Field Old and Young Leaves Lipid Analysis

Kenia E. Segura Aba

4/30/2020

This file was run in R version 3.5.3. The packages used are `tidyverse` version 1.3.0, `readr` version 1.3.1, `RRPP` version 0.4.2.9000, `mixOmics` version 6.6.2, and `labdsv` version 2.0-1. This file must be in the same directory as the Box sync folder in order to run. The following analysis of lipid metabolites was conducted using a split-plot analysis of variance (ANOVA) of Young and Old *P. virgatum* leaves using residual randomization permutation procedure (`RRPP`). Patterns in metabolite classification were visualized using `mixOmics` for principle component analysis (PCA) and partial least squares discriminant analysis (PLS-DA). Dufrene-Legendre indicator analysis was performed to identify specific metabolites indicative of plant response to water treatment and fungal treatment (`labdsv`).

1. Load necessary packages

```
library(tidyverse)
library(readr)
library(RRPP)
library(mixOmics)
library(labdsv)
```

Lipids (Neg)

RRPP

2. Define dependent variable matrix and class matrix.

```
path <- "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/"

O_L_neg <- read_tsv(paste(path, "XCMS Online Results/O_L_Neg/XCMS.annotated.Report_1394368.tsv",
    sep = ""))

Y_L_neg <- read_tsv(paste(path, "XCMS Online Results/Y_L_Neg/XCMS.annotated.Report_1394408.tsv",
    sep = ""))

# dependent variable: metabolite intensities
Y_old <- O_L_neg[, c(2, 12:54)] %>% data.frame(row.names = 1) %>%
    t %>% data.frame()
scaled_Y_old <- scale(Y_old)

Y_young <- Y_L_neg[, c(2, 12:54)] %>% data.frame(row.names = 1) %>%
    t %>% data.frame()
scaled_Y_young <- scale(Y_young)

# class: sample factors
class <- read.csv(paste(path, "XCMS Online Results/class.csv",
    sep = ""), header = T, row.names = 1)
```

3. Define and run multivariate regression models, then print out the results.

```r
O_LMneg <- lm.rrpp(scaled_Y_old ~ Block * Water * Fungus, data = class,
    SS.type = "III", print.progress = F)
summary(O_LMneg)
```

```
##
## Linear Model fit with lm.rrpp
##
## Number of observations: 43
## Number of dependent variables: 1540
## Data space dimensions: 42
## Sums of Squares and Cross-products: Type III
## Number of permutations: 1000
##
## Full Model Analysis of Variance
##
##                       Df Residual Df       SS Residual SS       Rsq        F
## Block * Water * Fungus  7          35 21192.11    43487.89 0.3276455 2.436553
##                       Z (from F) Pr(>F)
## Block * Water * Fungus    4.77486  0.001
##
##
## Redundancy Analysis (PCA on fitted values and residuals)
##
##               Trace Proportion Rank
## Fitted     504.5741  0.3276455    7
## Residuals 1035.4259  0.6723545   35
## Total     1540.0000  1.0000000   42
##
## Eigenvalues
##
##                PC1      PC2      PC3      PC4      PC5      PC6      PC7
## Fitted    217.2294 121.9682  74.1905  33.2772  22.0303  20.7254  15.1531
## Residuals 245.3912 125.0163  92.2257  65.1381  52.3223  36.2104  33.0830
## Total     338.7356 283.4179 172.5747  91.0421  61.7638  46.3547  38.5056
##                PC8      PC9     PC10     PC11     PC12     PC13     PC14
## Fitted
## Residuals  28.4884  25.5026  22.9833  20.8924  19.9502  17.8665  16.5646
## Total      34.4012  28.9196  28.3485  27.4033  22.3948  21.6792  19.5353
##               PC15     PC16     PC17     PC18     PC19     PC20     PC21
## Fitted
## Residuals  16.0867  15.3907  14.8116  13.9240  13.6737  13.1603  12.2774
## Total      18.6819  17.5700  16.9363  16.6384  15.3587  14.4541  13.8768
##               PC22     PC23     PC24     PC25     PC26     PC27     PC28
## Fitted
## Residuals  11.9370  11.6861  11.3297  10.6924  10.6138  10.1909  10.0011
## Total      13.5260  12.8876  12.7079  12.3537  11.7770  11.5493  11.1944
##               PC29     PC30     PC31     PC32     PC33     PC34     PC35
## Fitted
## Residuals   9.5219   9.1020   8.7849   8.4150   8.0646   7.4229   6.7041
## Total      10.9598  10.6132  10.5684  10.2297   9.6834   9.3750   9.2255
##               PC36     PC37     PC38     PC39     PC40     PC41     PC42
## Fitted
## Residuals
```

```
## Total       8.8336   8.5891   8.3525   8.0419   7.4380   7.0363   6.4651
```

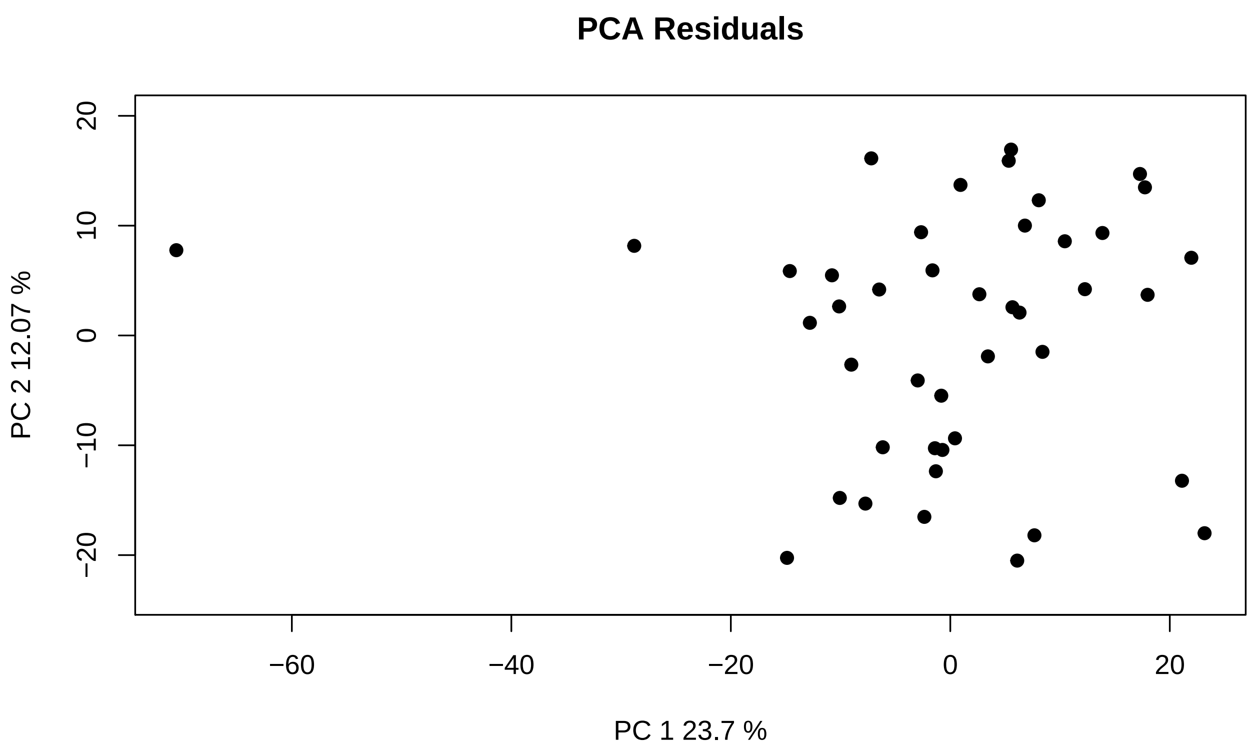
```r
Y_LMneg <- lm.rrpp(scaled_Y_young ~ Block * Water * Fungus, data = class,
    SS.type = "III", print.progress = F)
summary(Y_LMneg)
```

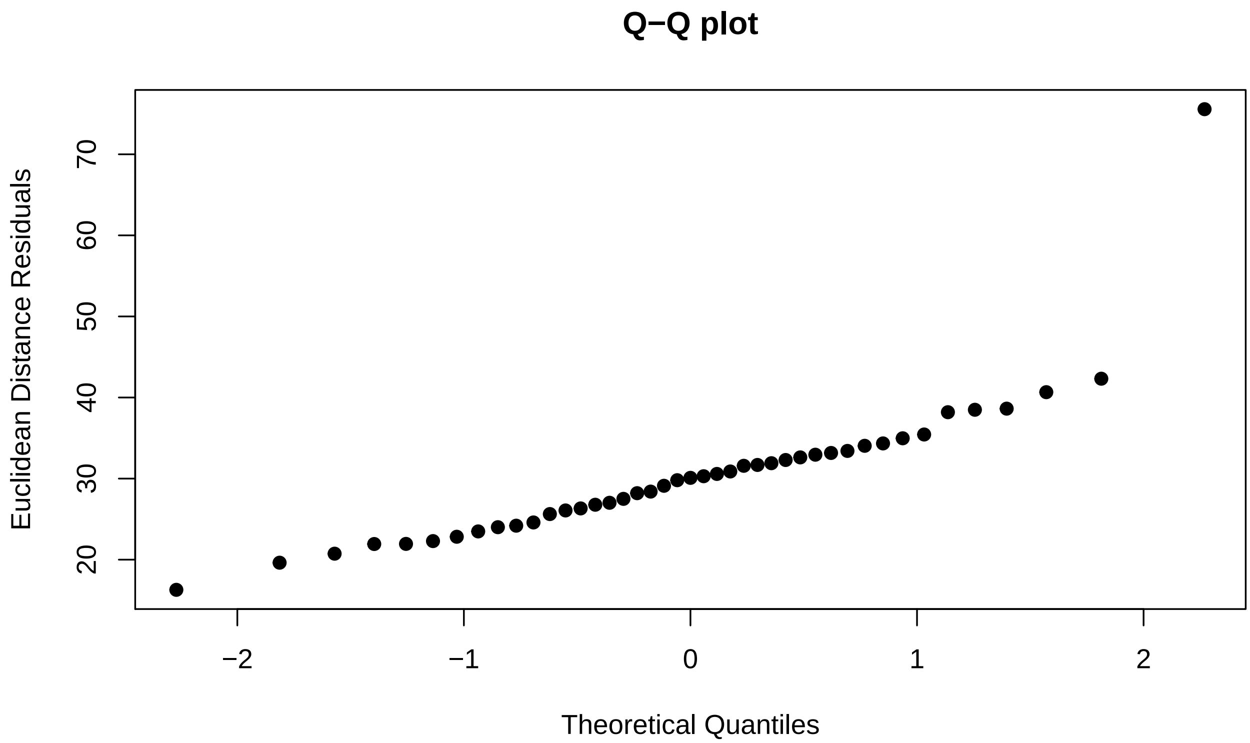
```
##
## Linear Model fit with lm.rrpp
##
## Number of observations: 43
## Number of dependent variables: 1603
## Data space dimensions: 42
## Sums of Squares and Cross-products: Type III
## Number of permutations: 1000
##
## Full Model Analysis of Variance
##
##                        Df Residual Df       SS Residual SS      Rsq        F
## Block * Water * Fungus  7          35 18630.05    48695.95 0.276714 1.912895
##                        Z (from F) Pr(>F)
## Block * Water * Fungus   3.805323  0.001
##
##
## Redundancy Analysis (PCA on fitted values and residuals)
##
##                Trace Proportion Rank
## Fitted     443.5726   0.276714    7
## Residuals 1159.4274   0.723286   35
## Total     1603.0000   1.000000   42
##
## Eigenvalues
##
##                PC1      PC2      PC3      PC4      PC5      PC6      PC7
## Fitted    188.2474 129.9745  43.0239  26.7656  21.9474  17.0881  16.5258
## Residuals 257.0000 179.6952 102.6573  72.5466  50.3806  46.8734  37.5463
## Total     419.6895 234.4477 140.1842 116.2279  67.9671  55.3037  45.4514
##                PC8      PC9     PC10     PC11     PC12     PC13     PC14
## Fitted
## Residuals  35.2538  30.2345  27.3585  24.9055  20.4310  19.8358  18.1858
## Total      38.0977  31.5379  30.7097  28.8956  26.7917  22.2296  22.0810
##               PC15     PC16     PC17     PC18     PC19     PC20     PC21
## Fitted
## Residuals  17.6850  16.5015  15.7274  14.8921  14.0276  13.6146  12.6536
## Total      20.7628  19.1927  18.0042  16.5583  16.0051  14.9467  14.2789
##               PC22     PC23     PC24     PC25     PC26     PC27     PC28
## Fitted
## Residuals  12.3258  11.6761  11.4278  10.8107  10.3672  10.0471   9.6833
## Total      14.0028  13.4649  12.8208  12.3105  12.0785  11.2245  10.8360
##               PC29     PC30     PC31     PC32     PC33     PC34     PC35
## Fitted
## Residuals   9.2572   8.7965   8.4393   7.8603   7.6556   7.0243   6.0502
## Total      10.4619  10.0447   9.9788   9.5432   9.2232   8.8299   8.6136
##               PC36     PC37     PC38     PC39     PC40     PC41     PC42
## Fitted
## Residuals
```

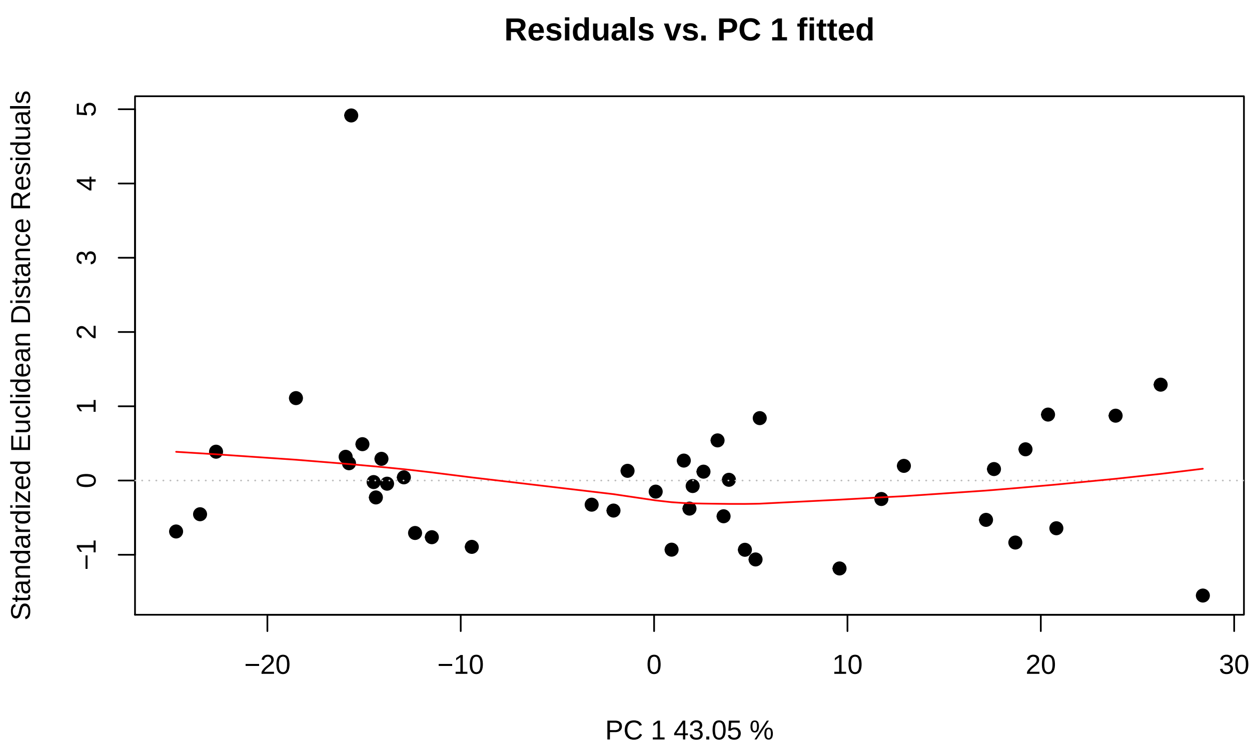
```
## Total        8.3533    7.8636    7.7013    7.4567    6.7107    6.4269    5.6909
```

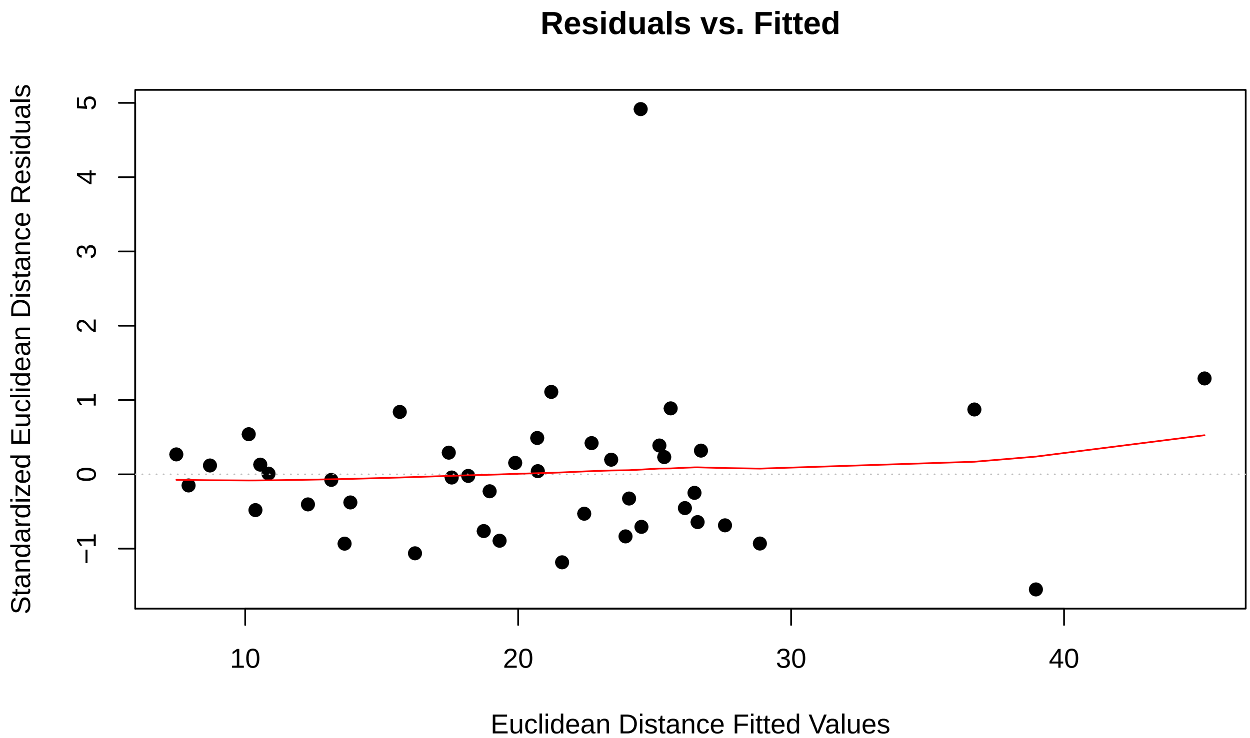
4. Examine RRPP plots to check for assumptions.

```r
## Old Leaves residuals vs fitted values (homoscedasticity
## check)
Odiagnostics <- plot(O_LMneg, type = "diagnostics")
```

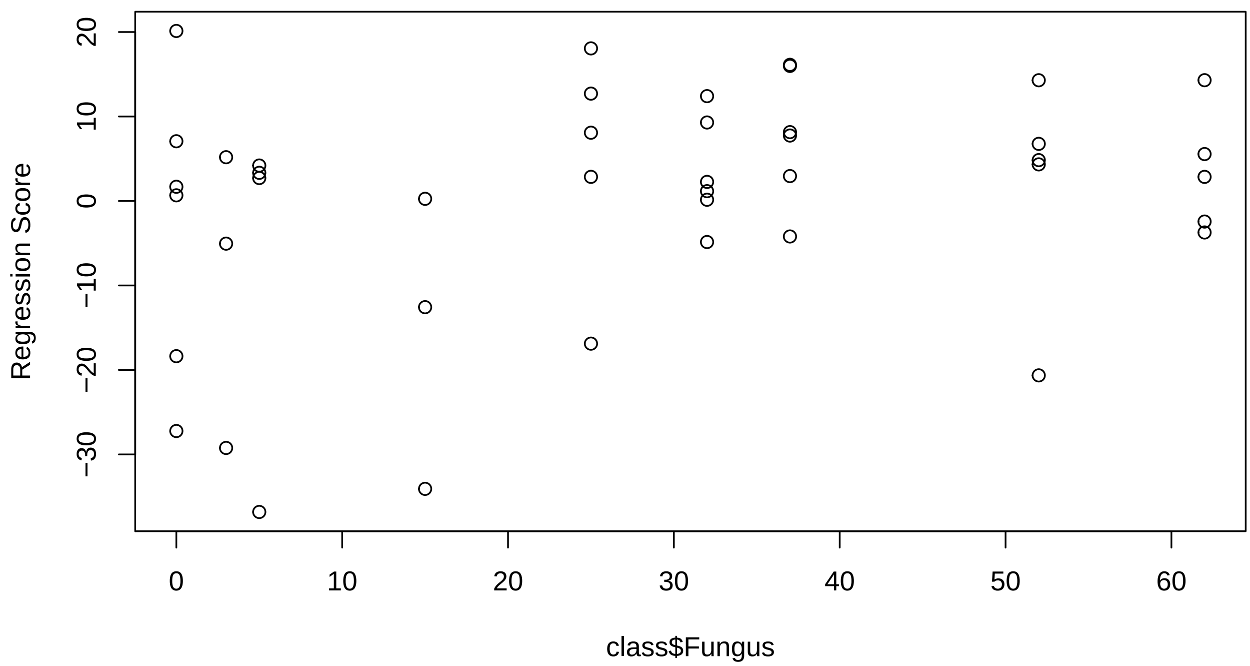



Q−Q plot
Euclidean Distance Residuals
20
30
40
50
60
70
−2
−1
0
1
2
Theoretical Quantiles

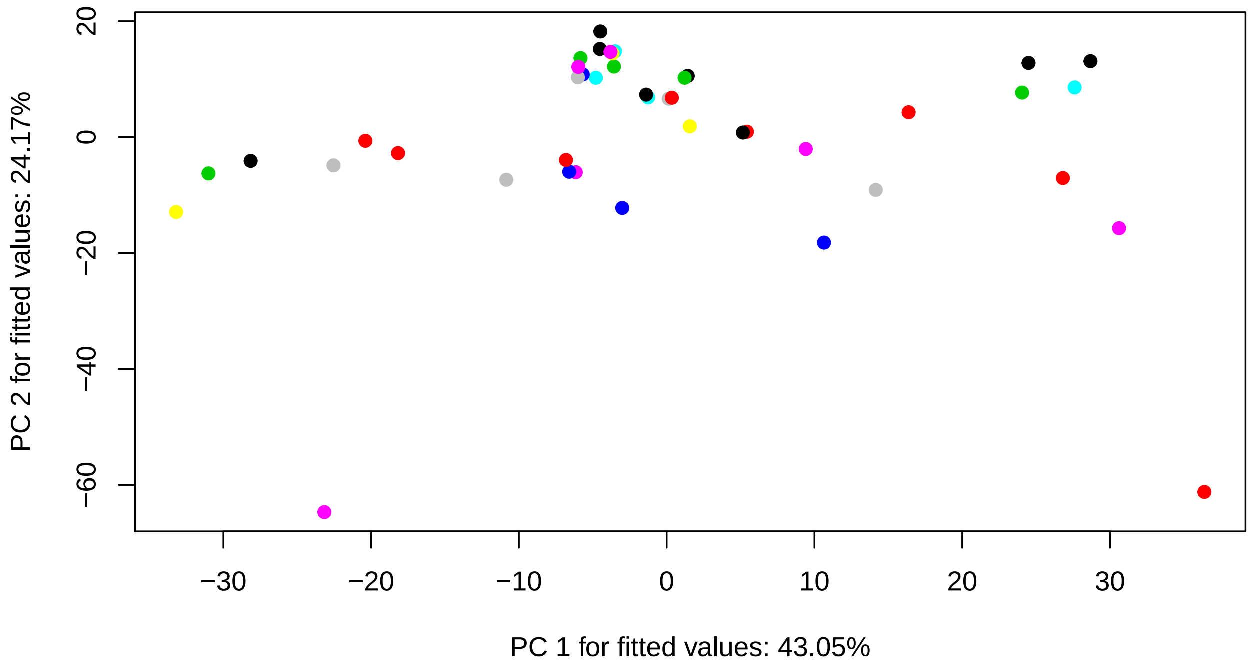

Residuals vs. PC 1 fitted
Standardized Euclidean Distance Residuals
−1
0
1
2
3
4
5
−20
−10
0
10
20
30
PC 1 43.05 %



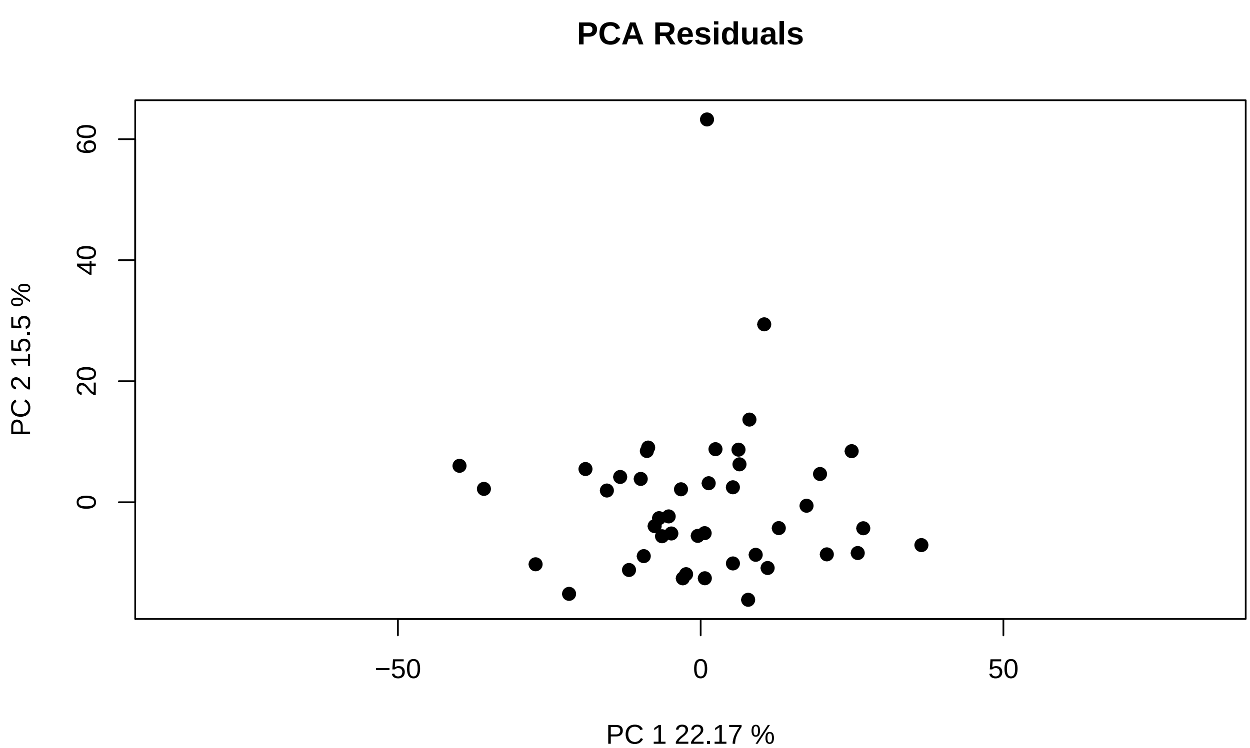

```r
# linear regression plot
Oregression <- plot(O_LMneg, type = "regression", predictor = class$Fungus,
    reg.type = "RegScore")
```

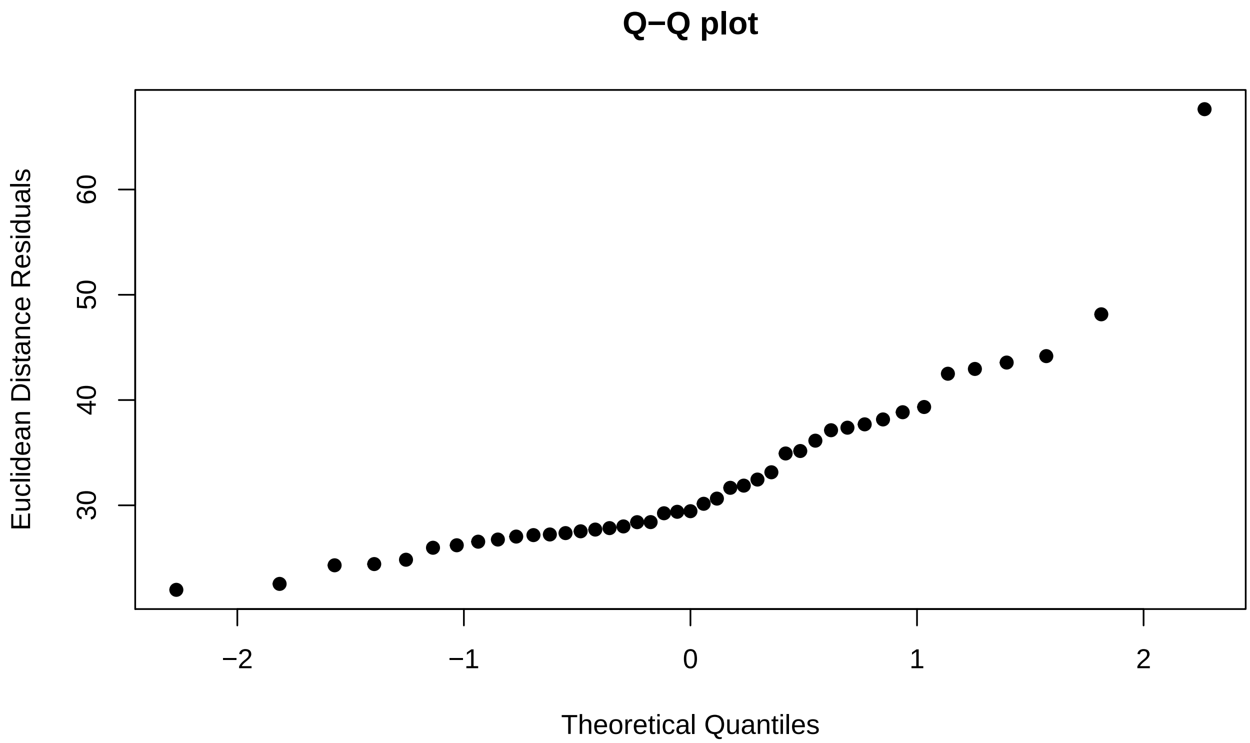


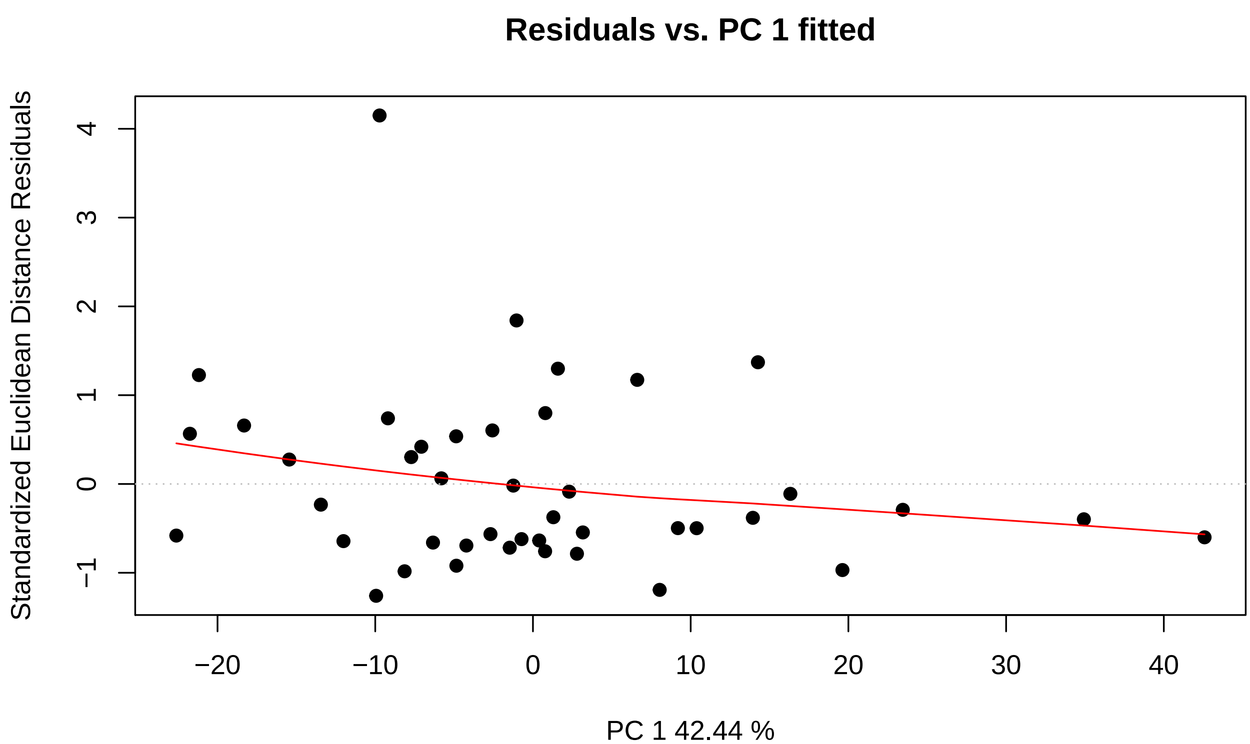

```r
# pca plot
Opcplot <- plot(O_LMneg, type = "PC", pch = 19, col = interaction(class$Water,
    class$Fungus))
```

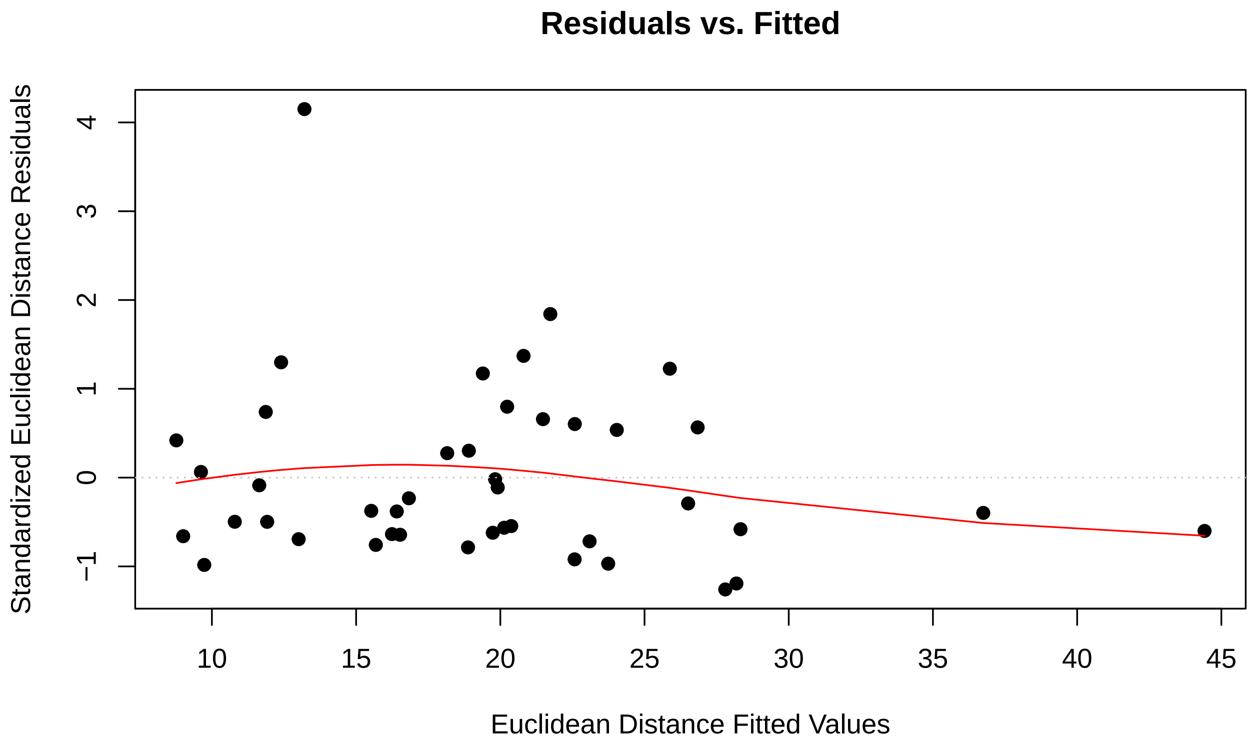



```
## Young Leaves residuals vs fitted values (homoscedasticity
## check)
Ydiagnostics <- plot(Y_LMneg, type = "diagnostics")
```

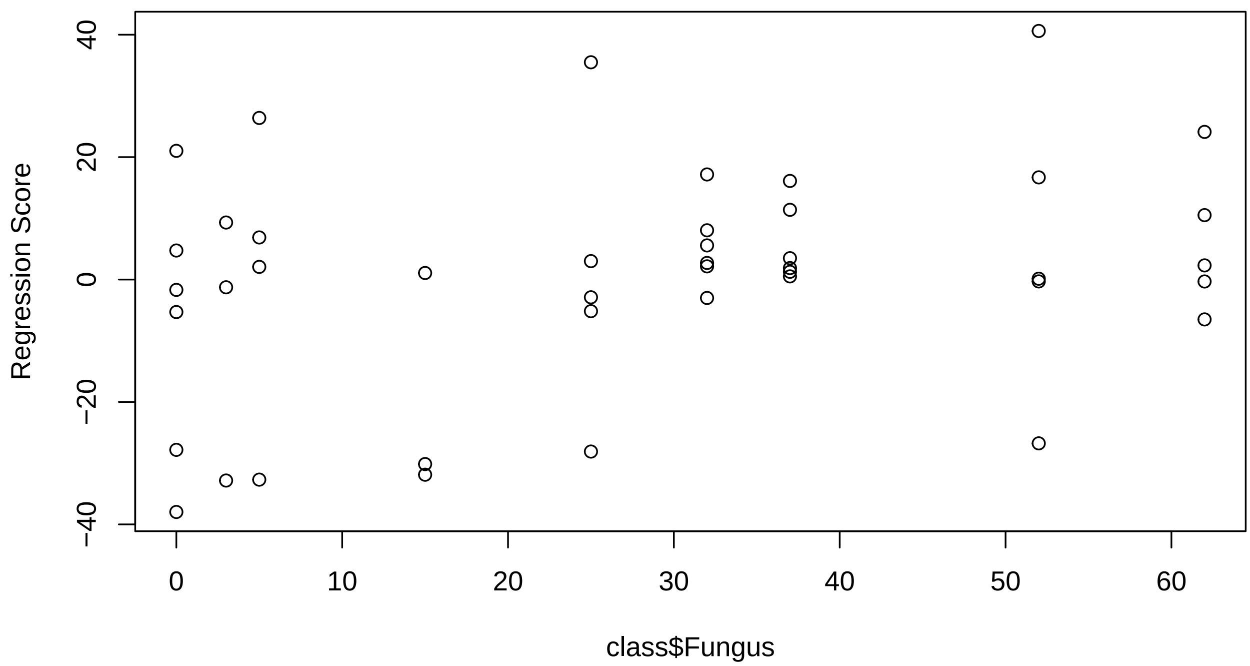



Q−Q plot
Euclidean Distance Residuals
30
40
50
60
−2
−1
0
1
2
Theoretical Quantiles

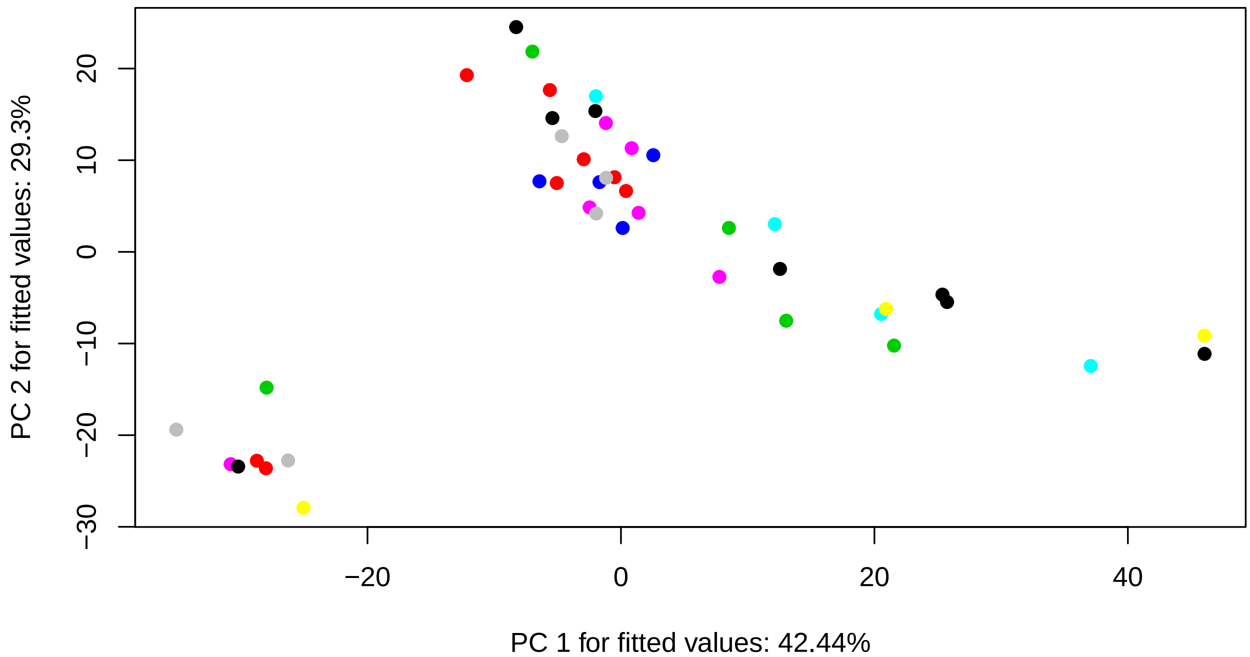

Residuals vs. PC 1 fitted
Standardized Euclidean Distance Residuals
−1
0
1
2
3
4
−20
−10
0
10
20
30
40
PC 1 42.44 %




```
# linear regression plot
Yregression <- plot(Y_LMneg, type = "regression", predictor = class$Fungus,
    reg.type = "RegScore")
```




```
# pca plot
Ypcplot <- plot(Y_LMneg, type = "PC", pch = 19, col = interaction(class$Water,
    class$Fungus))
```



5. Perform an RRPP ANOVA and print results.

```
## Old Leaves
OnegANOVA <- anova(O_LMneg, effect.type = "F", error = c("Residuals",
    "Block:Water", "Block:Water:Fungus", "Residuals", "Block:Water:Fungus",
    "Block:Water:Fungus", "Residuals"))
summary(OnegANOVA, formula = T)
```

```
##
## Analysis of Variance, using Residual Randomization
## Permutation procedure: Randomization of null model residuals
## Number of permutations: 1000
## Estimation method: Ordinary Least Squares
## Sums of Squares and Cross-products: Type III
## Effect sizes (Z) based on F distributions
##
##                    Df    SS     MS     Rsq      F       Z Pr(>F)
## Block               1  4167 4167.5 0.06443 3.3541  3.2868  0.009 **
## Water               1  1731 1731.4 0.02677 1.0292  0.1343  0.450
## Fungus              1  2084 2083.8 0.03222 1.1017  0.3095  0.377
## Block:Water         1  1682 1682.3 0.02601 1.3540  1.0381  0.146
## Block:Fungus        1  1744 1744.4 0.02697 0.9223 -0.1898  0.567
## Water:Fungus        1  1593 1593.0 0.02463 0.8422 -0.7313  0.784
## Block:Water:Fungus  1  1891 1891.4 0.02924 1.5223  1.3137  0.106
## Residuals          35 43488 1242.5 0.67235
## Total              42 64680
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call: lm.rrpp(f1 = scaled_Y_old ~ Block * Water * Fungus, SS.type = "III",
##     data = class, print.progress = F)
```

```
## Young Leaves
YnegANOVA <- anova(Y_LMneg, effect.type = "F", error = c("Residuals",
```

```r
    "Block:Water", "Block:Water:Fungus", "Residuals", "Block:Water:Fungus",
    "Block:Water:Fungus", "Residuals"))
summary(YnegANOVA, formula = T)
```

```
##
## Analysis of Variance, using Residual Randomization
## Permutation procedure: Randomization of null model residuals
## Number of permutations: 1000
## Estimation method: Ordinary Least Squares
## Sums of Squares and Cross-products: Type III
## Effect sizes (Z) based on F distributions
##
##                    Df    SS     MS     Rsq      F        Z Pr(>F)
## Block               1  2341 2341.3 0.03478 1.6828  1.60747  0.070 .
## Water               1  1323 1323.0 0.01965 1.2429  1.00561  0.160
## Fungus              1  3180 3180.2 0.04724 2.5291  2.36145  0.012 *
## Block:Water         1  1064 1064.4 0.01581 0.7650 -0.58460  0.704
## Block:Fungus        1  2296 2296.2 0.03411 1.8261  1.70026  0.041 *
## Water:Fungus        1  1391 1390.7 0.02066 1.1060  0.49091  0.305
## Block:Water:Fungus  1  1257 1257.4 0.01868 0.9038 -0.13380  0.527
## Residuals          35 48696 1391.3 0.72329
## Total              42 67326
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call: lm.rrpp(f1 = scaled_Y_young ~ Block * Water * Fungus, SS.type = "III",
##     data = class, print.progress = F)
```

6. Test lm.rrpp model coefficients. "d" is the amount of change in a variable for the coefficient indicated.

```r
## Old Leaves test model coefficients
Onegcoef <- coef(O_LMneg, test = T)
summary(Onegcoef)
```

```
##
## Linear Model fit with lm.rrpp
##
## Number of observations: 43
## Number of dependent variables: 1540
## Data space dimensions: 42
## Sums of Squares and Cross-products: Type III
## Number of permutations: 1000
##
## Statistics (distances) of coefficients with 95 percent confidence intervals,
## effect sizes, and probabilities of exceeding observed values based on
## 1000 random permutations using RRPP
##
##                          d.obs  UCL (95%)        Zd Pr(>d)
## (Intercept)         56.7851643 68.2393729 0.1881122  0.404
## Block               30.7839565 21.3721323 4.6692021  0.001
## WaterLow            51.3669825 52.1862425 1.7171593  0.064
## Fungus               1.1988549  1.1643577 2.1308690  0.043
## Block:WaterLow      26.0901902 27.3651382 1.5179096  0.077
## Block:Fungus         0.5625282  0.5990145 1.4492688  0.084
## WaterLow:Fungus      1.4541906  1.5461400 1.3736079  0.095
```

```
## Block:WaterLow:Fungus  0.7837857  0.7935525 1.8974730  0.059
```

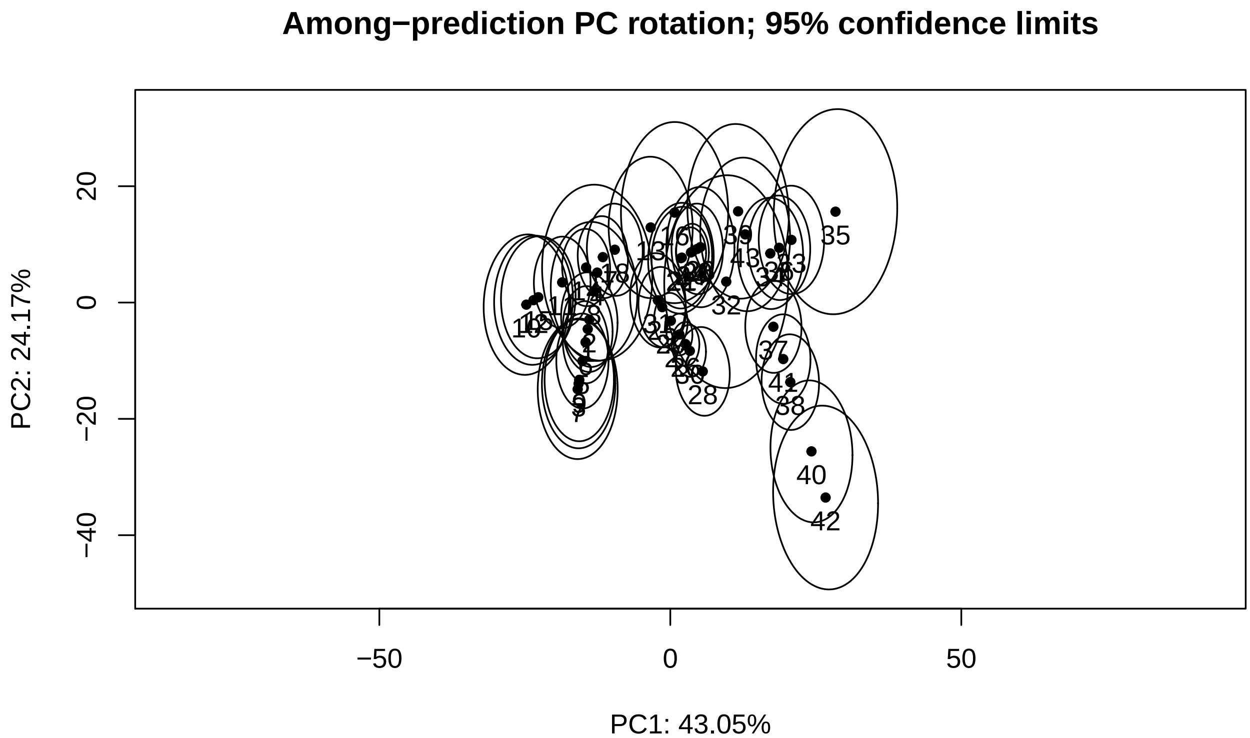
```r
## Young Leaves test model coefficients
Ynegcoef <- coef(Y_LMneg, test = T)
summary(Ynegcoef)
```

```
##
## Linear Model fit with lm.rrpp
##
## Number of observations: 43
## Number of dependent variables: 1603
## Data space dimensions: 42
## Sums of Squares and Cross-products: Type III
## Number of permutations: 1000
##
## Statistics (distances) of coefficients with 95 percent confidence intervals,
## effect sizes, and probabilities of exceeding observed values based on
## 1000 random permutations using RRPP
##
##                             d.obs  UCL (95%)         Zd Pr(>d)
## (Intercept)           45.4608543 61.7344120 -0.8080941  0.784
## Block                 23.0735567 22.1138801  2.1888278  0.034
## WaterLow              44.9023881 55.6540553  0.3883270  0.308
## Fungus                 1.4810450  1.2459086  3.2164939  0.006
## Block:WaterLow        20.7529414 28.6532397 -0.2077464  0.525
## Block:Fungus           0.6453841  0.6435496  1.9750292  0.048
## WaterLow:Fungus        1.3587323  1.6653060  0.5256945  0.269
## Block:WaterLow:Fungus  0.6390634  0.8120751  0.2173253  0.359
```

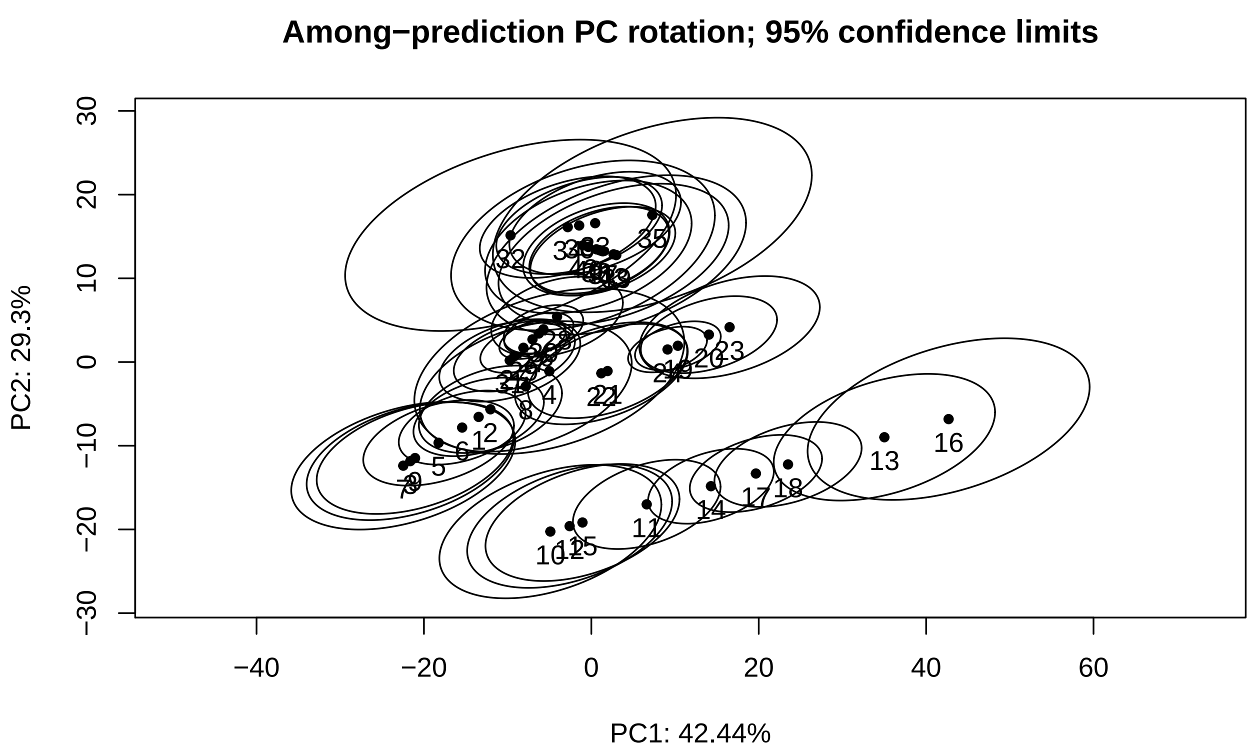
Fungus has the largest effect on the model for young leaves. The standard is the mean for High water treatment. Block has the largest effect on the model for old leaves. Fungus coming in second.

7. Compute predicted values from the lm.rrpp model fit using bootstrapped residuals to generate confidence intervals (precision of group mean estimates).

```r
O_pred <- predict(O_LMneg)
plot(O_pred, PC = T, ellipse = T)
```




```r
Y_pred <- predict(Y_LMneg)
plot(Y_pred, PC = T, ellipse = T)
```



8. Test pairwise differences between least squares means. Similar to tukeyHSD function in the r stats package. The pairwise function will generate tables with confidence intervals and p-values for the pairwise statistic, Euclidean distance between least-squares means.

```r
## Old Leaves pairwise differences of water
Onegpw <- pairwise(O_LMneg, groups = class$Fungus)
summary(Onegpw, confidence = 0.95, stat.table = T)
```

```
##
## Pairwise comparisons
##
## Groups: 0 3 5 15 25 32 37 52 62
##
## RRPP: 1000 permutations
##
## LS means:
## Vectors hidden (use show.vectors = TRUE to view)
##
## Pairwise distances between means, plus statistics
##               d UCL (95%)          Z Pr > d
## 0:3   14.717160 18.966988 -0.7815989  0.774
## 0:5    9.615246 13.252094 -0.7853229  0.769
## 0:15  14.626625 19.114954 -0.7245472  0.770
## 0:25   8.732025 13.942737 -1.3339615  0.924
## 0:32  10.619979 16.998242 -1.4035443  0.941
## 0:37  12.279350 19.654217 -1.4035443  0.941
## 0:52  18.756031 28.552349 -1.1672332  0.885
## 0:62  20.016731 32.390554 -1.4877295  0.951
## 3:5    9.923031 15.817301 -2.0283716  0.994
## 3:15   6.277265 10.465867 -2.1826828  1.000
## 3:25  16.385586 21.804195 -1.0069745  0.836
## 3:32  17.707795 23.602854 -1.0088981  0.832
## 3:37  18.684051 25.319513 -1.0684106  0.849
## 3:52  22.119230 30.587804 -0.9714698  0.825
## 3:62  24.622923 35.480760 -1.2853263  0.902
## 5:15   7.059009 10.364090 -1.7977428  0.976
## 5:25  10.740394 13.929872 -0.4854619  0.681
## 5:32  12.727351 17.076660 -0.7444280  0.771
## 5:37  13.900573 18.932075 -0.8553641  0.803
## 5:52  18.368463 25.548722 -0.8458061  0.791
## 5:62  20.587684 30.089895 -1.2176369  0.886
## 15:25 13.953746 16.976759 -0.5398929  0.696
## 15:32 15.316247 18.867668 -0.6440036  0.739
## 15:37 16.068288 20.125304 -0.7459548  0.775
## 15:52 18.657421 24.070424 -0.6520620  0.738
## 15:62 21.516178 29.492937 -1.1429056  0.880
## 25:32  2.629988  3.983467 -1.5425395  0.945
## 25:37  4.057069  6.329472 -1.5417956  0.953
## 25:52 10.413237 15.140523 -1.0867176  0.865
## 25:62 11.980143 19.353471 -1.5544049  0.956
## 32:37  1.659372  2.655975 -1.4035443  0.941
## 32:52  8.596237 12.486425 -1.0022531  0.844
## 32:62  9.593783 15.789034 -1.5592614  0.956
## 37:52  7.130951 10.151648 -1.0160471  0.849
## 37:62  8.012080 13.215906 -1.5802204  0.959
## 52:62  5.474832 10.015951 -2.1279157  1.000
```

```r
## Young Leaves pairwise differences of water
Ynegpw <- pairwise(Y_LMneg, groups = class$Fungus)
summary(Ynegpw, confidence = 0.95, stat.table = T)
```

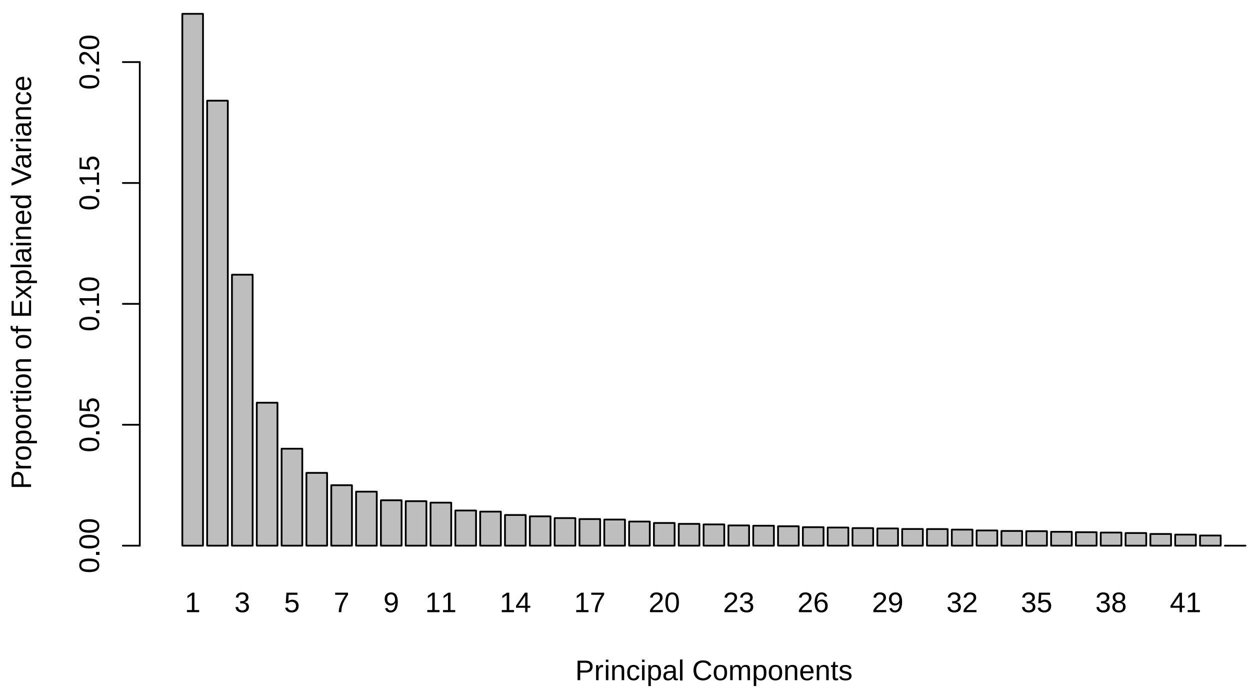
```
##
## Pairwise comparisons
##
## Groups: 0 3 5 15 25 32 37 52 62
##
## RRPP: 1000 permutations
##
## LS means:
## Vectors hidden (use show.vectors = TRUE to view)
##
## Pairwise distances between means, plus statistics
##               d UCL (95%)          Z Pr > d
## 0:3   11.702201 17.143285 -0.9091293  0.807
## 0:5    7.454235 11.790878 -1.0018822  0.889
## 0:15  10.235526 16.142513 -1.0245425  0.849
## 0:25   7.397407 12.437263 -1.7038175  0.978
## 0:32   9.712869 16.199092 -1.4884754  0.957
## 0:37  11.230505 18.730200 -1.4884754  0.957
## 0:52  17.439416 27.781528 -1.2067170  0.886
## 0:62  19.589381 32.388115 -1.3805056  0.937
## 3:5   10.079907 16.187297 -1.5572750  0.980
## 3:15   7.020215 11.014540 -1.9778254  0.994
## 3:25  15.636545 21.740261 -0.8700043  0.801
## 3:32  17.352414 24.284286 -0.7964528  0.776
## 3:37  18.541867 26.288697 -0.8238997  0.786
## 3:52  23.109898 33.090374 -0.7804791  0.774
## 3:62  25.793837 38.341462 -0.9107488  0.811
## 5:15   5.595434  9.809929 -1.4204310  0.952
## 5:25   9.854916 14.339773 -0.8564060  0.787
## 5:32  12.403808 18.250001 -0.8102353  0.765
## 5:37  13.647560 20.156268 -0.8760472  0.789
## 5:52  19.028556 28.208242 -0.7532405  0.758
## 5:62  21.353970 32.512155 -0.9817034  0.823
## 15:25 11.699219 15.679707 -0.6329744  0.725
## 15:32 13.638580 18.335328 -0.6077092  0.702
## 15:37 14.688404 19.835274 -0.6473870  0.721
## 15:52 18.925545 26.078639 -0.5541992  0.695
## 15:62 21.776994 31.515025 -0.8025172  0.775
## 25:32  3.077239  4.700321 -0.9376598  0.808
## 25:37  4.409372  7.019984 -1.1026361  0.865
## 25:52 10.900584 16.264982 -0.8094077  0.775
## 25:62 12.923649 20.870452 -1.1971576  0.890
## 32:37  1.517636  2.531108 -1.4884754  0.957
## 32:52  8.441198 12.660063 -0.8985559  0.795
## 32:62 10.066807 16.533220 -1.2875500  0.911
## 37:52  7.204876 10.581178 -0.8034993  0.761
## 37:62  8.617361 14.151604 -1.2577294  0.910
## 52:62  6.065070 10.786938 -1.5793897  0.993
```

PCA

8. Identify the major source of variation in data and determine if the variation is sourced from experimental bias or biological conditions.

```r
# Old Leaf Lipids (Neg) tune how many components to use
tune.pca(scaled_Y_old)
```

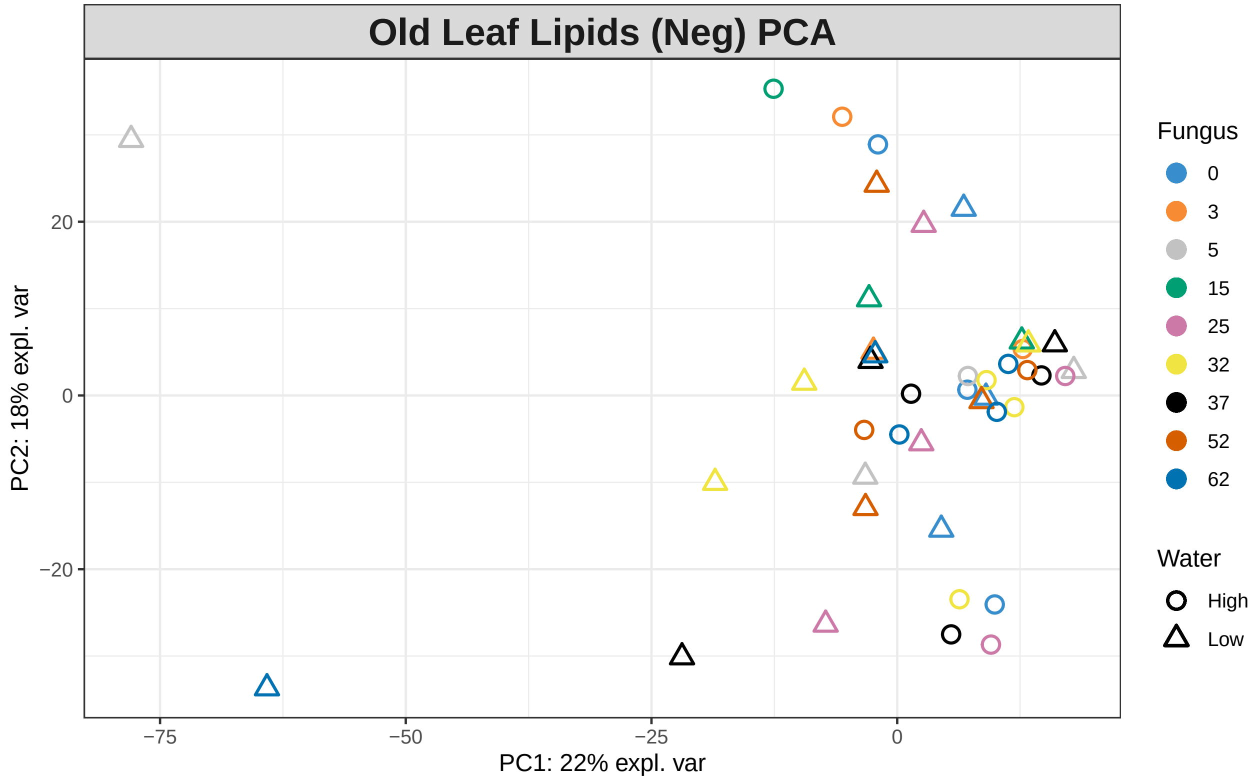
```
## Eigenvalues for the first 10 principal components, see object$sdev^2:
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7       PC8
## 338.73562 283.41789 172.57471  91.04209  61.76376  46.35468  38.50556  34.40119
##       PC9      PC10
##  28.91962  28.34847
##
## Proportion of explained variance for the first 10 principal components, see object$explained_varianc
##        PC1        PC2        PC3        PC4        PC5        PC6        PC7
## 0.21995819 0.18403759 0.11206150 0.05911824 0.04010634 0.03010044 0.02500361
##        PC8        PC9       PC10
## 0.02233843 0.01877898 0.01840810
##
## Cumulative proportion explained variance for the first 10 principal components, see object$cum.var:
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7       PC8
## 0.2199582 0.4039958 0.5160573 0.5751755 0.6152819 0.6453823 0.6703859 0.6927243
##       PC9      PC10
## 0.7115033 0.7299114
##
##  Other available components:
##  --------------------
##  loading vectors: see object$rotation
```



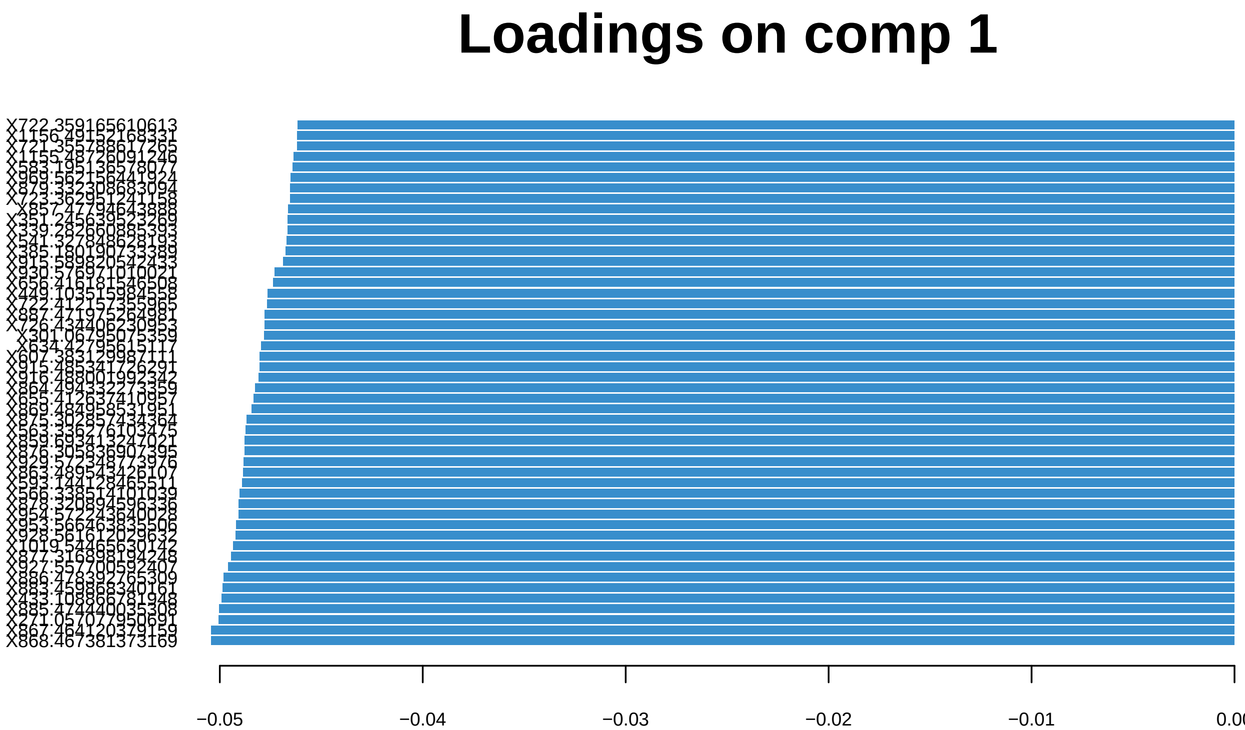

```r
pca.res <- mixOmics::pca(scaled_Y_old, ncomp = 4, scale = F)

# plot pca
```

```r
plotIndiv(pca.res, group = class$Fungus, ind.names = F, pch = as.factor(class$Water),
    legend = T, legend.title = "Fungus", legend.title.pch = "Water",
    title = "Old Leaf Lipids (Neg) PCA")
```

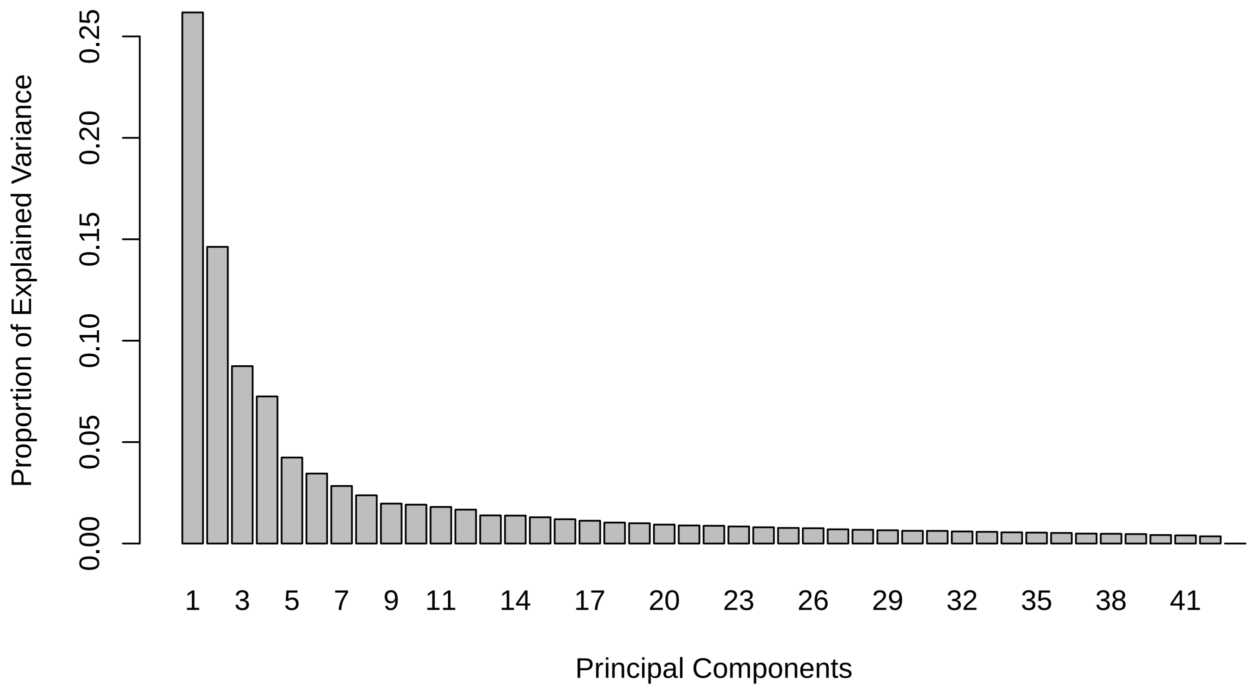



```r
# Look at variable coefficients in each component with the
# loading vectors The absolute value of loading vectors
# represent the importance of each variable to define each PC
plotLoadings(pca.res, ndisplay = 50)
```



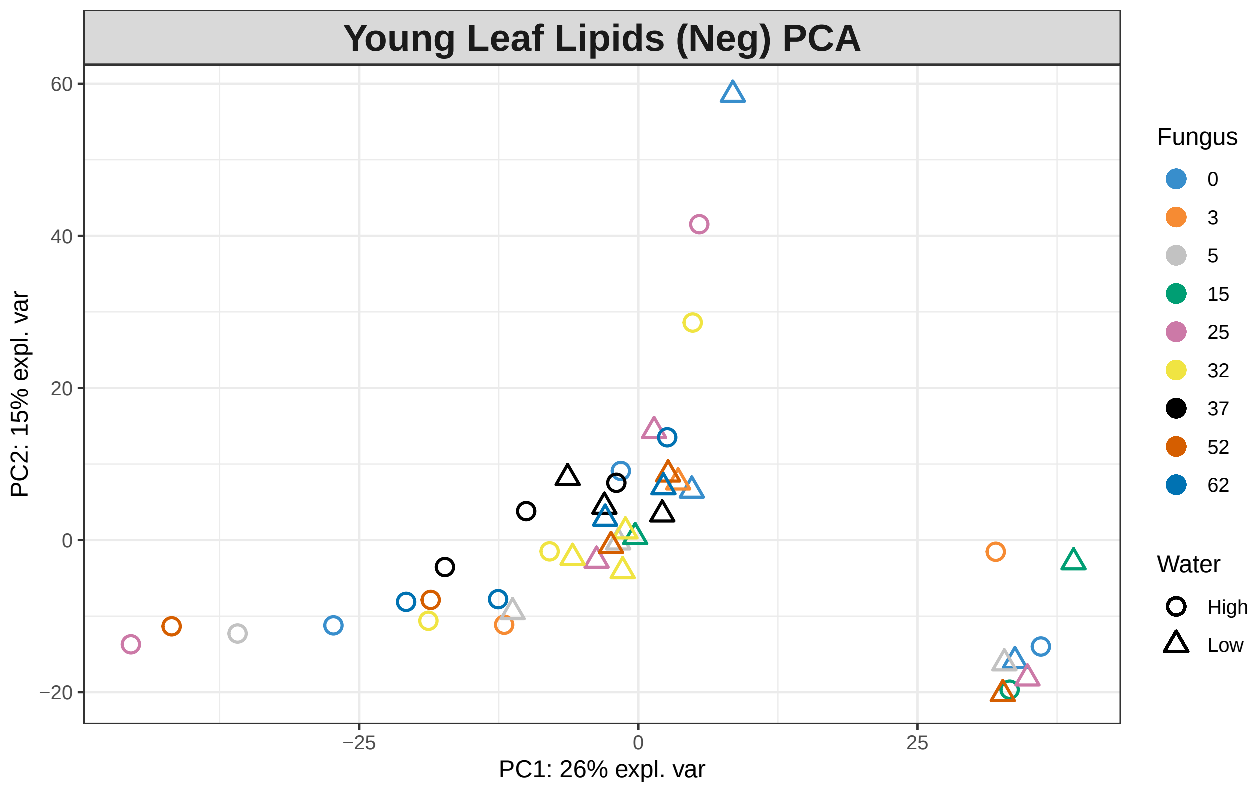

```r
# Young Leaf Lipids (Neg) tune how many components to use
tune.pca(scaled_Y_young)
```

```
## Eigenvalues for the first 10 principal components, see object$sdev^2:
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7       PC8
## 419.68951 234.44766 140.18419 116.22791  67.96706  55.30370  45.45138  38.09774
##       PC9      PC10
##  31.53786  30.70965
##
## Proportion of explained variance for the first 10 principal components, see object$explained_varianc
##        PC1        PC2        PC3        PC4        PC5        PC6        PC7
## 0.26181504 0.14625556 0.08745115 0.07250649 0.04239991 0.03450012 0.02835395
##        PC8        PC9       PC10
## 0.02376652 0.01967427 0.01915761
##
## Cumulative proportion explained variance for the first 10 principal components, see object$cum.var:
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7       PC8
## 0.2618150 0.4080706 0.4955218 0.5680282 0.6104282 0.6449283 0.6732822 0.6970488
##       PC9      PC10
## 0.7167230 0.7358806
##
##  Other available components:
##  --------------------
##  loading vectors: see object$rotation
```

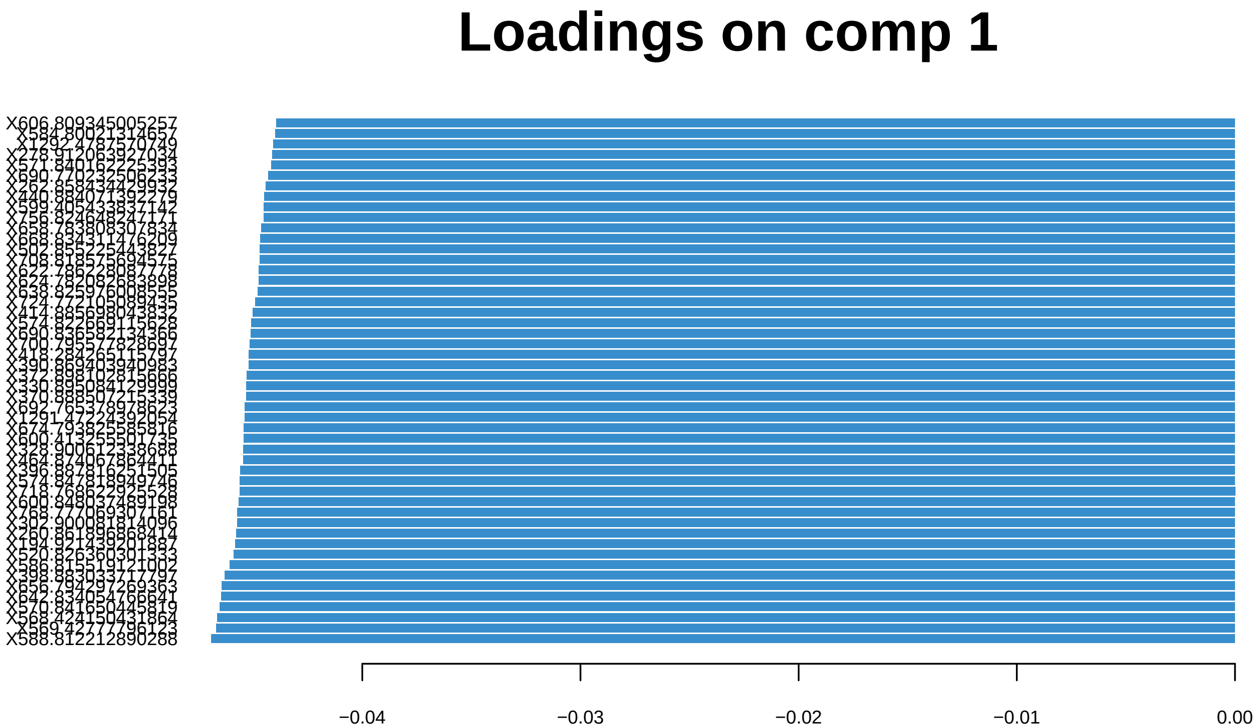



```r
pca.res <- mixOmics::pca(scaled_Y_young, ncomp = 3, scale = F)

# plot pca
plotIndiv(pca.res, group = class$Fungus, ind.names = F, pch = as.factor(class$Water),
    legend = T, legend.title = "Fungus", legend.title.pch = "Water",
    title = "Young Leaf Lipids (Neg) PCA")
```



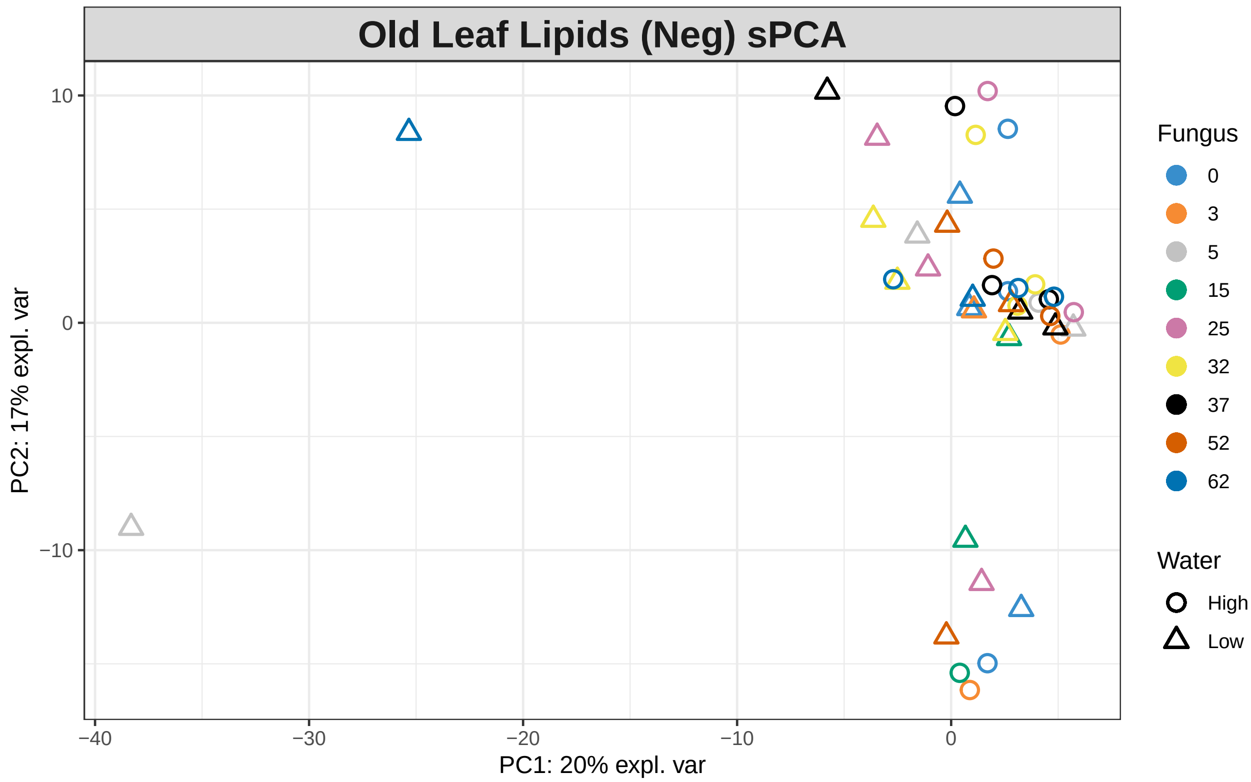
```r
# Look at variable coefficients in each component with the
# loading vectors The absolute value of loading vectors
# represent the importance of each variable to define each PC
plotLoadings(pca.res, ndisplay = 50)
```



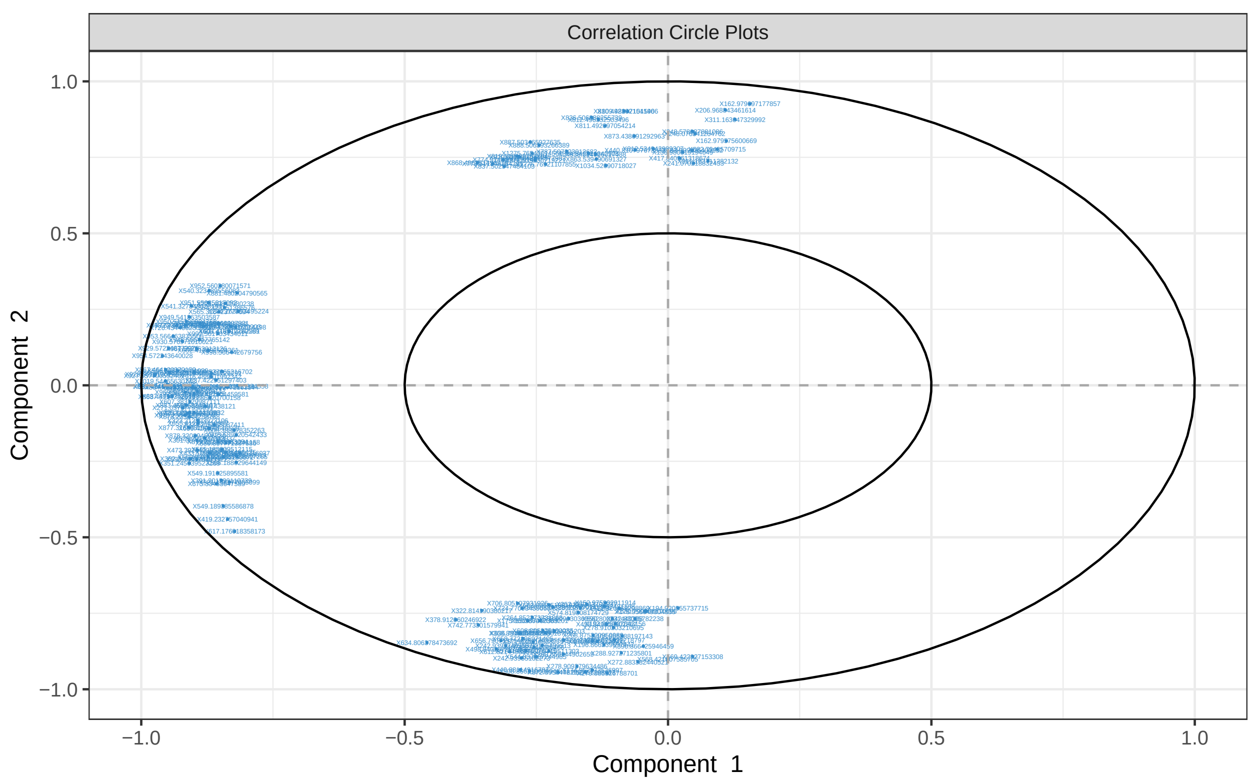
9. Identify the variables that contribute to the explained variance.

```r
# Old Leaf
spca.res <- mixOmics::spca(scaled_Y_old, ncomp = 4, keepX = c(100,
    100, 10, 10))

# plot spca
plotIndiv(spca.res, group = class$Fungus, ind.names = F, pch = as.factor(class$Water),
    legend = T, legend.title = "Fungus", legend.title.pch = "Water",
    title = "Old Leaf Lipids (Neg) sPCA")
```




```r
# variables contributing to each component
plotVar(spca.res, cex = 1)
```

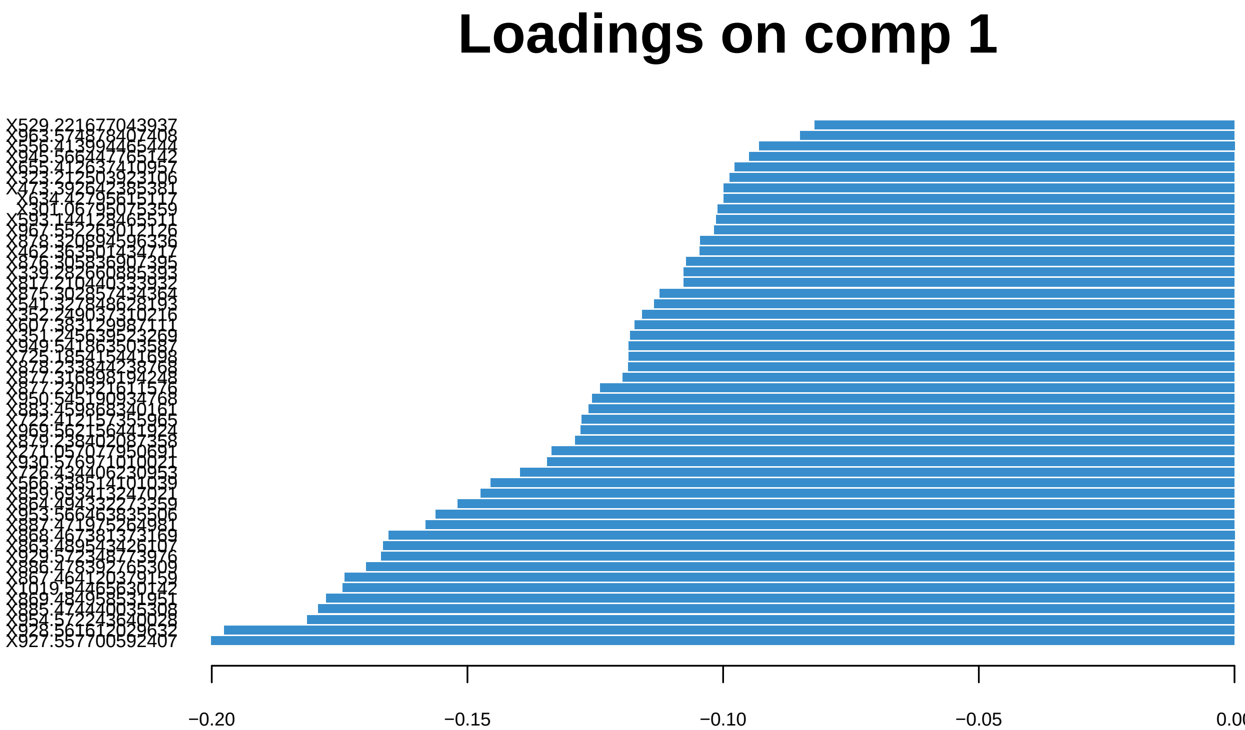


```r
selectVar(spca.res, comp = 1)$value  # view loading value of each metabolite
```

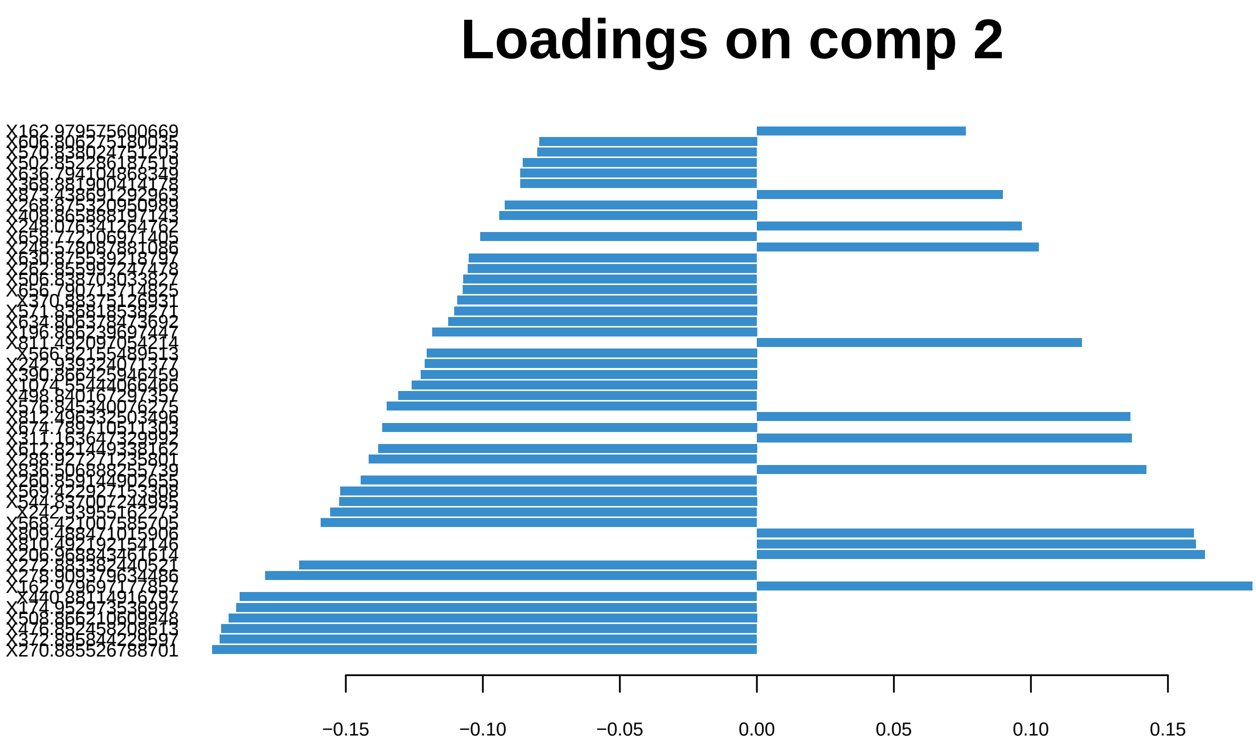
```
##                        value.var
## X927.557700592407 -0.200129024
## X928.561612029632 -0.197546772
## X954.572243640028 -0.181364972
## X885.474440035308 -0.179181509
## X869.484958531951 -0.177636767
## X1019.54465630142 -0.174351431
## X867.464120379159 -0.173962220
## X886.478392765309 -0.169780741
## X929.572348773976 -0.166888358
## X863.489543426107 -0.166493126
## X868.467381373169 -0.165440477
## X887.471975264981 -0.158163861
## X953.566463835506 -0.156163460
## X864.494332273359 -0.151929159
## X859.693413247021 -0.147434503
## X566.338514101039 -0.145410706
## X726.434406230953 -0.139649889
## X930.576971010021 -0.134385202
## X271.057077950691 -0.133499870
## X879.238402087358 -0.128877411
## X969.562156441924 -0.127875447
## X722.412157355965 -0.127653579
## X883.459868340161 -0.126325284
## X950.545190934768 -0.125545884
## X877.230321611576 -0.124019635
## X877.316898194248 -0.119613512
## X878.233844238768 -0.118591646
## X725.185415441698 -0.118457157
## X949.541863503587 -0.118439810
## X351.245639523269 -0.118133898
## X607.383129987111 -0.117305588
## X352.249037310216 -0.115822041
## X541.327848628193 -0.113514255
## X875.302857434364 -0.112398294
## X817.210440333932 -0.107730043
## X339.282660885393 -0.107706405
## X876.305836907395 -0.107230853
## X462.363501434717 -0.104597413
## X878.320894596336 -0.104444881
## X967.552263012126 -0.101782798
## X593.144128465511 -0.101302939
## X301.06795075359  -0.101030712
## X634.42795615117  -0.099879512
## X473.392642385381 -0.099871489
## X323.212503923106 -0.098713804
## X655.412637410957 -0.097764519
## X945.566447765142 -0.094896284
## X556.413994465444 -0.092987695
## X963.574878407408 -0.084933227
## X529.221677043937 -0.082089794
## X481.202981438121 -0.081378884
```

```
## X563.336276103475 -0.074493595
## X602.418692524361 -0.074409967
## X951.55635317092  -0.073939069
## X656.416181546508 -0.072708587
## X540.323489556083 -0.072268161
## X433.108866781948 -0.071262503
## X923.236620700158 -0.070057520
## X916.488001992342 -0.067765357
## X915.485341726291 -0.067193999
## X324.216268087411 -0.060551519
## X637.422951297403 -0.056338992
## X673.33453647589  -0.055056628
## X997.561703434011 -0.052421153
## X879.332308683094 -0.052343999
## X549.191625895581 -0.051897799
## X633.427504459581 -0.050900908
## X857.47794643888  -0.050475336
## X283.259468997921 -0.050069033
## X565.326423777694 -0.049311754
## X952.560280071571 -0.046534625
## X391.201599110739 -0.043685191
## X565.185009512115 -0.042462913
## X799.577065216702 -0.042284605
## X549.189385586878 -0.039567745
## X505.165901140672 -0.039414731
## X564.323251386576 -0.036617339
## X721.40842980238  -0.034558124
## X1065.55096405435 -0.033819246
## X880.337771387686 -0.033450733
## X482.20668851116  -0.032480717
## X419.232757040941 -0.029535342
## X713.529772163544 -0.028002491
## X608.416972280569 -0.026237761
## X607.413210162281 -0.025258474
## X521.158177035099 -0.025041234
## X723.362951241158 -0.024509449
## X619.413360639903 -0.022448051
## X998.566442679756 -0.019612586
## X608.388078352263 -0.013678125
## X617.176018358173 -0.012781306
## X894.543916014498 -0.010310911
## X722.359165610613 -0.010003806
## X566.188629644149 -0.008803619
## X915.589820542433 -0.008773624
## X721.355788617265 -0.007323427
## X881.480204790565 -0.007287137
## X449.103515984558 -0.005442569
## X284.263053495224 -0.004109811
## X861.234165316027 -0.001259647
```

```r
# plot loadings for comp 1
plotLoadings(spca.res, ndisplay = 50)
```

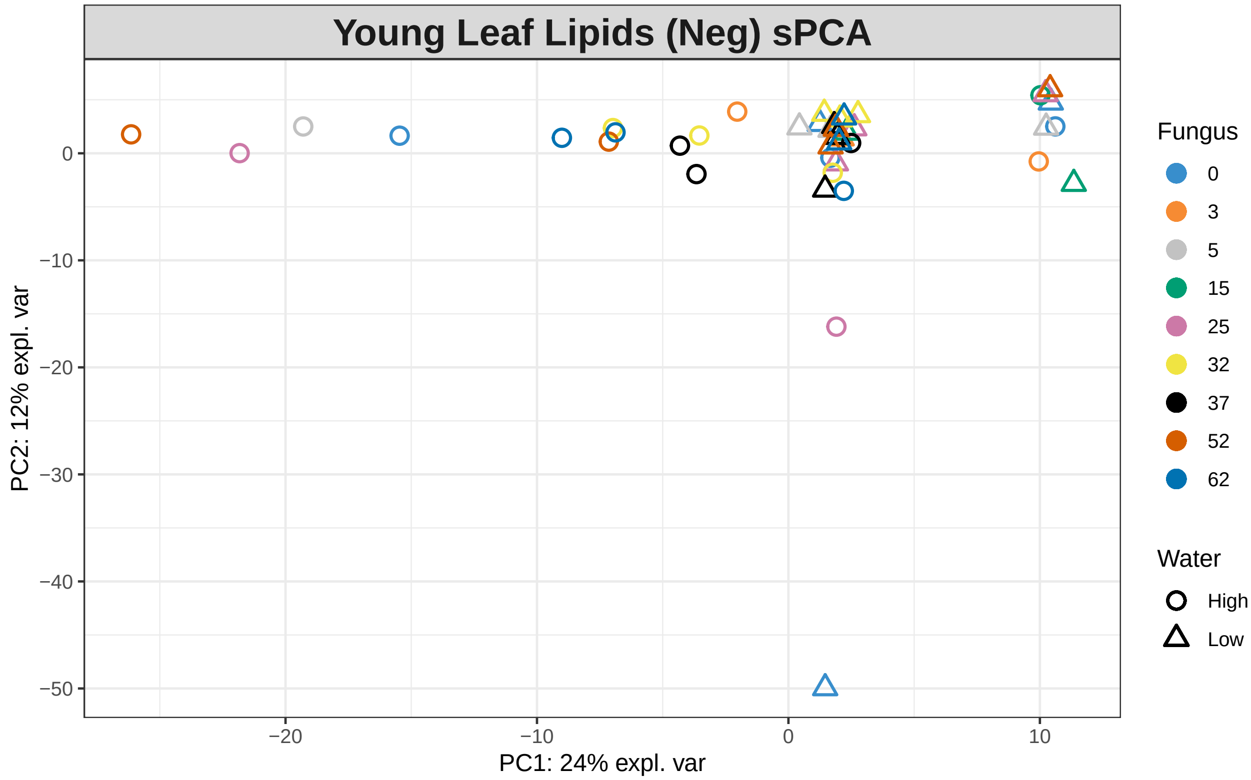



```r
# plot loadings for comp 2
plotLoadings(spca.res, comp = 2, ndisplay = 50)
```

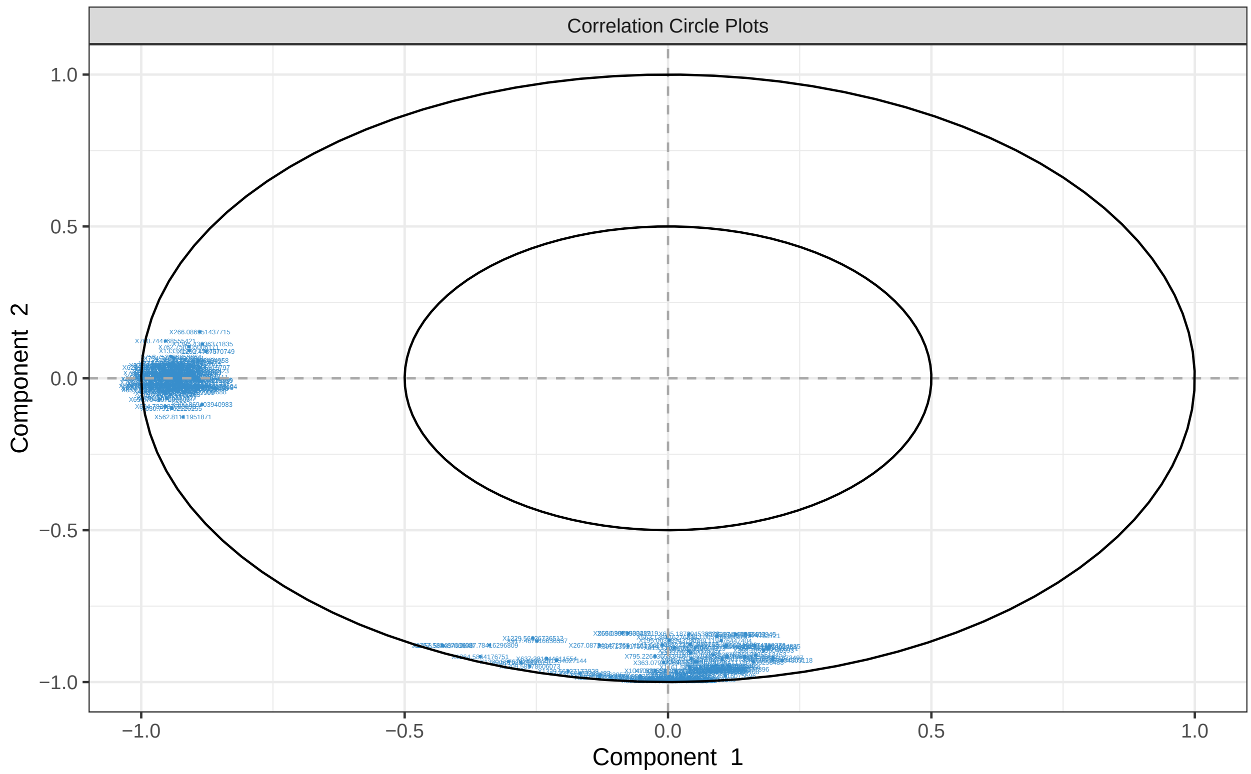



```r
# Young Leaf
spca.res <- mixOmics::spca(scaled_Y_young, ncomp = 3, keepX = c(100,
    100, 10))
```

```r
# plot spca
plotIndiv(spca.res, group = class$Fungus, ind.names = F, pch = as.factor(class$Water),
    legend = T, legend.title = "Fungus", legend.title.pch = "Water",
    title = "Young Leaf Lipids (Neg) sPCA")
```




```r
# variables contributing to each component
plotVar(spca.res, cex = 1)
```



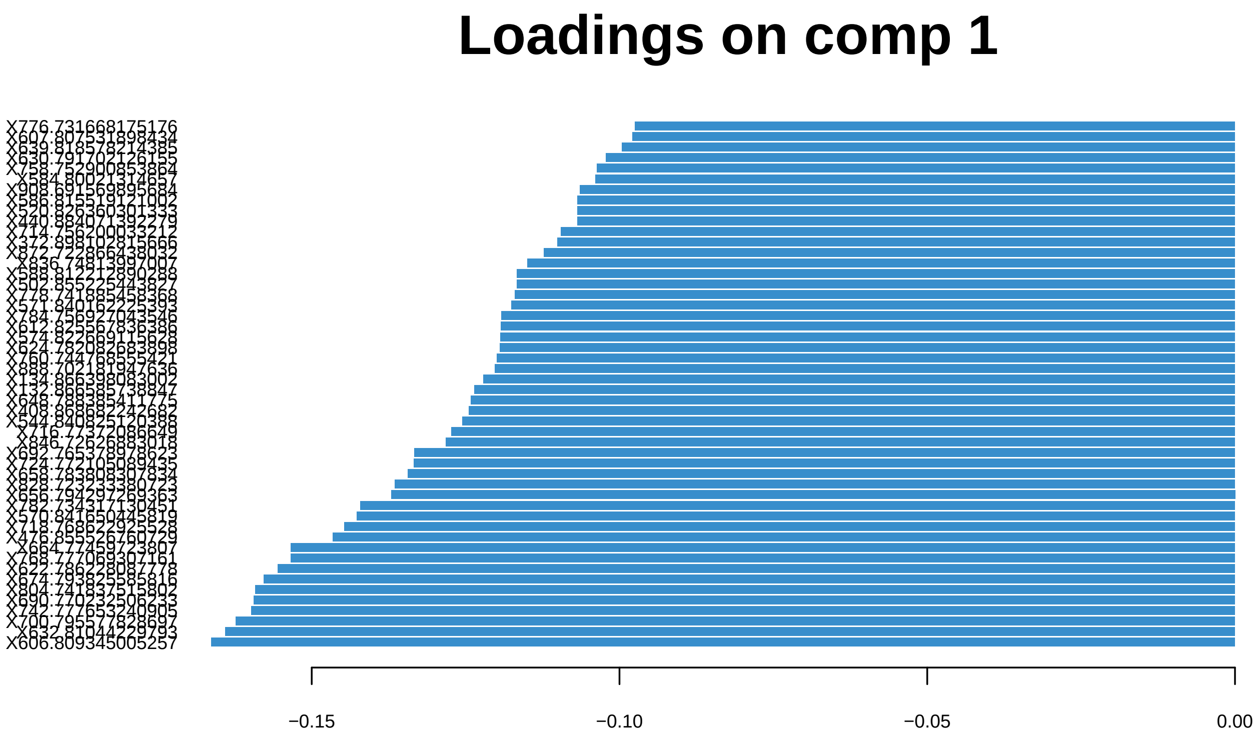

```
selectVar(spca.res, comp = 1)$value  # view loading value of each metabolite
```

```
##                      value.var
## X606.809345005257 -0.166332525
## X632.81044229793  -0.164033296
## X700.795577828697 -0.162314655
## X742.777653240905 -0.159823555
## X690.770232506233 -0.159441702
## X804.741837515802 -0.159152506
## X674.793825585816 -0.157780840
## X622.786228087778 -0.155511100
## X768.777069307161 -0.153378628
## X664.77459723807  -0.153359576
## X476.855526760729 -0.146572162
## X718.768622925528 -0.144715040
## X570.841650445819 -0.142629123
## X782.734317130451 -0.142079026
## X656.794297269363 -0.137096812
## X828.723233380723 -0.136495784
## X658.783808307834 -0.134358323
## X724.772105089435 -0.133418607
## X692.765378978623 -0.133304029
## X846.72626883018  -0.128205891
## X716.77372086649  -0.127284024
## X544.840825120388 -0.125507826
## X408.868682242682 -0.124423402
## X648.788385411775 -0.124154660
## X132.866585738847 -0.123533659
## X134.866398083002 -0.122096708
```

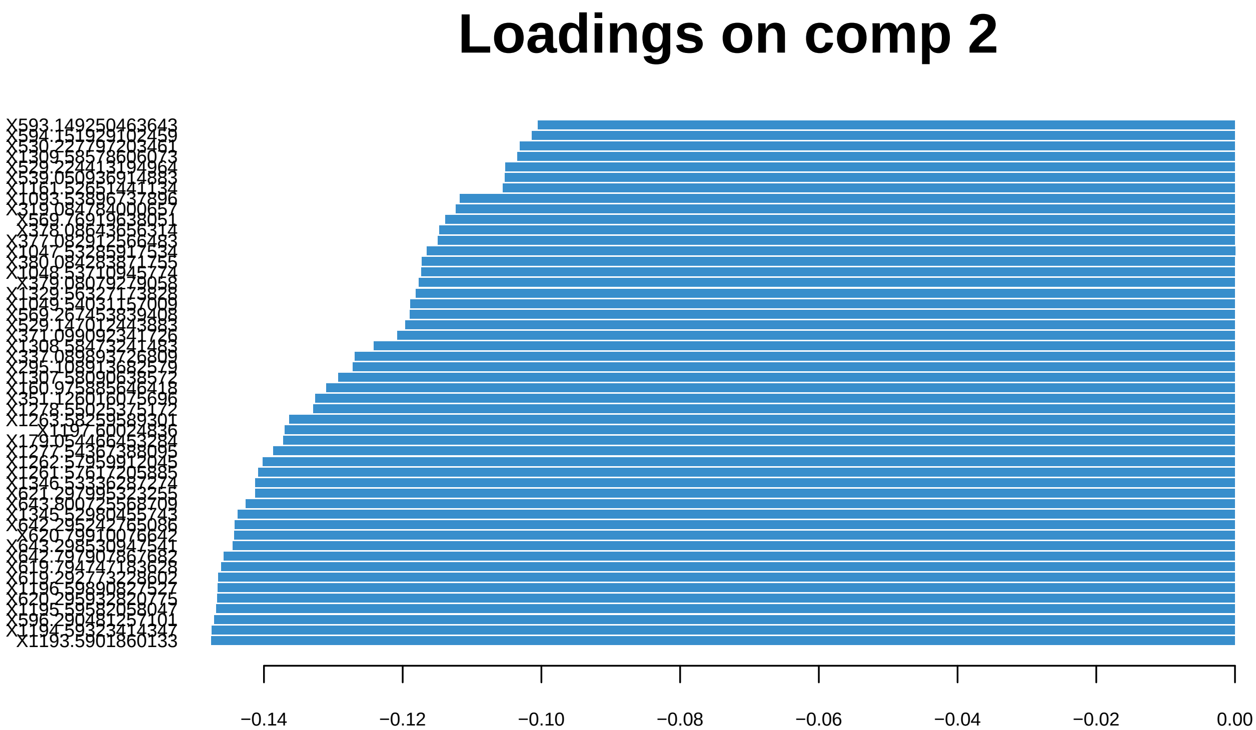
```
## X888.702181947636 -0.120228072
## X760.744768555421 -0.119932446
## X624.782082683898 -0.119434431
## X574.822669115628 -0.119342557
## X612.825567836386 -0.119295175
## X784.756927043546 -0.119176063
## X571.840162225393 -0.117561227
## X778.741885458368 -0.116951708
## X502.855225443827 -0.116641505
## X588.812212890288 -0.116620585
## X836.74813997007  -0.114936639
## X872.722866438032 -0.112241264
## X372.898102815666 -0.110050698
## X714.756200033212 -0.109481027
## X440.884071392279 -0.106852732
## X520.826360301333 -0.106806024
## X586.815519121002 -0.106783293
## X908.691569895684 -0.106421325
## X584.80021314657  -0.103905078
## X758.752900853864 -0.103689450
## X630.791702126155 -0.102151310
## X639.818578214385 -0.099602104
## X607.807531898434 -0.097912652
## X776.731668175176 -0.097490903
## X414.885698043832 -0.097005764
## X736.759279117259 -0.095809948
## X734.746504008811 -0.095232260
## X840.71244925072  -0.095067631
## X802.726364171284 -0.090732374
## X538.822954317218 -0.088564543
## X792.742406190328 -0.088486469
## X654.801893963813 -0.085999724
## X820.721435524574 -0.080331711
## X710.762932017726 -0.078194311
## X726.758924925928 -0.076854156
## X1291.47224392054 -0.075280342
## X386.860488067248 -0.074606708
## X650.783771288888 -0.073277302
## X786.745557215078 -0.070611027
## X562.81111951871  -0.069375629
## X330.895084129999 -0.066332121
## X192.927431485322 -0.063199397
## X666.770417878546 -0.062578639
## X633.807819066513 -0.060452417
## X638.825976008555 -0.060061860
## X278.912063927034 -0.059847620
## X396.887816251505 -0.055278165
## X762.758305006171 -0.052866742
## X1333.45977152431 -0.051188223
## X134.045412963815 -0.048529538
## X398.883033717797 -0.046987272
## X652.782727491955 -0.046883818
## X646.775033785773 -0.046681332
## X642.834054766641 -0.044180357
```

```
## X194.921439201887 -0.040248198
## X328.900612338688 -0.032131675
## X394.886566245554 -0.031544178
## X720.766074903034 -0.030350572
## X464.874067864411 -0.027062857
## X503.856090738483 -0.026237924
## X1177.48208484958 -0.025591727
## X871.562283488523 -0.024961052
## X708.75666274246  -0.024936629
## X596.796752072116 -0.024694133
## X322.883782987482 -0.022060861
## X418.284265115797 -0.021872289
## X266.086951437715 -0.021037380
## X574.847818949746 -0.015663369
## X390.869403940983 -0.014829995
## X370.888507215339 -0.013332876
## X1395.43636371835 -0.013203358
## X600.848037489198 -0.011855578
## X1292.4787570749  -0.003139404
## X190.926636178104 -0.001177666
```

```r
# plot loadings for comp 1
plotLoadings(spca.res, ndisplay = 50)
```




```r
# plot loadings for comp 2
plotLoadings(spca.res, comp = 2, ndisplay = 50)
```

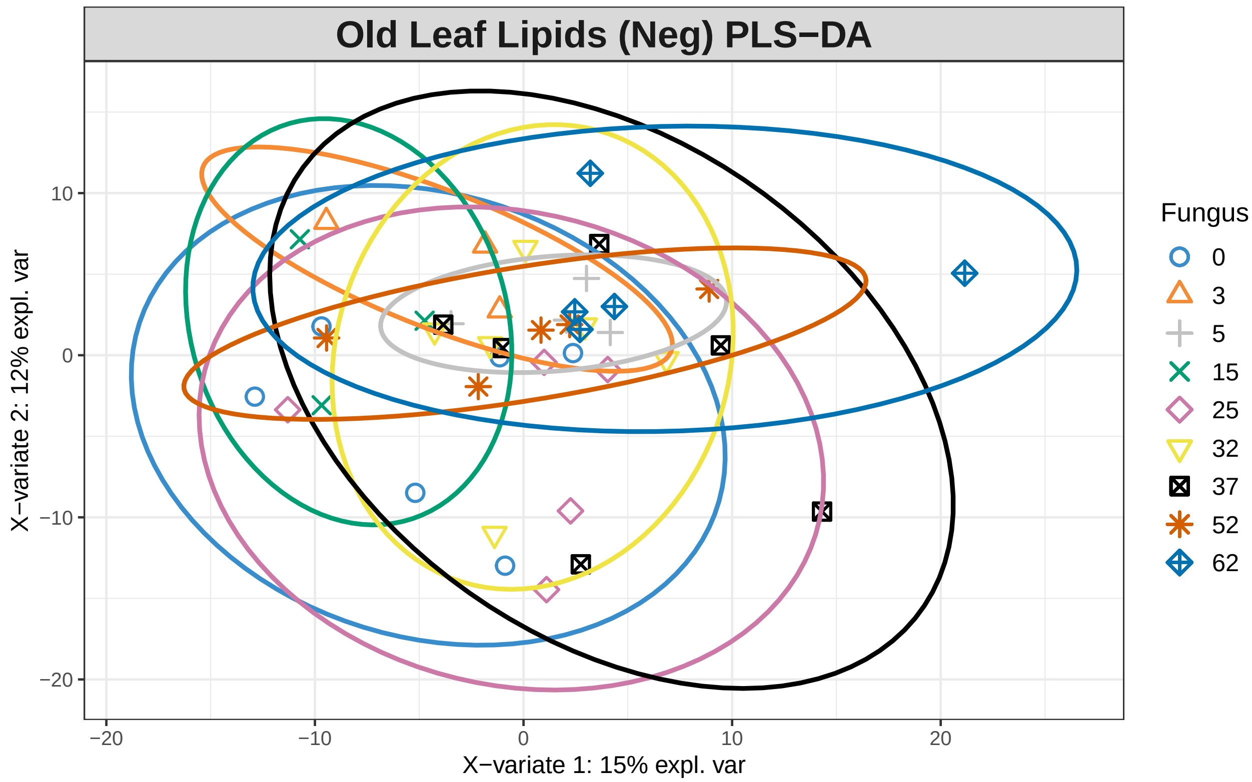


PLS-DA

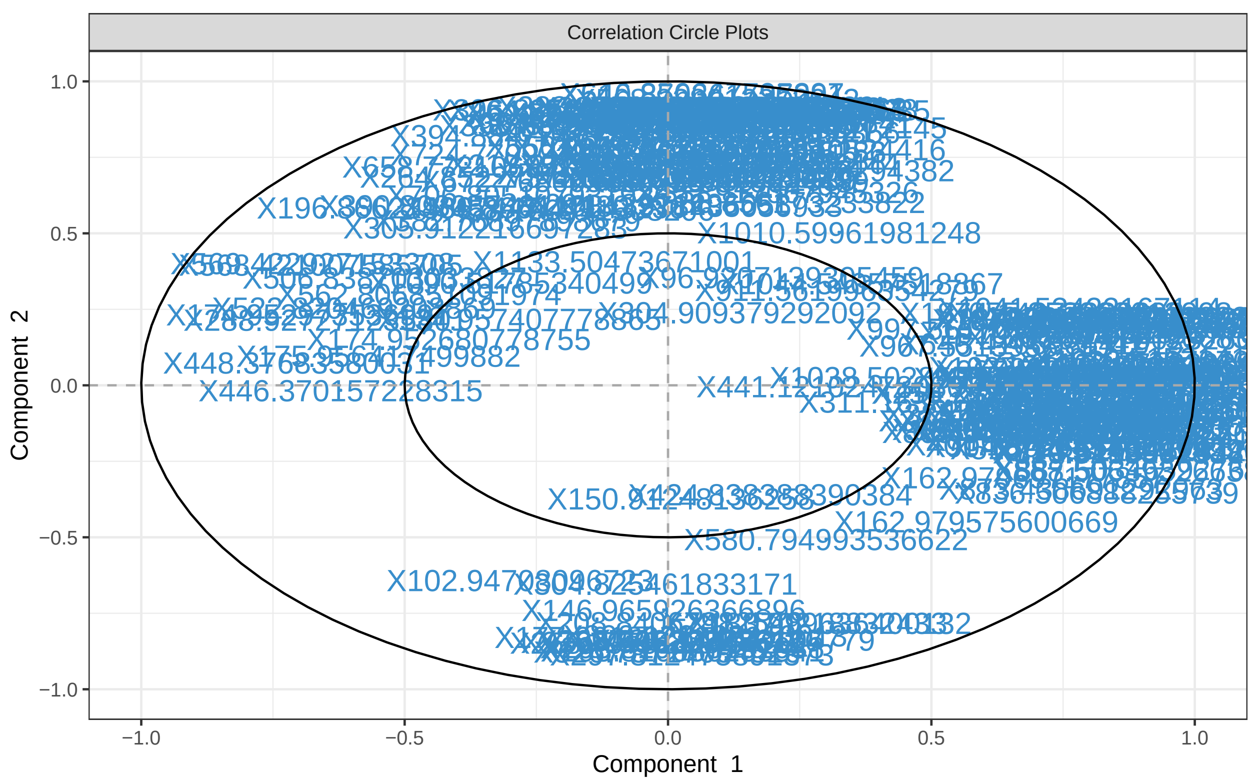
10. Classify samples into known groups and predict the class of new samples.

```r
# Old Leaf
old.splsda <- mixOmics::splsda(scaled_Y_old, class$Fungus, keepX = c(100,
    100))

# plot pls-da
plotIndiv(old.splsda, ind.names = F, legend = T, title = "Old Leaf Lipids (Neg) PLS-DA",
    legend.title = "Fungus", ellipse = T)
```




```r
# plot and select the variables
plotVar(old.splsda)
```



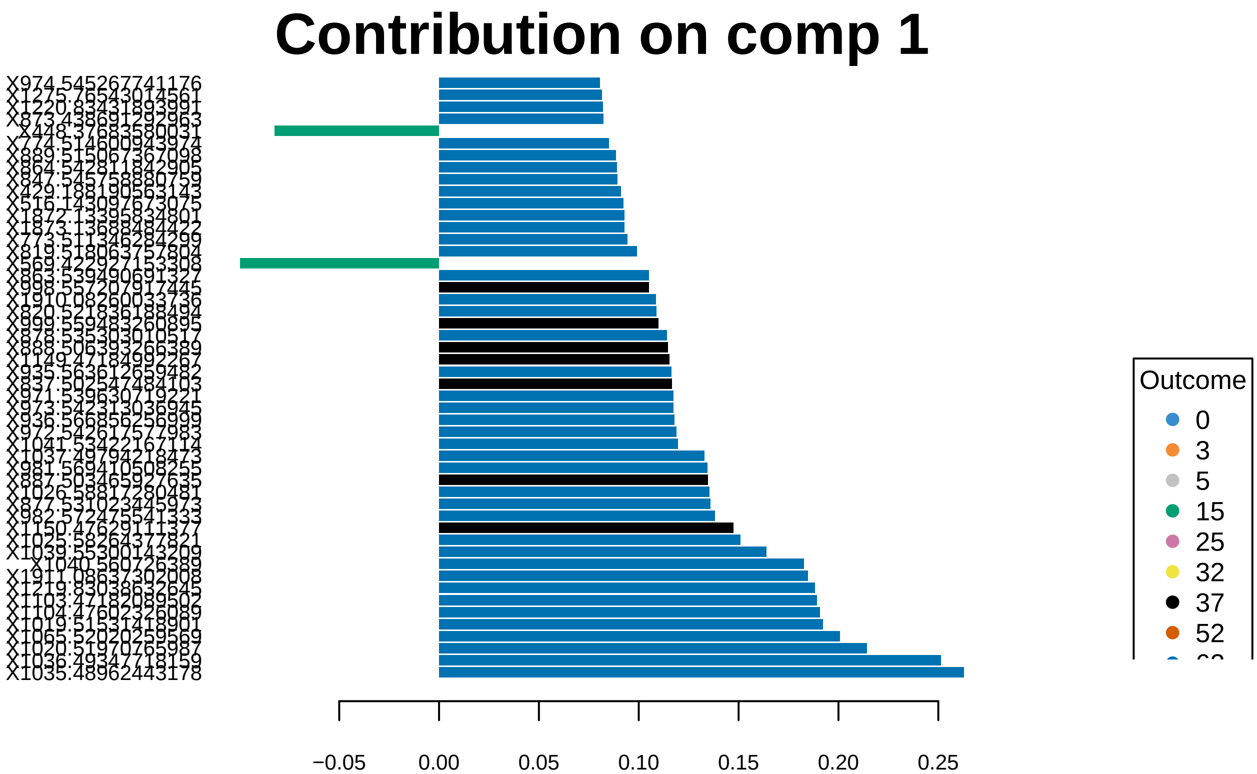
```r
selectVar(old.splsda, comp = 1)
```

```
## $name
##   [1] "X1035.48962443178" "X1036.49347718159" "X1020.51970765987"
##   [4] "X1065.52020259569" "X1019.51531418901" "X1104.47602326089"
##   [7] "X1103.47182089502" "X1219.83038632645" "X1911.08637302008"
##  [10] "X1040.560726389"   "X1039.55300143209" "X1025.58264377821"
##  [13] "X1150.47629111377" "X982.572475541333" "X877.531023445973"
##  [16] "X1026.58817280481" "X887.503465927635" "X981.569410508255"
##  [19] "X1037.49794218473" "X1041.53422167114" "X972.542617577983"
##  [22] "X936.566856256999" "X973.542313036945" "X971.539630719221"
##  [25] "X837.502547484103" "X935.563612659482" "X1149.47184992267"
##  [28] "X888.506393266389" "X878.535303010517" "X999.559483260895"
##  [31] "X820.521836188494" "X1910.08260033736" "X998.557207917445"
##  [34] "X863.539490691327" "X569.422927153308" "X819.518063757804"
##  [37] "X773.511346284299" "X1873.13688484422" "X1872.13395834801"
##  [40] "X516.143097673075" "X429.188190563143" "X847.545758880759"
##  [43] "X864.542811842905" "X889.515067367098" "X774.514600943974"
##  [46] "X448.37683580031"  "X873.438691292963" "X1220.83431893991"
##  [49] "X1275.76543014561" "X974.545267741176" "X401.157603916207"
##  [52] "X1034.52090718027" "X1549.03462530533" "X277.212850028684"
##  [55] "X175.956417499882" "X288.927271235801" "X1548.03166234851"
##  [58] "X1172.4614409432"  "X522.820469208869" "X848.548965929847"
##  [61] "X797.535600228753" "X675.352968613998" "X1550.03973700109"
##  [64] "X1276.76921107855" "X986.566025947242" "X1874.14077933693"
##  [67] "X386.188379924065" "X385.184945937595" "X162.979697177857"
##  [70] "X805.500384721158" "X311.163518476063" "X821.529032305061"
##  [73] "X441.121024780807" "X568.421007585705" "X515.140118980629"
##  [76] "X1066.52417490083" "X1261.75232985919" "X836.506888255739"
##  [79] "X174.952973536997" "X174.952680778755" "X134.957407778865"
##  [82] "X798.539218284514" "X506.838703033827" "X967.551395631042"
##  [85] "X967.552162080649" "X904.502555864161" "X1038.50285038777"
##  [88] "X751.527269681302" "X96.9207139305459" "X786.420127187216"
##  [91] "X752.530608553351" "X414.219499460298" "X446.370157228315"
##  [94] "X646.410608141015" "X256.231994916559" "X997.561703434011"
##  [97] "X799.547287485227" "X562.806851091974" "X162.979575600669"
## [100] "X835.510246875002"
## 
## $value
##                         value.var
## X1035.48962443178  2.627399e-01
## X1036.49347718159  2.513051e-01
## X1020.51970765987  2.143222e-01
## X1065.52020259569  2.007954e-01
## X1019.51531418901  1.920906e-01
## X1104.47602326089  1.906226e-01
## X1103.47182089502  1.890679e-01
## X1219.83038632645  1.881703e-01
## X1911.08637302008  1.846493e-01
## X1040.560726389    1.825775e-01
## X1039.55300143209  1.638365e-01
## X1025.58264377821  1.507778e-01
## X1150.47629111377  1.473418e-01
## X982.572475541333  1.381628e-01
```

```
## X877.531023445973  1.358959e-01
## X1026.58817280481  1.353119e-01
## X887.503465927635  1.345732e-01
## X981.569410508255  1.342745e-01
## X1037.49794218473  1.327396e-01
## X1041.53422167114  1.194617e-01
## X972.542617577983  1.187777e-01
## X936.566856256999  1.176803e-01
## X973.542313036945  1.174144e-01
## X971.539630719221  1.172818e-01
## X837.502547484103  1.166456e-01
## X935.563612659482  1.162204e-01
## X1149.47184992267  1.153189e-01
## X888.506393266389  1.146029e-01
## X878.535303010517  1.140008e-01
## X999.559483260895  1.098241e-01
## X820.521836188494  1.088266e-01
## X1910.08260033736  1.085166e-01
## X998.557207917445  1.050519e-01
## X863.539490691327  1.049808e-01
## X569.422927153308 -9.952883e-02
## X819.518063757804  9.891003e-02
## X773.511346284299  9.423627e-02
## X1873.13688484422  9.283331e-02
## X1872.13395834801  9.276950e-02
## X516.143097673075  9.217410e-02
## X429.188190563143  9.113610e-02
## X847.545758880759  8.926203e-02
## X864.542811842905  8.912992e-02
## X889.515067367098  8.855838e-02
## X774.514600943974  8.492731e-02
## X448.37683580031  -8.234819e-02
## X873.438691292963  8.219250e-02
## X1220.83431893991  8.199857e-02
## X1275.76543014561  8.158614e-02
## X974.545267741176  8.052994e-02
## X401.157603916207  7.362438e-02
## X1034.52090718027  7.132232e-02
## X1549.03462530533  6.788135e-02
## X277.212850028684  6.533014e-02
## X175.956417499882 -6.345424e-02
## X288.927271235801 -5.924263e-02
## X1548.03166234851  5.879000e-02
## X1172.4614409432   5.830235e-02
## X522.820469208869 -5.508273e-02
## X848.548965929847  5.493371e-02
## X797.535600228753  4.666910e-02
## X675.352968613998  4.662287e-02
## X1550.03973700109  4.524372e-02
## X1276.76921107855  4.386131e-02
## X986.566025947242  4.305194e-02
## X1874.14077933693  3.981875e-02
## X386.188379924065  3.819454e-02
## X385.184945937595  3.630213e-02
```

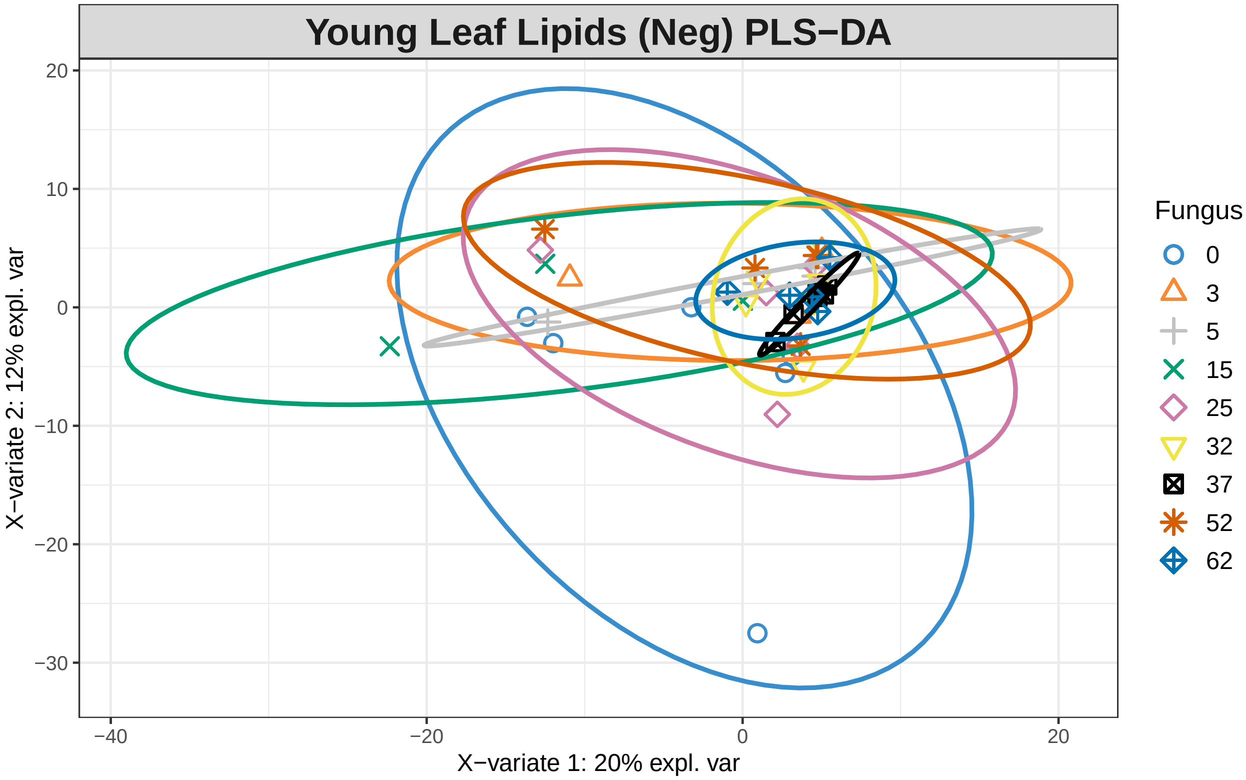
```
## X162.979697177857  3.319037e-02
## X805.500384721158  2.912366e-02
## X311.163518476063  2.870198e-02
## X821.529032305061  2.864186e-02
## X441.121024780807  2.846782e-02
## X568.421007585705 -2.785190e-02
## X515.140118980629  2.771866e-02
## X1066.52417490083  2.707963e-02
## X1261.75232985919  2.690965e-02
## X836.506888255739  2.628472e-02
## X174.952973536997 -2.594320e-02
## X174.952680778755 -2.571045e-02
## X134.957407778865 -2.538667e-02
## X798.539218284514  2.475376e-02
## X506.838703033827 -2.372125e-02
## X967.551395631042  2.332715e-02
## X967.552162080649  2.115915e-02
## X904.502555864161  2.035914e-02
## X1038.50285038777  1.723431e-02
## X751.527269681302  1.080361e-02
## X96.9207139305459  1.048247e-02
## X786.420127187216  9.797547e-03
## X752.530608553351  8.389689e-03
## X414.219499460298  5.647848e-03
## X446.370157228315 -5.260206e-03
## X646.410608141015  4.322310e-03
## X256.231994916559  4.038711e-03
## X997.561703434011  1.724372e-03
## X799.547287485227  1.699523e-03
## X562.806851091974 -1.521586e-03
## X162.979575600669  1.117011e-03
## X835.510246875002  6.719878e-05
##
## $comp
## [1] 1
```

```r
plotLoadings(old.splsda, contrib = "max", method = "mean", ndisplay = 50)
```

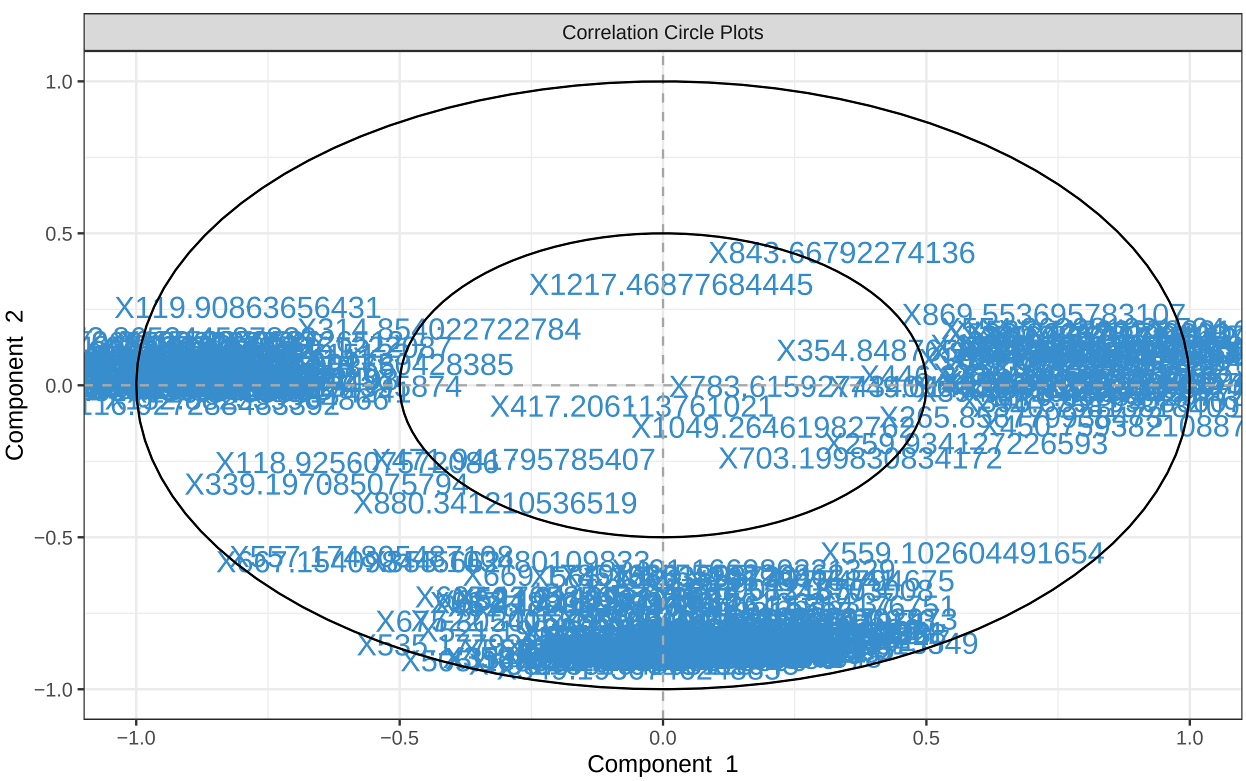



```r
# Young Leaf
young.splsda <- mixOmics::splsda(scaled_Y_young, class$Fungus,
    keepX = c(100, 100))

# plot pls-da
plotIndiv(young.splsda, ind.names = F, legend = T, title = "Young Leaf Lipids (Neg) PLS-DA",
    legend.title = "Fungus", ellipse = T)
```




```r
# plot and select the variables
plotVar(young.splsda)
```



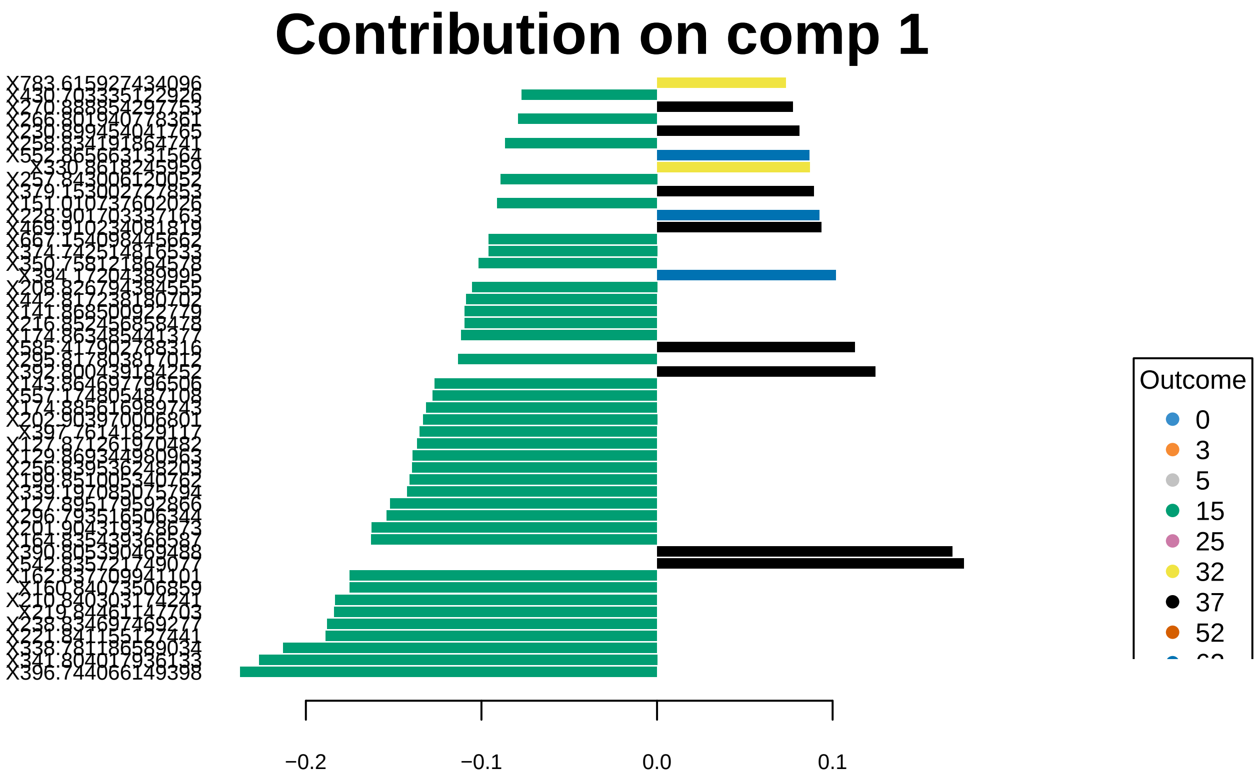
```r
selectVar(young.splsda, comp = 1)
```

```
## $name
##   [1] "X396.744066149398" "X341.804017936133" "X338.781186589034"
##   [4] "X221.841155127441" "X238.834697469277" "X219.84461147703"
##   [7] "X210.840303174241" "X160.84073506859"  "X162.837709941101"
##  [10] "X542.835721749077" "X390.805390469488" "X164.835439366587"
##  [13] "X201.904319378673" "X296.793516506344" "X127.895179592866"
##  [16] "X339.197085075794" "X199.851005340762" "X256.839536248203"
##  [19] "X129.869344980963" "X127.871261970482" "X397.76141829117"
##  [22] "X202.903970006801" "X174.885616989743" "X557.174805487108"
##  [25] "X143.864697796506" "X392.800439184252" "X295.817803817012"
##  [28] "X585.417902788316" "X174.863485441377" "X216.852456858478"
##  [31] "X141.868500922779" "X442.817238180702" "X208.826794384555"
##  [34] "X394.17204389995"  "X350.758121864578" "X374.742514816533"
##  [37] "X667.154098445662" "X469.910234081819" "X228.901703337163"
##  [40] "X151.010737602026" "X379.153002727853" "X257.843006120052"
##  [43] "X330.8618245959"   "X552.865663131564" "X258.834191864741"
##  [46] "X230.899454041765" "X266.801940778361" "X270.888854297753"
##  [49] "X430.703335122926" "X783.615927434096" "X255.828026038127"
##  [52] "X259.934127226593" "X450.759382108877" "X312.857261222087"
##  [55] "X172.865244587839" "X116.927288483392" "X293.821571674538"
##  [58] "X869.553695783107" "X584.415323461393" "X270.795957188362"
##  [61] "X119.90863656431"  "X548.787426737795" "X354.848765964605"
##  [64] "X314.854022722784" "X488.659982912674" "X109.003224395874"
##  [67] "X268.799154723268" "X232.762175151337" "X316.783455451835"
##  [70] "X152.882339964866" "X323.884189924197" "X314.784985265431"
##  [73] "X286.860667596866" "X325.014908362498" "X510.8792724562"
##  [76] "X474.851017726839" "X170.866558879586" "X240.926846526528"
##  [79] "X1028.5795946312"  "X355.103480109833" "X583.410941115392"
##  [82] "X442.891741260384" "X540.758138184099" "X372.742954316427"
##  [85] "X749.614838652779" "X840.786703094097" "X396.835199086007"
##  [88] "X109.003013726512" "X620.852616467634" "X336.786501994141"
##  [91] "X554.862981021704" "X940.601750428385" "X102.95537878551"
##  [94] "X265.856779938473" "X716.836524572973" "X646.853854191413"
##  [97] "X588.893578190363" "X1150.48670587775" "X444.888177851031"
## [100] "X446.373066658593"
##
## $value
##                     value.var
## X396.744066149398 -0.237346099
## X341.804017936133 -0.226712611
## X338.781186589034 -0.212839599
## X221.841155127441 -0.188603224
## X238.834697469277 -0.187887930
## X219.84461147703  -0.183754137
## X210.840303174241 -0.183293976
## X160.84073506859  -0.175122234
## X162.837709941101 -0.174967631
## X542.835721749077  0.174675387
## X390.805390469488  0.168096512
## X164.835439366587 -0.162856507
## X201.904319378673 -0.162392668
## X296.793516506344 -0.153997773
```

```
## X127.895179592866 -0.152040830
## X339.197085075794 -0.142236059
## X199.851005340762 -0.140928992
## X256.839536248203 -0.139463873
## X129.869344980963 -0.139035299
## X127.871261970482 -0.136633914
## X397.76141829117  -0.135226336
## X202.903970006801 -0.133281705
## X174.885616989743 -0.131405620
## X557.174805487108 -0.127847482
## X143.864697796506 -0.126591305
## X392.800439184252  0.124418002
## X295.817803817012 -0.113167199
## X585.417902788316  0.112654838
## X174.863485441377 -0.111515640
## X216.852456858478 -0.109465643
## X141.868500922779 -0.109408381
## X442.817238180702 -0.108616673
## X208.826794384555 -0.105398768
## X394.17204389995   0.101902740
## X350.758121864578 -0.101454603
## X374.742514816533 -0.096021798
## X667.154098445662 -0.095905408
## X469.910234081819  0.093491653
## X228.901703337163  0.092339950
## X151.010737602026 -0.091016171
## X379.153002727853  0.089268814
## X257.843006120052 -0.089182296
## X330.8618245959    0.087029685
## X552.865663131564  0.086750537
## X258.834191864741 -0.086420449
## X230.899454041765  0.081098315
## X266.801940778361 -0.078957710
## X270.888854297753  0.077217475
## X430.703335122926 -0.077089520
## X783.615927434096  0.073302432
## X255.828026038127 -0.070733902
## X259.934127226593  0.067820144
## X450.759382108877  0.067621153
## X312.857261222087 -0.067451324
## X172.865244587839 -0.065297269
## X116.927288483392 -0.064666845
## X293.821571674538 -0.061532091
## X869.553695783107  0.060766086
## X584.415323461393  0.056123672
## X270.795957188362 -0.055077183
## X119.90863656431  -0.054443833
## X548.787426737795  0.054088532
## X354.848765964605  0.053425957
## X314.854022722784 -0.045246971
## X488.659982912674 -0.043988685
## X109.003224395874 -0.043356673
## X268.799154723268 -0.041148826
## X232.762175151337 -0.039610014
```

```
## X316.783455451835 -0.036219106
## X152.882339964866 -0.036147621
## X323.884189924197 -0.035881662
## X314.784985265431 -0.032139638
## X286.860667596866  0.032071960
## X325.014908362498 -0.031772580
## X510.8792724562    0.031126297
## X474.851017726839  0.030329181
## X170.866558879586 -0.027043350
## X240.926846526528  0.025879617
## X1028.5795946312   0.023187879
## X355.103480109833 -0.022220089
## X583.410941115392  0.021770971
## X442.891741260384  0.019491794
## X540.758138184099  0.019226779
## X372.742954316427 -0.016650713
## X749.614838652779  0.015099968
## X840.786703094097  0.012489160
## X396.835199086007  0.008285053
## X109.003013726512 -0.007789797
## X620.852616467634  0.007220824
## X336.786501994141 -0.007203476
## X554.862981021704  0.006193023
## X940.601750428385 -0.004406166
## X102.95537878551   0.004031028
## X265.856779938473  0.002904594
## X716.836524572973  0.002813464
## X646.853854191413  0.002501612
## X588.893578190363  0.002479180
## X1150.48670587775 -0.001761621
## X444.888177851031  0.001537349
## X446.373066658593  0.001380297
##
## $comp
## [1] 1
```

```r
plotLoadings(young.splsda, contrib = "max", method = "mean",
    ndisplay = 50)
```



Heatmaps of Averaged Data

11. Create averaged metabolite matrices and rerun PLS-DA to create a heatmap.

```
## Young Leaves
av_Y_young <- aggregate(Y_young, by = list(class$Water, class$Fungus),
    FUN = "mean", simplify = T, data = class)
av.young.plsda <- mixOmics::plsda(av_Y_young[, 3:1605], av_Y_young$Group.2)  # fungus

# heatmap
youngcim <- cim(av.young.plsda, title = "Young Leaf Lipids (neg) Averaged Over Fungi",
    col.names = F, xlab = "Lipids", save = "png", name.save = "~/Box/Summer 2018 TX Endo Field Samples
```

Indicator Analysis

12. Identify indicator metabolites characteristic of each treatment using Dufrene-Legendre Indicator Analysis.

```
# Old Leaf
indicator_Fungus <- indval(Y_young, clustering = class$Fungus,
    numitr = 999, type = "long")

# Young Leaf
indicator_Fungus <- indval(Y_young, clustering = class$Fungus,
    numitr = 999, type = "long")
```

13. Disect indval object.

```
Orelfrq <- indicator_Fungus$relfrq  # relative frequency of species in classes
Orelabu <- indicator_Fungus$relabu  # relative abundance of species in classes
Oindval <- indicator_Fungus$indval  # the indicator value for each species
```

```
Omaxcls <- data.frame(indicator_Fungus$maxcls)  # the class each species has max indicator value for
Oindcls <- data.frame(indicator_Fungus$indcls)  # the indicator value for each species to its max class
Opval <- data.frame(indicator_Fungus$pval)  # the probability of obtaining as high an indicator value a

Yrelfrq <- indicator_Fungus$relfrq  # relative frequency of species in classes
Yrelabu <- indicator_Fungus$relabu  # relative abundance of species in classes
Yindval <- indicator_Fungus$indval  # the indicator value for each species
Ymaxcls <- data.frame(indicator_Fungus$maxcls)  # the class each species has max indicator value for
Yindcls <- data.frame(indicator_Fungus$indcls)  # the indicator value for each species to its max class
Ypval <- data.frame(indicator_Fungus$pval)  # the probability of obtaining as high an indicator value a
```

14. Export results to a csv file.

```
write.csv(cbind(Orelfrq, Orelabu, Oindval, Omaxcls, Oindcls,
    Opval), "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/Lipi
write.csv(cbind(Yrelfrq, Yrelabu, Yindval, Ymaxcls, Yindcls,
    Ypval), "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/Lipi
```

Lipids (Pos)

RRPP

2. Define dependent variable matrix and class matrix.

```
path <- "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/"

O_L_pos <- read_tsv(paste(path, "XCMS Online Results/O_L_Pos/XCMS.annotated.Report_1394374.tsv",
    sep = ""))

Y_L_pos <- read_tsv(paste(path, "XCMS Online Results/Y_L_Pos/XCMS.annotated.Report_1394370.tsv",
    sep = ""))

# dependent variable: metabolite intensities
Y_old <- O_L_pos[, c(2, 12:54)] %>% data.frame(row.names = 1) %>%
    t %>% data.frame()
scaled_Y_old <- scale(Y_old)

Y_young <- Y_L_pos[, c(2, 12:54)] %>% data.frame(row.names = 1) %>%
    t %>% data.frame()
scaled_Y_young <- scale(Y_young)

# class: sample factors
class <- read.csv(paste(path, "XCMS Online Results/class.csv",
    sep = ""), header = T, row.names = 1)
```

3. Define and run multivariate regression models, then print out the results.

```
O_LMpos <- lm.rrpp(scaled_Y_old ~ Block * Water * Fungus, data = class,
    SS.type = "III", print.progress = F)
summary(O_LMpos)
```

```
##
## Linear Model fit with lm.rrpp
##
```

```
## Number of observations: 43
## Number of dependent variables: 1823
## Data space dimensions: 42
## Sums of Squares and Cross-products: Type III
## Number of permutations: 1000
##
## Full Model Analysis of Variance
##
##                          Df Residual Df       SS Residual SS       Rsq        F
## Block * Water * Fungus  7           35 20563.31    56002.69 0.2685698 1.835922
##                          Z (from F) Pr(>F)
## Block * Water * Fungus   2.761617  0.006
##
##
## Redundancy Analysis (PCA on fitted values and residuals)
##
##                Trace Proportion Rank
## Fitted     489.6027  0.2685698    7
## Residuals 1333.3973  0.7314302   35
## Total     1823.0000  1.0000000   42
##
## Eigenvalues
##
##                PC1      PC2      PC3      PC4      PC5      PC6      PC7
## Fitted    203.7511 142.4724  62.4085  28.1300  21.0207  17.0790  14.7411
## Residuals 402.1919 150.3132 125.0098  79.2707  62.6699  44.2508  39.8865
## Total     513.8609 258.2186 186.1442 115.4959  89.3826  51.6885  48.5305
##                PC8      PC9     PC10     PC11     PC12     PC13     PC14
## Fitted
## Residuals  36.5208  28.1299  27.5893  25.0897  23.4977  20.4194  19.0704
## Total      44.9001  35.5187  31.1739  28.2841  27.2468  25.7007  22.5417
##               PC15     PC16     PC17     PC18     PC19     PC20     PC21
## Fitted
## Residuals  18.2407  17.7419  16.7325  15.3389  14.4980  13.5948  12.9893
## Total      21.0600  19.6929  18.6159  18.1548  16.8559  16.0430  15.4500
##               PC22     PC23     PC24     PC25     PC26     PC27     PC28
## Fitted
## Residuals  12.7425  12.3680  11.8230  11.0709  10.9782  10.7037  10.2142
## Total      14.4329  13.4848  12.8637  12.8111  12.6443  12.1077  11.5810
##               PC29     PC30     PC31     PC32     PC33     PC34     PC35
## Fitted
## Residuals   9.4416   9.3422   9.2476   8.8057   8.3472   7.7692   7.4973
## Total      11.3027  10.6443  10.6252  10.1859   9.6192   9.4706   9.3261
##               PC36     PC37     PC38     PC39     PC40     PC41     PC42
## Fitted
## Residuals
## Total       9.0376   8.9028   8.4398   8.1941   8.0756   7.4086   7.2823
```

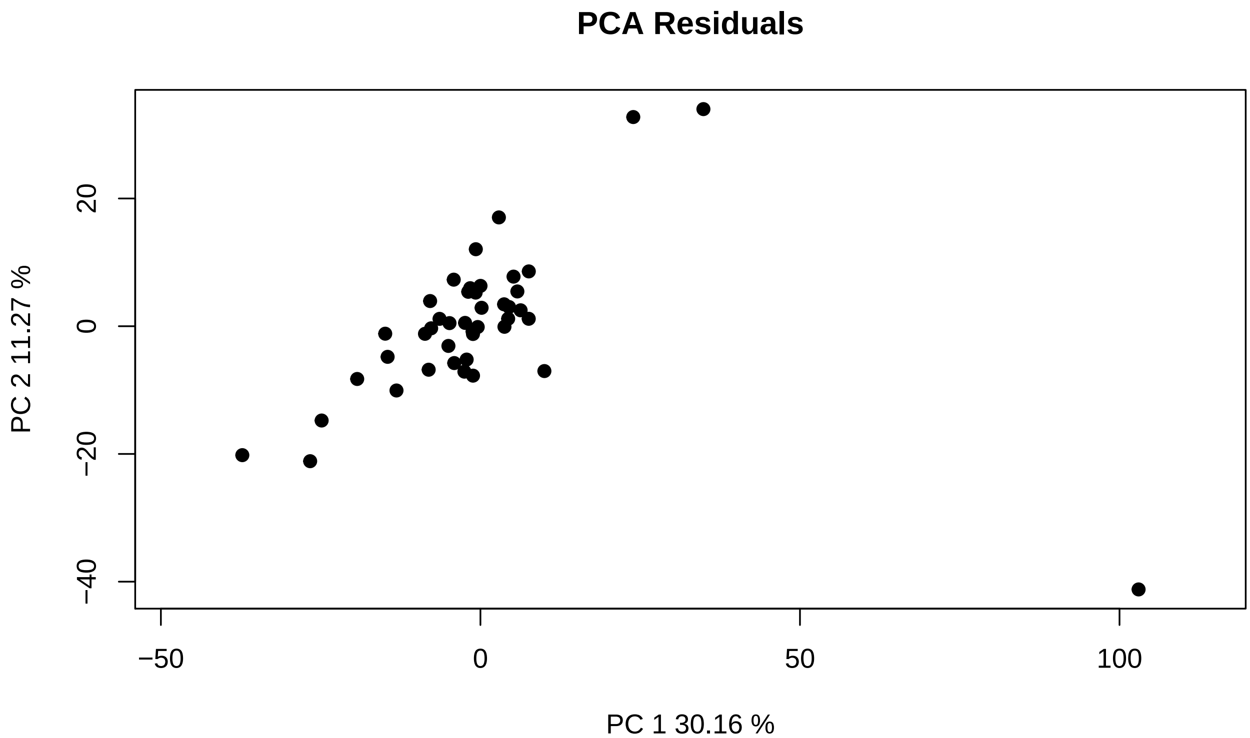
```r
Y_LMpos <- lm.rrpp(scaled_Y_young ~ Block * Water * Fungus, data = class,
    SS.type = "III", print.progress = F)
summary(Y_LMpos)
```

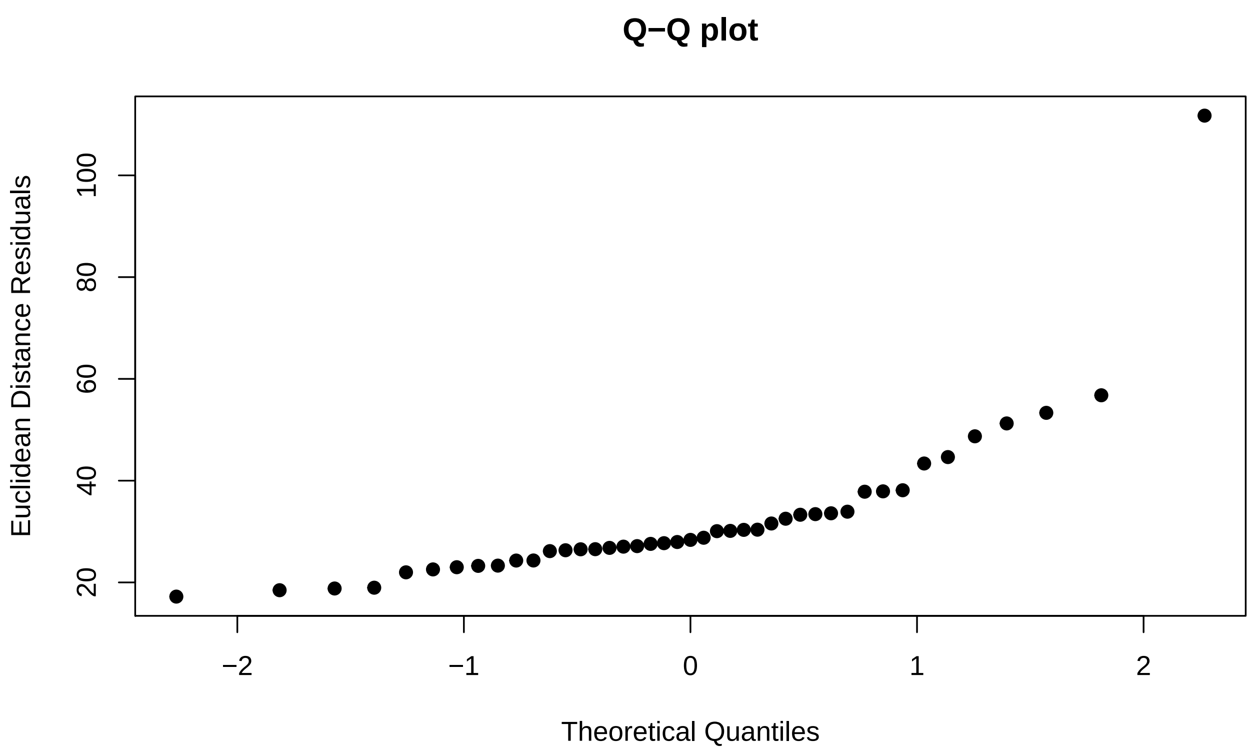
```
##
## Linear Model fit with lm.rrpp
##
```

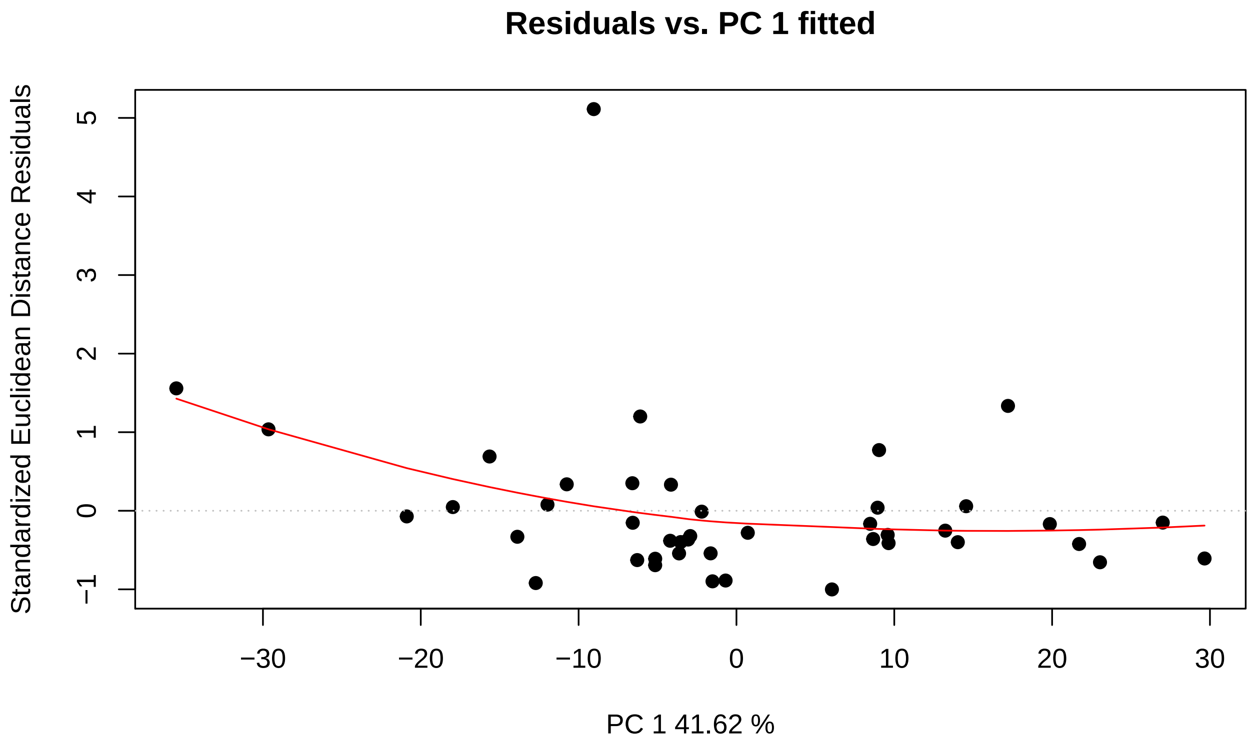
```
## Number of observations: 43
## Number of dependent variables: 1753
## Data space dimensions: 42
## Sums of Squares and Cross-products: Type III
## Number of permutations: 1000
##
## Full Model Analysis of Variance
##
##                      Df Residual Df       SS Residual SS       Rsq        F
## Block * Water * Fungus  7          35 16697.86    56928.14 0.2267929 1.466573
##                      Z (from F) Pr(>F)
## Block * Water * Fungus   2.839574  0.007
##
##
## Redundancy Analysis (PCA on fitted values and residuals)
##
##              Trace Proportion Rank
## Fitted     397.568  0.2267929    7
## Residuals 1355.432  0.7732071   35
## Total     1753.000  1.0000000   42
##
## Eigenvalues
##
##                PC1      PC2      PC3      PC4      PC5      PC6      PC7
## Fitted    142.6474  84.0019  50.0644  42.8687  29.4336  25.2679  23.2842
## Residuals 265.5398 157.3606  91.6686  74.9197  59.4149  50.7397  47.3443
## Total     355.1520 183.2308 149.0236  94.5782  80.3546  59.2781  58.3481
##                PC8      PC9     PC10     PC11     PC12     PC13     PC14
## Fitted
## Residuals  41.1301  39.2101  36.4674  32.3485  28.5313  28.0699  27.5505
## Total      49.3092  40.8127  38.5361  36.1489  35.4990  30.7850  30.2615
##               PC15     PC16     PC17     PC18     PC19     PC20     PC21
## Fitted
## Residuals  26.2549  24.0054  22.1261  21.8733  21.7556  20.9410  19.7891
## Total      29.0613  26.8140  25.8752  24.6339  23.2060  22.2234  21.7140
##               PC22     PC23     PC24     PC25     PC26     PC27     PC28
## Fitted
## Residuals  19.0491  18.1612  17.7255  17.1035  16.7104  16.6269  16.1307
## Total      21.2347  21.0382  19.8690  19.4048  18.6921  18.4397  17.2023
##               PC29     PC30     PC31     PC32     PC33     PC34     PC35
## Fitted
## Residuals  15.6475  15.3318  14.7780  13.9033  12.7712  12.3730  12.0795
## Total      16.8355  16.7241  16.3987  15.7905  15.6259  15.3121  14.8427
##               PC36     PC37     PC38     PC39     PC40     PC41     PC42
## Fitted
## Residuals
## Total      14.6820  14.0847  13.7266  13.0061  12.4638  11.7355  11.0454
```

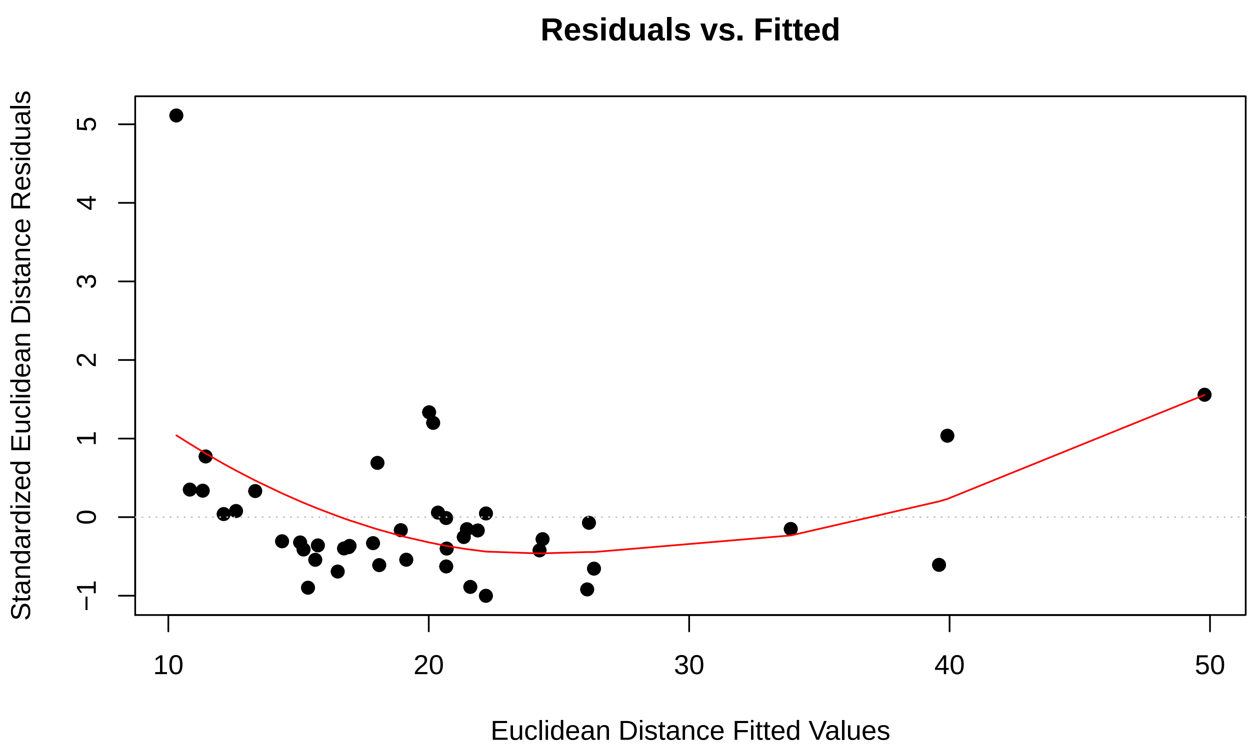
4. Examine RRPP plots to check for assumptions.

```r
## Old Leaves residuals vs fitted values (homoscedasticity
## check)
Odiagnostics <- plot(O_LMpos, type = "diagnostics")
```

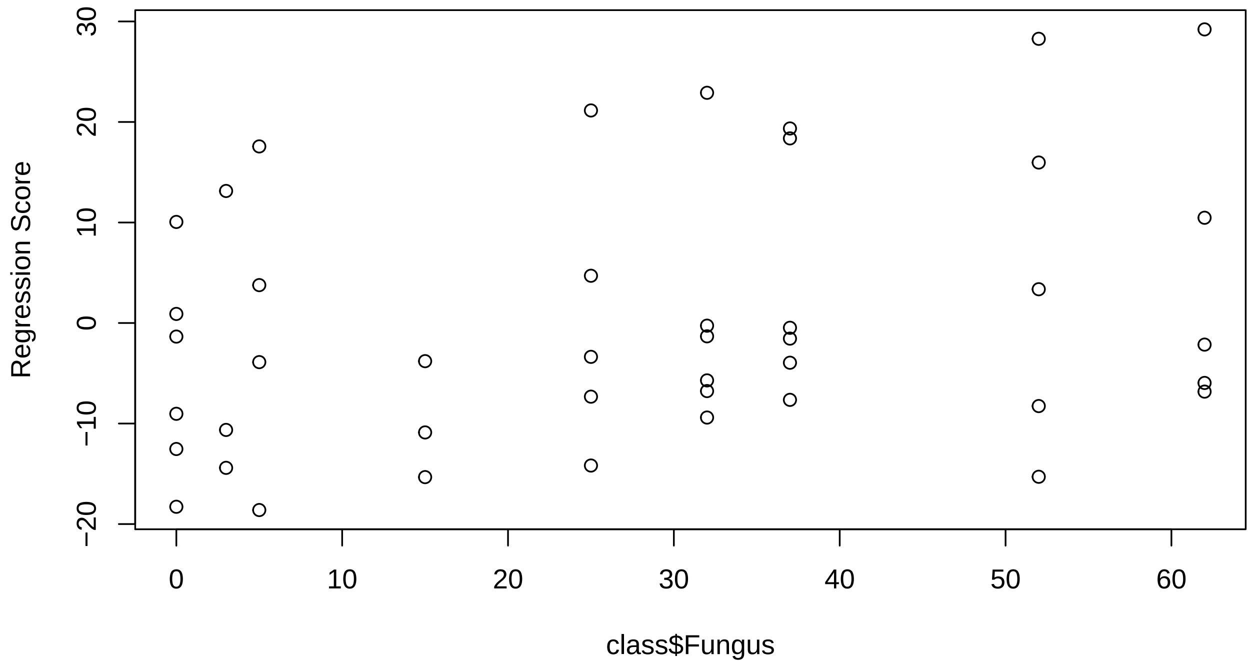

PCA Residuals
PC 2 11.27 %
PC 1 30.16 %
−50
0
50
100
−40
−20
0
20


Q−Q plot
Euclidean Distance Residuals
Theoretical Quantiles
−2
−1
0
1
2
20
40
60
80
100

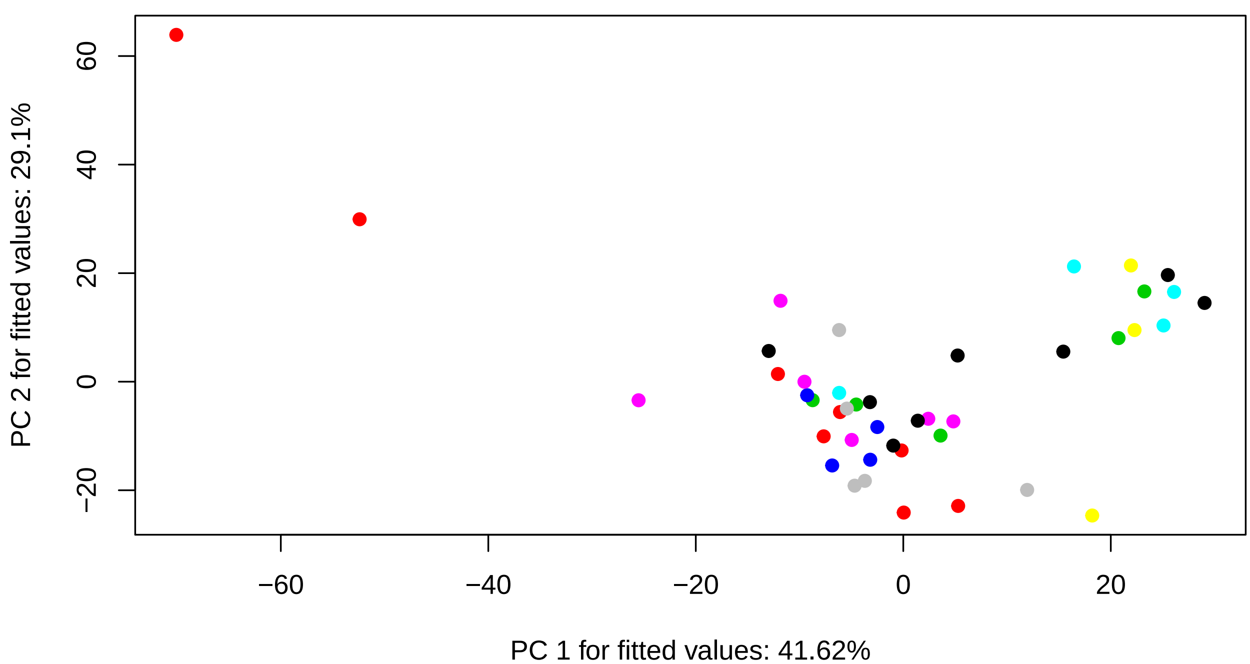




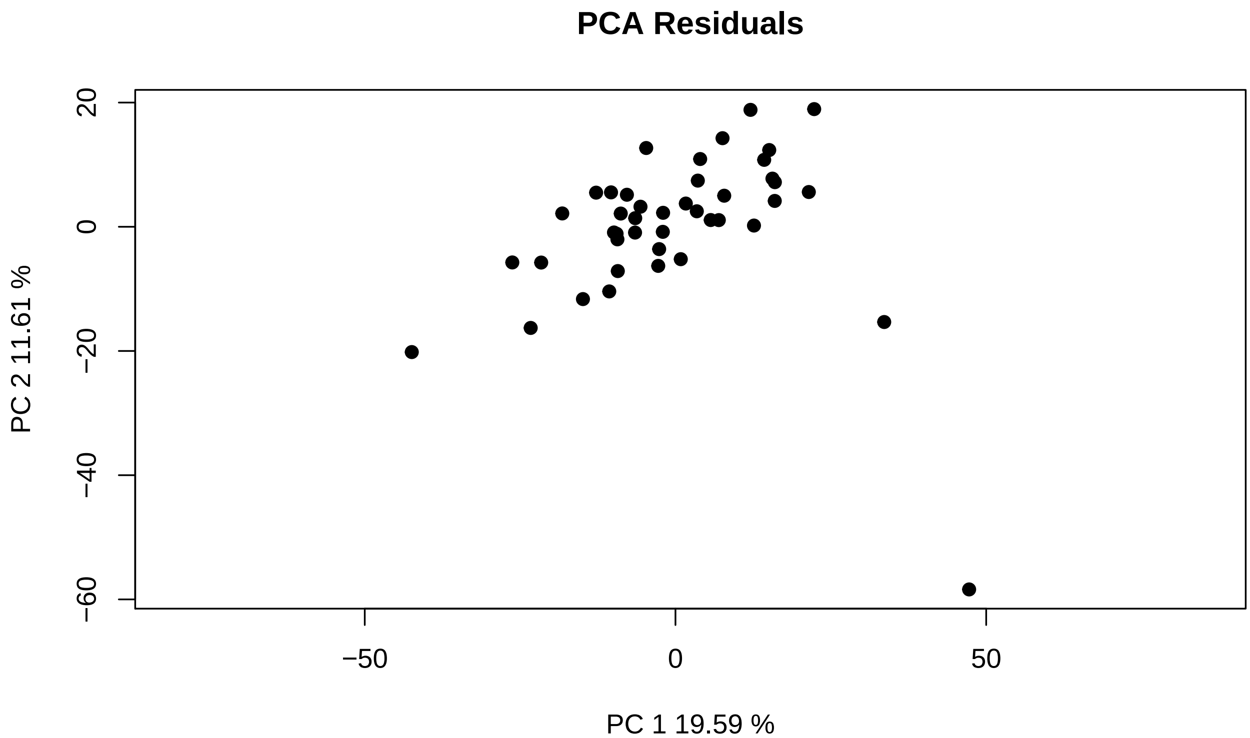

```
# linear regression plot
Oregression <- plot(O_LMpos, type = "regression", predictor = class$Fungus,
    reg.type = "RegScore")
```

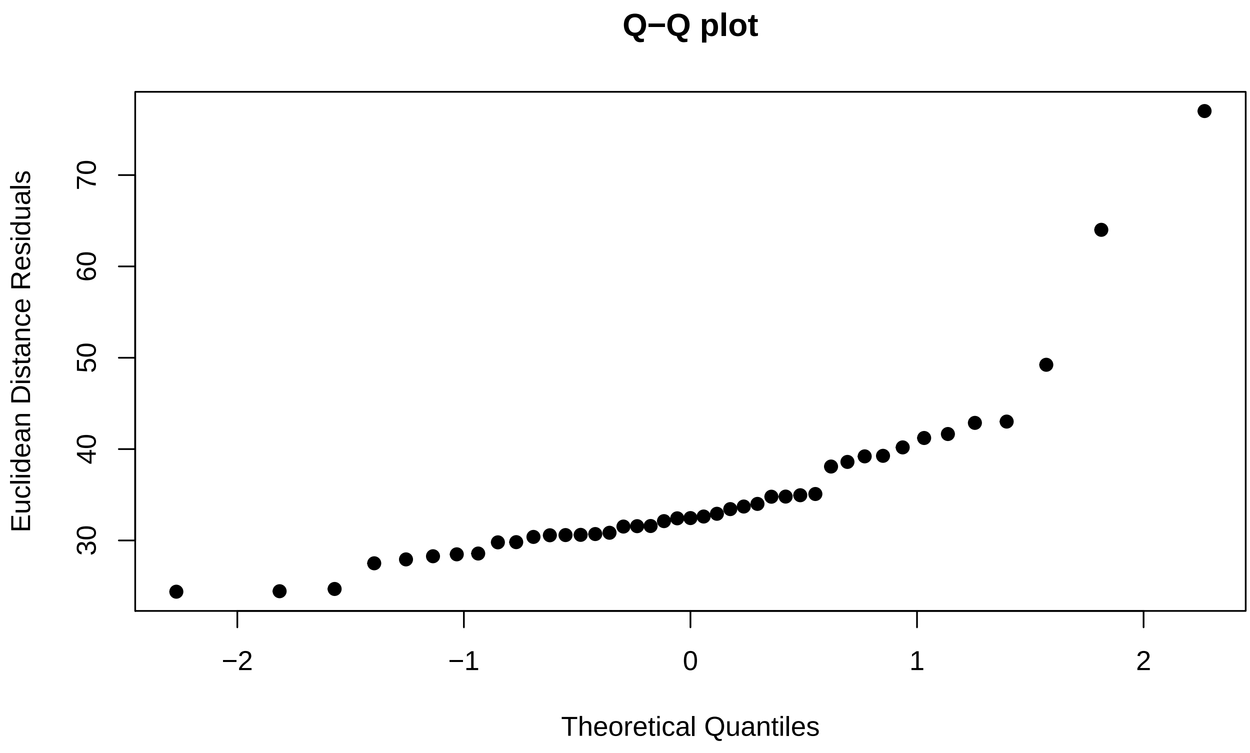


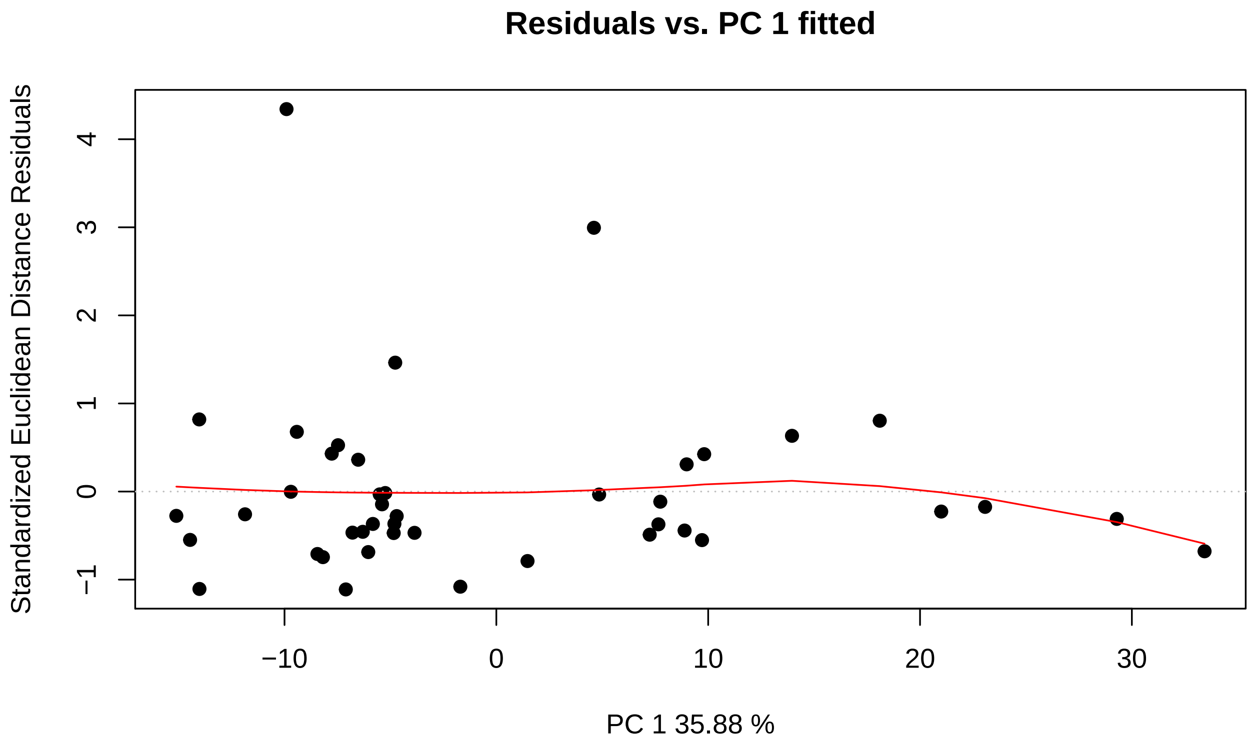

```r
# pca plot
Opcplot <- plot(O_LMpos, type = "PC", pch = 19, col = interaction(class$Water,
    class$Fungus))
```

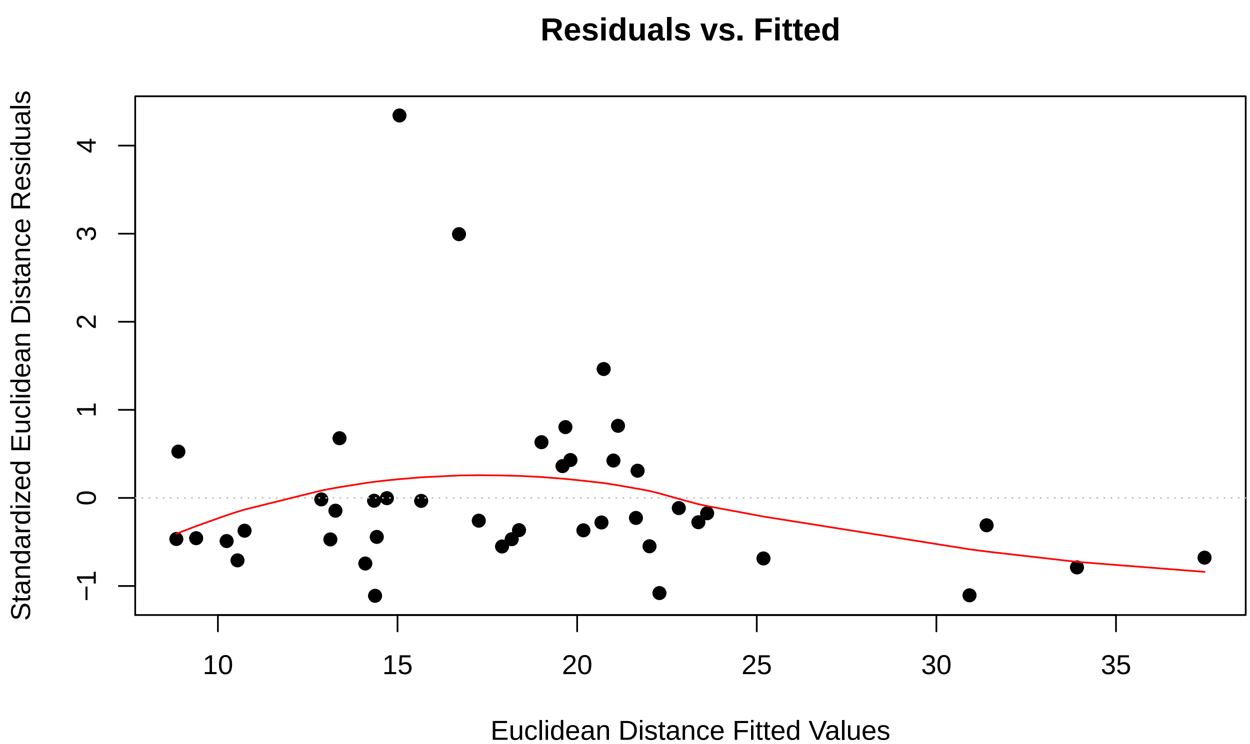



```r
## Young Leaves residuals vs fitted values (homoscedasticity
## check)
Ydiagnostics <- plot(Y_LMpos, type = "diagnostics")
```

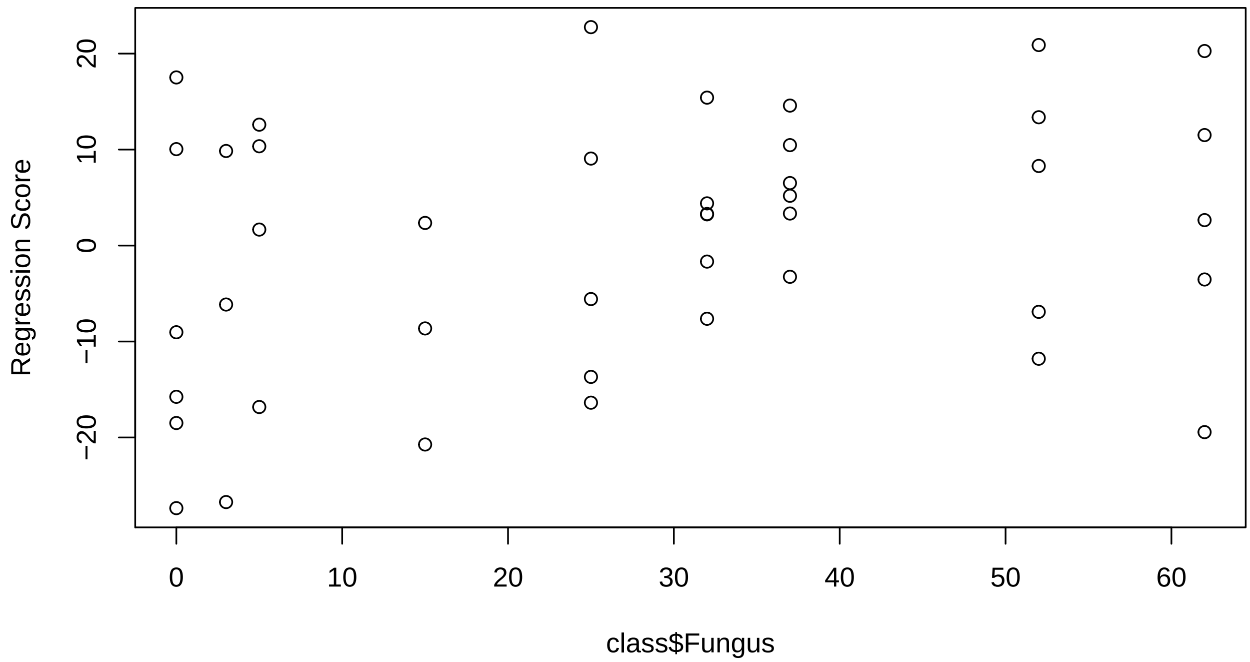

PCA Residuals
PC 2 11.61 %
PC 1 19.59 %
20
0
−20
−40
−60
−50
0
50


Q−Q plot
Euclidean Distance Residuals
Theoretical Quantiles
70
60
50
40
30
−2
−1
0
1
2

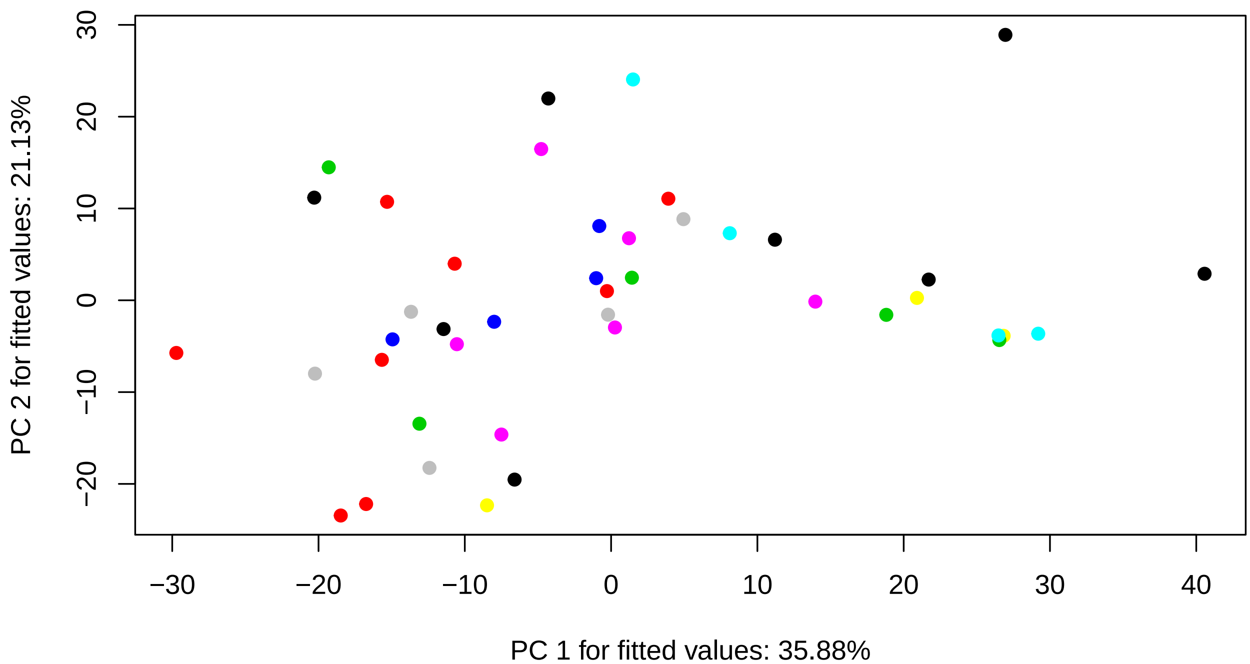





```r
# linear regression plot
Yregression <- plot(Y_LMpos, type = "regression", predictor = class$Fungus,
    reg.type = "RegScore")
```




```r
# pca plot
Ypcplot <- plot(Y_LMpos, type = "PC", pch = 19, col = interaction(class$Water,
    class$Fungus))
```



5. Perform an RRPP ANOVA and print results.

```r
## Old Leaves
OposANOVA <- anova(O_LMpos, effect.type = "F", error = c("Residuals",
    "Block:Water", "Block:Water:Fungus", "Residuals", "Block:Water:Fungus",
    "Block:Water:Fungus", "Residuals"))
summary(OposANOVA, formula = T)
```

```
##
## Analysis of Variance, using Residual Randomization
## Permutation procedure: Randomization of null model residuals
## Number of permutations: 1000
## Estimation method: Ordinary Least Squares
## Sums of Squares and Cross-products: Type III
## Effect sizes (Z) based on F distributions
##
##                    Df    SS      MS     Rsq      F        Z Pr(>F)
## Block               1  1070 1069.78 0.01397 0.6686 -0.57292  0.689
## Water               1  1330 1329.63 0.01737 1.3944  1.24172  0.107
## Fungus              1  1769 1768.66 0.02310 1.0018  0.06379  0.465
## Block:Water         1   954  953.56 0.01245 0.5959 -1.02809  0.863
## Block:Fungus        1  1305 1304.63 0.01704 0.7389 -0.65351  0.758
## Water:Fungus        1  1483 1483.16 0.01937 0.8401 -0.68090  0.768
## Block:Water:Fungus  1  1766 1765.55 0.02306 1.1034  0.41487  0.308
## Residuals          35 56003 1600.08 0.73143
## Total              42 76566
##
## Call: lm.rrpp(f1 = scaled_Y_old ~ Block * Water * Fungus, SS.type = "III",
##     data = class, print.progress = F)
```

```
## Young Leaves
YposANOVA <- anova(Y_LMpos, effect.type = "F", error = c("Residuals",
    "Block:Water", "Block:Water:Fungus", "Residuals", "Block:Water:Fungus",
    "Block:Water:Fungus", "Residuals"))
summary(YposANOVA, formula = T)
```

```
##
## Analysis of Variance, using Residual Randomization
## Permutation procedure: Randomization of null model residuals
## Number of permutations: 1000
## Estimation method: Ordinary Least Squares
## Sums of Squares and Cross-products: Type III
## Effect sizes (Z) based on F distributions
##
##                    Df    SS     MS     Rsq      F        Z Pr(>F)
## Block               1  2447 2446.9 0.03323 1.5044  1.45850  0.089 .
## Water               1  1644 1643.8 0.02233 1.1327  0.63780  0.279
## Fungus              1  2423 2423.0 0.03291 1.6588  1.67310  0.044 *
## Block:Water         1  1451 1451.3 0.01971 0.8923 -0.25299  0.568
## Block:Fungus        1  2248 2248.4 0.03054 1.5393  1.58953  0.047 *
## Water:Fungus        1  1225 1224.7 0.01663 0.8384 -1.00498  0.843
## Block:Water:Fungus  1  1461 1460.7 0.01984 0.8981 -0.23220  0.556
## Residuals          35 56928 1626.5 0.77321
## Total              42 73626
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Call: lm.rrpp(f1 = scaled_Y_young ~ Block * Water * Fungus, SS.type = "III",
##     data = class, print.progress = F)
```

6. Test lm.rrpp model coefficients. “d” is the amount of change in a variable for the coefficient indicated.

```
## Old Leaves test model coefficients
Oposcoef <- coef(O_LMpos, test = T)
```

```r
summary(Oposcoef)
```

```
##
## Linear Model fit with lm.rrpp
##
## Number of observations: 43
## Number of dependent variables: 1823
## Data space dimensions: 42
## Sums of Squares and Cross-products: Type III
## Number of permutations: 1000
##
## Statistics (distances) of coefficients with 95 percent confidence intervals,
## effect sizes, and probabilities of exceeding observed values based on
## 1000 random permutations using RRPP
##
##                           d.obs  UCL (95%)          Zd Pr(>d)
## (Intercept)          34.1599256 61.7702203 -1.93180616  0.993
## Block                15.5968289 24.7476777 -0.30721242  0.535
## WaterLow             45.0149621 60.4726540  0.05398047  0.420
## Fungus                1.1044955  1.5009254  0.59727782  0.212
## Block:WaterLow       19.6425423 32.5116629 -0.65799528  0.738
## Block:Fungus          0.4864742  0.7972160  0.02949339  0.363
## WaterLow:Fungus       1.4031743  1.8939047  0.27778080  0.324
## Block:WaterLow:Fungus  0.7572552  0.9467192  0.72492867  0.205
```

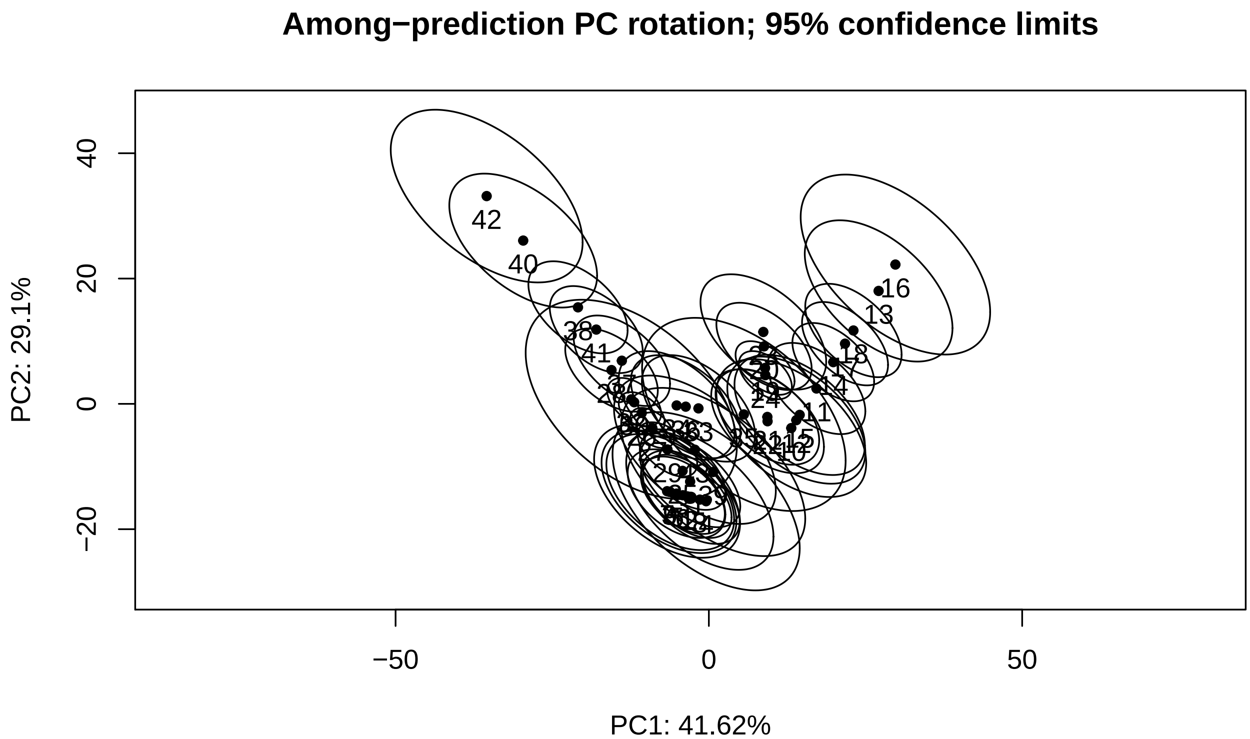
```r
## Young Leaves test model coefficients
Yposcoef <- coef(Y_LMpos, test = T)
summary(Yposcoef)
```

```
##
## Linear Model fit with lm.rrpp
##
## Number of observations: 43
## Number of dependent variables: 1753
## Data space dimensions: 42
## Sums of Squares and Cross-products: Type III
## Number of permutations: 1000
##
## Statistics (distances) of coefficients with 95 percent confidence intervals,
## effect sizes, and probabilities of exceeding observed values based on
## 1000 random permutations using RRPP
##
##                           d.obs  UCL (95%)         Zd Pr(>d)
## (Intercept)          43.3123692 57.3355802 -1.2919657  0.924
## Block                23.5881635 23.1463030  2.0389293  0.043
## WaterLow             50.0515288 58.4802917  0.6276905  0.229
## Fungus                1.2927626  1.2743742  1.9727754  0.043
## Block:WaterLow       24.2326549 30.0046238  0.2116341  0.354
## Block:Fungus          0.6386411  0.6701481  1.6266467  0.074
## WaterLow:Fungus       1.2750398  1.7195192 -0.3543871  0.578
## Block:WaterLow:Fungus  0.6887842  0.8529508  0.2071387  0.360
```

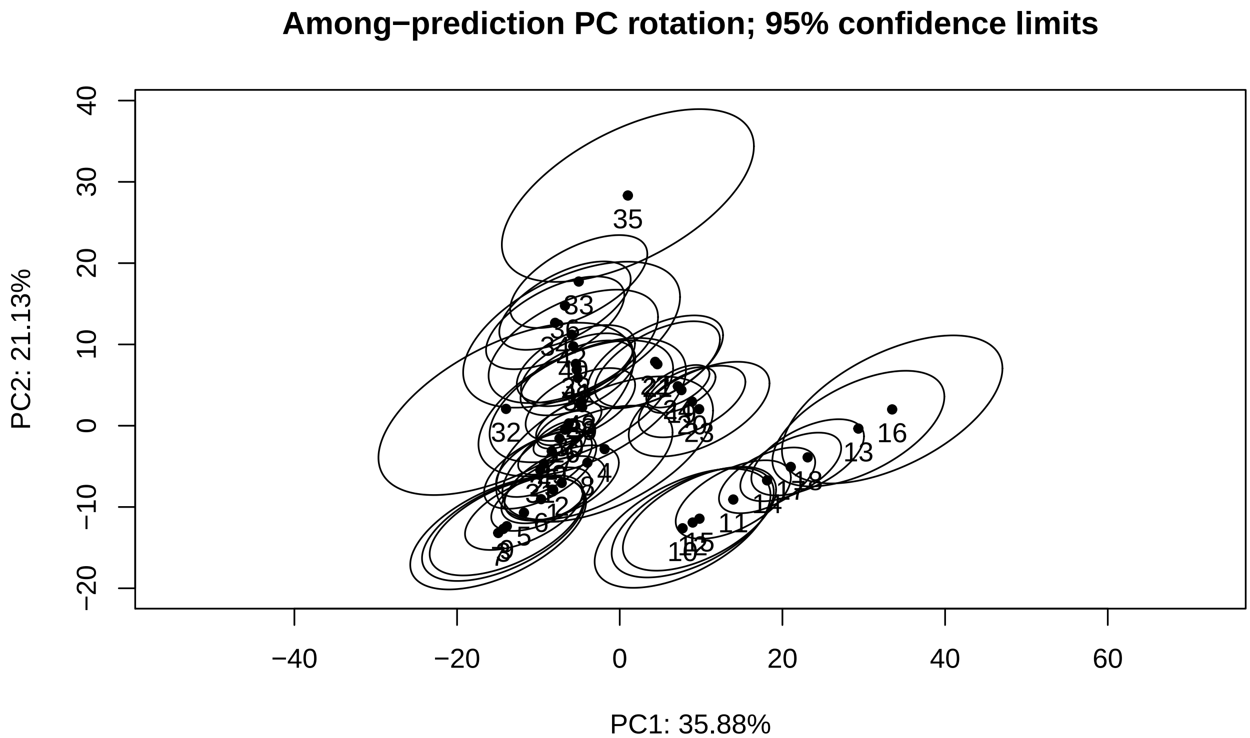
Block and Fungus have the largest effect on the model for young leaves, but not their interaction. The standard is the mean for High water treatment.

7. Compute predicted values from the lm.rrpp model fit using bootstrapped residuals to generate confidence intervals (precision of group mean estimates).

```
O_pred <- predict(O_LMpos)
plot(O_pred, PC = T, ellipse = T)
```




```
Y_pred <- predict(Y_LMpos)
plot(Y_pred, PC = T, ellipse = T)
```



8. Test pairwise differences between least squares means. Similar to tukeyHSD function in the r stats package. The pairwise function will generate tables with confidence intervals and p-values for the pairwise statistic, Euclidean distance between least-squares means.

```
## Old Leaves pairwise differences of fungus
Opospw <- pairwise(O_LMpos, groups = class$Fungus)
summary(Opospw, confidence = 0.95, stat.table = T)
```

```
##
## Pairwise comparisons
##
## Groups: 0 3 5 15 25 32 37 52 62
##
## RRPP: 1000 permutations
##
## LS means:
## Vectors hidden (use show.vectors = TRUE to view)
##
## Pairwise distances between means, plus statistics
##                d UCL (95%)          Z Pr > d
## 0:3    9.326918 15.347732 -1.6828998  0.987
## 0:5    6.413682 13.026220 -1.8395290  0.996
## 0:15   8.537287 15.908701 -1.8479211  0.999
## 0:25   9.682422 16.732834 -1.0034673  0.834
## 0:32  11.812877 20.237608 -1.0711518  0.867
## 0:37  13.658639 23.399735 -1.0711518  0.867
## 0:52  20.109497 33.261520 -0.8589151  0.790
## 0:62  22.508012 38.725115 -1.1569500  0.896
## 3:5   10.741033 18.549288 -0.9767991  0.897
## 3:15   7.242409 12.496847 -1.5689213  0.976
## 3:25  13.176750 20.514654 -1.1752098  0.892
```

```
## 3:32  14.364873 22.533596 -1.1724271  0.888
## 3:37  15.818024 25.043488 -1.1266412  0.878
## 3:52  20.555725 33.500601 -0.9047426  0.804
## 3:62  23.677676 39.000291 -1.1144980  0.871
## 5:15   6.053913  9.898580 -1.0482176  0.943
## 5:25   9.152150 13.920104 -0.9638321  0.851
## 5:32  11.806098 17.688517 -0.9905101  0.850
## 5:37  13.414391 20.379021 -0.9533571  0.849
## 5:52  19.464398 29.006170 -0.6208238  0.707
## 5:62  21.953727 34.730775 -1.0010884  0.850
## 15:25  8.889463 12.927338 -1.3055774  0.924
## 15:32 10.767077 15.491585 -1.2327690  0.917
## 15:37 12.160027 17.784690 -1.1345066  0.888
## 15:52 17.098166 25.346009 -0.7002696  0.740
## 15:62 20.263196 31.420478 -1.0658210  0.856
## 25:32  3.156363  4.745157 -1.2999704  0.914
## 25:37  4.699102  7.597054 -1.2444151  0.910
## 25:52 11.316338 17.277691 -0.6817045  0.724
## 25:62 13.769592 23.437792 -1.2468365  0.917
## 32:37  1.845762  3.162126 -1.0711518  0.867
## 32:52  8.828454 13.898701 -0.6016455  0.673
## 32:62 10.940495 18.847249 -1.2377334  0.912
## 37:52  7.221105 11.133099 -0.5383293  0.644
## 37:62  9.189634 15.918771 -1.2646036  0.922
## 52:62  5.837055 11.466108 -1.5197845  0.997
```

```r
## Young Leaves pairwise differences of fungus
Ypospw <- pairwise(Y_LMpos, groups = class$Fungus)
summary(Ypospw, confidence = 0.95, stat.table = T)
```

```
##
## Pairwise comparisons
##
## Groups: 0 3 5 15 25 32 37 52 62
##
## RRPP: 1000 permutations
##
## LS means:
## Vectors hidden (use show.vectors = TRUE to view)
##
## Pairwise distances between means, plus statistics
##               d UCL (95%)          Z Pr > d
## 0:3    9.854093 14.743698 -2.0747297  0.997
## 0:5    9.353807 13.331403 -0.7641721  0.779
## 0:15  10.700420 16.115496 -1.4689387  0.966
## 0:25   7.562621 12.422986 -2.4228412  1.000
## 0:32   9.535169 15.674433 -2.2859229  1.000
## 0:37  11.025039 18.123563 -2.2859229  1.000
## 0:52  16.992197 27.307179 -1.8838351  0.987
## 0:62  18.946444 31.241987 -2.2571727  1.000
## 3:5   10.611811 16.135336 -1.9080841  0.998
## 3:15   8.159435 12.371299 -1.9526129  0.997
## 3:25  12.031577 17.637315 -2.3172738  0.998
## 3:32  13.017518 19.508655 -2.1354904  0.999
## 3:37  14.042320 21.285137 -2.1455179  0.999
```

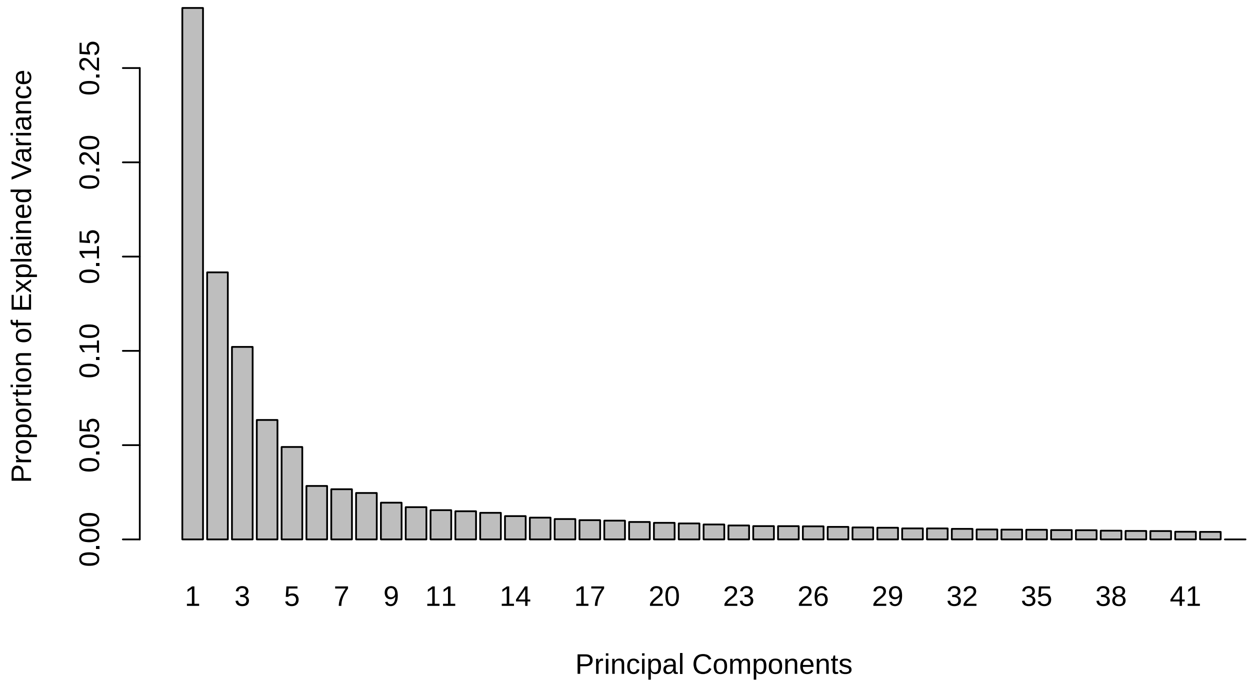
```
## 3:52  18.632569 28.544843 -1.8361515  0.990
## 3:62  20.424364 31.794098 -2.1968168  1.000
## 5:15   5.706131  9.750433 -1.8043770  0.998
## 5:25   9.764308 14.012366 -1.1767303  0.913
## 5:32  11.972073 17.810433 -1.2391327  0.931
## 5:37  12.980020 19.621502 -1.3747098  0.960
## 5:52  18.049742 27.496967 -1.1798161  0.893
## 5:62  19.303690 30.143465 -1.7895809  0.992
## 15:25  9.174418 12.726981 -1.4971598  0.963
## 15:32 10.645667 15.139172 -1.3982002  0.952
## 15:37 11.383035 16.590292 -1.4888181  0.967
## 15:52 15.349692 22.908809 -1.1912952  0.899
## 15:62 17.100786 26.349416 -1.8452052  0.994
## 25:32  2.894715  4.506400 -1.4851014  0.952
## 25:37  4.144646  6.646085 -1.7568113  0.987
## 25:52 10.383804 15.773489 -1.3350691  0.913
## 25:62 12.119246 19.533196 -2.0797469  1.000
## 32:37  1.489870  2.449130 -2.2859229  1.000
## 32:52  8.312294 12.463310 -1.4371659  0.931
## 32:62  9.612051 15.564903 -2.2298692  1.000
## 37:52  7.155507 10.519312 -1.2747762  0.894
## 37:62  8.195418 13.160463 -2.2191516  1.000
## 52:62  6.749666 10.159396 -1.9051218  0.999
```

PCA

8. Identify the major source of variation in data and determine if the variation is sourced from experimental bias or biological conditions.

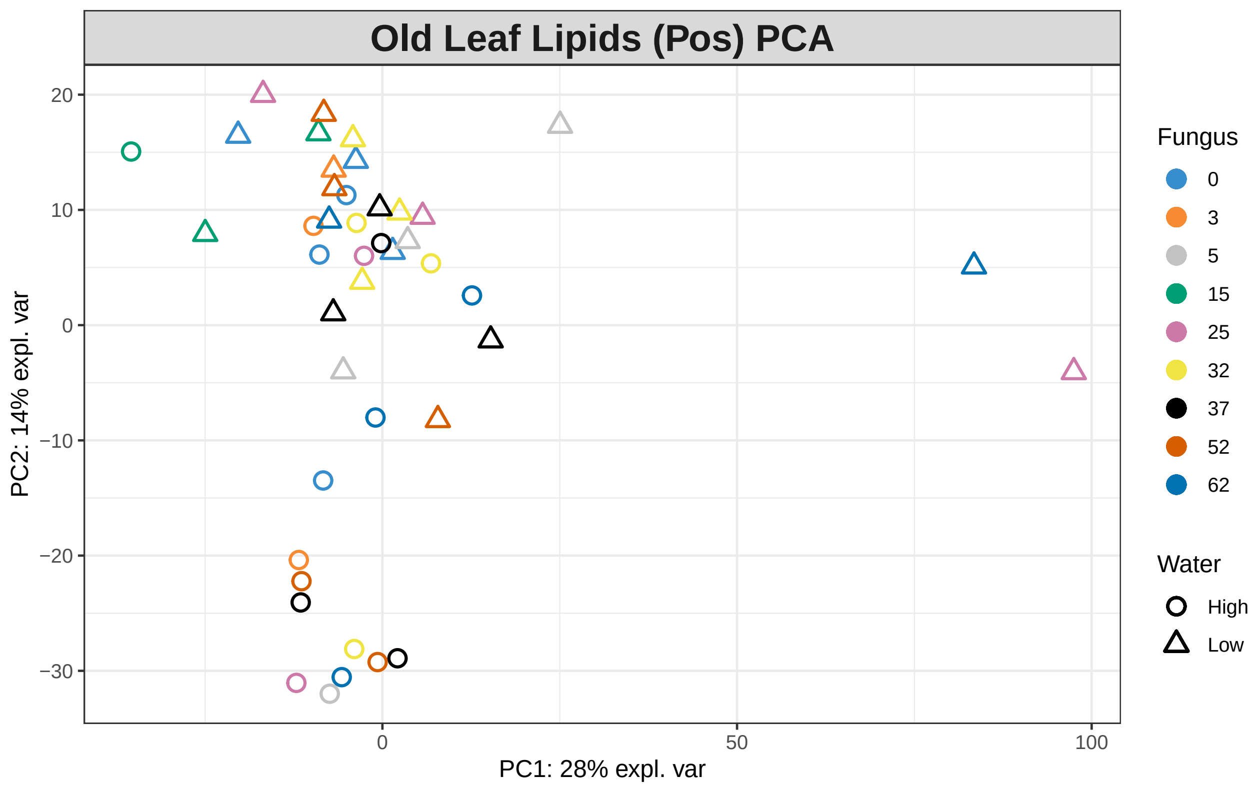
```r
# Old Leaf Lipids (Pos) tune how many components to use
tune.pca(scaled_Y_old)
```

```
## Eigenvalues for the first 10 principal components, see object$sdev^2:
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7       PC8
## 513.86089 258.21864 186.14420 115.49587  89.38257  51.68852  48.53052  44.90014
##       PC9      PC10
##  35.51871  31.17389
##
## Proportion of explained variance for the first 10 principal components, see object$explained_varianc
##        PC1        PC2        PC3        PC4        PC5        PC6        PC7
## 0.28187652 0.14164489 0.10210872 0.06335484 0.04903048 0.02835355 0.02662124
##        PC8        PC9       PC10
## 0.02462981 0.01948366 0.01710032
##
## Cumulative proportion explained variance for the first 10 principal components, see object$cum.var:
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7       PC8
## 0.2818765 0.4235214 0.5256301 0.5889850 0.6380155 0.6663690 0.6929902 0.7176200
##       PC9      PC10
## 0.7371037 0.7542040
##
##  Other available components:
##  --------------------
##  loading vectors: see object$rotation
```

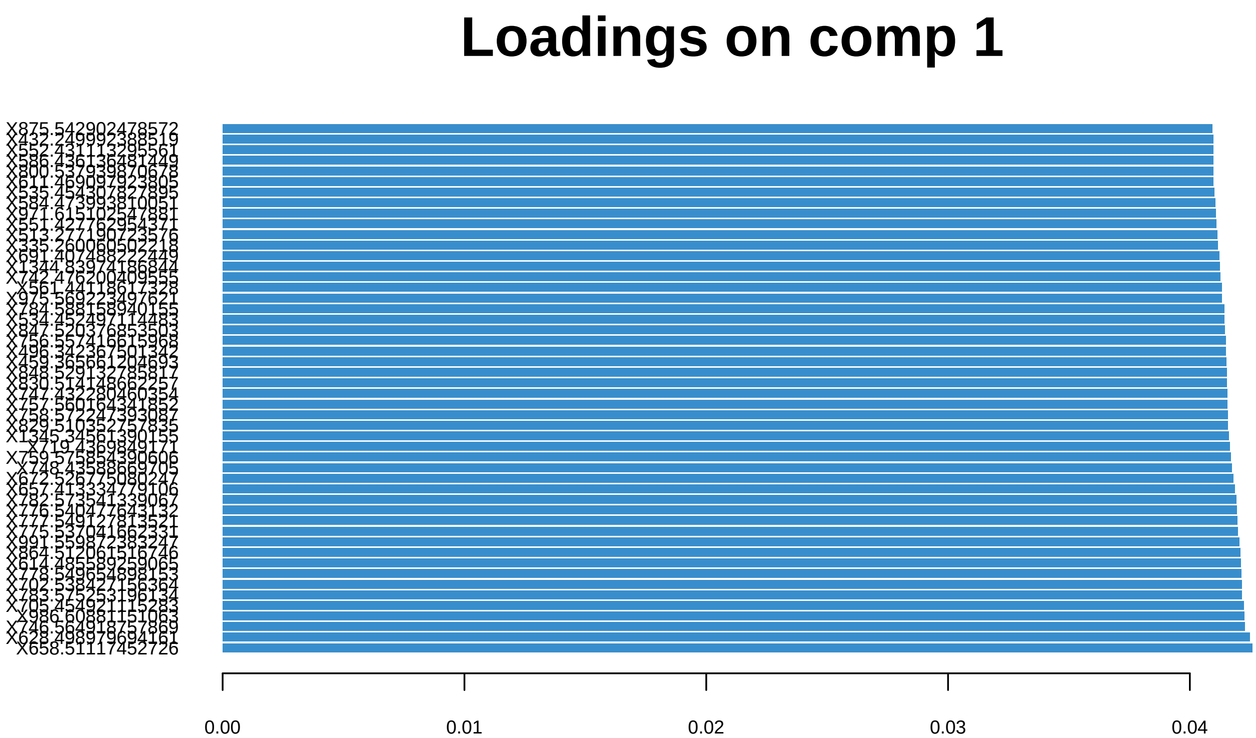



```r
pca.res <- mixOmics::pca(scaled_Y_old, ncomp = 3, scale = F)

# plot pca
plotIndiv(pca.res, group = class$Fungus, ind.names = F, pch = as.factor(class$Water),
    legend = T, legend.title = "Fungus", legend.title.pch = "Water",
    title = "Old Leaf Lipids (Pos) PCA")
```

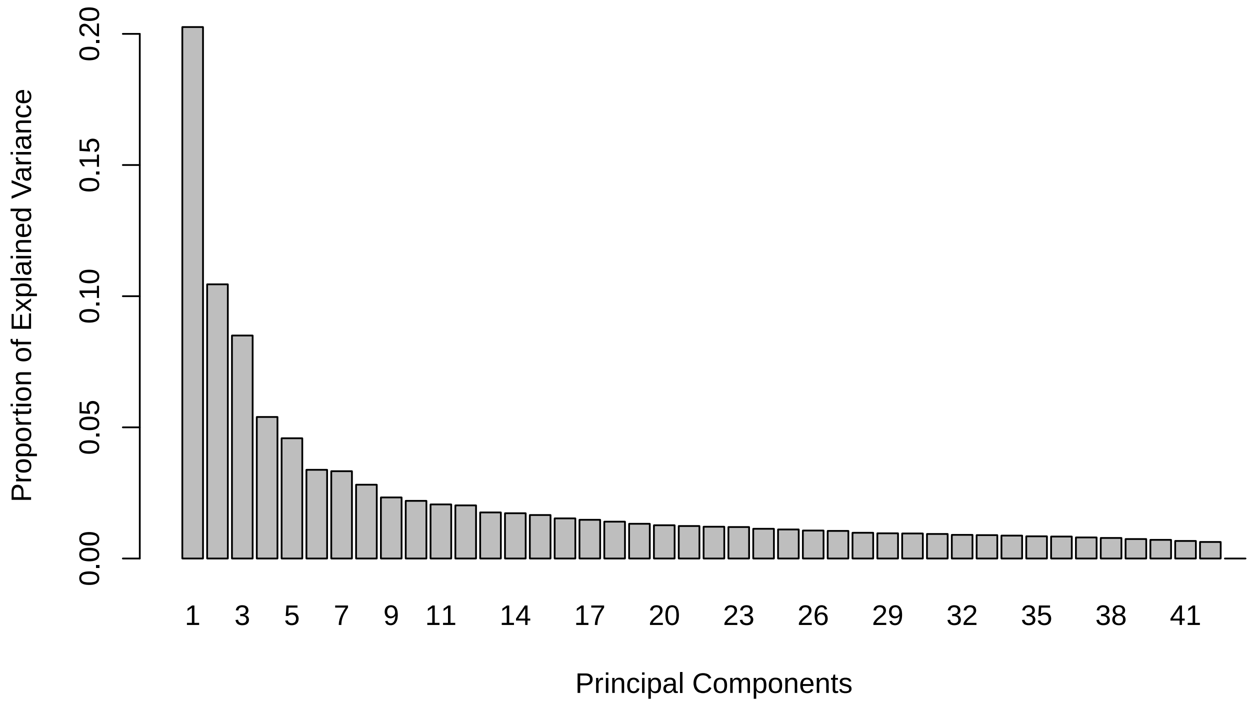


```r
# Look at variable coefficients in each component with the
# loading vectors The absolute value of loading vectors
# represent the importance of each variable to define each PC
plotLoadings(pca.res, ndisplay = 50)
```



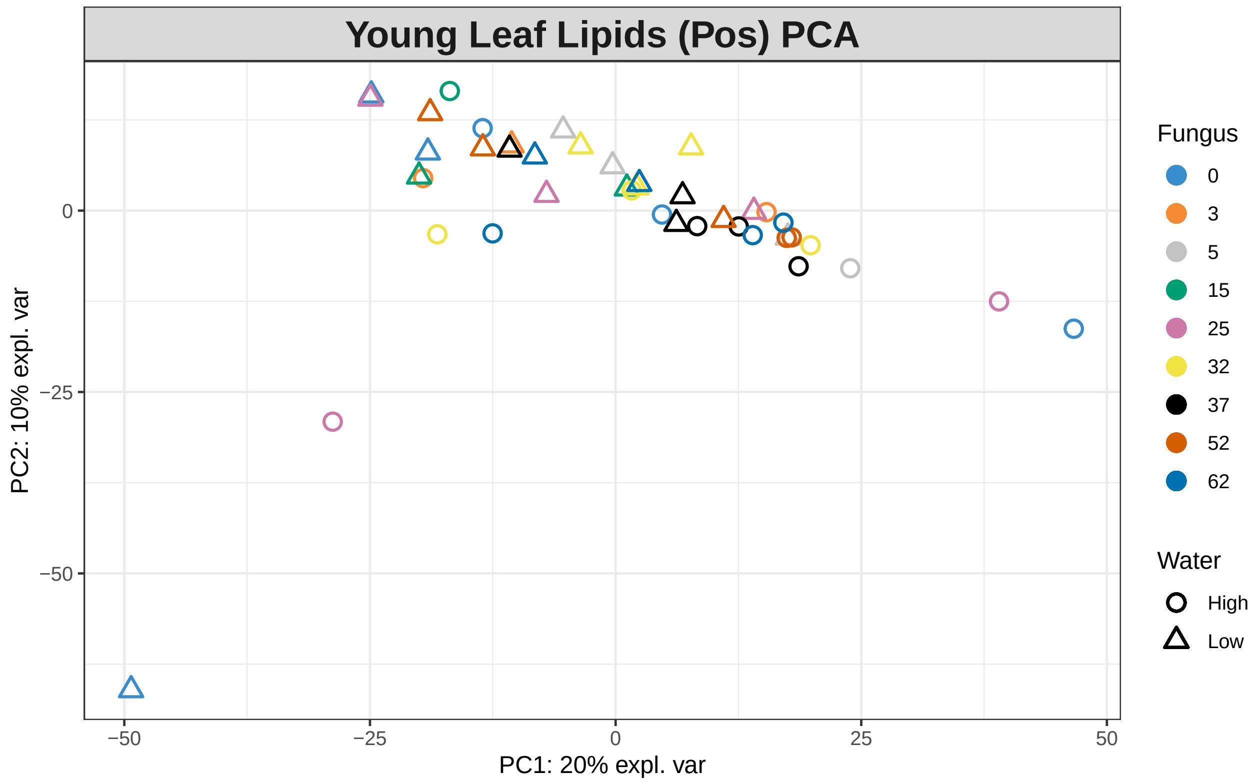

```r
# Young Leaf Lipids (Pos) tune how many components to use
tune.pca(scaled_Y_young)
```

```
## Eigenvalues for the first 10 principal components, see object$sdev^2:
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7       PC8
## 355.15199 183.23080 149.02358  94.57819  80.35455  59.27809  58.34810  49.30924
##       PC9      PC10
##  40.81271  38.53615
##
## Proportion of explained variance for the first 10 principal components, see object$explained_varianc
##        PC1        PC2        PC3        PC4        PC5        PC6        PC7
## 0.20259669 0.10452413 0.08501060 0.05395219 0.04583831 0.03381522 0.03328471
##        PC8        PC9       PC10
## 0.02812849 0.02328164 0.02198297
##
## Cumulative proportion explained variance for the first 10 principal components, see object$cum.var:
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7       PC8
## 0.2025967 0.3071208 0.3921314 0.4460836 0.4919219 0.5257371 0.5590219 0.5871503
##       PC9      PC10
## 0.6104320 0.6324149
##
##  Other available components:
##  --------------------
##  loading vectors: see object$rotation
```

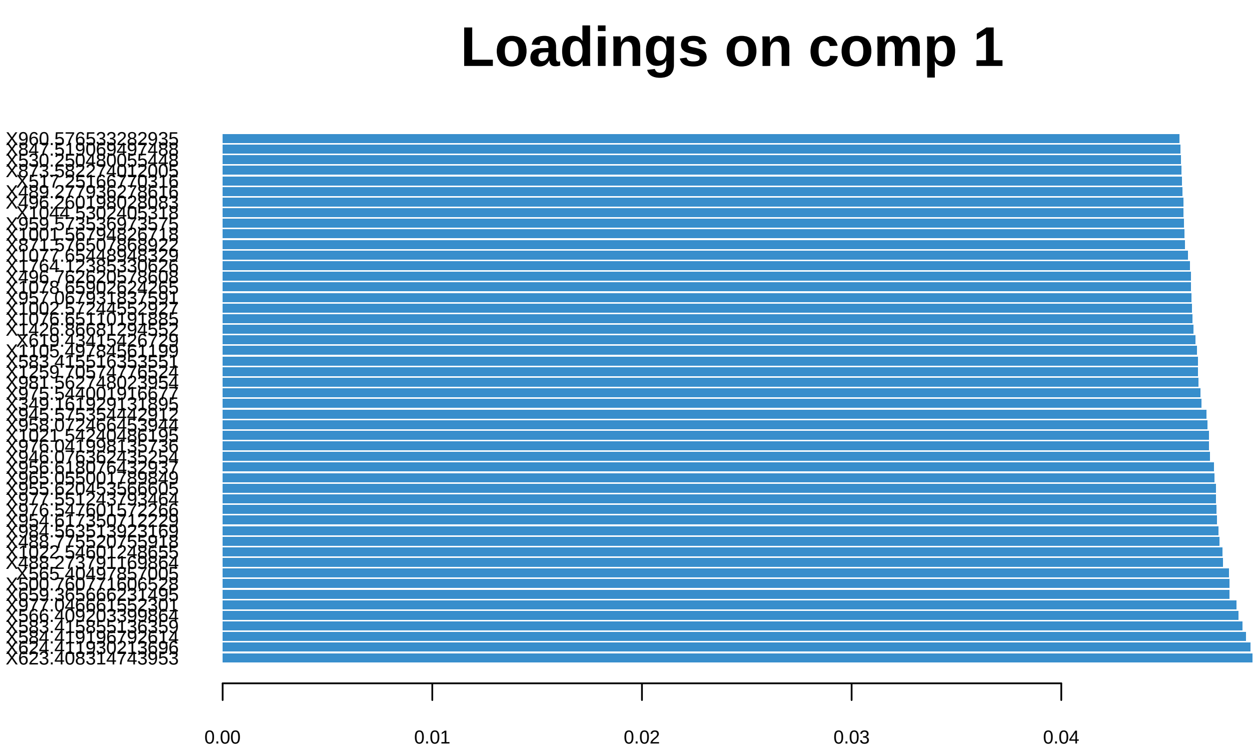



```
pca.res <- mixOmics::pca(scaled_Y_young, ncomp = 3, scale = F)

# plot pca
plotIndiv(pca.res, group = class$Fungus, ind.names = F, pch = as.factor(class$Water),
    legend = T, legend.title = "Fungus", legend.title.pch = "Water",
    title = "Young Leaf Lipids (Pos) PCA")
```



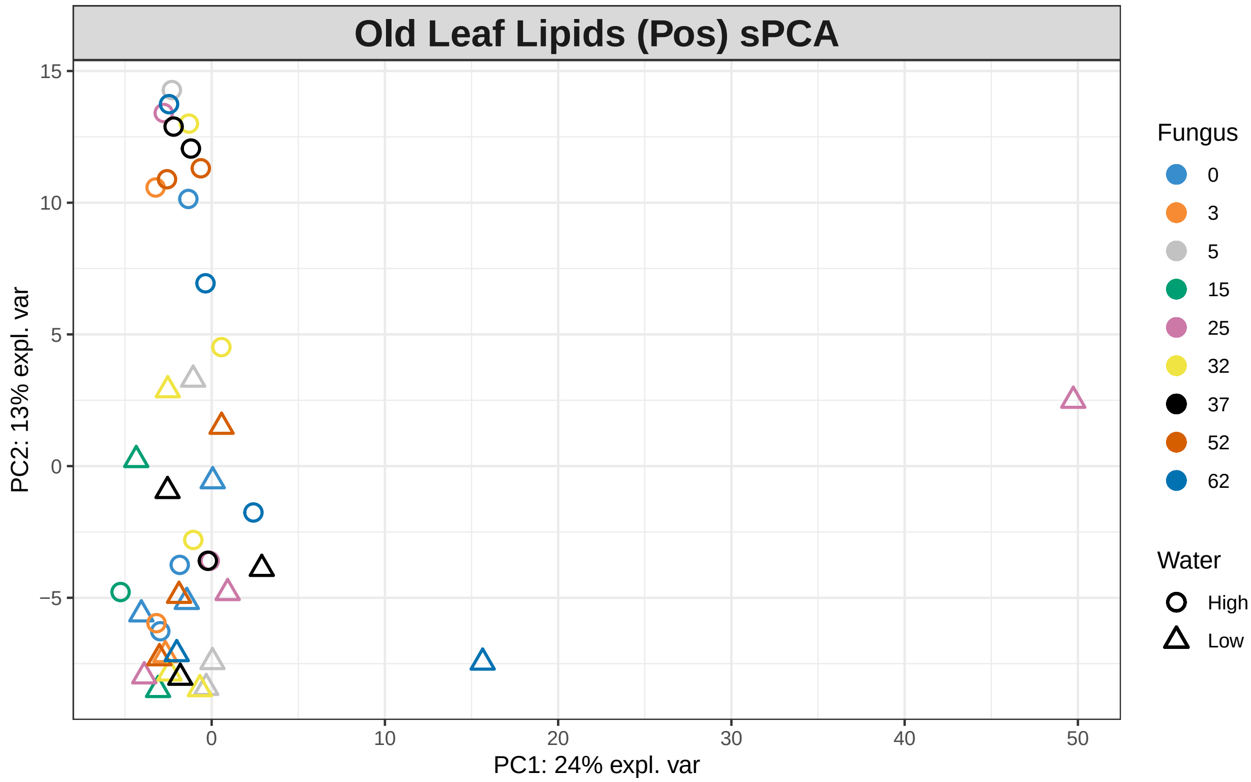

```r
# Look at variable coefficients in each component with the
# loading vectors The absolute value of loading vectors
# represent the importance of each variable to define each PC
plotLoadings(pca.res, ndisplay = 50)
```



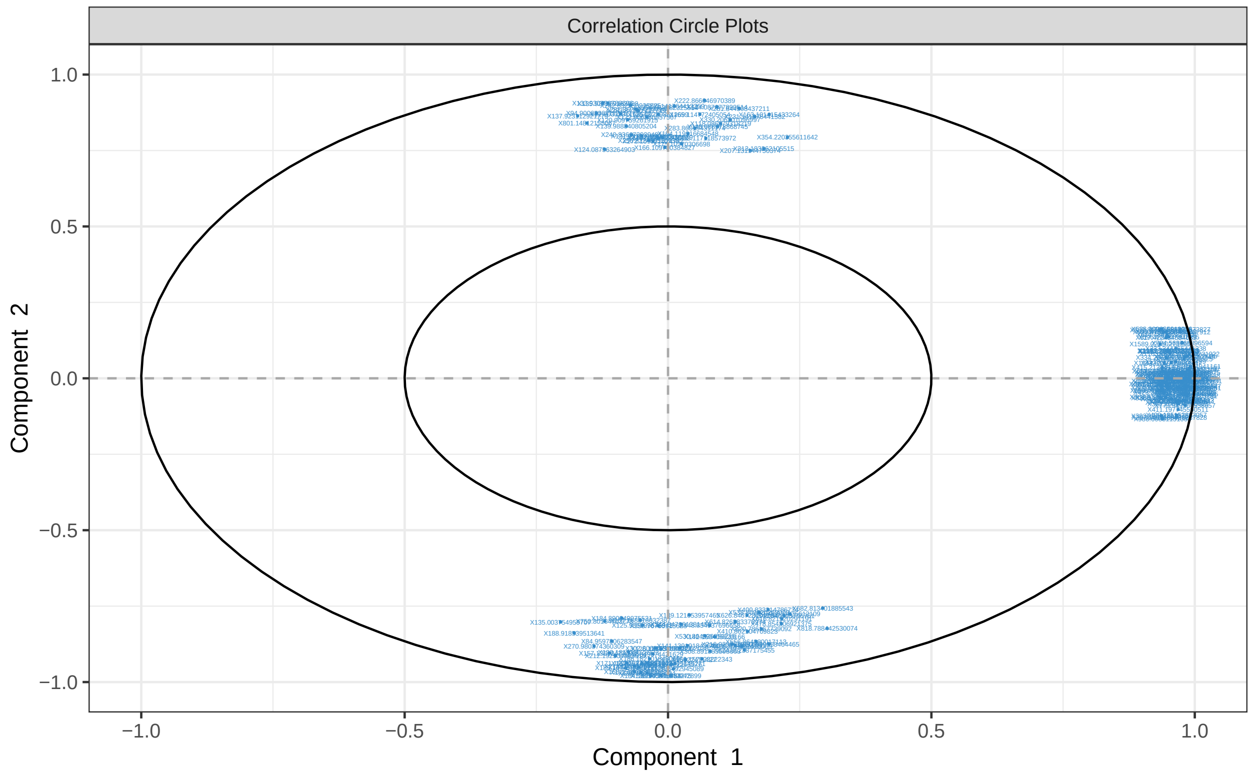
9. Identify the variables that contribute to the explained variance.

```
# Old Leaf
spca.res <- mixOmics::spca(scaled_Y_old, ncomp = 3, keepX = c(100,
    100, 10))

# plot spca
plotIndiv(spca.res, group = class$Fungus, ind.names = F, pch = as.factor(class$Water),
    legend = T, legend.title = "Fungus", legend.title.pch = "Water",
    title = "Old Leaf Lipids (Pos) sPCA")
```




```
# variables contributing to each component
plotVar(spca.res, cex = 1)
```



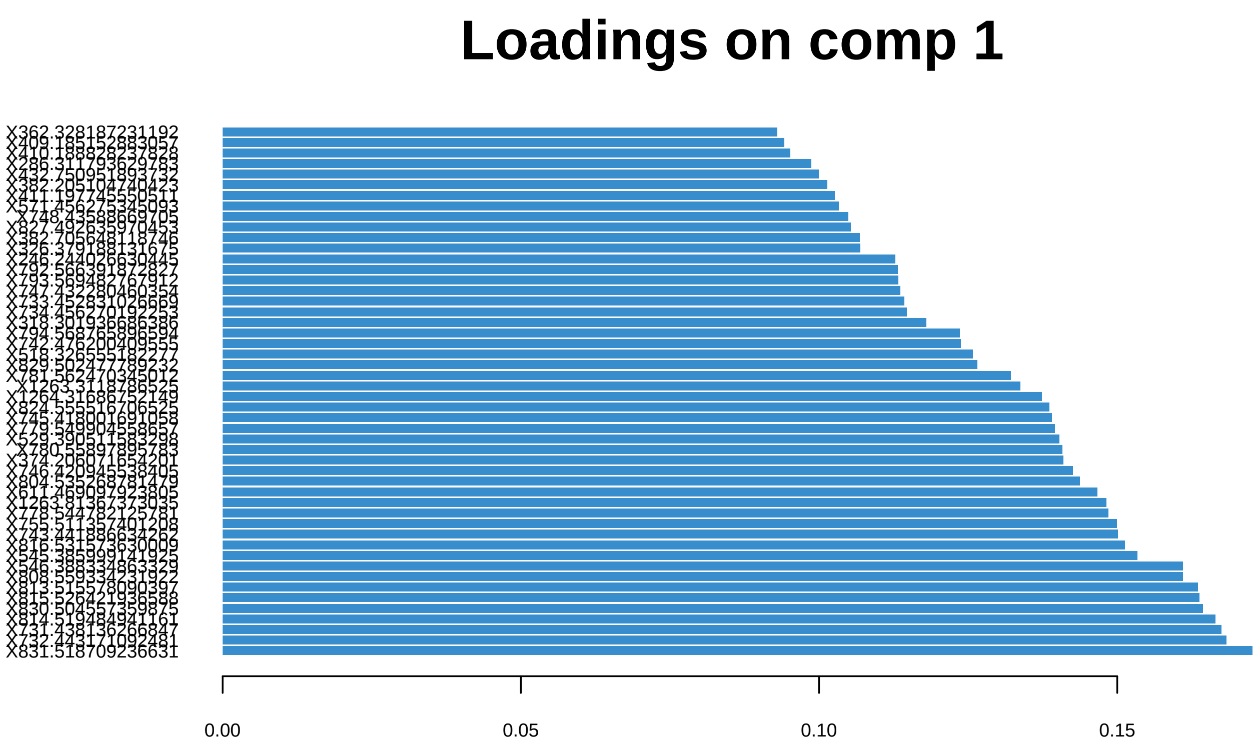

```r
selectVar(spca.res, comp = 1)$value  # view loading value of each metabolite
```

```
##                      value.var
## X831.518709236631 0.172620458
## X732.443171092481 0.168296057
## X731.438136266847 0.167454065
## X814.519484941161 0.166438746
## X830.504557359875 0.164320709
## X815.526421936588 0.163737218
## X813.515578090397 0.163508986
## X808.559334231922 0.161009134
## X546.388334863329 0.160984742
## X545.385999141925 0.153342022
## X816.531573630009 0.151288630
## X743.441886634262 0.150052144
## X755.511357401208 0.149947430
## X778.544782125781 0.148521054
## X1263.81367373035 0.148160308
## X611.469097923805 0.146656030
## X804.535268781479 0.143688197
## X746.420945538405 0.142549845
## X374.206071654201 0.140943889
## X780.55897895783  0.140750076
## X529.390511583298 0.140242603
## X779.549904558657 0.139479912
## X745.418001691058 0.138998524
## X824.555516706525 0.138571412
## X1264.31686752149 0.137319501
## X1263.3118786525  0.133758175
```

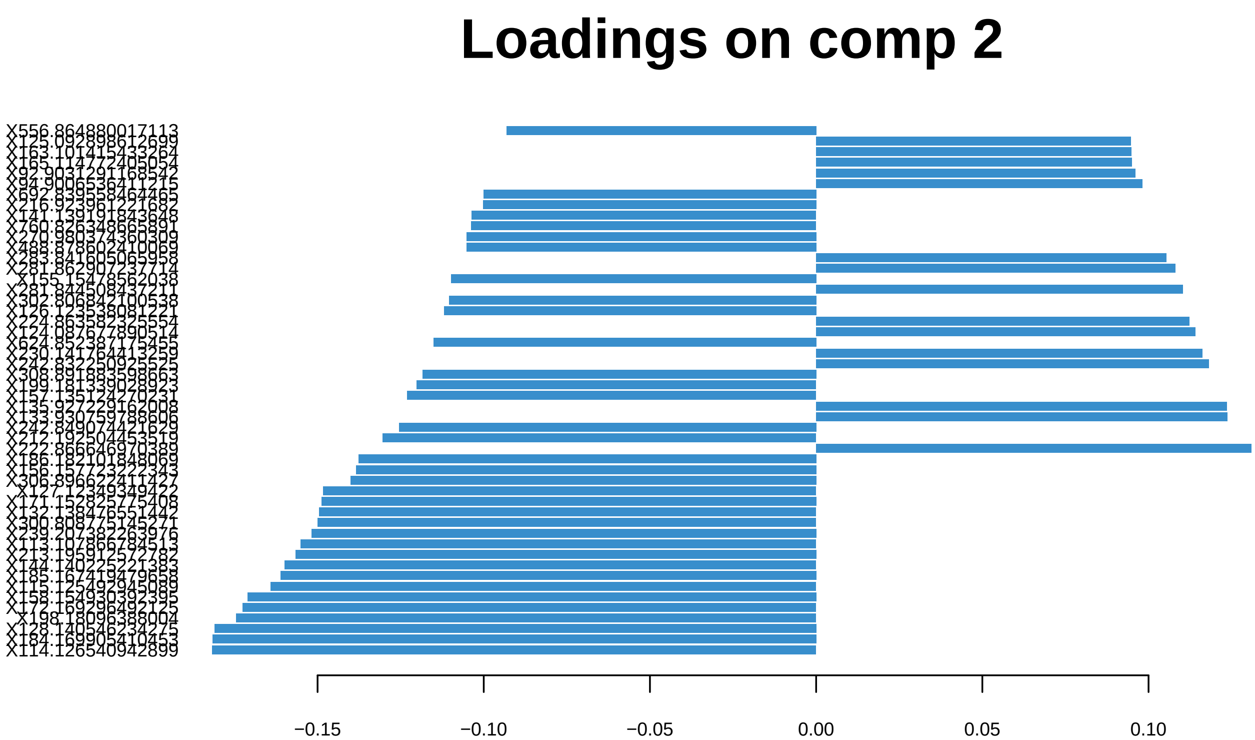
```
## X781.562470345012 0.132132235
## X829.502477789232 0.126553139
## X518.326555182277 0.125737893
## X742.476200409555 0.123710029
## X794.568765896594 0.123558466
## X318.301936686386 0.117975433
## X734.456270192253 0.114664379
## X733.452831026669 0.114303773
## X747.432280460354 0.113584596
## X793.569482767912 0.113278695
## X792.566391872827 0.113173936
## X246.244026630445 0.112776140
## X326.379188131675 0.106900159
## X382.705648118746 0.106833961
## X827.492635970453 0.105282182
## X748.43588669705  0.104892985
## X571.456275345093 0.103289959
## X411.197745550511 0.102627905
## X382.205104740423 0.101332014
## X432.750951893732 0.099918659
## X286.311793629783 0.098645924
## X410.188828237828 0.095175090
## X409.185152883057 0.094132284
## X362.328187231192 0.092960946
## X765.445581943467 0.091584912
## X333.244766441338 0.090869187
## X519.329371282499 0.089682623
## X363.331227311425 0.080353159
## X532.410240682773 0.079368347
## X414.739690648256 0.067791427
## X309.253134340825 0.067106545
## X332.332789682947 0.065975546
## X1183.78973796016 0.060171240
## X1182.78522288059 0.056580213
## X1182.28366567391 0.055405026
## X596.478037801955 0.054532813
## X281.222474726652 0.053342219
## X1181.78090187063 0.052643191
## X340.283564752252 0.052030673
## X423.74507270066  0.051520306
## X387.181977224108 0.050024375
## X1183.28910429712 0.048964580
## X375.216239987552 0.048325820
## X850.532999703027 0.044751553
## X614.489382227607 0.044712660
## X432.249992388519 0.044096815
## X851.536200338849 0.042511539
## X334.296207062515 0.041511119
## X617.422434584689 0.040892373
## X600.423865799132 0.040450993
## X531.406294563801 0.039938451
## X388.185474387062 0.038037599
## X375.715726205822 0.037751982
## X795.424433772223 0.033949094
```

```
## X1344.83974186844 0.031175120
## X423.246011998791 0.029381703
## X274.275878507362 0.027867860
## X595.475042420081 0.026042544
## X657.413334779106 0.022971788
## X986.60881151063  0.021944818
## X415.213245299454 0.020714708
## X207.140250943762 0.018507543
## X409.164099604973 0.018150977
## X538.309635918706 0.017914136
## X393.192032923759 0.017455099
## X896.51647657354  0.017379232
## X875.487665919143 0.017004776
## X800.537939870678 0.016738006
## X830.514148662257 0.016441777
## X537.305924903602 0.010246862
## X864.512061516746 0.009965227
## X1589.02230273985 0.008643349
## X410.167418682302 0.006399847
## X550.44710345109  0.002474784
```

```r
# plot loadings for comp 1
plotLoadings(spca.res, ndisplay = 50)
```

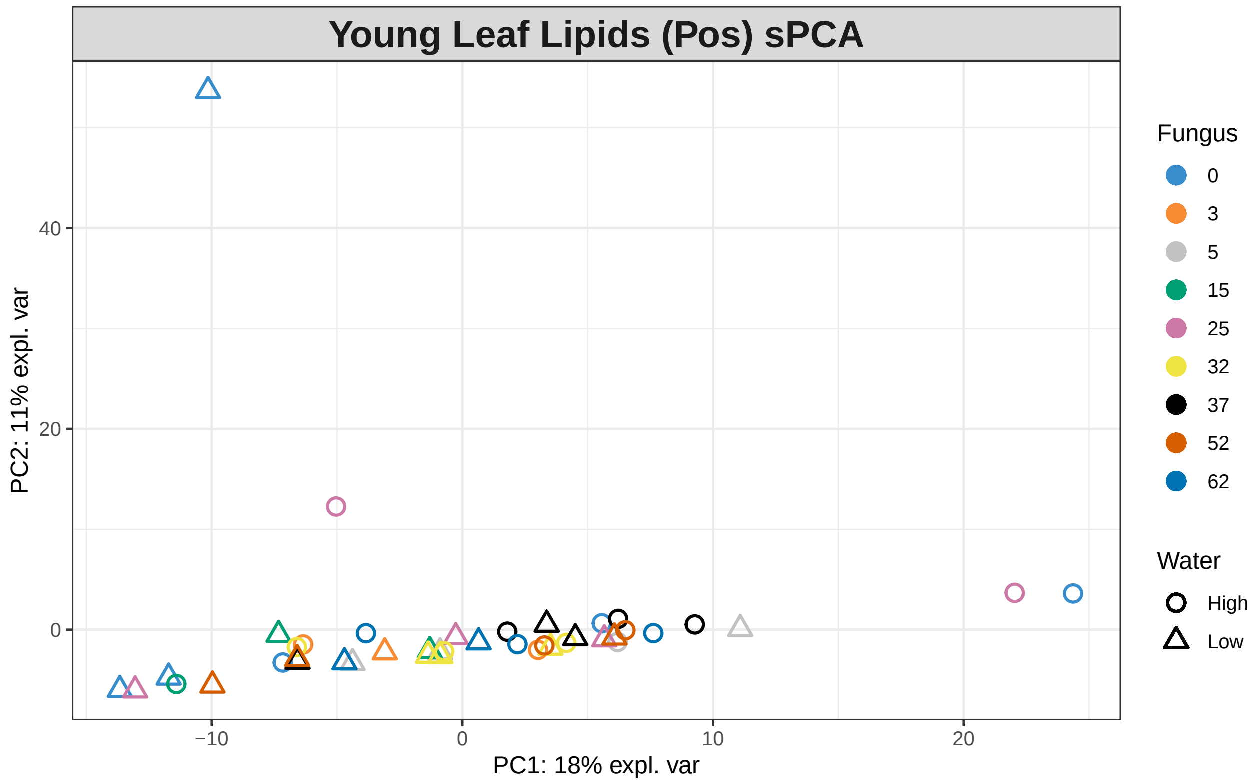



```r
# plot loadings for comp 2
plotLoadings(spca.res, comp = 2, ndisplay = 50)
```

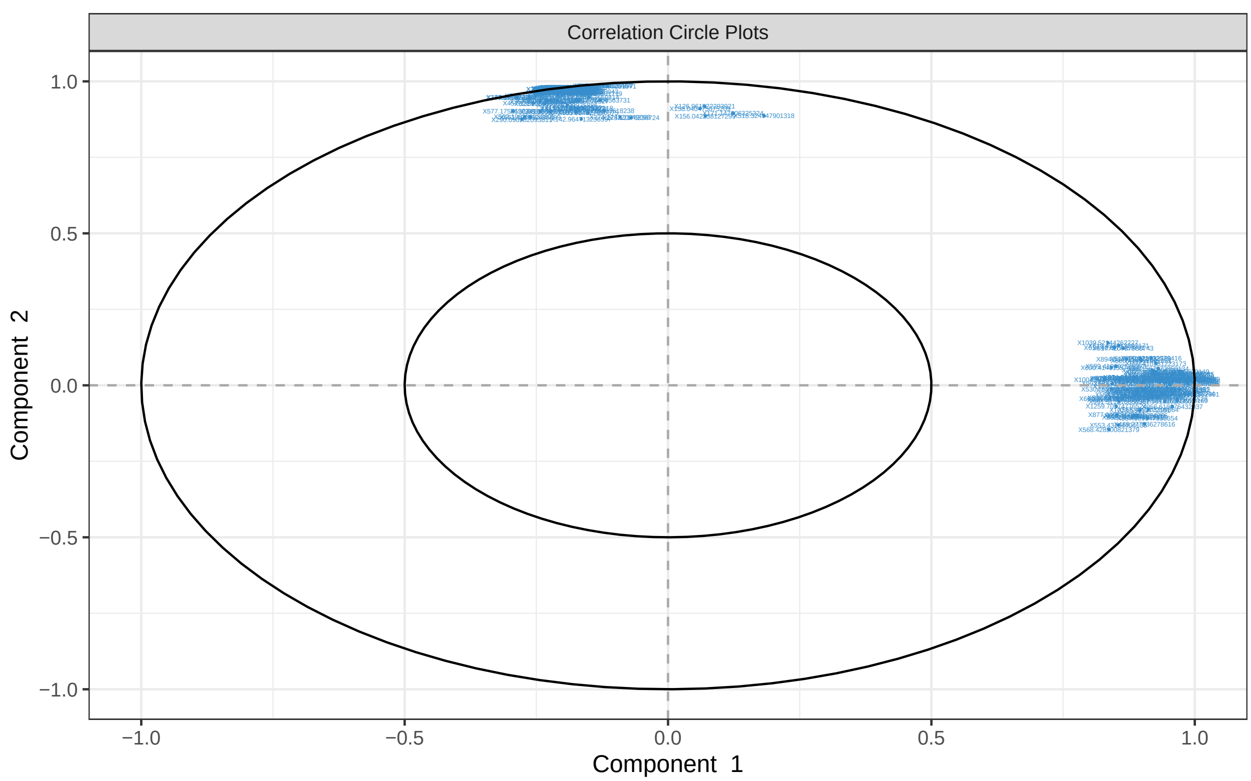



```
# Young Leaf
spca.res <- mixOmics::spca(scaled_Y_young, ncomp = 3, keepX = c(100,
    100, 10))

# plot spca
plotIndiv(spca.res, group = class$Fungus, ind.names = F, pch = as.factor(class$Water),
    legend = T, legend.title = "Fungus", legend.title.pch = "Water",
    title = "Young Leaf Lipids (Pos) sPCA")
```




```r
# variables contributing to each component
plotVar(spca.res, cex = 1)
```

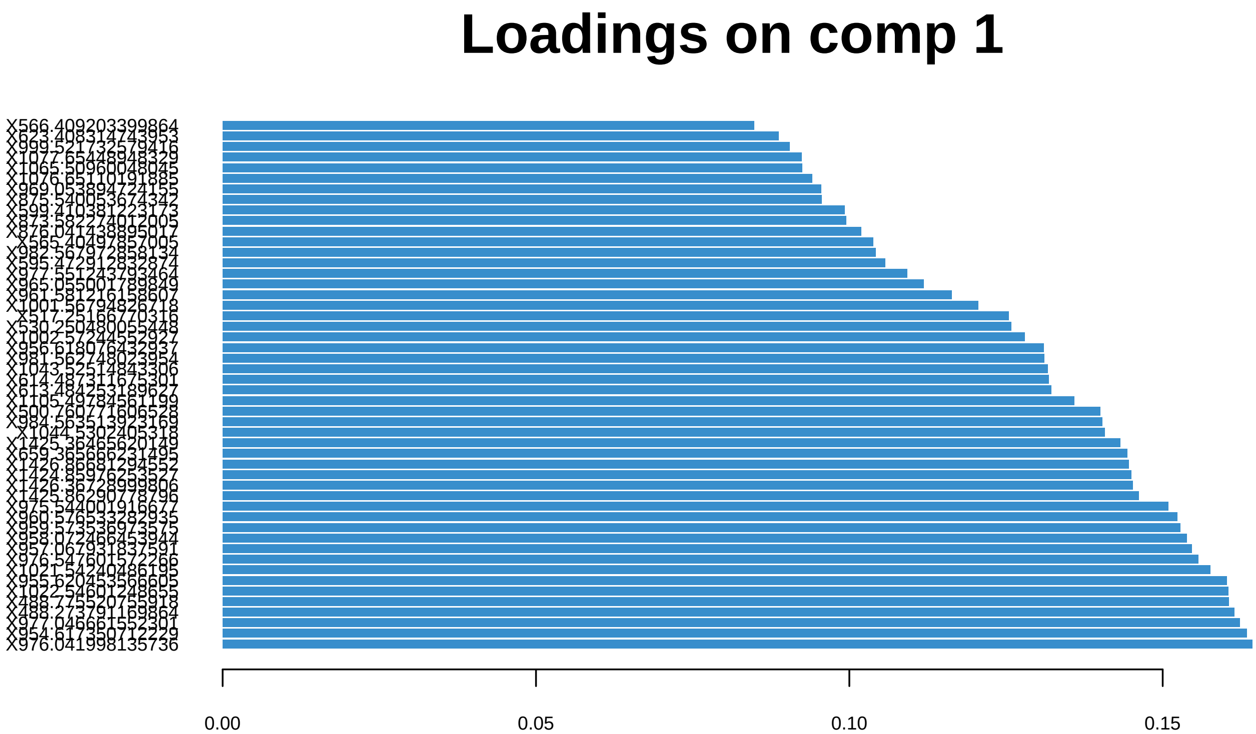


```r
selectVar(spca.res, comp = 1)$value  # view loading value of each metabolite
```

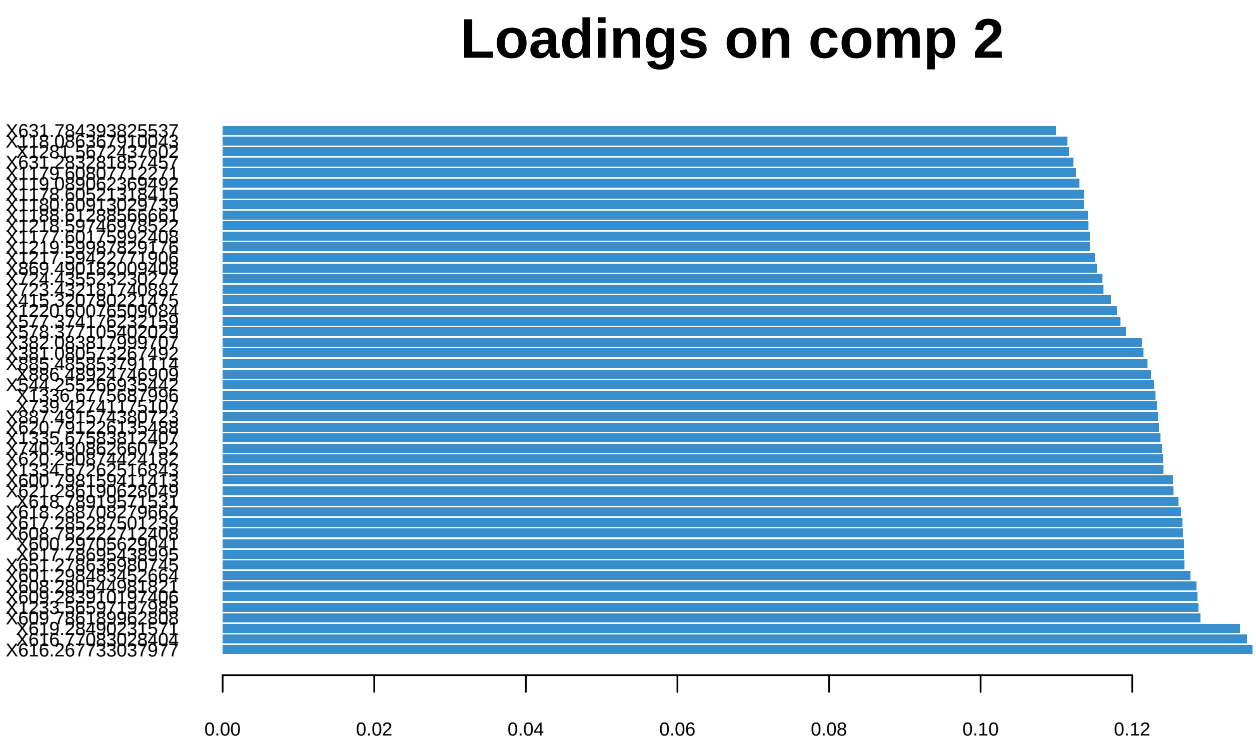
```
##                      value.var
## X976.041998135736 0.164272822
## X954.617350712229 0.163448534
## X977.046661552301 0.162306422
## X488.273791169864 0.161458674
## X488.775520755918 0.160536781
## X1022.54601248655 0.160489386
## X955.620453566605 0.160190374
## X1021.54240486195 0.157593782
## X976.547601572266 0.155665956
## X957.067931837591 0.154670263
## X958.072466453944 0.153812952
## X959.573536973575 0.152794534
## X960.576533282935 0.152322088
## X975.544001916677 0.150883330
## X1425.86290778796 0.146142511
## X1426.36728999806 0.145229770
## X1424.85976253527 0.144990862
## X1426.86681294552 0.144565735
## X659.365666231495 0.144306082
## X1425.36465620149 0.143198475
## X1044.5302405318  0.140736973
## X984.563513923169 0.140323494
## X500.760771606528 0.140069527
## X1105.49784561199 0.135875269
## X613.484253189627 0.132178575
## X614.487311675301 0.131782873
## X1043.52514843306 0.131659191
## X981.562748023954 0.131115658
## X956.618076432937 0.131013597
## X1002.57244552927 0.127964995
## X530.250480055448 0.125847541
## X517.25166770316  0.125451005
## X1001.56794826718 0.120599063
## X961.581216158607 0.116334098
## X965.055001789849 0.111857681
## X977.551243793464 0.109211089
## X595.472912832874 0.105682274
## X982.567972858134 0.104196176
## X565.40497857005  0.103787999
## X876.041438895017 0.101865450
## X873.582274012005 0.099474539
## X599.410381223173 0.099252065
## X875.540053674342 0.095547062
## X969.053894724155 0.095534693
## X1076.65110191885 0.094061557
## X1065.50960048045 0.092477157
## X1077.65448948329 0.092406666
## X999.521732579416 0.090490988
## X623.408314743953 0.088684055
## X566.409203399864 0.084787818
## X957.609347356654 0.083278521
```

```
## X624.411930213696 0.080258317
## X1027.55778323275 0.080073455
## X489.277936278616 0.078066626
## X978.555223013531 0.076957472
## X1028.56349217506 0.076554593
## X480.790480251846 0.074970226
## X349.161929131895 0.074074623
## X600.414332602851 0.071518021
## X517.752878732788 0.070620078
## X1023.55012772181 0.069077483
## X876.543723964092 0.066186012
## X946.076362435254 0.064245671
## X1078.65902624265 0.064236605
## X584.419196792614 0.063199104
## X551.425843612517 0.063176518
## X945.575354442912 0.062090676
## X552.429062975079 0.058517505
## X583.415855136359 0.058038828
## X871.576507868922 0.057795471
## X874.584651876624 0.057747623
## X962.586217446947 0.055082584
## X895.015665583442 0.046338032
## X894.514915085236 0.043939574
## X496.260198028083 0.043827329
## X657.408346877498 0.043428948
## X502.259240592042 0.041649504
## X583.415516353551 0.038963095
## X1106.50260362597 0.038025338
## X617.420737980743 0.037213469
## X496.762620578608 0.036499356
## X909.532512628987 0.032323205
## X619.43415426729  0.031295823
## X507.750369455364 0.030877346
## X991.511218688387 0.029628460
## X618.424413292171 0.028967963
## X553.43164404136  0.028058213
## X847.519069497488 0.027288267
## X985.567543188756 0.026943520
## X877.04633470551  0.025191092
## X1259.70574776524 0.024073045
## X599.410692576378 0.023695334
## X614.487577459701 0.020638426
## X992.515712573244 0.016347061
## X530.752892698683 0.015324619
## X600.414315574091 0.014573064
## X660.368635116281 0.011751384
## X568.428500821379 0.010047001
## X1039.52144262227 0.008062114
## X1004.54062609616 0.001874783
```

```r
# plot loadings for comp 1
plotLoadings(spca.res, ndisplay = 50)
```



```r
# plot loadings for comp 2
plotLoadings(spca.res, comp = 2, ndisplay = 50)
```

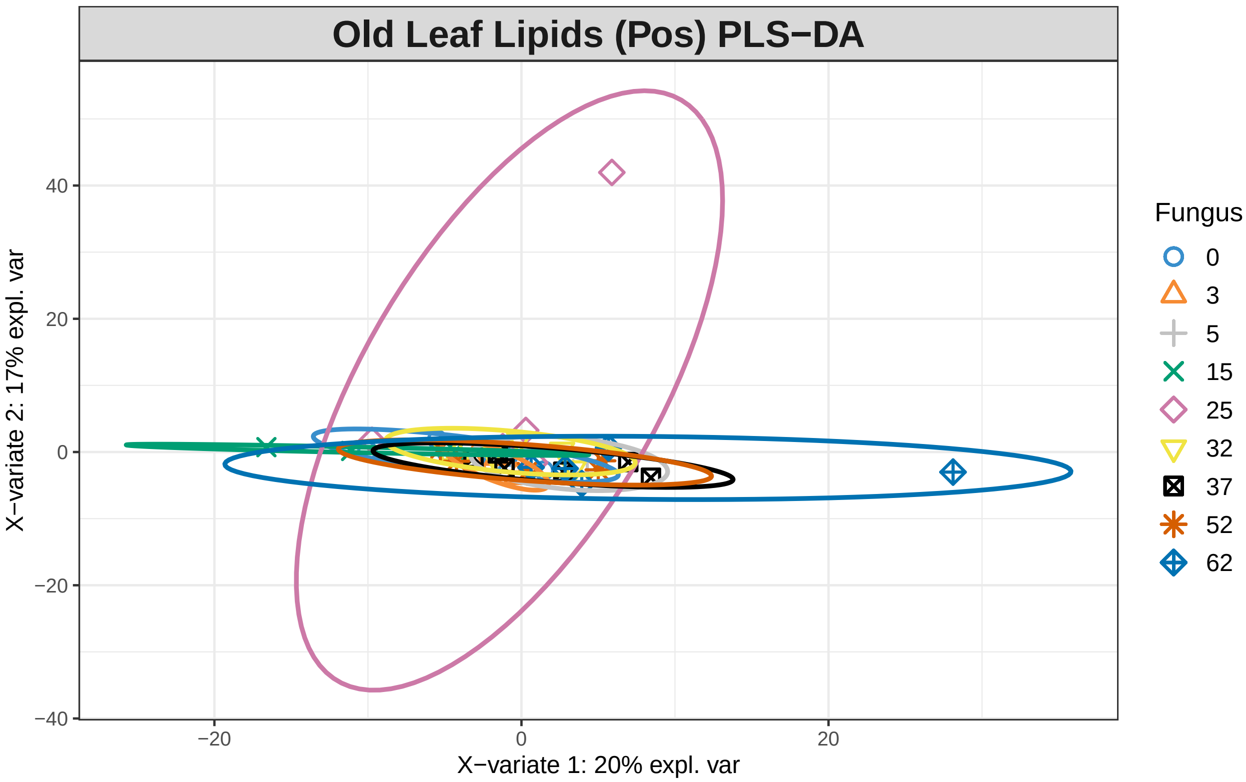


PLS-DA

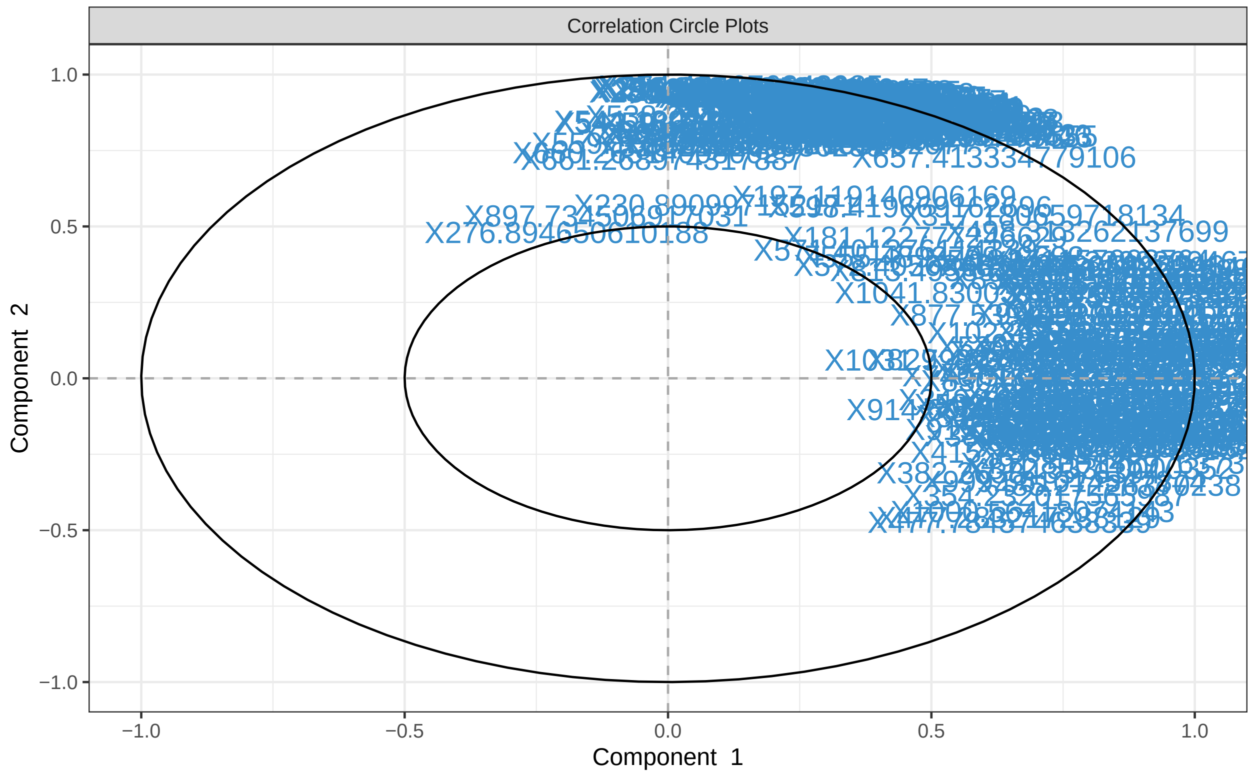
10. Classify samples into known groups and predict the class of new samples.

```r
# Old Leaf
old.splsda <- mixOmics::splsda(scaled_Y_old, class$Fungus, keepX = c(100,
    100))

# plot pls-da
plotIndiv(old.splsda, ind.names = F, legend = T, title = "Old Leaf Lipids (Pos) PLS-DA",
    legend.title = "Fungus", ellipse = T)
```




```r
# plot and select the variables
plotVar(old.splsda)
```




```r
selectVar(old.splsda, comp = 1)
```

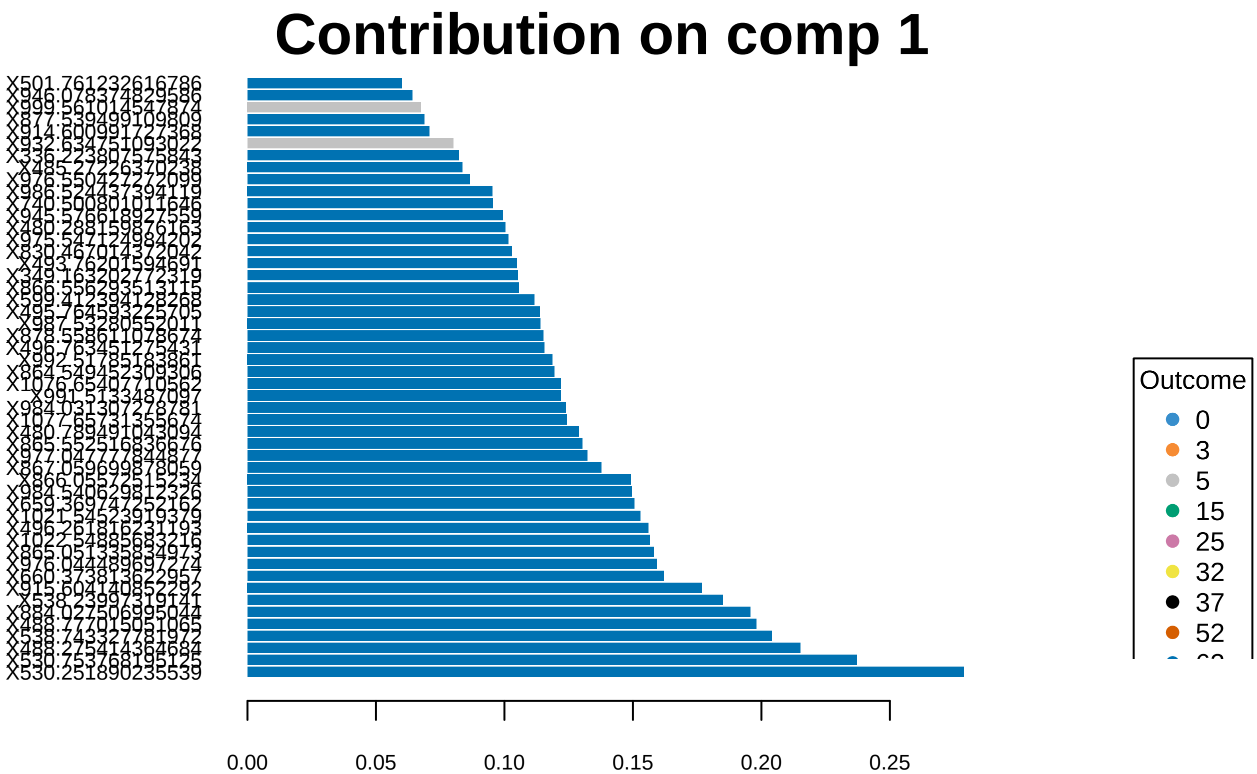
```
## $name
##   [1] "X530.251890235539" "X530.753768195125" "X488.275414364684"
##   [4] "X538.743327781972" "X488.777015051065" "X884.027506995044"
##   [7] "X538.23997319141"  "X915.604140852292" "X660.373813622957"
##  [10] "X976.044489697274" "X865.051335834973" "X1022.54885683216"
##  [13] "X496.261816231193" "X1021.54523919379" "X659.369747252162"
##  [16] "X984.540629812326" "X866.05572515234"  "X867.059699878059"
##  [19] "X977.047777844877" "X865.552516836676" "X480.789491043094"
##  [22] "X1077.65731355674" "X984.031307278781" "X991.5133487097"
##  [25] "X1076.65407710562" "X864.549452309306" "X992.51785183861"
##  [28] "X496.763451275431" "X878.558611078674" "X987.53280552011"
##  [31] "X495.764593225705" "X599.412394128268" "X866.556293513115"
##  [34] "X349.163202772319" "X493.76201594691"  "X830.467014372042"
##  [37] "X975.547124984202" "X480.288159876163" "X945.576618927559"
##  [40] "X740.500801011646" "X986.524437394119" "X976.550427272099"
##  [43] "X485.27226370238"  "X336.223807575843" "X932.634751093022"
##  [46] "X914.600991727368" "X877.539499109809" "X999.561014547874"
##  [49] "X946.078374829586" "X501.761232616786" "X539.249113810979"
##  [52] "X600.4158642877"   "X364.254362067022" "X295.196082045934"
##  [55] "X613.48620846313"  "X879.065097543885" "X415.7365415304"
##  [58] "X317.180659718134" "X1445.34303632503" "X496.775134400763"
##  [61] "X502.25999467882"  "X1041.83003958831" "X872.53310000352"
##  [64] "X489.27954788019"  "X994.523205661239" "X946.582521635004"
##  [67] "X955.623184556673" "X877.04903182984"  "X1000.56418074143"
##  [70] "X959.575997758434" "X893.618592737698" "X477.283217238129"
##  [73] "X497.273508649734" "X878.055130721582" "X354.232017565987"
##  [76] "X957.070210444299" "X335.260157312681" "X954.619922134736"
```

```
##  [79] "X504.271157003764" "X960.578907599357" "X829.462641361221"
##  [82] "X342.239823636993" "X1023.55357784034" "X880.070800931978"
##  [85] "X977.554810339346" "X601.427805825546" "X349.663922662639"
##  [88] "X1444.33993604679" "X491.800145976373" "X758.531383058781"
##  [91] "X813.495852628813" "X1031.79972614852" "X296.199163537393"
##  [94] "X382.263843576394" "X498.313262137699" "X958.074259394391"
##  [97] "X477.784574638339" "X595.4746818897"   "X933.638024103796"
## [100] "X1444.84082391465"
##
## $value
##                      value.var
## X530.251890235539 0.2787399150
## X530.753768195125 0.2371262941
## X488.275414364684 0.2150767431
## X538.743327781972 0.2039737100
## X488.777015051065 0.1979515943
## X884.027506995044 0.1957315327
## X538.23997319141  0.1850202728
## X915.604140852292 0.1768660602
## X660.373813622957 0.1619652834
## X976.044489697274 0.1592902240
## X865.051335834973 0.1580346338
## X1022.54885683216 0.1565880346
## X496.261816231193 0.1560505238
## X1021.54523919379 0.1528475130
## X659.369747252162 0.1504923648
## X984.540629812326 0.1494937360
## X866.05572515234  0.1492380737
## X867.059699878059 0.1377467341
## X977.047777844877 0.1321578287
## X865.552516836676 0.1303586523
## X480.789491043094 0.1289828655
## X1077.65731355674 0.1242133843
## X984.031307278781 0.1238722544
## X991.5133487097   0.1219637941
## X1076.65407710562 0.1219308145
## X864.549452309306 0.1193376111
## X992.51785183861  0.1187054405
## X496.763451275431 0.1154751633
## X878.558611078674 0.1150664680
## X987.53280552011  0.1140340310
## X495.764593225705 0.1138238999
## X599.412394128268 0.1115587261
## X866.556293513115 0.1056436222
## X349.163202772319 0.1051028921
## X493.76201594691  0.1047411455
## X830.467014372042 0.1029089014
## X975.547124984202 0.1015381266
## X480.288159876163 0.1002706043
## X945.576618927559 0.0993092193
## X740.500801011646 0.0954256877
## X986.524437394119 0.0953498581
## X976.550427272099 0.0864485551
## X485.27226370238  0.0836839142
```

```
## X336.223807575843 0.0822740327
## X932.634751093022 0.0800578018
## X914.600991727368 0.0707202362
## X877.539499109809 0.0688353460
## X999.561014547874 0.0675343722
## X946.078374829586 0.0641602966
## X501.761232616786 0.0601006955
## X539.249113810979 0.0589843204
## X600.4158642877   0.0588990331
## X364.254362067022 0.0578590721
## X295.196082045934 0.0568541536
## X613.48620846313  0.0566359178
## X879.065097543885 0.0543763280
## X415.7365415304   0.0531126915
## X317.180659718134 0.0490692525
## X1445.34303632503 0.0485941801
## X496.775134400763 0.0480602511
## X502.25999467882  0.0476392303
## X1041.83003958831 0.0465025280
## X872.53310000352  0.0443540775
## X489.27954788019  0.0424362509
## X994.523205661239 0.0385777808
## X946.582521635004 0.0375775847
## X955.623184556673 0.0354372477
## X877.04903182984  0.0344928726
## X1000.56418074143 0.0317210173
## X959.575997758434 0.0301766867
## X893.618592737698 0.0288806933
## X477.283217238129 0.0288038601
## X497.273508649734 0.0265632684
## X878.055130721582 0.0261134717
## X354.232017565987 0.0252260680
## X957.070210444299 0.0232778198
## X335.260157312681 0.0230247719
## X954.619922134736 0.0222957611
## X504.271157003764 0.0213450974
## X960.578907599357 0.0210706752
## X829.462641361221 0.0203366521
## X342.239823636993 0.0175492491
## X1023.55357784034 0.0146647413
## X880.070800931978 0.0133300332
## X977.554810339346 0.0122406310
## X601.427805825546 0.0112007391
## X349.663922662639 0.0111267919
## X1444.33993604679 0.0104216536
## X491.800145976373 0.0099582362
## X758.531383058781 0.0097578262
## X813.495852628813 0.0094676399
## X1031.79972614852 0.0074113278
## X296.199163537393 0.0069973762
## X382.263843576394 0.0066865175
## X498.313262137699 0.0057358045
## X958.074259394391 0.0045261897
## X477.784574638339 0.0042601516
```

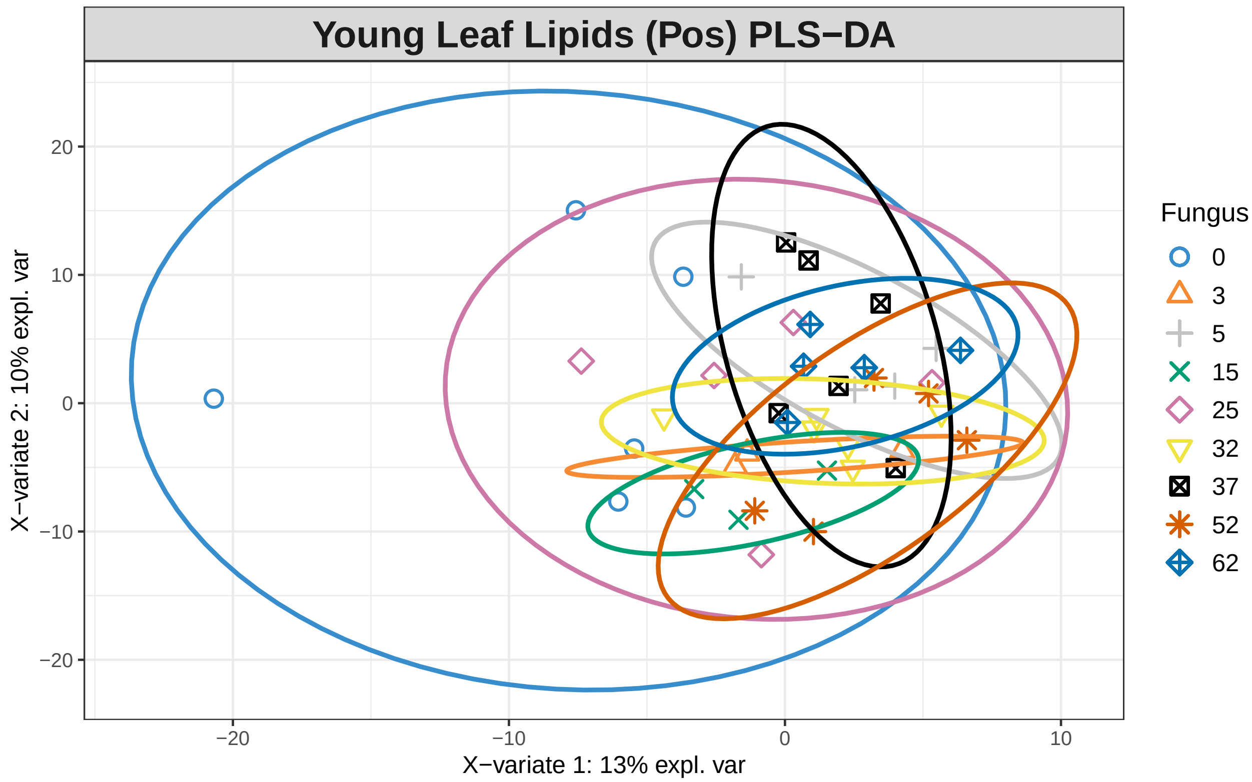
```
## X595.4746818897   0.0025486460
## X933.638024103796 0.0006263414
## X1444.84082391465 0.0004178260
##
## $comp
## [1] 1
```

```r
plotLoadings(old.splsda, contrib = "max", method = "mean", ndisplay = 50)
```

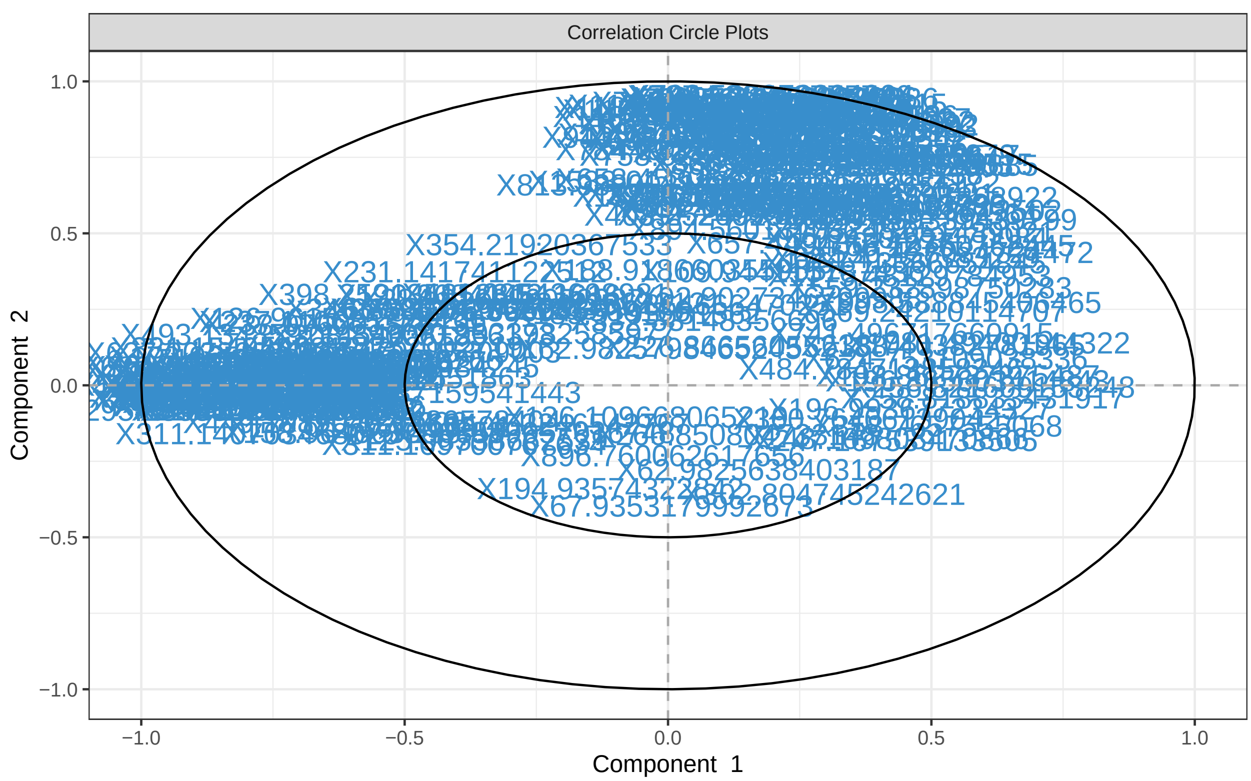



```r
# Young Leaf
young.splsda <- mixOmics::splsda(scaled_Y_young, class$Fungus,
    keepX = c(100, 100))

# plot pls-da
plotIndiv(young.splsda, ind.names = F, legend = T, title = "Young Leaf Lipids (Pos) PLS-DA",
    legend.title = "Fungus", ellipse = T)
```




```r
# plot and select the variables
plotVar(young.splsda)
```



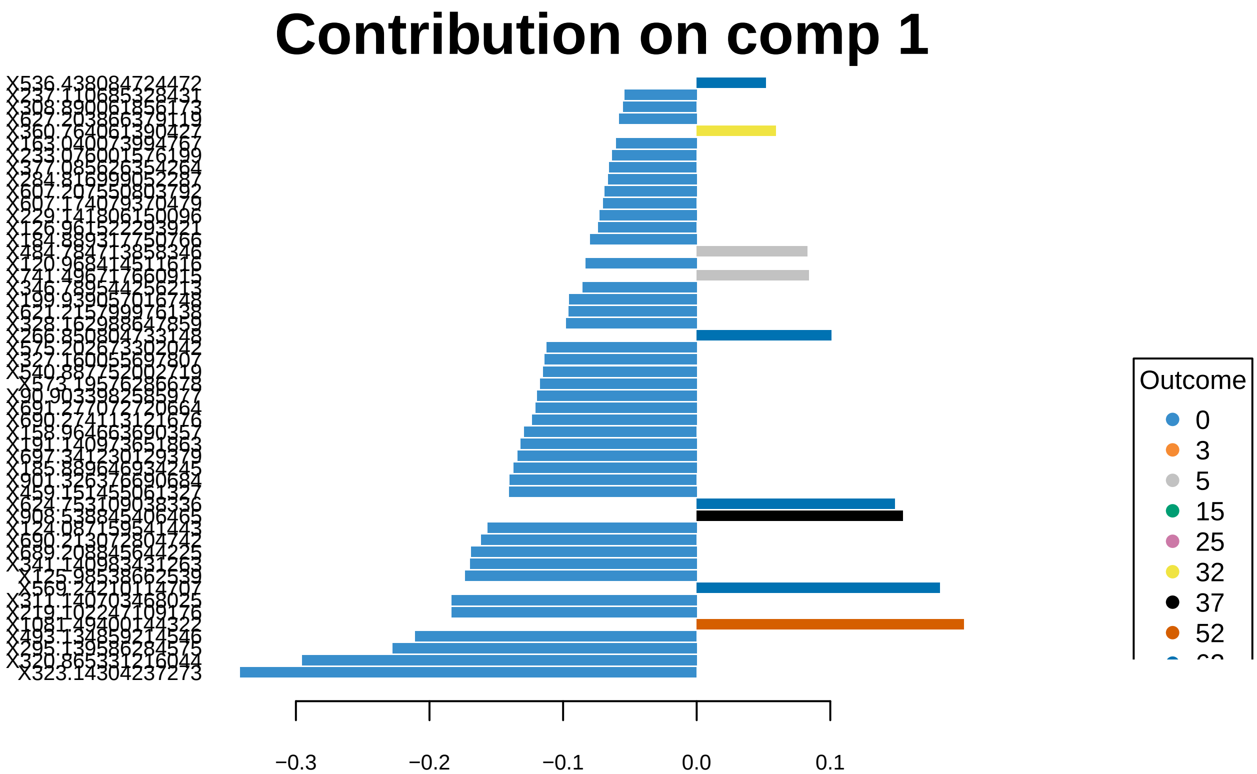
```r
selectVar(young.splsda, comp = 1)
```

```
## $name
##   [1] "X323.14304237273"  "X320.865331216044" "X295.139586284575"
##   [4] "X493.134859214546" "X1081.49400144322" "X219.102247109176"
##   [7] "X311.140703468025" "X569.24210114707"  "X125.98538662539"
##  [10] "X341.140983431263" "X689.208845644225" "X690.213072804742"
##  [13] "X124.087159541443" "X908.538845406465" "X624.753109038336"
##  [16] "X459.151455061327" "X901.326376690684" "X185.889646934245"
##  [19] "X697.341230129379" "X191.140973651863" "X158.964663690357"
##  [22] "X690.274113121676" "X691.277072720664" "X90.9033982585977"
##  [25] "X573.19576286678"  "X540.887752002719" "X327.160055697807"
##  [28] "X575.202673302042" "X266.850804733148" "X328.162988647859"
##  [31] "X621.215799976138" "X199.939057016748" "X346.789544256213"
##  [34] "X741.496717660915" "X120.968414511616" "X484.784713858346"
##  [37] "X184.889317750766" "X126.961522293921" "X229.141806150096"
##  [40] "X607.174079370479" "X607.207550803792" "X284.816999052287"
##  [43] "X377.085626354264" "X233.076001576199" "X163.040073994767"
##  [46] "X360.764061390427" "X627.203866379119" "X308.890061856173"
##  [49] "X237.110685328431" "X536.438084724472" "X358.7657555068"
##  [52] "X561.192552485699" "X282.094893241756" "X711.191695035036"
##  [55] "X196.993677234327" "X273.081274829421" "X120.088896864637"
##  [58] "X177.016364210501" "X549.218901747332" "X562.196381837805"
##  [61] "X281.088456911411" "X179.065348185315" "X231.141741122512"
##  [64] "X609.189377269662" "X196.865603414843" "X378.164038356871"
##  [67] "X190.126486578404" "X559.232898750283" "X638.740382781965"
##  [70] "X377.160255395164" "X137.060663669955" "X660.80142026303"
##  [73] "X198.862737375487" "X311.169700767634" "X569.199478252238"
##  [76] "X119.089062369492" "X354.21920367533"  "X265.106604762037"
##  [79] "X1175.48129784953" "X547.215180396862" "X548.21957840366"
##  [82] "X617.150214310121" "X334.179688309003" "X359.149843804862"
##  [85] "X302.804745242621" "X379.112996191006" "X247.167539133505"
##  [88] "X411.09543608981"  "X177.055302334209" "X274.079850557197"
##  [91] "X256.821609153248" "X634.452246542767" "X596.177215143824"
##  [94] "X595.17823477301"  "X449.108904670796" "X248.169808470866"
##  [97] "X331.15494958124"  "X278.895045500166" "X258.818833431917"
## [100] "X621.286190628049"
##
## $value
##                       value.var
## X323.14304237273  -0.3416167438
## X320.865331216044 -0.2955018499
## X295.139586284575 -0.2276259767
## X493.134859214546 -0.2106263799
## X1081.49400144322  0.1998892746
## X219.102247109176 -0.1834834966
## X311.140703468025 -0.1834650636
## X569.24210114707   0.1820321373
## X125.98538662539  -0.1733721746
## X341.140983431263 -0.1697275623
## X689.208845644225 -0.1688705084
## X690.213072804742 -0.1612449742
## X124.087159541443 -0.1564560851
## X908.538845406465  0.1543282802
```

```
## X624.753109038336  0.1482127350
## X459.151455061327 -0.1405048279
## X901.326376690684 -0.1399217497
## X185.889646934245 -0.1370141931
## X697.341230129379 -0.1339946588
## X191.140973651863 -0.1318479906
## X158.964663690357 -0.1290596063
## X690.274113121676 -0.1231780789
## X691.277072720664 -0.1206881434
## X90.9033982585977 -0.1195010844
## X573.19576286678  -0.1172479650
## X540.887752002719 -0.1150723215
## X327.160055697807 -0.1139831185
## X575.202673302042 -0.1123408534
## X266.850804733148  0.1010303807
## X328.162988647859 -0.0979358756
## X621.215799976138 -0.0960407330
## X199.939057016748 -0.0956320989
## X346.789544256213 -0.0853501324
## X741.496717660915  0.0839611945
## X120.968414511616 -0.0833105448
## X484.784713858346  0.0827437432
## X184.889317750766 -0.0798891147
## X126.961522293921 -0.0737213683
## X229.141806150096 -0.0727801918
## X607.174079370479 -0.0699263459
## X607.207550803792 -0.0689839161
## X284.816999052287 -0.0662973673
## X377.085626354264 -0.0654950136
## X233.076001576199 -0.0632242865
## X163.040073994767 -0.0603304097
## X360.764061390427  0.0591449838
## X627.203866379119 -0.0580548334
## X308.890061856173 -0.0550110532
## X237.110685328431 -0.0539797209
## X536.438084724472  0.0519614676
## X358.7657555068    0.0492060182
## X561.192552485699 -0.0485448680
## X282.094893241756 -0.0479308899
## X711.191695035036 -0.0476566818
## X196.993677234327  0.0470686061
## X273.081274829421 -0.0454925159
## X120.088896864637 -0.0454761313
## X177.016364210501 -0.0452816943
## X549.218901747332 -0.0432890869
## X562.196381837805 -0.0418095894
## X281.088456911411 -0.0415812997
## X179.065348185315 -0.0409394208
## X231.141741122512 -0.0403002997
## X609.189377269662 -0.0399387350
## X196.865603414843  0.0376777031
## X378.164038356871 -0.0369785833
## X190.126486578404 -0.0351672677
## X559.232898750283  0.0341601357
```

```
## X638.740382781965  0.0333015118
## X377.160255395164 -0.0329288226
## X137.060663669955 -0.0310865370
## X660.80142026303  -0.0302963470
## X198.862737375487  0.0288442438
## X311.169700767634 -0.0255553101
## X569.199478252238 -0.0254921289
## X119.089062369492 -0.0211187434
## X354.21920367533  -0.0209744779
## X265.106604762037 -0.0184500339
## X1175.48129784953  0.0182995905
## X547.215180396862 -0.0181184559
## X548.21957840366  -0.0178552376
## X617.150214310121 -0.0173720762
## X334.179688309003 -0.0167524117
## X359.149843804862 -0.0166710091
## X302.804745242621  0.0155077124
## X379.112996191006 -0.0154490252
## X247.167539133505  0.0143220385
## X411.09543608981  -0.0122302715
## X177.055302334209 -0.0114308940
## X274.079850557197 -0.0111453338
## X256.821609153248  0.0111005249
## X634.452246542767 -0.0103238264
## X596.177215143824 -0.0102794991
## X595.17823477301  -0.0100377883
## X449.108904670796 -0.0085082813
## X248.169808470866  0.0059334461
## X331.15494958124  -0.0048366868
## X278.895045500166 -0.0036005552
## X258.818833431917  0.0031083231
## X621.286190628049 -0.0008631579
##
## $comp
## [1] 1
```

```
plotLoadings(young.splsda, contrib = "max", method = "mean",
    ndisplay = 50)
```



Heatmaps of Averaged Data

11. Create averaged metabolite matrices and rerun PLS-DA to create a heatmap.

```
## Young Leaves
av_Y_young <- aggregate(Y_young, by = list(class$Water, class$Fungus),
    FUN = "mean", simplify = T, data = class)
av.young.plsda <- mixOmics::plsda(av_Y_young[, 3:1755], av_Y_young$Group.2)  # fungus

# heatmap
youngcim <- cim(av.young.plsda, title = "Young Leaf Lipids (pos) Averaged Over Fungi",
    col.names = F, xlab = "Lipids", save = "png", name.save = "~/Box/Summer 2018 TX Endo Field Samples
```

Indicator Analysis

12. Identify indicator metabolites characteristic of each treatment using Dufrene-Legendre Indicator Analysis.

```
# Old Leaf
indicator_Fungus <- indval(Y_young, clustering = class$Fungus,
    numitr = 999, type = "long")

# Young Leaf
indicator_Fungus <- indval(Y_young, clustering = class$Fungus,
    numitr = 999, type = "long")
```

13. Disect indval object.

```
Orelfrq <- indicator_Fungus$relfrq  # relative frequency of species in classes
Orelabu <- indicator_Fungus$relabu  # relative abundance of species in classes
Oindval <- indicator_Fungus$indval  # the indicator value for each species
```

```r
Omaxcls <- data.frame(indicator_Fungus$maxcls)  # the class each species has max indicator value for
Oindcls <- data.frame(indicator_Fungus$indcls)  # the indicator value for each species to its max class
Opval <- data.frame(indicator_Fungus$pval)  # the probability of obtaining as high an indicator value a

Yrelfrq <- indicator_Fungus$relfrq  # relative frequency of species in classes
Yrelabu <- indicator_Fungus$relabu  # relative abundance of species in classes
Yindval <- indicator_Fungus$indval  # the indicator value for each species
Ymaxcls <- data.frame(indicator_Fungus$maxcls)  # the class each species has max indicator value for
Yindcls <- data.frame(indicator_Fungus$indcls)  # the indicator value for each species to its max class
Ypval <- data.frame(indicator_Fungus$pval)  # the probability of obtaining as high an indicator value a
```

14. Export results to a csv file.

```r
write.csv(cbind(Orelfrq, Orelabu, Oindval, Omaxcls, Oindcls,
    Opval), "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/Lipi
write.csv(cbind(Yrelfrq, Yrelabu, Yindval, Ymaxcls, Yindcls,
    Ypval), "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/Lipi
```

References

Collyer, M.L., Adams, D.C. 2018. RRPP: An r package for fitting linear models to high-dimensional data using residual randomization. Methods in Ecology and Evolution. 9(7):1772-1779.

Dufrene, M. and Legendre, P. 1997. Species assemblages and indicator species: the need for a flexible asymmetrical approach. Ecol. Monogr. 67(3):345-366.

Rohart, F., Gautier, B., Singh, A., & Lê Cao, K. A. 2017. mixOmics: An R package for ‘omics feature selection and multiple data integration. PLoS computational biology, 13(11):e1005752.