Introduction to R and the CPCBP package

Mike McCoy

April 8, 2011

Load the package:

```
> library(cpcbp)
```

Set the random number seed (to make all results identical across sessions):

```
> set.seed(1001)
```

First simulate data. `simdata` creates identically shaped but offset multivariate normal groups. The data file `x` below is representative of data for 3 traits that are offset along the 1st principal component axis by 2 units and the 2nd principal component axis by 5 units.

```
> x = simdata(offset = 2, offset2 = 5)
```

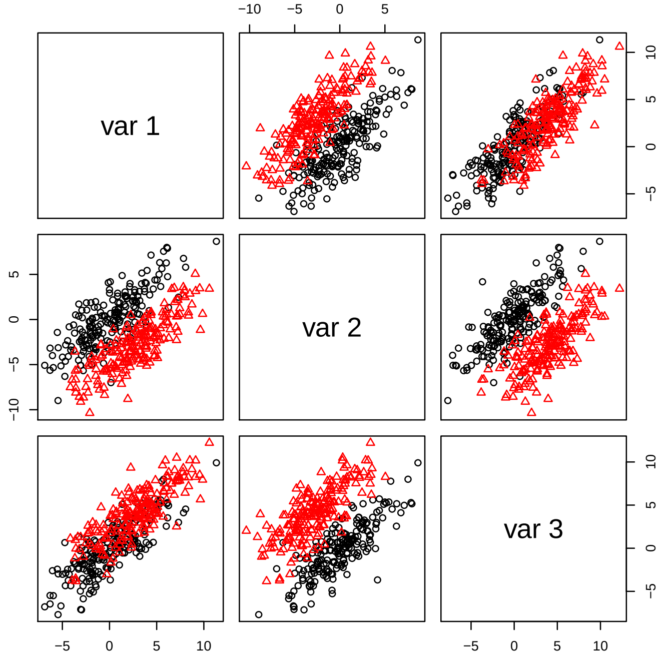
`x` is structured so that the data are in 4 columns. The first 3 simulate log transformed morphometric data The fourth is a factor (e.g., predator or no-predator) Use `str` to view the structure of the data

```
> str(x)
```

```
List of 2
 $ data: num [1:400, 1:3] 6.141 -0.597 -1.422 -5.458 -0.233 ...
 $ f   : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
```

To view all pairwise correlations use `plot_multigrp`. If `x` is a list of the data matrix and a grouping factor (i.e., as produced by `simdata` then you can just give it as a single argument to `plot_multigrp`: in this case `plot\_multigrp(x$data,x$f)` would produce the same answer. This convention holds for all of the functions in the package.

```
> plot_multigrp(x)
```



Next test to determine if groups share the first CPC in common. The null hypothesis for this test is that the PC1s are shared so a significant pvalue tells you that the within groups PC1s are not shared; this is equivalent to heterogeneous slopes in ANCOVA.

```
> phillips.cpc(x)

$evecs.CPC
        [,1]     [,2]     [,3]
[1,] 0.60786 -0.20677 -0.76665
[2,] 0.55973 -0.57324  0.59841
[3,] 0.56321  0.79287  0.23271

$cpc1.pval
[1] 0.3173

$datlist
$datlist[[1]]
$datlist[[1]]$crit
[1] 1774.332

$datlist[[1]]$par
[1] 9
```

```
$datlist[[1]]$testmat
          chi-square df      p
CPC(1)         0.719  1 0.3966
Unrelated      3.014  3 0.3894

$datlist[[1]]$evals
$datlist[[1]]$evals[[1]]
[1] 25.2890  1.7286  2.1377

$datlist[[1]]$evals[[2]]
[1] 23.4960  2.1598  1.5714


$datlist[[1]]$evecs
        [,1]     [,2]     [,3]
[1,] 0.60786 -0.20677 -0.76665
[2,] 0.55973 -0.57324  0.59841
[3,] 0.56321  0.79287  0.23271

$datlist[[1]]$cov
$datlist[[1]]$cov[[1]]
        [,1]   [,2]   [,3]
[1,] 10.6740 7.8285 7.9930
[2,]  7.8285 9.2566 7.4843
[3,]  7.9930 7.4843 9.2243

$datlist[[1]]$cov[[2]]
       [,1]   [,2]   [,3]
[1,] 9.6976 7.5295 7.4096
[2,] 7.5295 8.6339 6.6444
[3,] 7.4096 6.6444 8.8960



$datlist[[2]]
$datlist[[2]]$crit
[1] 1773.613

$datlist[[2]]$par
[1] 10

$datlist[[2]]$testmat
          chi-square df      p
Unrelated      2.296  2 0.3173
```

```
$datlist[[2]]$evals
$datlist[[2]]$evals[[1]]
[1] 25.2680  1.7083  2.1794

$datlist[[2]]$evals[[2]]
[1] 23.5040  2.1661  1.5574


$datlist[[2]]$evecs
$datlist[[2]]$evecs[[1]]
        [,1]     [,2]      [,3]
[1,] 0.61864 -0.35971 -0.698500
[2,] 0.55122 -0.43481  0.712110
[3,] 0.55986  0.82556  0.070712

$datlist[[2]]$evecs[[2]]
        [,1]     [,2]     [,3]
[1,] 0.61864 -0.12831 -0.77513
[2,] 0.55122 -0.63214  0.54457
[3,] 0.55986  0.76416  0.32034


$datlist[[2]]$cov
$datlist[[2]]$cov[[1]]
        [,1]   [,2]   [,3]
[1,] 10.9550 7.7995 8.1366
[2,]  7.7995 9.1054 7.2943
[3,]  8.1366 7.2943 9.0953

$datlist[[2]]$cov[[2]]
       [,1]   [,2]   [,3]
[1,] 9.9667 7.5333 7.5416
[2,] 7.5333 8.4689 6.4788
[3,] 7.5416 6.4788 8.7920




$pool
$pool$crit
[1] 1781.012

$pool$par
[1] 6

$pool$testmat
```

```
                 chi-square df      p
Proportionality      0.390  1 0.5326
Common PC            6.680  3 0.0828
CPC(1)               7.399  4 0.1162
Unrelated            9.695  6 0.1381

$pool$evals
[1]  1.9899  1.8086 24.3930

$pool$evecs
         [,1]     [,2]     [,3]
[1,]  0.20714  0.76874 -0.60509
[2,] -0.79504 -0.22815 -0.56202
[3,]  0.57010 -0.59748 -0.56392

$pool$cov
        [,1]   [,2]   [,3]
[1,] 10.0850 7.6505 7.7277
[2,]  7.6505 9.0569 7.0756
[3,]  7.7277 7.0756 9.0494


$propdat
$propdat$crit
[1] 1780.623

$propdat$par
[1] 7

$propdat$testmat
           chi-square df      p
Common PC       6.291  2 0.0431
CPC(1)          7.009  3 0.0716
Unrelated       9.305  5 0.0975

$propdat$evals
[1]  1.8525  2.0446 25.0140

$propdat$evecs
         [,1]     [,2]     [,3]
[1,] -0.77443  0.18325 -0.60554
[2,]  0.25279 -0.78778 -0.56169
[3,]  0.57996  0.58807 -0.56376

$propdat$cov
$propdat$cov[[1]]
```

```
        [,1]   [,2]   [,3]
[1,] 10.3520 7.8500 7.9275
[2,]  7.8500 9.2791 7.2452
[3,]  7.9275 7.2452 9.2801

$propdat$cov[[2]]
       [,1]   [,2]   [,3]
[1,] 9.8321 7.4560 7.5296
[2,] 7.4560 8.8134 6.8816
[3,] 7.5296 6.8816 8.8144
```

In this example you should get a non-significant p-value. This output also includes the vectors for CPC1.

Now you can do the back projection (i.e. size correction) and test for differences among groups for the three traits: `bp.anova` provides a list of ANOVA tables, one for each trait.

```
> bp.anova(x)

[[1]]
Analysis of Variance Table

Response: bpx[, i]
           Df Sum Sq Mean Sq F value    Pr(>F)
f           1 150.76 150.758   131.3 < 2.2e-16 ***
Residuals 398 456.97   1.148
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

[[2]]
Analysis of Variance Table

Response: bpx[, i]
           Df  Sum Sq Mean Sq F value    Pr(>F)
f           1 1742.82 1742.82  1284.3 < 2.2e-16 ***
Residuals 398  540.08    1.36
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

[[3]]
Analysis of Variance Table

Response: bpx[, i]
           Df Sum Sq Mean Sq F value    Pr(>F)
f           1 797.36  797.36  616.39 < 2.2e-16 ***
Residuals 398 514.85    1.29
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

To see the mean differences among groups of the back projected (i.e. size-corrected) data for each trait, as well as the uncorrected (`sd.raw`) and error corrected (`sd.corr`) standard deviations, use `bp.means`:

```
> bp.means(x)

$meandiffs
[1]  0.6005963 -4.1768344  2.0558096

$sd.raw
[1] 0.1041395 0.1481069 0.1100967

$sd.corr
       [,1]      [,2]      [,3]
1 0.1084071 0.1104516 0.1524689
2 0.1511381 0.1045147 0.1158983
```