Testing prcr and tidymixmod

Joshua Rosenberg

10/11/2017

Loading, setting up

First, run these three lines of code in order to install `prcr` (for cluster analysis-based profile analysis) and `tidymixmod` (for model-based profile analysis, or Latent Profile Analysis):

```
install.packages("prcr")
install.packages("devtools")
devtools::install_github("jrosen48/tidymixmod)
```

Next, load the packages:

```
library(tidymixmod)
library(prcr)
```

Getting started with prcr

Now, we'll start with `prcr`. Run this line of code:

```
?create_profiles
```

This should give us an idea of how to use the function `create_profiles()`. Here's an example:

```
m2 <- create_profiles(mtcars, disp, hp, wt, n_profiles = 2, to_scale = TRUE, to_center = TRUE)
```

```
## Prepared data: Removed 0 incomplete cases
```

```
## Hierarchical clustering carried out on: 32 cases
```

```
## K-means algorithm converged: 1 iteration
```

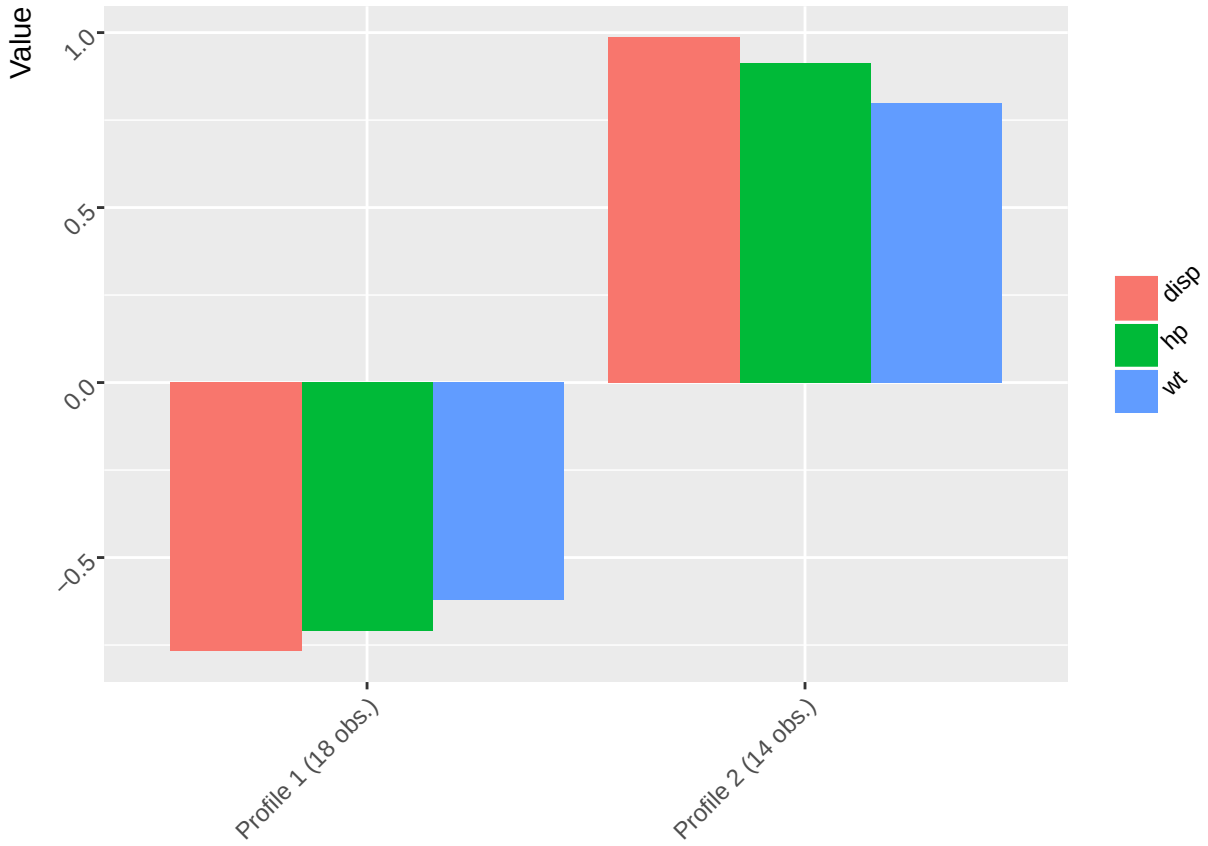
```
## Clustered data: Using a 2 cluster solution
```

```
## Calculated statistics: R-squared = 0.654
```

```
summary(m2)
```

```
## 2 cluster solution (R-squared = 0.654)
##
## Profile n and means:
##
## # A tibble: 2 x 4
##               Cluster       disp         hp         wt
##                 <chr>      <dbl>      <dbl>      <dbl>
## 1 Profile 1 (18 obs.) -0.7679844 -0.7093044 -0.6215850
## 2 Profile 2 (14 obs.)  0.9874085  0.9119628  0.7991807
```

```
plot(m2)
```




```
m2$.data
```

```
## # A tibble: 32 x 12
##      mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  21.0     6 160.0   110  3.90 2.620 16.46     0     1     4     4
##  2  21.0     6 160.0   110  3.90 2.875 17.02     0     1     4     4
##  3  22.8     4 108.0    93  3.85 2.320 18.61     1     1     4     1
##  4  21.4     6 258.0   110  3.08 3.215 19.44     1     0     3     1
##  5  18.7     8 360.0   175  3.15 3.440 17.02     0     0     3     2
##  6  18.1     6 225.0   105  2.76 3.460 20.22     1     0     3     1
##  7  14.3     8 360.0   245  3.21 3.570 15.84     0     0     3     4
##  8  24.4     4 146.7    62  3.69 3.190 20.00     1     0     4     2
##  9  22.8     4 140.8    95  3.92 3.150 22.90     1     0     4     2
## 10  19.2     6 167.6   123  3.92 3.440 18.30     1     0     4     4
## # ... with 22 more rows, and 1 more variables: cluster <int>
```

Getting started with tidymixmod

We'll do the same thing with `tidymixmod` that we did with `prcr`, starting with the help file:

```
?create_profiles_mclust
```

Here's some example code:

```
m3 <- create_profiles_mclust(iris, Sepal.Length, Sepal.Width, Petal.Length, n_profiles = 3, model = 1,
```

```
## [[1]]
## <quosure: global>
## ~Sepal.Length
##
## [[2]]
## <quosure: global>
## ~Sepal.Width
##
## [[3]]
## <quosure: global>
## ~Petal.Length
##
## attr(,"class")
## [1] "quosures"
```

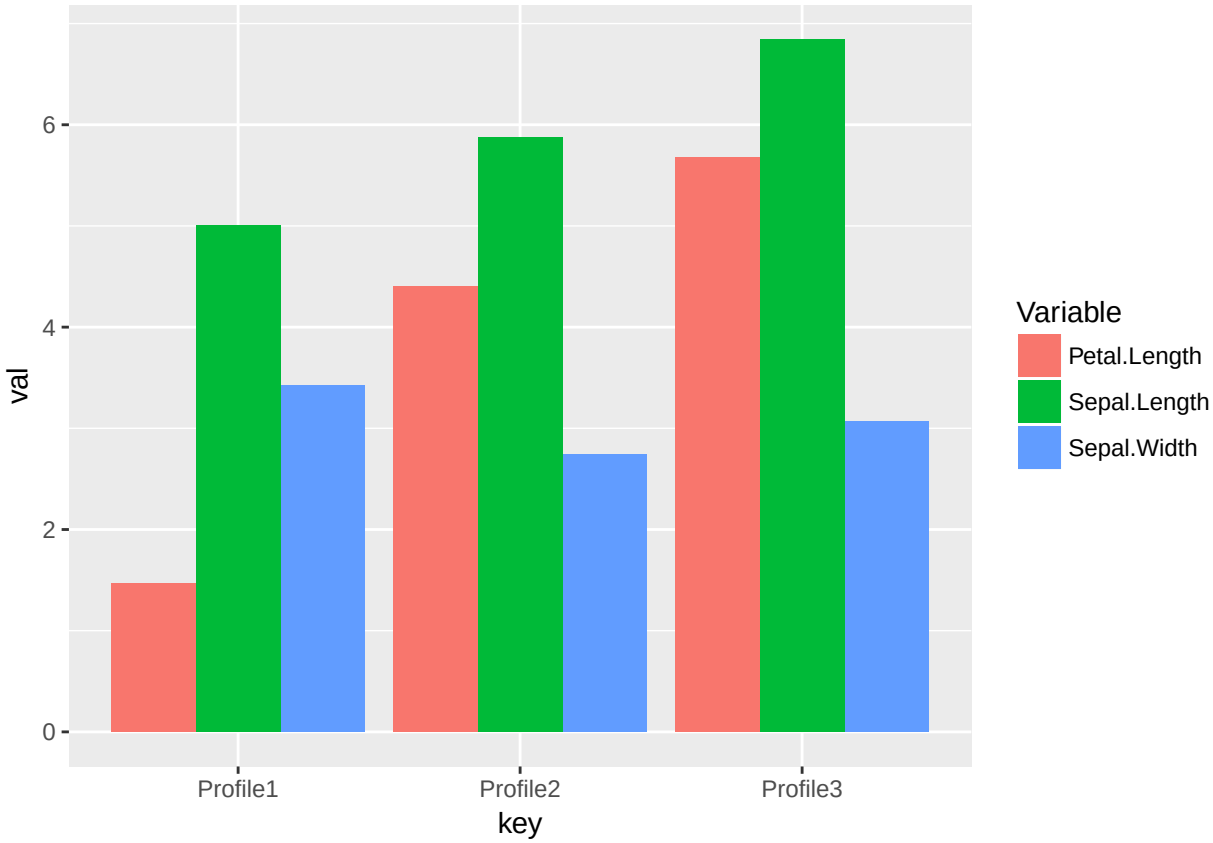
```
## Fit model with 3 profiles using the 'constrained variance' model.
```

```
## Model BIC is 807.309
```

```
calculate_centroids_mclust(m3)
```

```
##       Variable Profile1 Profile2 Profile3
## 1 Sepal.Length 5.006056 5.879192 6.845031
## 2  Sepal.Width 3.427429 2.740044 3.072953
## 3 Petal.Length 1.462947 4.398460 5.679276
```

```
plot_mclust(m3)
```



Determining the number of clusters (for prcr) or mixture components (for tidymixmod)

In `prcr`, we can use the `plot_r_squared()` function

```
?plot_r_squared
```

Here's an example:

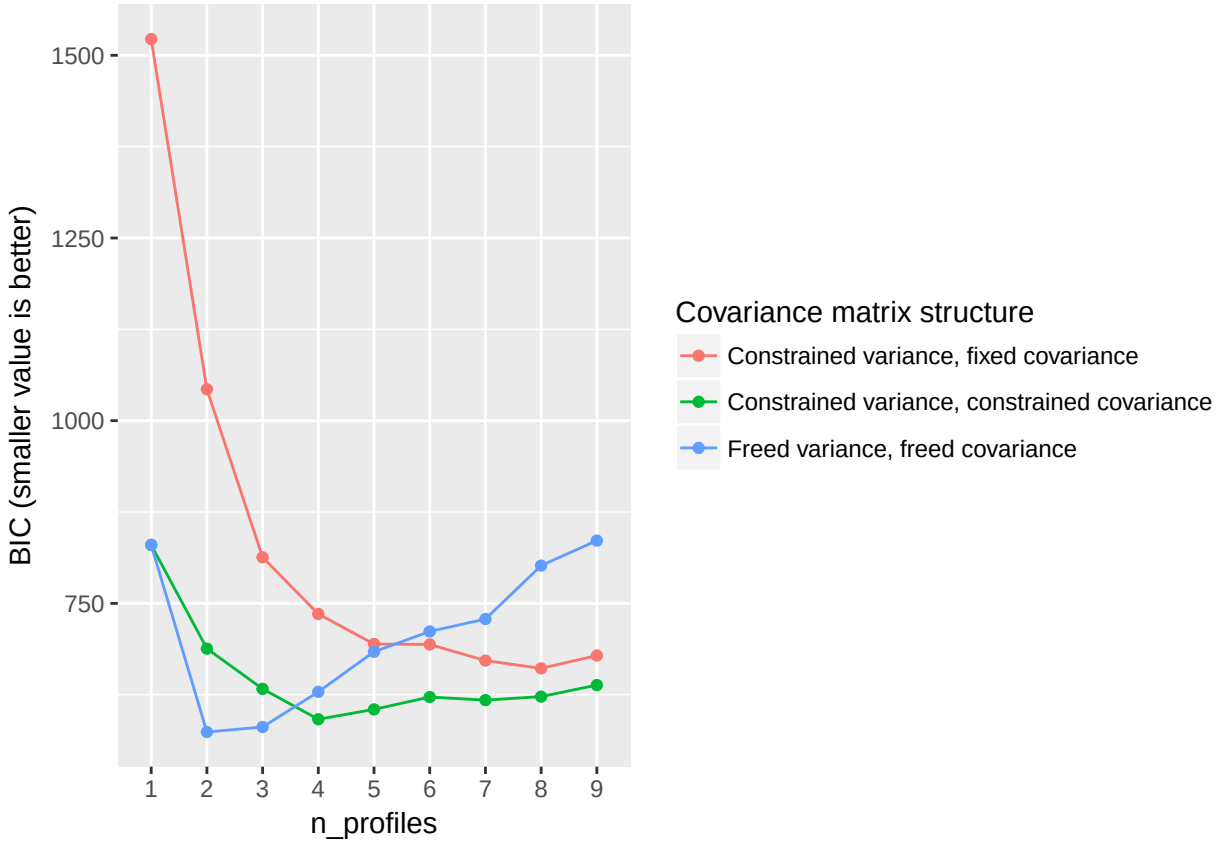
```
plot_r_squared(mtcars,
               disp, hp, wt,
               to_scale = T,
               lower_bound = 2, upper_bound = 7,
               r_squared_table = TRUE)
```

```
##   cluster r_squared_value
## 1       2           0.654
## 2       3           0.750
## 3       4           0.833
## 4       5              NA
## 5       6              NA
## 6       7              NA
```

There is also a very powerful function for performing cross-validation, `cross_validate()`.

In `tidymixmod`, we can use the `explore_models_mclust()` function:

```
df <- dplyr::select(iris, -Species)
explore_models_mclust(df)
```



Future directions

We are actively working to combine `prcr` and `tidymixmod` and to improve the user interface for the combined package. We are also working to add additional functionality.