Regional frequency analysis of the annual flows in Piemonte and Valle d'Aosta

Alberto Viglione

Abstract

TO BE WRITTEN

Introduction

Many practical hydrological problems require reliable models for estimation of mean annual runoff in a region. Runoff cannot be interpolated like purely distributed variables, as precipitation or temperature, because runoff in a cross section is representative of the whole contributing basin. Therefore, usual spatial interpolation methods cannot be used for estimation in ungauged basins. As regards the statistical approach, one of the firsts and more popular methods in regional frequency analysis is the "index-flood" technique (Dalrymple, 1960). Many Regional Flood estimation projects (see e.g. Rossi and Villani, 1995; Robson and Reed, 1999) are based on Dalrymple's methodology, but also flow duration curves can be referred to the index flow method (Claps and Fiorentino, 1997; Castellarin et al., 2004a,b).

In this work we are interested in the annual flow, that is the amount of water crossing a river section in one year. If compared with hydrological extremes, applications of regional analysis to average variables, like the annual flow, are much less frequent in literature. Vogel and Wilson (1996) present some applications related to the US, while in Italy some previous works can be traced back to Ferraresi et al. (1988), Claps and Mancino (2002) and Brath et al. (2004). The purpose of the Regional frequency analysis of the annual flow is the estimation of its probability distribution in basins with few or no data.

The fundamental hypothesis of Dalrymple's method is that the distribution of a variable in different sites belonging to a "homogeneous region" is identical, with the exception of the scale parameter, the index-flow. In this document we show how the `nsRFA` package can be used to:

1. regionalize the index-flow;
2. regionalize the growth curve, i.e. the rescaled distribution function.

The methodology has been applied to Piemonte and Valle dâĂŹAosta, two contiguous regions in the North-West of Italy. This territory is characterized by a marked heterogeneity. In this relatively small region, very different orographic and climatic conditions coexist: in few hundreds kilometres the climate changes from the appenninic-mediterranean one in the south-eastern hills to the alpine-continental one in the mountainous Valle d'Aosta, passing from all the intermediate conditions. For this reason, a regional frequency analysis in this territory is both complex and interesting.

The following results are documented in Viglione et al. (2006) and Viglione (2007).

TO BE WRITTEN

Data

In `nsRFA` data referred to 47 basins in Piemonte and Valle d'Aosta are in:

```
> data(hydroSIMN)
```

To have some information on these data

```
> ls()
> help(hydroSIMN)
```

The object used in this work are `annualflows`, a data.frame containing the annual flows of 47 hydrometric stations in Piemonte and Valle d'Aosta, measured by the SIMN (Servizio Idrografico e Mareografico Nazionale), and `parameters`, a data.frame containing morphometric and climatic descriptors that have been derived for all these river basins.

Regionalization of the index-flow

The "index-flow" parameter can be either the sample mean (e.g. Hosking and Wallis, 1997) or the sample median (e.g. Robson and Reed, 1999). Viglione et al. (2007) show that, for variables characterized by low skewness coefficients, the estimation of the mean is less biased than that of the median. For this reason in this work the sample mean is used as the index-flow. Due to its simplicity, the most frequently used method to estimate the index-flow is the multiregressive approach (see e.g. Kottegoda and Rosso, 1997), that relates the index-flow to catchment characteristics, such as climatic indices, geologic and morphologic parameters, land cover type, etc., through linear (used here) or non-linear equations.

The choice of the best linear regressions between the mean annual flow and the catchment attributes is performed using the function `bestlm()`. Different types of linear models are investigated. The candidate dependent variable is selected between 4 possibilities:

```
> Dm <- parameters[, "Dm"]
> logDm <- log(Dm)
> sqrtDm <- sqrt(Dm)
> sqrt3Dm <- Dm^(1/3)
```

The candidate regressors are:

```
> attributes <- parameters[, -c(1, 2)]
> logattributes <- log(attributes[, -c(7:9)])
> mixedattributes <- cbind(attributes, logattributes[, 1])
> names(mixedattributes) <- c(names(attributes), "lnAm")
```

The best regressions with the non transformed mean annual flows are:

```
> nontrasfregr <- bestlm(Dm, mixedattributes, kmax = 3, nbest = 4)
> nontrasfregr
```

```
                model     R2adj
1    Am + S2000 + IT 0.8851092
2  S2000 + IT + lnAm 0.8845462
3    Am + S2000 + IB 0.8793968
4  Am + S2000 + lnAm 0.8789050
5       S2000 + lnAm 0.8773823
6         Am + S2000 0.8761521
7          Hm + lnAm 0.8648352
8            Am + Hm 0.8588129
9                 IT 0.8467383
```

```
10                IB 0.6094562
11             lnAm 0.5440745
12                Am 0.5323001
```

Other diagnostics of these regressions can be obtained using the functions in `REGRDIAGNOSTICS`. Here we calculate the Root Mean Squared Error (RMSE), and the Root Mean Squared Error of the cross validation (RMSEjk)

```
> nregr <- dim(nontrasfregr$subselect)[1]
> diagn <- data.frame(matrix(NA, nrow = nregr, ncol = 2))
> names(diagn) <- c("RMSE", "RMSEjk")
> for (i in 1:nregr) {
+     f <- paste("Dm ~", paste(colnames(nontrasfregr$subselect)[nontrasfregr$subselect[i,
+         ]], collapse = " + "))
+     regr <- lm(f, mixedattributes)
+     diagn[i, 1] <- RMSE.lm(regr)
+     diagn[i, 2] <- RMSEjk.lm(regr)
+ }
> diagn
```

```
       RMSE   RMSEjk
1  104.0258 113.4305
2  104.2804 114.1658
3  106.5805 116.2866
4  106.7976 116.3828
5  108.7094 116.6226
6  109.2534 116.9209
7  114.1359 122.2157
8  116.6509 124.9485
9  122.9100 127.3007
10 196.2029 203.4998
11 211.9910 219.7052
12 214.7109 222.3708
```

The best multiplicative regressions (linearized with the logarithms) are:

```
> multregr <- bestlm(logDm, logattributes, kmax = 3, nbest = 4)
> multregr
```

```
            model     R2adj
1  Hm + LLDP + IB 0.8468383
2     S + Hm + IB 0.8467204
3  Am + Hm + LLDP 0.8458270
4     Am + S + Hm 0.8447653
5         Am + Hm 0.8330193
6         Hm + IB 0.8245532
7         IT + IB 0.8069278
8         Am + IT 0.8015272
9              IT 0.7787654
10             IB 0.5571305
11             Am 0.5172267
12             Pm 0.3590828
```

whose RMSE and RMSEjk are:

```
> nregr <- dim(multregr$subselect)[1]
> diagn <- data.frame(matrix(NA, nrow = nregr, ncol = 2))
> names(diagn) <- c("RMSE", "RMSEjk")
> for (i in 1:nregr) {
+     f <- paste("logDm ~", paste(colnames(multregr$subselect)[multregr$subselect[i,
+         ]], collapse = " + "))
+     regr <- lm(f, logattributes)
+     fitt <- regr$fitted.values
+     crossval <- jackknife1.lm(regr)
+     diagn[i, 1] <- RMSE(Dm, exp(fitt))
+     diagn[i, 2] <- RMSE(Dm, exp(crossval))
+ }
> diagn

       RMSE   RMSEjk
1  129.3508 141.2024
2  128.9386 141.0293
3  127.9561 139.3443
4  127.5461 139.2905
5  130.8245 139.3654
6  135.7904 145.6343
7  140.2605 148.5320
8  141.6007 150.6099
9  149.1973 156.0839
10 214.6567 224.6109
11 217.2887 225.2206
12 266.0818 275.1440
```

The best regressions with the transformed mean annual flows are:

```
> trasfregr_log <- bestlm(logDm, mixedattributes, kmax = 3, nbest = 4)
> trasfregr_log

                model     R2adj
1     Hm + NORD + IB 0.9001788
2   Hm + NORD + lnAm 0.8922544
3    Hm + EST + lnAm 0.8893036
4  Hm + S2000 + lnAm 0.8885439
5       S2000 + lnAm 0.8838486
6          Hm + lnAm 0.8828080
7            Hm + IB 0.8823572
8         S2000 + IB 0.8723769
9                 IT 0.8382124
10                IB 0.5944382
11             lnAm 0.5172267
12               Am 0.4877632

> nregr <- dim(trasfregr_log$subselect)[1]
> diagn <- data.frame(matrix(NA, nrow = nregr, ncol = 2))
> names(diagn) <- c("RMSE", "RMSEjk")
```

```
> for (i in 1:nregr) {
+     f <- paste("logDm ~", paste(colnames(trasfregr_log$subselect)[trasfregr_log$subselect[i,
+         ]], collapse = " + "))
+     regr <- lm(f, mixedattributes)
+     fitt <- regr$fitted.values
+     crossval <- jackknife1.lm(regr)
+     diagn[i, 1] <- RMSE(Dm, exp(fitt))
+     diagn[i, 2] <- RMSE(Dm, exp(crossval))
+ }
> diagn

       RMSE   RMSEjk
1  101.7852 110.5497
2  103.2933 112.9716
3  101.7492 111.8125
4  103.0333 112.0052
5  106.2369 113.4683
6  108.6799 116.1945
7  111.3227 118.5316
8  117.3381 125.3689
9  136.1306 144.1668
10 202.3507 210.0492
11 217.2887 225.2206
12 226.6542 236.5474

> trasfregr_sqrt <- bestlm(sqrtDm, mixedattributes, kmax = 3, nbest = 4)
> trasfregr_sqrt

                 model     R2adj
1    S2000 + IT + lnAm 0.8936962
2    Hm + S2000 + lnAm 0.8893312
3       Hm + NORD + IB 0.8893250
4         S2000 + lnAm 0.8877719
5  S2000 + NORD + lnAm 0.8876825
6            Hm + lnAm 0.8801759
7              Hm + IB 0.8747952
8           Am + S2000 0.8738923
9                   IT 0.8490673
10                  IB 0.6065118
11                lnAm 0.5348015
12                  Am 0.5137850

> nregr <- dim(trasfregr_sqrt$subselect)[1]
> diagn <- data.frame(matrix(NA, nrow = nregr, ncol = 2))
> names(diagn) <- c("RMSE", "RMSEjk")
> for (i in 1:nregr) {
+     f <- paste("sqrtDm ~", paste(colnames(trasfregr_sqrt$subselect)[trasfregr_sqrt$subselect
+         ]], collapse = " + "))
+     regr <- lm(f, mixedattributes)
+     fitt <- regr$fitted.values
+     crossval <- jackknife1.lm(regr)
```

```
+     diagn[i, 1] <- RMSE(Dm, fitt^2)
+     diagn[i, 2] <- RMSE(Dm, crossval^2)
+ }
> diagn

       RMSE   RMSEjk
1  100.5960 109.7789
2  103.8121 113.0390
3  105.1730 114.7534
4  105.9916 113.4752
5  105.2430 115.6963
6  109.2252 116.6429
7  112.8295 120.2861
8  110.4650 118.5180
9  124.9163 129.9945
10 197.6366 204.9570
11 213.2123 220.8817
12 218.5031 226.6602

> trasfregr_sqrt3 <- bestlm(sqrt3Dm, mixedattributes, kmax = 3,
+     nbest = 4)
> trasfregr_sqrt3

                model     R2adj
1     Hm + NORD + IB 0.8944924
2  S2000 + IT + lnAm 0.8936365
3  Hm + S2000 + lnAm 0.8906543
4   Hm + NORD + lnAm 0.8906427
5       S2000 + lnAm 0.8881545
6          Hm + lnAm 0.8825746
7            Hm + IB 0.8788041
8         S2000 + IB 0.8725909
9                 IT 0.8469916
10                IB 0.6035465
11             lnAm 0.5298848
12               Am 0.5059347

> nregr <- dim(trasfregr_sqrt3$subselect)[1]
> diagn <- data.frame(matrix(NA, nrow = nregr, ncol = 2))
> names(diagn) <- c("RMSE", "RMSEjk")
> for (i in 1:nregr) {
+     f <- paste("sqrt3Dm ~", paste(colnames(trasfregr_sqrt3$subselect)[trasfregr_sqrt3$subsel
+         ]], collapse = " + "))
+     regr <- lm(f, mixedattributes)
+     fitt <- regr$fitted.values
+     crossval <- jackknife1.lm(regr)
+     diagn[i, 1] <- RMSE(Dm, fitt^3)
+     diagn[i, 2] <- RMSE(Dm, crossval^3)
+ }
> diagn
```

```
       RMSE   RMSEjk
1  103.6245 112.8929
2  100.1596 109.2270
3  103.1318 112.2067
4  103.9504 113.9002
5  105.7137 113.0890
6  108.4943 115.8495
7  111.8762 119.2041
8  116.0246 124.1258
9  127.2240 132.8930
10 198.7465 206.1318
11 214.1709 221.8712
12 220.5780 229.0946
```

The choice of the best regression is based on the RMSE of the cross-validation (function `RMSEjk.lm` or `jackknife1.lm` plus `RMSE`). So the best regression is:

```
> bestregr <- lm(sqrt3Dm ~ S2000 + IT + lnAm, mixedattributes)
> bestregr

Call:
lm(formula = sqrt3Dm ~ S2000 + IT + lnAm, data = mixedattributes)

Coefficients:
(Intercept)        S2000           IT         lnAm
  -14.86525      0.01601      0.71038      3.32829

> summary(bestregr)

Call:
lm(formula = sqrt3Dm ~ S2000 + IT + lnAm, data = mixedattributes)

Residuals:
     Min       1Q   Median       3Q      Max
-0.84607 -0.16793 -0.01297  0.14732  0.80262

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -14.86525    5.72525  -2.596 0.012840 *
S2000         0.01601    0.00344   4.655 3.11e-05 ***
IT            0.71038    0.39298   1.808 0.077653 .
lnAm          3.32829    0.83259   3.998 0.000247 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 0.3457 on 43 degrees of freedom
Multiple R-squared: 0.9006,        Adjusted R-squared: 0.8936
F-statistic: 129.8 on 3 and 43 DF,  p-value: < 2.2e-16
```

that we check with the following tests: the Variance inflation factors (if VIF > 5 there is a problem of multicollinearity) and correlation between the regressors:

```
> vif.lm(bestregr)

    S2000        IT      lnAm
 4.692027 11.211280 13.021852

> cor(bestregr$model[-1])

           S2000        IT      lnAm
S2000  1.0000000 0.1500960 -0.398042
IT     0.1500960 1.0000000  0.804859
lnAm  -0.3980420 0.8048590  1.000000
```

the Student t test of significance of the coefficients (probability $Pr(>|t|)$ of the significance test, the smallest-the best):

```
> prt.lm(bestregr)

       S2000           IT         lnAm
3.105746e-05 7.765271e-02 2.469395e-04
```

So there is a correlation problem between IT and lnAm, that can cause collinearity, and that causes the non-significance of the coefficient of IT in the model.

Therefore we choose:

```
> bestregr <- lm(logDm ~ Hm + NORD + IB, mixedattributes)
> bestregr

Call:
lm(formula = logDm ~ Hm + NORD + IB, data = mixedattributes)

Coefficients:
(Intercept)           Hm         NORD           IB
  7.8577186    0.0002910    0.0722161   -1.6956356

> summary(bestregr)

Call:
lm(formula = logDm ~ Hm + NORD + IB, data = mixedattributes)

Residuals:
      Min        1Q    Median        3Q       Max
-0.273533 -0.059161  0.007672  0.041782  0.244361

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.858e+00  7.849e-02 100.117  < 2e-16 ***
Hm           2.910e-04  2.465e-05  11.807 4.42e-15 ***
NORD         7.222e-02  2.427e-02   2.976  0.00478 **
IB          -1.696e+00  9.503e-02 -17.843  < 2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 0.1018 on 43 degrees of freedom
Multiple R-squared: 0.9067,        Adjusted R-squared: 0.9002
F-statistic: 139.3 on 3 and 43 DF,  p-value: < 2.2e-16
```

```
> prt.lm(bestregr)

          Hm         NORD           IB
4.420213e-15 4.780480e-03 1.707744e-21

> vif.lm(bestregr)

      Hm     NORD       IB
1.148364 1.330344 1.336530

> cor(bestregr$model[-1])

              Hm       NORD        IB
Hm    1.0000000 -0.1887886 0.2002600
NORD -0.1887886  1.0000000 0.4140169
IB    0.2002600  0.4140169 1.0000000
```

We also check the normality of the residuals (using a goodness-of-fit test):

```
> p_norm <- A2_GOFlaio(bestregr$residuals, dist = "NORM")
> p_norm

       A2     p(A2)
0.5963062 0.8765066
```

and calculate the root mean square errors:

```
> rmse <- RMSE(Dm, exp(bestregr$fitted.values))
> predicted <- jackknife1.lm(bestregr)
> rmse_jk <- RMSE(Dm, exp(predicted))
```

Some diagnostics can be plotted as (Figure 1):

```
> op <- par(mfrow = c(2, 2))
> plot(bestregr$fitted.values, bestregr$residuals, xlab = "Fitted",
+     ylab = "Residuals")
> abline(0, 0, lty = 3)
> normplot(bestregr$residuals, xlab = "Residuals")
> plot(parameters[, c("Dm")], exp(bestregr$fitted.values), xlab = "Originals",
+     ylab = "Fitted")
> abline(0, 1, lty = 3)
> intervals <- predinterval.lm(bestregr)
> intervals <- intervals[order(intervals[, 1]), ]
> plot(parameters[, c("Dm")], exp(predicted), xlab = "Originals",
+     ylab = "Predicted")
> abline(0, 1, lty = 3)
> lines(exp(intervals[, c(1, 2)]), lty = 2)
> lines(exp(intervals[, c(1, 3)]), lty = 2)
> par(op)
```

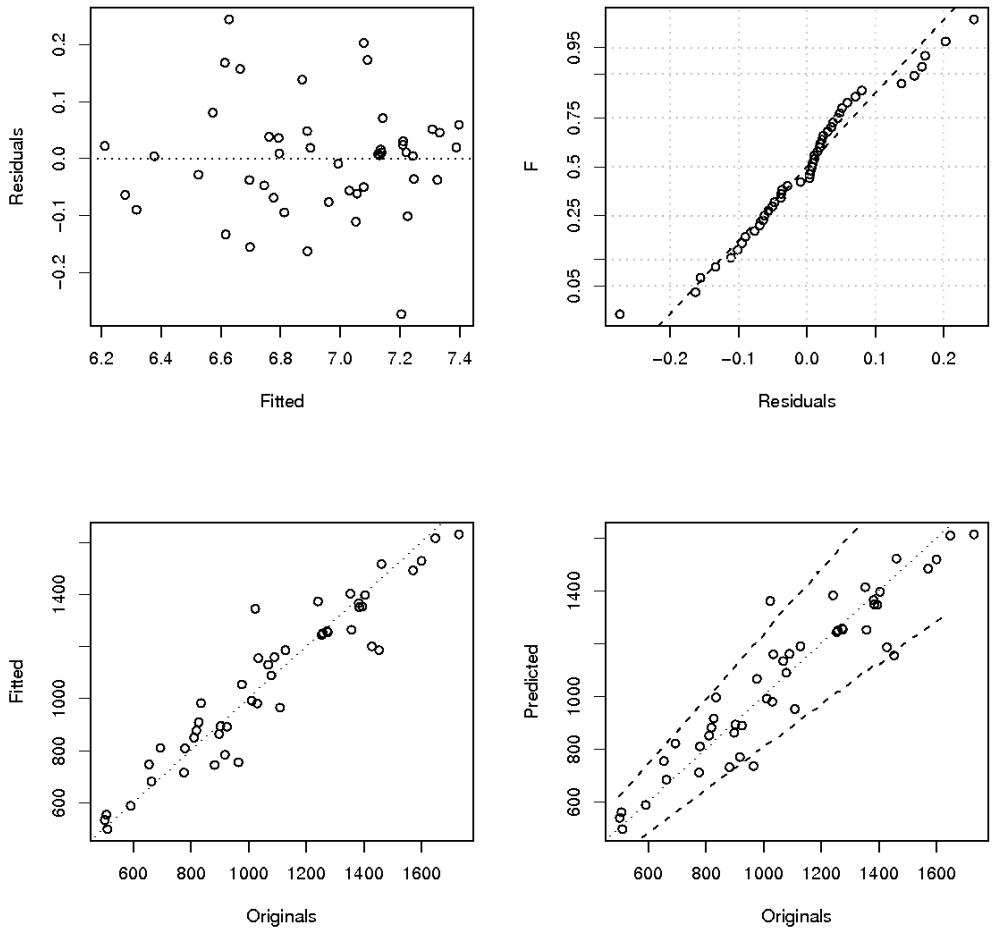

Figure 1: Diagnostic plots of the best regression model. Counterclockwise from upper left: residuals as a function of the estimated values; originals against the fitted values; result of cross-validation and normal plot of residuals.

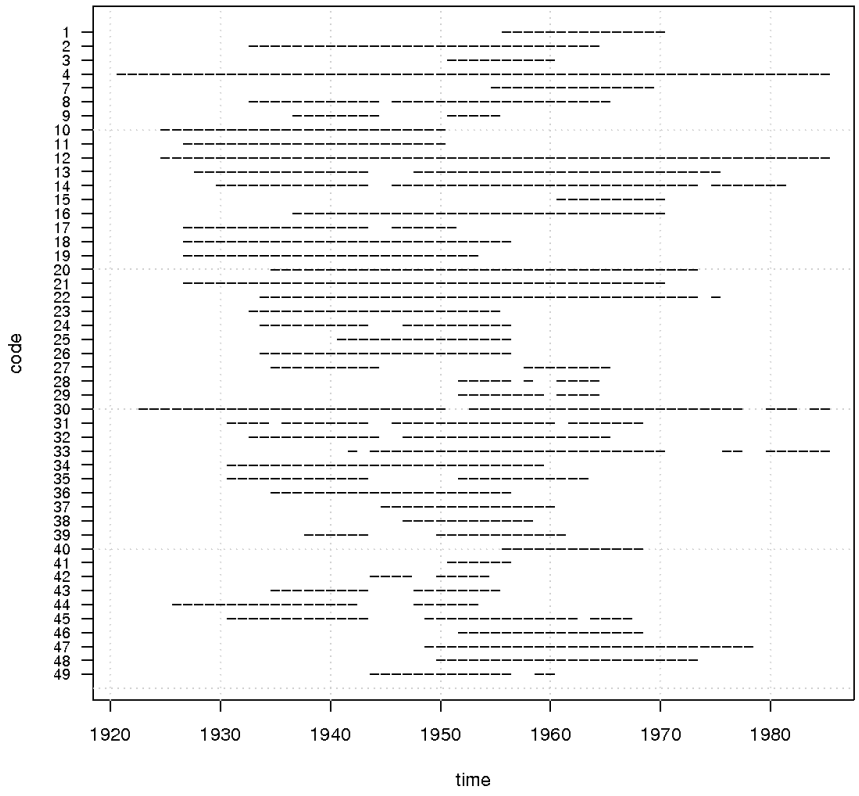

Figure 2: Data consistency.

Regionalization of the growth-curve

TO BE COMPLETED...

```
> D <- annualflows["dato"][, ]
> y <- annualflows["anno"][, ]
> cod <- annualflows["cod"][, ]
```

Plot consistency of data series (Figure 2):

```
> consistencyplot(y, cod)
```

Choice of sites with more than 15 records:

```
> ni <- tapply(D, cod, length)
> annualflows15 <- annualflows[unsplit(ni, cod) >= 15, ]
> parameters15 <- parameters[ni >= 15, ]
> D15 <- annualflows15["dato"][, ]
> cod15 <- annualflows15["cod"][, ]
```

L-moments of the series:

```
> LM15 <- data.frame(t(sapply(split(D15, cod15), Lmoments)))
```

L-moment ratios plot (Figure 3):

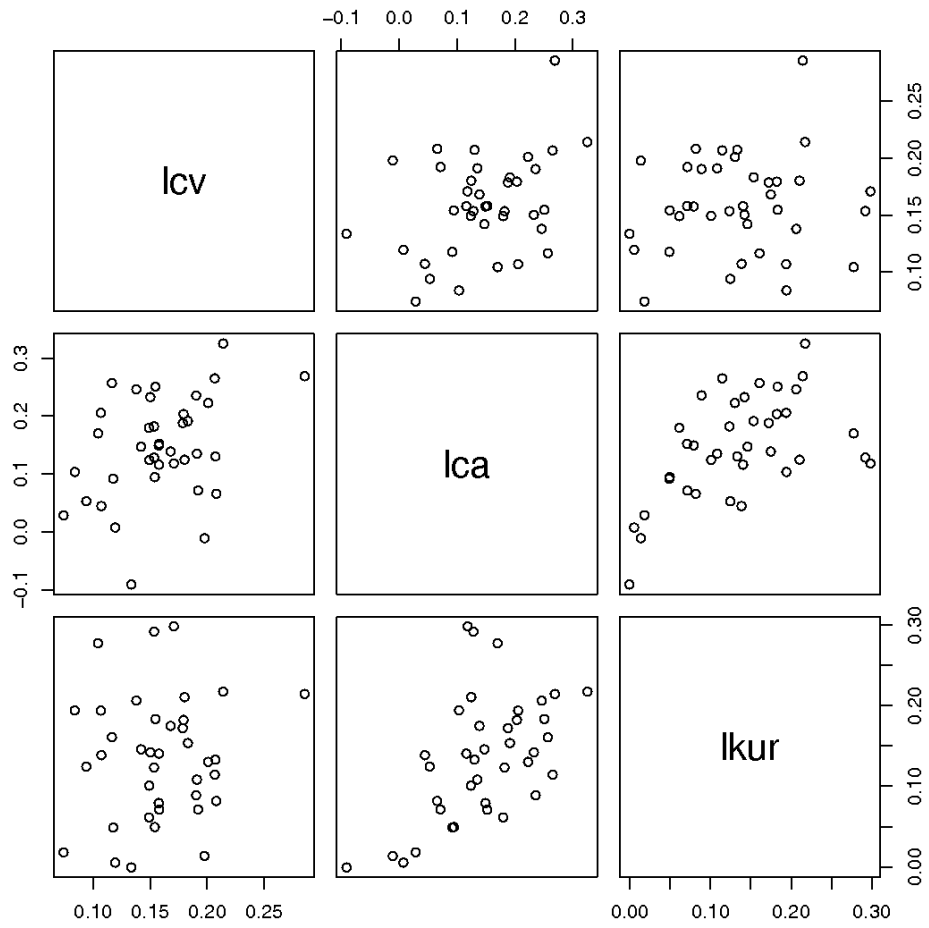

Figure 3: L-moment ratios plot.

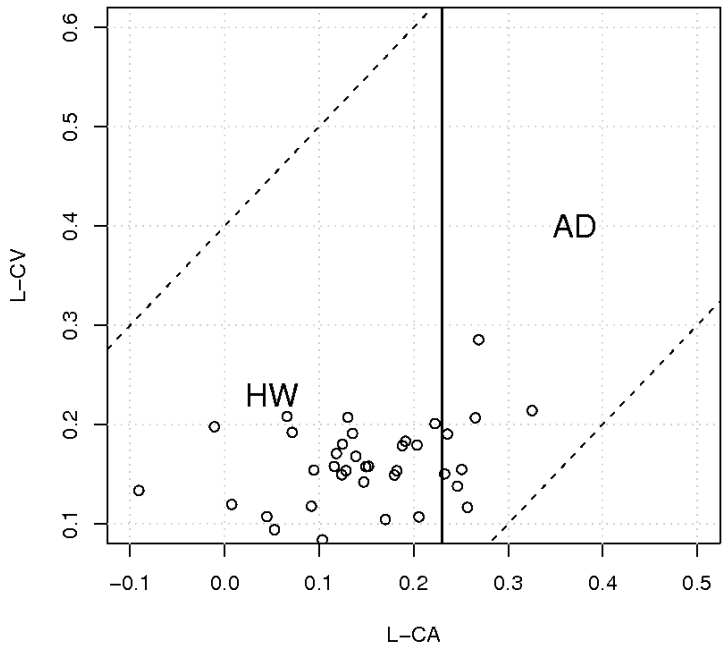

Figure 4: L-moment ratios plot.

```
> plot(LM15[3:5])
```

Which homogeneity test do I use:

```
> Lspace.HWvsAD()
> points(LM15[, 4:3])
```

Homogeneity test on the entire region:

```
> D15adim <- D15/unsplit(tapply(D15, cod15, mean), cod15)
> HWs <- HW.tests(D15adim, cod15)[1]
> HWs

      H1
7.776372
```

Choice of the classification variables through multi-regression approach:

```
> bestlm(LM15[, "lcv"], parameters15[, 3:16], kmax = 3)

               model     R2adj
1   S2000 + Rc + IB 0.6780821
2   Am + S2000 + Rc 0.6767409
3 Am + LLDP + S2000 0.6762082
4      S2000 + Ybar 0.6595223
5         Hm + Ybar 0.6435752
6        Am + S2000 0.6306392
```

```
7              S2000 0.5394290
8                 Hm 0.5321995
9               Ybar 0.3584742
```

or reasoning with distance matrices:

```
> bestlm(as.numeric(AD.dist(D15, cod15)), data.frame(apply(parameters15[,
+     3:16], 2, dist)), kmax = 3)

                model      R2adj
1  Am + S2000 + Ybar 0.16116558
2 S2000 + EST + Ybar 0.15822746
3  Pm + S2000 + Ybar 0.15718520
4       S2000 + Ybar 0.14887002
5          Hm + Ybar 0.14154585
6        S2000 + EST 0.11684503
7              S2000 0.10917391
8                 Hm 0.09862238
9               Ybar 0.05313668
```

We choose Hm and Ybar as classification variables. Mantel test:

```
> Y <- AD.dist(D15, cod15)
> X <- data.frame(apply(parameters15[, c("Hm", "Ybar")], 2, dist))
> datamantel <- cbind(as.numeric(Y), X)
> regrmantel <- lm(Y ~ Hm + Ybar, datamantel)
> mantel.lm(regrmantel, Nperm = 100)

  P.Hm P.Ybar
     1      1
```

Cluster formation:

```
> param <- parameters15[c("Hm", "Ybar")]
> n <- dim(param)[1]
> k <- dim(param)[2]
> param.norm <- (param - matrix(mean(param), nrow = n, ncol = k,
+     byrow = TRUE))/matrix(sd(param), nrow = n, ncol = k, byrow = TRUE)

> nclusters = 1
> while (max(HWs) > 2.1) {
+     nclusters <- nclusters + 1
+     clusters <- traceWminim(param.norm, nclusters)
+     indclusters <- unsplit(clusters, cod15)
+     HWs <- rep(NA, nclusters)
+     for (i in unique(clusters)) {
+         HWs[i] <- HW.tests(D15adim[indclusters == i], cod15[indclusters ==
+             i])[1]
+     }
+     print(HWs)
+ }
```

```
[1] 4.426389 1.617630
[1]  4.7201696 -0.1214777  0.4974403
[1]  1.7858483  1.9609622 -0.2935966  0.4070506
```

So the number of regions is 4.

Regional L-moments: My regional L-moment ratios are not calculated as in Hosking and Wallis, but using the pooled growth-curves.

```
> regLM15 <- t(sapply(split(D15adim, indclusters), Lmoments))[,
+     3:5]
> regLM15

        lcv       lca      lkur
1 0.1143714 0.1506529 0.1543626
2 0.1544286 0.1005249 0.1224144
3 0.1648291 0.1361256 0.1197992
4 0.1999977 0.1872976 0.1400214
```

If I calculate them with the method of Hosking and Wallis:

```
> for (i in 1:nclusters) {
+     print(regionalLmoments(D15adim[indclusters == i], cod15[indclusters ==
+         i])[3:5])
+ }

     lcvR      lcaR     lkurR
0.1156601 0.1550459 0.1489688
     lcvR      lcaR     lkurR
0.1557402 0.1021739 0.1096460
     lcvR      lcaR     lkurR
0.1680399 0.1406213 0.1180470
     lcvR      lcaR     lkurR
0.2032194 0.1938267 0.1438153
```

Plot of clusters (Figure 5):

```
> op <- par(mfrow = c(2, 2))
> plot(parameters15[c("Hm", "Ybar")], col = clusters, pch = clusters,
+     cex = 0.6, main = "Clusters in the space of classification variables",
+     cex.main = 1, font.main = 1)
> grid()
> points(tapply(parameters15["Hm"][, ], clusters, mean), tapply(parameters15["Ybar"][,
+     ], clusters, mean), col = c(1:nclusters), pch = c(1:nclusters))
> legend("topleft", paste("clust ", c(1:nclusters)), col = c(1:nclusters),
+     pch = c(1:nclusters), bty = "n")
> plot(parameters15[c("Xbar", "Ybar")], col = clusters, pch = clusters,
+     cex = 0.6, main = "Clusters in geographical space", cex.main = 1,
+     font.main = 1)
> grid()
> plot(LM15[, 4:3], pch = clusters, col = clusters, cex = 0.6,
+     main = "Clusters in L-moments space", cex.main = 1, font.main = 1)
> points(regLM15[, 2:1], col = c(1:nclusters), pch = c(1:nclusters))
```

```
> grid()
> plot(LM15[, 4:5], pch = clusters, col = clusters, cex = 0.6,
+     main = "Clusters in L-moments space", cex.main = 1, font.main = 1)
> points(regLM15[, 2:3], col = c(1:nclusters), pch = c(1:nclusters))
> grid()
> par(op)
```

Model selection (L-moments ratio diagram):

```
> Lmoment.ratio.diagram()
> points(regLM15[, 2:3], col = c(1:nclusters), pch = c(1:nclusters))
> legend("bottomleft", paste("clust ", c(1:nclusters)), col = c(1:nclusters),
+     pch = c(1:nclusters), bty = "n")
```

The points are around the Pearson type III distribution. If we apply the Anderson-Darling goodness-of-fit test, we obtain:

```
> for (i in 1:nclusters) {
+     GOFA2_P3 <- A2_GOFlaio(D15adim[indclusters == i], dist = "P3")
+     cat(paste("\np(A2) for Cluster ", i, ":\n", sep = ""))
+     print(GOFA2_P3)
+ }

p(A2) for Cluster 1:
       A2     p(A2)
0.3152134 0.5608133

p(A2) for Cluster 2:
       A2     p(A2)
0.2531199 0.3859892

p(A2) for Cluster 3:
       A2     p(A2)
0.5713187 0.9220134

p(A2) for Cluster 4:
       A2     p(A2)
0.6638851 0.9532153
```

For the 4-th cluster, the goodness of fit test is not passed with a 5% significance level.

Parameters of the Pearson type III distributions using the method of L-moments:

```
> paramgamma = NULL
> for (i in 1:nclusters) {
+     paramgamma[[i]] <- par.gamma(1, regLM15[i, 1], regLM15[i,
+         2])
+     cat(paste("\nCluster", i, ":\n"))
+     print(format(paramgamma[[i]][1:3]))
+ }

Cluster 1 :
          xi         beta         alfa
```

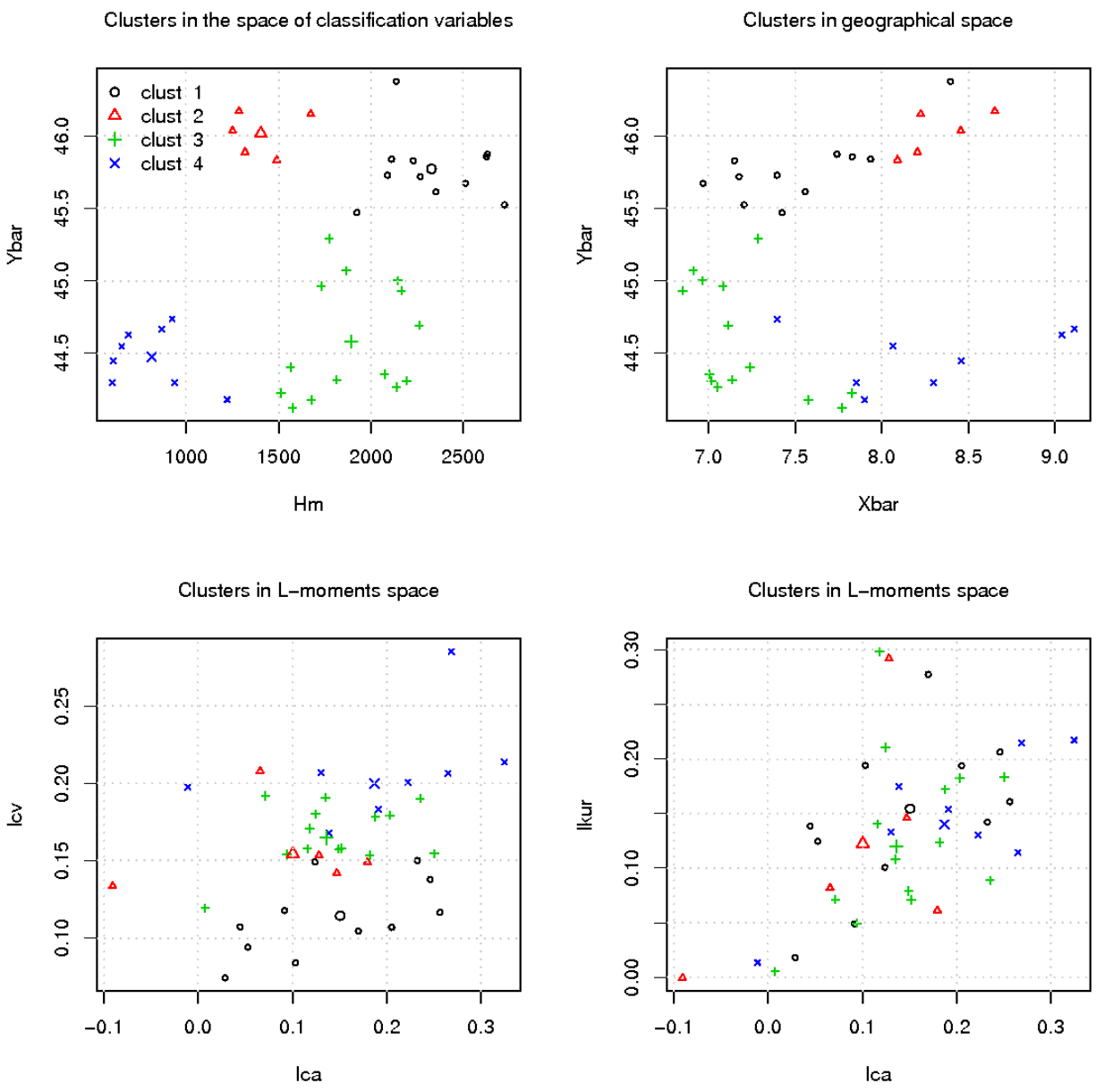

Figure 5: Clusters.

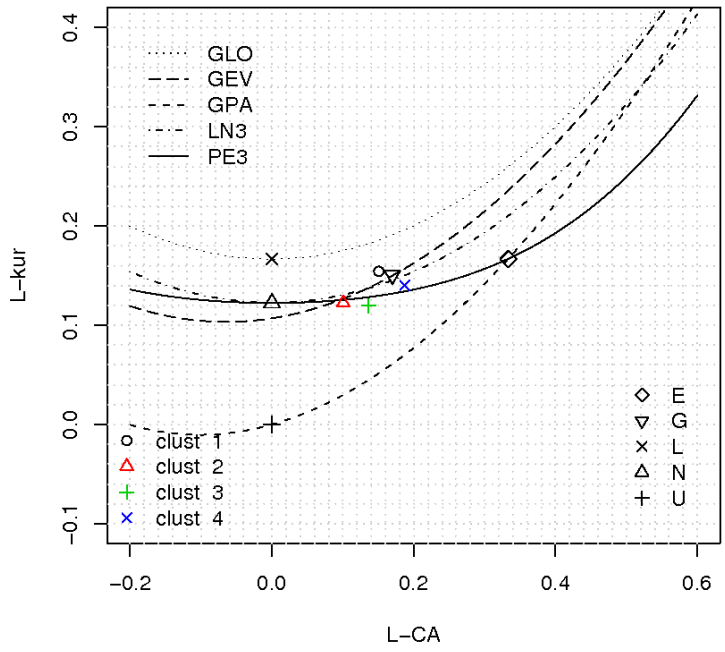

Figure 6: L-moments ratio diagram.

```
 "0.5457898" "0.09534035"   "4.764092"

Cluster 2 :
          xi         beta         alfa
"0.09842929" "0.08508402"   "10.59624"

Cluster 3 :
         xi        beta        alfa
"0.2800572" "0.1237543"  "5.817518"

Cluster 4 :
         xi        beta        alfa
"0.3496308" "0.2093341"  "3.106847"
```

Expressed in the other way:

```
> for (i in 1:nclusters) {
+     cat(paste("\nCluster", i, ":\n"))
+     print(format(par2mom.gamma(paramgamma[[i]]$xi, paramgamma[[i]]$beta,
+         paramgamma[[i]]$alfa)))
+ }

Cluster 1 :
          mu        sigma         gamm
  "4.816127" "0.07043515"   "2.707181"
```

```
Cluster 2 :
          mu        sigma         gamm
  "10.60462" "0.02669378"   "6.374819"

Cluster 3 :
          mu        sigma         gamm
  "5.852176" "0.06549131"   "3.779259"

Cluster 4 :
         mu       sigma        gamm
 "3.180037" "0.1237784"  "3.382401"
```

Regional growth-curves:

```
> op <- par(mfrow = c(2, 2))
> for (i in 1:nclusters) {
+     FF <- F.gamma(D15adim[indclusters == i], paramgamma[[i]]$xi,
+         paramgamma[[i]]$beta, paramgamma[[i]]$alfa)
+     regionalplotpos(D15adim[indclusters == i], cod15[indclusters ==
+         i], xlab = paste("cluster", i), main = "Empirical distributions",
+         cex.main = 1, font.main = 1)
+     lines(sort(D15adim[indclusters == i]), sort(FF))
+     nomi <- names(clusters)[clusters == i]
+     legend("bottomright", legend = nomi, pch = c(1:length(nomi)),
+         col = c(1:length(nomi)), bty = "n", cex = 0.9)
+ }
> par(op)
```

Regional growth-curves:

```
> op <- par(mfrow = c(2, 2))
> for (i in 1:nclusters) {
+     Fs <- seq(0.001, 0.999, by = 0.001)
+     regionalnormplot(D15adim[indclusters == i], cod15[indclusters ==
+         i], xlab = paste("cluster", i), main = "Empirical distributions",
+         cex.main = 1, font.main = 1)
+     normpoints(invF.gamma(Fs, paramgamma[[i]]$xi, paramgamma[[i]]$beta,
+         paramgamma[[i]]$alfa), type = "l")
+     nomi <- names(clusters)[clusters == i]
+     legend("bottomright", legend = nomi, pch = c(1:length(nomi)),
+         col = c(1:length(nomi)), bty = "n", cex = 0.9)
+ }
> par(op)
```

Comparison between regional growth-curves:

```
> spess = c(1, 1.5, 2, 1.3)
> Fs <- seq(0.001, 0.999, by = 0.001)
> lognormplot(D15adim, line = FALSE, type = "n", )
> for (i in 1:nclusters) {
+     qq <- invF.gamma(Fs, paramgamma[[i]]$xi, paramgamma[[i]]$beta,
```

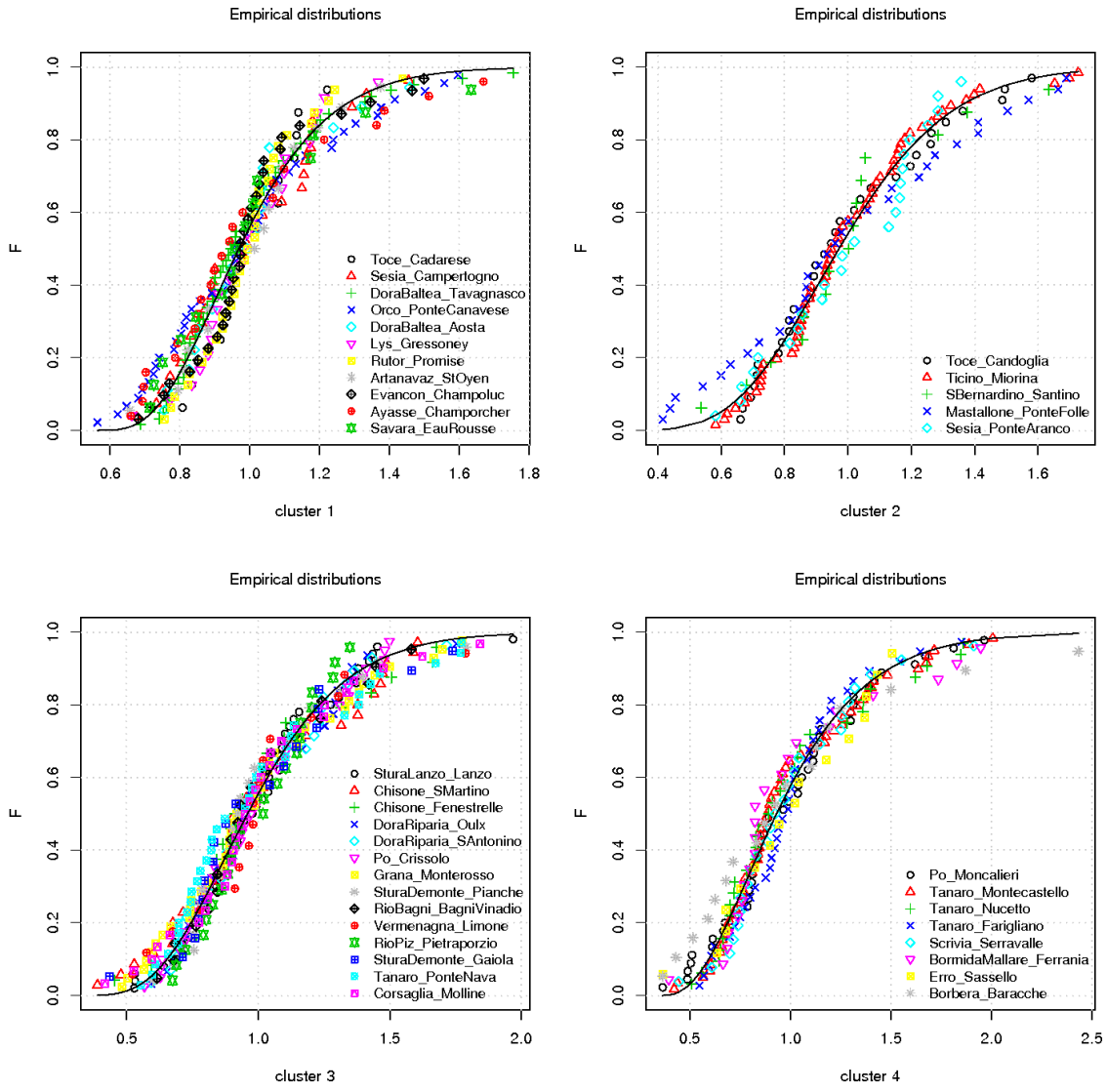

Figure 7: GrowthCurves1.png.

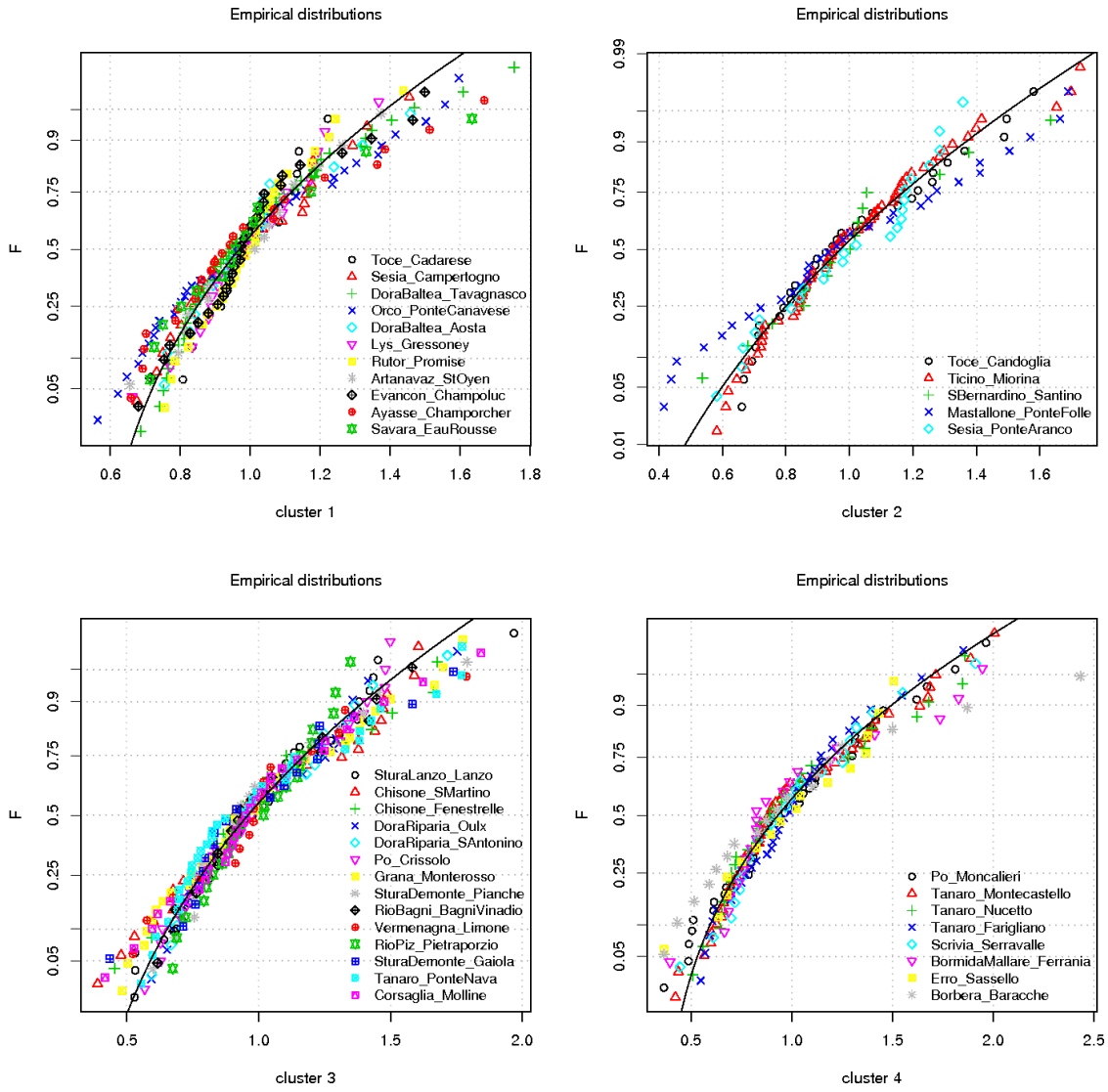

Figure 8: GrowthCurves2.png.

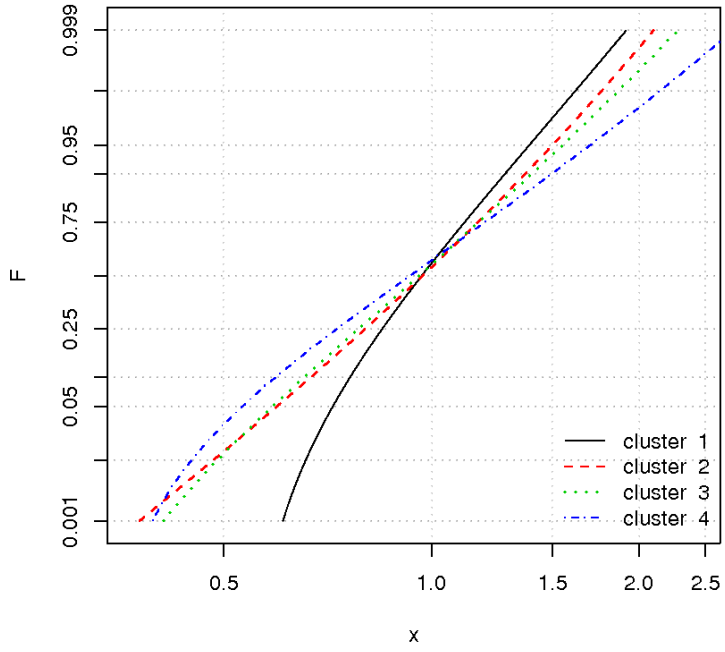

Figure 9: comparGrowthCurves.png.

```
+         paramgamma[[i]]$alfa)
+     normpoints(qq, type = "l", lty = i, col = i, lwd = spess[i])
+ }
> legend("bottomright", paste("cluster ", c(1:nclusters)), col = c(1:nclusters),
+     lty = c(1:nclusters), lwd = spess, bty = "n")
```

References

Brath, A., Camorani, G., and Castellarin, A. (2004). Una tecnica di stima regionale della curva di durata delle portate in bacini non strumentati. In *XXIX Convegno di Idraulica e Costruzioni Idrauliche*, volume 2, pages 391–398, Trento. Università di Trento.

Castellarin, A., Galeati, G., Brandimarte, L., Brath, A., and Montanari, A. (2004a). Regional flow-duration curve: realiability for ungauged basins. *Advances in Water Resources*, 27(10):953–965.

Castellarin, A., Vogel, R., and Brath, A. (2004b). A stochastic index flow model of flow duration curves. *Water Resources Research*, 40(3):W03104.

Claps, P. and Fiorentino, M. (1997). *Probabilistic Flow Duration Curvers for use in Environmental Planning and Management*, volume 2 (31) of *NATO-ASI series*, pages 255–266. Harmancioglu et al., Kluwer, Dordrecht, The Netherlands.

Claps, P. and Mancino, L. (2002). Impiego di classificazioni climatiche quantitative nell'analisi regionale del deflusso annuo. In *XXVIII Convegno di Idraulica e Costruzioni Idrauliche*, pages 169–178, Potenza. 16-19 settembre 2002.

Dalrymple, T. (1960). *Flood frequency analyses*, volume 1543-A of *Water Supply Paper*. U.S. Geological Survey, Reston, Va.

Ferraresi, M., Todini, E., and Franchini, M. (1988). Un metodo per la regionalizzazione dei deflussi medi. In *XXI Convegno di Idraulica*, L'Aquila.

Hosking, J. and Wallis, J. (1997). *Regional Frequency Analysis: An Approach Based on L-Moments*. Cambridge University Press.

Kottegoda, N. T. and Rosso, R. (1997). *Statistics, Probability, and Reliability for Civil and Environmental Engineers*. McGraw-Hill Companies, international edition.

Robson, A. and Reed, D. (1999). Statistical procedures for flood frequency estimation. In *Flood Estimation HandBook*, volume 3. Institute of Hydrology Crowmarsh Gifford, Wallingford, Oxfordshire.

Rossi, F. and Villani, P. (1995). Valutazione delle piene in campania. Technical report, CNR-GNDCI e Dipartimento di Ingegneria Civile dell'Università di Salerno, Salerno.

Viglione, A. (2007). *Metodi statistici non-supervised per la stima di grandezze idrologiche in siti non strumentati*. PhD thesis, Politecnico di Torino.

Viglione, A., Claps, P., and Laio, F. (2006). Utilizzo di criteri di prossimità nell'analisi regionale del deflusso annuo. In *XXX Convegno di Idraulica e Costruzioni Idrauliche - IDRA 2006*.

Viglione, A., Laio, F., and Claps, P. (2007). A comparison of homogeneity tests for regional frequency analysis. *Water Resources Research*, 43(3).

Vogel, R. and Wilson, I. (1996). Probability distribution of annual maximum, mean, and minimum streamflows in the united states. *Journal of Hydrologic Engineering*, 1(2):69–76.