Multivariate Methods

Dave Lorenz

July 14, 2015

These examples demonstrate some of the functions that can be used to prepared water-quality data for multivariate analysis that are available in the `smwrQW` package. The examples in this vignette use the DDT in fish dataset (table 13.1 in Helsel 2012). The examples in this vignette use the function `as.lcens` to convert those data to a form used by the functions demonstrated; the class "lcens" is most appropriate for these data as they are only left-censored and have only the value and an indicator of censoring. The functions demonstrated in these examples will also accept data of class "qw." The R code following this paragraph creates the data and recodes the water-quality data to class "lcens." There are many contributed pacakges that can be used for multivariate anslysis; the examples in this vignette use only packages supplied with base R and are intended only to demonstrate the multivariate functionality of the smwrQW pacakge.

```
> # Load the smwrQW package
> library(smwrQW)
> # Create the dataset omitting Age, code as character
> DDT <- data.frame(Site=1:32,
+   opDDD=c("<5","<5","5.3","<5","<5","<5","<5","<5",
+           "<5","5.1","<5","<5","<5","5.1","<5","<5",
+           "9","<5","9.8","<5","<5","5.1","8","<5",
+           "<5","<5","5.7","<5","<5","<5","<5","<5"),
+   ppDDD=c("<5","42","38","12","<5","<5","14","15",
+           "12","39","5.7","9.4","18","27","10","7.6",
+           "46","22","41","13","26","24","100","<5",
+           "<5","<5","27","15","20","22","31","15"),
+   opDDE=c("<5","8.4","<5","<5","<5","<5","<5","<5",
+           "<5","<5","<5","<5","<5","<5","<5","<5",
+           "<5","<5","6.9","<5","<5","250","8","<5",
+           "<5","<5","<5","6","7.5","<5","<5","<5"),
+   ppDDE=c("14","130","250","57","16","<5","52","48",
+           "110","100","87","53","210","140","24","15",
+           "110","51","50","66","110","38","160","23",
+           "17","16","140","8.1","22","190","42","23"),
+   opDDT=c("<5","<5","<5","<5","<5","<5","<5","<5",
+           "<5","<5","18","<5","<5","<5","<5","<5",
+           "<5","<5","<5","<5","<5","<5","<5","<5",
+           "<5","<5","<5","<5","<5","5.2","<5","<5"),
+   ppDDT=c("<5","31","11","<5","<5","<5","14","<5",
+           "20","24","<5","7.3","30","33","5.8","10",
+           "21","7.4","11","8.6","26","11","<5","<5",
+           "<5","<5","14","<5","6.4","27","25","5.6"),
+   stringsAsFactors=FALSE)
> # Convert concentrations to class "lcens"
> DDT <- transform(DDT, opDDD=as.lcens(opDDD), ppDDD=as.lcens(ppDDD),
+   opDDE=as.lcens(opDDE), ppDDE=as.lcens(ppDDE),
+   opDDT=as.lcens(opDDT), ppDDT=as.lcens(ppDDT))
> # Load the libraries
> library(cluster)
```

1 Binary Methods

The most conceptually simple method to prepare censored data for multivariate analysis recodes values as presence/absence or as the numerical values 1 and 0, respectively. The values are typically recoded based on the largest reporting limit, which works well for moderately and highly censored data. The recoding criterion can be set to any value greater than the largest reporting level, which can be useful for un- or lightly censored data.

The recoded presence/absence, often referred to as 0/1 (reversing the sense of presence/absence), data usually converted into similarity or dissimilarity meterics. Helsel (2012) points out that many of the techniques for computing the similarity or dissimilarity meterics were developed for the biological sciences and are not appropriate for water-quality data because they downweight absence/absence pairs. The simple matching coefficient for computing dissimilarities described by Helsel (2012) can be computed using the `daisy` function in the `cluster` package and is illustrated in the R code following this paragraph. The distances between the first 10 sites are shown in the last line of R code, for illustration only. For example the 0.5 dissimilarity between site 1 and 2 indicates that one-half the values are the same (opDDD, ppDDE, and opDDT) and the other one-half are different.

```
> # Compute the 0/1 values
> DDT01 <- with(DDT, code01(opDDD, ppDDD, opDDE, ppDDE, opDDT, ppDDT))
> # Compute the dissimilarities using simple matching
> DDT.diss <- daisy(DDT01, metric="gower", type=list(symm=1:6))
> # Show the first 10 rows of the 0/1 data and the distances
> head(DDT01, 10)

   opDDD ppDDD opDDE ppDDE opDDT ppDDT
1      0     0     0     1     0     0
2      0     1     1     1     0     1
3      1     1     0     1     0     1
4      0     1     0     1     0     0
5      0     0     0     1     0     0
6      0     0     0     0     0     0
7      0     1     0     1     0     1
8      0     1     0     1     0     0
9      0     1     0     1     0     1
10     1     1     0     1     0     1

> print(as.dist(as.matrix(DDT.diss)[1:10,1:10]), digits=4)

        1      2      3      4      5      6      7      8      9
2  0.5000
3  0.5000 0.3333
4  0.1667 0.3333 0.3333
5  0.0000 0.5000 0.5000 0.1667
6  0.1667 0.6667 0.6667 0.3333 0.1667
7  0.3333 0.1667 0.1667 0.1667 0.3333 0.5000
8  0.1667 0.3333 0.3333 0.0000 0.1667 0.3333 0.1667
9  0.3333 0.1667 0.1667 0.1667 0.3333 0.5000 0.0000 0.1667
10 0.5000 0.3333 0.0000 0.3333 0.5000 0.6667 0.1667 0.3333 0.1667
```

The distance matrix can be used to construct a hieracical cluster analysis. The `stats` package in base R has the `hclust` function and the `cluster` pacakge, supplied with base R has the `agnes` function that has more options than `hclust`. The example following this paragrah uses the `hclust` function. The details of the cluster merging sequence are printed to help understand the dendrogram. The cluster merging sequence has three columns, the first column is the left cluster number, the second column is the right cluster number and the third column is the height or distance between those cluster. For the left and right columns, negative values refer to the object number and positive values refer to the row number in the matrix. In this example the first cluster is formed from the merger of object 1 and 5 and the second cluster is formed from the merger of object 24 and the fist cluster. In both cases, the distances are 0 (no difference between the values). That pattern can be seen in the dendrogram.

```
> # The cluster analysis
> DDT.hclust <- hclust(DDT.diss, method="average")
> # Details for the cluster merging sequence
> cbind(DDT.hclust$merge, DDT.hclust$height)

      [,1] [,2]      [,3]
 [1,]   -1   -5 0.0000000
 [2,]  -24    1 0.0000000
 [3,]  -25    2 0.0000000
 [4,]  -26    3 0.0000000
 [5,]   -2  -29 0.0000000
 [6,]   -3  -10 0.0000000
 [7,]  -14    6 0.0000000
 [8,]  -17    7 0.0000000
 [9,]  -27    8 0.0000000
[10,]   -4   -8 0.0000000
[11,]   -7   -9 0.0000000
[12,]  -12   11 0.0000000
[13,]  -13   12 0.0000000
[14,]  -15   13 0.0000000
[15,]  -16   14 0.0000000
[16,]  -18   15 0.0000000
[17,]  -20   16 0.0000000
[18,]  -21   17 0.0000000
[19,]  -31   18 0.0000000
[20,]  -32   19 0.0000000
[21,]  -19  -22 0.0000000
[22,]    4   10 0.1666667
[23,]    5   21 0.1666667
[24,]  -11  -30 0.1666667
[25,]  -23  -28 0.1666667
[26,]    9   20 0.1666667
[27,]   -6   22 0.2142857
[28,]   23   26 0.2500000
[29,]   24   28 0.3416667
[30,]   25   29 0.3939394
[31,]   27   30 0.4027778

> # Plot the dendrogram
> setSweave("graph01", 6 ,6)
> # Create the graph,
> dendGram(DDT.hclust, ytitle="Average Merging Dissimilarity")
> graphics.off()
```

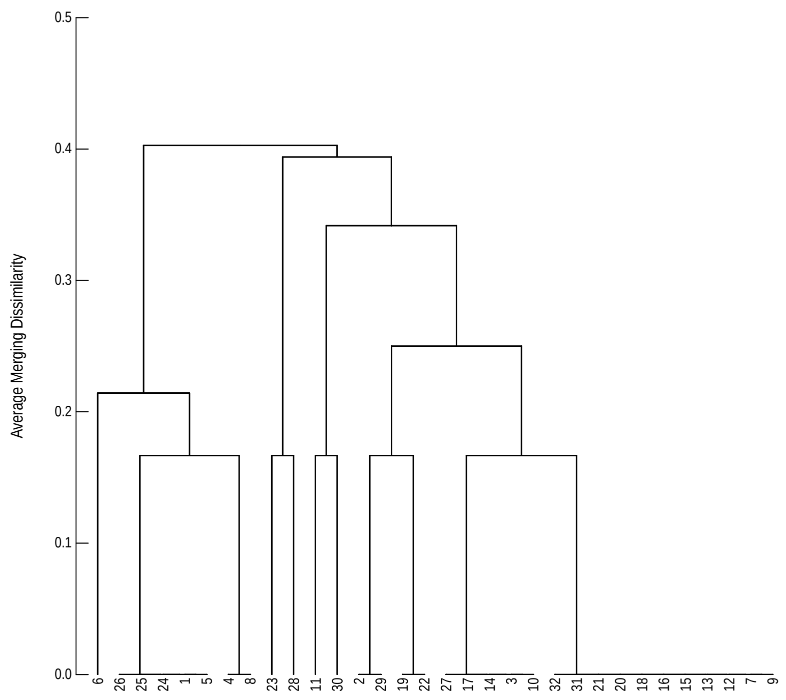

Figure 1. The dendrogram from the binary method of distance calculation.

2 Ordinal Methods

Gehan's u-score, described in Helsel (2012) is an affine transform of the rank for uncensored and single-reporting level censored values. The equation is u = (r-(n+1)/2)*2 or r = u/2 + (n+1)/2, where u is the u-score, r is the rank, and n is the number of observations. The advantage for the u-score is that is computable for multiple reporting levels and multiply censored data. The exampes in this section use the u-score computed by the `codeU` function.

The u-scores can be used directly in a principal component analysis or used to compute distances for a cluster analysis. The code following this paragraph computes the u-scores and the computes the Euclidean distance between sites.

```
> # Compute the u scores
> DDTU <- with(DDT, codeU(opDDD, ppDDD, opDDE, ppDDE, opDDT, ppDDT))
> # Compute the dissimilarities using simple matching
> DDTU.dist <- dist(DDTU)
> # Show the first 8 rows of the 0/1 data and the distances
> head(DDTU, 8)

  opDDD ppDDD opDDE ppDDE opDDT ppDDT
1    -8   -26    -6   -27    -2   -21
2    -8    27    29    19    -2    29
3    23    21    -6    31    -2     7
4    -8   -10    -6     5    -2   -21
5    -8   -26    -6   -22    -2   -21
6    -8   -26    -6   -31    -2   -21
7    -8    -5    -6     1    -2    12
8    -8    -1    -6    -5    -2   -21

> print(as.dist(as.matrix(DDTU.dist)[1:8,1:8]), digits=4)

      1     2     3     4     5     6     7
2 93.01
3 85.55 53.39
4 35.78 72.73 58.15
5  5.00 90.64 82.24 31.38
6  4.00 95.05 88.31 39.40  9.00
7 48.10 53.50 50.62 33.62 45.38 50.54
8 33.30 71.31 59.37 13.45 30.23 36.07 33.78
```

Hierarchical cluster analysis can be computed using either the `hclust` or `agnes` function. The R code following this paragraph uses the `hclust` function to compute the cluster analysis and uses the "average" method for merging clusters. For this example, the cluster merging sequence is not shown as there are no multiple obervations separated by a distance of 0. But sites 5 and 26 have identical u-scores.

```
> # The cluster analysis
> DDTU.hclust <- hclust(DDTU.dist, method="average")
> # Plot the dendrogram
> setSweave("graph02", 6 ,6)
> # Create the graph,
> dendGram(DDTU.hclust, ytitle="Average Merging Distance",
+          xtitle="Site Number")
> graphics.off()
```

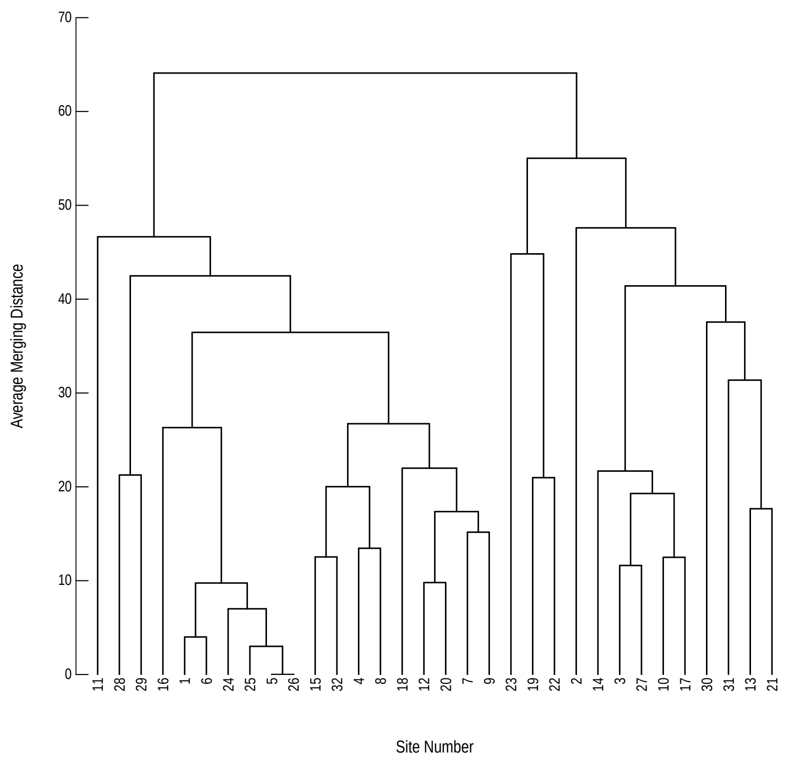

Figure 2. The dendrogram from the ordinal method of distance calculation.

Principal component analysis can be computed using either the `princomp` or `prcomp` function. The R code following this paragraph uses the `princomp` function to compute the analysis, but the `prcomp` function has more flexibility and numerical stability.

```
> # The PCA.
> DDTU.pca <- princomp(DDTU, cor=TRUE)
> print(DDTU.pca)

Call:
princomp(x = DDTU, cor = TRUE)

Standard deviations:
   Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6
```

```
1.6665666 1.1735949 0.9383574 0.7647978 0.5003405 0.3598053

 6  variables and  32 observations.

> print(loadings(DDTU.pca), digits=4, cutoff=0)

Loadings:
      Comp.1  Comp.2  Comp.3  Comp.4  Comp.5  Comp.6 
opDDD -0.4469  0.2636  0.0291  0.7251 -0.3496  0.2863
ppDDD -0.5657  0.1241  0.0368 -0.0793 -0.0085 -0.8105
opDDE -0.2105  0.5796  0.5806 -0.4148  0.1410  0.3012
ppDDE -0.4793 -0.3922 -0.0211  0.1013  0.7350  0.2559
opDDT  0.0065 -0.5902  0.7509  0.0808 -0.2773 -0.0659
ppDDT -0.4540 -0.2774 -0.3106 -0.5284 -0.4906  0.3172

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
SS loadings    1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
Proportion Var 0.1667 0.1667 0.1667 0.1667 0.1667 0.1667
Cumulative Var 0.1667 0.3333 0.5000 0.6667 0.8333 1.0000

> # Plot the PCA
> setSweave("graph03", 6 ,6)
> # Create the graph,
> AA.pl <- biPlot(DDTU.pca, Scale="distance", range.factor=1.1,
+      obsPlotLabels=list(labels="none"))
> # Manually assign label direction to reduce ambiguity
> dir <- c("N","NE","NW","S","N","E","W","NE","E","E","NE","SW","NE","E","S","S",
+   "N","NE","NE","NE","NE","NE","NW","SE","NW","SE","N","NE","NE","NE","NE","NW")
> with(AA.pl, labelPoints(x, y, labels=1:32, dir=dir, offset=0.5, current=AA.pl))
> graphics.off()
```

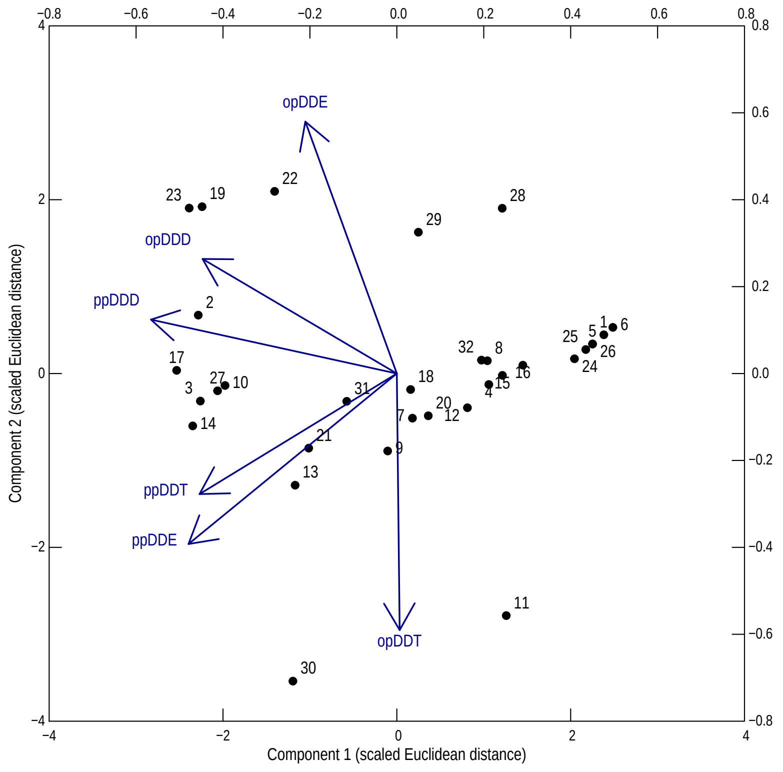

Figure 3. The biplot of the principal component analyses of the u-sores of the DDT data.

The x-axis (Component 1) represents the total concentration, but primarily the pp-type compunds. Sites on the left-hand side have larger concentrations than those on the right-hand side. The y-axis (Component 2) represents whether opDDE or opDDT has the larger concentration.

3 Imputation Methods

For relatively small percentages of left-censored water-quality data, substitute values can be estimated for left-censored values based on the overall structure of the data and the complete data used in multivariate procedures. The methods for estimating sensible substitution strategies for left-censored values are described by Palarea-Albaladejo (2013). The zCompositions package implements those methods. The functions `imputeLessThans` and `mImputeLessThans` in the `smwrQW` package can be used to estimate values for data of class ”qw” or ”lcens.”

The example in this section uses an altered version of the DDT dataset. Estimated values for the left-censored were synthesized to approximately maintain the correlations structure of the u-sores and the data were recensored at 2. However, the intent of the altered values was to illustrate the process of estiamting values rather than produce a similar analysis. The R code immediately following this paragraph creates those data.

```
> # Create the altered dataset, code as character
> DDTalt <- data.frame(Site=1:32,
+   opDDD=c("<2","4","5.3","2","<2","<2","2.3","2.3",
+           "2.1","5.1","<2","<2","2.8","5.1","<2","<2",
+           "9","2.8","9.8","2.1","3.3","5.1","8","<2",
+           "<2","<2","5.7","2.1","2.6","3","3.5","2.2"),
+   ppDDD=c("3.1","42","38","12","3.1","2.4","14","15",
+           "12","39","5.7","9.4","18","27","10","7.6",
+           "46","22","41","13","26","24","100","3.2",
+           "3.1","3.1","27","15","20","22","31","15"),
+   opDDE=c("2.1","8.4","<2","<2","<2","4.9","<2","<2",
+           "<2","<2","<2","<2","<2","<2","<2","<2",
+           "<2","<2","6.9","<2","<2","250","8","<2",
+           "<2","<2","<2","6","7.5","<2","<2","<2"),
+   ppDDE=c("14","130","250","57","16","3.4","52","48",
+           "110","100","87","53","210","140","24","15",
+           "110","51","50","66","110","38","160","23",
+           "17","16","140","8.1","22","190","42","23"),
+   opDDT=c("<2","3.9","2.4","2","<2","<2","2.4","2",
+           "2.5","2.6","18","2.2","2.6","2.7","2.1","2.2",
+           "4.9","2.3","2.4","2.3","2.6","2.4","2.3","<2",
+           "<2","<2","2.5","2","2.2","5.2","2.6","2.2"),
+   ppDDT=c("2.2","31","11","3.2","2.3","<2","14","3.2",
+           "20","24","3","7.3","30","33","5.8","10",
+           "21","7.4","11","8.6","26","11","3.9","2.5",
+           "2.3","2.3","14","2.6","6.4","27","25","5.6"),
+   stringsAsFactors=FALSE)
> # Convert concentrations to class "lcens"
> DDTalt <- transform(DDTalt, opDDD=as.lcens(opDDD), ppDDD=as.lcens(ppDDD),
+   opDDE=as.lcens(opDDE), ppDDE=as.lcens(ppDDE),
+   opDDT=as.lcens(opDDT), ppDDT=as.lcens(ppDDT))
```

The `imputeLessThans` function can estimate sensible substitution values for left-censored values. The Rcode immediately following this paragraph uses that function to estiamte complete data for the DDTal dataset. Those data are used in a principal component analysis, `prcomp` and the biplot is created.

```
> # Impute the less-than values, multRepl required because opDDE is heavily censored
> DDTaltImp <- with(DDTalt, imputeLessThans(opDDD, ppDDD, opDDE, ppDDE, opDDT, ppDDT,
+   initial="multRepl"))
> # Print alternate and imputed values
> head(DDTalt)

  Site opDDD ppDDD opDDE ppDDE opDDT ppDDT
1    1    <2   3.1   2.1    14    <2   2.2
2    2     4    42   8.4   130   3.9    31
3    3   5.3    38    <2   250   2.4    11
4    4     2    12    <2    57     2   3.2
5    5    <2   3.1    <2    16    <2   2.3
6    6    <2   2.4   4.9   3.4    <2    <2

> head(DDTaltImp)

      opDDD ppDDD        opDDE ppDDE     opDDT     ppDDT
1 0.8264093   3.1 2.100000e+00  14.0 0.6707273  2.200000
2 4.0000000  42.0 8.400000e+00 130.0 3.9000000 31.000000
3 5.3000000  38.0 3.606025e-04 250.0 2.4000000 11.000000
4 2.0000000  12.0 5.418405e-04  57.0 2.0000000  3.200000
5 0.6242727   3.1 9.279211e-05  16.0 0.5257752  2.300000
6 0.9788568   2.4 4.900000e+00   3.4 0.7194524  1.109294

> # The PCA.
> DDTaltImp.pca <- prcomp(log(DDTaltImp), center=TRUE, scale=TRUE)
> print(DDTaltImp.pca)

Standard deviations:
[1] 1.9420539 1.1617123 0.6774500 0.5222167 0.3201583 0.2114263

Rotation:
             PC1         PC2        PC3        PC4          PC5          PC6
opDDD 0.47258000 -0.27859713  0.1047690 -0.1760945  0.417454523 -0.694836443
ppDDD 0.47665081 -0.23195432  0.2110143 -0.2436388  0.332319115  0.710407145
opDDE 0.01771871 -0.84634903 -0.0259327  0.1287362 -0.515857999  0.004936371
ppDDE 0.44149195  0.33085708  0.1181277 -0.4924028 -0.657292076 -0.084681158
opDDT 0.40585607  0.06218435 -0.8914845  0.1771177  0.001630072  0.072774226
ppDDT 0.43534343  0.19733437  0.3675938  0.7869096 -0.130960763 -0.005714139

> # Plot the PCA
> setSweave("graph04", 6 ,6)
> # Create the graph,
> biPlot(DDTaltImp.pca, Scale="distance", range.factor=1.1)
> graphics.off()
```

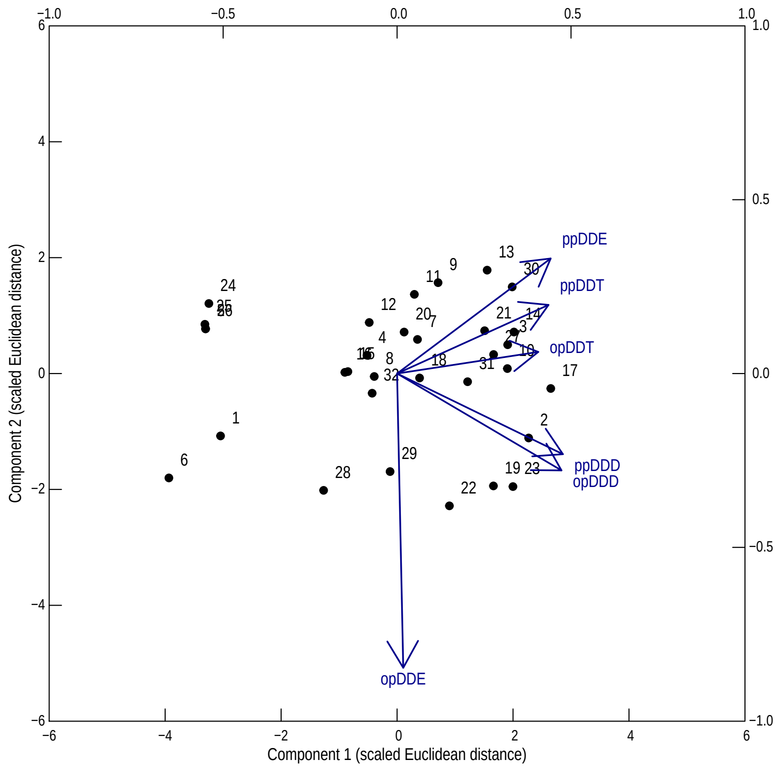

Figure 4. The biplot of the principal component analyses of the of the DDTalt data.

The x-axis (Component 1) represents the total concentration. Sites on the right-hand side have larger concentrations than those on the left-hand side. The y-axis (Component 2) represents the cponcetration of opDDE. Sites with negative scores have larger concentrations than those with positive scores.

References

H12 Helsel, D.R. 2012, Statistics for Censored Environmental Data Using Minitab and R: New York, Wiley, 324 p. HH Helsel, D.R., and Hirsch, R.M., 2002, Statistical methods in water resources: U.S. Geological Survey Techniques of Water-Resources Investigations, book 4, chap. A3, 522 p. DL Lorenz, D.L., 2015, smwrStats–An R package for the analysis of hydrologic data, Version 0.7.3: U.S. Geological Survey Open File Report, ? p. PA Palarea-Albaladejo J. and Martin-Fernandez J.A., 2013, Values below detection limit in compositional chemical data: Analytica Chimica Acta 2013, v. 764, p. 32-43. accessed July 6, 2015 at http://dx.doi.org/10.1016/j.aca.2012.12.029.