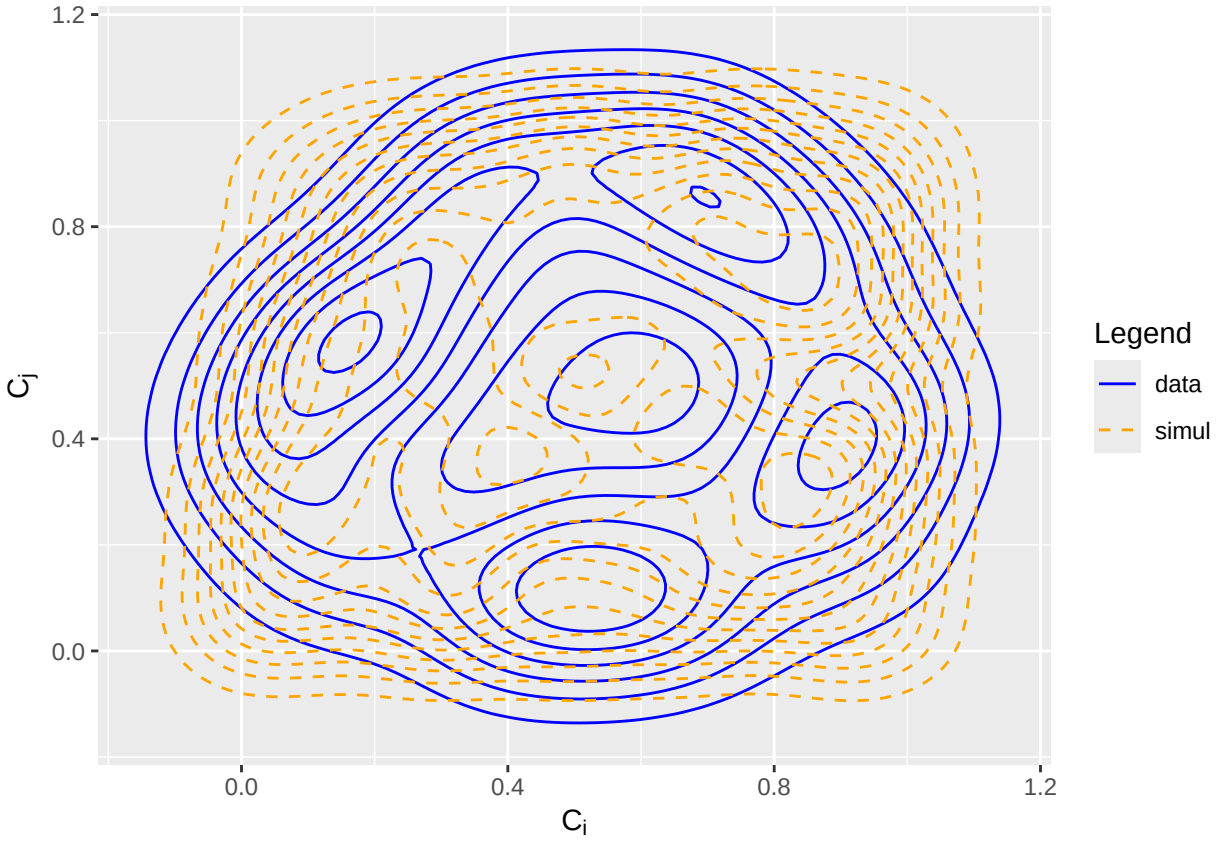
```
### Coordinates in the PCA basis
J<-4
Scores_<-PCA_analysis$ind$coord[,c(1:J)]
Matrix_C<-function_Structure_Matrice(NB_dim = J)
```

```
### Uniform margins
Unifs<-VineCopula::pobs(Scores_)
### Choose the model to use (AIC criteria)
if(J>2){
  Famille_choisie<-VineCopula::RVineCopSelect(data=Unifs,Matrix = Matrix_C)
}
if(J<=2)
{
  Famille_choisie<-BiCopSelect(Unifs[,1],Unifs[,2])
}
```

We can plot the isodensity curves we obtain for each couple of coordinates in our simulations and in our data.

```
N_for_test<-1000
if(J>2){
  i<-1
  j<-3
  Family<-Famille_choisie$family[i,j]
  PAR1<-Famille_choisie$par[i,j]
  PAR2<-Famille_choisie$par2[i,j]
  Simul_for_test<-VineCopula::RVineSim(N=N_for_test,
```

```
                                          RVM = Famille_choisie)[,c(i,j)]
}
if(J==2)
{
  i<-1
  j<-2
  Simul_for_test<-BiCopSim(N=N_for_test,family =Famille_choisie$family,par = Famille_choisie$par,par2 =
}

MIN_y<-min(apply(Simul_for_test,MARGIN = 1,FUN = min),
           apply(Unifs,MARGIN = 1,FUN = min))
MAX_y<-max(apply(Simul_for_test,MARGIN = 1,FUN = max),
           apply(Unifs,MARGIN = 1,FUN = max))

colnames(Simul_for_test)<-c("first","second")
data<-cbind.data.frame(Unifs[,i],Unifs[,j])
colnames(data)<-c("first","second")

ggplot(data=data,aes(x = first, y =second,col="data")) +
  xlim(c(-0.15,1.15))+
  ylim(c(-0.15,1.15))+
  geom_density_2d()+
  geom_density2d(data=Simul_for_test,aes(x=first,                          y=second,col="simul"),
                 linetype=2)+
  scale_color_manual(values=c("data"="blue",
                              "simul"="orange"))+
  labs(col="Legend")+
  xlab(expression("C"[i]))+
  ylab(expression("C"[j]))
```



As we are able to approach the law of angle with the PCA, we now use the result in regular variations to simulate extreme time series with heavy-tail margins. ### (II) Simulation of extreme time series

(1) Simulate new angles

```
#### the chosen number of simulated extreme time series.
Nsimul<-1000

### We use here Frechet margins so we will check that we are above 1 at each time.
Function_margin_check<-function(series){
  resultat_test<-all(series>0)
  return(resultat_test)
}
Nfound<-0
New_obs_P_acp<-matrix(NA,nrow = Nsimul,ncol = ncol(Theta))
while (Nfound<Nsimul){
  Nb<-Nsimul-Nfound
  if(J<=2){
  Realisations_copules<-BiCopSim(N=Nb,family =Famille_choisie$family,par = Famille_choisie$par,par2 =Fa
  }
  if(J>2){
    Realisations_copules<-VineCopula::RVineSim(N=Nb,RVM = Famille_choisie)
  }
  Matrice_coords_ALL<-matrix(NA,ncol = J,nrow = Nb)
  for(l in c(1:J)){
    if(Nb!=1){
```

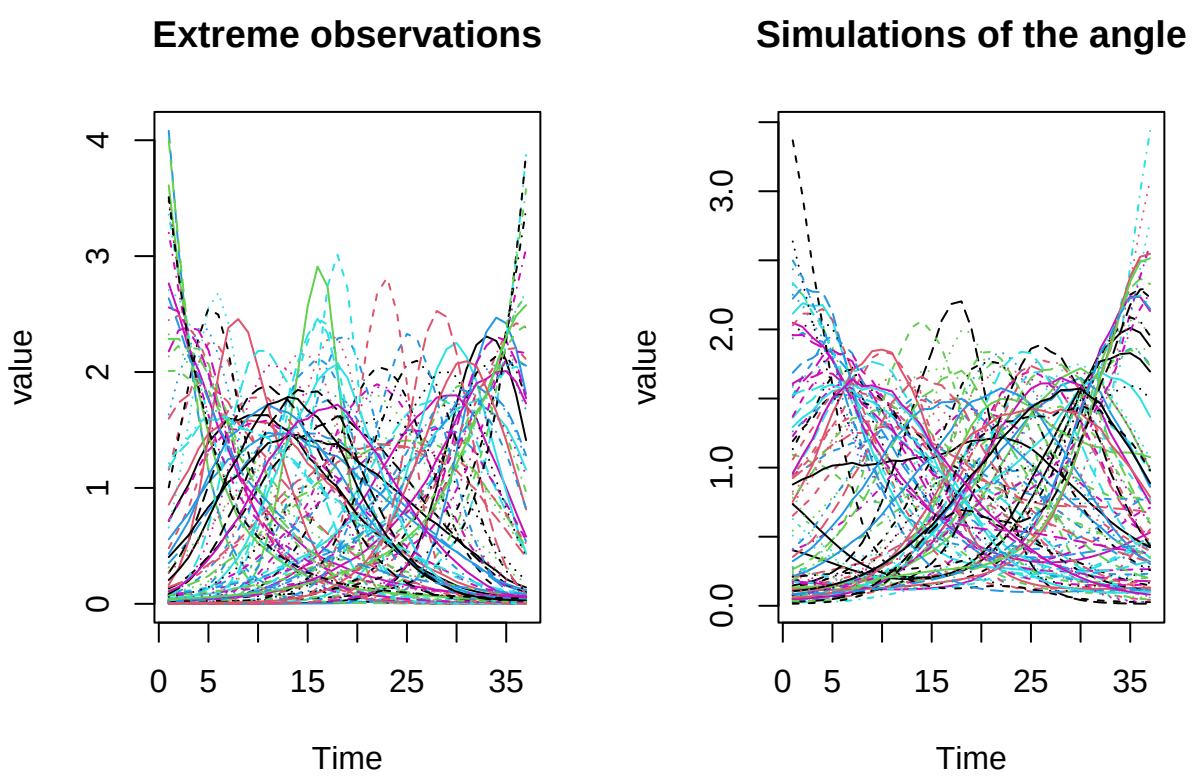
```
      Matrice_coords_ALL[,l]<-quantile(Scores_[,l],probs=Realisations_copules[,l])
    }
    else{
      Matrice_coords_ALL[,l]<-quantile(Scores_[,l],probs=Realisations_copules[l])
    }

  }
  ### Simulate new angles with the PCA decomposition
  Shape_PCA<-exp(function_reconstitution_trajectory_std(Vector_coords=Matrice_coords_ALL,
                                                        mu_t = esp_time,
                                                        sd_t = sd_time,BOOL_STD=STD_PCA))

  ### Impose the condition on the L2 norm of the angle
  L2_simul<-apply(X =Shape_PCA,MARGIN = 1,FUN = calcul_norme_L2)
  Shape_PCA<-t(t(Shape_PCA)%*%diag(L2_simul^(-1),ncol = Nb))

  ### Simulate a new radius with a Pareto law.
  valeurs_unif<-runif(n = Nb)
  reals_P_std<-sapply(valeurs_unif,Generation_Pareto_std)
  Scale_Pareto<-reals_P_std*as.numeric(Threshold_l)
  New_P_candidates<-t(t(Shape_PCA)%*%diag(Scale_Pareto,ncol = Nb))
  Inds_Pareto_verified<-apply(X = New_P_candidates,MARGIN = 1,FUN =Function_margin_check)

  ### We check that we are above 1.
  Selected<-which(Inds_Pareto_verified==TRUE)
  ### Apply the selection
  if(length(Selected)>0){
    Sub<-New_P_candidates[Selected,]
    if(is.null(nrow(Sub))){
      Sub<-matrix(Sub,ncol=ncol(New_obs_P_acp),nrow=1)
    }
    ### fill the empty dataset with the convenient time series.
    beg_indexes<-Nfound+1
    end<-Nfound+length(Selected)
    New_obs_P_acp[c(beg_indexes:end),]<-Sub
    Nfound<-end
  }
}
```

2) Applying the reverse marginal transformation

```
INDICES_ACP<-1:nrow(New_obs_P_acp)
### Pareto margins.
New_obs_P_acp<-(1-exp(-New_obs_P_acp^(-1)))^(-1)
NO_ACP<-lapply(INDICES_ACP,FUN = fnct_select_colonne,df=New_obs_P_acp)
Variables_reconversion_PCA<-lapply(NO_ACP,function_reconversion_Pareto,K=K_dens,list_evt=L_EVT)
Variables_reconversion_PCA<-t(cbind.data.frame(Variables_reconversion_PCA))
```

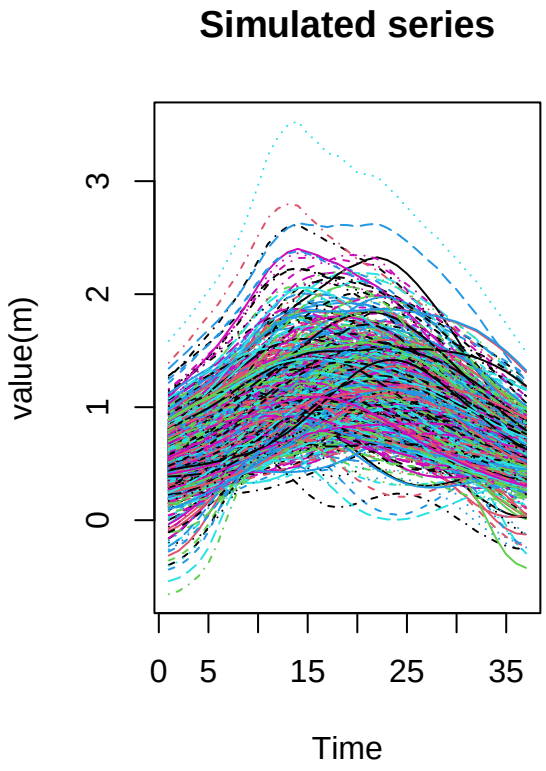
```
### Observe the Angle we obtain.
Shape_simulated<-t(t(New_obs_P_acp)%*%diag(apply(X=New_obs_P_acp,
                                                 MARGIN = 1,FUN = calcul_norme_L2)^(-1)))
par(mfrow=c(1,2))
matplot(t(Theta[Inds_drawn,]),type="l",ylab="value",
```

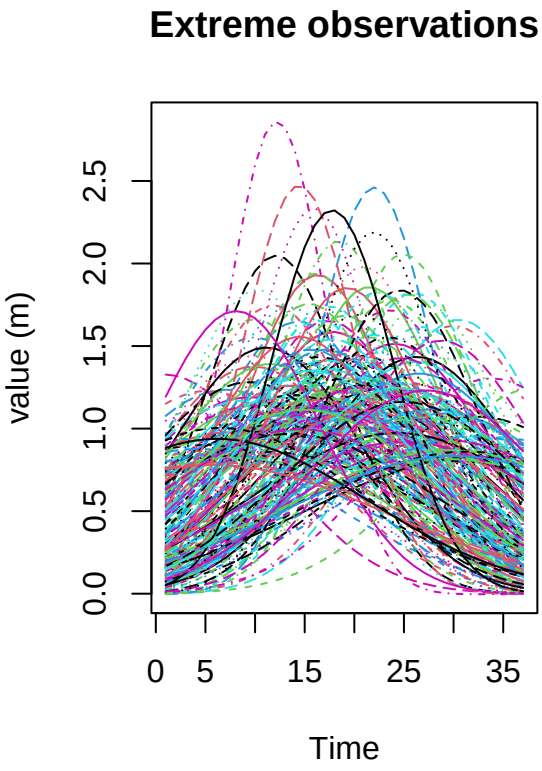
```
        main="Extreme observations",xlab="Time")
matplot(t(Shape_simulated[Inds_drawn,]),
        type="l",ylab="value",main="Simulations of the angle",
        xlab="Time")
```




```
par(mfrow=c(1,1))
```

```
par(mfrow=c(1,2))

matplot(t(Variables_reconversion_PCA),type="l",ylab="value(m)",
        main="Simulated series",xlab="Time")
matplot(t(Residuals_AR[Index_extremes,]),type="l",ylab="value (m)",
        main="Extreme observations",xlab="Time")
```






```
par(mfrow=c(1,1))
```

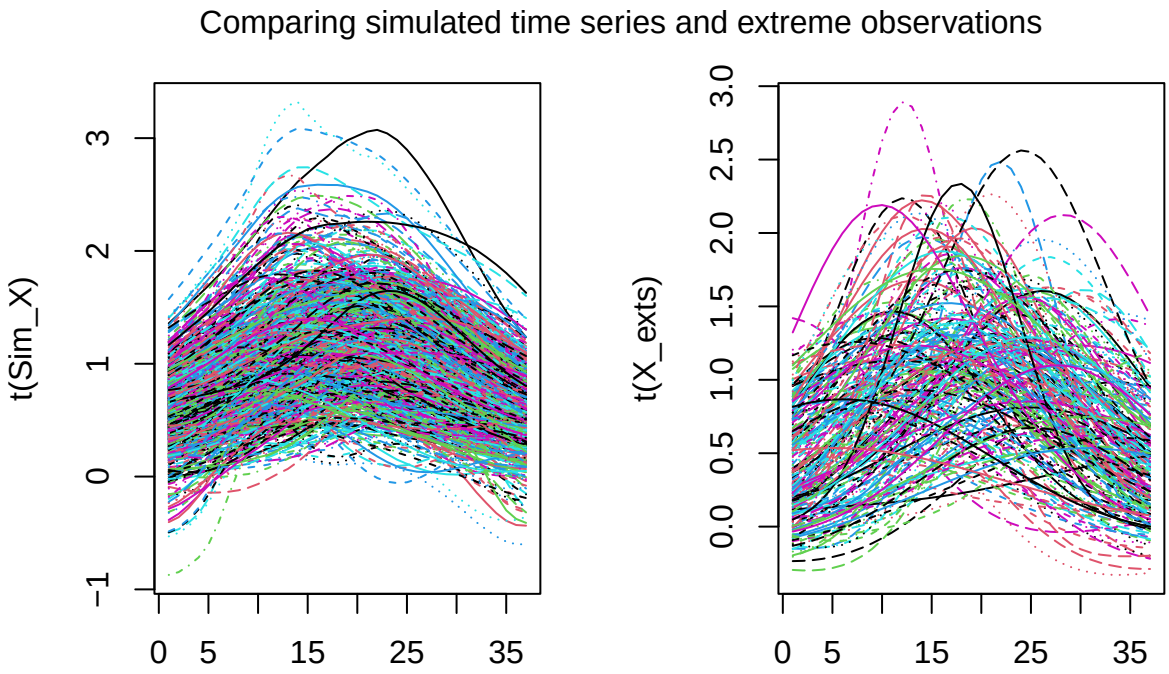
As we simulate new extreme residuals, we need to choose an initialisation. ### (3) Choice of the previous time series

```
### Conditional sampling
SW<-20
l_X_eve<-apply(X_variable,MARGIN = 1,FUN = calcul_norme_L2)
L<-length(l_X_eve)-1
l_Epsilon_data<-apply(Residuals_AR,MARGIN = 1,
                      FUN = calcul_norme_L2)[2:length(l_X_eve)]
l_Epsilon_simul<-apply(X = Variables_reconversion_PCA,MARGIN = 1,
                       FUN = calcul_norme_L2)
Indexes<-as.numeric(sapply(l_Epsilon_simul ,FUN = Sample_window,x_window = l_Epsilon_data,y = l_X_eve[1

### Choice done !
```

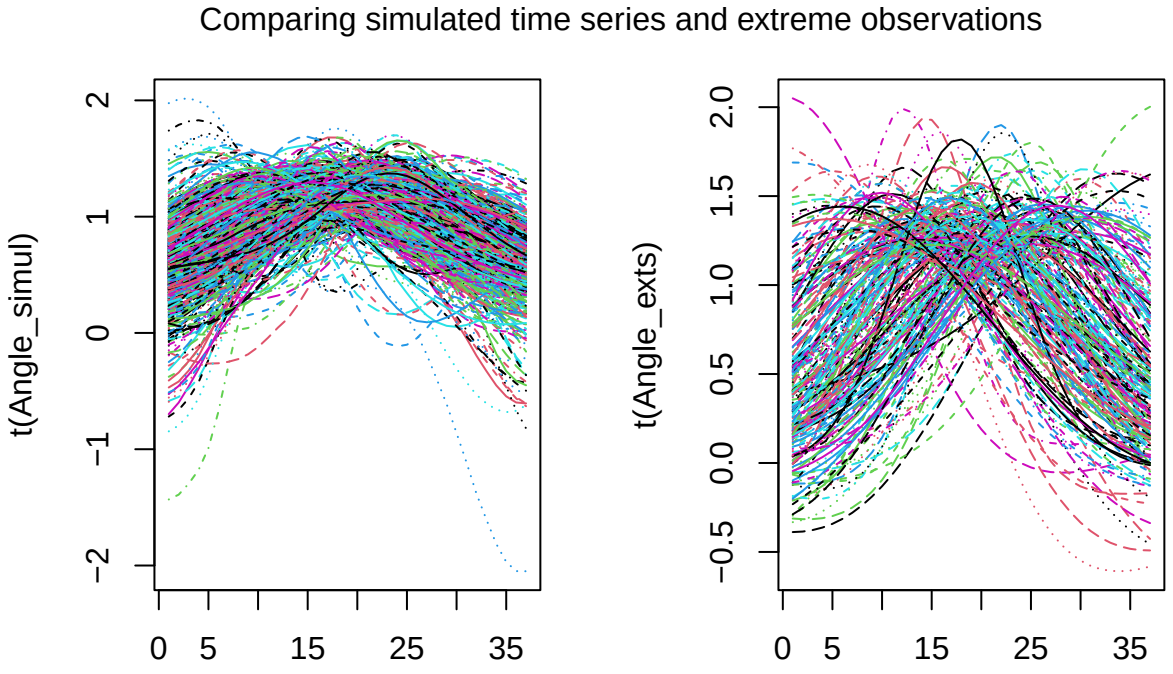
```
obs_taken<-X_variable[Indexes,]
l_taken<-apply(X = obs_taken,MARGIN = 1,
               FUN = calcul_norme_L2)
### Use the AR formula at each time.
Sim_X<-matrix(NA,nrow =nrow(Variables_reconversion_PCA),
              ncol=ncol(Variables_reconversion_PCA))
for(j in c(1:nrow(Variables_reconversion_PCA))){
  obs_EVE<-obs_taken[j,]
  Prediction<-as.numeric(sapply(X = c(1:ncol(X_variable)),FUN = Function_AR_p, obs_eve=obs_EVE,epsilon_
  Sim_X[j,]<-Prediction
```

```
}
```

```
X_exts<-X_variable[Index_extremes,]
par(mfrow=c(1,2))
matplot(t(Sim_X),type="l")
matplot(t(X_exts),type="l")
mtext("Comparing simulated time series and extreme observations",
      outer=TRUE,line=-3)
```




```
par(mfrow=c(1,2))
L2_simul<-apply(Sim_X,FUN = calcul_norme_L2,
                                        MARGIN=1)
L2_exts<-apply( X=X_exts,MARGIN = 1,FUN = calcul_norme_L2)
Inds_obs<-which(L2_exts>0)
Inds_sim<-which(L2_simul>0)
Angle_simul<-t(t(Sim_X[Inds_sim,])%*%diag(L2_simul[Inds_sim]^(-1)))
Angle_exts<-t(t(X_exts[Inds_obs,])%*%diag(L2_exts[Inds_obs]^(-1)))

matplot(t(Angle_simul),type="l")
matplot(t(Angle_exts),type="l")
mtext("Comparing simulated time series and extreme observations",
      outer=TRUE,line=-3)
```



(III) Consistency of our simulations.

##(1) Quantiles value

```
N_rep<-500
B<-nrow(X_exts)
N_rep<-500
lq<-c(0.05,0.50,0.95,0.975)
Tend<-replicate(n =N_rep,expr = Resamples_tendencies(B = B,extremes_indus= X_exts,list_Q =lq))
```

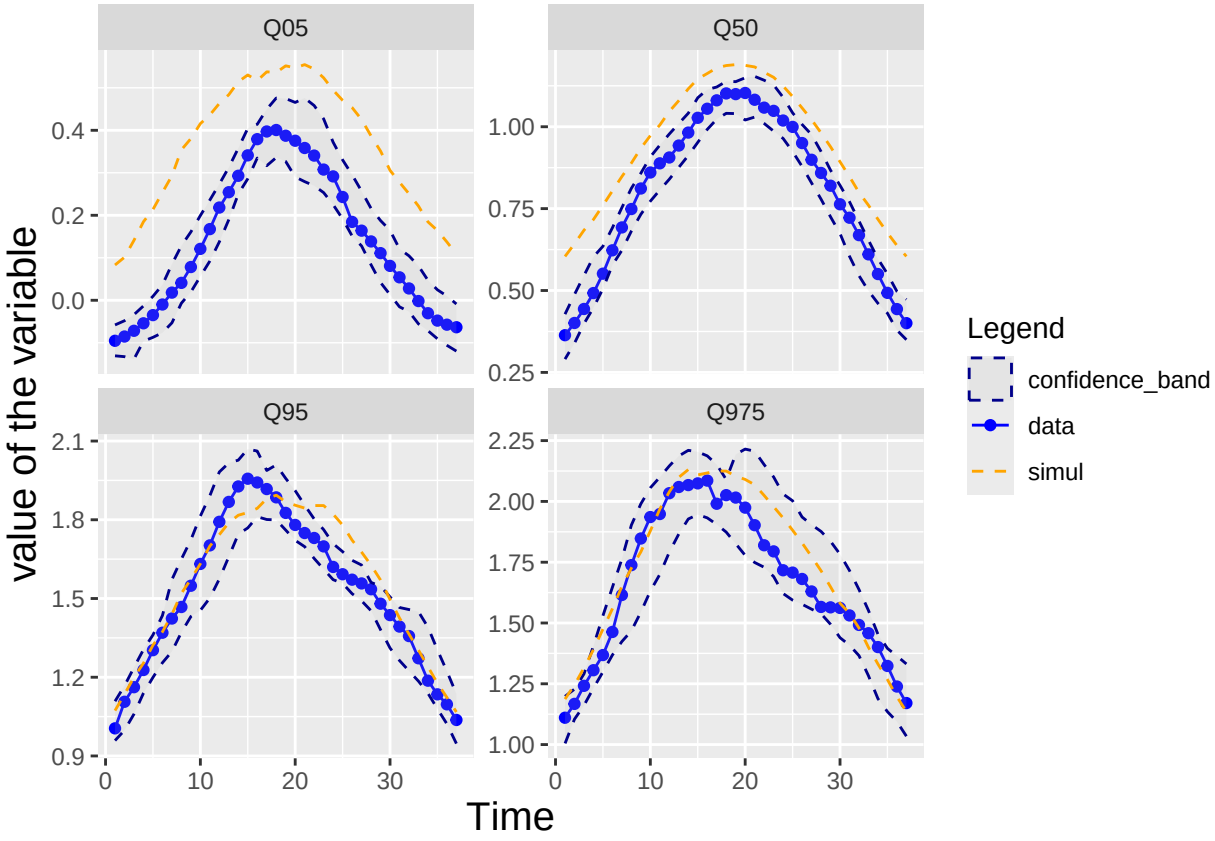
```
estimator_95_simul<-apply(Sim_X,MARGIN =2,
                          function(x){return(as.numeric(quantile(x,0.95)))})
estimator_05_simul<-apply(Sim_X,MARGIN =2,
                          function(x){return(as.numeric(quantile(x,0.05)))})
estimator_50_simul<-apply(Sim_X,MARGIN =2,
                          function(x){return(as.numeric(quantile(x,0.50)))})
estimator_975_simul<-apply(Sim_X,MARGIN =2,
                           function(x){return(as.numeric(quantile(x,0.975)))})


### data estimator
estimator_05<-apply(X_exts,MARGIN =2,
                    function(x){return(as.numeric(quantile(x,0.05)))})
estimator_95<-apply(X_exts,MARGIN =2,
                    function(x){return(as.numeric(quantile(x,0.95)))})
estimator_975<-apply(X_exts,MARGIN =2,
                     function(x){return(as.numeric(quantile(x,0.975)))})
```

```
estimator_05<-apply(X_exts,MARGIN =2,
                    function(x){return(as.numeric(quantile(x,0.05)))})
estimator_50<-apply(X_exts,MARGIN =2,
                    function(x){return(as.numeric(quantile(x,0.50)))})
```

```
bound_minus05<-apply(Tend[1,,],MARGIN = 1,
                     FUN = function(x){return(quantile(x,0.025))})
bound_plus05<-apply(Tend[1,,],MARGIN = 1,
                    FUN = function(x){return(quantile(x,0.975))})

##########
bound_minus50<-apply(Tend[2,,],MARGIN = 1,
                     FUN = function(x){return(quantile(x,0.025))})
bound_plus50<-apply(Tend[2,,],MARGIN = 1,
                    FUN = function(x){return(quantile(x,0.975))})

########
bound_minus95<-apply(Tend[3,,],MARGIN = 1,
                     FUN = function(x){return(quantile(x,0.025))})
bound_plus95<-apply(Tend[3,,],MARGIN = 1,
                    FUN = function(x){return(quantile(x,0.975))})

######
bound_minus975<-apply(Tend[4,,],MARGIN = 1,
                      FUN = function(x){return(quantile(x,0.025))})
bound_plus975<-apply(Tend[4,,],MARGIN = 1,
                     FUN = function(x){return(quantile(x,0.975))})
```

```
df<-cbind.data.frame(c(estimator_05,estimator_50,estimator_95,estimator_975),
                       c(bound_minus05,bound_minus50,
                         bound_minus95,bound_minus975),
                       c(bound_plus05,bound_plus50,
                         bound_plus95,bound_plus975))
colnames(df)<-c("estimator","bound_minus","bound_plus")

cols_<-c("data"="blue","simul"="orange","confidence_band"="darkblue")
df$estimator_simul<-c(estimator_05_simul,estimator_50_simul
                        ,estimator_95_simul,estimator_975_simul)
df$Time<-rep(c(1:ncol(X_exts)),4)
df$percent<-c(rep("Q05",ncol(X_exts)),
                rep("Q50",ncol(X_exts)),
                rep("Q95",ncol(X_exts)),
                rep("Q975",ncol(X_exts)))
GG_percent<-ggplot(data=df,aes(x=Time,y=estimator,group=interaction(percent),col="data"))+
    facet_wrap(~percent,scales="free_y")+
    geom_line()+
    geom_point()+
    geom_ribbon(mapping = aes(ymin=bound_minus,ymax=bound_plus,col="confidence_band"),alpha=0.15,
                fill="grey", linetype = "dashed")+
    geom_line(aes(x=Time,y=estimator_simul,col="simul"),linetype=2)+
    theme(axis.title=element_text(size=15))+
    ylab("value of the variable")+
    scale_color_manual(values=cols_)+
    xlab("Time")+
```

```
    labs(col="Legend")
GG_percent
```

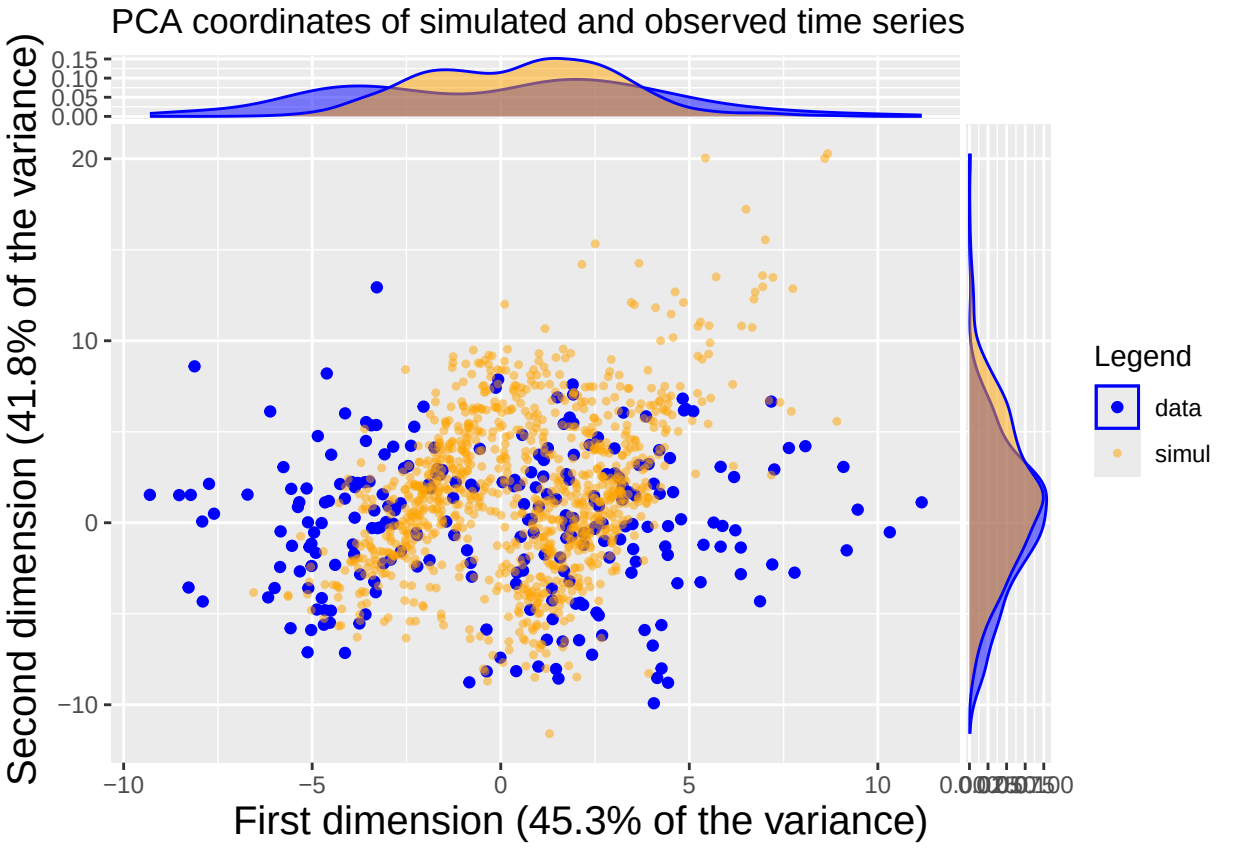


###(2) Functional decomposition of simulated and recorded time series

```
PCA_plan<-FactoMineR::PCA(X = X_exts,scale.unit = TRUE,graph=FALSE)
Mu<-colMeans(X_exts)
SD<-apply(X_exts,MARGIN = 2,FUN = sd)
F_eigen<-PCA_plan$svd$V[,c(1:2)]
Coordinates_obs<-PCA_plan$ind$coord[,c(1:2)]
percent_variance<-PCA_plan$eig[c(1:J),2]
### Standardise the simulations
STD_Simul<-scale(Sim_X,center = Mu,scale = SD)
Coordinates_simul<-STD_Simul%*%(F_eigen)
```

```
colnames(Coordinates_simul)<-c("Score_1","Score_2")
colnames(Coordinates_obs)<-c("Score_1","Score_2")
df_combo<-rbind.data.frame(Coordinates_obs,Coordinates_simul)
df_combo$type<-c(rep("data",nrow(Coordinates_obs)),rep("simul",nrow(Coordinates_simul)))
GG0<-ggplot(data=Coordinates_obs,aes(x=Score_1,y=Score_2,col="data"))+
  geom_point()+
  geom_point(data=Coordinates_simul,aes(x=Score_1,y=Score_2,col="simul"),
             alpha=0.5,
             pch=20)+
  geom_xsidedensity(data=df_combo,aes(fill=type), alpha = 0.5)+
  geom_ysidedensity(data=df_combo,aes(fill=type), alpha = 0.5)+
  ggtitle("PCA coordinates of simulated and observed time series")+
```

```
  xlab(paste0("First dimension (",round(percent_variance[1],1),"% of the variance)"))+
  ylab(paste0("Second dimension (",round(percent_variance[2],1),"% of the variance)"))+
  labs(col="Legend")+
  scale_color_manual(values=cols_)+
  scale_fill_manual(values=cols_)+
  guides(fill="none")+
  theme(axis.title=element_text(size=15))
GGO
```



We can now see if some algorithms are able to distinguish our simulations from extreme observations. We use several entries such as the time series, its norm and its pattern represented by its angle.

(3) Correlation of extreme values

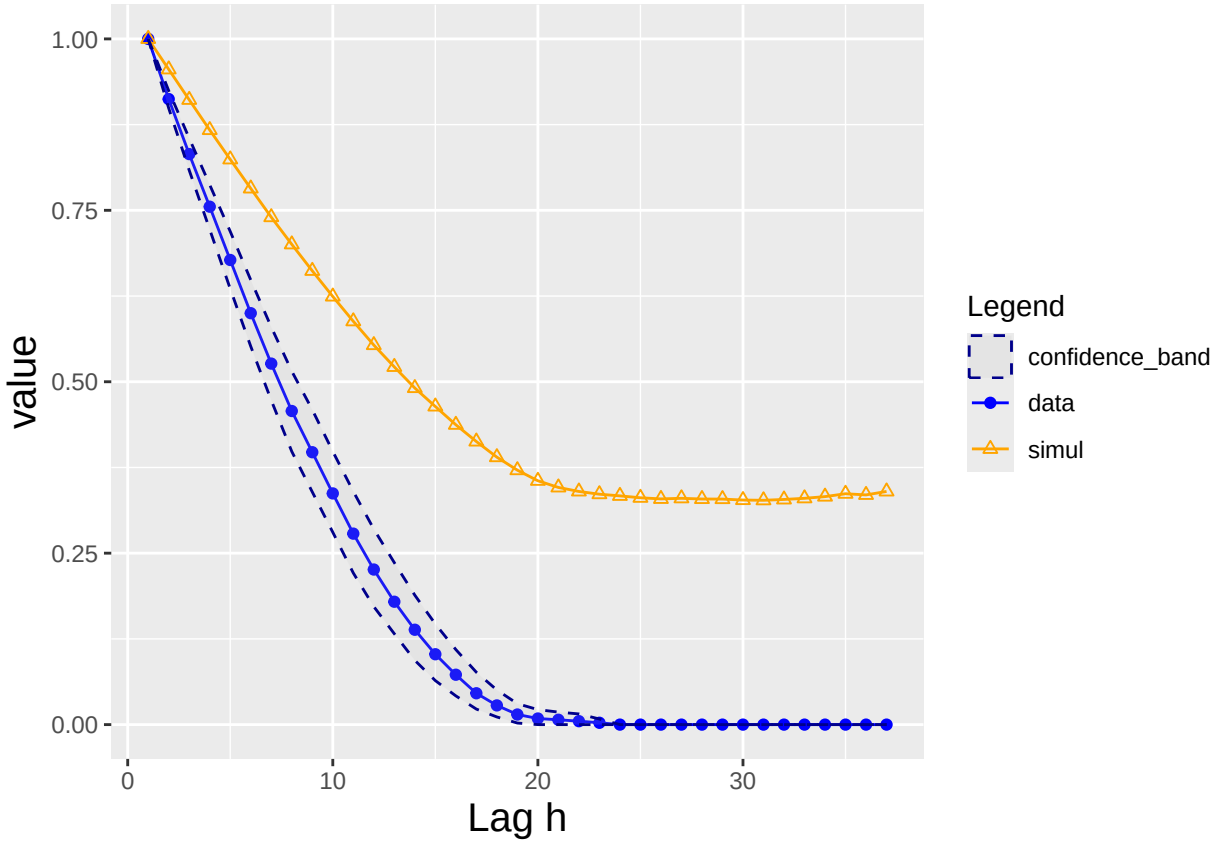
```
Matrice_couples<-list()
L<-ncol(X_exts)
z<-1
vecteur_distances<-c()
for(j in 1:L){
    #condition imposée sur le deuxième temps.
    valeurs_t_plus_h<-j:L
    for (i in valeurs_t_plus_h){
      Matrice_couples[[z]]<-c(i,j)
      vecteur_distances<-c(vecteur_distances,abs(i-j))
      z<-z+1
    }
```

```
}
q_chosen<-0.90
Tau1<-sapply(c(1:37),function(x,q,t){
  return(as.numeric(quantile(x[,t],q)))},q=q_chosen,
  x=Sim_X)

Tau2<-sapply(c(1:37),function(x,q,t){
  return(as.numeric(quantile(x[,t],q)))},q=q_chosen,
  x=X_exts)
```

```
Extremogram_simulations<-empirical_extremogram(Matrix_couples =Matrice_couples,Tau = Tau1,inds_select= S
Extremogram_data<-empirical_extremogram(Matrix_couples=Matrice_couples,inds_select = X_exts,Tau = Tau2)
N_rep<-500
B<-nrow(X_exts)
Result<-replicate(n =N_rep,expr = fnct_estim_extremo_resample(B = B,inds_ext = X_exts,Tau = Tau2,Matrix_
```

```
fnct_quantile<-function(x,q){
    M<-apply(X = x,MARGIN = 2,FUN = function(X){return(as.numeric(quantile(X,q)))})
    return(M)
  }
Q05<-fnct_quantile(x = t(Result),q=0.025)
Q95<-fnct_quantile(x = t(Result),q=0.975)
df<-cbind.data.frame(Extremogram_simulations,Extremogram_data,vecteur_distances)
colnames(df)<-c("simulation_value","data_value","delta")

result_delta<-df %>% group_by(delta) %>% summarise(val_data=mean(data_value),
                                                   val_simul=mean(simulation_value))
result_delta$bound_inf<-Q05
result_delta$bound_sup<-Q95
extremo_<-as.data.frame(result_delta[,c("val_data","val_simul","bound_inf","bound_sup")])
extremo_$Time<-c(1:nrow(extremo_))
```

```
GG_extremo<-ggplot(extremo_,aes(x=Time,y=val_data,col="data"))+
  geom_line()+
  geom_point()+
  geom_line(aes(y=val_simul,col="simul"))+
  geom_point(aes(y=val_simul,col="simul"),pch=2)+
  geom_ribbon(mapping = aes(ymin=bound_inf,ymax=bound_sup,col="confidence_band"),alpha=0.15,
              fill="grey", linetype = "dashed")+
  scale_color_manual(values=cols_)+
  theme(axis.title=element_text(size=15))+
  labs(col="Legend")+
  xlab("Lag h")+
  ylab("value")
GG_extremo
```



(4) Identifying simulated time series with machine learning algorithms

```
par(mfrow=c(1,1))
### Hyper-parameters of our machine learning algorithms.
H_Params<-c("radial","logit",500)

### Parameters of our experience.
NB_times<-100
ALPHA_PROP<-0.10
typeS<-"simple"
result<-matrix(NA,ncol=3,nrow=3)
```

```
# Use the time series as input
Result_X<-Running_perfs_ML(Base_simul =Sim_X,
                  Base_data = X_exts,
                  hyp_param = H_Params,
                  NB_times = NB_times,
                  alpha_prop = ALPHA_PROP,
                  K = K,
                  type_sampling = typeS,
                  title_ROC = "ROC curves",
                  NB_shown_ROC = 20)
```

```
## Warning: glm.fit: l'algorithme n'a pas convergé
```

```
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
```

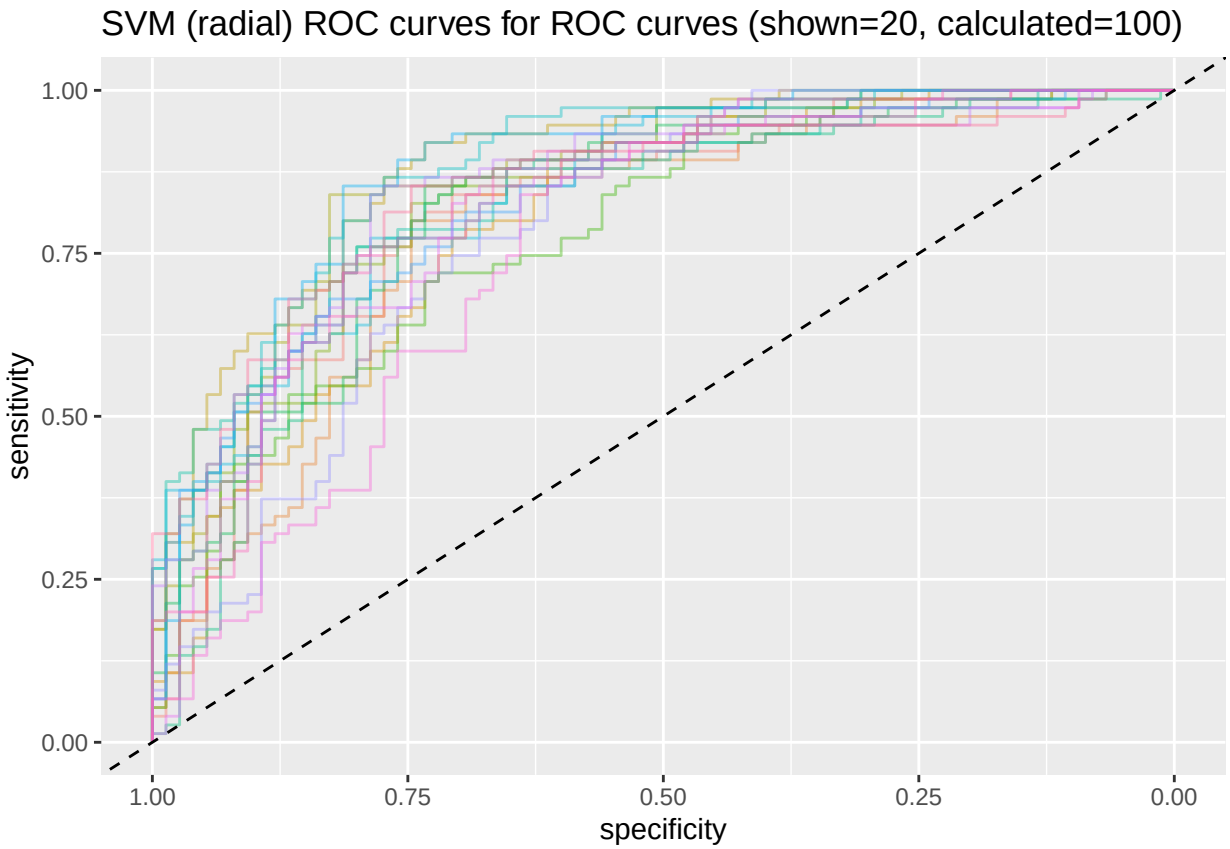
```
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
```

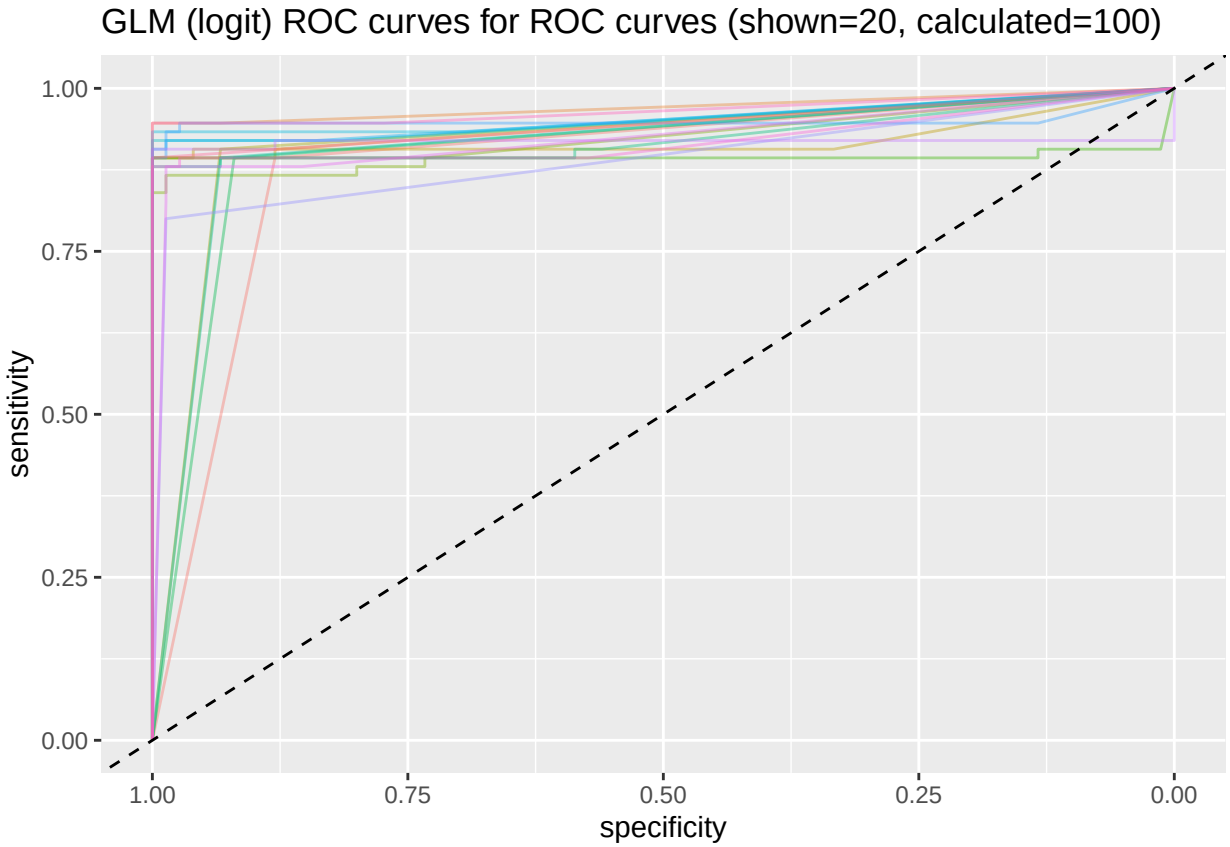
```
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé
```

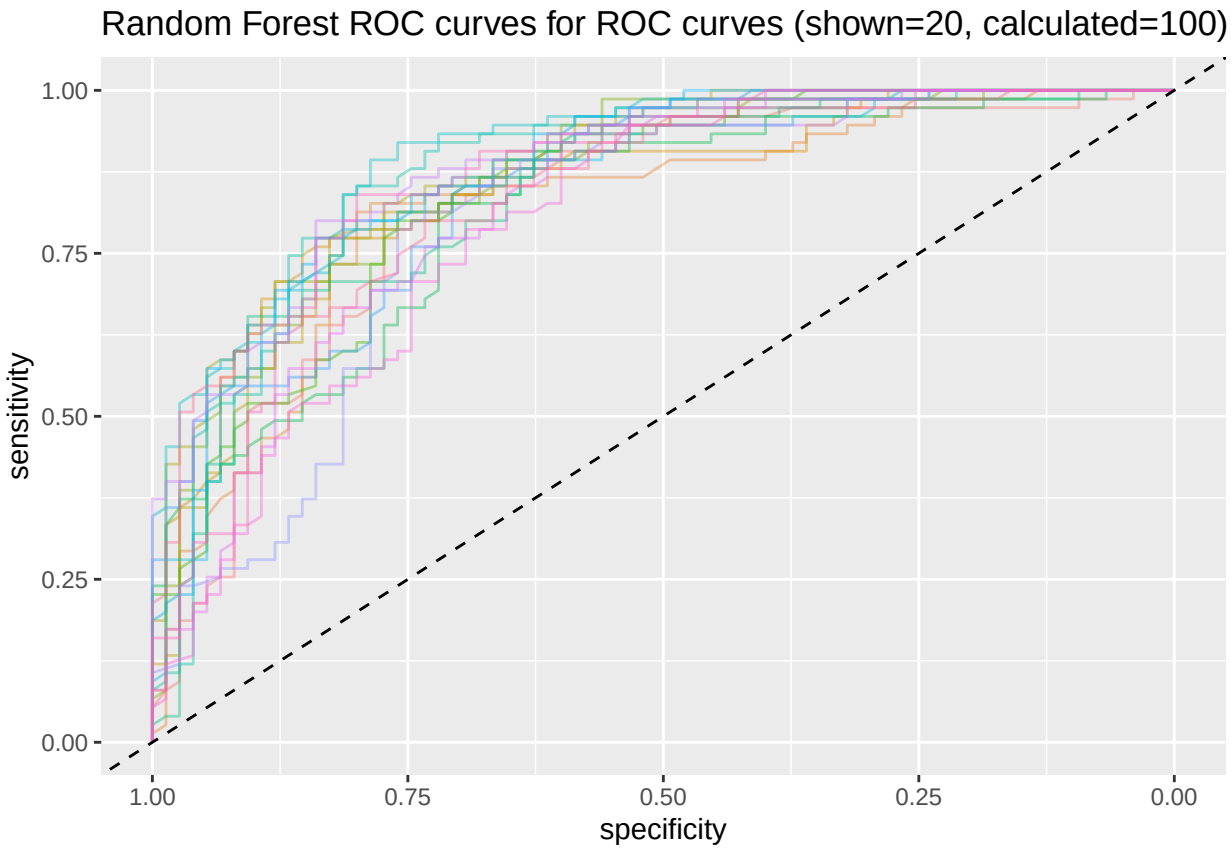
```
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé
```

```
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
```

```
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
```

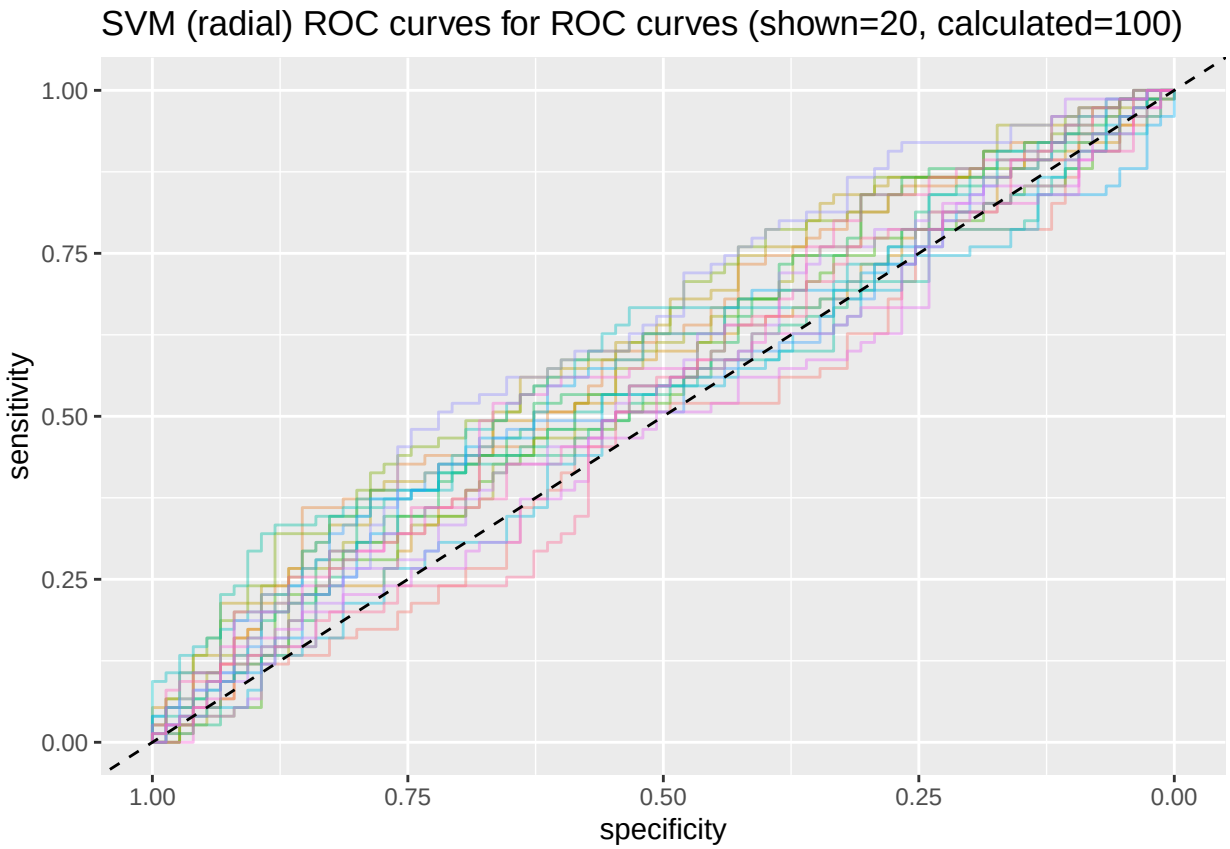

SVM (radial) ROC curves for ROC curves (shown=20, calculated=100)
sensitivity
1.00
0.75
0.50
0.25
0.00
specificity
1.00
0.75
0.50
0.25
0.00

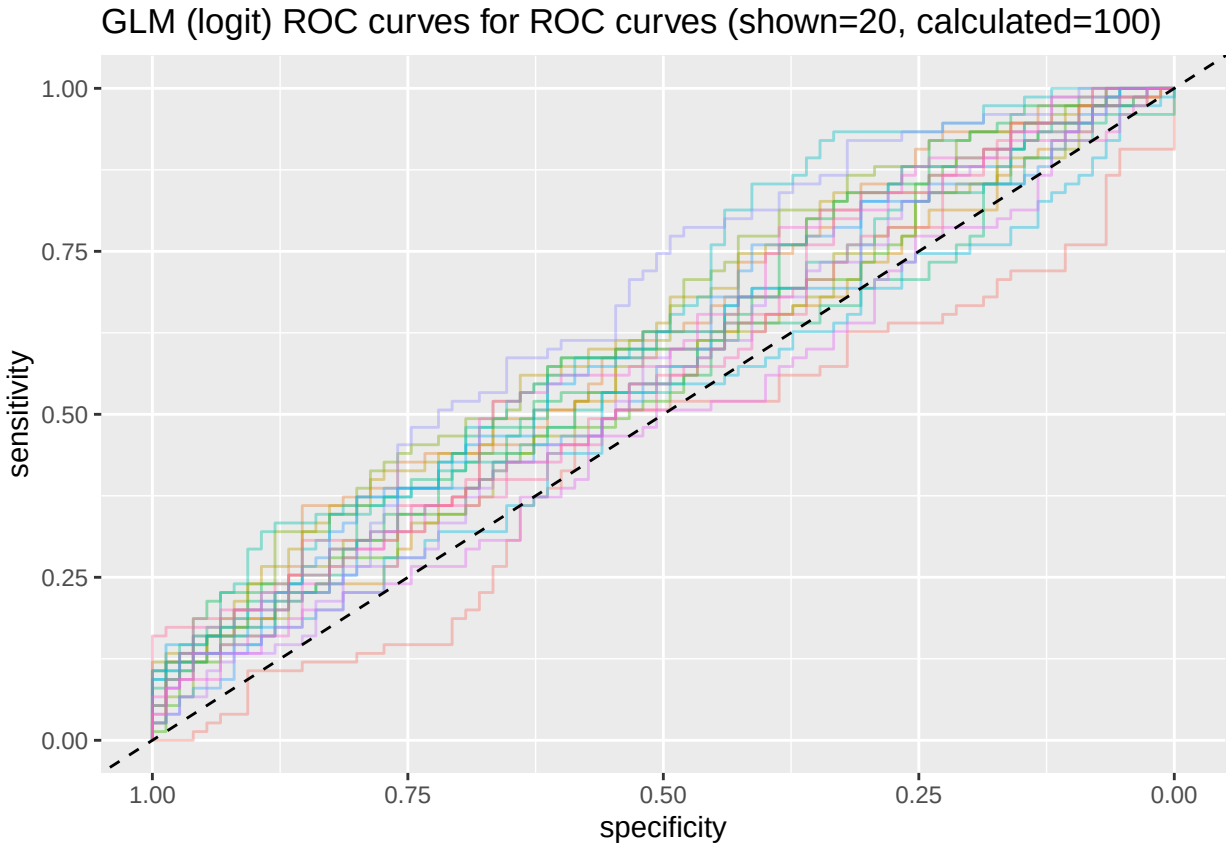

GLM (logit) ROC curves for ROC curves (shown=20, calculated=100)
sensitivity
1.00
0.75
0.50
0.25
0.00
specificity
1.00
0.75
0.50
0.25
0.00

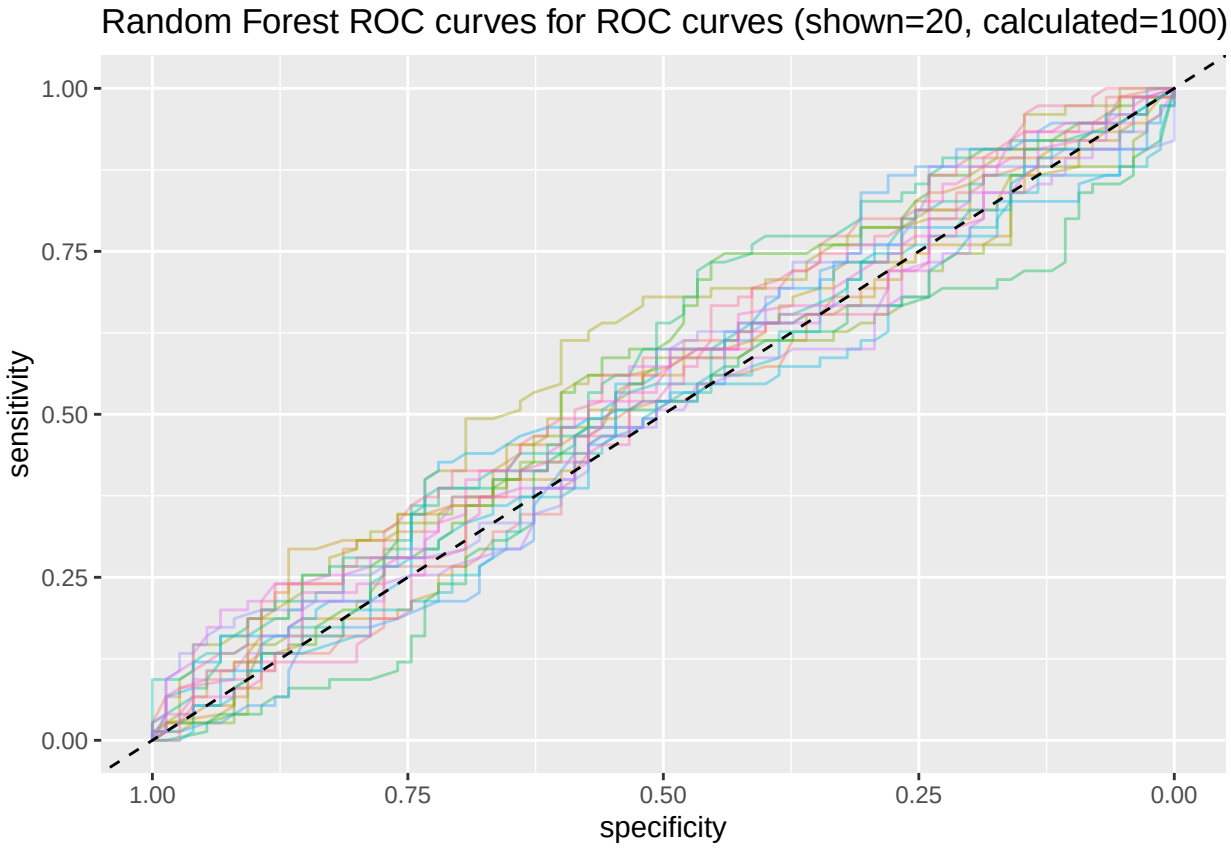



```
result[1,]<-Result_X$Accuracy$qu_accuracy
```

```
### Use the L2 norm as input
Result_l_f<-Running_perfs_ML(Base_simul =L2_simul,
                  Base_data =L2_exts,
                  hyp_param = H_Params,
                  NB_times = NB_times,
                  alpha_prop = ALPHA_PROP,
                  K = K,
                  type_sampling = typeS,
                  title_ROC = "ROC curves",
                  NB_shown_ROC = 20)
```


SVM (radial) ROC curves for ROC curves (shown=20, calculated=100)
sensitivity
1.00
0.75
0.50
0.25
0.00
specificity
1.00
0.75
0.50
0.25
0.00


GLM (logit) ROC curves for ROC curves (shown=20, calculated=100)
sensitivity
1.00
0.75
0.50
0.25
0.00
specificity
1.00
0.75
0.50
0.25
0.00




```
result[2,]<-Result_l_f$Accuracy$qu_accuracy
```

```
### Use the angle as input.
Result_A_f<-Running_perfs_ML(Base_simul =Angle_simul,
                  Base_data = Angle_exts,
                  hyp_param = H_Params,
                  NB_times = NB_times,
                  alpha_prop = ALPHA_PROP,
                  K = K,
                  type_sampling = typeS,
                  title_ROC = "ROC curves",
                  NB_shown_ROC = 50)
```

```
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
```

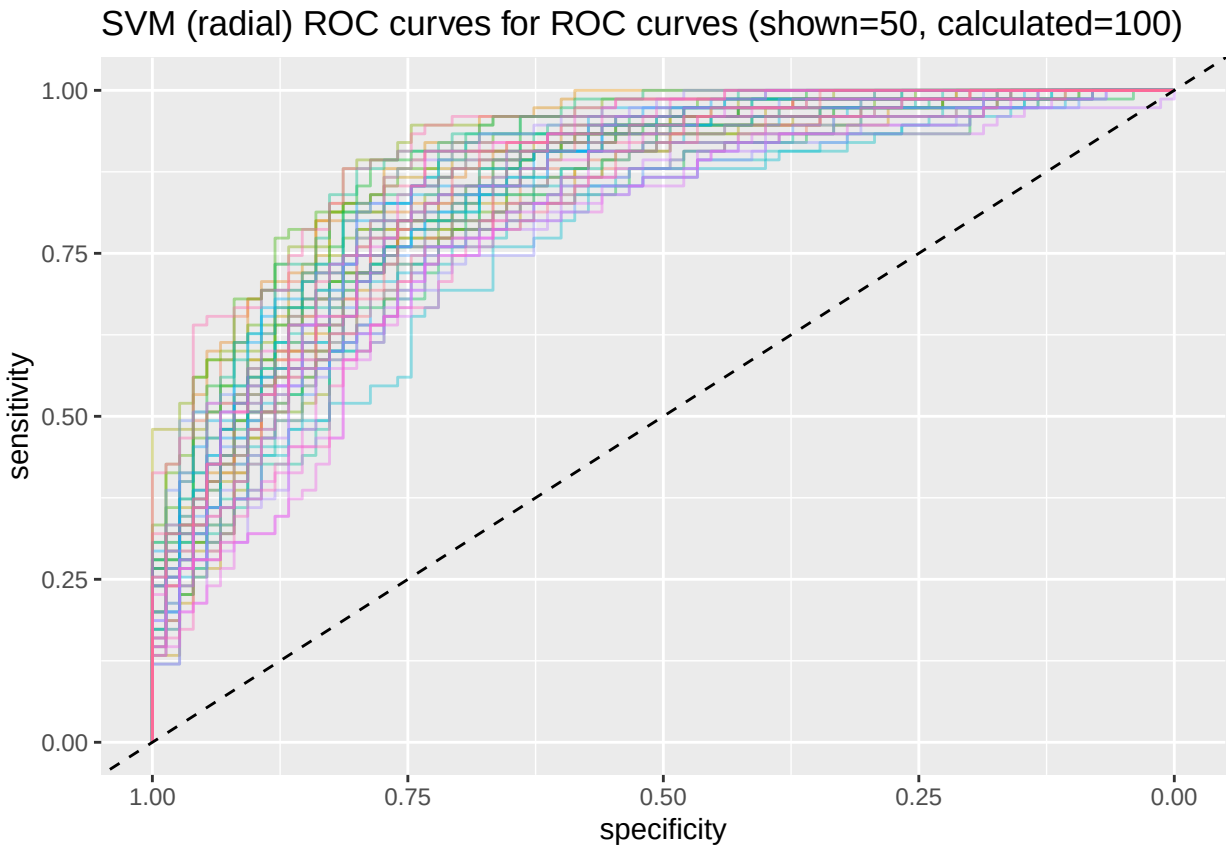
```
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé
```

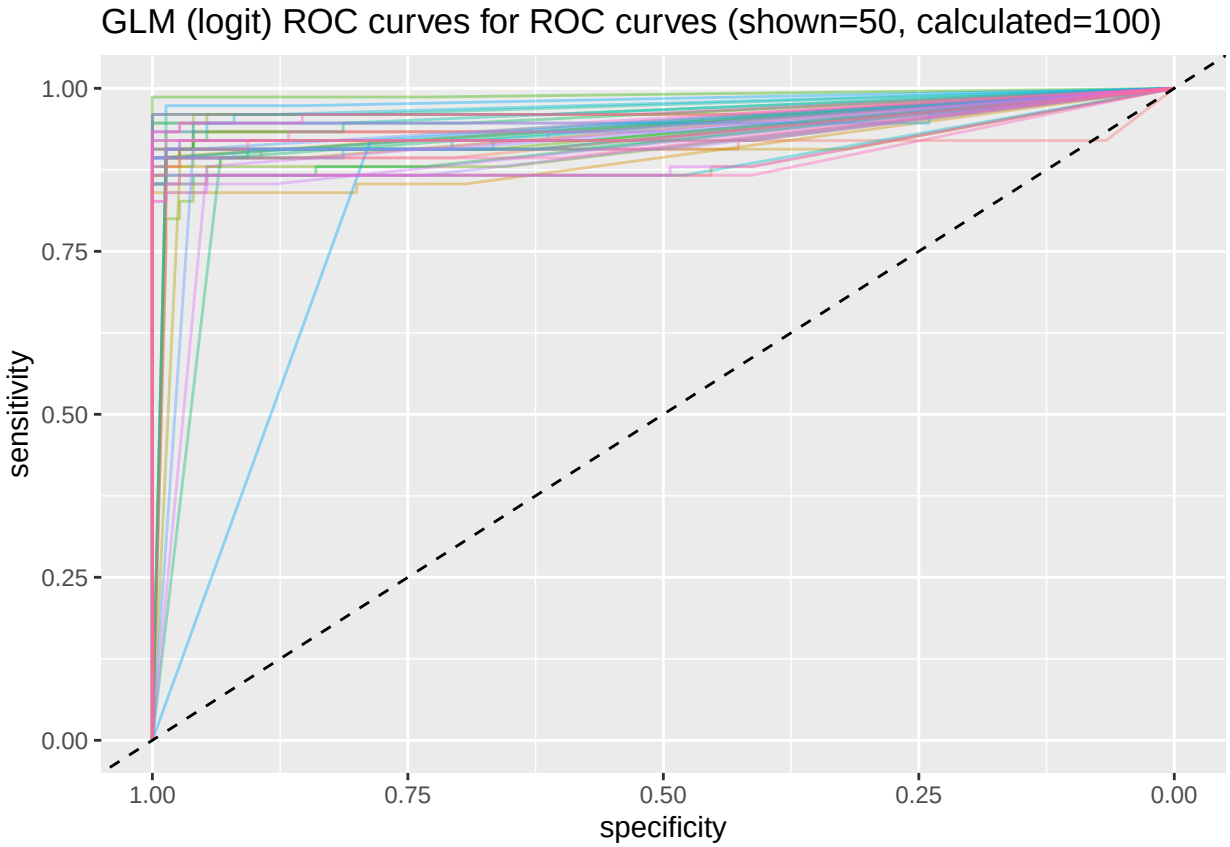
```
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
```

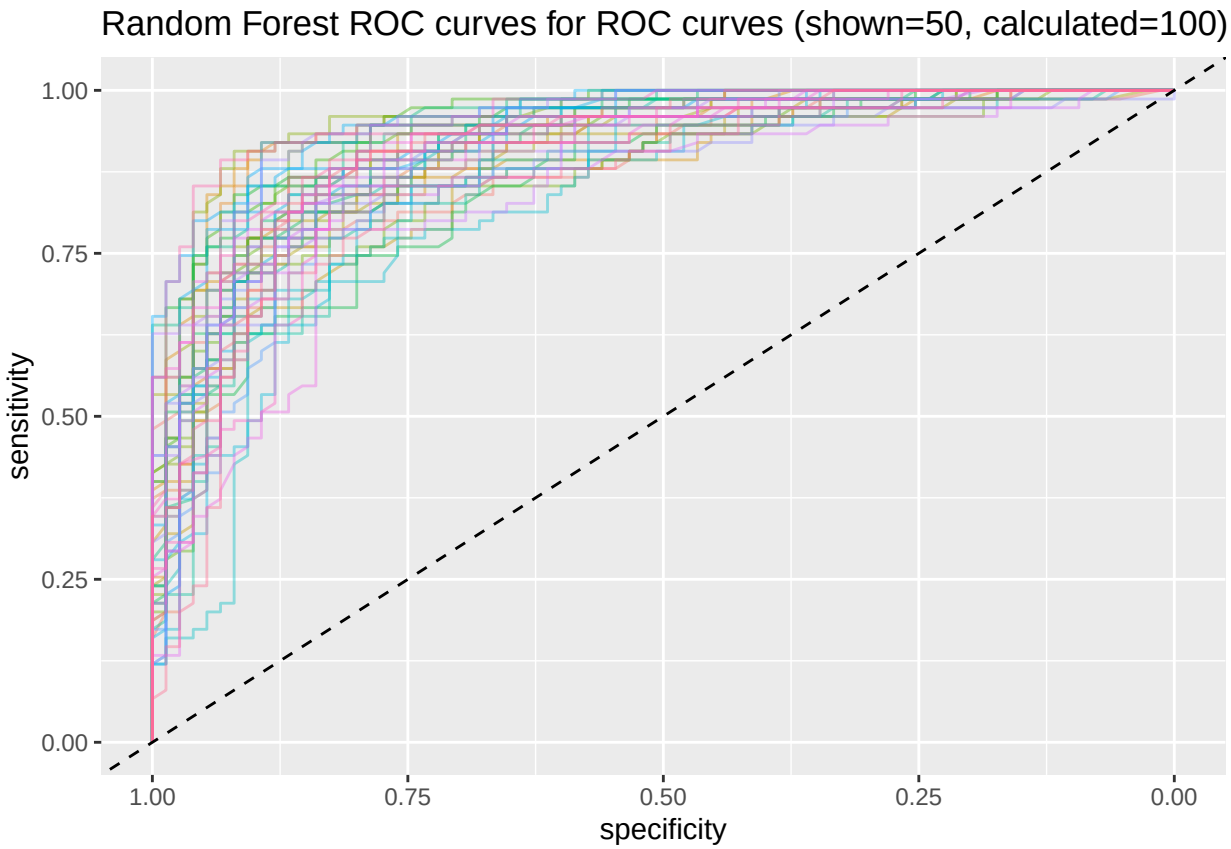
```
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
```

```
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: l'algorithme n'a pas convergé
```

```
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1

## Warning: glm.fit: l'algorithme n'a pas convergé

## Warning: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
```


SVM (radial) ROC curves for ROC curves (shown=50, calculated=100)
sensitivity
1.00
0.75
0.50
0.25
0.00
specificity
1.00
0.75
0.50
0.25
0.00


GLM (logit) ROC curves for ROC curves (shown=50, calculated=100)
sensitivity
1.00
0.75
0.50
0.25
0.00
specificity
1.00
0.75
0.50
0.25
0.00




```
result[3,]<-Result_A_f$Accuracy$qu_accuracy
```

Summary of our resuts

```
result
```

```
##      [,1]    [,2]    [,3]
## [1,] "71-81" "88-96" "71-81"
## [2,] "47-60" "49-59" "43-56"
## [3,] "72-82" "90-97" "77-89"
```