

Field Old and Young Leaves Secondary Metabolites 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. The following analysis of secondary 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)
```

Secondary Metabolites (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_SM_neg <- read_tsv(paste(path, "XCMS Online Results/O_SM_Neg/XCMS.annotated.Report_1394387.tsv",
  sep = ""))

Y_SM_neg <- read_tsv(paste(path, "XCMS Online Results/Y_SM_Neg/XCMS.annotated.Report_1394397.tsv",
  sep = ""))

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

Y_young <- Y_SM_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.

```
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: 3734
## 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 46010.13    110817.9 0.2933796 2.075935
##              Z (from F)      Pr(>F)
## Block * Water * Fungus  4.337837 0.0005714286
##
##
## Redundancy Analysis (PCA on fitted values and residuals)
##
##              Trace Proportion Rank
## Fitted      1095.479  0.2933795    7
## Residuals 2638.521  0.7066205   35
## Total      3734.000  1.0000000  42
##
## Eigenvalues
##
##              PC1      PC2      PC3      PC4      PC5      PC6      PC7
## Fitted      564.1858 153.9587 125.7033  96.0683  72.5213  48.1336  34.9085
## Residuals   603.9575 244.3193 216.0016 156.1993 129.7743 107.1920 106.3001
## Total      1006.1235 296.2684 272.7497 229.4345 201.7779 159.0190 137.0564
##              PC8      PC9      PC10     PC11     PC12     PC13     PC14
## Fitted
## Residuals   84.3344  83.2693  73.9679  64.4639  60.8570  54.7399  50.4119
## Total      117.3648  99.5651  83.3967  82.1249  73.8888  70.7485  62.0878
##              PC15     PC16     PC17     PC18     PC19     PC20     PC21
## Fitted
## Residuals   48.0455  45.1743  44.2448  38.6085  36.2209  33.6145  32.5220
## Total      56.5008  54.4177  49.8007  46.9897  42.8943  41.9063  38.2559
##              PC22     PC23     PC24     PC25     PC26     PC27     PC28
## Fitted
## Residuals   31.0776  29.3703  28.4226  26.8550  25.5566  24.1440  23.7667
## Total      37.3648  34.9422  32.5619  31.2924  30.5917  28.3578  27.7198
##              PC29     PC30     PC31     PC32     PC33     PC34     PC35
## Fitted
## Residuals   21.6762  21.4874  20.8524  20.3145  18.5993  17.4643  14.7149
## Total      26.3828  26.1749  24.7238  23.5483  22.9331  22.1958  21.1991
##              PC36     PC37     PC38     PC39     PC40     PC41     PC42
## Fitted
## Residuals
```

```
## Total      19.8343   19.0164   18.6900   17.8323   17.2688   15.2883   13.7105
```

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

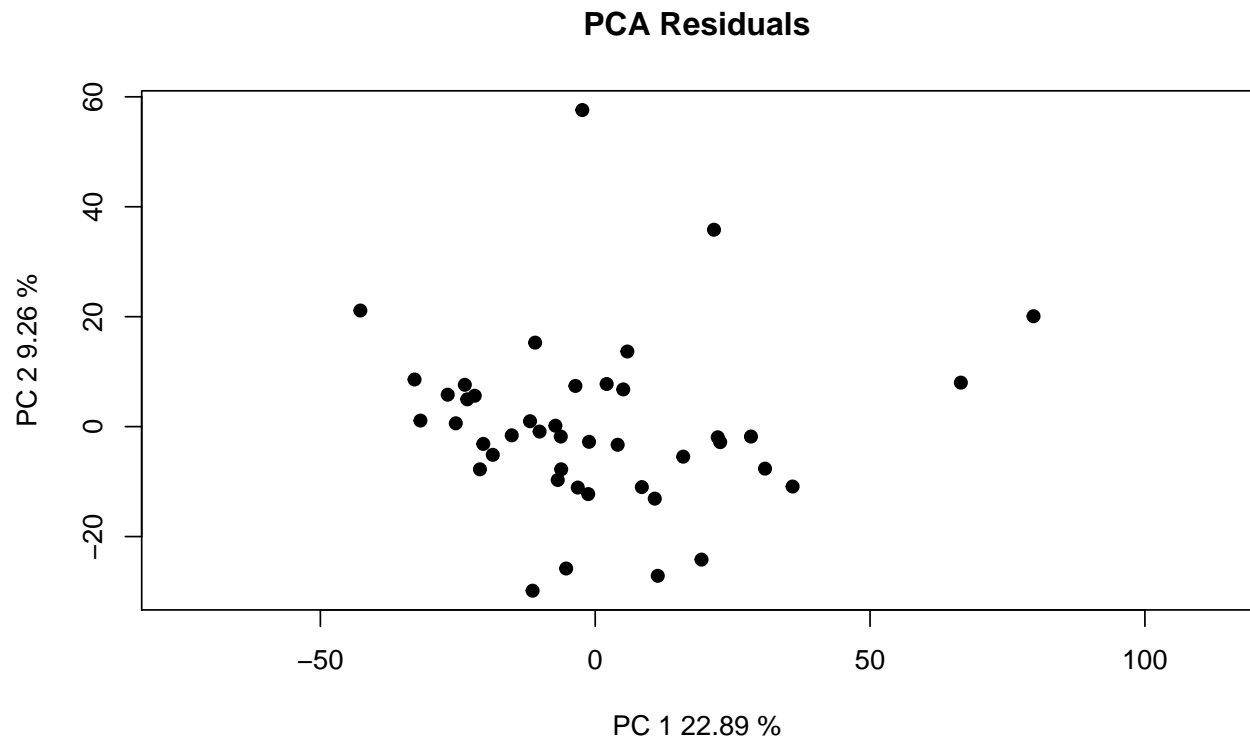
```
##
## Linear Model fit with lm.rrpp
##
## Number of observations: 43
## Number of dependent variables: 2564
## Data space dimensions: 42
## Sums of Squares and Cross-products: Type III
## Number of permutations: 1000
##
## Full Model Analysis of Variance
##
##              Df Residual Df      SS Residual SS      Rsq      F
## Block * Water * Fungus  7          35 31681.28    76006.72 0.2941951 2.084111
##              Z (from F)      Pr(>F)
## Block * Water * Fungus  3.585621 0.0005714286
##
##
## Redundancy Analysis (PCA on fitted values and residuals)
##
##              Trace Proportion Rank
## Fitted      754.3162 0.2941951    7
## Residuals 1809.6838 0.7058049   35
## Total      2564.0000 1.0000000   42
##
## Eigenvalues
##
##              PC1      PC2      PC3      PC4      PC5      PC6      PC7
## Fitted      479.1294  91.1236  60.3231  48.3523  34.7519  22.3644  18.2714
## Residuals  453.7519 244.4531 180.4838 108.3626  99.0782  86.2482  70.5464
## Total      741.2963 412.4162 243.1708 131.7368 117.9416  99.9943  90.3451
##              PC8      PC9      PC10      PC11      PC12      PC13      PC14
## Fitted
## Residuals  64.5324  43.4411  42.9112  32.4887  29.6020  27.9724  26.3722
## Total      80.2716  46.9199  45.5122  42.9432  35.3433  34.7238  32.1156
##              PC15      PC16      PC17      PC18      PC19      PC20      PC21
## Fitted
## Residuals  24.6029  23.8238  21.4968  20.4037  18.8184  17.5014  16.1502
## Total      28.6321  27.6384  26.6384  22.9269  21.5609  20.5311  19.2592
##              PC22      PC23      PC24      PC25      PC26      PC27      PC28
## Fitted
## Residuals  15.6273  14.9691  13.2887  13.0737  12.8391  11.8027  11.5692
## Total      18.1625  17.3365  16.2670  15.2323  14.2357  13.7407  13.6344
##              PC29      PC30      PC31      PC32      PC33      PC34      PC35
## Fitted
## Residuals  10.4926  10.3096   9.4148   9.1406   8.4518   7.8754   7.7879
## Total      12.3507  11.8622  11.3577  10.9042  10.5546  10.1456   9.5823
##              PC36      PC37      PC38      PC39      PC40      PC41      PC42
## Fitted
## Residuals
```

```
## Total      9.2737   8.8000   8.6340   8.0915   7.7601   7.6078   6.5488
```

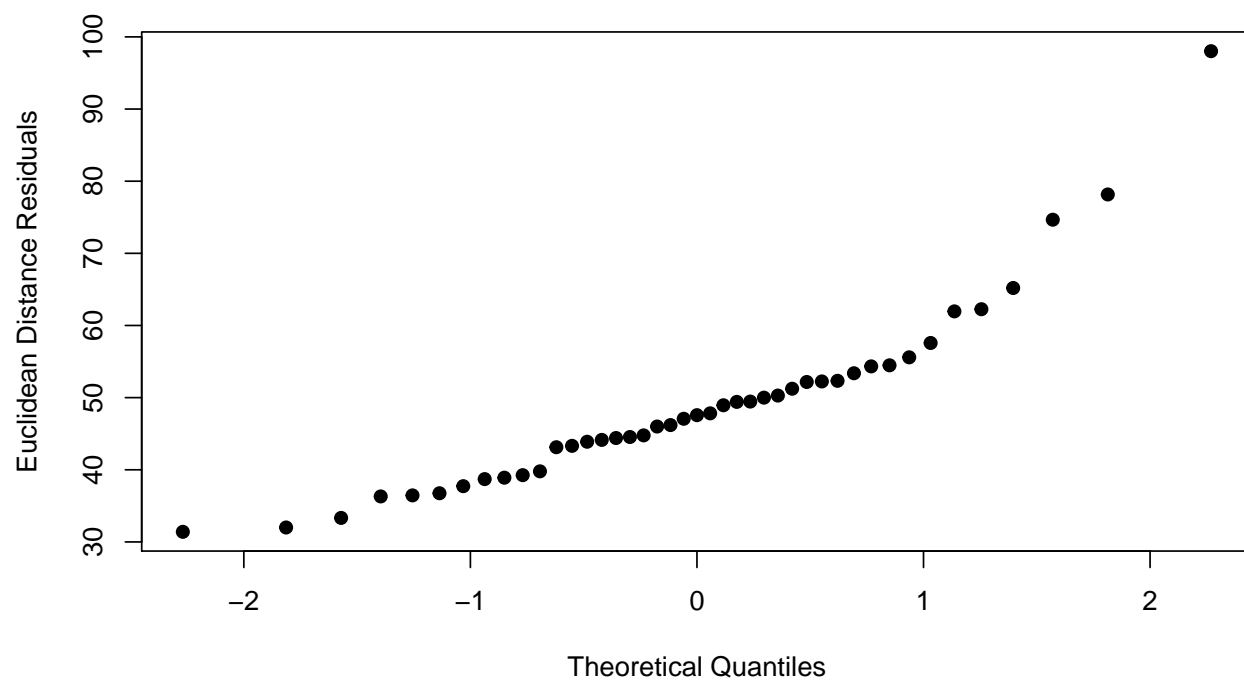
4. Examine RRPP plots to check for assumptions.

```
## Old Leaves residuals vs fitted values (homoscedasticity  
## check)
```

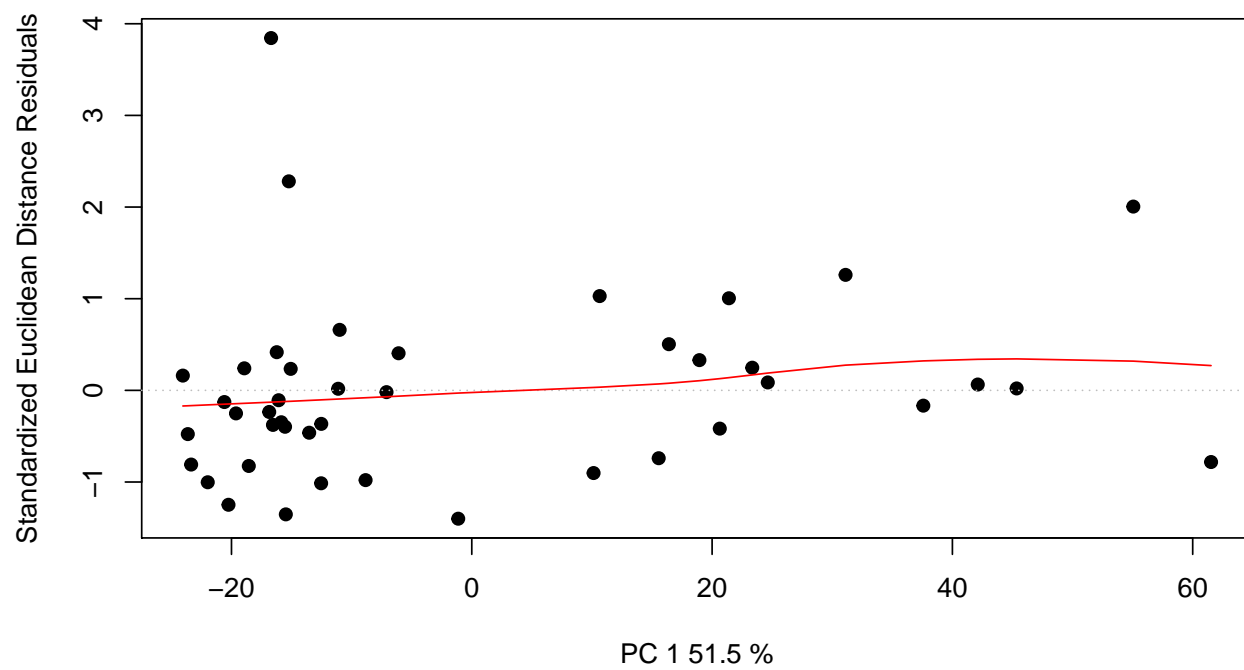
```
Odiagnostics <- plot(O_LMneg, type = "diagnostics")
```



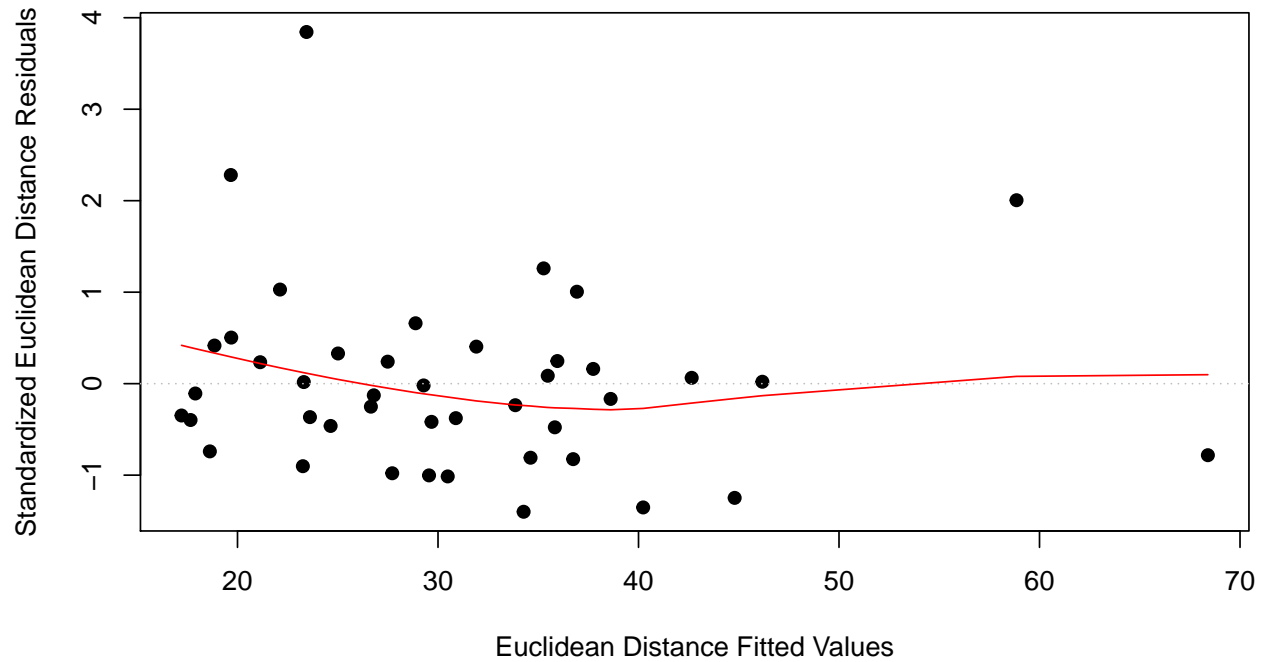
Q-Q plot



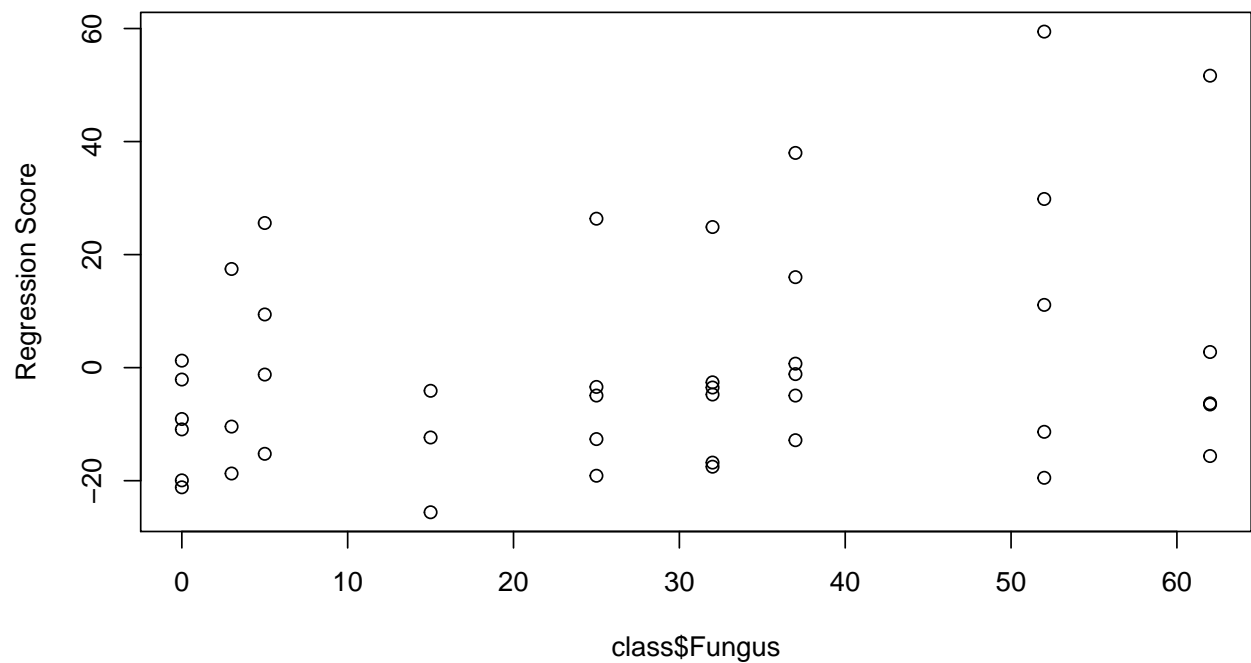
Residuals vs. PC 1 fitted



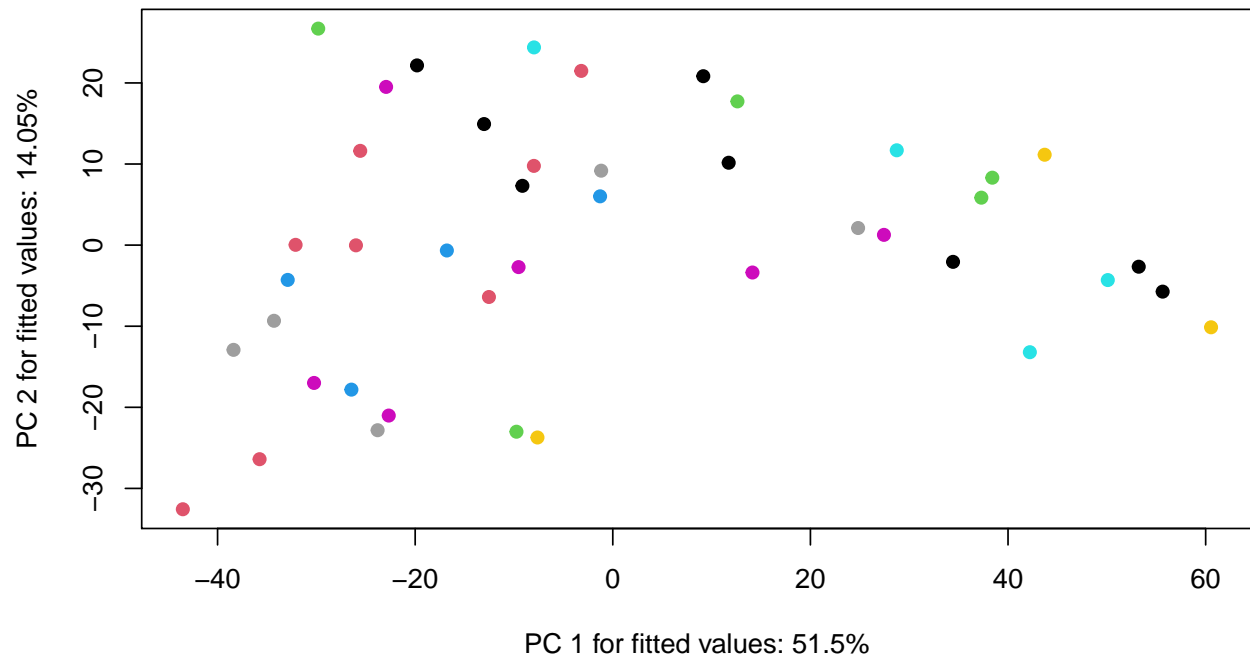
Residuals vs. Fitted



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

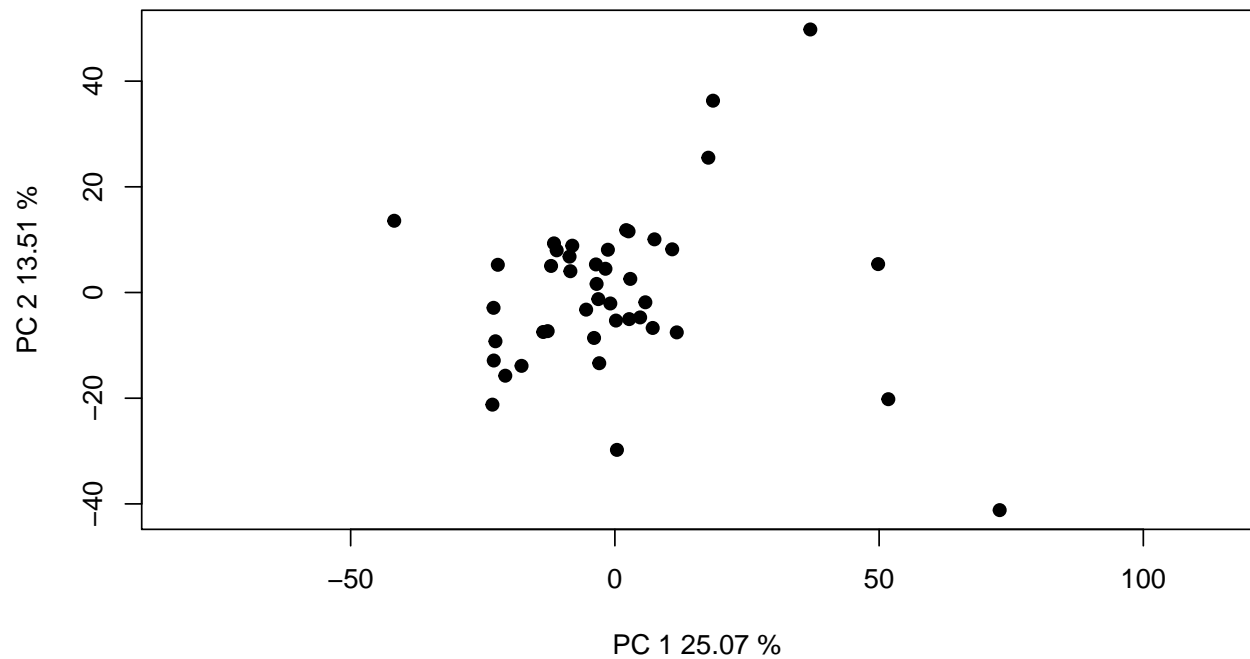


```
# pca plot
Opcplot <- plot(O_LMneg, type = "PC", pch = 19, col = interaction(class$Water,
  class$Fungus))
```

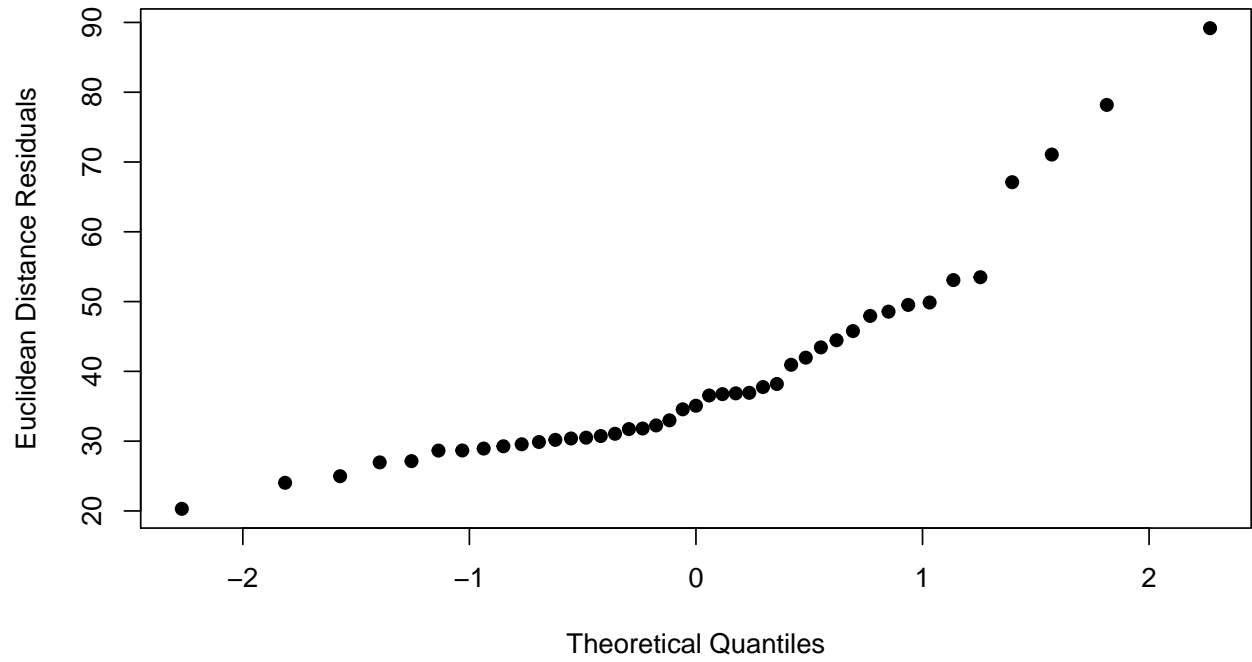


```
## Young Leaves residuals vs fitted values (homoscedasticity
## check)
Ydiagnostics <- plot(Y_LMneg, type = "diagnostics")
```

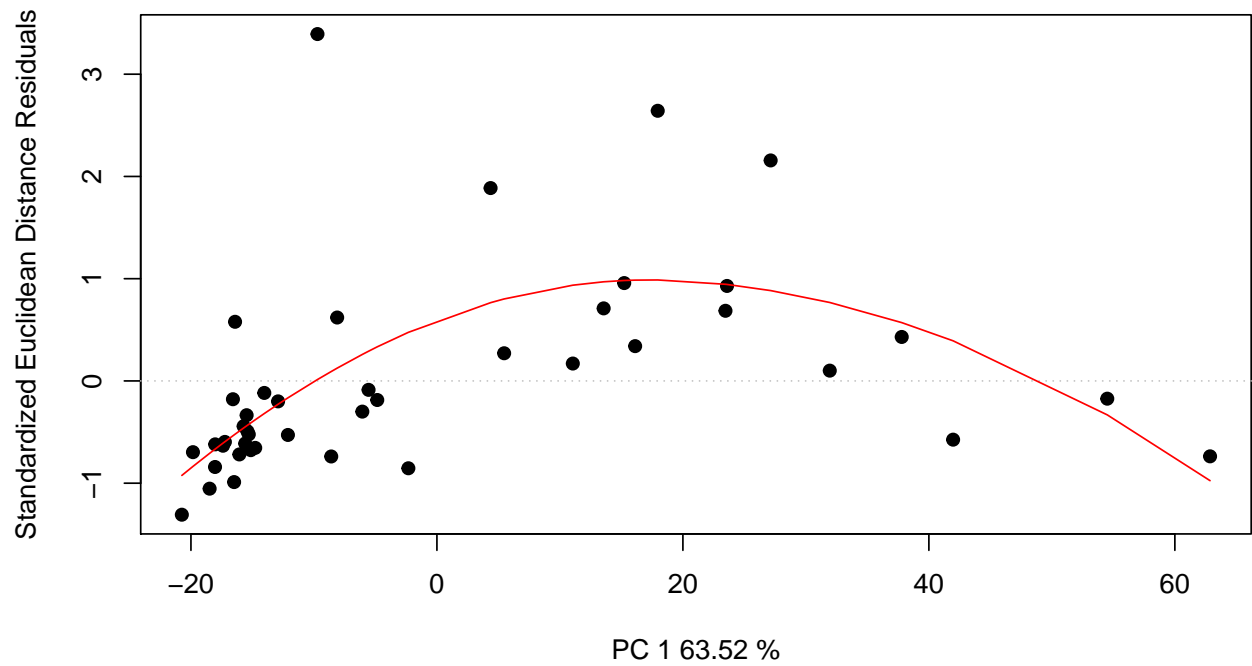
PCA Residuals



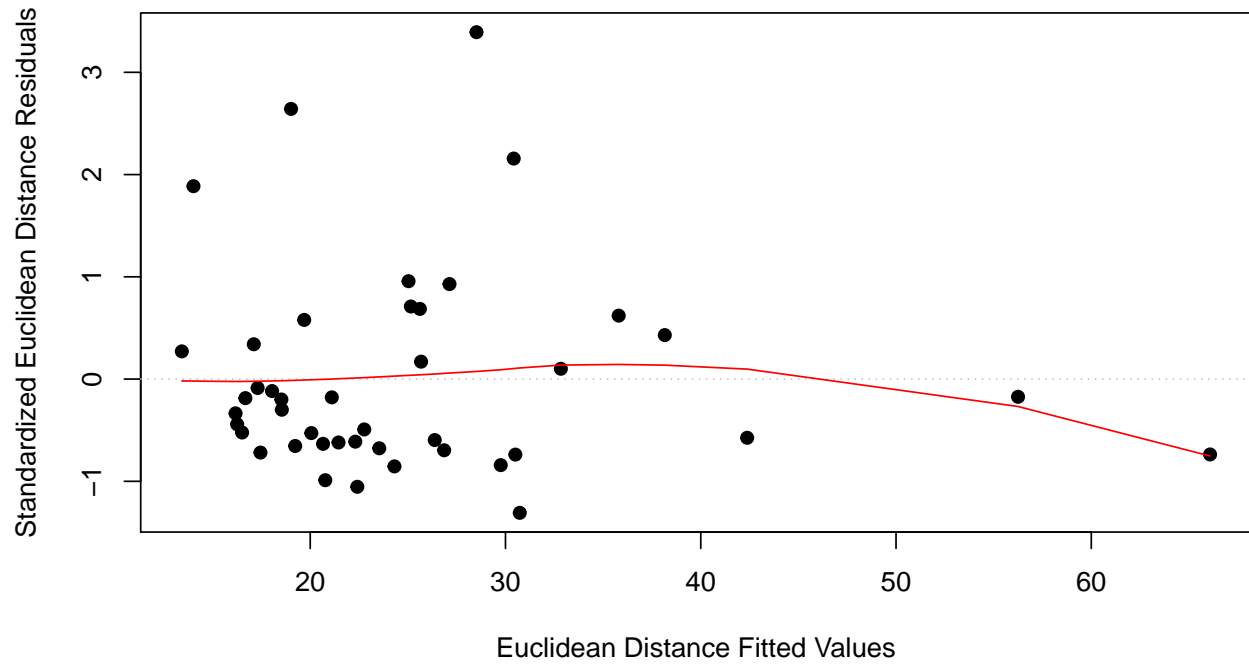
Q-Q plot



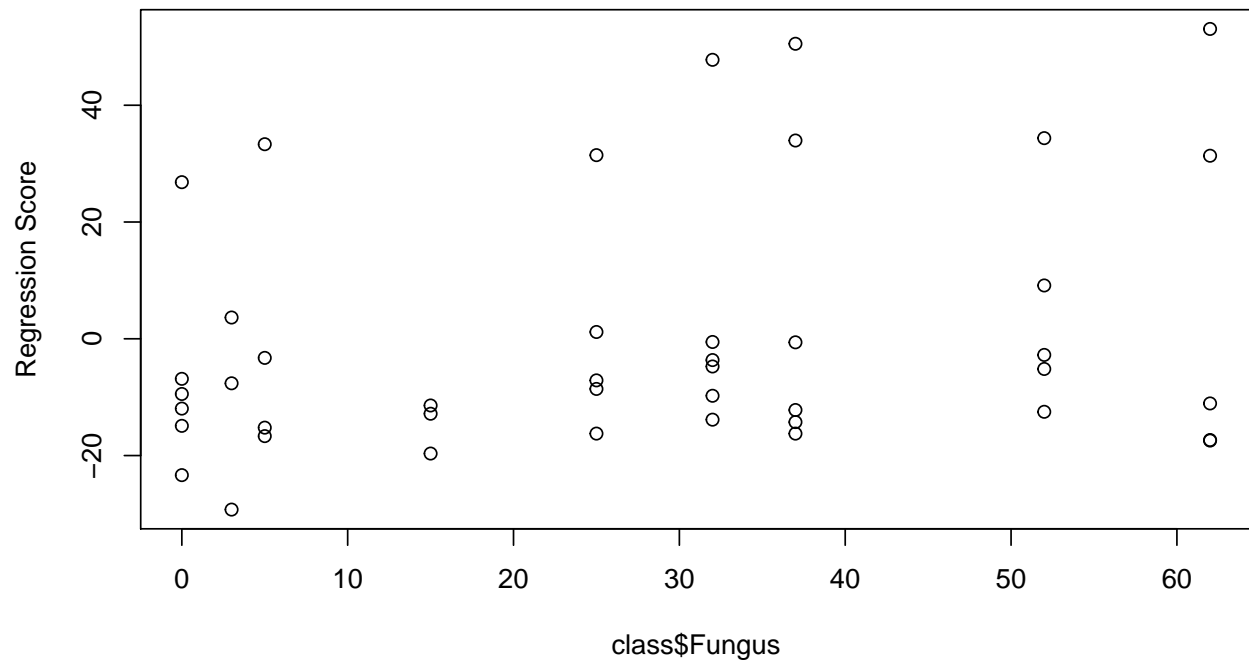
Residuals vs. PC 1 fitted



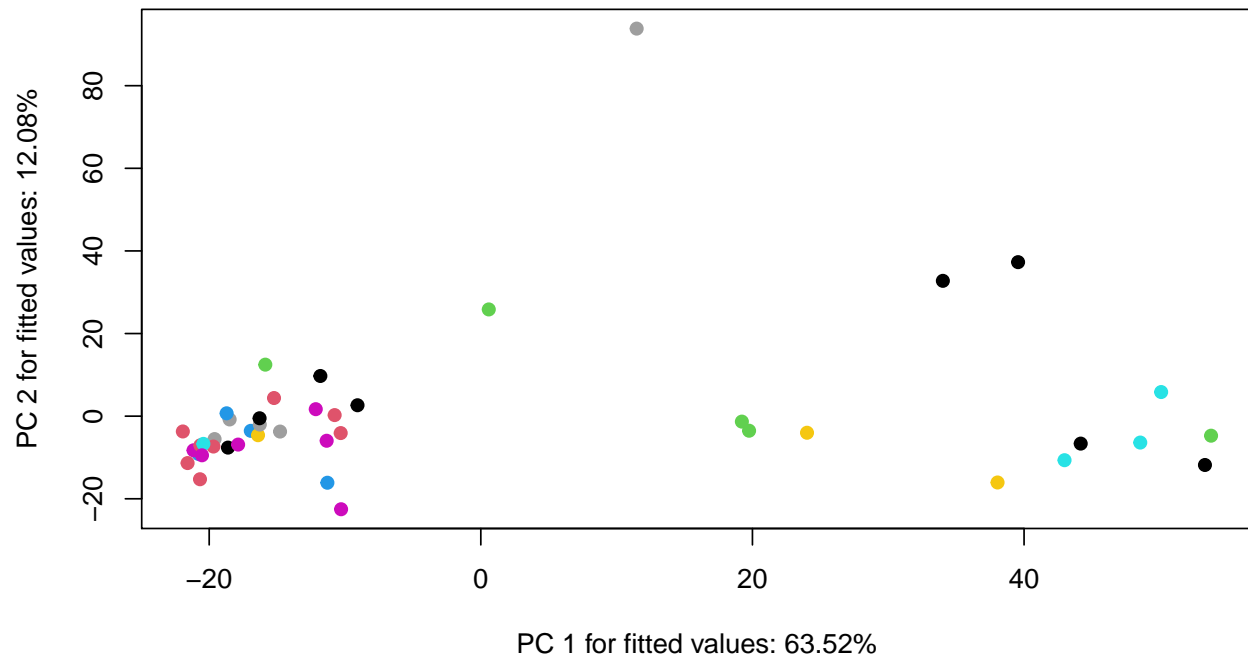
Residuals vs. Fitted



```
# 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    3202 3201.5 0.02041 1.0112 0.20301 0.370
## Water      1    5610 5610.4 0.03577 1.5380 2.09648 0.022 *
## Fungus     1    5175 5174.6 0.03300 1.4954 1.20511 0.113
## Block:Water 1    3648 3647.8 0.02326 1.1521 0.56563 0.263
## Block:Fungus 1    3692 3691.6 0.02354 1.0669 0.29363 0.407
## Water:Fungus 1    3935 3935.5 0.02509 1.1373 0.71081 0.238
## Block:Water:Fungus 1    3460 3460.3 0.02206 1.0929 0.35576 0.320
## Residuals   35 110818 3166.2 0.70662
## Total      42 156828
## ---
## 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",
```

```

"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   1885 1885.0 0.01750 0.8680 -0.11886 0.517
## Water           1   2056 2056.1 0.01909 1.5851  1.86954 0.030 *
## Fungus          1   4496 4496.5 0.04175 1.8526  1.55113 0.057 .
## Block:Water     1   1297 1297.1 0.01205 0.5973 -1.17211 0.889
## Block:Fungus    1   2939 2939.1 0.02729 1.2109  0.56704 0.277
## Water:Fungus    1   2993 2992.7 0.02779 1.2330  0.83351 0.206
## Block:Water:Fungus 1   2427 2427.1 0.02254 1.1177  0.43148 0.318
## Residuals      35  76007 2171.6 0.70580
## Total          42 107688
## ---
## 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
Onegcoef <- coef(O_LMneg, test = T)
summary(Onegcoef)

##
## Linear Model fit with lm.rrpp
##
## Number of observations: 43
## Number of dependent variables: 3734
## 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)  60.3518088 95.1576780 -1.6147590 0.979
## Block        26.9816761 32.6377501  0.5810541 0.215
## WaterLow     92.4671749 83.0478713  2.7401007 0.012
## Fungus       1.8892146  1.8291740  2.1574420 0.036
## Block:WaterLow 38.4181644 43.0241417  1.0338704 0.137
## Block:Fungus  0.8183189  0.9240055  0.9726995 0.151
## WaterLow:Fungus 2.2856809  2.4625126  1.2781698 0.108

```

```
## Block:WaterLow:Fungus 1.0601248 1.2260900 0.7919779 0.192
```

```
## Young Leaves test model coefficients
```

```
Ynegcoef <- coef(Y_LMneg, test = T)
```

```
summary(Ynegcoef)
```

```
##
```

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

```
##
```

```
## Number of observations: 43
```

```
## Number of dependent variables: 2564
```

```
## 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) | 44.4893714 | 77.035800 | -1.6998557 | 0.983 |
| ## Block | 20.7034191 | 28.851289 | 0.1383966 | 0.364 |
| ## WaterLow | 55.9771863 | 70.500843 | 0.3669193 | 0.299 |
| ## Fungus | 1.7610770 | 1.600200 | 2.6444701 | 0.018 |
| ## Block:WaterLow | 22.9093040 | 36.583231 | -0.7695439 | 0.776 |
| ## Block:Fungus | 0.7301684 | 0.819176 | 1.2218881 | 0.117 |
| ## WaterLow:Fungus | 1.9932055 | 2.146520 | 1.4119705 | 0.084 |
| ## Block:WaterLow:Fungus | 0.8878702 | 1.063849 | 0.7867375 | 0.187 |

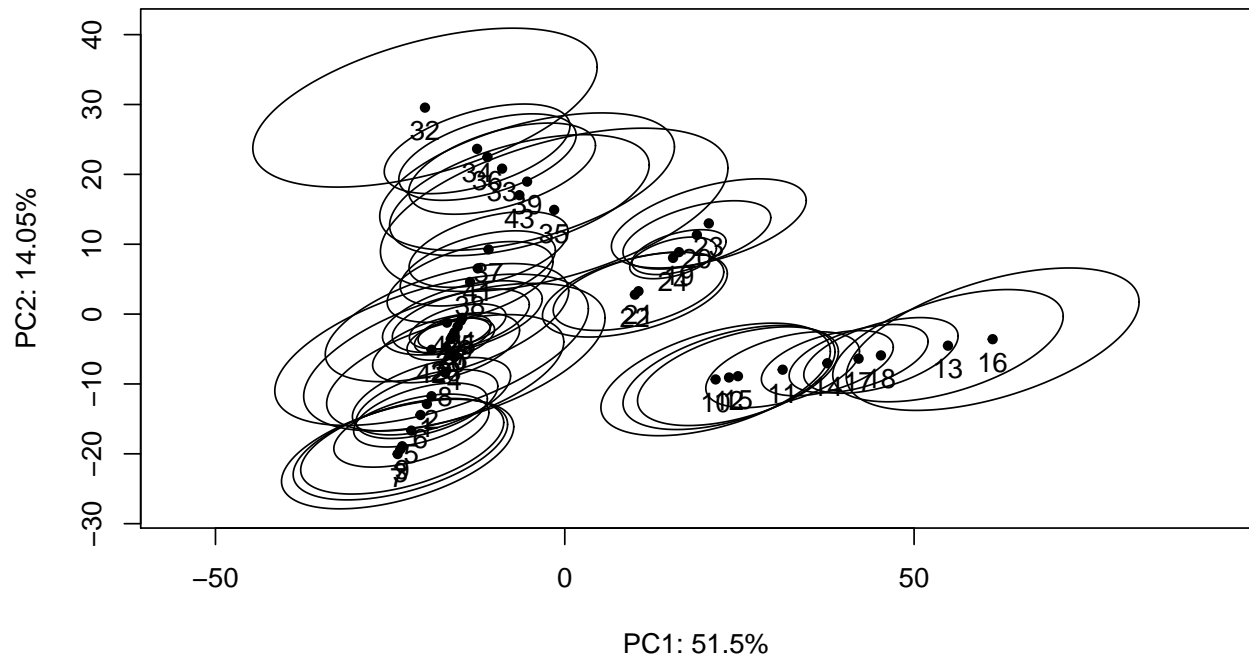
WaterLow has the largest effect on the model. The standard is the mean for High water treatment.

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

```
O_pred <- predict(O_LMneg)
```

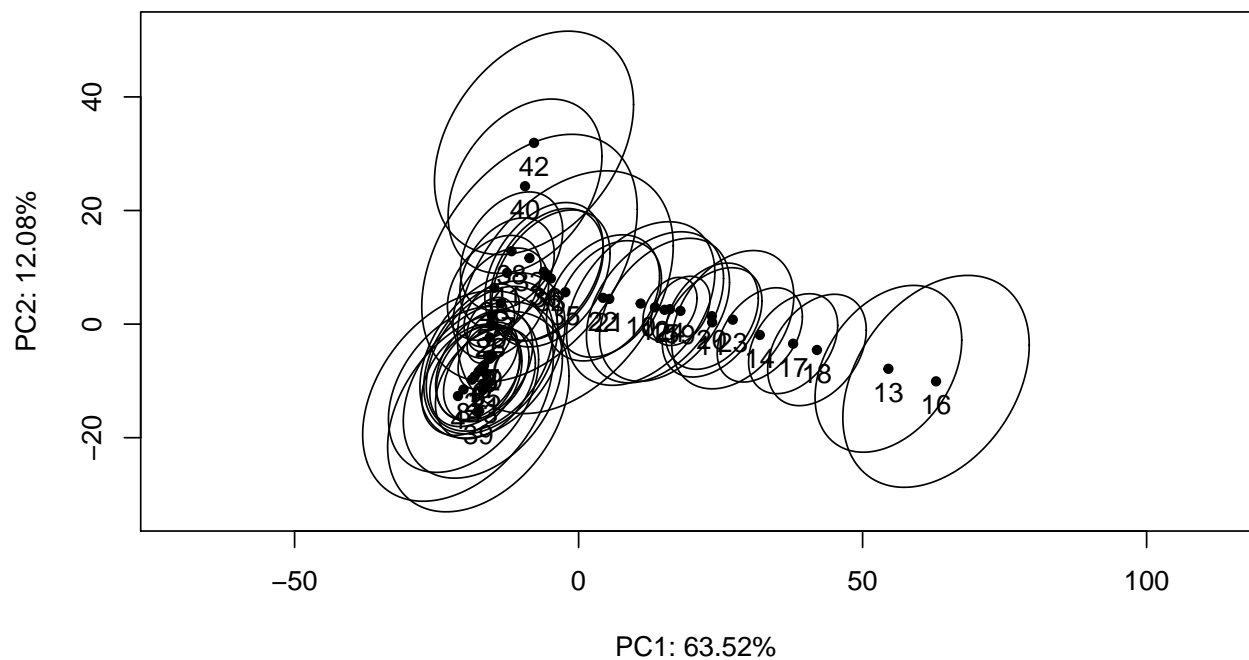
```
plot(O_pred, PC = T, ellipse = T)
```

Among-prediction PC rotation; 95% confidence limits



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

Among-prediction PC rotation; 95% confidence limits



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 water
Onegpw <- pairwise(O_LMneg, groups = class$Water)
summary(Onegpw, confidence = 0.95, stat.table = T)

##
## Pairwise comparisons
##
## Groups: High Low
##
## 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
## High:Low 38.34694 49.87041 -0.5777425 0.693

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

##
## Pairwise comparisons
##
## Groups: High Low
##
## 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
## High:Low 34.62897 44.12792 -0.5132088 0.689

Ynegpw2 <- pairwise(Y_LMneg, groups = class$Fungus)
summary(Ynegpw2, 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    11.967287 18.355618 -1.5437998 0.960
## 0:5     7.073755 13.489150 -1.6170467 0.992
## 0:15    8.463239 15.962885 -2.6648850 1.000
## 0:25    7.249913 15.317896 -1.9298912 0.999

```

```
## 0:32 10.797658 20.945008 -1.4658976 0.972
## 0:37 12.484792 24.217666 -1.4658976 0.972
## 0:52 19.500474 34.669590 -1.2158107 0.922
## 0:62 23.410947 43.584857 -1.3451742 0.937
## 3:5 14.149716 20.201283 -0.8557056 0.797
## 3:15 8.923562 14.450566 -1.4093037 0.950
## 3:25 12.647085 19.839114 -1.9958906 0.989
## 3:32 12.397093 20.875031 -2.1728750 0.997
## 3:37 13.281103 22.899146 -2.1188893 0.996
## 3:52 18.290922 31.825494 -1.6669728 0.987
## 3:62 20.897656 38.825164 -1.6497027 0.990
## 5:15 7.559166 11.178094 -0.9053605 0.821
## 5:25 10.045844 15.942290 -0.7283301 0.744
## 5:32 13.913808 21.451907 -0.5312014 0.661
## 5:37 15.397669 24.184723 -0.6010529 0.681
## 5:52 22.064981 33.955665 -0.4852377 0.637
## 5:62 25.678135 41.355658 -0.7593210 0.765
## 15:25 7.934798 12.773747 -1.7897362 0.989
## 15:32 9.834148 16.490910 -1.1818510 0.917
## 15:37 11.031471 18.671089 -1.1332872 0.902
## 15:52 16.772019 26.892873 -0.7745148 0.761
## 15:62 20.435333 35.006860 -1.0294169 0.855
## 25:32 4.796356 6.969520 -0.6801663 0.744
## 25:37 6.280954 9.987427 -0.8752117 0.803
## 25:52 13.445915 20.416946 -0.6367510 0.711
## 25:62 17.520881 29.313591 -1.0832902 0.868
## 32:37 1.687134 3.272658 -1.4658976 0.972
## 32:52 9.437433 14.848263 -0.8593867 0.813
## 32:62 12.949955 22.882762 -1.2418133 0.916
## 37:52 8.039163 12.128741 -0.7536081 0.764
## 37:62 11.372417 19.770866 -1.2092249 0.908
## 52:62 7.696395 15.320555 -1.5126342 0.979
```

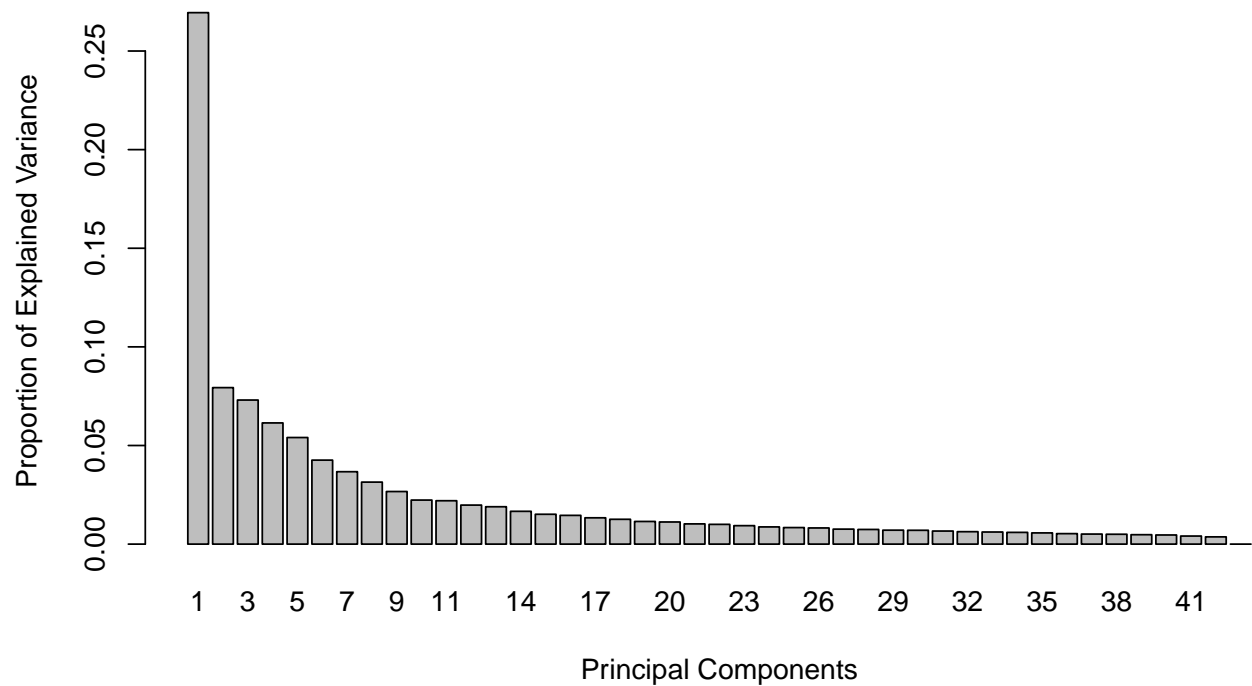
PCA

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

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

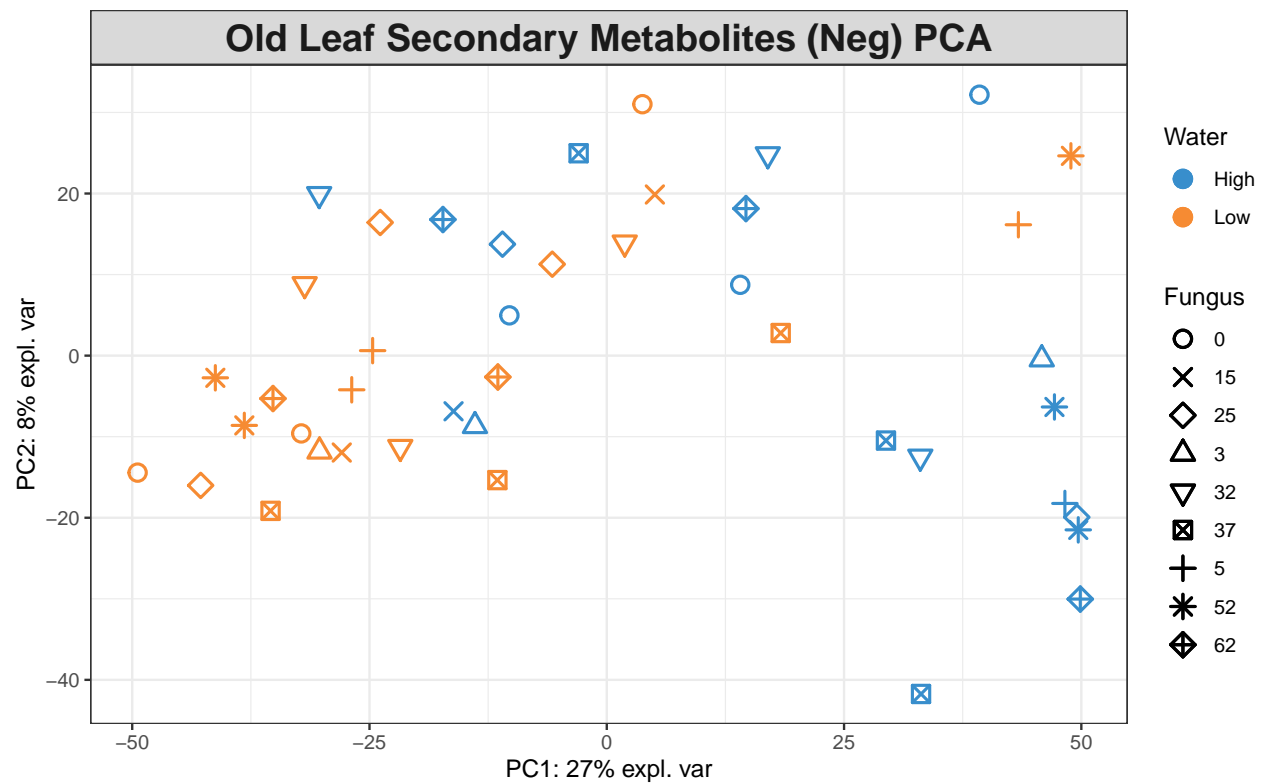
```
## Eigenvalues for the first 10 principal components, see object$sdev^2:
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7
## 1006.12346 296.26837 272.74967 229.43449 201.77791 159.01902 137.05643
##      PC8      PC9      PC10
## 117.36477 99.56505 83.39669
##
## Proportion of explained variance for the first 10 principal components, see object$explained_variance
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7
## 0.26944924 0.07934343 0.07304490 0.06144470 0.05403801 0.04258678 0.03670499
##      PC8      PC9      PC10
## 0.03143138 0.02666445 0.02233441
##
## Cumulative proportion explained variance for the first 10 principal components, see object$cum.var:
```

```
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8
## 0.2694492 0.3487927 0.4218376 0.4832823 0.5373203 0.5799070 0.6166120 0.6480434
##      PC9      PC10
## 0.6747079 0.6970423
##
## Other available components:
## -----
## loading vectors: see object$rotation
```



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

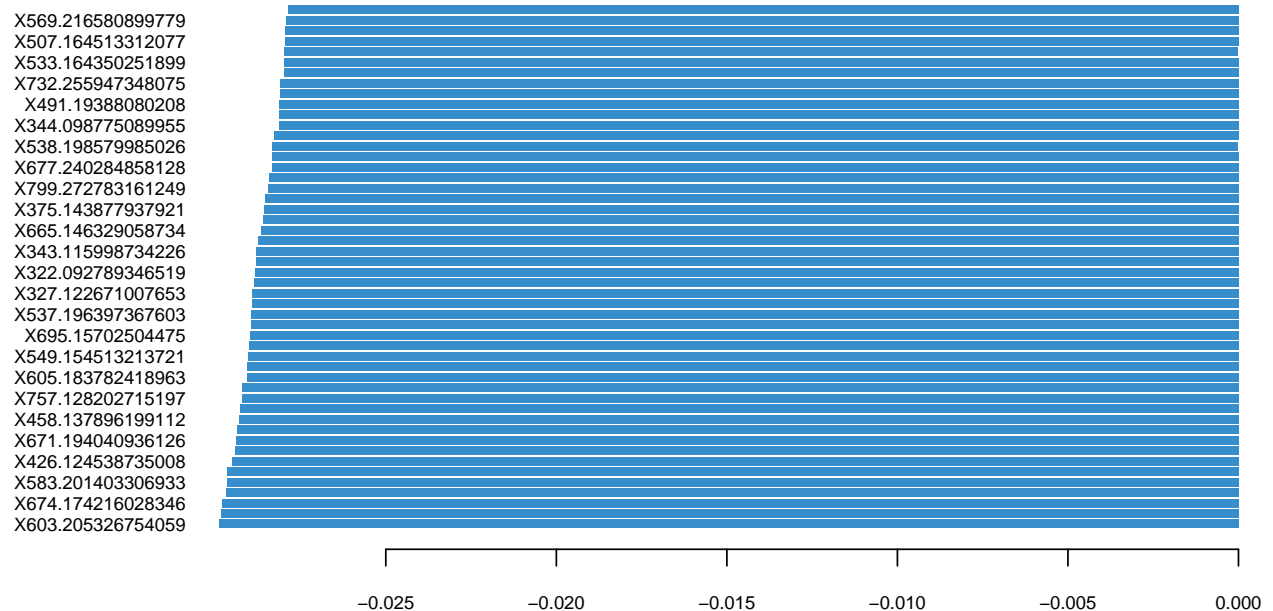
# plot pca
plotIndiv(pca.res, group = class$Water, ind.names = F, pch = as.factor(class$Fungus),
  legend = T, legend.title = "Water", legend.title.pch = "Fungus",
  title = "Old Leaf Secondary Metabolites (Neg) PCA")
```

*# 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*

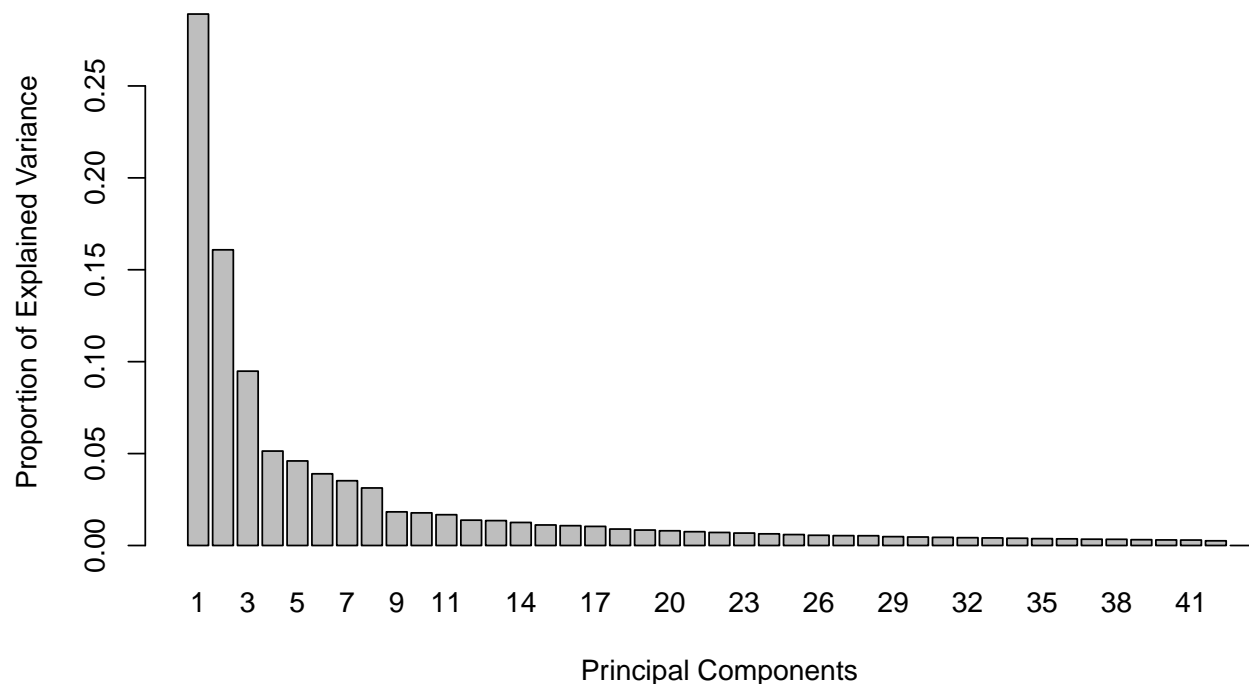
```
plotLoadings(pca.res, ndisplay = 50)
```

Loadings on comp 1



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

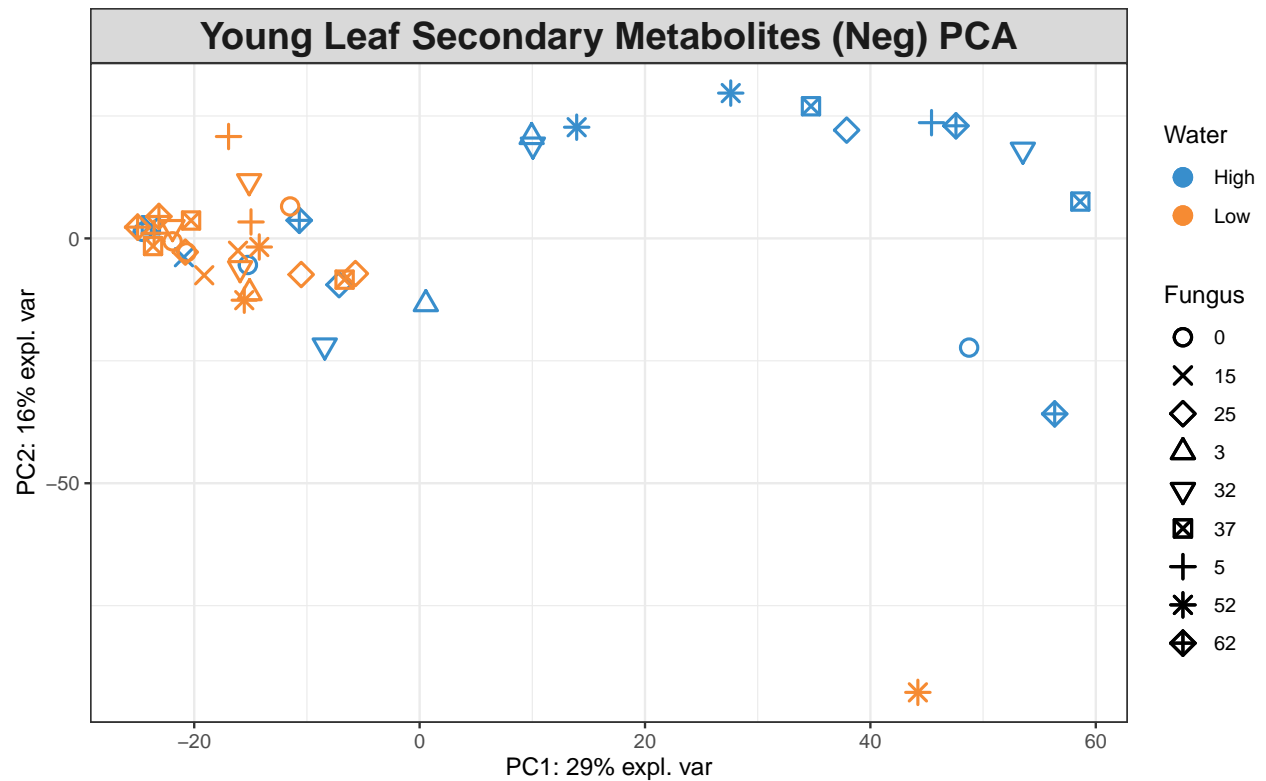
```
## Eigenvalues for the first 10 principal components, see object$sdev^2:
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8
## 741.29628 412.41616 243.17081 131.73675 117.94156  99.99428  90.34507  80.27160
##      PC9      PC10
##  46.91985  45.51224
##
## Proportion of explained variance for the first 10 principal components, see object$explained_variance:
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7
## 0.28911711 0.16084874 0.09484041 0.05137939 0.04599905 0.03899933 0.03523599
##      PC8      PC9      PC10
## 0.03130718 0.01829947 0.01775048
##
## Cumulative proportion explained variance for the first 10 principal components, see object$cum.var:
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8
## 0.2891171 0.4499658 0.5448063 0.5961856 0.6421847 0.6811840 0.7164200 0.7477272
##      PC9      PC10
## 0.7660267 0.7837771
##
## Other available components:
## -----
## loading vectors: see object$rotation
```



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

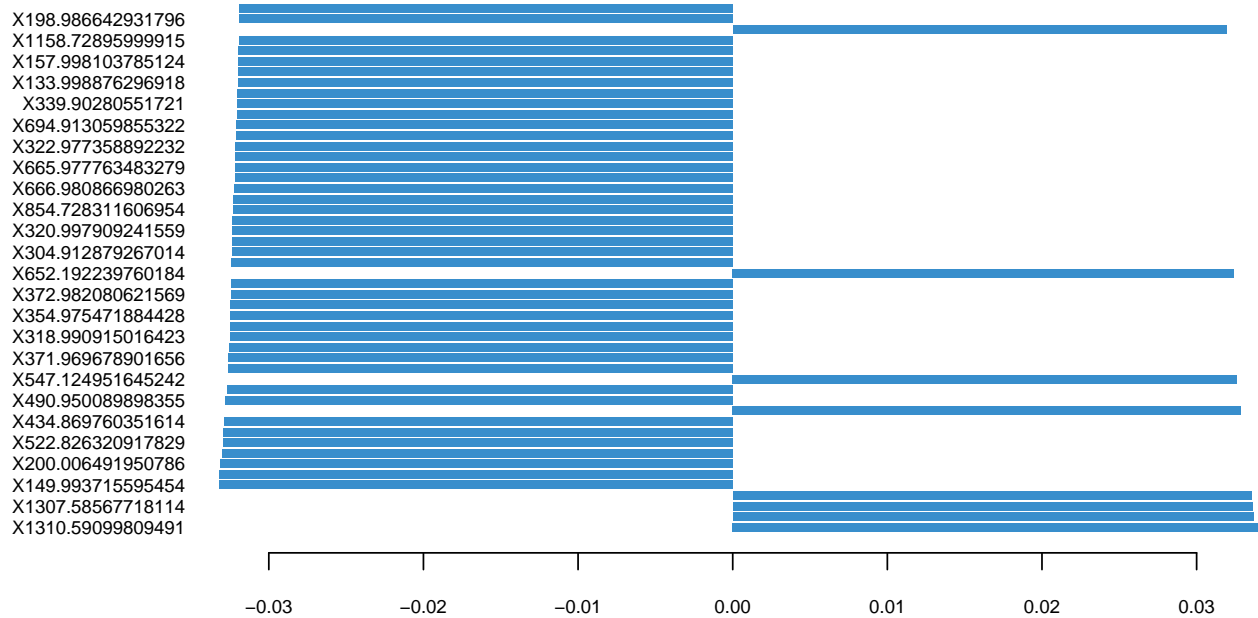
# plot pca
plotIndiv(pca.res, group = class$Water, ind.names = F, pch = as.factor(class$Fungus),
  legend = T, legend.title = "Water", legend.title.pch = "Fungus",
```

```
title = "Young Leaf Secondary Metabolites (Neg) PCA")
```



```
# 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)
```

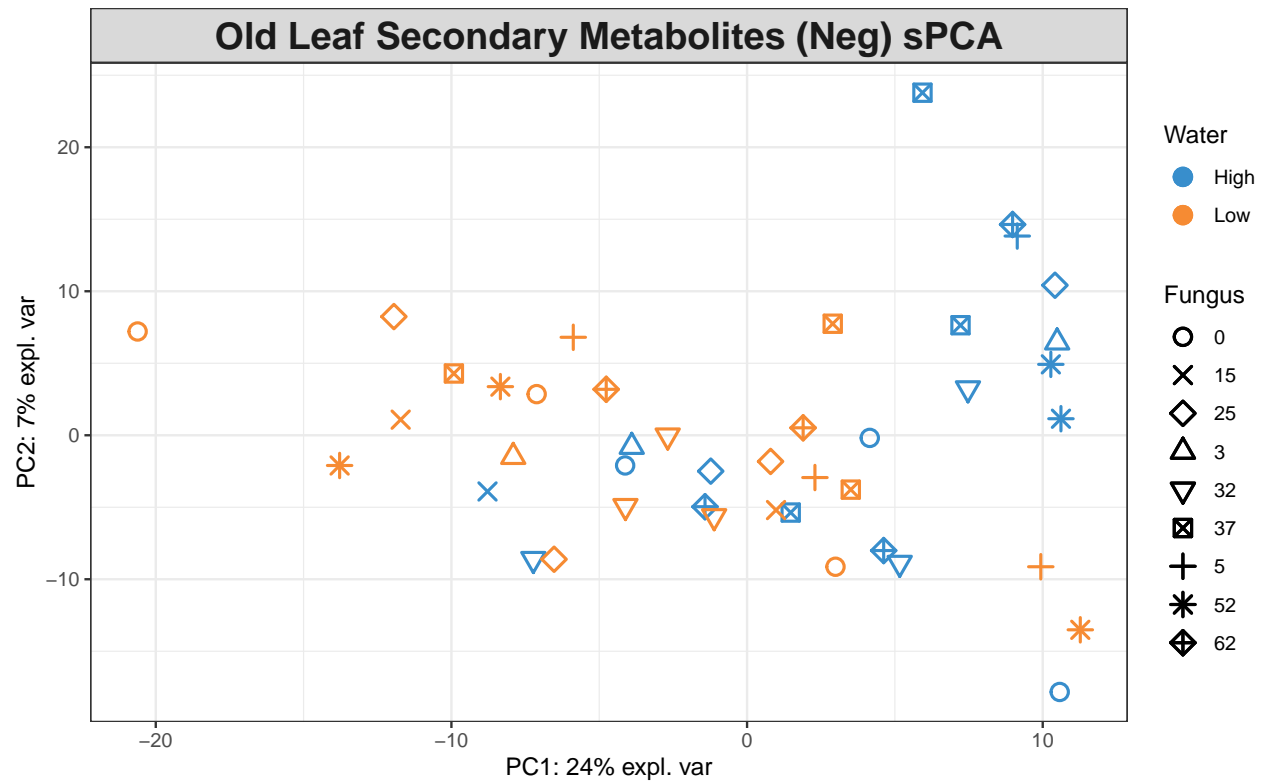
Loadings on comp 1



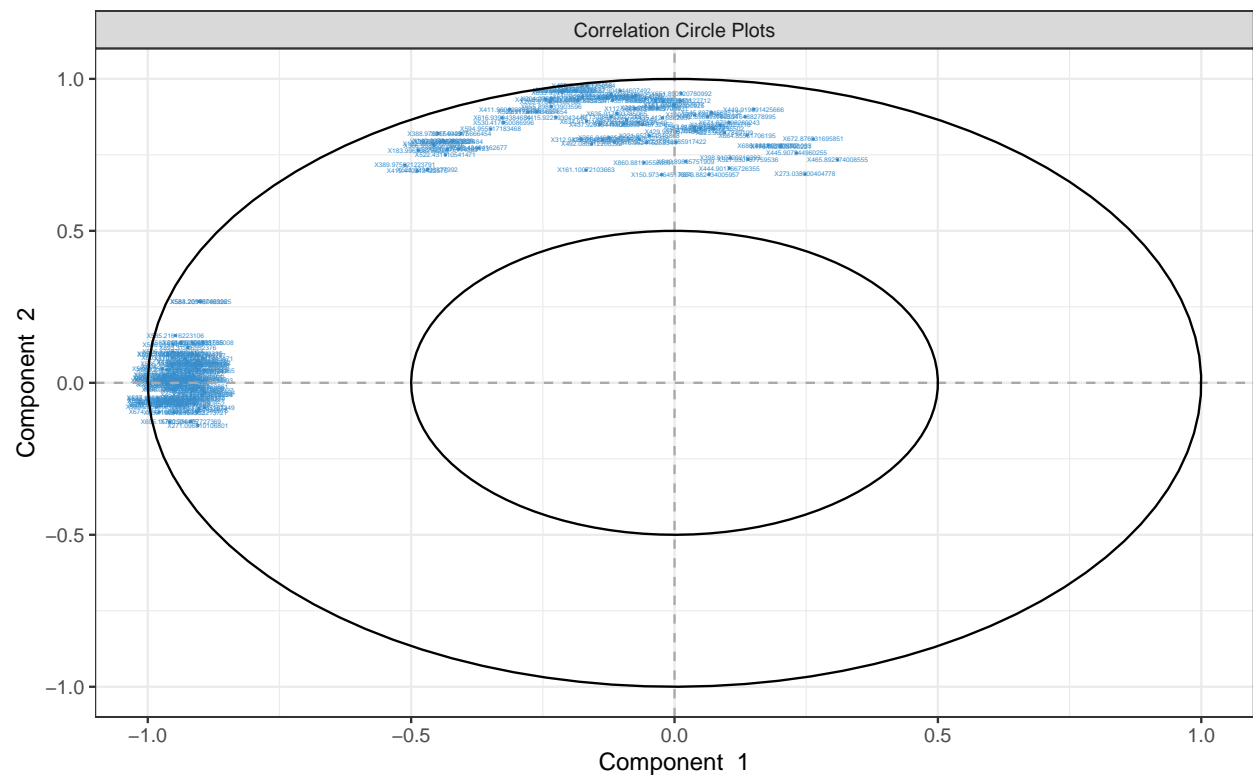
9. Identify the variables that contribute to the explained variance.

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

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



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



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

```
##          value.var
## X539.200396017582 -0.181082951
## X537.196397367603 -0.177118557
## X689.139568989934 -0.176379501
## X605.183782418963 -0.174908432
## X538.199827871219 -0.174730140
## X606.187523014034 -0.173086641
## X673.171512158096 -0.170842206
## X674.174216028346 -0.170227465
## X757.128202715197 -0.164840111
## X673.170424561307 -0.162356814
## X566.195673145549 -0.158134925
## X606.186978663872 -0.155743982
## X538.199494433559 -0.154379050
## X508.189553957924 -0.152111850
## X605.183549351429 -0.150829867
## X537.196150052541 -0.148851518
## X507.186328234882 -0.148529026
## X605.183770693646 -0.146771927
## X674.174026648318 -0.144312231
## X569.215412674731 -0.141371483
## X345.134077182166 -0.140059056
## X565.192217935211 -0.139170771
## X621.154757691745 -0.138209140
## X535.180816690394 -0.137719228
## X375.143877937921 -0.136720111
## X695.15702504475  -0.132672803
## X536.183781917591 -0.127150509
## X667.156131811396 -0.126185948
## X603.170193450933 -0.123359380
## X604.173915958927 -0.123282579
## X896.338024188277 -0.122276357
## X586.219348942499 -0.120032704
## X506.173570612014 -0.119838045
## X537.195007794578 -0.117466739
## X719.284172253939 -0.116608140
## X665.146329058734 -0.115605747
## X606.186970859362 -0.115319414
## X603.205326754059 -0.114899908
## X668.159851943747 -0.111049129
## X585.21616223106  -0.110936526
## X895.335059889362 -0.106853883
## X343.115998734226 -0.101118943
## X505.170366397634 -0.100061246
## X567.207353749129 -0.097753184
## X538.198579985026 -0.097218515
## X327.122671007653 -0.095158658
## X568.210597694665 -0.094806479
## X604.208532818709 -0.094428877
## X583.201403306933 -0.090439718
## X610.182262413814 -0.076973370
## X346.140373954959 -0.069376435
```

```

## X570.219543443364 -0.067921138
## X149.059073075148 -0.065194016
## X893.31996552376 -0.065091823
## X833.181144939018 -0.062452460
## X727.178420224907 -0.058258959
## X671.194040936126 -0.054279193
## X720.286487727369 -0.053767576
## X457.134224463719 -0.052892192
## X878.329634092335 -0.049771260
## X731.252742703494 -0.048911141
## X735.146845005385 -0.048670427
## X540.215185799472 -0.047788147
## X803.137206404634 -0.046133144
## X613.210520581568 -0.045375543
## X653.178879325056 -0.042429292
## X507.164513312077 -0.042115852
## X508.174573318256 -0.042074589
## X817.214095360852 -0.040853774
## X458.137896199112 -0.040246045
## X481.170644965179 -0.039170292
## X877.32639228893 -0.037250992
## X677.166894727587 -0.036928503
## X716.259282733113 -0.032230304
## X549.154513213721 -0.031683394
## X879.340062720795 -0.030671329
## X376.132023520547 -0.030608230
## X894.32300346217 -0.030321790
## X527.175065407586 -0.028133102
## X732.255947348075 -0.027703876
## X271.096810106801 -0.027347261
## X528.177807869861 -0.027319604
## X521.201790047362 -0.026797953
## X589.18940461419 -0.025996732
## X516.148124468326 -0.025001523
## X584.20513749326 -0.023886833
## X717.21276012974 -0.023767090
## X698.250989243722 -0.018573314
## X515.14524575838 -0.016156106
## X583.201760969925 -0.015767460
## X880.34365929126 -0.014726899
## X796.233580680884 -0.012552486
## X439.157924608228 -0.010533966
## X489.162112886893 -0.010227638
## X359.148741949471 -0.009070205
## X426.124538735008 -0.007881174
## X795.230947111927 -0.007269924
## X799.272783161249 -0.003948915
## X330.141938124165 -0.003220751
## X993.322387841374 -0.002683019

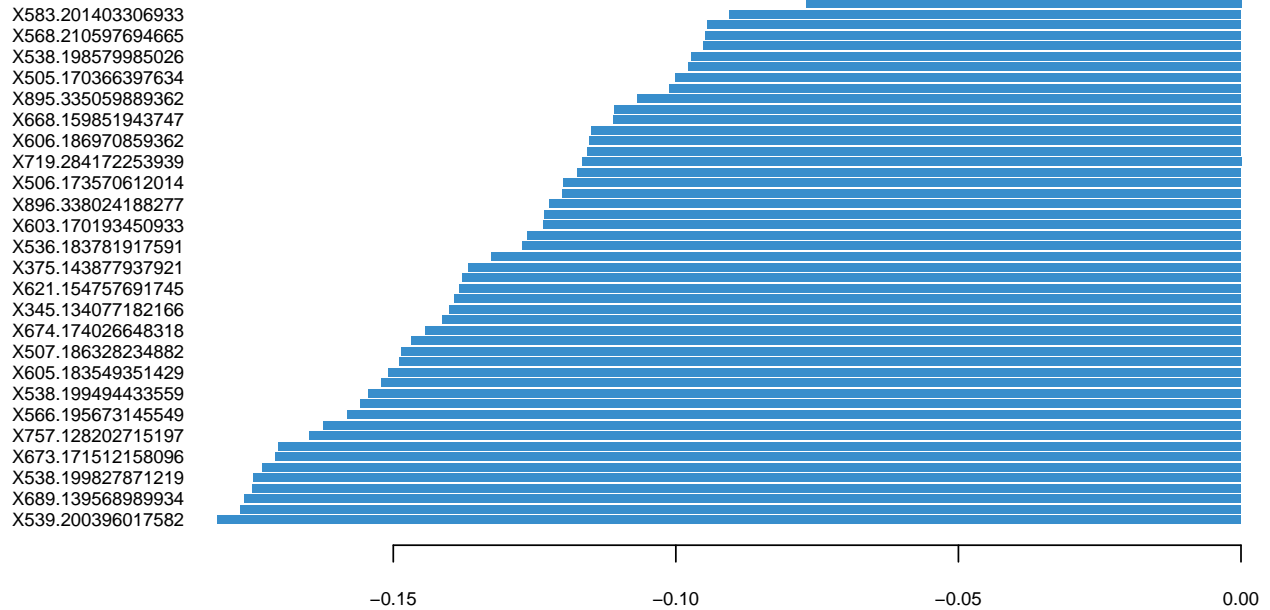
```

```

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

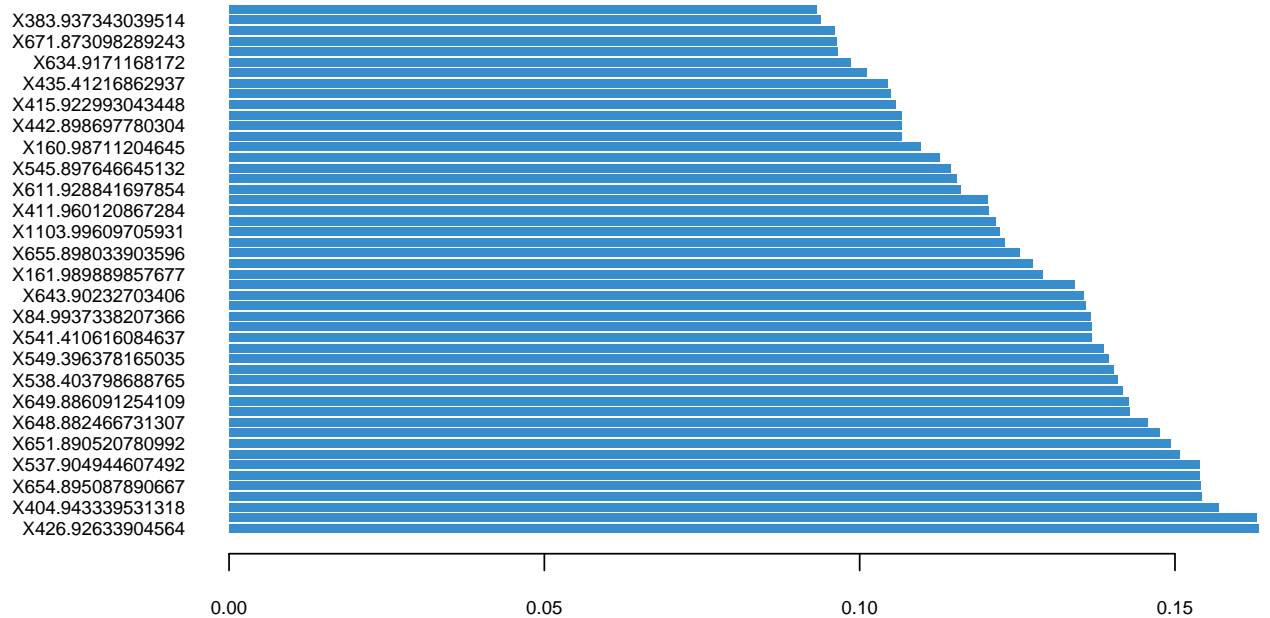
```

Loadings on comp 1



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

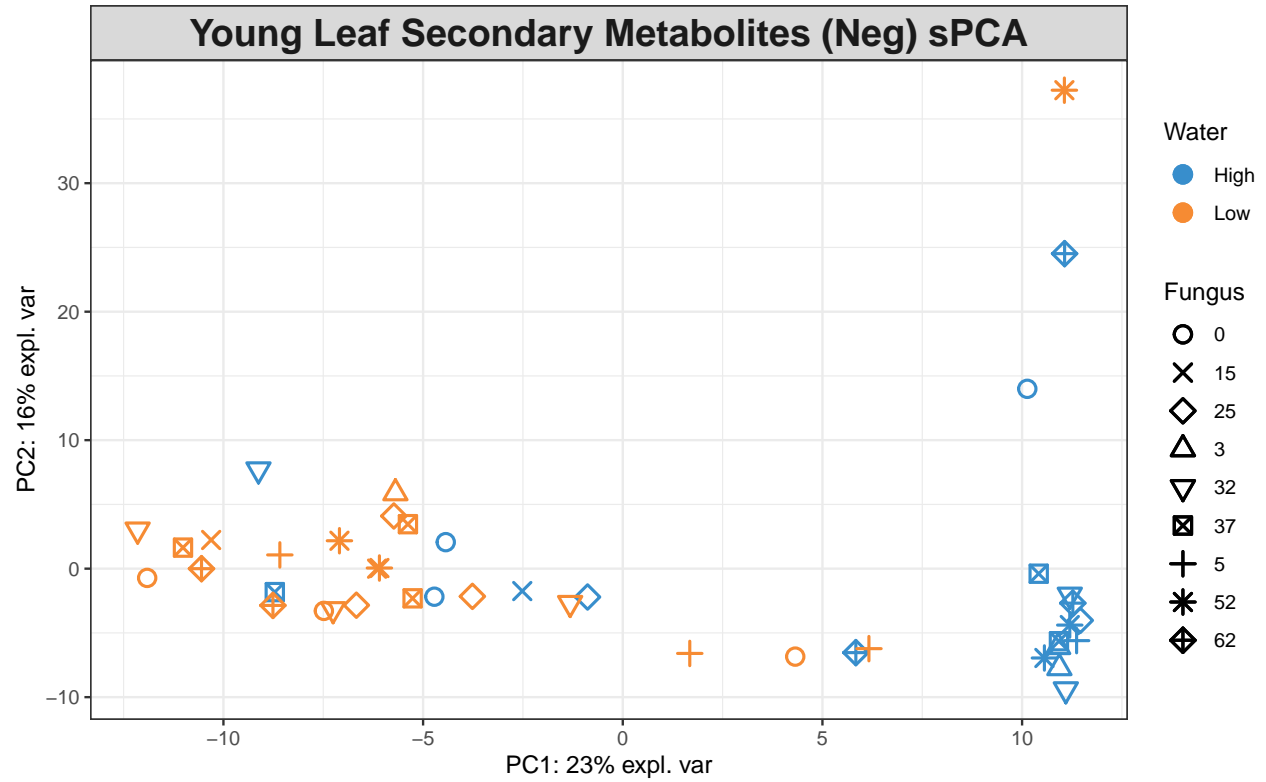
Loadings on comp 2



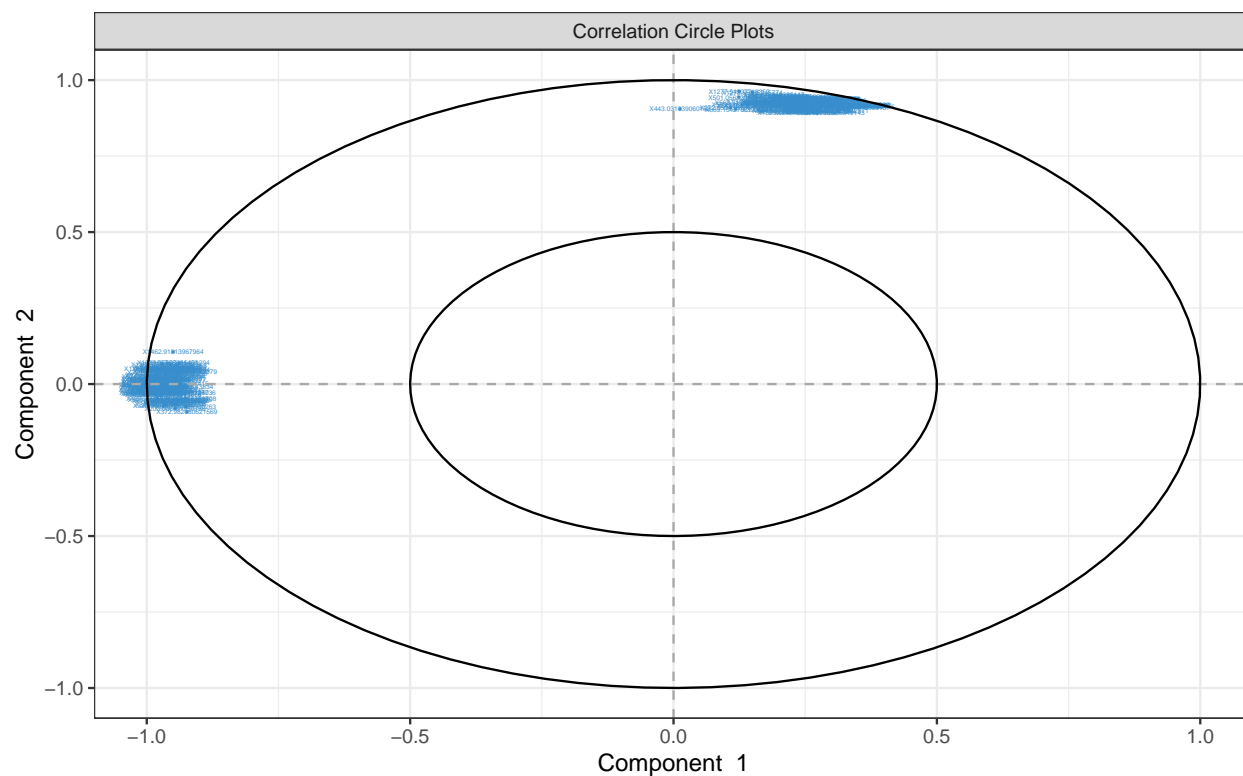
```
# Young Leaf
spca.res <- mixOmics::spca(scaled_Y_young, ncomp = 3, keepX = c(100,
100, 10))
```



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



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



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

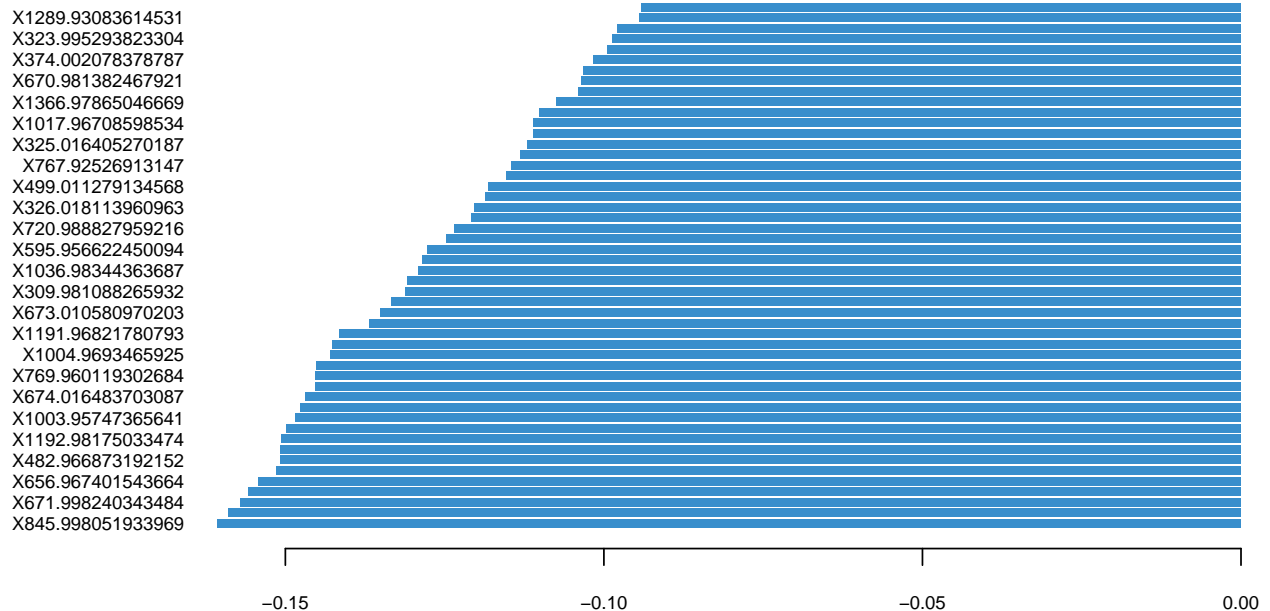
```
##          value.var
## X845.998051933969 -0.1606330732
## X657.982854820673 -0.1588913910
## X671.998240343484 -0.1570044653
## X1018.98243765579 -0.1558509302
## X656.967401543664 -0.1542198021
## X847.009461460586 -0.1513489387
## X482.966873192152 -0.1507840595
## X768.942262814837 -0.1507794924
## X1192.98175033474 -0.1505850124
## X942.945320263561 -0.1498940009
## X1003.95747365641 -0.1483768457
## X1019.99560196699 -0.1475759336
## X674.016483703087 -0.1468869753
## X1115.92973111084 -0.1452322948
## X769.960119302684 -0.1452289240
## X941.927736931021 -0.1451711041
## X1004.9693465925 -0.1429543394
## X844.982584773594 -0.1425938797
## X1191.96821780793 -0.1414755075
## X1193.99113116688 -0.1367830091
## X673.010580970203 -0.1350161100
## X546.988591117921 -0.1333992604
## X309.981088265932 -0.1311783631
## X1365.96770825737 -0.1308576944
## X1036.98344363687 -0.1291475627
## X497.995790827568 -0.1284198692
```

X595.956622450094 -0.1276729024
X421.955540488549 -0.1247260506
X720.988827959216 -0.1233755750
X654.985813378408 -0.1208239906
X326.018113960963 -0.1202548234
X893.974082323864 -0.1185839446
X499.011279134568 -0.1180404580
X831.984075771083 -0.1152274812
X767.92526913147 -0.1144326983
X827.971822574841 -0.1130674947
X325.016405270187 -0.1120384107
X658.999544129286 -0.1111185579
X1017.96708598534 -0.1110301332
X632.970118158626 -0.1101401694
X1366.97865046669 -0.1073823625
X1116.94487167345 -0.1039494146
X670.981382467921 -0.1035217899
X500.018284225597 -0.1032308499
X374.002078378787 -0.1015634926
X1021.00089889269 -0.0993725768
X323.995293823304 -0.0987291486
X283.006624782518 -0.0979302194
X1289.93083614531 -0.0944013782
X322.011417255538 -0.0940995915
X1002.95537959577 -0.0924342460
X943.961229494259 -0.0915650916
X320.997909241559 -0.0907305440
X849.977486523324 -0.0900179868
X548.002503827387 -0.0864849042
X1364.95342014471 -0.0848014939
X667.985410425584 -0.0825819686
X1539.96588681403 -0.0818807443
X439.973315198955 -0.0747832594
X1177.95950779409 -0.0705672962
X1537.94035488602 -0.0702813560
X1176.95174154844 -0.0693124850
X311.000291548275 -0.0673917656
X1190.95179773088 -0.0660990078
X297.965942084671 -0.0659084293
X151.01222317911 -0.0641287500
X828.975599111753 -0.0627652369
X1462.91813967964 -0.0582347094
X492.989329994173 -0.0568171097
X655.964796055963 -0.0556936893
X1209.9715082933 -0.0554816541
X1001.96773728958 -0.0511680114
X496.976355787476 -0.0504382102
X843.965940610961 -0.0496897756
X200.006491950786 -0.0493434127
X485.000453042681 -0.0492835816
X1005.98287863099 -0.0484741503
X894.987582775265 -0.0484167703
X1288.91771445373 -0.0466471937
X892.958412895177 -0.0465485025

```
## X840.974050128923 -0.0462979834
## X675.97602596745 -0.0462678146
## X493.984197587741 -0.0442490380
## X832.995567847355 -0.0441367913
## X1066.96015657453 -0.0439418244
## X483.981165431415 -0.0373671386
## X862.985775931247 -0.0364052212
## X653.970909319968 -0.0362206773
## X839.970909295556 -0.0334228170
## X1067.97416491294 -0.0331244928
## X494.99544329672 -0.0317928369
## X1538.95350383304 -0.0308769093
## X1114.9157769402 -0.0284955747
## X1016.95098645776 -0.0219369799
## X506.971749443834 -0.0166512968
## X676.816971205036 -0.0063743103
## X666.980866980263 -0.0054246355
## X308.001278383708 -0.0041169721
## X1187.95480089979 -0.0008532640
## X372.982080621569 -0.0001506316
```

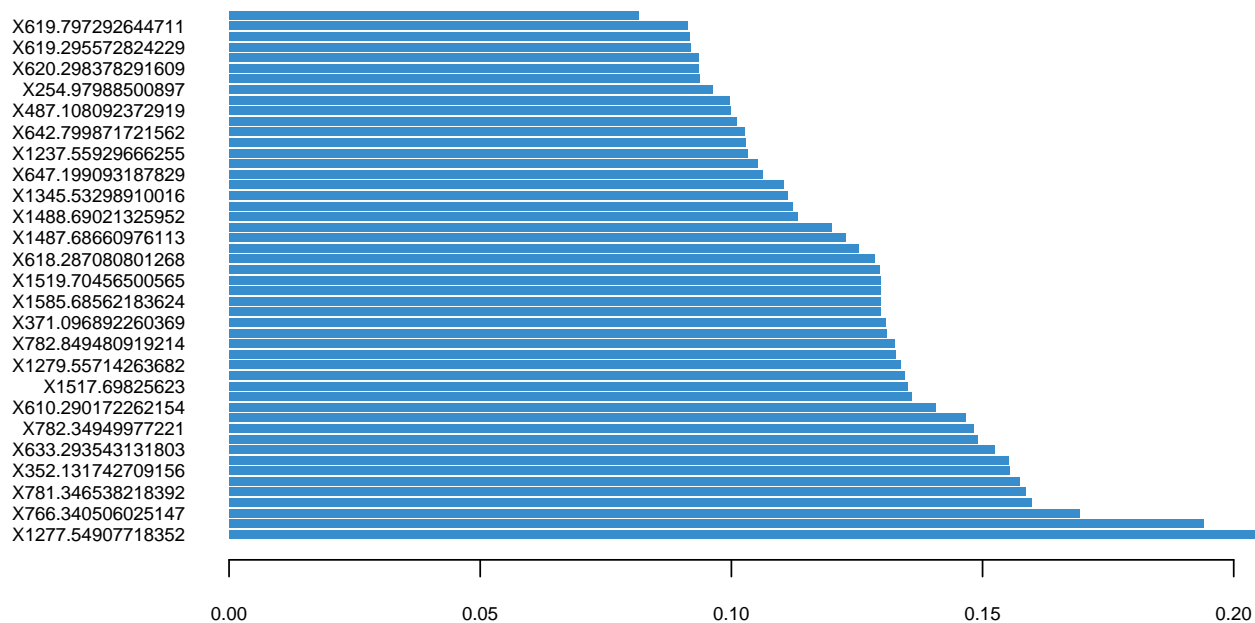
```
# plot loadings for comp 1
plotLoadings(spca.res, ndisplay = 50)
```

Loadings on comp 1



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

Loadings on comp 2



PLS-DA

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

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

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



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

```
## $name
## [1] "X360.100855094492" "X689.406786965204" "X313.092613575399"
## [4] "X325.111894653303" "X185.190588615838" "X324.137371517562"
## [7] "X323.133879784681" "X690.410098388308" "X389.160740120785"
## [10] "X691.422542368164" "X319.157448204942" "X367.123501601156"
## [13] "X321.173284297382" "X555.093874891555" "X391.122971286377"
## [16] "X944.449967310942" "X317.074072751279" "X322.17642106672"
## [19] "X1240.15789775369" "X639.046287279312" "X183.175015349613"
## [22] "X351.128821579971" "X169.989509429336" "X556.097536191383"
## [25] "X420.119920756105" "X1242.17308762521" "X365.145723604088"
## [28] "X232.95540138555" "X373.115564667084" "X503.071140727181"
## [31] "X564.931775721697" "X623.08265073133" "X943.448842600956"
## [34] "X419.116546770163" "X571.058061473006" "X1058.40882060084"
## [37] "X671.194040936126" "X344.098775089955" "X352.132376994544"
## [40] "X483.86951634573" "X351.129035832649" "X324.173145983115"
## [43] "X459.11029979044" "X1178.17585807417" "X487.104967151007"
## [46] "X578.160197428851" "X320.160371151912" "X359.097883259062"
## [49] "X349.115949924823" "X431.154386005126" "X604.208532818709"
## [52] "X340.075145186925" "X779.244559972528" "X537.082677183086"
## [55] "X481.088866272944" "X435.08241905927" "X1243.17676615566"
## [58] "X707.029978476026" "X488.109293490248" "X461.06951805573"
## [61] "X380.158184370181" "X561.03604963324" "X737.232626061852"
## [64] "X403.160178776661" "X648.205609857975" "X402.11215295727"
## [67] "X483.119144506346" "X341.965577057768" "X531.066415927912"
## [70] "X727.164615818962" "X379.328828228171" "X253.107386469566"
## [73] "X582.225274749511" "X538.198579985026" "X509.112370096629"
## [76] "X797.114458482459" "X170.985886728112" "X339.118846056087"
## [79] "X605.192195695578" "X581.222187754695" "X317.075176414668"
## [82] "X480.872661584228" "X191.106884410638" "X337.077569276891"
## [85] "X813.080943994947" "X274.831245724891" "X1244.17196607701"
## [88] "X125.083041728111" "X403.124191936883" "X745.249207080709"
## [91] "X492.892293308974" "X173.831930748219" "X420.157767082518"
## [94] "X1411.56721539583" "X173.117683745721" "X326.186522086171"
## [97] "X177.054713421273" "X363.089853013778" "X338.08099104949"
## [100] "X379.056256037961"
##
## $value
## value.var
## X360.100855094492 -0.343087113
## X689.406786965204 -0.243980813
## X313.092613575399 -0.212848057
## X325.111894653303 -0.210978045
## X185.190588615838 -0.201115999
## X324.137371517562 -0.195958410
## X323.133879784681 -0.179196447
## X690.410098388308 -0.163741506
## X389.160740120785 -0.161347090
## X691.422542368164 -0.153900693
## X319.157448204942 -0.152947273
## X367.123501601156 -0.147500421
## X321.173284297382 -0.144279341
## X555.093874891555 -0.144164122
```

X391.122971286377 -0.143791645
X944.449967310942 -0.143389701
X317.074072751279 -0.138157675
X322.17642106672 -0.127191868
X1240.15789775369 -0.126435373
X639.046287279312 -0.124081960
X183.175015349613 -0.122999679
X351.128821579971 -0.122763557
X169.989509429336 -0.120186465
X556.097536191383 -0.117034237
X420.119920756105 -0.115473266
X1242.17308762521 -0.112731694
X365.145723604088 -0.111058298
X232.95540138555 -0.110330724
X373.115564667084 -0.104760137
X503.071140727181 -0.104718524
X564.931775721697 -0.102057012
X623.08265073133 -0.099408492
X943.448842600956 -0.097763217
X419.116546770163 -0.096815661
X571.058061473006 -0.095732735
X1058.40882060084 -0.092767379
X671.194040936126 -0.090464559
X344.098775089955 -0.088669610
X352.132376994544 -0.085510281
X483.86951634573 -0.082308021
X351.129035832649 -0.082287182
X324.173145983115 -0.077698371
X459.11029979044 -0.077395460
X1178.17585807417 -0.076698722
X487.104967151007 -0.075986692
X578.160197428851 -0.075288966
X320.160371151912 -0.074244898
X359.097883259062 -0.074035136
X349.115949924823 -0.073591353
X431.154386005126 -0.072490068
X604.208532818709 -0.072236378
X340.075145186925 -0.071596818
X779.244559972528 -0.071183880
X537.082677183086 -0.069883004
X481.088866272944 -0.068707232
X435.08241905927 -0.068085888
X1243.17676615566 -0.056126462
X707.029978476026 -0.055657940
X488.109293490248 -0.055151086
X461.06951805573 -0.053574443
X380.158184370181 -0.051874729
X561.03604963324 -0.050211614
X737.232626061852 -0.049763044
X403.160178776661 -0.049653036
X648.205609857975 -0.048848009
X402.11215295727 -0.045746915
X483.119144506346 -0.043102012
X341.965577057768 -0.042460376


```

## X531.066415927912 -0.040675698
## X727.164615818962 -0.040650908
## X379.328828228171 -0.033741877
## X253.107386469566 -0.033311519
## X582.225274749511 -0.029564631
## X538.198579985026 -0.028853103
## X509.112370096629 -0.028625407
## X797.114458482459 -0.028095612
## X170.985886728112 -0.027017507
## X339.118846056087 -0.026546405
## X605.192195695578 -0.026545106
## X581.222187754695 -0.026514681
## X317.075176414668 -0.025846378
## X480.872661584228 -0.024447803
## X191.106884410638 -0.019600042
## X337.077569276891 -0.017707061
## X813.080943994947 -0.015917969
## X274.831245724891 -0.013837994
## X1244.17196607701 -0.012832722
## X125.083041728111 -0.011628397
## X403.124191936883 -0.011363122
## X745.249207080709 -0.010654157
## X492.892293308974 -0.008953790
## X173.831930748219 -0.008696817
## X420.157767082518 -0.007923736
## X1411.56721539583 -0.006699450
## X173.117683745721 -0.005112417
## X326.186522086171 -0.002636670
## X177.054713421273 -0.002103826
## X363.089853013778 -0.002103020
## X338.08099104949 -0.001091795
## X379.056256037961 -0.001047459
##
## $comp
## [1] 1

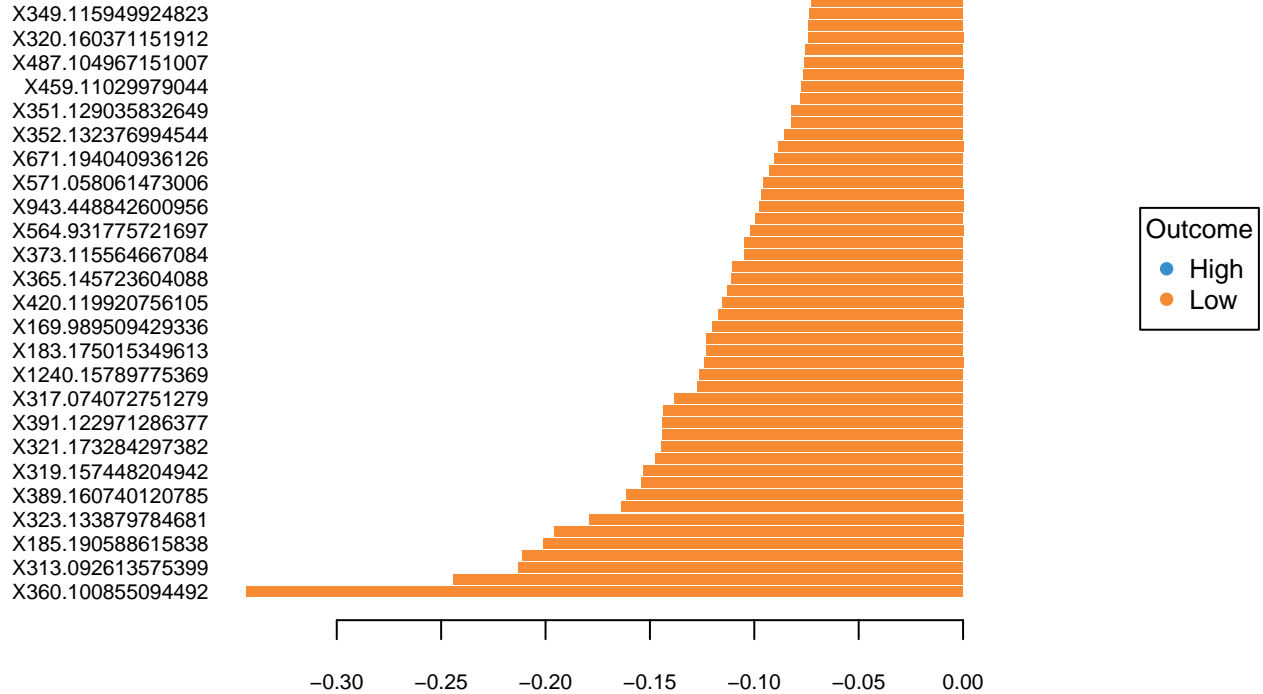
```

```

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

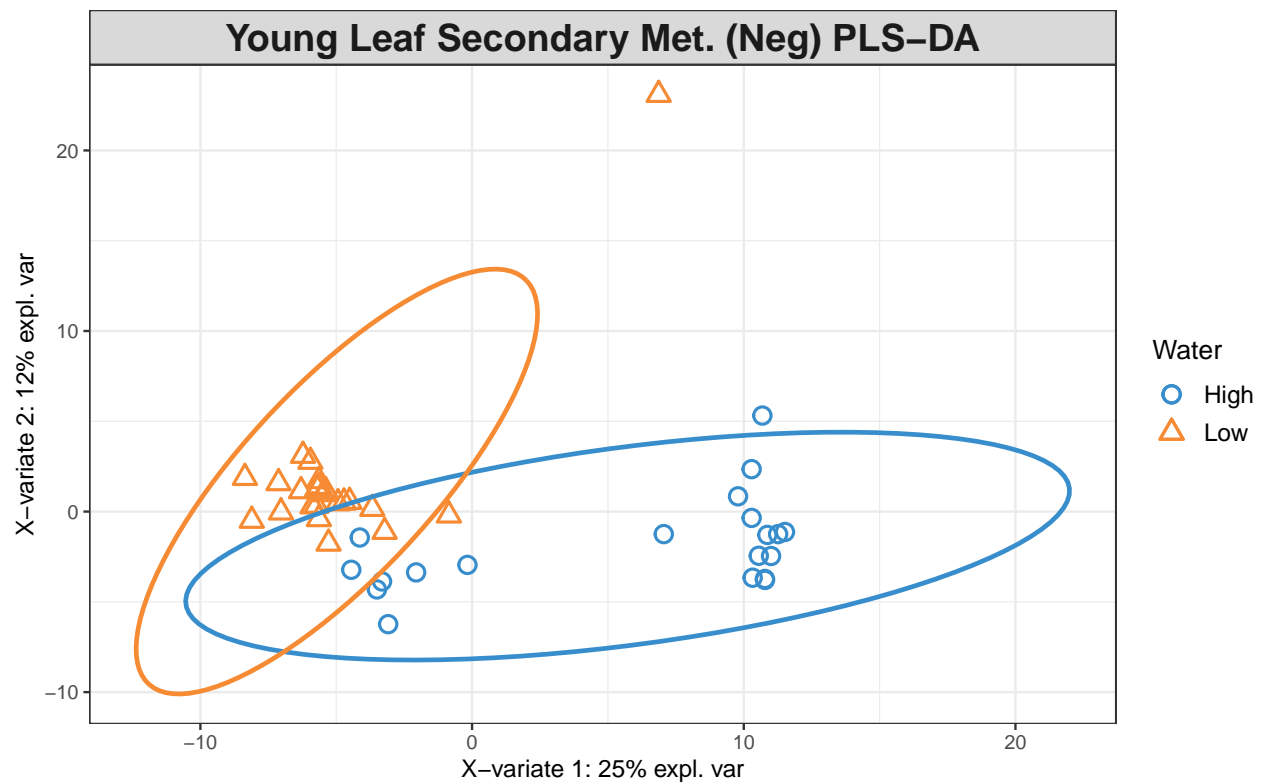
```

Contribution on comp 1

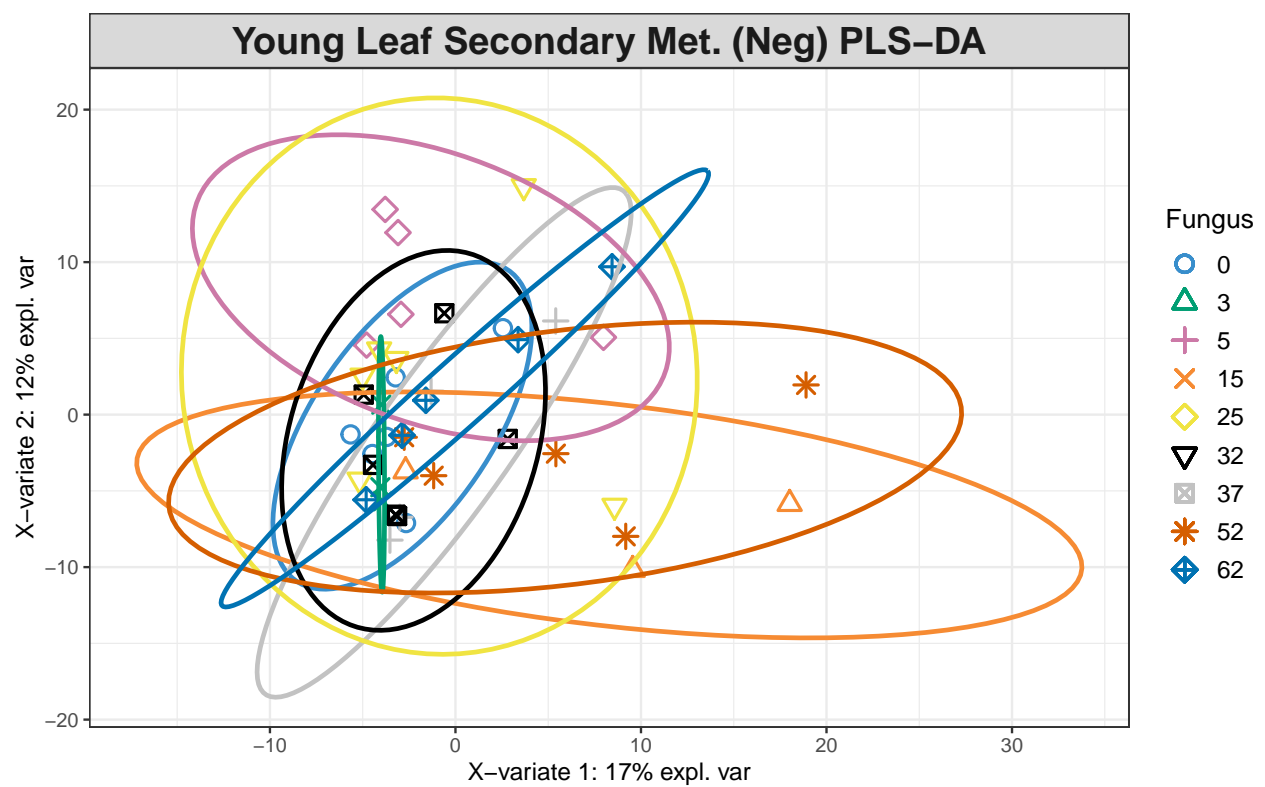


```
# Young Leaf
young.splsda <- mixOmics::splsda(scaled_Y_young, class$Water,
  keepX = c(100, 100))
young.splsda2 <- mixOmics::splsda(scaled_Y_young, class$Fungus,
  keepX = c(100, 100))

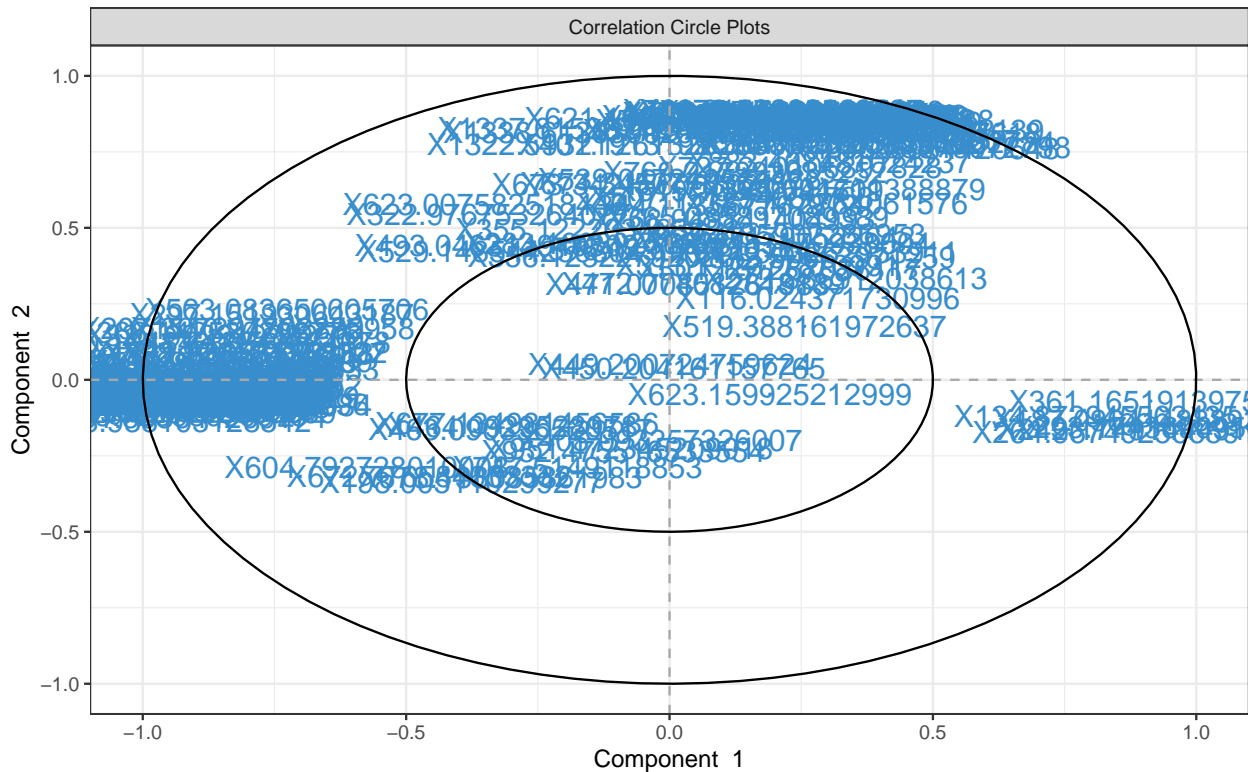
# plot pls-da
plotIndiv(young.splsda, ind.names = F, legend = T, title = "Young Leaf Secondary Met. (Neg) PLS-DA",
  legend.title = "Water", ellipse = T)
```



```
plotIndiv(young.splsda2, ind.names = F, legend = T, title = "Young Leaf Secondary Met. (Neg) PLS-DA",
  legend.title = "Fungus", ellipse = T)
```



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



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

```
## $name
## [1] "X150.105130035688" "X802.732174968048" "X242.174914449158"
## [4] "X1124.61055906716" "X676.816971205036" "X1150.60468753138"
## [7] "X838.711000805533" "X782.740233674195" "X151.01222317911"
## [10] "X325.016405270187" "X849.977486523324" "X872.724302253429"
## [13] "X149.993715595454" "X1192.58529322529" "X1540.97796285958"
## [16] "X754.740789349138" "X200.006491950786" "X293.174369095004"
## [19] "X1294.70679432442" "X1348.5287504802" "X1386.51213623856"
## [22] "X848.720751962966" "X1158.59946171176" "X121.99862199071"
## [25] "X794.733724869254" "X110.006290772417" "X323.992919982561"
## [28] "X1090.73888347575" "X1154.60400860062" "X798.726524348207"
## [31] "X1110.61644505682" "X323.995293823304" "X283.006624782518"
## [34] "X804.740427603084" "X1066.63653837698" "X1160.5969683167"
## [37] "X674.818219943495" "X627.923106883369" "X1036.98344363687"
## [40] "X756.735148419103" "X844.982584773594" "X1116.61329569819"
## [43] "X326.018113960963" "X1364.95342014471" "X852.733071582736"
## [46] "X322.976781158022" "X297.307224802892" "X866.707975254094"
## [49] "X497.995790827568" "X822.723393027696" "X830.737149656595"
## [52] "X133.998876296918" "X482.966873192152" "X874.707571794695"
## [55] "X424.843687850262" "X1158.72895999915" "X671.998240343484"
## [58] "X369.877177619296" "X1350.52689216505" "X1539.96588681403"
## [61] "X297.151930643187" "X850.821164746369" "X484.12840584078"
## [64] "X1078.62469612196" "X134.872945563435" "X845.998051933969"
```

```

## [67] "X840.71024774808" "X514.986512120569" "X421.955540488549"
## [70] "X751.803644269174" "X828.732413615017" "X563.083650605706"
## [73] "X1019.99560196699" "X1191.96821780793" "X152.014000365393"
## [76] "X1018.98243765579" "X1138.61401223754" "X892.958412895177"
## [79] "X1192.98175033474" "X632.970118158626" "X361.165191397563"
## [82] "X558.938103128542" "X854.728311606954" "X896.710609558084"
## [85] "X843.965940610961" "X670.981382467921" "X657.982854820673"
## [88] "X366.142530296776" "X538.815638527027" "X264.967492566591"
## [91] "X187.096414888931" "X1128.61230614359" "X824.820684339506"
## [94] "X826.731366769468" "X1177.95950779409" "X337.904579660098"
## [97] "X1711.93957530353" "X916.69988248716" "X1103.61628997104"
## [100] "X339.90280551721"
##
## $value
##          value.var
## X150.105130035688 -0.282039719
## X802.732174968048 -0.227414352
## X242.174914449158  0.215910257
## X1124.61055906716 -0.198782419
## X676.816971205036 -0.192655075
## X1150.60468753138 -0.183273749
## X838.711000805533 -0.181332267
## X782.740233674195 -0.180686657
## X151.01222317911  -0.176989099
## X325.016405270187 -0.175849728
## X849.977486523324 -0.174259645
## X872.724302253429 -0.160769967
## X149.993715595454 -0.156421706
## X1192.58529322529 -0.154190387
## X1540.97796285958 -0.151174229
## X754.740789349138 -0.149717546
## X200.006491950786 -0.149376121
## X293.174369095004  0.144319007
## X1294.70679432442 -0.140253834
## X1348.5287504802  -0.137292240
## X1386.51213623856 -0.134315079
## X848.720751962966 -0.131984710
## X1158.59946171176 -0.128594472
## X121.99862199071  -0.126133016
## X794.733724869254 -0.124014822
## X110.006290772417 -0.122819961
## X323.992919982561 -0.115547980
## X1090.73888347575 -0.114224424
## X1154.60400860062 -0.109269030
## X798.726524348207 -0.102943924
## X1110.61644505682 -0.100529543
## X323.995293823304 -0.100100667
## X283.006624782518 -0.098992349
## X804.740427603084 -0.092445129
## X1066.63653837698 -0.088592213
## X1160.5969683167  -0.088291119
## X674.818219943495 -0.085416018
## X627.923106883369 -0.083266055
## X1036.98344363687 -0.083040165

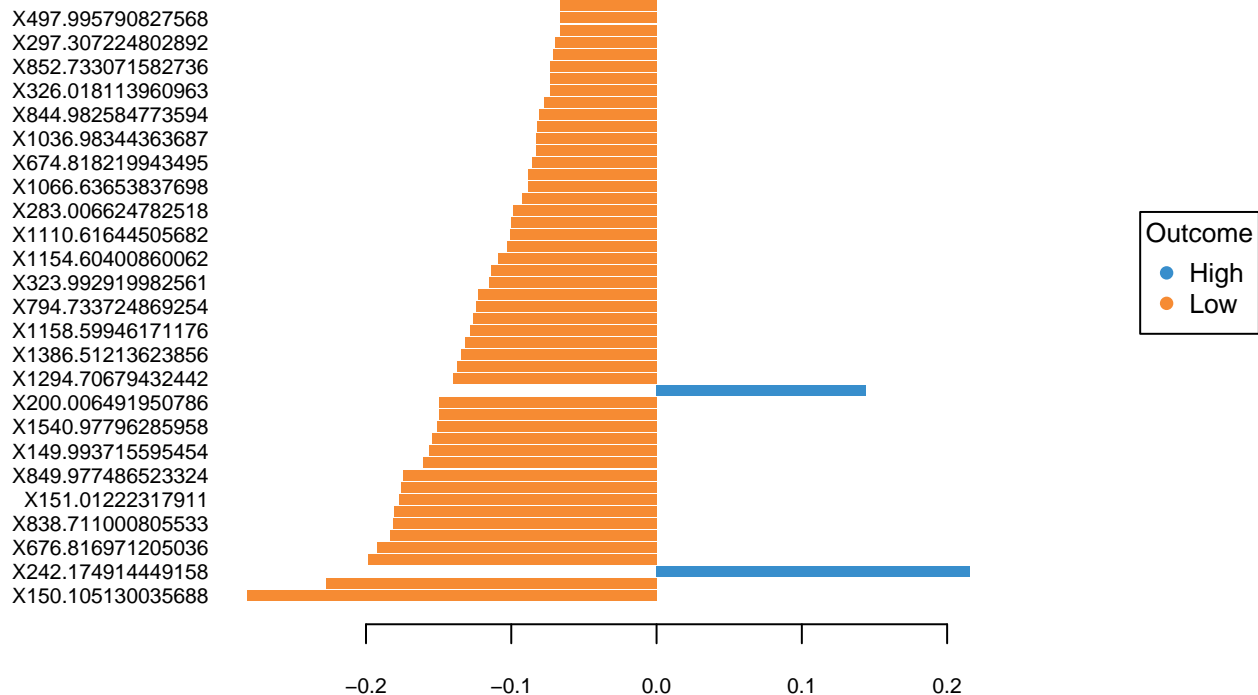
```

X756.735148419103 -0.081939466
 ## X844.982584773594 -0.081009163
 ## X1116.61329569819 -0.077158656
 ## X326.018113960963 -0.073578175
 ## X1364.95342014471 -0.073276382
 ## X852.733071582736 -0.073097286
 ## X322.976781158022 -0.071190101
 ## X297.307224802892 -0.070158011
 ## X866.707975254094 -0.066433552
 ## X497.995790827568 -0.066224702
 ## X822.723393027696 -0.066184178
 ## X830.737149656595 -0.063541450
 ## X133.998876296918 -0.063294993
 ## X482.966873192152 -0.063122686
 ## X874.707571794695 -0.059919434
 ## X424.843687850262 -0.059134367
 ## X1158.72895999915 -0.058494755
 ## X671.998240343484 -0.058106054
 ## X369.877177619296 -0.053828920
 ## X1350.52689216505 -0.052171922
 ## X1539.96588681403 -0.049501193
 ## X297.151930643187 -0.048493255
 ## X850.821164746369 -0.047328889
 ## X484.12840584078 -0.046426405
 ## X1078.62469612196 -0.043632423
 ## X134.872945563435 0.042506082
 ## X845.998051933969 -0.041902578
 ## X840.71024774808 -0.041651192
 ## X514.986512120569 -0.041251783
 ## X421.955540488549 -0.040089622
 ## X751.803644269174 -0.039688088
 ## X828.732413615017 -0.038866785
 ## X563.083650605706 -0.038776647
 ## X1019.99560196699 -0.038232314
 ## X1191.96821780793 -0.037011527
 ## X152.014000365393 -0.036591298
 ## X1018.98243765579 -0.034731819
 ## X1138.61401223754 -0.033671300
 ## X892.958412895177 -0.033540932
 ## X1192.98175033474 -0.033489661
 ## X632.970118158626 -0.028243242
 ## X361.165191397563 0.026629148
 ## X558.938103128542 -0.026275125
 ## X854.728311606954 -0.025659016
 ## X896.710609558084 -0.023886971
 ## X843.965940610961 -0.023711050
 ## X670.981382467921 -0.021465443
 ## X657.982854820673 -0.020212723
 ## X366.142530296776 -0.019123888
 ## X538.815638527027 -0.017573798
 ## X264.967492566591 0.015608624
 ## X187.096414888931 0.015526290
 ## X1128.61230614359 -0.014892070
 ## X824.820684339506 -0.013553869

```
## X826.731366769468 -0.012032830
## X1177.95950779409 -0.011616170
## X337.904579660098 -0.011374628
## X1711.93957530353 -0.009008417
## X916.69988248716 -0.008695822
## X1103.61628997104 -0.006663306
## X339.90280551721 -0.006540303
##
## $comp
## [1] 1
```

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

Contribution on comp 1



Heatmaps of Averaged Data

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

```
## Old Leaves
av_Y_old <- aggregate(Y_old, by = list(class$Water, class$Fungus),
  FUN = "mean", simplify = T, data = class)
av.old.plsda <- mixOmics::plsda(av_Y_old[, 3:3735], av_Y_old$Group.1) # water

# heatmap
oldcim <- cim(av.old.plsda, title = "Old Leaf Secondary Met. (neg) Averaged Over Water",
  col.names = F, xlab = "Secondary Metabolites", save = "png",
  name.save = "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/")

## 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[, 3:2566], av_Y_young$Group.1) # water
av.young.plsda2 <- mixOmics::plsda(av_Y_young[, 3:2566], av_Y_young$Group.2) # fungus

# heatmap
youngcim <- cim(av.young.plsda, title = "Young Leaf Secondary Met. (neg) Averaged Over Water",
  col.names = F, xlab = "Secondary Metabolites", save = "png",
  name.save = "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/")

# heatmap
youngcim2 <- cim(av.young.plsda2, title = "Young Leaf Secondary Met. (neg) Averaged Over Fungi",
  col.names = F, xlab = "Secondary Metabolites", save = "png",
  name.save = "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/")

```

Indicator Analysis

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

```

# Old Leaf
indicator_Water0 <- indval(Y_old, clustering = class$Water, numitr = 999,
  type = "long")
summary(indicator_Water0)

```

```

##
## Sum of probabilities          = 866.638638638639
##
## Sum of Indicator Values      = 2138
##
## Sum of Significant Indicator Values = 1054.04
##
## Number of Significant Indicators   = 1691
##
## Significant Indicator Distribution
##
##      1      2
## 111 1580

```

```

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

```

```

##
## Sum of probabilities          = 706.783783783784
##
## Sum of Indicator Values      = 1526.49
##
## Sum of Significant Indicator Values = 721.05
##
## Number of Significant Indicators   = 1085
##

```



```
## Significant Indicator Distribution
```

```
##
```

```
## 1 2
```

```
## 468 617
```

```
summary(indicator_Fungus)
```

```
##          cluster indicator_value probability
## X126.978565369291      2          0.1961 0.033033033
## X174.955281450343      3          0.1998 0.031031031
## X362.876199428467      3          0.1721 0.040040040
## X732.753622137795      3          0.1641 0.022022022
## X563.890207842266      3          0.1579 0.045045045
## X570.842116351837      3          0.1436 0.039039039
## X719.973261950079      4          0.3228 0.009009009
## X495.959962551678      4          0.2521 0.049049049
## X114.935142929173      4          0.2214 0.027027027
## X290.987559951518      4          0.2110 0.041041041
## X297.965942084671      4          0.2017 0.022022022
## X506.971749443834      4          0.2013 0.040040040
## X1016.95098645776      4          0.1952 0.045045045
## X839.970909295556      4          0.1841 0.049049049
## X300.152669002914      4          0.1832 0.033033033
## X174.95528178745      4          0.1823 0.041041041
## X157.998103785124      4          0.1790 0.026026026
## X314.169131919955      4          0.1782 0.045045045
## X433.126629159221      4          0.1775 0.044044044
## X451.856709396092      4          0.1570 0.027027027
## X288.935299439552      4          0.1514 0.007007007
## X442.893065530899      4          0.1462 0.035035035
## X150.914649588063      4          0.1432 0.043043043
## X672.776584033332      4          0.1426 0.045045045
## X399.13009701304      5          0.2728 0.018018018
## X582.182538128675      5          0.2635 0.024024024
## X561.235165916746      5          0.2544 0.043043043
## X439.157916063253      5          0.2198 0.022022022
## X596.203846215805      5          0.2053 0.026026026
## X462.168292694096      5          0.2049 0.022022022
## X791.274337542169      5          0.2033 0.022022022
## X415.159188680521      5          0.1996 0.046046046
## X461.164714419708      5          0.1995 0.034034034
## X616.22881415752      5          0.1966 0.042042042
## X595.199358972177      5          0.1941 0.013013013
## X793.288299990309      5          0.1907 0.023023023
## X443.115738244395      5          0.1835 0.037037037
## X375.127006310969      5          0.1786 0.034034034
## X376.129076720164      5          0.1762 0.025025025
## X438.12133847032      5          0.1670 0.035035035
## X170.867102638045      7          0.1541 0.039039039
## X177.038796119533      8          0.2596 0.006006006
## X207.05046346149      9          0.2156 0.031031031
##
## Sum of probabilities          = 1537.42042042042
##
## Sum of Indicator Values      = 445.52
```

```
##
## Sum of Significant Indicator Values = 8.4
##
## Number of Significant Indicators = 43
##
## Significant Indicator Distribution
##
## 2 3 4 5 7 8 9
## 1 5 18 16 1 1 1
```

13. Disect indval object and export results to a csv file.

```
Orelfrq <- indicator_Water0$relfrq # relative frequency of species in classes
Orelabu <- indicator_Water0$relabu # relative abundance of species in classes
Oindval <- indicator_Water0$indval # the indicator value for each species
Omaxcls <- data.frame(indicator_Water0$maxcls) # the class each species has max indicator value for
Oindcls <- data.frame(indicator_Water0$indcls) # the indicator value for each species to its max class
Opval <- data.frame(indicator_Water0$pval) # the probability of obtaining as high an indicator value as

write.csv(cbind(Orelfrq, Orelabu, Oindval, Omaxcls, Oindcls,
  Opval), "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/Secor

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

write.csv(cbind(Yrelfrq, Yrelabu, Yindval, Ymaxcls, Yindcls,
  Ypval), "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/Secor

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

write.csv(cbind(relfrq, relabu, indval, maxcls, indcls, pval),
  "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/Secondary Me
```

Secondary Metabolites (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_SM_pos <- read_tsv(paste(path, "XCMS Online Results/O_SM_Pos/XCMS.annotated.Report_1394418.tsv",
  sep = ""))
```

```

Y_SM_pos <- read_tsv(paste(path, "XCMS Online Results/Y_SM_Pos/XCMS.annotated.Report_1394440.tsv",
  sep = ""))

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

Y_young <- Y_SM_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: 5800
## 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 66102.35    177497.7 0.2713561 1.862063
##              Z (from F)      Pr(>F)
## Block * Water * Fungus  3.745304 0.0005714286
##
##
## Redundancy Analysis (PCA on fitted values and residuals)
##
##              Trace Proportion Rank
## Fitted      1573.865    0.271356    7
## Residuals  4226.135    0.728644   35
## Total      5800.000    1.000000   42
##
## Eigenvalues
##
##              PC1      PC2      PC3      PC4      PC5      PC6      PC7
## Fitted      613.0058  377.3605  245.4204  108.9341   90.8147   75.5811   62.7487
## Residuals  1125.9274  456.5069  301.3978  228.3176  221.2095  152.5997  143.8765
## Total      1563.9592  668.0796  447.6527  365.5013  244.1181  236.9154  180.9496
##              PC8      PC9      PC10     PC11     PC12     PC13     PC14
## Fitted

```

```
## Residuals 138.6110 120.4927 105.6991 100.1973 86.5102 79.7117 76.6642
## Total 164.0427 148.5100 136.1229 116.6738 109.9659 96.2741 89.0995
## PC15 PC16 PC17 PC18 PC19 PC20 PC21
## Fitted
## Residuals 71.8936 67.2005 61.6455 60.0418 56.5602 54.0174 51.0623
## Total 84.0573 78.2452 75.8352 68.1756 62.8428 61.6040 59.1024
## PC22 PC23 PC24 PC25 PC26 PC27 PC28
## Fitted
## Residuals 49.5758 45.6077 44.4495 42.9826 38.1437 35.0037 32.8700
## Total 56.0853 53.7059 49.4072 48.9840 46.9481 46.5683 43.3817
## PC29 PC30 PC31 PC32 PC33 PC34 PC35
## Fitted
## Residuals 32.0639 28.5369 27.4785 24.8828 23.9346 21.1897 19.2722
## Total 41.0579 37.8943 34.1106 32.7965 31.8990 29.7583 28.4563
## PC36 PC37 PC38 PC39 PC40 PC41 PC42
## Fitted
## Residuals
## Total 26.1381 25.2956 24.1190 24.0956 22.7724 20.3350 18.4638
```

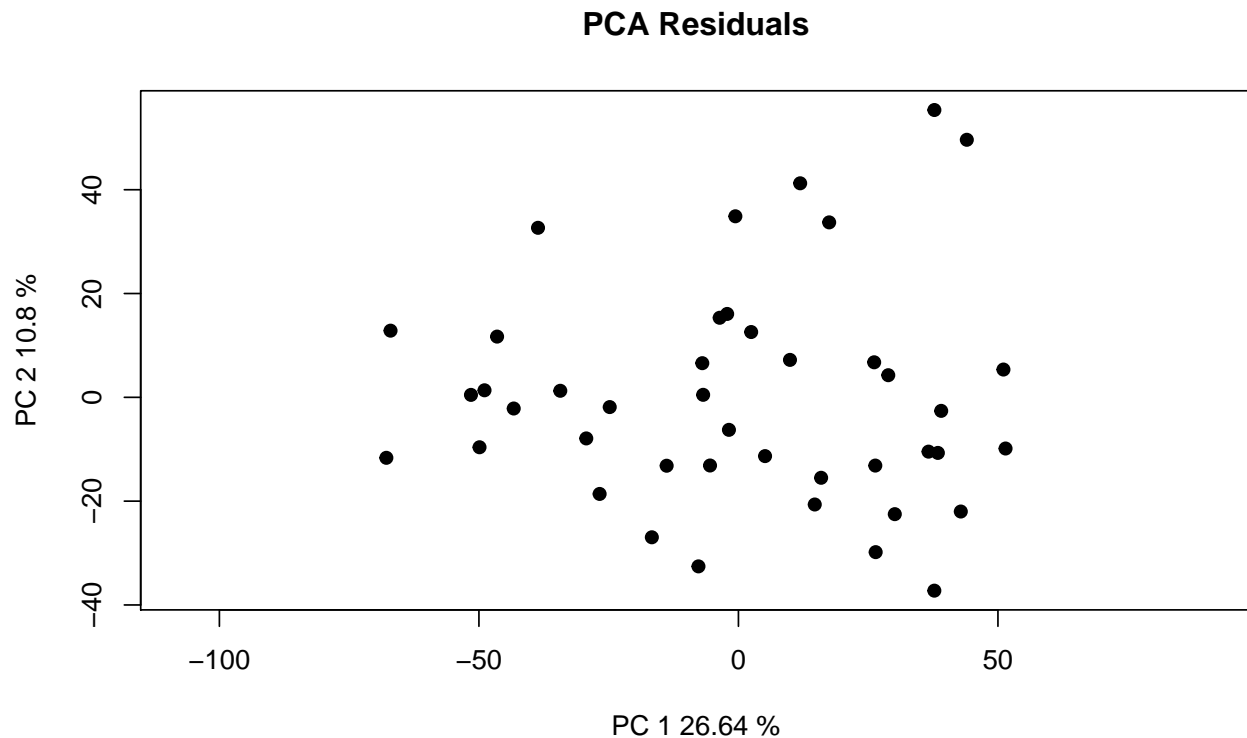
```
Y_LMpos <- lm.rrpp(scaled_Y_young ~ Block * Water * Fungus, data = class,
  SS.type = "III", print.progress = F)
summary(Y_LMpos)
```

```
##
## Linear Model fit with lm.rrpp
##
## Number of observations: 43
## Number of dependent variables: 3559
## 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 38642.68    110835.3 0.2585175 1.743248
##              Z (from F)      Pr(>F)
## Block * Water * Fungus  3.399081 0.001571429
##
##
## Redundancy Analysis (PCA on fitted values and residuals)
##
##              Trace Proportion Rank
## Fitted      920.064 0.2585176    7
## Residuals 2638.936 0.7414824   35
## Total      3559.000 1.0000000   42
##
## Eigenvalues
##
##              PC1      PC2      PC3      PC4      PC5      PC6      PC7
## Fitted      397.7443 240.1652 87.8892 66.1666 47.2456 44.1174 36.7354
## Residuals 633.3110 249.1598 226.5129 162.1190 123.8237 98.7414 86.2987
## Total      812.6211 580.6484 274.7611 195.5419 135.8900 111.6667 91.3427
##              PC8      PC9      PC10      PC11      PC12      PC13      PC14
## Fitted
```

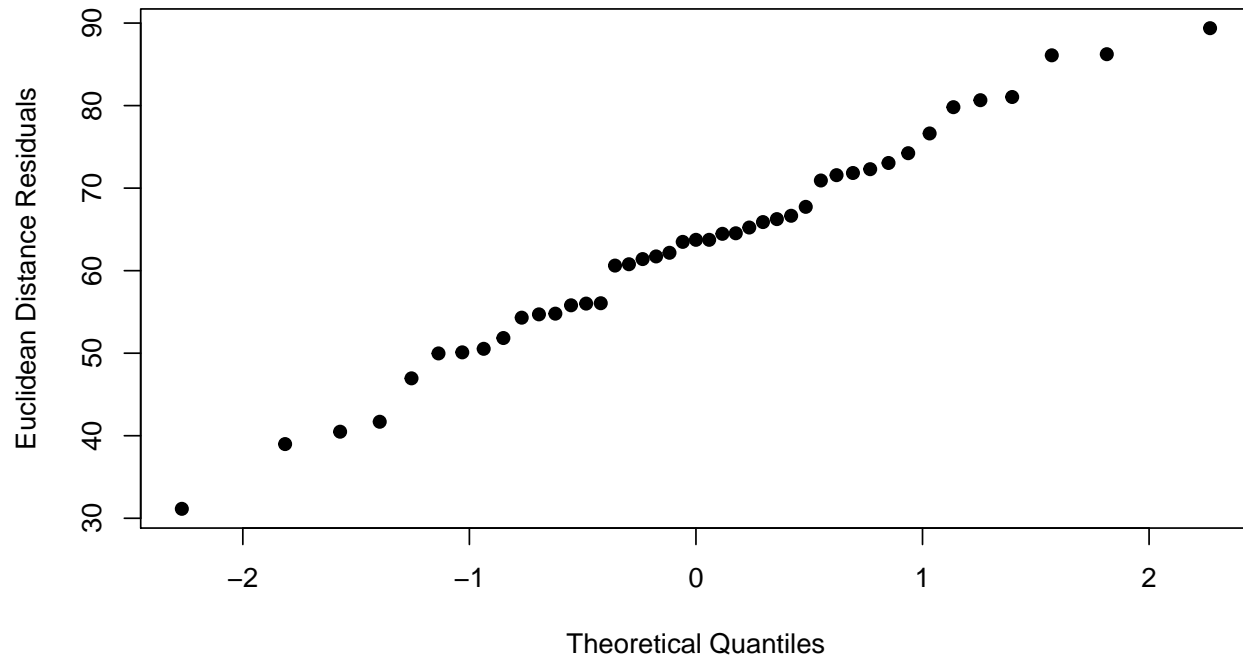
```
## Residuals 74.0692 62.7979 62.2331 54.0214 50.0254 49.2183 47.9739
## Total     81.8073 74.4560 68.9230 65.8313 58.9405 52.9731 51.5130
##           PC15    PC16    PC17    PC18    PC19    PC20    PC21
## Fitted
## Residuals 46.9945 42.0306 40.1348 39.0245 37.6032 36.6949 35.0499
## Total     49.3036 48.1786 45.4179 43.1774 42.9165 40.3182 39.0791
##           PC22    PC23    PC24    PC25    PC26    PC27    PC28
## Fitted
## Residuals 33.5945 32.0128 31.2296 30.3538 29.6785 28.2886 27.7751
## Total     37.6835 35.9477 35.1338 33.9837 33.6972 32.1289 31.3560
##           PC29    PC30    PC31    PC32    PC33    PC34    PC35
## Fitted
## Residuals 27.0825 26.2620 25.4541 24.0211 23.4611 22.1253 19.7593
## Total     30.5374 29.9458 28.8928 28.0758 27.1667 26.5044 26.1782
##           PC36    PC37    PC38    PC39    PC40    PC41    PC42
## Fitted
## Residuals
## Total     25.4687 25.1620 23.5154 22.1223 21.2005 19.9071 19.0846
```

4. Examine RRPP plots to check for assumptions.

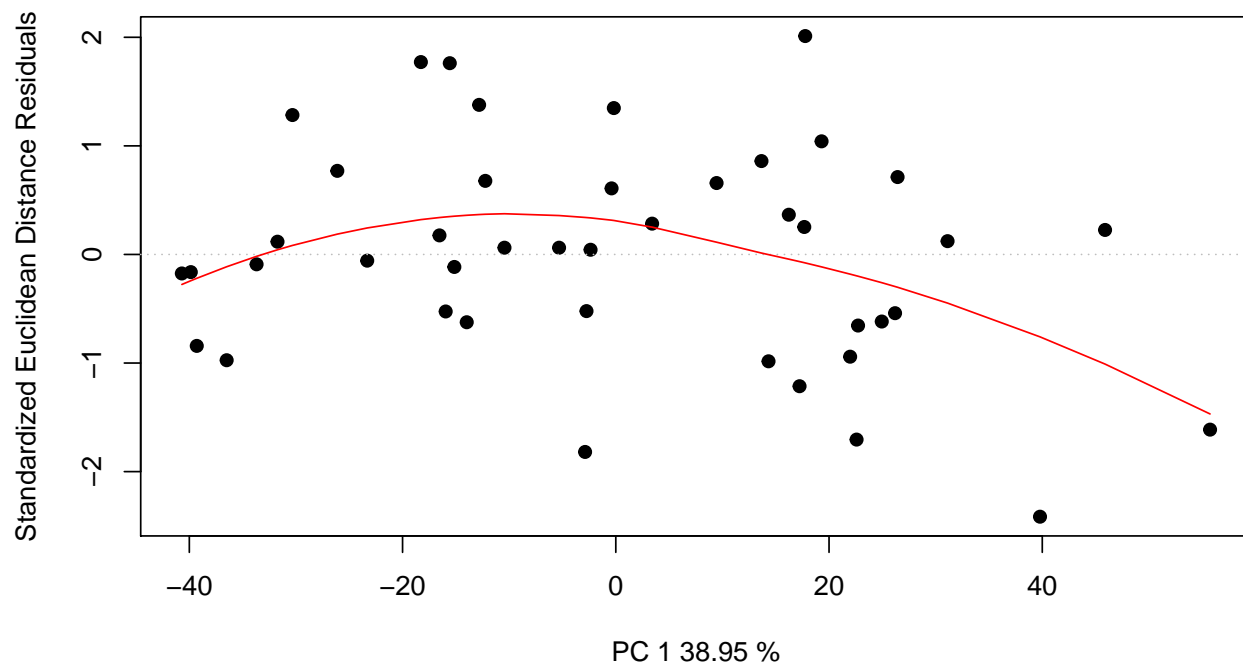
```
## Old Leaves residuals vs fitted values (homoscedasticity
## check)
Odiagnostics <- plot(O_LMpos, type = "diagnostics")
```



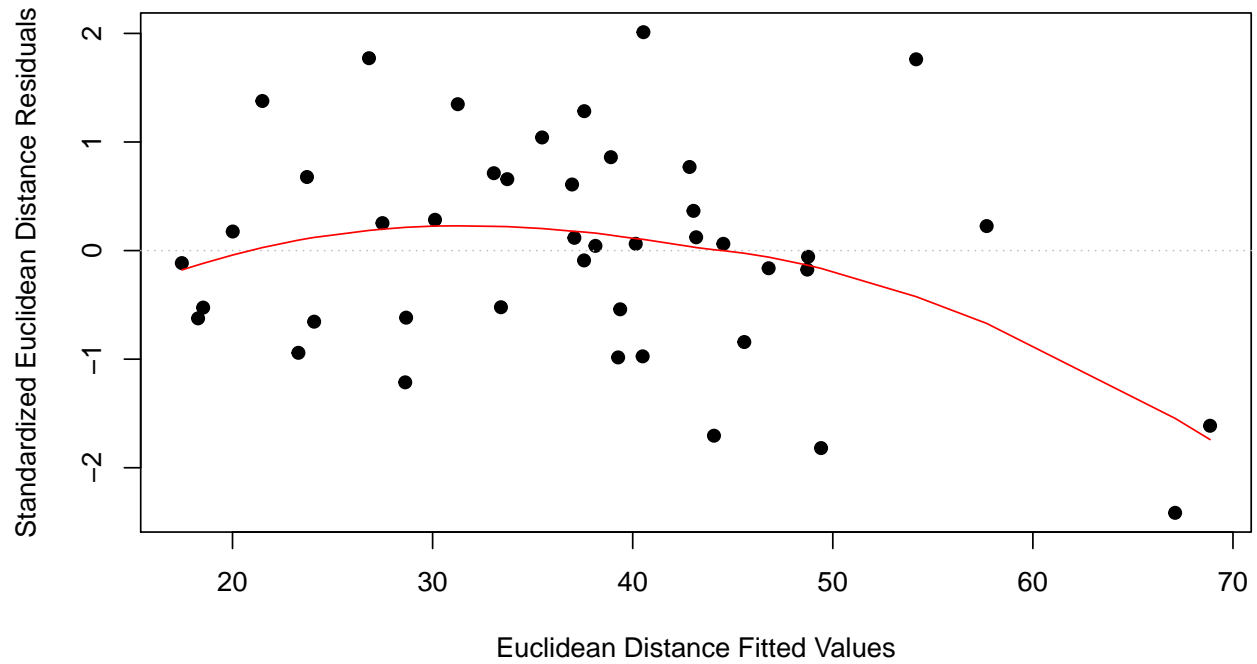
Q-Q plot



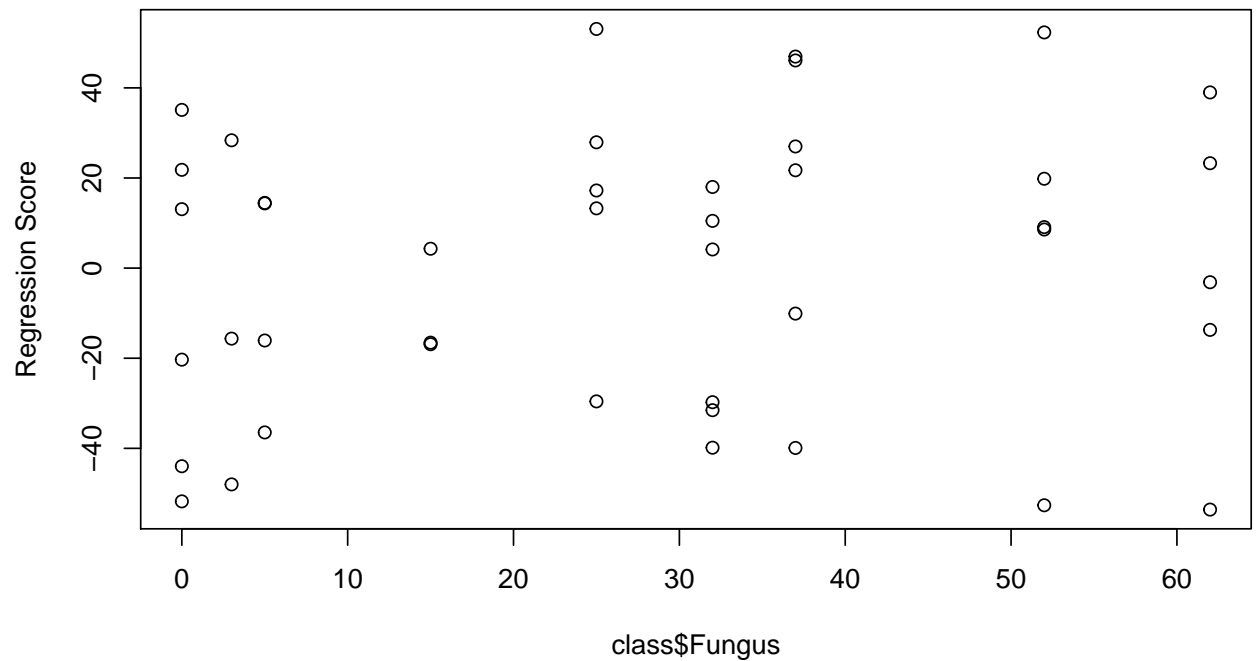
Residuals vs. PC 1 fitted



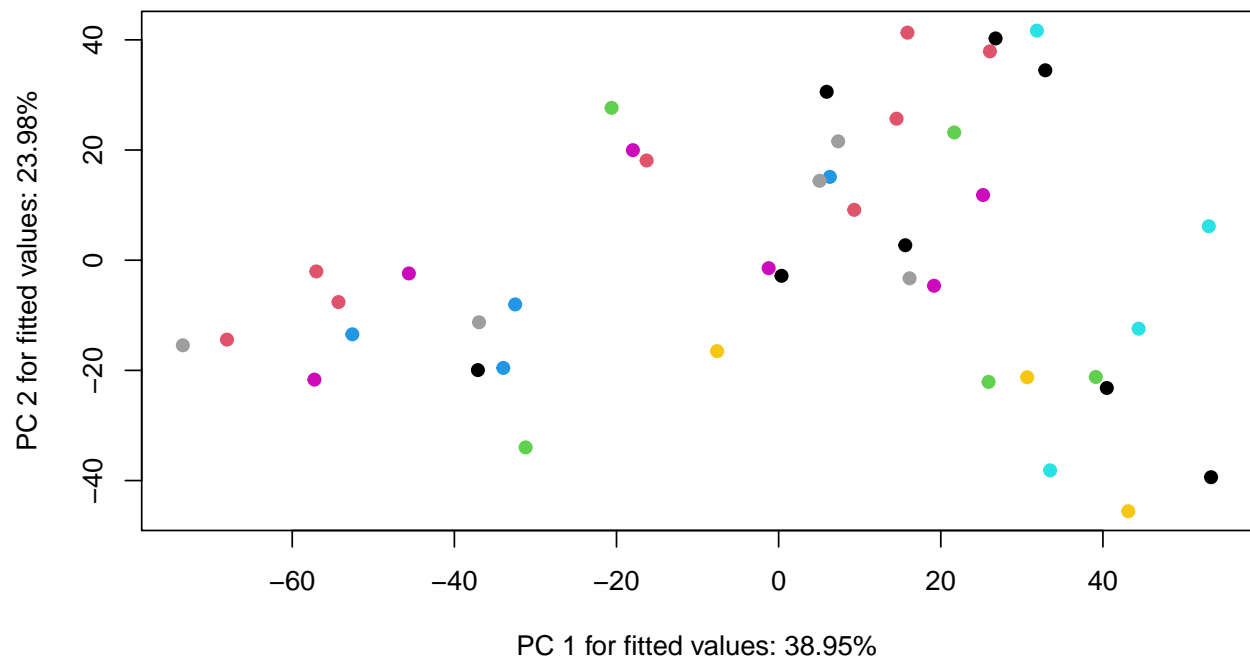
Residuals vs. Fitted



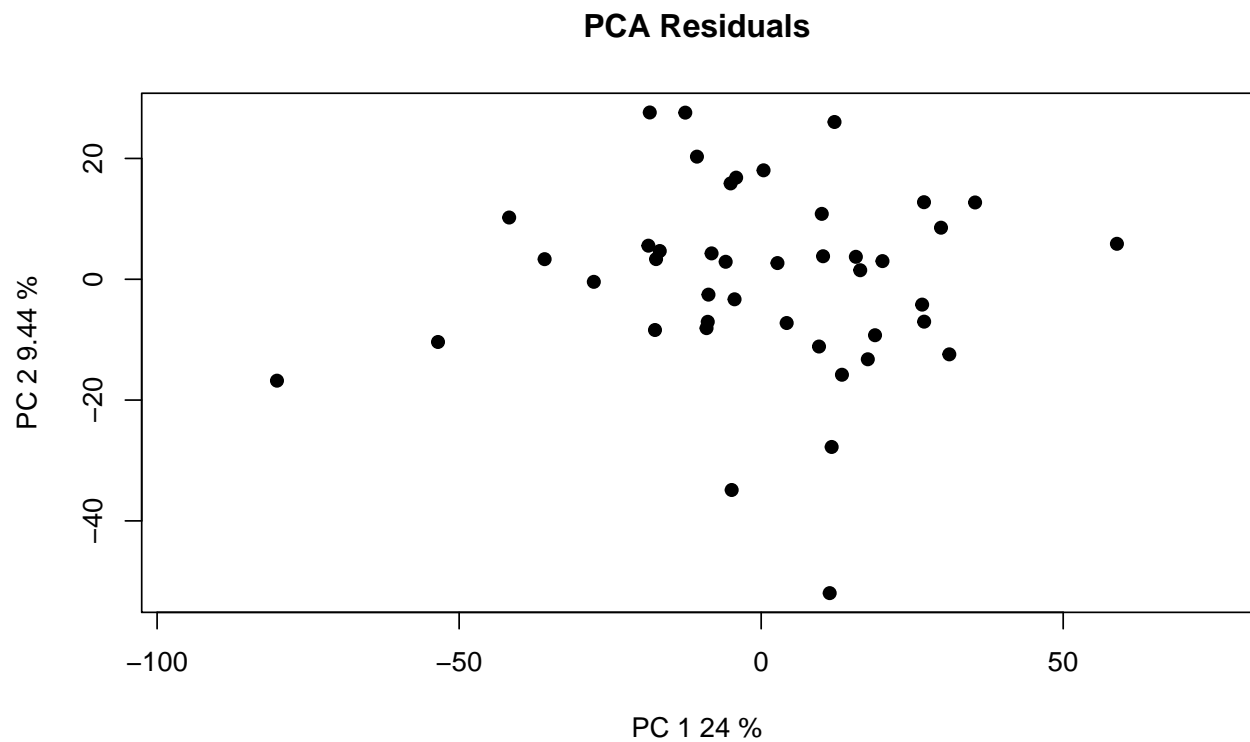
```
# linear regression plot
Oregression <- plot(O_LMpos, type = "regression", predictor = class$Fungus,
  reg.type = "RegScore")
```



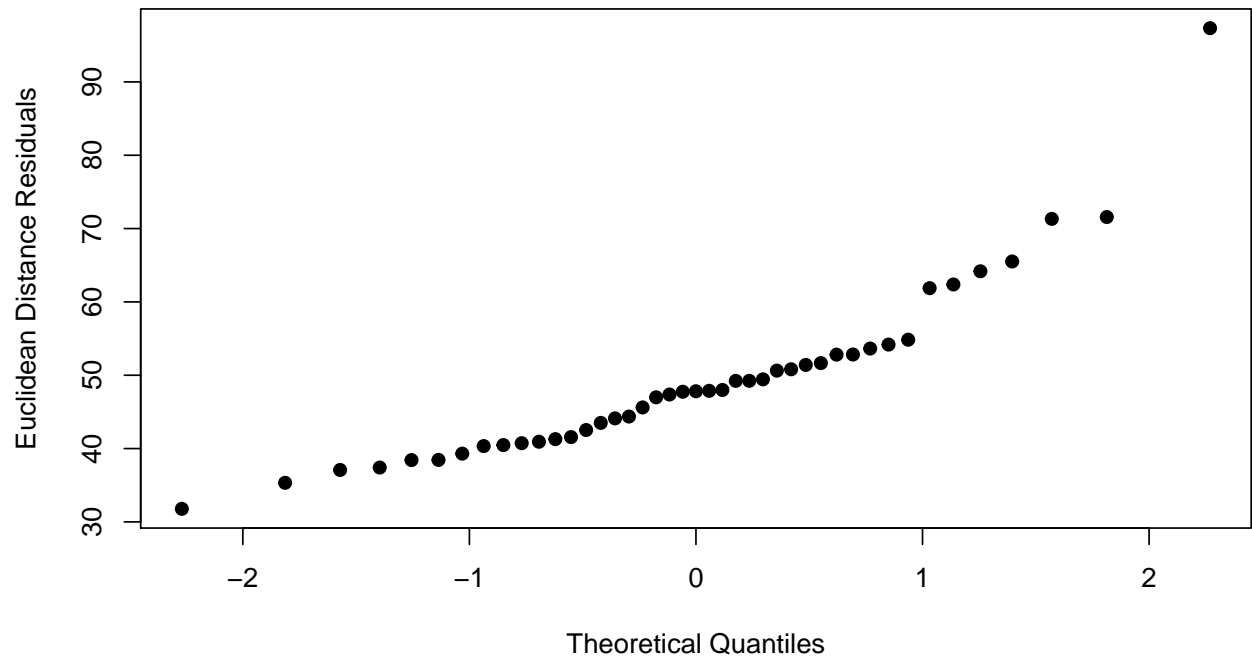
```
# pca plot
Opplot <- plot(O_LMpos, type = "PC", pch = 19, col = interaction(class$Water,
  class$Fungus))
```



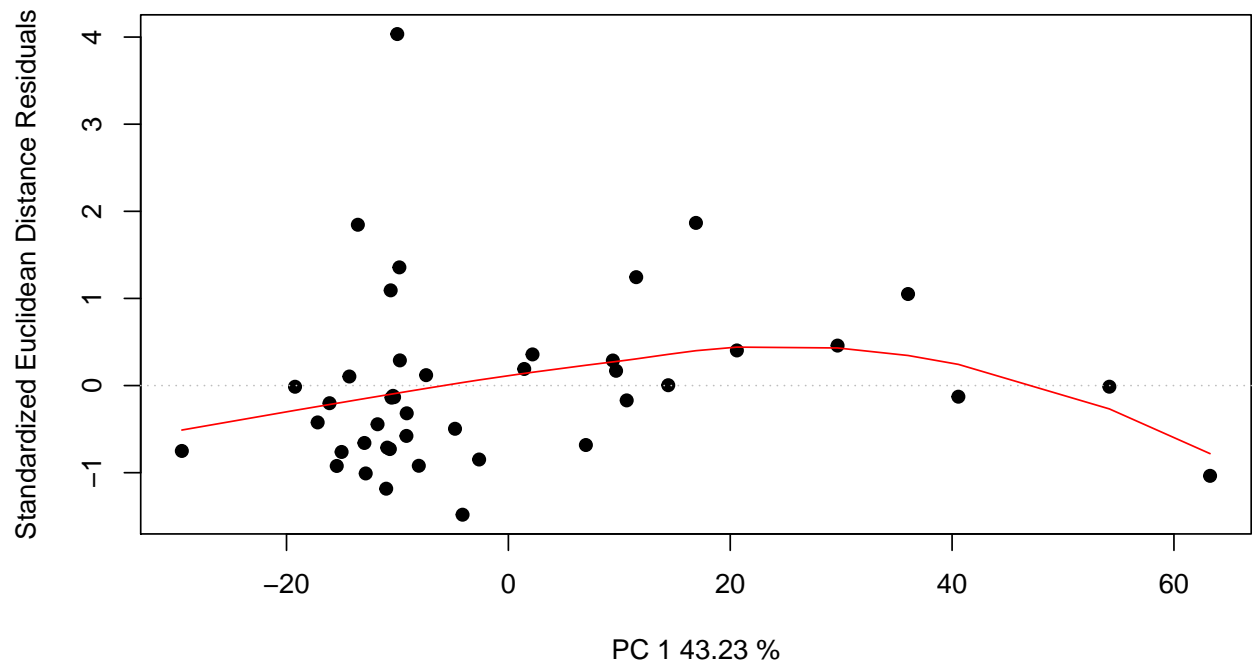
```
## Young Leaves residuals vs fitted values (homoscedasticity
## check)
Ydiagnostics <- plot(Y_LMpos, type = "diagnostics")
```

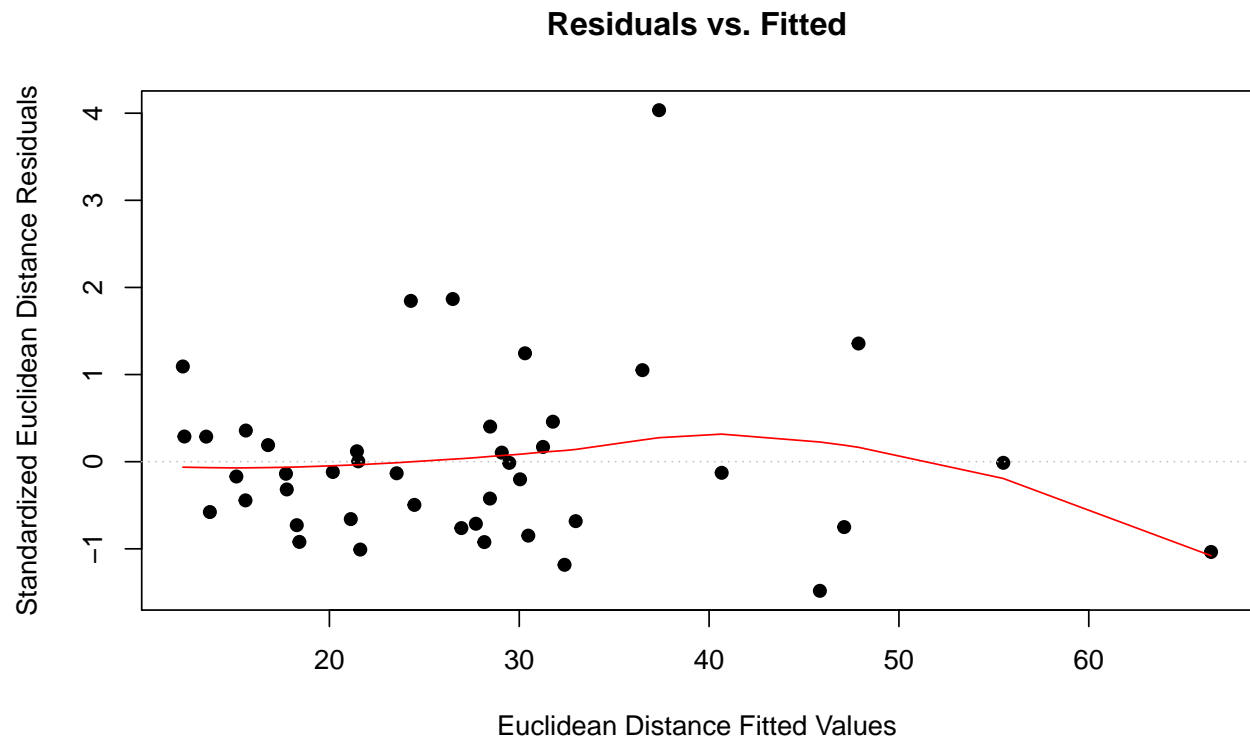


Q-Q plot

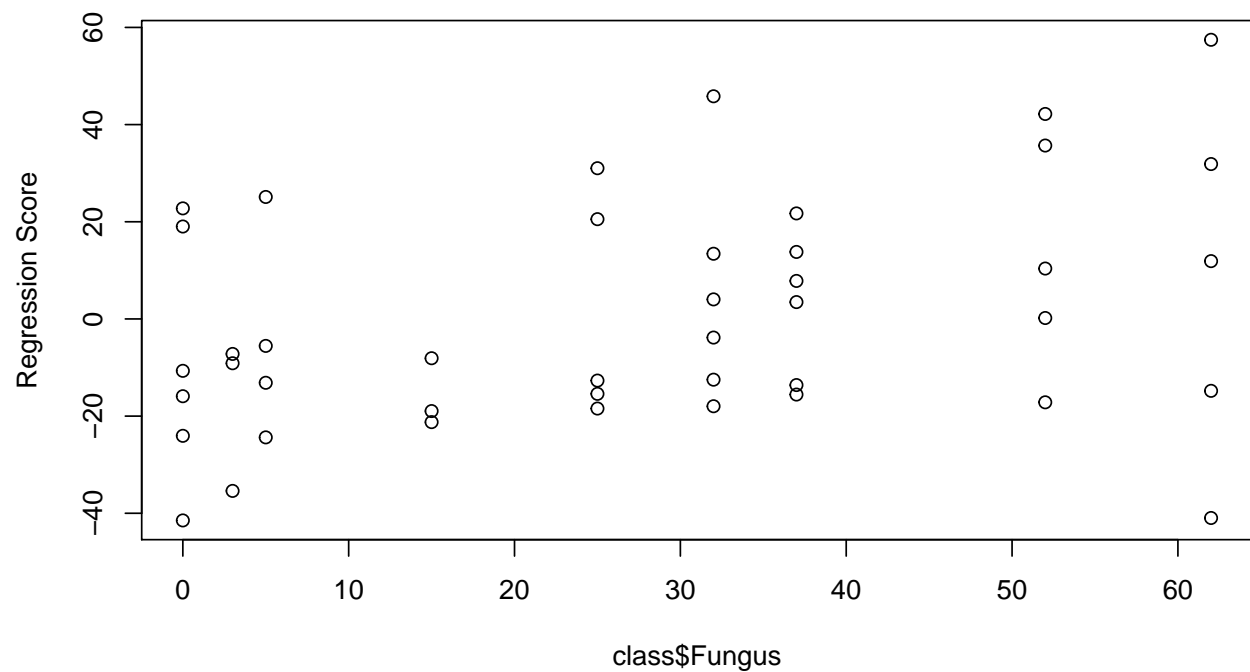


Residuals vs. PC 1 fitted

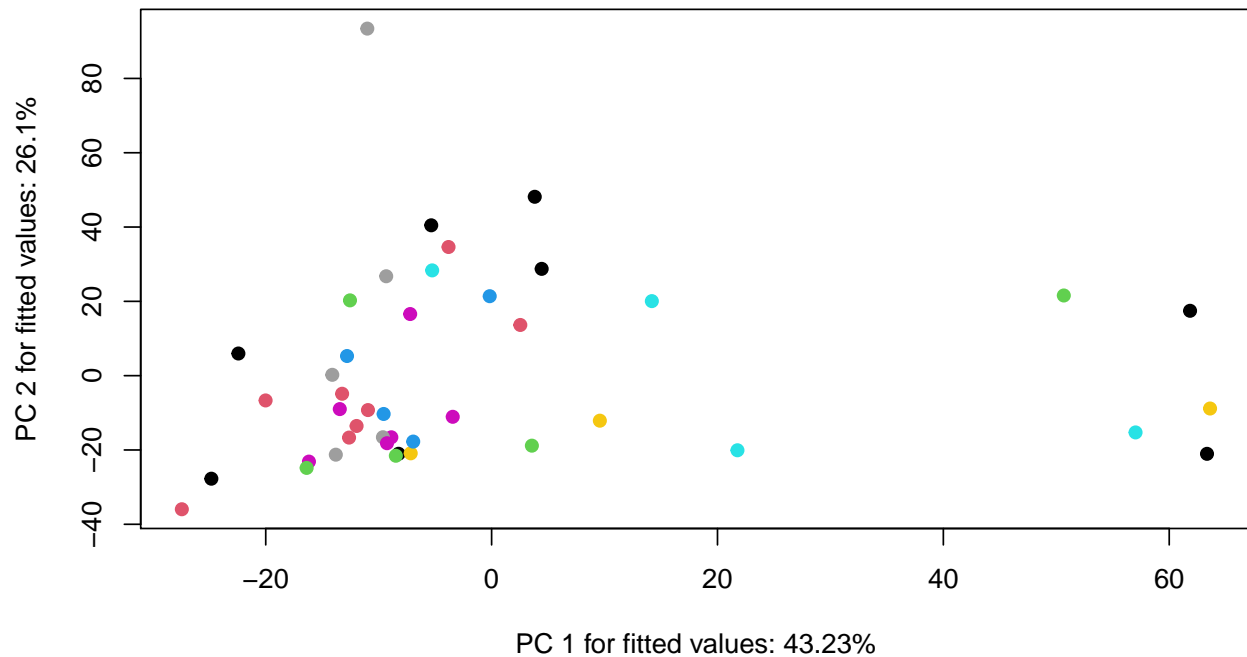




```
# linear regression plot
Yregression <- plot(Y_LMpos, type = "regression", predictor = class$Fungus,
  reg.type = "RegScore")
```



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



5. Perform an RRPP ANOVA and print results.

```
## 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   9842 9842.3 0.04040 1.9408   1.93563 0.043 *
## Water      1   6015 6014.6 0.02469 1.2215   0.96850 0.174
## Fungus     1   8415 8414.6 0.03454 1.9912   1.74049 0.044 *
## Block:Water 1   4924 4923.9 0.02021 0.9709   0.06545 0.428
## Block:Fungus 1   7570 7569.9 0.03108 1.7913   1.68520 0.044 *
## Water:Fungus 1   4625 4625.0 0.01899 1.0944   0.39905 0.352
## Block:Water:Fungus 1   4226 4225.9 0.01735 0.8333  -0.37299 0.607
## Residuals  35 177498 5071.4 0.72864
## Total     42 243600
## ---
## 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
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   5774 5773.8 0.03863 1.8233  1.91902 0.045 *
## Water           1   3326 3325.5 0.02225 0.8231 -0.92641 0.828
## Fungus          1   7286 7285.7 0.04874 1.4310  0.95467 0.155
## Block:Water     1   4040 4040.2 0.02703 1.2758  0.91450 0.161
## Block:Fungus    1   6299 6299.0 0.04214 1.2372  0.61786 0.247
## Water:Fungus    1   3839 3839.1 0.02568 0.7540 -1.29822 0.913
## Block:Water:Fungus 1   5091 5091.4 0.03406 1.6078  1.57153 0.088 .
## Residuals      35 110835 3166.7 0.74148
## Total          42 149478
## ---
## 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)
summary(Oposcoef)

##
## Linear Model fit with lm.rrpp
##
## Number of observations: 43
## Number of dependent variables: 5800
## 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)  82.950278 121.985522 -0.9656612 0.830
## Block        47.308227  42.862108  2.6530050 0.018
## WaterLow     95.740528 108.500604  1.0580682 0.133
## Fungus       2.409128   2.387798  2.0517450 0.047
## Block:WaterLow 44.635517 56.517070  0.4231836 0.266
## Block:Fungus  1.171821  1.211972  1.6991995 0.071
## WaterLow:Fungus 2.477853  3.166755  0.2264772 0.354

```

```
## Block:WaterLow:Fungus 1.171555 1.616995 -0.0172714 0.426
```

```
## Young Leaves test model coefficients
```

```
Yposcoef <- coef(Y_LMpos, test = T)
```

```
summary(Yposcoef)
```

```
##
```

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

```
##
```

```
## Number of observations: 43
```

```
## Number of dependent variables: 3559
```

```
## 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) | 67.224931 | 86.6097468 | -0.2926484 | 0.560 |
| ## Block | 36.234157 | 33.0439840 | 2.6250965 | 0.026 |
| ## WaterLow | 71.190457 | 85.0930030 | 0.7408915 | 0.184 |
| ## Fungus | 2.241702 | 1.8664608 | 3.4234113 | 0.005 |
| ## Block:WaterLow | 40.431917 | 43.9953314 | 1.3804395 | 0.090 |
| ## Block:Fungus | 1.068939 | 0.9499183 | 2.8673697 | 0.016 |
| ## WaterLow:Fungus | 2.257521 | 2.4969515 | 1.1822329 | 0.129 |
| ## Block:WaterLow:Fungus | 1.285945 | 1.2558817 | 2.2174577 | 0.039 |

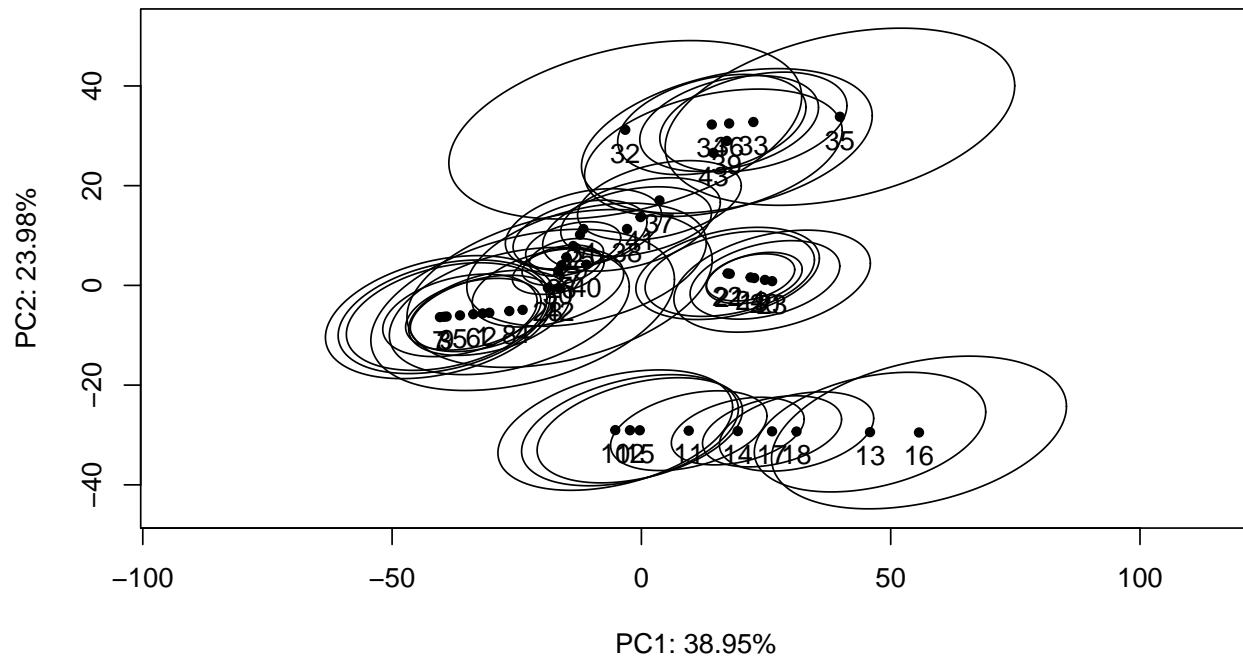
Block and Fungus have the largest effect on the model for old leaves, but not their interaction. The standard is the mean for High water treatment. For young leaves, Block, Fungus, Block:Fungus, and Block:WaterLow:Fungus have the largest effect on the model.

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

```
O_pred <- predict(O_LMpos)
```

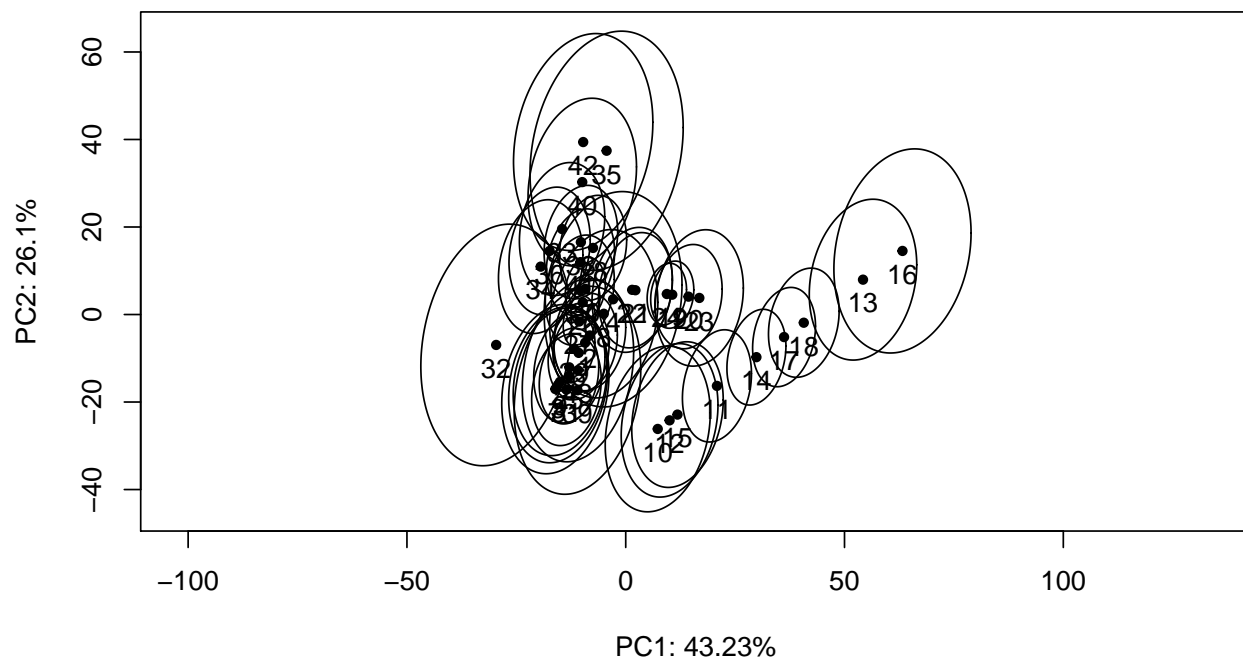
```
plot(O_pred, PC = T, ellipse = T)
```

Among-prediction PC rotation; 95% confidence limits



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

Among-prediction PC rotation; 95% confidence limits



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    22.190429 32.595628 -1.0363762 0.855
## 0:5    17.798195 27.382869 -0.6216817 0.685
## 0:15   24.519599 35.503011 -0.7383296 0.735
## 0:25   13.615879 23.329685 -1.7071115 0.989
## 0:32   16.139136 27.683841 -2.1297186 0.999
## 0:37   18.660876 32.009442 -2.1297186 0.999
## 0:52   29.207175 48.094881 -1.8235328 0.988
## 0:62   31.133050 53.016598 -2.2578357 1.000
## 3:5    17.786466 27.649058 -2.4409760 1.000
## 3:15   12.322880 20.954505 -1.8739528 0.998
## 3:25   22.264469 32.751009 -1.8759385 0.994
## 3:32   24.176171 36.230015 -1.7347681 0.986
## 3:37   25.415180 38.428930 -1.9047376 0.994
## 3:52   30.040035 46.896491 -2.2945570 1.000
## 3:62   35.208499 55.712597 -2.1353313 0.999
## 5:15   12.706131 19.039764 -1.6275339 0.978
## 5:25   15.112813 23.916840 -1.5967148 0.978
## 5:32   19.099319 30.150213 -1.5400493 0.973
## 5:37   20.477870 32.715432 -1.7831798 0.989
## 5:52   27.654743 43.338550 -2.0241162 0.995
## 5:62   31.206660 50.890285 -2.0874857 0.996
## 15:25   19.500368 27.175373 -0.8392479 0.782
## 15:32   22.187833 31.320643 -0.8590205 0.784
## 15:37   22.854770 32.866767 -1.0275958 0.843
## 15:52   25.865764 38.786671 -1.6016917 0.979
## 15:62   31.688647 48.856164 -1.5866827 0.976
## 25:32    5.003006  7.918074 -1.5516400 0.966
## 25:37    6.857942 11.202856 -2.0733970 0.996
## 25:52   17.391686 26.832045 -1.6677330 0.961
## 25:62   19.775068 33.336462 -2.2183112 0.999
## 32:37    2.521740  4.325600 -2.1297186 0.999
## 32:52   14.379824 22.337418 -1.4140734 0.940
## 32:62   15.526671 26.660486 -2.2842883 0.999
## 37:52   12.363495 18.849090 -1.2765064 0.917
## 37:62   13.204385 22.539244 -2.2663387 0.999
## 52:62   11.324188 19.564326 -1.6802913 0.988

```

```

## 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    12.197518 19.548605 -2.21426609  1.000
## 0:5    12.541925 17.225178 -0.47283484  0.654
## 0:15   12.091486 19.987403 -2.36695288  1.000
## 0:25   10.411956 20.263874 -1.65552803  0.999
## 0:32   14.320542 26.602293 -1.45327257  0.979
## 0:37   16.558127 30.758901 -1.45327257  0.979
## 0:52   27.084454 44.050817 -1.07373045  0.876
## 0:62   28.135960 51.703487 -1.50946116  0.984
## 3:5     13.070454 20.318975 -1.87146016  0.999
## 3:15     7.883536 14.583644 -3.11617956  1.000
## 3:25    16.269685 27.130362 -1.49774738  0.970
## 3:32    18.616016 31.117053 -1.31890013  0.929
## 3:37    20.359490 34.288275 -1.30481590  0.932
## 3:52    28.906708 46.012589 -0.94479866  0.825
## 3:62    30.000546 52.384649 -1.37704057  0.960
## 5:15     9.303381 13.134942 -0.84737414  0.790
## 5:25    18.191487 24.199294  0.03175081  0.466
## 5:32    21.913465 29.918287 -0.13703125  0.515
## 5:37    23.826525 33.358293 -0.25533576  0.566
## 5:52    33.680353 45.827215 -0.12074749  0.509
## 5:62    33.745873 51.622886 -0.71636918  0.737
## 15:25   14.077529 20.419563 -0.73636931  0.748
## 15:32   16.806159 24.627849 -0.68351024  0.727
## 15:37   18.460367 27.436876 -0.72360147  0.741
## 15:52   26.902869 38.154578 -0.38098664  0.604
## 15:62   27.791185 44.895936 -1.01504849  0.825
## 25:32    4.657594  7.335702 -1.02140225  0.847
## 25:37    6.722654 11.152621 -1.14600448  0.894
## 25:52   17.852807 25.754944 -0.55041928  0.681
## 25:62   18.564809 32.818300 -1.42451606  0.965
## 32:37    2.237585  4.156608 -1.45327257  0.979
## 32:52   13.890120 19.957496 -0.51957622  0.659
## 32:62   14.160482 25.826805 -1.56487272  0.989
## 37:52   12.062830 16.549363 -0.31644760  0.583
## 37:62   12.050231 21.844793 -1.58476970  0.989
## 52:62    9.354537 16.054715 -1.53544059  0.997

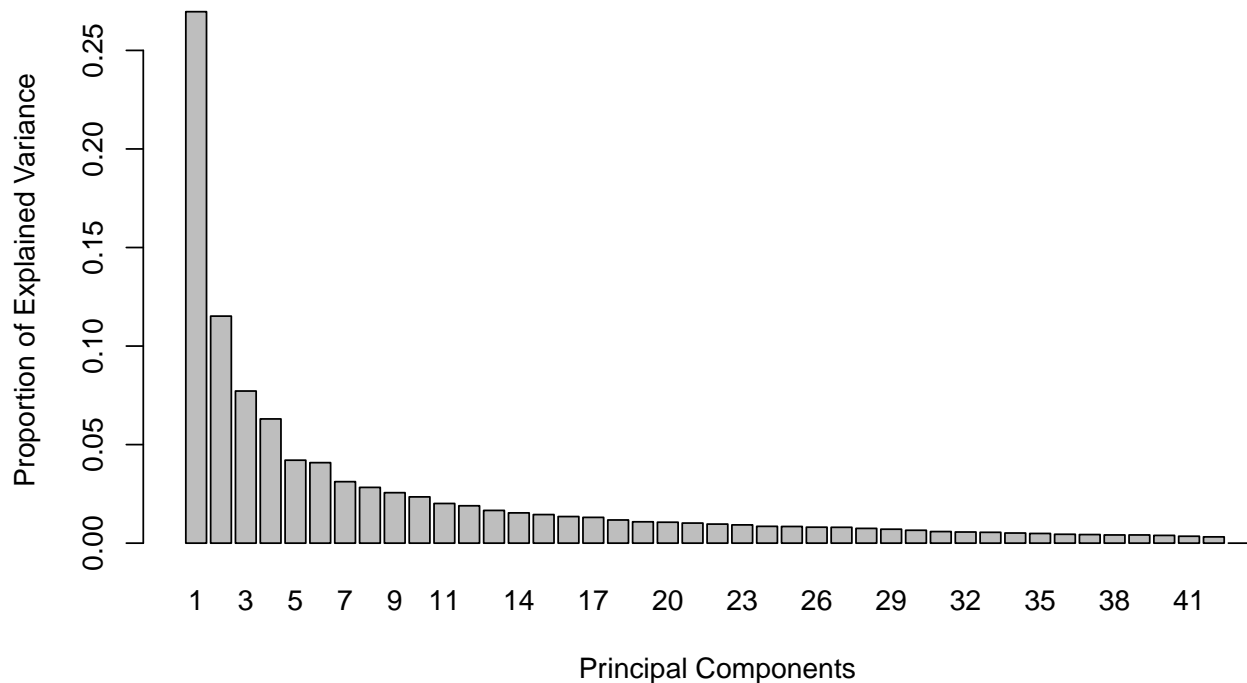
```


PCA

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

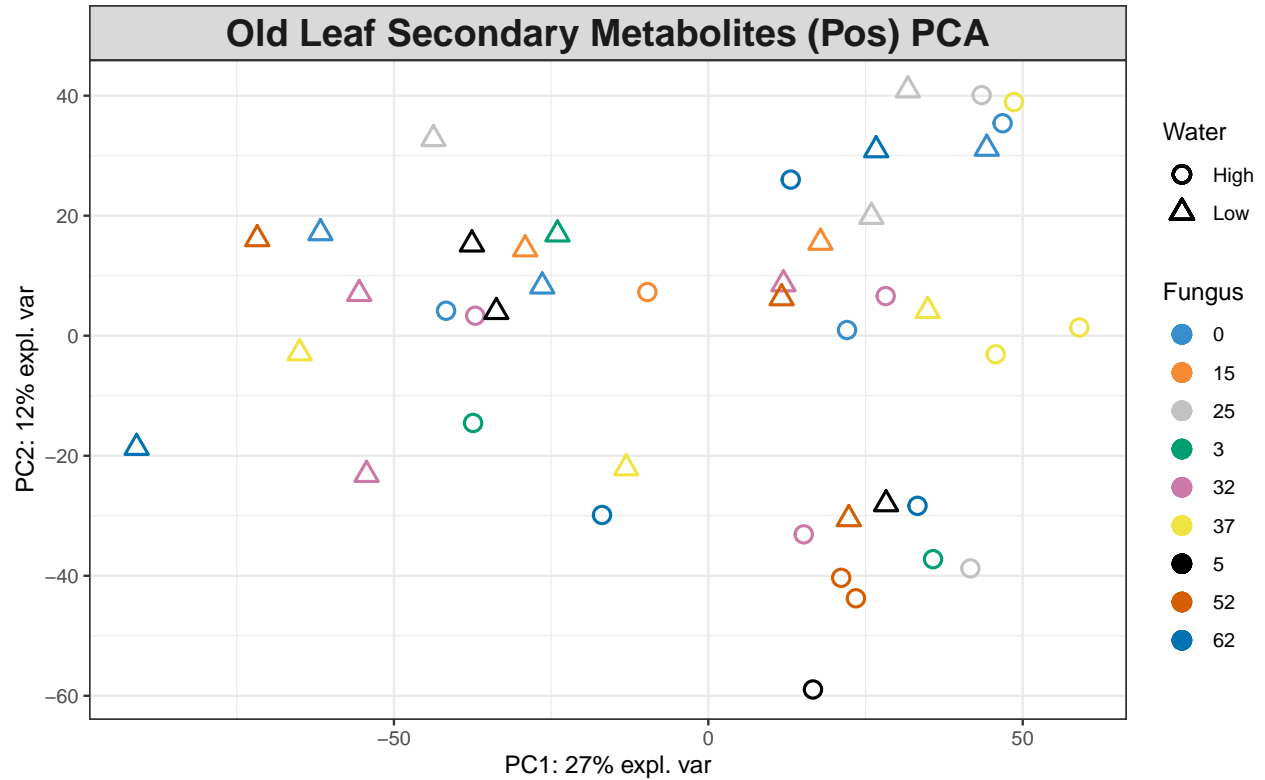
```
# Old Leaf Secondary Metabolites (Pos) tune how many  
# components to use  
tune.pca(scaled_Y_old)
```

```
## Eigenvalues for the first 10 principal components, see object$sdev^2:  
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8  
## 1563.9592 668.0796 447.6527 365.5013 244.1181 236.9154 180.9496 164.0427  
##      PC9      PC10  
## 148.5100 136.1229  
##  
## Proportion of explained variance for the first 10 principal components, see object$explained_variance:  
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7  
## 0.26964813 0.11518614 0.07718151 0.06301746 0.04208933 0.04084749 0.03119821  
##      PC8      PC9      PC10  
## 0.02828322 0.02560518 0.02346946  
##  
## Cumulative proportion explained variance for the first 10 principal components, see object$cum.var:  
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8  
## 0.2696481 0.3848343 0.4620158 0.5250332 0.5671226 0.6079701 0.6391683 0.6674515  
##      PC9      PC10  
## 0.6930567 0.7165261  
##  
## Other available components:  
## -----  
## loading vectors: see object$rotation
```



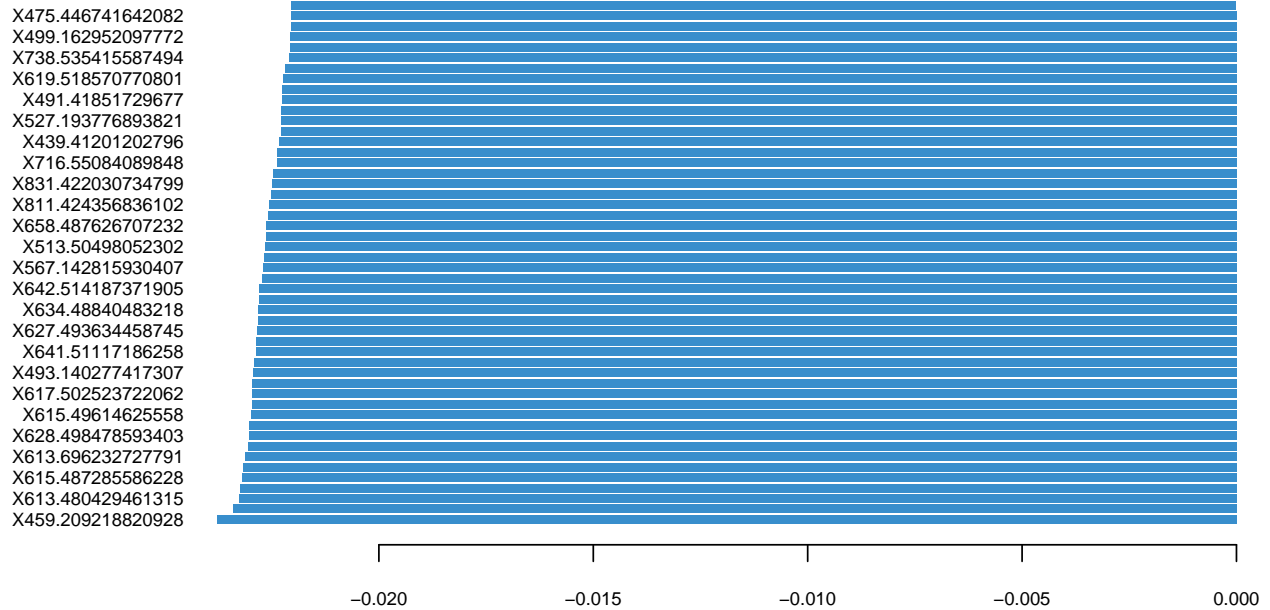
```
pca.res <- mixOmics::pca(scaled_Y_old, ncomp = 3, scale = F)
```

```
# 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 Secondary Metabolites (Pos) PCA")
```



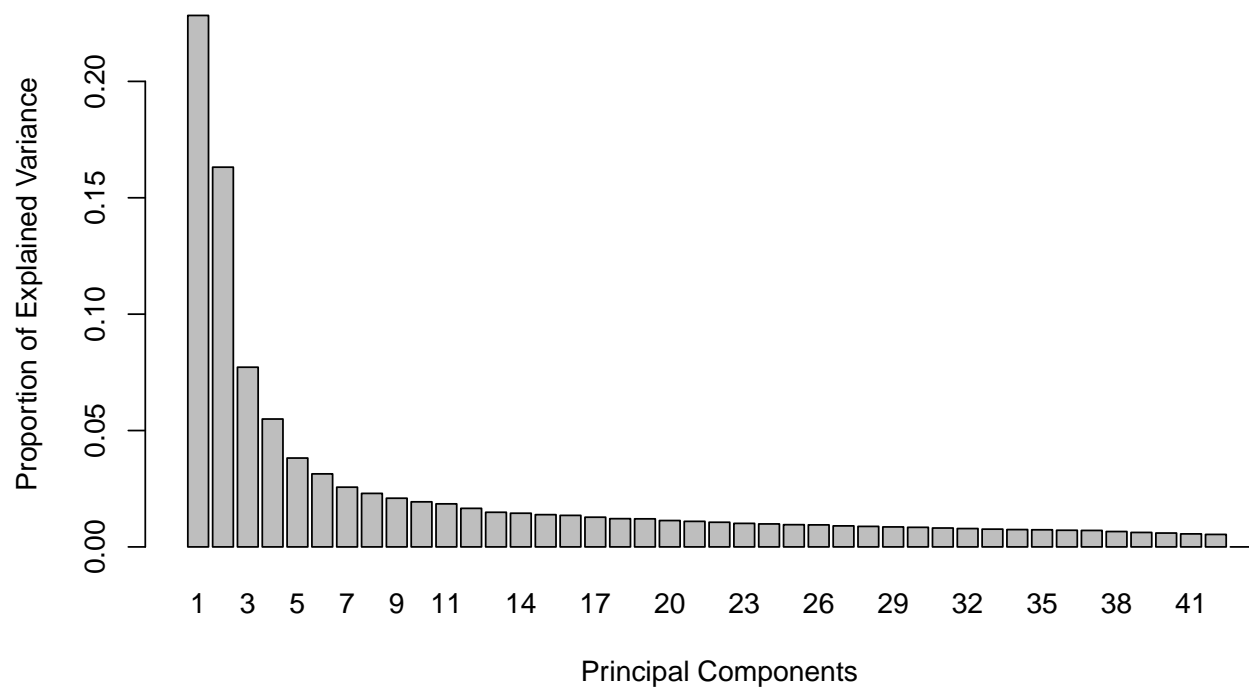
```
# Look at variable coefficients in each component with the
# loading vectors The absolute value of loading vectors
# represent the importance of each variable to define each PC
plotLoadings(pca.res, ndisplay = 50)
```

Loadings on comp 1



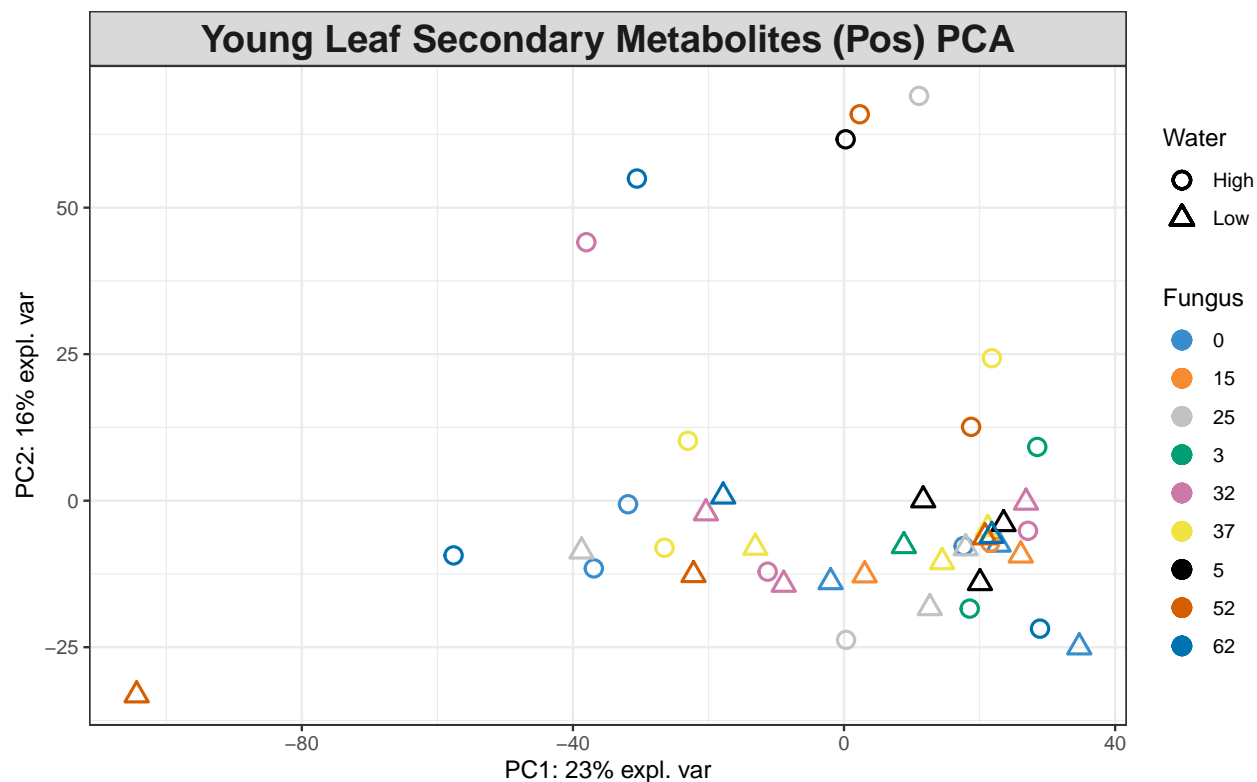
```
# Young Leaf Secondary Metabolites (Pos) tune how many
# components to use
tune.pca(scaled_Y_young)
```

```
## Eigenvalues for the first 10 principal components, see object$sdev^2:
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8
## 812.62108 580.64841 274.76111 195.54194 135.89003 111.66667  91.34267  81.80728
##      PC9      PC10
##  74.45598  68.92299
##
## Proportion of explained variance for the first 10 principal components, see object$explained_variance:
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7
## 0.22832849 0.16314931 0.07720177 0.05494294 0.03818208 0.03137586 0.02566526
##      PC8      PC9      PC10
## 0.02298603 0.02092048 0.01936583
##
## Cumulative proportion explained variance for the first 10 principal components, see object$cum.var:
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8
## 0.2283285 0.3914778 0.4686796 0.5236225 0.5618046 0.5931805 0.6188457 0.6418317
##      PC9      PC10
## 0.6627522 0.6821181
##
## Other available components:
## -----
## loading vectors: see object$rotation
```



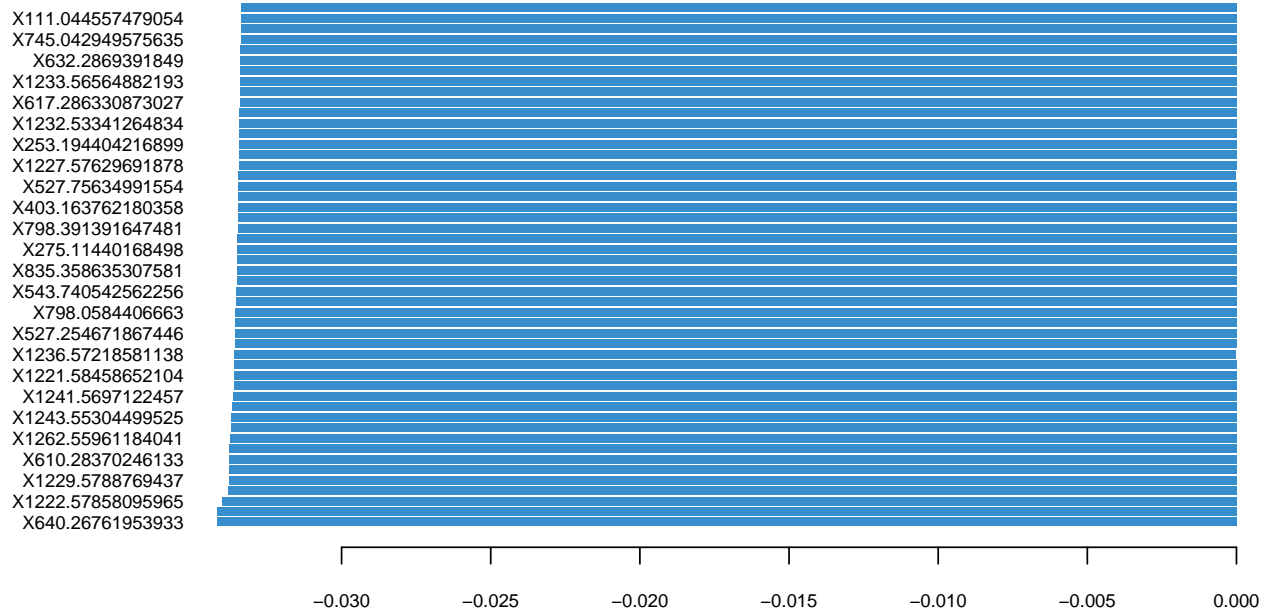
```
pca.res <- mixOmics::pca(scaled_Y_young, ncomp = 3, scale = F)

# 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 Secondary Metabolites (Pos) PCA")
```



```
# 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)
```

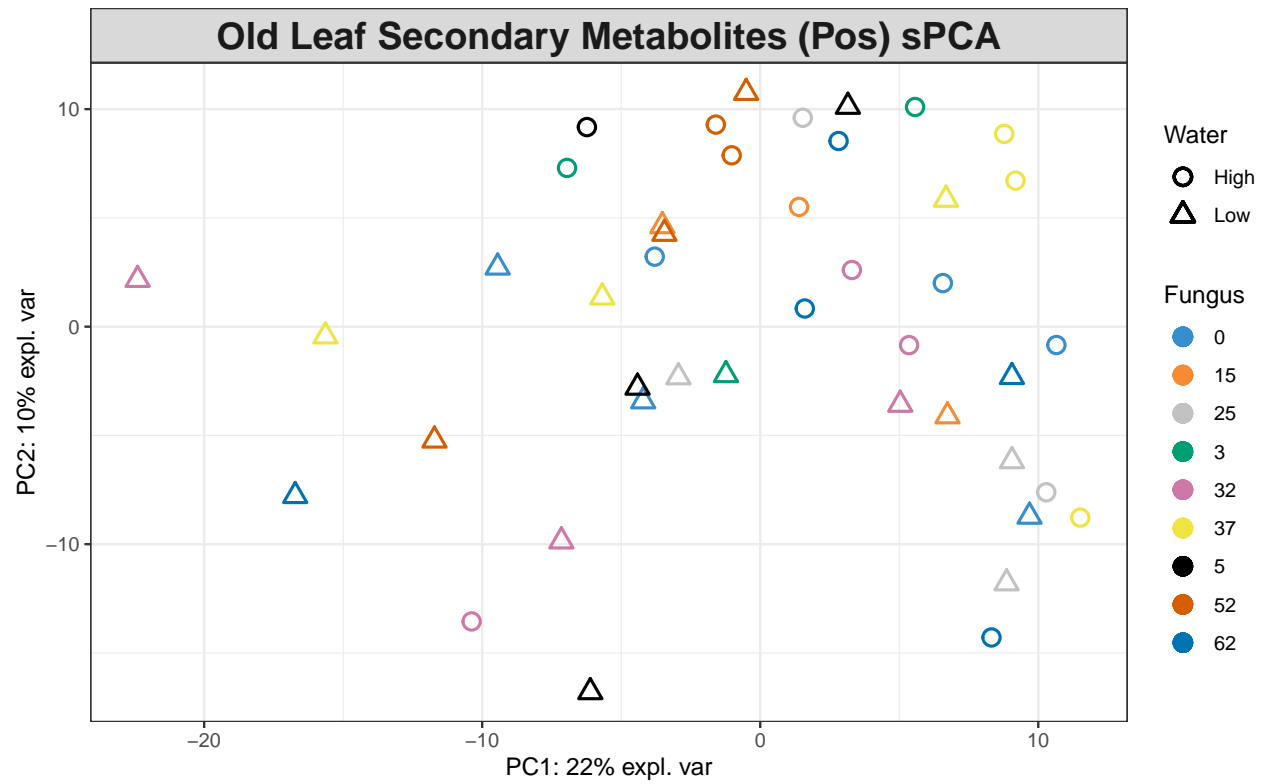
Loadings on comp 1



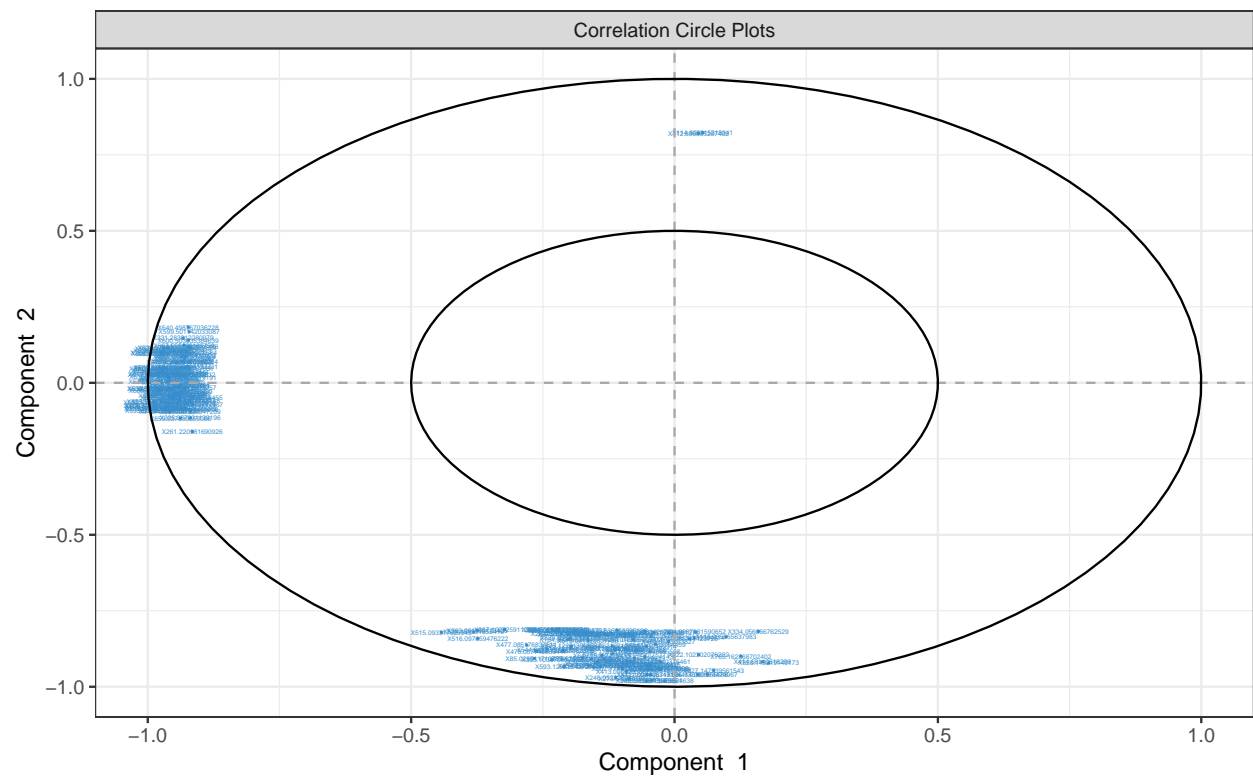
9. Identify the variables that contribute to the explained variance.

```
# 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 Secondary Metabolites (Pos) sPCA")
```



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



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

```
##          value.var
## X623.5427146501  -0.182055114
## X622.539887164198 -0.181404088
## X314.276975384773 -0.180391891
## X313.273601537502 -0.177913462
## X609.528418469459 -0.173659127
## X593.503695335635 -0.173111385
## X608.525125744233 -0.173085253
## X610.530931596464 -0.170019964
## X636.555782347412 -0.164596649
## X617.502523722062 -0.161488471
## X615.49614625558  -0.161372201
## X637.554974068755 -0.160689106
## X636.555044413109 -0.157801079
## X616.499669547884 -0.157311441
## X638.554170230362 -0.156500296
## X611.543535737333 -0.154368634
## X263.236566731675 -0.149120932
## X650.568768355466 -0.146359898
## X626.562565312764 -0.145730463
## X624.554998209121 -0.144372113
## X625.55795874975  -0.142472966
## X575.502966681956 -0.137889376
## X637.48032502828  -0.136679304
## X638.483173644822 -0.135515131
## X576.505762391681 -0.135267793
## X618.507103088589 -0.129459479
## X811.424356836102 -0.125604358
## X624.546393245245 -0.119384220
## X610.540587713529 -0.118351586
## X633.526384652885 -0.117956330
## X638.571010920895 -0.115276197
## X631.50710466344  -0.113748627
## X574.490589263477 -0.111560071
## X601.517812801478 -0.110208018
## X613.696232727791 -0.109824109
## X615.487285586228 -0.108647615
## X746.457453097686 -0.107262835
## X743.436513200601 -0.105806758
## X637.557509911664 -0.102124150
## X602.521531638131 -0.097096716
## X745.453691240662 -0.094310581
## X585.446721188158 -0.090565970
## X630.507793882762 -0.089309333
## X632.524036885204 -0.087580472
## X767.440469599277 -0.082429634
## X617.723248632496 -0.082259964
## X595.507447349485 -0.079387805
## X635.464542075429 -0.079367125
## X643.526311756535 -0.078010992
## X613.480429461315 -0.076824176
## X659.48872253972  -0.076643589
```

```

## X577.506622179924 -0.075085714
## X614.48413993025 -0.071693614
## X645.515984064549 -0.071351379
## X651.564511662028 -0.069707813
## X636.467534222444 -0.069163816
## X626.570188705153 -0.065764285
## X573.487260848141 -0.065535248
## X829.406046163401 -0.062453335
## X658.538868443561 -0.060418449
## X580.492441620954 -0.059736350
## X641.51117186258 -0.058555426
## X642.514187371905 -0.058042934
## X659.537339659066 -0.058033262
## X768.443158772313 -0.055913361
## X598.456409633744 -0.055656048
## X664.585958751482 -0.053832451
## X652.58279371971 -0.053087021
## X643.514428788009 -0.052008040
## X673.509123378608 -0.051285990
## X331.283812280979 -0.045554142
## X644.529496767651 -0.045034037
## X657.484442714608 -0.043683074
## X667.525186185759 -0.042049347
## X660.503817215267 -0.038786678
## X633.484726945181 -0.037949634
## X658.487626707232 -0.037612021
## X744.439887639683 -0.036573902
## X663.551426151608 -0.035986321
## X665.510410370928 -0.035924141
## X665.586601431923 -0.034656733
## X644.514172333257 -0.033719869
## X666.546022519191 -0.032138035
## X662.551874509279 -0.031995479
## X639.565404224584 -0.029863137
## X620.52328522026 -0.025732640
## X668.53044299629 -0.025132634
## X645.530094137731 -0.024834541
## X640.498767036228 -0.022619778
## X600.503635384639 -0.021720501
## X599.501142033087 -0.018378543
## X638.57096397224 -0.017188350
## X617.511254268748 -0.016288781
## X634.48840483218 -0.015300095
## X603.53233885458 -0.014696303
## X335.257931139196 -0.013030594
## X635.487834947209 -0.012811617
## X336.261554038455 -0.004335658
## X261.220981690926 -0.003217427
## X262.224903017467 -0.001379185

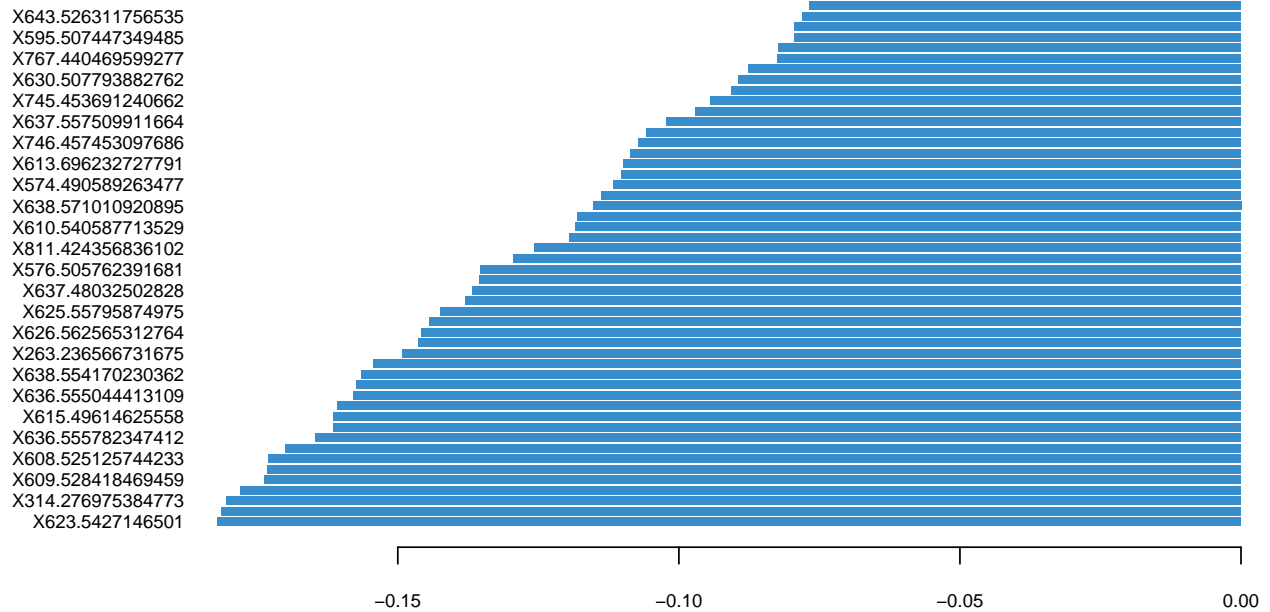
```

```

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

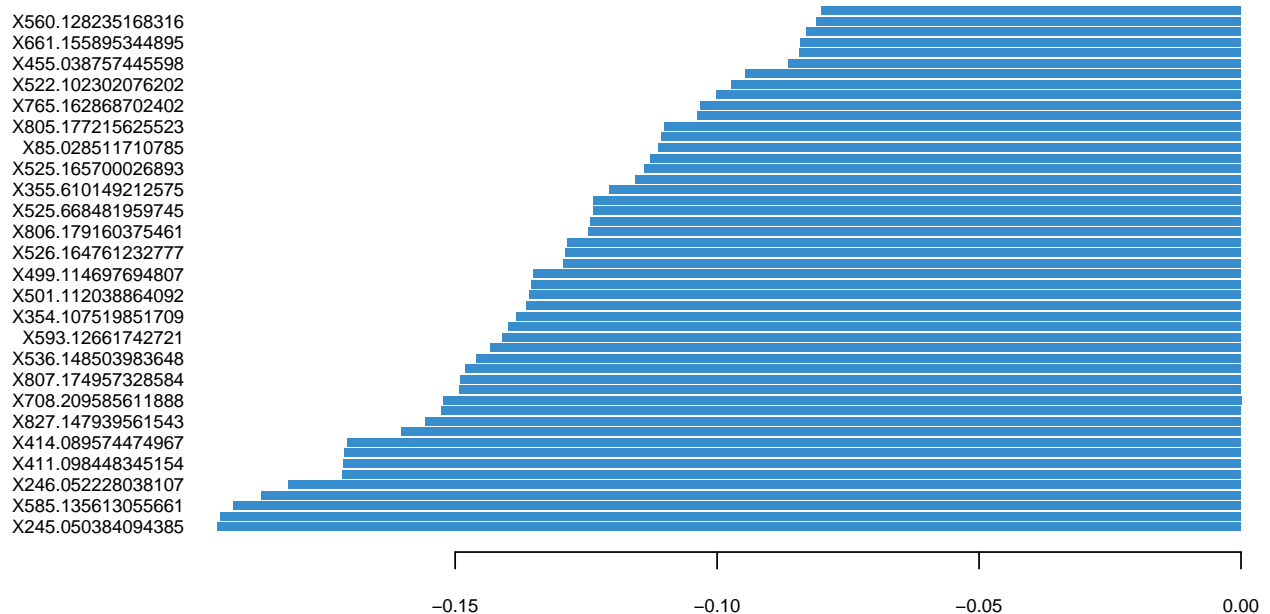
```


Loadings on comp 1



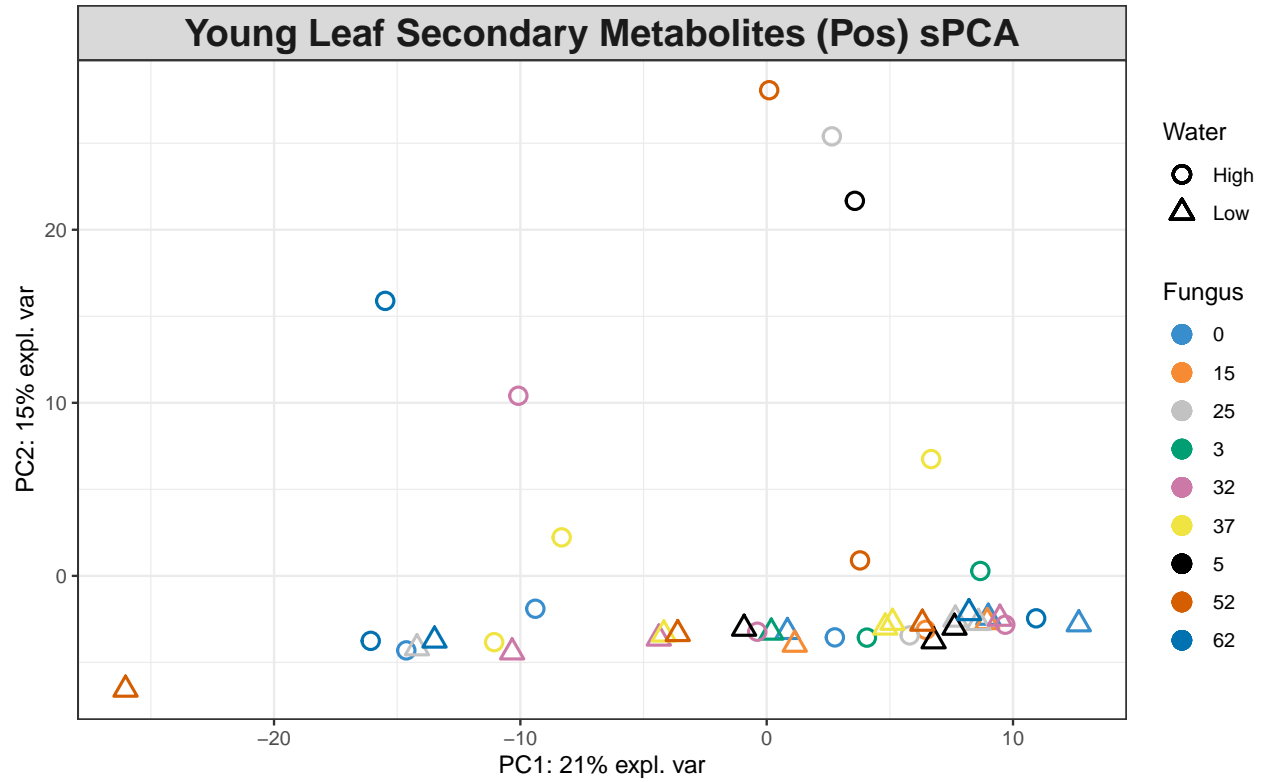
```
# plot loadings for comp 2
plotLoadings(sPCA.res, comp = 2, ndisplay = 50)
```

Loadings on comp 2

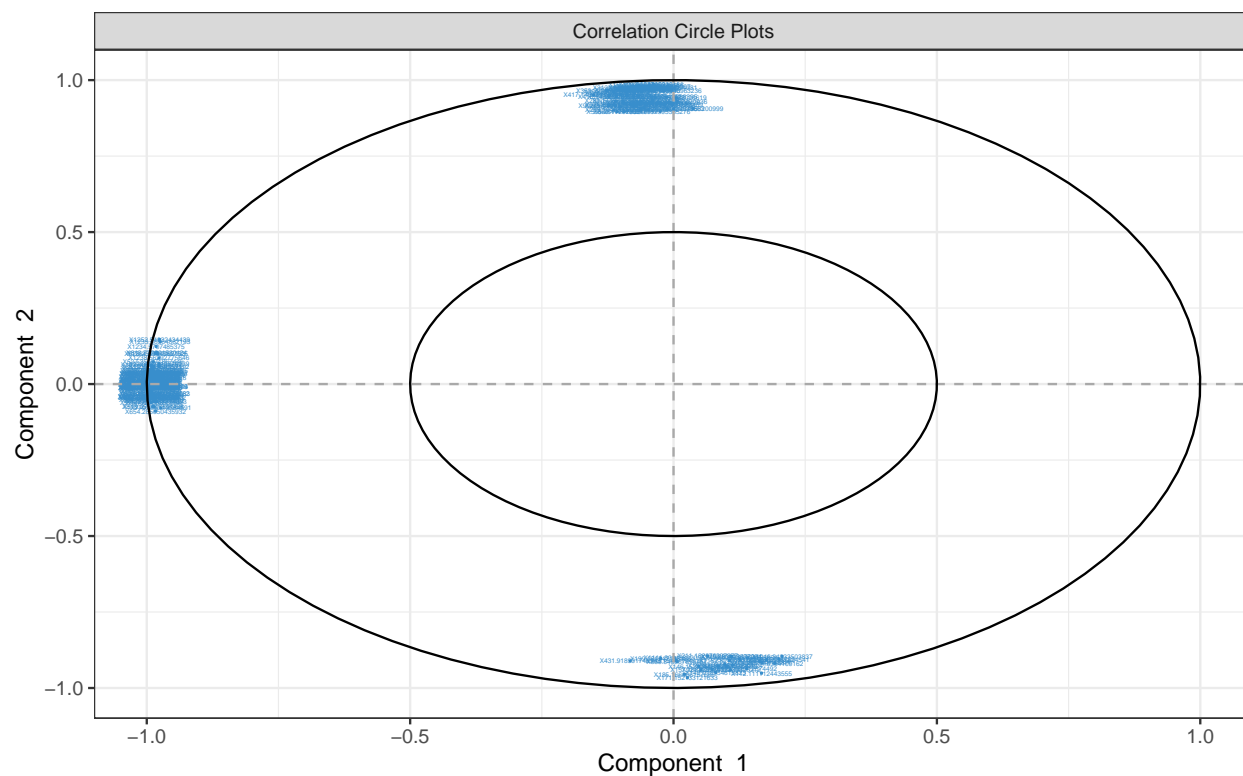


```
# Young Leaf
sPCA.res <- mixOmics::sPCA(scaled_Y_young, ncomp = 3, keepX = c(100,
  100, 10))
```

```
# 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 Secondary Metabolites (Pos) sPCA")
```



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



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

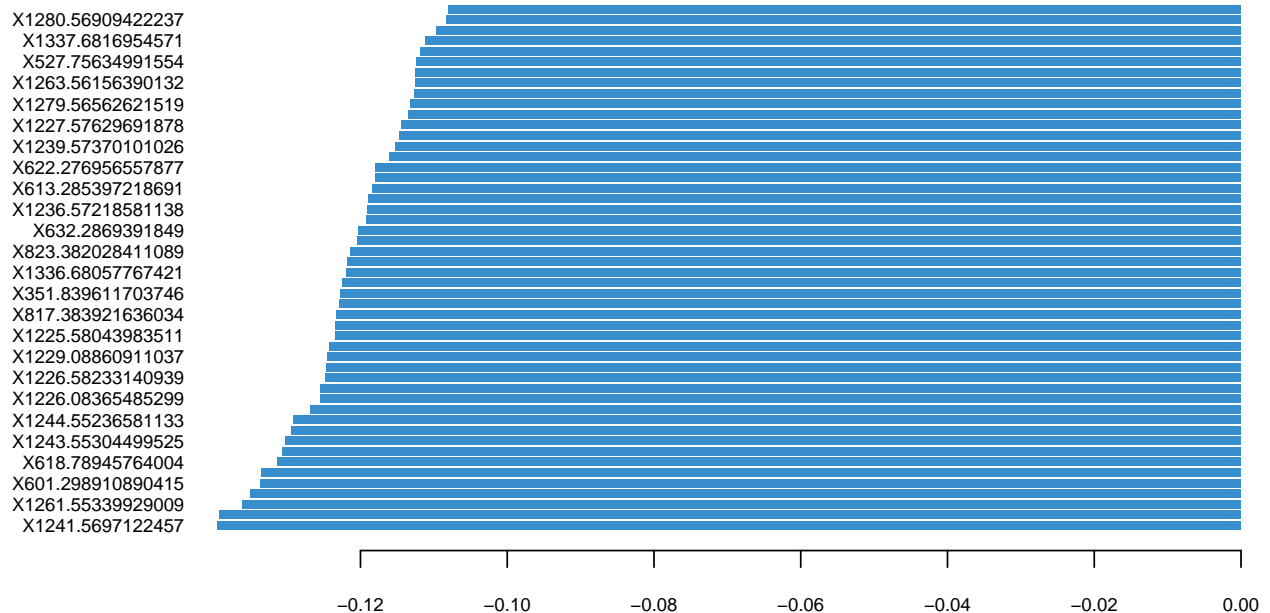
```
##          value.var
## X1241.5697122457 -0.139470661
## X600.298086140669 -0.139175646
## X1261.55339929009 -0.136132665
## X600.799613678288 -0.134941320
## X601.298910890415 -0.133563771
## X1227.08434351864 -0.133521416
## X618.78945764004 -0.131283246
## X823.048160007084 -0.130587035
## X1243.55304499525 -0.130195212
## X1262.55961184041 -0.129412563
## X1244.55236581133 -0.129116682
## X617.286330873027 -0.126845950
## X1226.08365485299 -0.125392084
## X1334.67495005457 -0.125382708
## X1226.58233140939 -0.124832612
## X817.723367787991 -0.124618167
## X1229.08860911037 -0.124494169
## X618.289313527896 -0.124274416
## X1225.58043983511 -0.123439455
## X822.714478304939 -0.123348570
## X817.383921636034 -0.123223780
## X617.78775135641 -0.122810308
## X351.839611703746 -0.122681247
## X609.78630978679 -0.122479407
## X1336.68057767421 -0.121862820
## X1228.58307046331 -0.121824091
```

X823.382028411089 -0.121349640
X1335.67840490914 -0.120383444
X632.2869391849 -0.120231642
X609.284289758841 -0.119187887
X1236.57218581138 -0.119006491
X405.857226571577 -0.118923159
X613.285397218691 -0.118398301
X1281.57026969661 -0.117923790
X622.276956557877 -0.117908484
X1240.57686400951 -0.116112366
X1239.57370101026 -0.115179222
X550.256803920959 -0.114748723
X1227.57629691878 -0.114347942
X823.71615864327 -0.113434602
X1279.56562621519 -0.113192877
X608.782741019481 -0.112651258
X1263.56156390132 -0.112569764
X1234.06478512482 -0.112525354
X527.75634991554 -0.112363717
X519.772768355994 -0.111812009
X1337.6816954571 -0.111093133
X608.28119716139 -0.109601512
X1280.56909422237 -0.108275587
X813.047644882569 -0.108059038
X406.192592684012 -0.107786086
X631.785508659045 -0.106529676
X817.043002326106 -0.105118952
X1235.06677264887 -0.104034681
X611.287897439172 -0.103000870
X824.049066021535 -0.102789556
X631.284156543571 -0.102303243
X352.174656446785 -0.101512752
X1054.50423500233 -0.099237844
X601.799037963433 -0.098831516
X1242.56522633851 -0.096006929
X1228.07642334574 -0.094508987
X613.035765932759 -0.093906496
X818.056781338958 -0.091141400
X612.785606728103 -0.087550336
X651.781625860394 -0.085040210
X527.254671867446 -0.084059004
X1237.57084462266 -0.081084178
X1230.08774541731 -0.079219075
X519.270307691408 -0.078746834
X816.707450462089 -0.078302171
X651.28036531701 -0.076416993
X632.786594301567 -0.074710660
X813.376589649202 -0.073640219
X520.264679977611 -0.073469371
X550.759359335189 -0.069502467
X1229.5788769437 -0.068918200
X528.257744262133 -0.064983429
X1283.55588708327 -0.060901976
X654.285650435932 -0.060037126

```
## X613.536347716772 -0.057915722
## X1234.5707485375 -0.051899728
## X838.372738715693 -0.049536575
## X613.772591530124 -0.045920191
## X535.247917849353 -0.044255968
## X1037.52045054275 -0.040974710
## X1305.53707639549 -0.039625742
## X1053.50107517804 -0.037969267
## X1264.55658857625 -0.037667817
## X1222.57858095965 -0.036063657
## X1237.0724873078 -0.034444605
## X610.28370246133 -0.027216591
## X1238.56292775646 -0.022181468
## X1224.56843575939 -0.021649724
## X812.714817179162 -0.020333869
## X1231.52739997192 -0.020296654
## X1252.54032434439 -0.015880601
## X621.789191807783 -0.015879632
## X1233.56564882193 -0.009690989
## X779.416119584591 -0.001035685
```

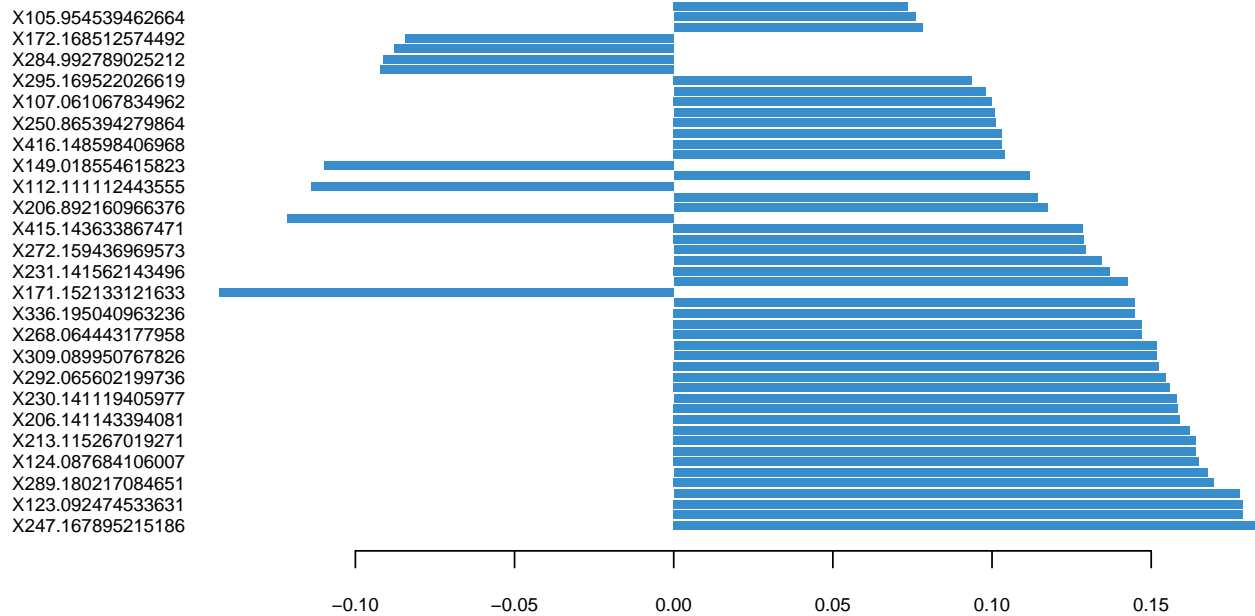
```
# plot loadings for comp 1
plotLoadings(spca.res, ndisplay = 50)
```

Loadings on comp 1



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

Loadings on comp 2



PLS-DA

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

```
# 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 Secondary Metabolites (Pos) PLS-DA",
legend.title = "Fungus", ellipse = T)
```



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

```
## $name
## [1] "X1487.73406300404" "X1426.07694312396" "X1116.05305660839"
## [4] "X1115.80460626783" "X1488.06760498806" "X1136.78228410413"
## [7] "X1104.06862691242" "X1116.30428005847" "X1103.8218996643"
## [10] "X1103.57319985526" "X1121.79675101528" "X1103.07299044082"
## [13] "X1111.8096321385" "X1125.54778244133" "X1102.573154447"
## [16] "X1116.80373735132" "X1132.03461103474" "X1122.29841750066"
## [19] "X1127.79341899364" "X1129.54675650191" "X1136.53403742727"
## [22] "X1131.28698128425" "X1108.81841229658" "X1110.06286552456"
## [25] "X1114.56442916912" "X1107.8169646921" "X1138.28303782195"
## [28] "X1102.82284449259" "X1426.40983069566" "X1122.04758243004"
## [31] "X1117.30475357162" "X1115.05543333062" "X1425.73993768413"
## [34] "X1132.53908246651" "X1109.81461712057" "X1133.54008365613"
## [37] "X1121.29895737339" "X1112.5604640628" "X1121.54819054951"
## [40] "X1125.79378838971" "X1126.29158820081" "X1108.06747930453"
## [43] "X1102.32311155925" "X1122.5497312715" "X1126.5441441766"
## [46] "X1470.09319447942" "X1123.30031266495" "X1112.81003971275"
## [49] "X1103.32325021034" "X1129.29565171194" "X1112.30949170708"
## [52] "X1108.31811054051" "X1488.40051943368" "X1471.42767786981"
## [55] "X1117.04932871798" "X1469.75976780245" "X1470.42704875471"
## [58] "X1127.04008421696" "X1131.78505133898" "X1470.76081907912"
## [61] "X1113.56329850353" "X1098.56831395271" "X1469.42570146433"
## [64] "X1135.53727745968" "X1112.05929178221" "X1113.31183863808"
## [67] "X1108.56822391305" "X1131.03795724898" "X1471.09532086136"
## [70] "X1127.29255917483" "X1114.06358094469" "X1129.79457145018"
## [73] "X1109.06854325948" "X1125.30053856127" "X1485.74884256861"
## [76] "X1135.28675162713" "X1489.399551935" "X1478.08739359233"
## [79] "X1133.28887059062" "X1132.78820284397" "X1476.75330076931"
## [82] "X1113.06007098305" "X1131.53755678079" "X1109.56623050873"
## [85] "X1136.28695855805" "X1478.42215498284" "X1132.28893553294"
## [88] "X1124.55354253246" "X1117.80491567323" "X1118.30495999252"
## [91] "X1109.31799268981" "X1114.31354083018" "X1118.05504416426"
## [94] "X1477.4205825873" "X1113.81293085655" "X1483.41064725665"
## [97] "X1119.80974545454" "X1126.79208167921" "X1490.40376922173"
## [100] "X1119.30886967612"
##
## $value
## value.var
## X1487.73406300404 0.2778109819
## X1426.07694312396 0.2644217191
## X1116.05305660839 0.2261700396
## X1115.80460626783 0.2173284797
## X1488.06760498806 0.1896116468
## X1136.78228410413 0.1650091785
## X1104.06862691242 0.1623645986
## X1116.30428005847 0.1580176449
## X1103.8218996643 0.1540609727
## X1103.57319985526 0.1512505345
## X1121.79675101528 0.1470020181
## X1103.07299044082 0.1437882774
## X1111.8096321385 0.1436423860
## X1125.54778244133 0.1379262355
```


X1102.573154447 0.1333348419
X1116.80373735132 0.1311062999
X1132.03461103474 0.1268407388
X1122.29841750066 0.1249396606
X1127.79341899364 0.1247654543
X1129.54675650191 0.1244549039
X1136.53403742727 0.1223155601
X1131.28698128425 0.1212771304
X1108.81841229658 0.1212334363
X1110.06286552456 0.1201315088
X1114.56442916912 0.1181915572
X1107.8169646921 0.1167675478
X1138.28303782195 0.1165180686
X1102.82284449259 0.1156617829
X1426.40983069566 0.1149926752
X1122.04758243004 0.1146371796
X1117.30475357162 0.1121319305
X1115.05543333062 0.1087284570
X1425.73993768413 0.1086871253
X1132.53908246651 0.1080661707
X1109.81461712057 0.1059478384
X1133.54008365613 0.1031746286
X1121.29895737339 0.1015076025
X1112.5604640628 0.0945782327
X1121.54819054951 0.0914253886
X1125.79378838971 0.0909061512
X1126.29158820081 0.0889916026
X1108.06747930453 0.0879526826
X1102.32311155925 0.0878277012
X1122.5497312715 0.0870585837
X1126.5441441766 0.0860669559
X1470.09319447942 0.0851941163
X1123.30031266495 0.0849895564
X1112.81003971275 0.0828129627
X1103.32325021034 0.0804834482
X1129.29565171194 0.0792222380
X1112.30949170708 0.0755750879
X1108.31811054051 0.0741827212
X1488.40051943368 0.0738000496
X1471.42767786981 0.0705795509
X1117.04932871798 0.0680631925
X1469.75976780245 0.0677773708
X1470.42704875471 0.0670220233
X1127.04008421696 0.0617318264
X1131.78505133898 0.0614552161
X1470.76081907912 0.0607021936
X1113.56329850353 0.0575110057
X1098.56831395271 0.0552270918
X1469.42570146433 0.0547297316
X1135.53727745968 0.0525023044
X1112.05929178221 0.0499647663
X1113.31183863808 0.0499299039
X1108.56822391305 0.0484990599
X1131.03795724898 0.0467007621

```

## X1471.09532086136 0.0465308757
## X1127.29255917483 0.0465304054
## X1114.06358094469 0.0430392118
## X1129.79457145018 0.0426737863
## X1109.06854325948 0.0425454117
## X1125.30053856127 0.0384018869
## X1485.74884256861 0.0361639550
## X1135.28675162713 0.0355602328
## X1489.399551935 0.0355471964
## X1478.08739359233 0.0338542992
## X1133.28887059062 0.0332983021
## X1132.78820284397 0.0314452831
## X1476.75330076931 0.0312544730
## X1113.06007098305 0.0307244558
## X1131.53755678079 0.0306259262
## X1109.56623050873 0.0297866618
## X1136.28695855805 0.0275403531
## X1478.42215498284 0.0271748784
## X1132.28893553294 0.0270411862
## X1124.55354253246 0.0221139805
## X1117.80491567323 0.0220497801
## X1118.30495999252 0.0208404964
## X1109.31799268981 0.0198891505
## X1114.31354083018 0.0182593046
## X1118.05504416426 0.0178664769
## X1477.4205825873 0.0169645660
## X1113.81293085655 0.0158796336
## X1483.41064725665 0.0132492924
## X1119.80974545454 0.0098193383
## X1126.79208167921 0.0071816142
## X1490.40376922173 0.0020508425
## X1119.30886967612 0.0006404074
##
## $comp
## [1] 1

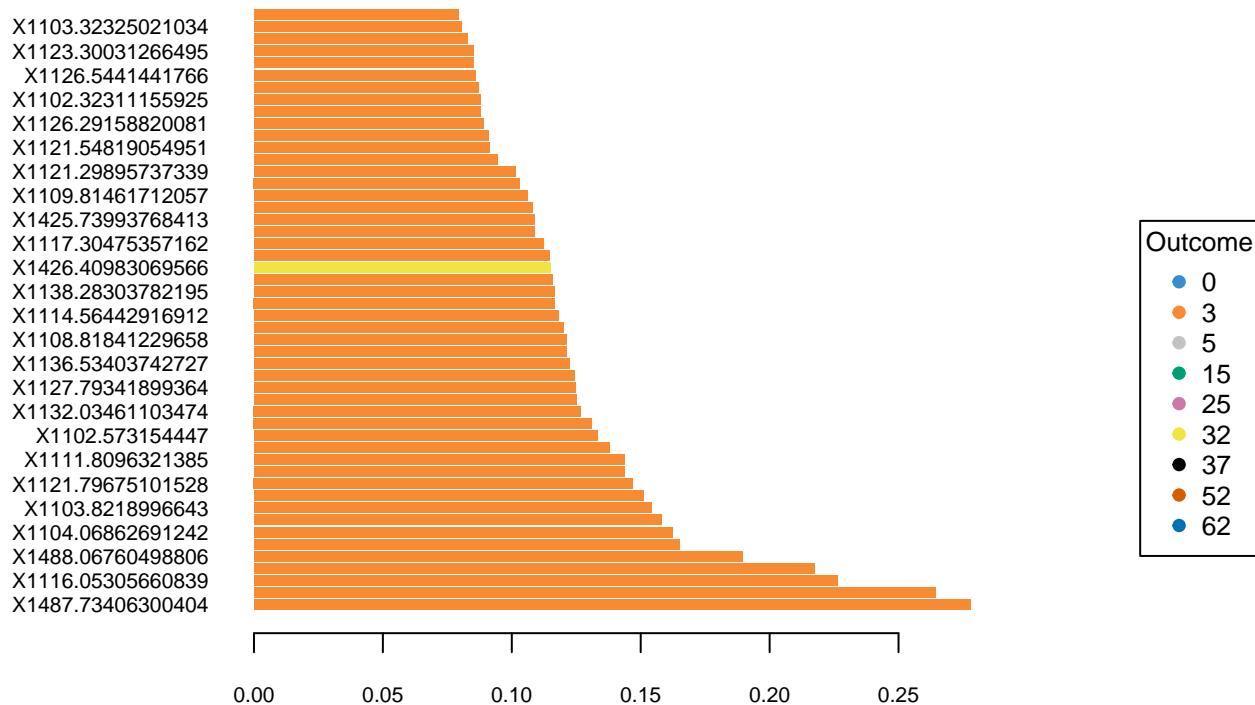
```

```

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

```

Contribution on comp 1



```
# 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 Secondary Metabolites (Pos) PLS-1",
  legend.title = "Fungus", ellipse = T)
```



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

```
## $name
## [1] "X435.265041264193" "X561.195153908566" "X191.142831050821"
## [4] "X562.198826698276" "X220.933637304516" "X411.186748289517"
## [7] "X433.024089572141" "X761.150139232025" "X417.049118108137"
## [10] "X79.94287983744" "X201.887880425105" "X542.258263036671"
## [13] "X562.199692173047" "X561.195521367934" "X592.2103604043"
## [16] "X529.274978890623" "X622.22112991077" "X817.290953139684"
## [19] "X273.167111911126" "X393.190249243473" "X229.141721419357"
## [22] "X559.216971541369" "X576.211656573802" "X787.279327745226"
## [25] "X137.061136253662" "X547.21588517482" "X147.042739890915"
## [28] "X759.162827899809" "X419.041079838955" "X338.342115928924"
## [31] "X608.783170750181" "X697.33374931113" "X559.181816091216"
## [34] "X609.188818880262" "X598.248439099125" "X546.204470010442"
## [37] "X361.091398476515" "X431.134225468165" "X214.253550469892"
## [40] "X608.280771730718" "X545.200300816215" "X548.219043560918"
## [43] "X559.308892423438" "X679.276723243348" "X327.160705477326"
## [46] "X126.965854212842" "X627.205841547367" "X387.163549203145"
## [49] "X591.206314720655" "X349.090515445971" "X678.27479236805"
## [52] "X149.095689289718" "X219.102286597348" "X579.103775337199"
## [55] "X295.087943490825" "X381.116290473037" "X396.136174296936"
## [58] "X769.269913578792" "X368.42562933552" "X281.589541788573"
## [61] "X771.283948428587" "X611.208506435124" "X607.209814001622"
## [64] "X575.208493952074" "X459.153301042731" "X549.219252720572"
## [67] "X282.095747416245" "X628.210632485529" "X547.216368593675"
## [70] "X159.101695076237" "X655.189885840188" "X395.132504275827"
## [73] "X330.368285312136" "X603.206683298622" "X402.358358777433"
## [76] "X430.390022741156" "X595.222226186161" "X548.220348742993"
## [79] "X157.044063854019" "X608.21522833073" "X179.070702928982"
## [82] "X723.248094849444" "X375.14623697476" "X707.430538895667"
## [85] "X540.245574294723" "X604.209992086111" "X593.221770649434"
## [88] "X346.111503639577" "X374.303001032379" "X851.311747586913"
## [91] "X498.217452455795" "X621.216851121727" "X500.210549877182"
## [94] "X594.2254897838" "X435.022999713606" "X501.138397567176"
## [97] "X1379.64356208646" "X441.244799903401" "X435.128173248124"
## [100] "X401.068518225837"
##
## $value
## value.var
## X435.265041264193 0.258963679
## X561.195153908566 0.246358522
## X191.142831050821 0.238693289
## X562.198826698276 0.237963492
## X220.933637304516 0.237248512
## X411.186748289517 0.222867792
## X433.024089572141 0.218606675
## X761.150139232025 0.212332555
## X417.049118108137 0.209195273
## X79.94287983744 0.195117208
## X201.887880425105 0.171412375
## X542.258263036671 0.163553226
## X562.199692173047 0.154299165
## X561.195521367934 0.152634222
```

```

## X592.2103604043    0.147199177
## X529.274978890623  0.141830874
## X622.22112991077   0.137377777
## X817.290953139684  0.134882524
## X273.167111911126  0.129870163
## X393.190249243473  0.120262756
## X229.141721419357 -0.120047587
## X559.216971541369  0.114792705
## X576.211656573802  0.106310189
## X787.279327745226  0.105770084
## X137.061136253662  0.104912721
## X547.21588517482   0.104665651
## X147.042739890915  0.104086004
## X759.162827899809  0.103561657
## X419.041079838955  0.098130782
## X338.342115928924  0.091404048
## X608.783170750181  0.090990343
## X697.33374931113   0.089838998
## X559.181816091216  0.082511640
## X609.188818880262  0.081541900
## X598.248439099125  0.081463692
## X546.204470010442  0.080079798
## X361.091398476515  0.078486608
## X431.134225468165  0.078440925
## X214.253550469892  0.076089624
## X608.280771730718  0.070627804
## X545.200300816215  0.063329224
## X548.219043560918  0.063225972
## X559.308892423438  0.062664063
## X679.276723243348  0.061463371
## X327.160705477326  0.058134774
## X126.965854212842  0.058067992
## X627.205841547367  0.056609462
## X387.163549203145  0.056225981
## X591.206314720655  0.055010844
## X349.090515445971  0.053976768
## X678.27479236805   0.053139991
## X149.095689289718  0.052029242
## X219.102286597348  0.051889353
## X579.103775337199  0.051834367
## X295.087943490825  0.050970519
## X381.116290473037  0.049560180
## X396.136174296936  0.048273374
## X769.269913578792  0.047746198
## X368.42562933552   -0.046740608
## X281.589541788573  0.045612957
## X771.283948428587  0.042215253
## X611.208506435124  0.041706940
## X607.209814001622  0.040487259
## X575.208493952074  0.040183163
## X459.153301042731  0.039493252
## X549.219252720572  0.038285218
## X282.095747416245  0.036745489
## X628.210632485529  0.036450168

```

```

## X547.216368593675 0.036022686
## X159.101695076237 0.033758763
## X655.189885840188 0.033010829
## X395.132504275827 0.032253564
## X330.368285312136 -0.031975536
## X603.206683298622 0.031005958
## X402.358358777433 0.030979833
## X430.390022741156 0.030323872
## X595.222226186161 0.029635867
## X548.220348742993 0.029628293
## X157.044063854019 0.028907790
## X608.21522833073 0.024580253
## X179.070702928982 0.024301928
## X723.248094849444 0.023929859
## X375.14623697476 0.020281866
## X707.430538895667 0.017609269
## X540.245574294723 0.016430810
## X604.209992086111 0.015782657
## X593.221770649434 0.014237139
## X346.111503639577 0.013165190
## X374.303001032379 0.011635550
## X851.311747586913 0.009264566
## X498.217452455795 0.008973775
## X621.216851121727 0.008646167
## X500.210549877182 0.007840429
## X594.2254897838 0.006764113
## X435.022999713606 0.005054371
## X501.138397567176 0.004307501
## X1379.64356208646 0.001845171
## X441.244799903401 0.001827014
## X435.128173248124 0.001329296
## X401.068518225837 0.001178817
##
## $comp
## [1] 1

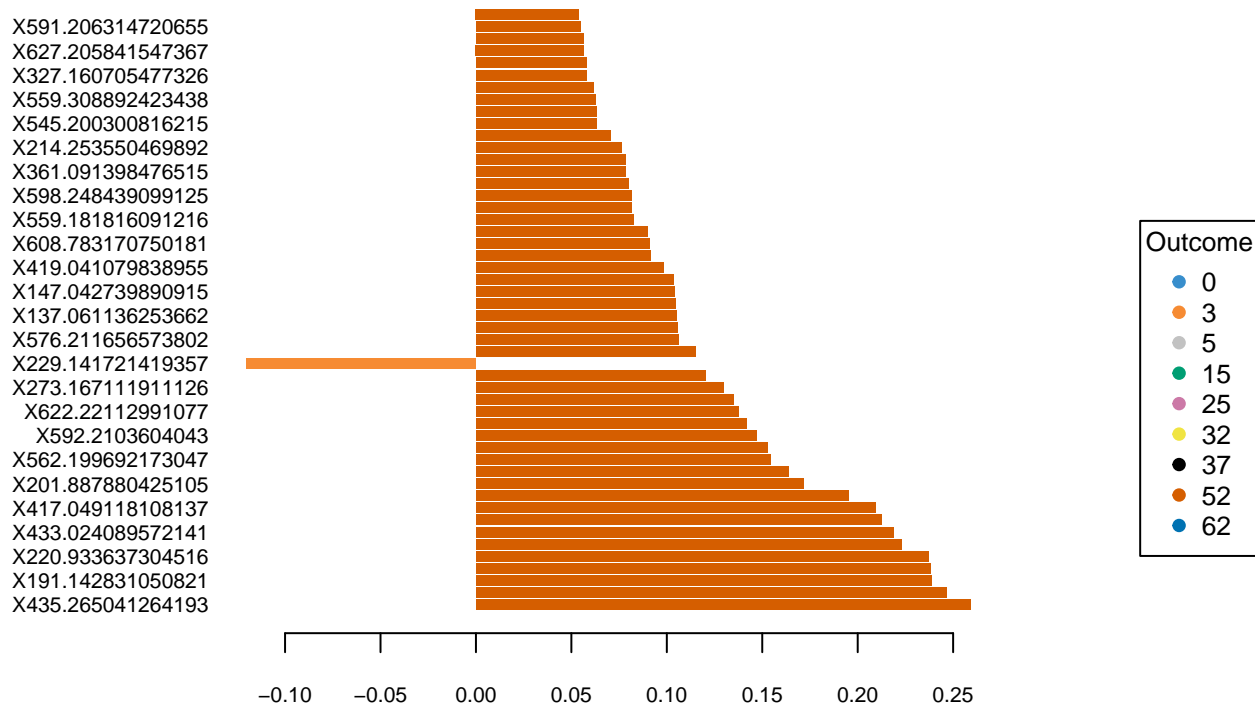
```

```

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

```

Contribution on comp 1



Heatmaps of Averaged Data

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

```
## Old Leaves
av_Y_old <- aggregate(Y_old, by = list(class$Water, class$Fungus),
  FUN = "mean", simplify = T, data = class)
av.old.plsda <- mixOmics::plsda(av_Y_old[, 3:5802], av_Y_old$Group.2) # fungus

# heatmap
oldcim <- cim(av.old.plsda, title = "Old Leaf Secondary Met. (pos) Averaged Over Fungi",
  col.names = F, xlab = "Secondary Metabolites", save = "png",
  name.save = "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/
```

Indicator Analysis

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

```
# Old Leaf
indicator_Fungus0 <- indval(Y_old, clustering = class$Fungus,
  numitr = 999, type = "long")
summary(indicator_Fungus0)
```

```
##
## Sum of probabilities          = 3271.37837837838
##
## Sum of Indicator Values      = 960.28
##
## Sum of Significant Indicator Values = 30.84
```



```
##
## Number of Significant Indicators      = 148
##
## Significant Indicator Distribution
##
## 2 3 4 5 8 9
## 95 4 41 5 1 2

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

| ## | cluster | indicator_value | probability |
|----|-------------------|-----------------|--------------------|
| ## | X738.296098922098 | 1 | 0.2081 0.011011011 |
| ## | X1276.8016935537 | 2 | 0.1851 0.047047047 |
| ## | X1072.8337334809 | 2 | 0.1787 0.039039039 |
| ## | X167.97551825434 | 2 | 0.1761 0.038038038 |
| ## | X1140.82165943663 | 2 | 0.1714 0.043043043 |
| ## | X444.930683630603 | 2 | 0.1659 0.001001001 |
| ## | X275.163918393716 | 2 | 0.1633 0.021021021 |
| ## | X156.890008237947 | 2 | 0.1499 0.021021021 |
| ## | X1153.79792571483 | 2 | 0.1407 0.029029029 |
| ## | X229.141721419357 | 2 | 0.1387 0.001001001 |
| ## | X540.818241281503 | 2 | 0.1363 0.034034034 |
| ## | X313.01815211212 | 2 | 0.1352 0.009009009 |
| ## | X1246.76817339089 | 2 | 0.1219 0.032032032 |
| ## | X113.96385752929 | 3 | 0.1700 0.012012012 |
| ## | X1180.43827513436 | 3 | 0.1656 0.007007007 |
| ## | X703.214675604816 | 3 | 0.1595 0.040040040 |
| ## | X633.151746090594 | 3 | 0.1557 0.045045045 |
| ## | X637.148466811858 | 3 | 0.1547 0.025025025 |
| ## | X702.214256834651 | 3 | 0.1543 0.023023023 |
| ## | X559.132251560287 | 3 | 0.1541 0.048048048 |
| ## | X131.962746648011 | 4 | 0.2466 0.014014014 |
| ## | X169.287975931043 | 4 | 0.1941 0.012012012 |
| ## | X174.170106921473 | 4 | 0.1834 0.002002002 |
| ## | X86.0969059629987 | 4 | 0.1785 0.017017017 |
| ## | X320.864934443148 | 4 | 0.1750 0.030030030 |
| ## | X173.164828416272 | 4 | 0.1626 0.006006006 |
| ## | X115.964583574333 | 4 | 0.1568 0.040040040 |
| ## | X1280.77652898027 | 4 | 0.1546 0.040040040 |
| ## | X228.215068132004 | 4 | 0.1502 0.029029029 |
| ## | X1357.7663209951 | 4 | 0.1499 0.010010010 |
| ## | X500.822798296509 | 4 | 0.1458 0.042042042 |
| ## | X305.908971541333 | 4 | 0.1431 0.048048048 |
| ## | X1315.76786868364 | 4 | 0.1405 0.041041041 |
| ## | X1017.81968016253 | 4 | 0.1383 0.005005005 |
| ## | X907.835107134673 | 4 | 0.1338 0.036036036 |
| ## | X219.214947899134 | 4 | 0.1249 0.038038038 |
| ## | X468.163826322867 | 5 | 0.1768 0.042042042 |
| ## | X344.876666294775 | 7 | 0.1436 0.046046046 |
| ## | X433.024089572141 | 8 | 0.1953 0.023023023 |
| ## | X220.933637304516 | 8 | 0.1605 0.033033033 |
| ## | X435.265041264193 | 8 | 0.1532 0.031031031 |

```
## X122.964222379764      9      0.2028 0.047047047
##
## Sum of probabilities      = 2143.79179179179
##
## Sum of Indicator Values  = 564.14
##
## Sum of Significant Indicator Values = 6.8
##
## Number of Significant Indicators   = 42
##
## Significant Indicator Distribution
##
## 1 2 3 4 5 7 8 9
## 1 12 7 16 1 1 3 1
```

13. Disect indval object.

```
Orelfrq <- indicator_Fungus0$relfrq # relative frequency of species in classes
Orelabu <- indicator_Fungus0$relabu  # relative abundance of species in classes
Oindval <- indicator_Fungus0$indval  # the indicator value for each species
Omaxcls <- data.frame(indicator_Fungus0$maxcls) # the class each species has max indicator value for
Oindcls <- data.frame(indicator_Fungus0$indcls) # the indicator value for each species to its max class
Opval <- data.frame(indicator_Fungus0$pval) # the probability of obtaining as high an indicator value as

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 as
```

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/Second",
write.csv(cbind(Yrelfrq, Yrelabu, Yindval, Ymaxcls, Yindcls,
  Ypval), "~/Box/Summer 2018 TX Endo Field Samples and Analysis/Statistics/Kenia_Thesis_Analysis/Second")
```

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.