Spider-Man, the Child, and the Trickster[1] – Archetypal Analysis in R

Manuel J. A. Eugster
Ludwig-Maximilians-Universität München

Friedrich Leisch
Ludwig-Maximilians-Universität München

Abstract

This article is a (slightly) modified version of Eugster and Leisch (2008).

Archetypal analysis has the aim to represent observations in a multivariate data set as convex combinations of extremal points. This approach was introduced by Cutler and Breiman (1994); they defined the concrete problem, laid out the theoretical foundations and presented an algorithm written in Fortran, which is available on request. In this paper we present the R package **archetypes** which is available on the Comprehensive R Archive Network. The package provides an implementation of the archetypal analysis algorithm within R and different exploratory tools to analyze the algorithm during its execution and its final result. The application of the package is demonstrated on two examples.

Keywords: archetypal analysis, convex hull, R.

1. Introduction

The Merriam-Webster Online Dictionary (2008) defines an archetype as *the original pattern or model of which all things of the same type are representations or copies.* Then, the aim of the archetypal analysis is to find "pure types", the archetypes, within a set defined in a specific context. The concept of archetypes is used in many different areas, the set can be defined in terms of literature, philosophy, psychology and also statistics. Here, the concrete problem is to find a few, not necessarily observed, points (archetypes) in a set of multivariate observations such that all the data can be well represented as convex combinations of the archetypes.

In statistics archetypal analysis was first introduced by Cutler and Breiman (1994). In their paper they laid out the theoretical foundations, defined the concrete problem as a nonlinear least squares problem and presented an alternating minimizing algorithm to solve it. It has found applications in different areas, with recently grown popularity in economics, e.g., Li, Wang, Louviere, and Carson (2003) and Porzio, Ragozini, and Vistocc (2006). In spite of the rising interest in this computer-intensive but numerically sensitive method, no "easy-to-use" and freely available software package has been developed yet; the only implementation is the original Fortran code by Cutler and Breiman (1994) which is available upon request only. In this paper we present the software package **archetypes** within the R statistical environment (R Development Core Team 2008) which provides an implementation of the archetypal analysis

[1] These are examples of archetypes in different contexts; see Wikipedia (2008).

algorithm. Additionally, the package provides exploratory tools to visualize the algorithm during the minimization steps and its final result. The newest released version of **archetypes** is always available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=archetypes`.

The paper is organized as follows: In Section 2 we outline the archetypal analysis with its different conceptual parts. We present the theoretical background as far as we need it for a sound introduction of our implementation; for a complete explanation we refer to the original paper. Section 3 demonstrates how to use **archetypes** based on a simple artificial data set, with details about numerical problems and the behavior of the algorithm. In Section 4 we show a real word example – the archetypes of human skeletal diameter measurements. Section 5 concludes the article with future investigations.

2. Archetypal analysis

Given is an $n \times m$ matrix X representing a multivariate data set with n observations and m attributes. For a given k the archetypal analysis finds the matrix Z of k m-dimensional archetypes according to the two fundamentals:

(1) The data are best approximated by combinations of the archetypes, i.e., they minimize

$$\text{RSS} = \|X - Z^T * \alpha\|_2$$

with α, the coefficients of the archetypes, a $k \times n$ matrix; the elements are required to be greater equal 0 and their sum must be 1, i.e., $\sum_{j=1}^{k} \alpha_{ij} = 1$ with $\alpha_{ij} \geq 0$ and $i = 1, \ldots, n$.

(2) The archetypes are mixtures of the data points:

$$Z = X^T * \beta$$

with β, the coefficients of the data set, a $n \times k$ matrix where the elements are required to be greater equal 0 and their sum must be 1, i.e., $\sum_{i=1}^{n} \beta_{ji} = 1$ with $\beta_{ji} \geq 0$ and $j = 1, \ldots, k$.

These two fundamentals define the basic principles of the algorithm: it alternates between finding the best α for given archetypes Z and finding the best archetypes Z for given α; at each step several convex least squares problems are solved, the overall RSS is reduced successively.

With a view to the implementation, the algorithm consists of the following steps:

Given the number of archetypes k:

1. Data preparation and initialization: scale data, add a dummy row (see below) and initialize β in a way that the the constraints are fulfilled to calculate the starting archetypes Z.

2. Loop until RSS reduction is sufficiently small or the number of maximum iterations is reached:

2.1. Find best α for the given set of archetypes Z: solve n convex least squares problems $(i = 1, \dots, n)$

$$\min_{\alpha_i} \frac{1}{2} \|Z * \alpha_i - X_i\|_2 \text{ subject to } \alpha_i \geq 0 \text{ and } \sum_{j=1}^{k} \alpha_{ij} = 1,$$

2.2. Recalculate archetypes $\tilde{Z}$: solve system of linear equations $X = \alpha * \tilde{Z}^T$.

2.3. Find best β for the given set of archetypes $\tilde{Z}$: solve k convex least squares problems $(j = 1, \dots, k)$

$$\min_{\beta_j} \frac{1}{2} \|X * \beta_j - \tilde{Z}_j\|_2 \text{ subject to } \beta_j \geq 0 \text{ and } \sum_{i=1}^{n} \beta_{ji} = 1,$$

2.4. Recalculate archetypes Z: $Z = X * \beta$.

2.5. Calculate residual sum of squares RSS.

3. Post-processing: remove dummy row and rescale archetypes.

The algorithm has to deal with several numerical problems, i.e. systems of linear equations and convex least squares problems. In the following we explain each step in detail.

Solving the convex least squares problems: In Step 2.1 and 2.3 several convex least squares problems have to be solved. Cutler and Breiman (1994) use a penalized version of the non-negative least squares algorithm by Lawson and Hanson (1974). The penalization is done by adding an extra observation, *the dummy row*, to X containing a "huge" value M at each element (see, e.g., Luenberger 1984). The hugeness of the value M varies from problem to problem and thus can be seen as a hyperparameter of the algorithm. Default value in the package is 200.

Solving the system of linear equations: In Step 2.2 the system of linear equations

$$\tilde{Z} = \alpha^{-1} * X$$

has to be solved. A lot of methods exist, one approach is the Moore-Penrose pseudoinverse which provides an approximated unique solution by a least square approach: given the pseudoinverse α^+ of α,

$$\tilde{Z} = \alpha^+ * X,$$

is solved. Another approach is the usage of QR decomposition: $\alpha = QR$, where Q is an orthogonal and R an upper triangular matrix, then

$$\tilde{Z} = Q^T * X * R^{-1},$$

is sovled. Default approach in the package is the QR decomposition using the `solve()` function.

Calculating the residual sum of squares: In Step 2.5 the RSS is calculated. It uses the spectral norm (see, e.g., Golub and Loan 1996)). The spectral norm of a matrix X is the largest singular value of X or the square root of the largest eigenvalue of $X^H X$,

$$\|X\|_2 = \sqrt{\lambda_{max}(X^H X)},$$

where X^H is the conjugate transpose of X.

Avoiding local minima: Cutler and Breiman (1994) shows that the algorithm converged in all cases, but not necessarily to a global minimum. Hence, the algorithm should be started several times with different initial archetypes. It is important that these are not too close together, this can cause slow convergence or convergence to a local minimum.

Choosing the correct number of archetypes: As in many cases there is no rule for the correct number of archetypes k. A simple method the determine the value of k is to run the algorithm for different numbers of k and use the "elbow criterion" on the RSS where a "flattening" of the curve indicates the correct value of k.

Approximation of the convex hull: Through the definition of the problem, archetypes lie on the boundary of the convex hull of the data. Let N be the number of data points which define the boundary of the convex hull, then Cutler and Breiman (1994) showed: if $1 < k < N$, there are k archetypes on the boundary which minimize RSS; if $k = N$, exactly the data points which define the convex hull are the archetypes with RSS $= 0$; and if $k = 1$, the sample mean minimizes the RSS. In practice, these theoretical results can not always be achieved as we will see in the following two sections.

3. Using archetypes

The package is loaded within R using the `library()` or `require()` command:

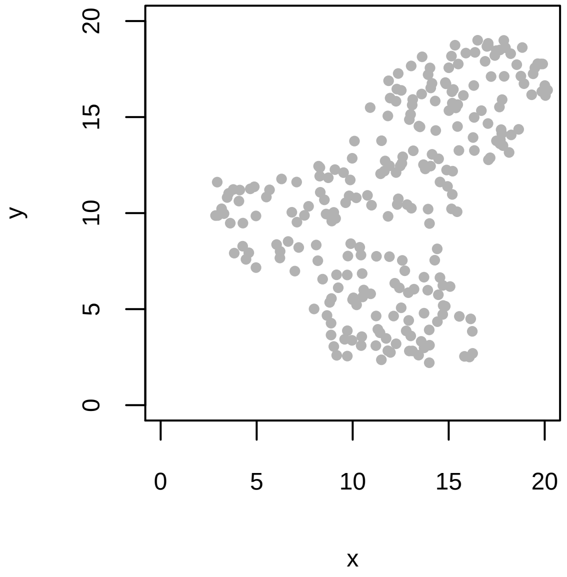
```
> library(archetypes)
```

```
Loading required package: nnls
```

It requires the packages **nnls** (Mullen and van Stokkum 2007) for solving the convex least square problems.

We use a simple artificial two-dimensional data set to explain the usage of the implementation, and the behavior of the archetypal analysis at all. The advantage is that we can imagine the results and simply visualize them, Section 4 then shows a more realistic example. Note that in the following the plot commands do not produce exactly the shown graphics concerning primitives coloring, width, etc.; due to readability we have reduced the presented commands to the significant arguments.

```
> data(toy)
> plot(toy)
```



Data set `toy` consists of the two attributes x and y, and 250 observations. According to the shape of the data, it seems to be a good idea to apply archetypal analysis with $k = 3$ archetypes.

```
> set.seed(1986)
> a <- archetypes(toy, 3)

1: rss = 0.02177873, improvement = 0.05689221
2: rss = 0.01411290, improvement = 0.00766583
3: rss = 0.01101285, improvement = 0.00310005
4: rss = 0.00915121, improvement = 0.00186164
5: rss = 0.00790619, improvement = 0.00124502
6: rss = 0.00741714, improvement = 0.00048906
7: rss = 0.00756394, improvement = -0.00014681
```

During the fit, the function reports its improvement and stops after a maximum number of iterations (default is `maxIterations = 100`) or if the improvement is less than a defined value (default is `minImprovement = sqrt(.Machine$double.eps)`). As basis for our further research, the implementation is a flexible framework where the problem solving mechanisms of the individual steps can be interchanged. The default values are the "original ones" described in the previous section (`family = archetypesFamily()`). The result is a S3 `archetypes` object,

```
> a

Archetypes object

archetypes(data = toy, k = 3)

Convergence after 7 iterations
with RSS = 0.007563943.
```

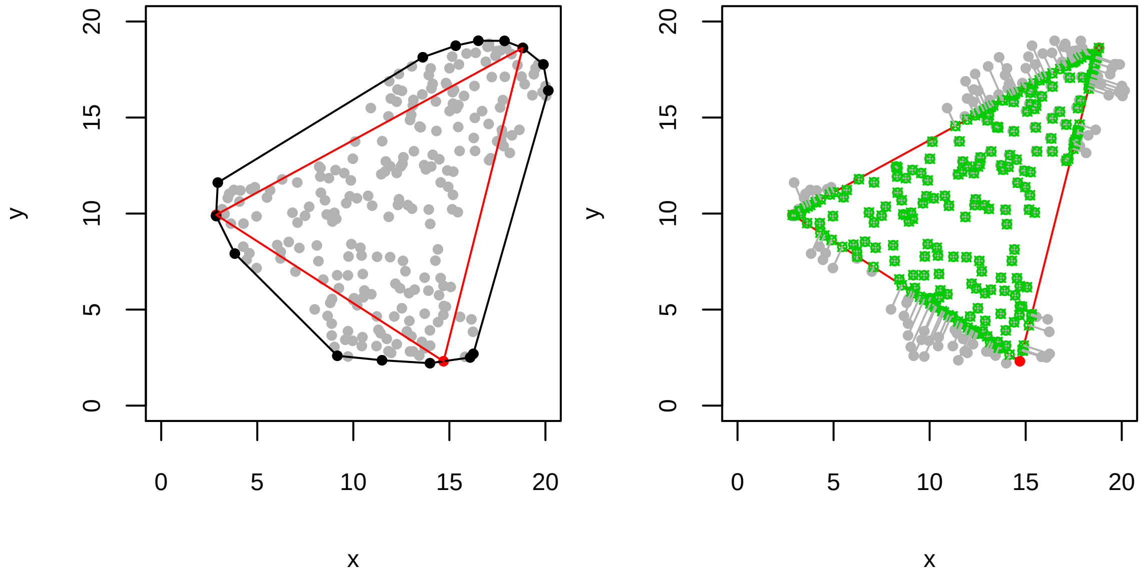
containing the three final archetypes:

```
> atypes(a)
```

```
             x         y
[1,] 14.696091  2.310303
[2,]  2.860579  9.935227
[3,] 18.810086 18.629479
```

The `plot()` function visualizes archetypes for two-dimensional data sets; for higher-dimensional data sets parallel coordinates are used.

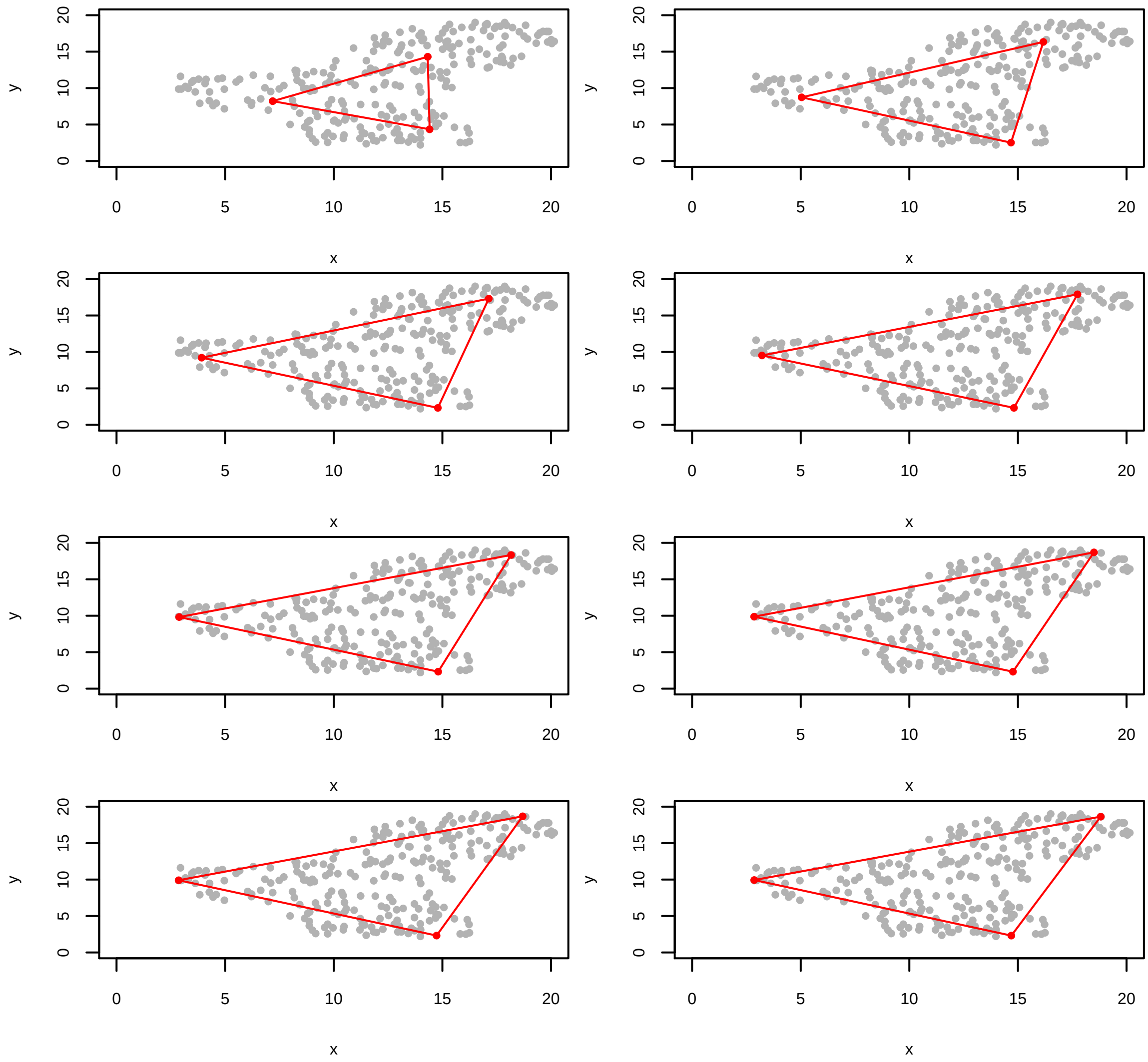
```
> plot(a, toy, chull=chull(toy))
> plot(a, toy, adata.show=TRUE)
```



The left plot shows the archetypes, their approximation of the convex hull (red dots and lines) and the convex hull (black dots and lines) of the data. The right plot additionally shows the approximation of the data through the archetypes and the corresponding α values (green symbols, and grey connection lines); as we can see, all data points outside the approximated convex hull are mapped on its boundary. This plot is based on an idea and Matlab source code of Bernard Pailthorpe (Pailthorpe 2008).

With `saveHistory = TRUE` (which is set per default) each step of the execution is saved and we can examine the archetypes in each iteration using the `ahistory()` command; the initial archetypes, for example, are `ahistory(a, step=0)`. This can be used to create an "evolution movie" of the archetypes,

```
> movieplot(a, toy)
```



The figure shows the plots of the eight steps (the random initialization and the seven iterations) from top to bottom and left to right. In each step the three archetypes move further to the three corners of the data set. A movie of the approximated data is shown when setting parameter `show = 'adata'`[1].

In the previous section we mentioned that the algorithm should be started several times to avoid local minima. This is done using the `stepArchetypes()` function; it passes all arguments to the `archetypes()` function and additionally has the argument `nrep` which specifies the number of repetitions.

```
> set.seed(1986)
> a4 <- stepArchetypes(data=toy, k=3, verbose=FALSE, nrep=4)
```

The result is a S3 `stepArchetypes` object,

```
> a4
```

[1] Real animations are as Flash movies available from http://www.statistik.lmu.de/~eugster/archetypes/.

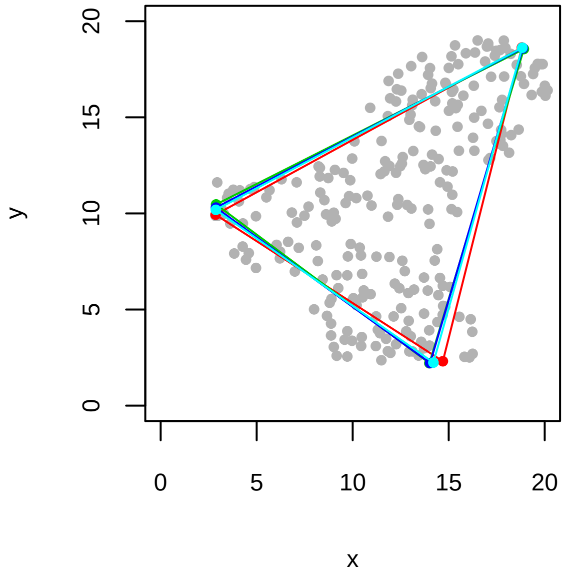
```
StepArchetypes object

stepArchetypes(data = toy, k = 3, nrep = 4, verbose = FALSE)
```

where `summary()` provides an overview of each repetition by showing the final residual sum of squares and number of iterations:

```
> summary(a4)

StepArchetypes object

stepArchetypes(data = toy, k = 3, nrep = 4, verbose = FALSE)

k=3:
Convergence after 7 iterations
with RSS = 0.007563943.
Convergence after 12 iterations
with RSS = 0.007254754.
Convergence after 9 iterations
with RSS = 0.007248349.
Convergence after 13 iterations
with RSS = 0.007255585.
```

There are no huge differences in the residual sum of squares, thus if there are different local minima then they are all equally good. But the following plot shows that the repetition starts all nearly found the same final archetypes (and thus the same local minima),

```
> plot(a4, toy)
```



However, the model of repetition 3 has the lowest residual sum of squares and is the best model:

```
> bestModel(a4)

Convergence after 9 iterations
with RSS = 0.007248349.
```

At the beginning of the example we decided by looking at the data that three archetypes may be a good choice. It is not given that this is the right choice, and with higher-dimensional data this is not possible at all. As already mentioned in the previous section, one simple way to choose the correct number of archetypes is to run the algorithm for different numbers of k and use the "elbow criterion" on the residual sum of squares. The `stepArchetypes()` function allows a vector as value of argument k and executes for each k_i the `archetypes()` function *nrep* times.

```
> set.seed(1986)
> as <- stepArchetypes(data=toy, k=1:10, verbose=FALSE, nrep=4)

There were 23 warnings (use warnings() to see)
```

The occurred warnings indicate that errors occured during the execution, in this case, singular matrizes in solving the linear equation system in Step 2.2 as from $k = 4$:

```
> warnings()

Warnings:
1: In archetypes(..., k = k[i], verbose = verbose) ... :
  k=4: Error in qr.solve(alphas %*% t(alphas)): singular matrix 'a' in solve
2: In archetypes(..., k = k[i], verbose = verbose) ... :
  k=5: Error in qr.solve(alphas %*% t(alphas)): singular matrix 'a' in solve
3: In archetypes(..., k = k[i], verbose = verbose) ... :
  k=5: Error in qr.solve(alphas %*% t(alphas)): singular matrix 'a' in solve
[...]
```

In these cases the residual sum of squares is `NA`:

```
> rss(as)

              r1          r2         r3          r4
k1  0.075569637 0.075569637 0.07556964 0.075569637
k2  0.047510402 0.047510490 0.04751053 0.047510417
k3  0.007370711 0.007370705 0.00737070 0.007630078
k4  0.005124407          NA 0.00594061 0.004970437
k5  0.005249507          NA         NA          NA
k6           NA          NA         NA          NA
k7           NA 0.001216508         NA          NA
k8           NA          NA         NA          NA
k9           NA          NA         NA          NA
k10          NA          NA         NA          NA
```

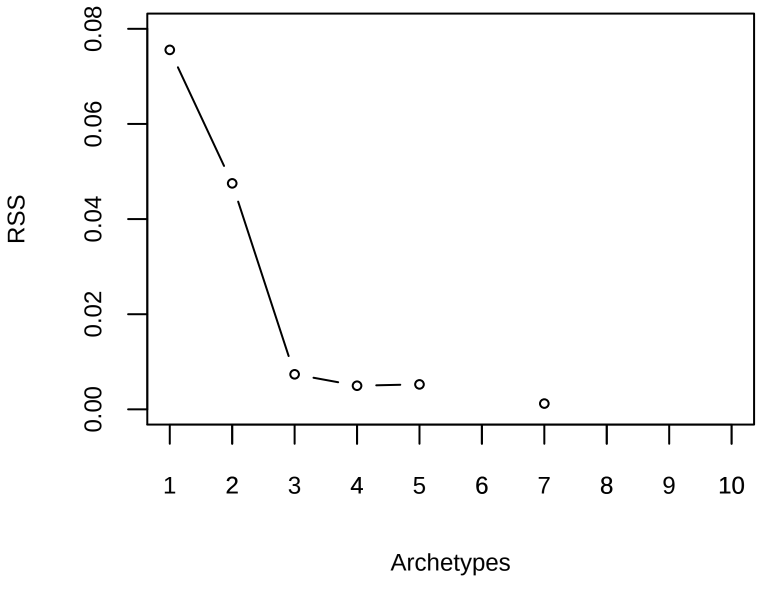
And all errors occured during the first iteration,

```
> iters(as)
```

```
     r1  r2 r3 r4
k1    3   3  3  3
k2   14   9  6 20
k3   73  73 74  8
k4   20   1  9 22
k5   69   1  1  1
k6    1   1  1  1
k7    1 100  1  1
k8    1   1  1  1
k9    1   1  1  1
k10   1   1  1  1
```

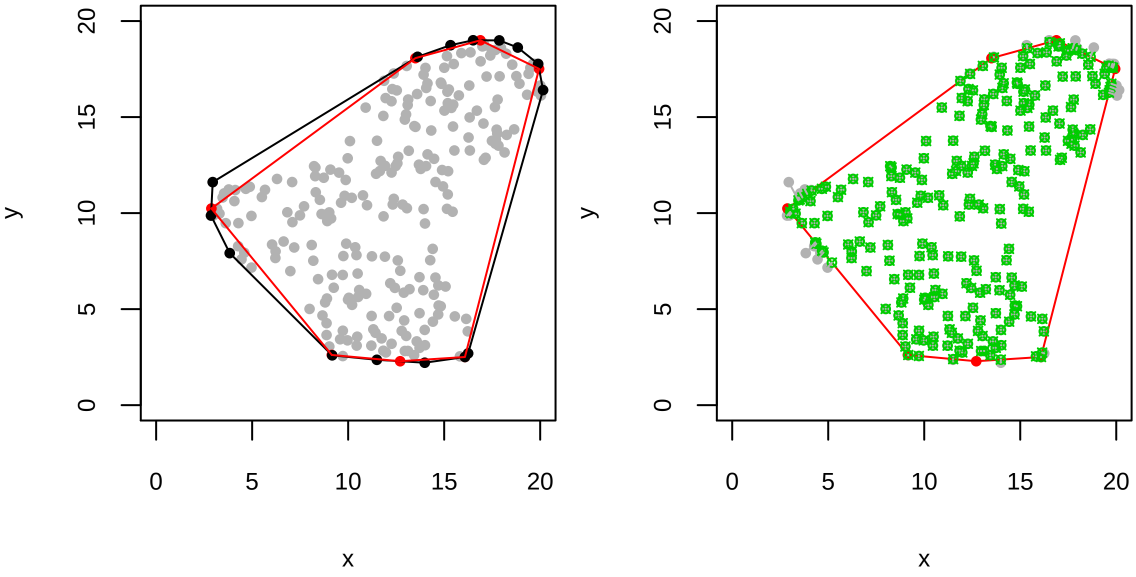
which is an indication for an afflicted random initialisation. But up to $k = 5$ there is always at least one start with a meaningful result and the residual sum of squares curve of the best models shows that by the “elbow criterion” three or maybe seven is the best number of archetypes:

```
> screeplot(as)
```



We already have seen the three archetypes in detail; the seven archetypes of the best repetition and their approximation of the data are:

```
> a7 <- bestModel(as[[7]])
> plot(a7, toy, chull=chull(toy))
> plot(a7, toy, adata.show=TRUE)
```



The approximation of the convex hull is now clearly visible.

As we mentioned in Section 2, there are many ways to solve linear equation systems. One other possibility is the Moore-Penrose pseudoinverse:

```
> set.seed(1986)
> gas <- stepArchetypes(data=toy, k=1:10, family=archetypesFamily('ginv'),
+                       verbose=FALSE, nrep=4)

Loading required package: MASS
There were 23 warnings (use warnings() to see)
```

We use the `ginv()` function from the **MASS** package to calculate the pseudoinverse. The function ignores ill-conditioned matrizes and "just solves the linear equation system", but the `archetypes` function throws warnings of ill-conditioned matrices if the matrix condition number κ is bigger than an upper bound (default is `maxKappa = 1000`):

```
> warnings()

Warnings:
1: In archetypes(..., k = k[i], verbose = verbose) ... :
  k=4: alphas > maxKappa
2: In archetypes(..., k = k[i], verbose = verbose) ... :
  k=5: alphas > maxKappa
3: In archetypes(..., k = k[i], verbose = verbose) ... :
  k=5: alphas > maxKappa
[...]
```

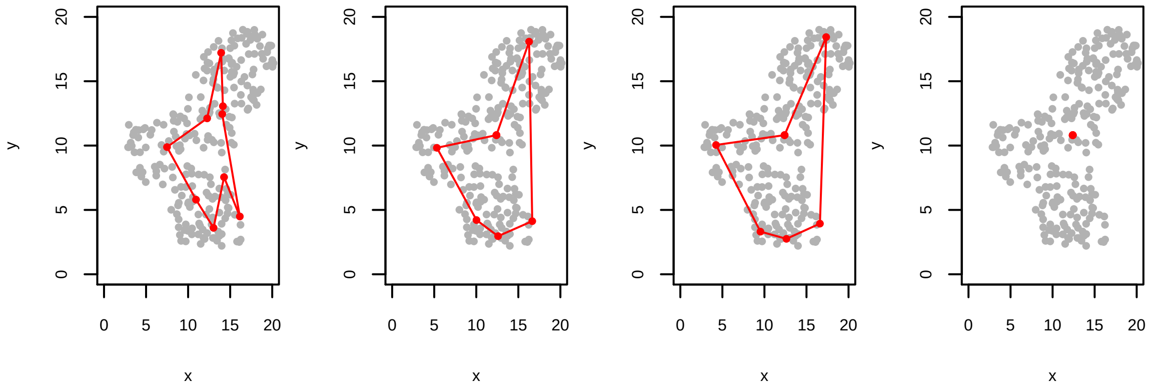
In comparison with the QR decomposition, the warnings occured for the same number of archetypes k_i during the same repetition. In most of these cases the residual sum of squares is about 12,

```
> rss(gas)
```

```
             r1           r2           r3           r4
k1   0.075569637  0.075569637  0.075569637  0.075569637
k2   0.047510402  0.047510490  0.047510530  0.047510417
k3   0.007370711  0.007370705  0.007370701  0.007630078
k4   0.005124407 12.144453252  0.005940610  0.004970437
k5   0.005249507  0.005396092 11.556157268  0.005121638
k6   0.005425348 11.391490573  0.004787822 12.162114653
k7  12.462270024  0.001216508  0.001210484  0.004136435
k8  12.219617787 12.440727546 12.314599508 12.537250751
k9  12.055954228 12.364046902 12.372383424  0.005218947
k10 12.378985500 12.548477210 12.564166317 12.464162790
```

and the randomly chosen initial archetypes "collapse" to the center of the data as we exemplarily see for $k = 9$, $r = 3$:

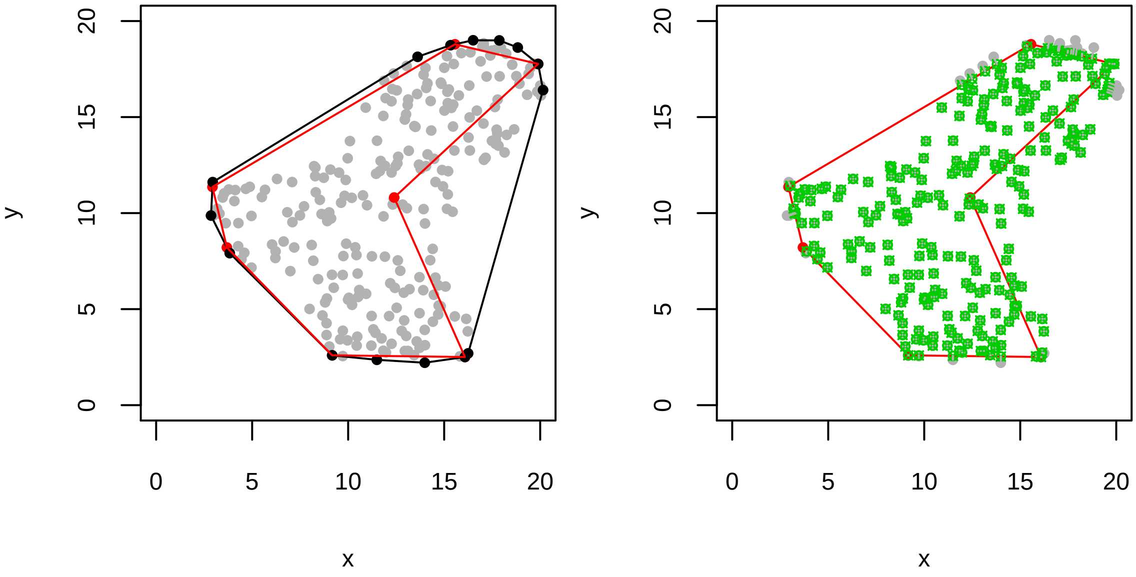
```
> movieplot(gas[[9]][[3]], toy)
```



The figure shows the four steps (from top to bottom and left to right), the random initialization and the three iterations until all archetypes are in the center of the data.

All other residual sum of squares are nearly equivalent to the ones calculated with QR decomposition. Further investigations would show that three or maybe seven is the best number of archetypes, and in case of $k = 3$ nearly the same three points are the best archetypes. An interesting exception is the case $k = 7, r = 2$; the residual sum of squares is exactly the same, but not the archetypes. The plots of the archetypes and their approximation of the data:

```
> ga7 <- bestModel(gas[[7]])
> plot(ga7, toy, chull=chull(toy))
> plot(ga7, toy, adata.show=TRUE)
```



Interesting is the one archetype in the center of the data set and especially the approximation of the data in the right area of it. As the data are approximated by a linear combination of archetypes and non-negative α, the only possibility for this kind of approximation is when α for this archetype is always zero:

```
> apply(alphas(ga7), 2, range)

     [,1]     [,2]      [,3]     [,4]      [,5]      [,6]      [,7]
[1,]    0 0.000000 0.0000000 0.000000 0.0000000 0.0000000 0.0000000
[2,]    0 0.999986 0.9636323 1.000027 0.9844227 0.9999634 0.9925399
```

As we can see, α of archetype 1 (column one) is 0 for all data points. Theoretically, this is not possible, but ill-conditioned matrices during the fit process lead to such results in practice. The occurred warnings (`k=7: alphas > max.kappa`) notify that solving the convex least squares problems lead to the ill-conditioned matrices. Our simulations showed that this behavior mostly appears when requesting a relatively large number of archetypes in relation to size of the data set.

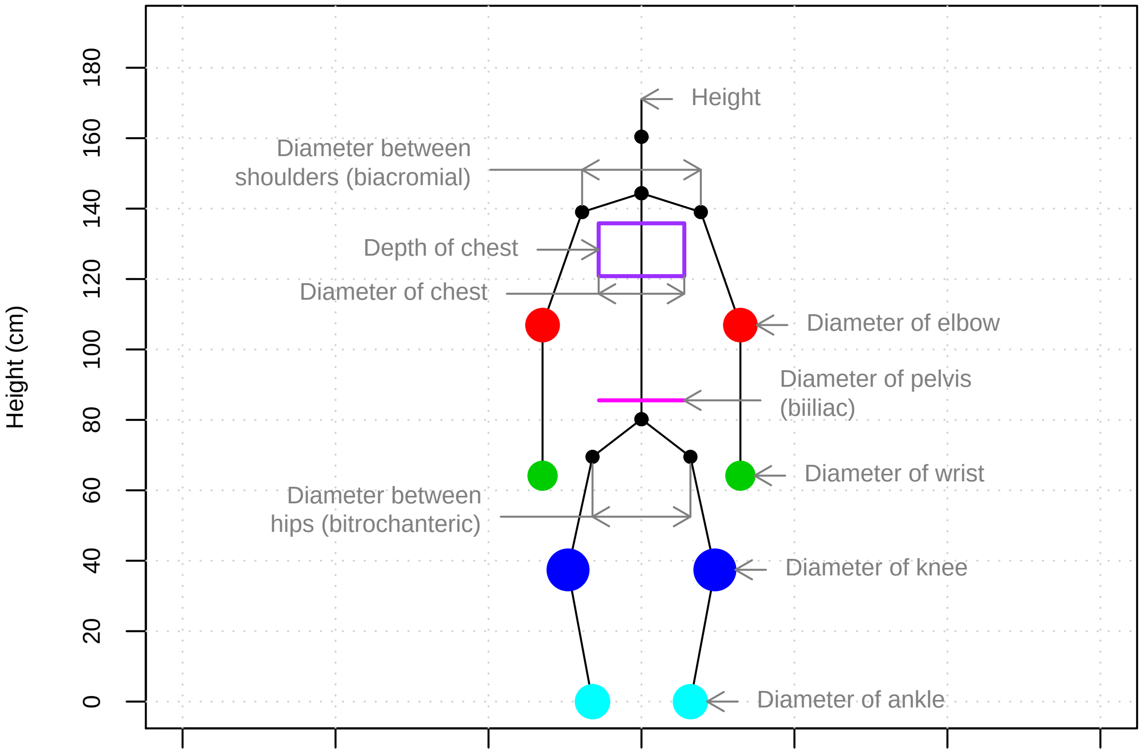
4. Example: Skeletal archetypes

In this section we apply archetypal analysis on an interesting real world example: in Heinz, Peterson, Johnson, and Kerk (2003) the authors took body girth measurements and skeletal diameter measurements, as well as age, weight, height and gender on 247 men and 260 women in their twenties and early thirties, with a scattering of older man and woman, and all physically active. The full data are available within the package as `data(body)`, but we are only interested in a subset, the skeletal measurements and the height (all measured in centimeters),

```
> data(skel)
> skel2 <- subset(skel, select=-Gender)
```

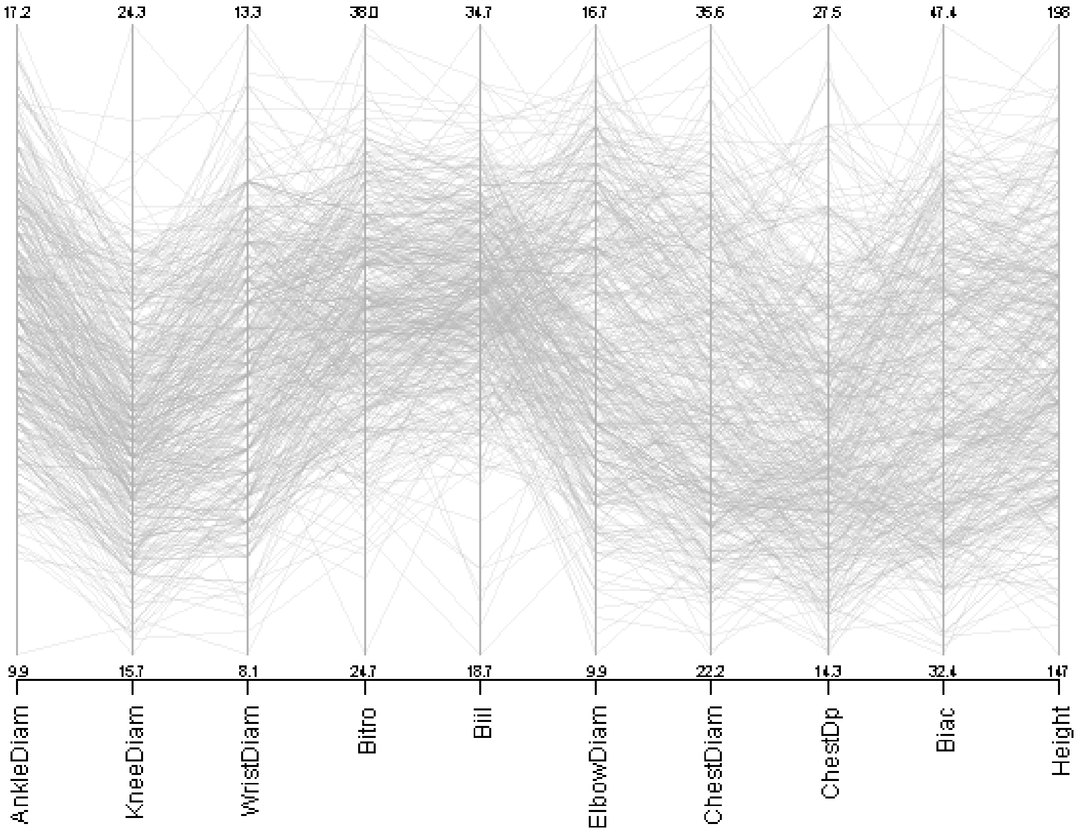
The skeletal measurements consist of nine diameter measurements: biacromial (`Biac`), shoulder diameter; biiliac (`Biil`), pelvis diameter; bitrochanteric (`Bitro`) hip diameter; chest depth between spine and sternum at nipple level, mid-expiration (`ChestDp`); chest diameter at nipple level, mid-expiration (`ChestDiam`); elbow diameter, sum of two elbows (`ElbowDiam`); wrist diameter, sum of two wrists (`WristDiam`); knee diameter, sum of two knees (`KneeDiam`); ankle diameter, sum of two ankles (`AnkleDiam`). See the original publication for a full anatomical explanation of the skeletal measurements and the process of measuring. We use basic elements of *Human Modeling and Animation* to model the skeleton and create a schematic representation of an individual, e.g., `skeletonplot(skel2[1,])` for observation number one. The function `jd()` (for “John Doe”) uses this plot and shows a generic individual with explanations of the measurements:

```
> jd()
```



For visualizing the full data set, parallel coordinates with axes arranged according to the “natural order” are used,

```
> pcplot(skel2)
```



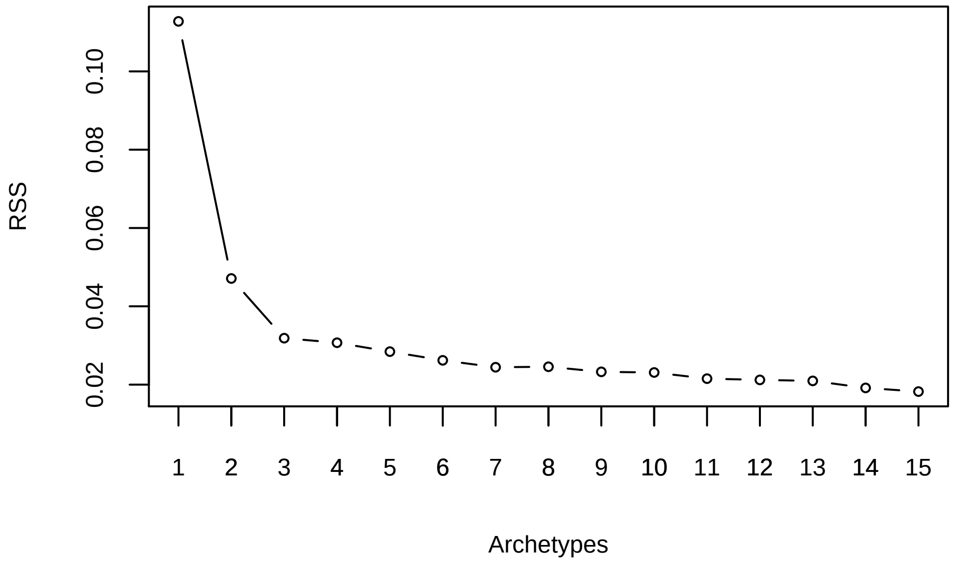
At first view no patterns are visible and it is not possible to guess a meaningful number of archetypes. Therefore, we calculate the archetypes for $k = 1, \dots, 15$ with three repetations each time,

```
> set.seed(1981)
> as <- stepArchetypes(skel2, k=1:15, verbose=FALSE, nrep=3)
```

```
There were 12 warnings (use warnings() to see)
```

The warnings indicate ill-conditioned matrices as from $k = 11$, but, not as in the previous section, the residual sum of squares contains no `NA` values. The corresponding curve of the best model in each case is:

```
> screeplot(as)
```



And according to the "elbow criterion" $k = 3$ or maybe $k = 7$ is the best number of archetypes. Corresponding to Occam's razor we proceed with three archetypes,

```
> a3 <- bestModel(as[[3]])
```

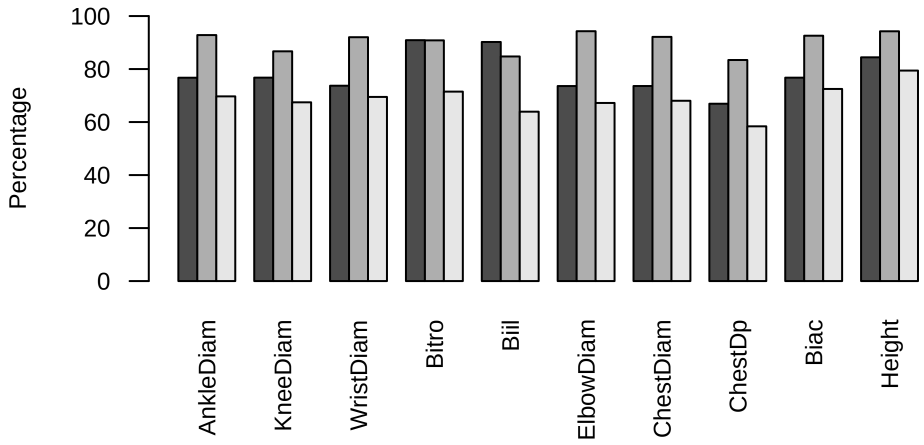
The three archetypes are (transposed for better readability):

```
> t(atypes(a3))
```

```
                [,1]      [,2]       [,3]
AnkleDiam  13.198350  15.96623  11.987396
KneeDiam   18.652792  21.06616  16.390670
WristDiam   9.804258  12.23864   9.241163
Bitro      34.541469  34.51450  27.165757
Biil       31.306794  29.40766  22.179091
ElbowDiam  12.289295  15.74556  11.224216
ChestDiam  26.201939  32.80120  24.219513
ChestDp    18.402293  22.93809  16.059033
Biac       36.378991  43.88447  34.363785
Height    167.250532 186.70914 157.355366
```

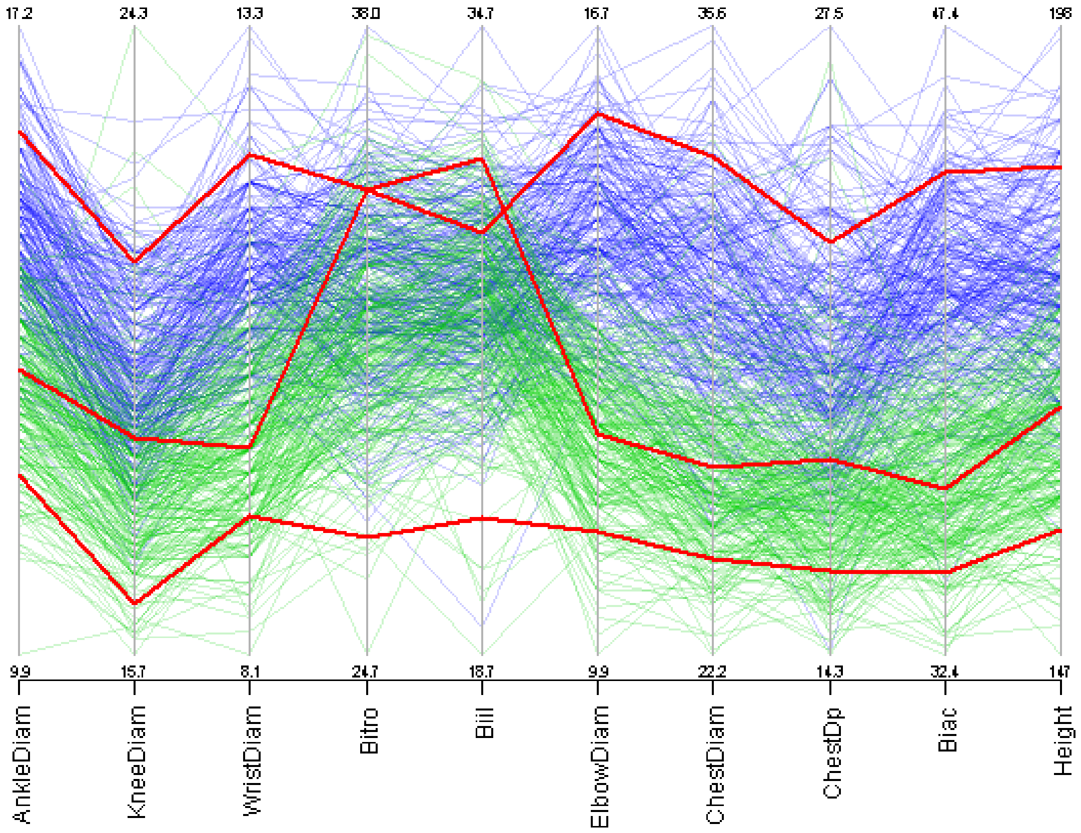
Or as a barplot in relation to the original data:

```
> barplot(a3, skel2, percentage=TRUE)
```



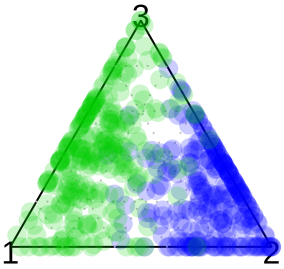
Archetype 2 (gray) represents individuals which are "huge" in all measurements; on the other hand, archetype 3 (lightgray) represents individuals which are "small". Archetype 1 (darkgray) represents individuals with average measures except the bitrochanteric and biiliac – the meaning of this is best visible when looking at the data with gender information (men are blue, women are green colored, with alpha transparency) and the archetypes (red),

```
> pcplot(a3, skel2, data.col=as.numeric(skel$Gender))
```



Archetype 2 reflects the primary difference between men and women in body structure – the comparatively wider hip and pelvis of women. A verification of this interpretation can be done by looking at the coefficients α and see how much each archetype contributes to the approximation of each individual observation. For three archetypes, a ternary plot is a usable graphical representation (e.g., package **vcd** by Meyer, Zeileis, and Hornik (2008)):

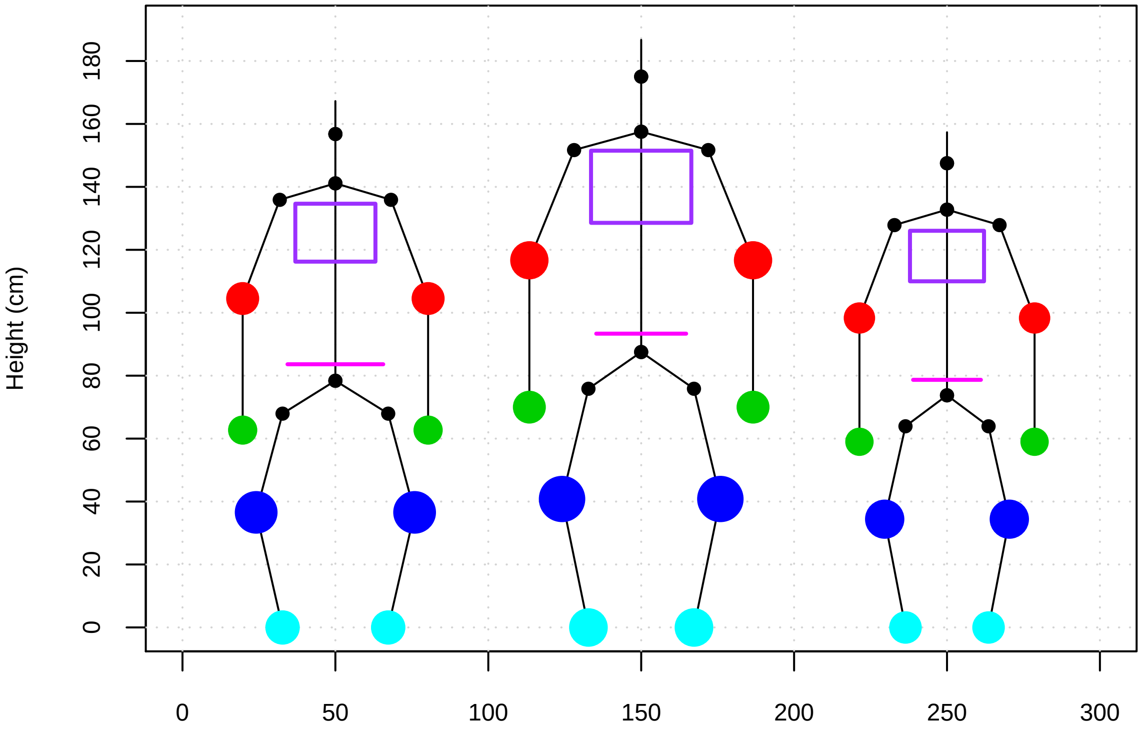
```
> ternaryplot(alphas(a3), col=as.numeric(skel$Gender))
```



Clearly, men observations cluster close to archetype 2 and women mixes mainly the first and the third archetype. For more than three archetypes parallel coordinates with an axis for each archetype projecting the corresponding coefficients (in range $[0, 1]$) can be used to investigate the coefficients α.

Finally, the `skeletonplot()` visualizes the three skeleton archetypes:

```
> skeletonplot(atypes(a3))
```



5. Summary and outlook

In Section 2 we explained the different steps of the algorithms according to the original paper. However, for each problem a number of methods exist and the choice of the right method often depends on the concrete data set. In Section 3 we have already used the `archetypesFamily()` function to use different linear equations solver, but the function has to be extendend to allow abritary problem solving blocks for each step of the algorithm.

Additionally, the two fundamentals defined in Section 2 can be generalized to allow for example arbitrary loss functions, arbitrary conditions on the coefficients α or arbitrary matrix norms. As the algorithm strategy is similar to the least squares formulation of the k-means algorithm, this leads to a general framework for this class of k-means-like algorithms (k-means, k-median, Fuzzy k-means, etc.; see, e.g., Steinly (2006)).

Altogether, the short-term goal for the package **archetypes** was the implementation of a general framework with interchangeable blocks for the concrete problem solving mechanisms of each algorithm step. Now, further work is the design of a clean archetypes family object, especially with a view to our long-term goal, the generalization towards the class of k-means-like algorithms.

Computational details

All computations and graphics in this paper have been done using R version 2.7.1 with the packages **archetypes** 0.1, **MASS** 7.2-43, **nnls** 1.1 and **tools** 2.7.1.

The newest released version of **archetypes** is always available from the Comprehensive R Archive Network at `http://CRAN.R-Project.org/package=archetypes`. The development version is hosted on R-Forge as project **Archetypal Analysis (archetypes)** and available from `http://r-forge.r-project.org/projects/archetypes`. The source code is documented using the Roxygen documentation system for R (Danenberg and Eugster 2008).

Acknowledgments

The authors thank Bernard Pailthorpe, University of Queensland, for providing his Matlab code for comparison with our code.

References

Cutler A, Breiman L (1994). "Archetypal Analysis." *Technometrics*, **36**(4), 338–347.

Danenberg P, Eugster MJA (2008). *roxygen: a documentation system for R.* R package version 1.0, URL `http://www.roxygen.org`.

Eugster MJA, Leisch F (2008). "Spider-Man, the Child and the Trickster – Archetypal Analsis in R." *Technical Report 44*, Institut für Statistik, Ludwig-Maximilians-Universität München, Germany. URL `http://epub.ub.uni-muenchen.de/8270/`.

Golub GH, Loan CFV (1996). *Matrix Computations.* The Johns Hopkins University Press, 3 edition.

Heinz G, Peterson LJ, Johnson RW, Kerk CJ (2003). "Exploring Relationships in Body Dimensins." *Journal of Statistics Education*, **11**(2). URL http://www.amstat.org/publications/jse/v11n2/datasets.heinz.html.

Lawson CL, Hanson RJ (1974). *Solving Least Squares Problems*. Prentice Hall.

Li S, Wang P, Louviere J, Carson R (2003). "Archetypal Analysis: a New Way to Segment Markets Based on Extreme Individuals." In "A Celebration of Ehrenberg and Bass: Marketing Knowledge, Discoveries and Contribution. Proceedings of the ANZMAC 2003 conference, December 1-3, 2003.", pp. 1674–1679.

Luenberger DG (1984). *Linear and Nonlinear Programming*. Addison-Wesley, 2 edition.

Merriam-Webster Online Dictionary (2008). "archetype." In "Merriam-Webster Online Dictionary," Retrieved October 6, 2008, from http://www.merriam-webster.com/dictionary/archetype.

Meyer D, Zeileis A, Hornik K (2008). *vcd: Visualizing Categorical Data*. R package version 1.0-9.

Mullen KM, van Stokkum IHM (2007). *nnls: The Lawson-Hanson algorithm for non-negative least squares (NNLS)*. R package version 1.1.

Pailthorpe B (2008). "Matlab source code." Matlab source code and personal communication.

Porzio GC, Ragozini G, Vistocc D (2006). "Archetypal Analysis for Data Driven Benchmarking." In "Data Analysis, Classification and the Forward Search. Proceedings of the Meeting of the Classification and Data Analysis Group (CLADAG) of the Italian Statistical Society, University of Parma, June 6, 2005," pp. 309–318. ISBN 978-3-540-35977-7.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Steinly D (2006). "K-means clustering: A half-century synthesis." *British Journal of Mathematical and Statistical Psychology*, pp. 1–34.

Wikipedia (2008). "Archetype." In "Wikipedia – The Free Encyclopedia," Retrieved October 7, 2008, from http://en.wikipedia.org/wiki/Archetype.

Affiliation:

Manuel J. A. Eugster
Department of Statistics
Ludwig-Maximilians-Universität München
80539 Munich, Germany
E-mail: Manuel.Eugster@stat.uni-muenchen.de
URL: http://www.statistik.lmu.de/~eugster/