Assignment 05

Erik Pak

2023-06-03

Import Libraries**

```
library(tidyverse)    # Data manipulation package
library(corrplot)     # correlation plot
library(psych)        # used for describe & pairs.panel plot
library(ggpubr)       # combine plots
library(MASS)         # "isoMDS"
# The CCA library has more extensive functionality
library(CCA)
```

Problem 1

1) (20 points) Download the "kellog.dat" data file which contains data on 22 cereals from Kellog. Each cereal has 9 metric values that measure various aspects of the cereal. We are not given the meanings of these variables, but in spite of this use the data to classify the cereals:

Problem 1 a)

```
# read data into dataframe
cereal <- read.table("kellog.dat", skip = 2, row.names = 1)

# display
head(cereal)
```

```
##                     V2  V3     V4     V5     V6     V7     V8  V9    V10 V11
## AllBran         0.1818 0.6 0.3333 0.8125 0.6429 0.0000 0.3333 1.0 0.9677   0
## AllBranFlakes   0.0000 0.6 0.0000 0.4375 1.0000 0.0667 0.0000 1.0 1.0000   0
## AppleJacks      0.5455 0.2 0.0000 0.3906 0.0714 0.2667 0.9333 0.5 0.0323   0
## CornFlakes      0.4545 0.2 0.0000 0.9063 0.0714 0.9333 0.1333 0.0 0.0484   0
## CorPops         0.5455 0.0 0.0000 0.2813 0.0714 0.4000 0.8000 0.5 0.0000   0
## CracklinOatBran 0.5455 0.4 1.0000 0.4375 0.2857 0.2000 0.4667 1.0 0.4516   0
```

```
# display
head(cereal)
```

```
##                     V2  V3     V4     V5     V6     V7     V8  V9    V10 V11
## AllBran         0.1818 0.6 0.3333 0.8125 0.6429 0.0000 0.3333 1.0 0.9677   0
## AllBranFlakes   0.0000 0.6 0.0000 0.4375 1.0000 0.0667 0.0000 1.0 1.0000   0
## AppleJacks      0.5455 0.2 0.0000 0.3906 0.0714 0.2667 0.9333 0.5 0.0323   0
## CornFlakes      0.4545 0.2 0.0000 0.9063 0.0714 0.9333 0.1333 0.0 0.0484   0
## CorPops         0.5455 0.0 0.0000 0.2813 0.0714 0.4000 0.8000 0.5 0.0000   0
## CracklinOatBran 0.5455 0.4 1.0000 0.4375 0.2857 0.2000 0.4667 1.0 0.4516   0
```

```
# summary
summary(cereal)
```

```
##       V2              V3              V4              V5
```

```
##   Min.   :0.0000    Min.   :0.0000    Min.    :0.0000    Min.    :0.0000
##   1st Qu.:0.4545    1st Qu.:0.2000    1st Qu.:0.0000    1st Qu.:0.4023
##   Median :0.5455    Median :0.3000    Median :0.0000    Median :0.5313
##   Mean   :0.5289    Mean   :0.3273    Mean    :0.2121    Mean    :0.5370
##   3rd Qu.:0.5455    3rd Qu.:0.4000    3rd Qu.:0.3333    3rd Qu.:0.6875
##   Max.   :1.0000    Max.   :1.0000    Max.    :1.0000    Max.    :1.0000
##        V6                V7                V8                V9
##   Min.   :0.0000    Min.   :0.0000    Min.    :0.0000    Min.    :0.0000
##   1st Qu.:0.0714    1st Qu.:0.3000    1st Qu.:0.2000    1st Qu.:0.5000
##   Median :0.0714    Median :0.5333    Median :0.4667    Median :0.7500
##   Mean   :0.1883    Mean   :0.5455    Mean    :0.4909    Mean    :0.6591
##   3rd Qu.:0.2143    3rd Qu.:0.8334    3rd Qu.:0.7833    3rd Qu.:1.0000
##   Max.   :1.0000    Max.   :1.0000    Max.    :1.0000    Max.    :1.0000
##        V10               V11
##   Min.   :0.0000    Min.   :0.0000
##   1st Qu.:0.0484    1st Qu.:0.0000
##   Median :0.1210    Median :0.0000
##   Mean   :0.2551    Mean   :0.1364
##   3rd Qu.:0.3387    3rd Qu.:0.0000
##   Max.   :1.0000    Max.   :1.0000
```

```r
# plot
plot(cereal, pch = 16)
```
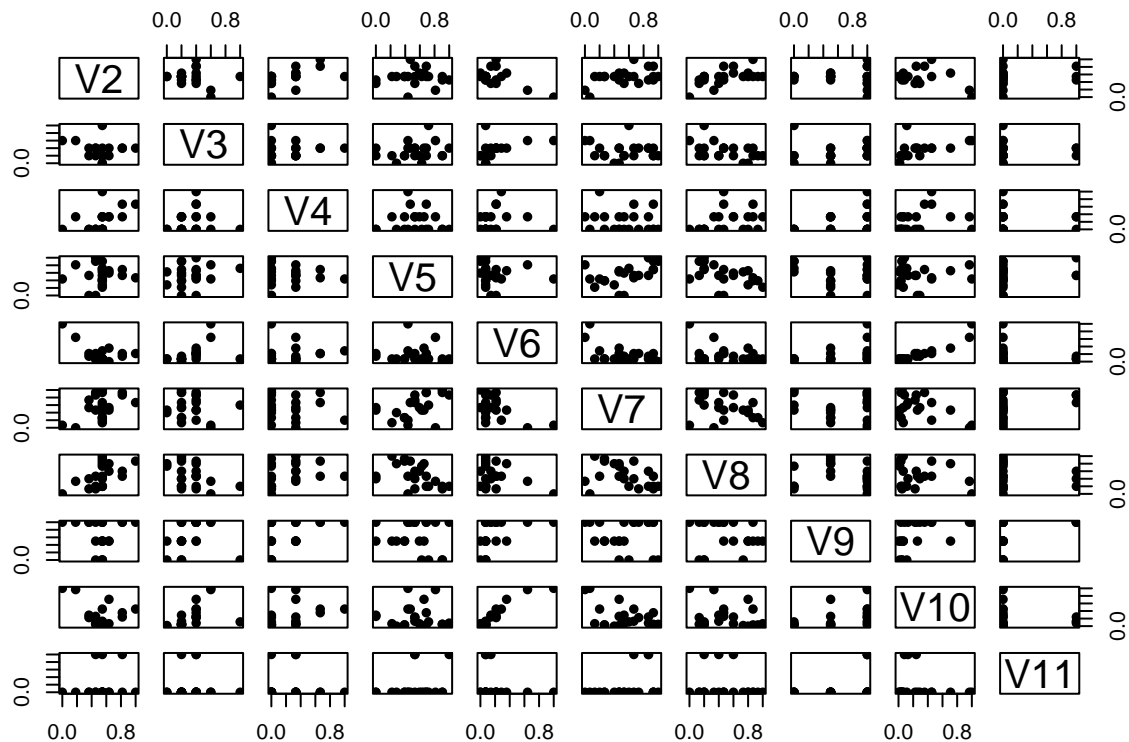


```r
# corrplot
corrplot(cor(cereal), method = "ellipse", order = "hclust")
```
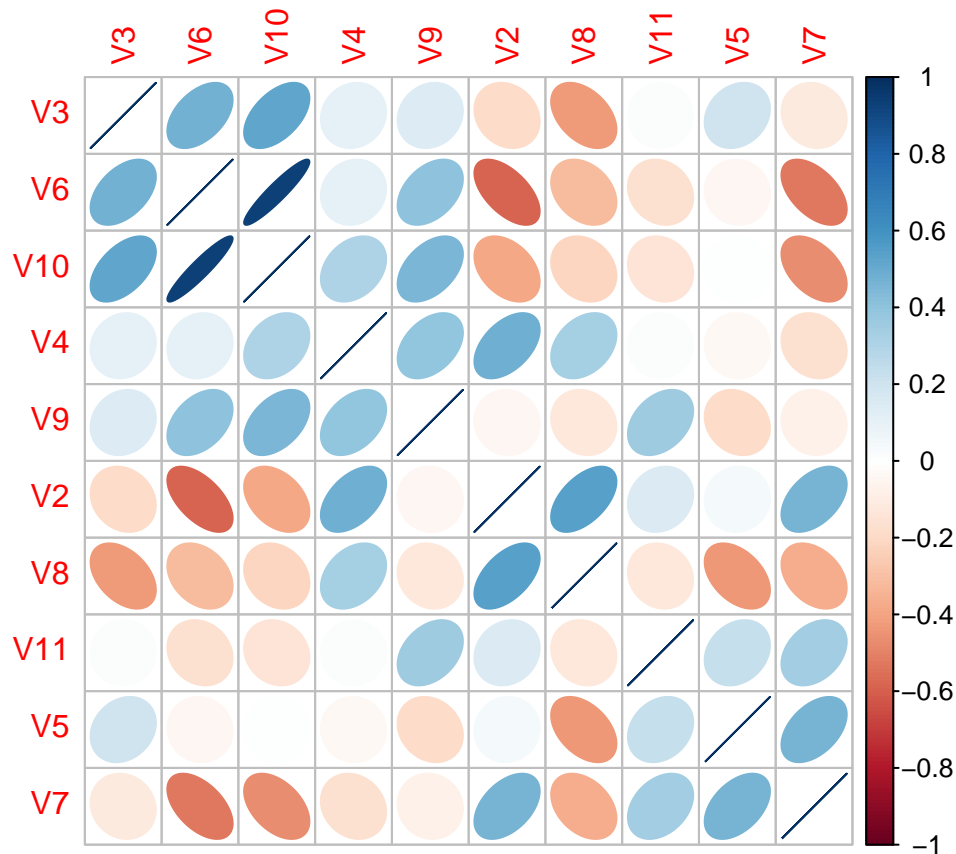
Problem 1 b)

```r
# Compute the distance function (since all are numeric)
cereal.dist <- dist(cereal)

# display distance matrix
cereal.dist
```

```
##                    AllBran AllBranFlakes AppleJacks CornFlakes   CorPops
## AllBranFlakes    0.7272322
## AppleJacks       1.5697390     1.8492052
## CornFlakes       1.8539219     2.0305045  1.2684015
## CorPops          1.6662546     1.8829115  0.2975358  1.1916989
## CracklinOatBran  1.0996082     1.5440580  1.3176119  1.7678896 1.3547525
## Crispix          1.5841069     1.7634645  1.1490144  1.0299818 1.0490237
## FrootLoops       1.5091412     1.8468267  0.3398889  1.2719303 0.4316207
## FrostedFlakes    1.8023558     2.0544616  0.6520781  0.8403389 0.6142367
## FrostedMiniWheats 1.4347282    1.4944984  0.7279911  1.2266398 0.6708088
## JustRight        1.6882887     1.9283144  1.3544907  1.5519973 1.3081891
## JustRightFruitNut 1.7672470    2.0343733  1.4196422  1.6397532 1.4069083
## Mueslix          1.4552656     1.8747545  1.1436096  1.6629513 1.1815905
## Nut&Honey        1.4997064     1.8448304  0.5907124  0.9354137 0.5692408
## NutriGrain       1.4212580     1.7941160  1.2974982  1.3748080 1.2789268
## NutFeast         1.2550124     1.3637588  1.1218121  1.1317151 1.0642234
## Product19        1.7728810     1.9876043  1.6030319  1.4348174 1.5871723
## RaisinBran       1.0511532     1.4668203  0.9100617  1.2953081 1.0048503
## RaisinWheats     1.4007230     1.4490070  0.9288793  1.4575319 0.8181519
## RiceKrispies     1.9205121     2.1198904  1.2636303  0.1492539 1.1866194
```

3

```
## HoneySmacks          1.5809483     1.8995743  0.4048276   1.4944746 0.5198887
## SpecialK             1.6873679     1.8735937  1.2857019   0.8961782 1.3616727
##                   CracklinOatBran   Crispix FrootLoops FrostedFlakes
## AllBranFlakes
## AppleJacks
## CornFlakes
## CorPops
## CracklinOatBran
## Crispix               1.3903578
## FrootLoops            1.0596577 1.1567554
## FrostedFlakes         1.6040403 1.2435004  0.7170009
## FrostedMiniWheats     1.2503425 1.0635731  0.7637698      0.9792681
## JustRight             1.3658479 1.1206756  1.2872281      1.5243700
## JustRightFruitNut     1.4332747 1.2096828  1.3654066      1.6087575
## Mueslix               0.8375832 1.2044443  0.9849233      1.4502875
## Nut&Honey             1.0618428 0.8392057  0.4448509      0.6633024
## NutriGrain            0.8945117 0.8685676  1.1368609      1.4514440
## NutFeast              1.2221023 0.4469872  1.1258005      1.3028190
## Product19             1.7389280 1.0736413  1.6085895      1.6605647
## RaisinBran            1.0022757 1.2363397  0.8389299      1.0476095
## RaisinWheats          1.1943371 0.8817143  0.9525259      1.2882294
## RiceKrispies          1.7930692 1.0284355  1.2706732      0.8331199
## HoneySmacks           1.1255875 1.3644799  0.2571313      0.8664377
## SpecialK              1.6825281 1.3261089  1.2926246      1.1490633
##                   FrostedMiniWheats JustRight JustRightFruitNut   Mueslix
## AllBranFlakes
## AppleJacks
## CornFlakes
## CorPops
## CracklinOatBran
## Crispix
## FrootLoops
## FrostedFlakes
## FrostedMiniWheats
## JustRight                 1.3316102
## JustRightFruitNut         1.3995626 0.4606777
## Mueslix                   1.2038583 1.3053475         1.1438155
## Nut&Honey                 0.8014891 1.1529193         1.2206573 0.9130613
## NutriGrain                1.2355117 1.1818840         1.0842522 0.5668569
## NutFeast                  0.8507743 1.1355754         1.2484837 1.2027871
## Product19                 1.5915576 0.6792585         0.8088116 1.6330287
## RaisinBran                0.9541119 1.3902247         1.3279287 0.8123873
## RaisinWheats              0.5597161 1.2146529         1.3392266 1.1730249
## RiceKrispies              1.2510384 1.5538330         1.6129819 1.6268183
## HoneySmacks               0.8128402 1.4129162         1.4820527 1.0690075
## SpecialK                  1.1248803 1.6825437         1.6832981 1.6302255
##                   Nut&Honey NutriGrain  NutFeast Product19 RaisinBran
## AllBranFlakes
## AppleJacks
## CornFlakes
## CorPops
## CracklinOatBran
## Crispix
## FrootLoops
```

```
## FrostedFlakes
## FrostedMiniWheats
## JustRight
## JustRightFruitNut
## Mueslix
## Nut&Honey
## NutriGrain         0.8665760
## NutFeast           0.8995366  0.9183382
## Product19          1.3695575  1.3585262 1.1365256
## RaisinBran         0.7950756  0.9329151 1.1051015 1.5842025
## RaisinWheats       0.9484267  1.1565091 0.6653321 1.4995015  1.1368606
## RiceKrispies       0.9192802  1.3498559 1.1690533 1.4425548  1.2965719
## HoneySmacks        0.6884662  1.3131275 1.2997845 1.7882332  0.9427259
## SpecialK           1.0940138  1.4632114 1.2180429 1.5873017  1.2445166
##                    RaisinWheats RiceKrispies HoneySmacks
## AllBranFlakes
## AppleJacks
## CornFlakes
## CorPops
## CracklinOatBran
## Crispix
## FrootLoops
## FrostedFlakes
## FrostedMiniWheats
## JustRight
## JustRightFruitNut
## Mueslix
## Nut&Honey
## NutriGrain
## NutFeast
## Product19
## RaisinBran
## RaisinWheats
## RiceKrispies       1.4802744
## HoneySmacks        1.0090331    1.4934093
## SpecialK           1.5069150    0.9189203    1.4530108
```

Problem 1 c)

The rule of thumb suggests that a stress value $<= 0.1$ is considered excellent, indicating a very good data fit. On the other hand, a stress value $>= 0.2$ is considered less tolerable, suggesting a poorer fit with more distortion. Since a stress value of 0.14, therefore stress value is reasonably good. Still, it indicates some discrepancy between the original distances and the distances in the reduced-dimensional space, but it is not severe. Overall, it shows a relatively good representation of the data in the lower-dimensional space.

```
# run distance matrix with the "isoMDS"
# k:    The desired dimension for the solution, passed to cmdscale.
cereal.mds  <-  isoMDS(cereal.dist, k = 2)
```

```
## initial  value 19.915627
## iter   5 value 14.639105
## iter  10 value 14.220989
## final  value 14.179482
## converged
```
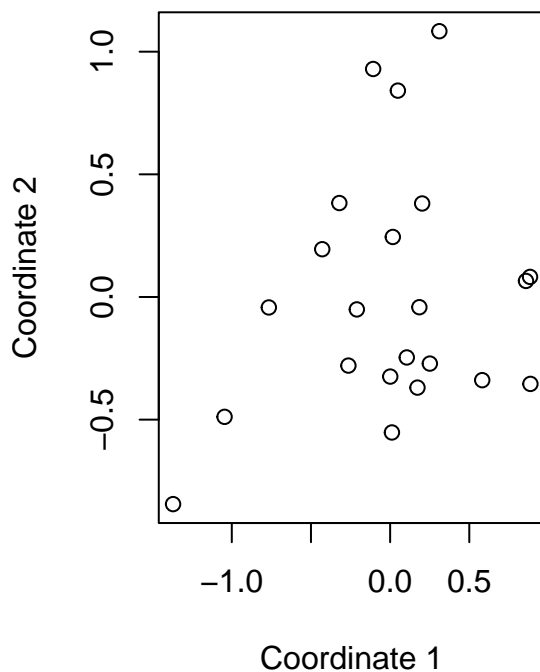
```r
# Note ... this is in %, .147
cereal.mds$stress
```
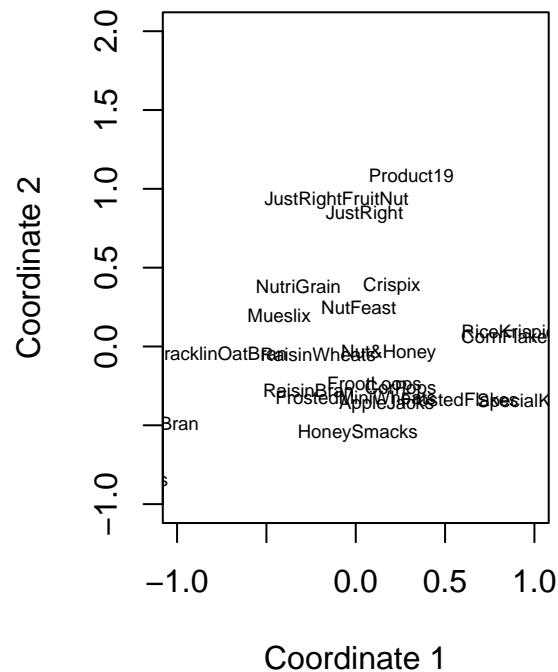
```
## [1] 14.17948
```

```r
par(mfrow = c(1,2))
# plot
plot(cereal.mds$points, xlab = "Coordinate 1", ylab = "Coordinate 2",
     main="Metric MDS Plot")
# Now plot the MDS
plot(cereal.mds$points, type = "n", xlab = "Coordinate 1", ylab = "Coordinate 2",
     xlim = c(-1.0, 1.0), ylim = c(-1.0, 2.0),  main="IsoMDS Plot")
text(cereal.mds$points, labels = row.names(cereal), cex = .6)
```

Metric MDS Plot   **IsoMDS Plot**


```r
par(mfrow = c(1,1))
```

```r
# just out of cu #riosity
# # Plotting a Shepard diagram happens in three steps
# # First we set up the plot with Shepard
# cereal.sh = Shepard(cereal.dist, cereal.mds$points)
#
# # Then we plot the points
# plot(cereal.sh, pch = ".")
#
# # Then we plot the lines
# lines(cereal.sh$x, cereal.sh$yf, type = "S", col="red")
```

Problem 1 d)

The plot can be spearated into three clusters with high sugar content cereal, healthier full balance option cereal, and "Crispy cereal" a type of breakfast cereal that has a crispy texture.

6

```
# Now plot the MDS
plot(cereal.mds$points, type = "n", xlab = "Coordinate 1", ylab = "Coordinate 2",
     xlim = c(-1.0, 1.0), ylim = c(-1.0, 2.0),  main="IsoMDS Plot")
text(cereal.mds$points, labels = row.names(cereal), cex = .6)
```
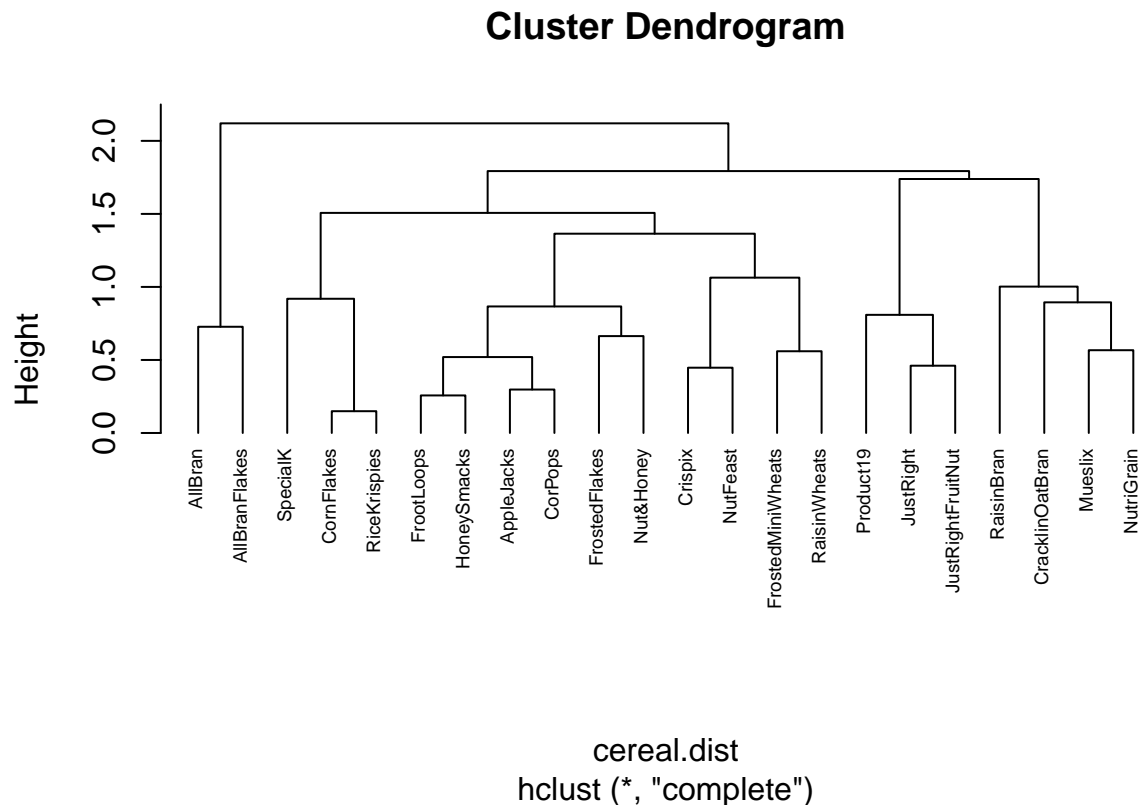
IsoMDS Plot
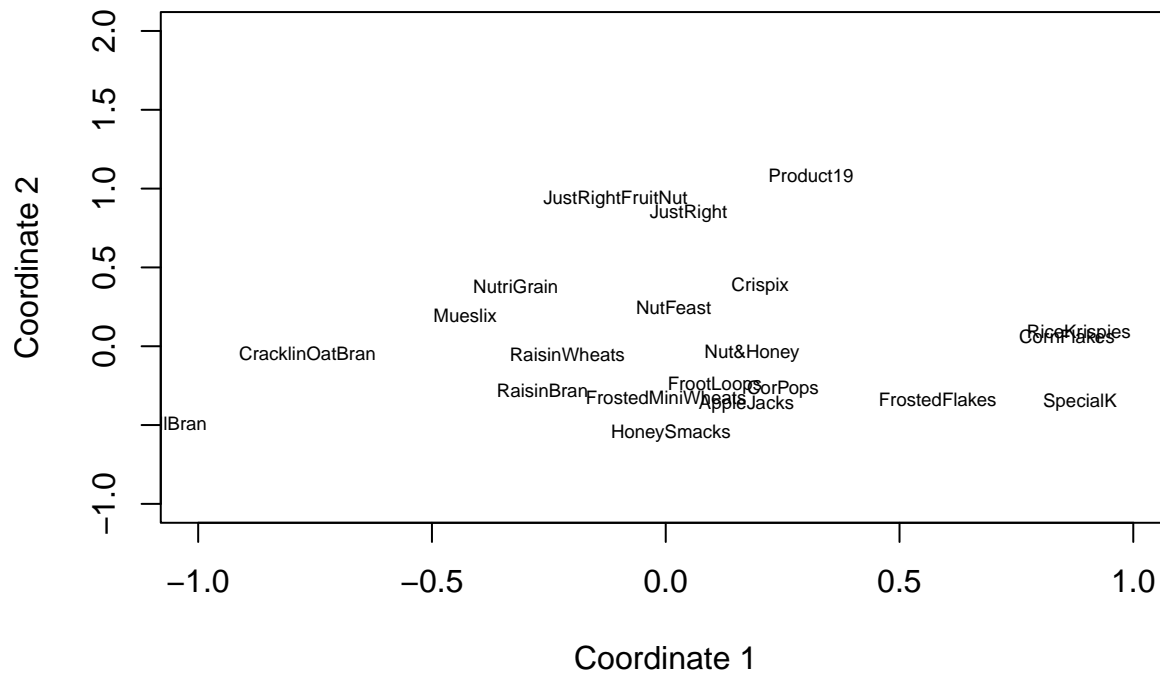
Problem 1 e)

```
# Hierarchical clustering
clusterH <- hclust(cereal.dist)

# plot dendogram
plot(clusterH, cex = 0.6, hang = -1)
```

Cluster Dendrogram

cereal.dist
hclust (*, "complete")

Problem 1 f)

All-Bran cereal, which is known for its high fiber content and its role in promoting regular bowel movements, is likely that the cereals in the red cluster share similar characteristics related to their fiber content or digestive benefits.

Cereals in the green cluster suggest that these products appeal to a target audience that prefers sweeter flavors, such as children. Therefore, cereal brands often add sugar or sweeteners to enhance the taste and make them more enjoyable for younger consumers.

Finally, the blue cluster represents cereals that are positioned as healthier choices and are formulated to provide a balanced combination of nutrients, including fiber, protein, essential vitamins, and minerals. Cereals in the blue cluster are likely designed to meet the needs of health-conscious consumers who prioritize a balanced and nutritious breakfast.

```r
# three clusters
CutclusterH <- cutree(clusterH, k = 3)

# Generate color palette
color_palette <- c("red", "green", "steelblue")

# Create dendrogram
dend <- as.dendrogram(clusterH)
dend_3 <- dendextend::color_labels(dend, col = color_palette, k = length(color_palette))

# Assign colors to dendrogram branches
colored_dend_3 <- dendextend::color_branches(dend_3, k = length(color_palette),
                                col = color_palette)

# plot
```
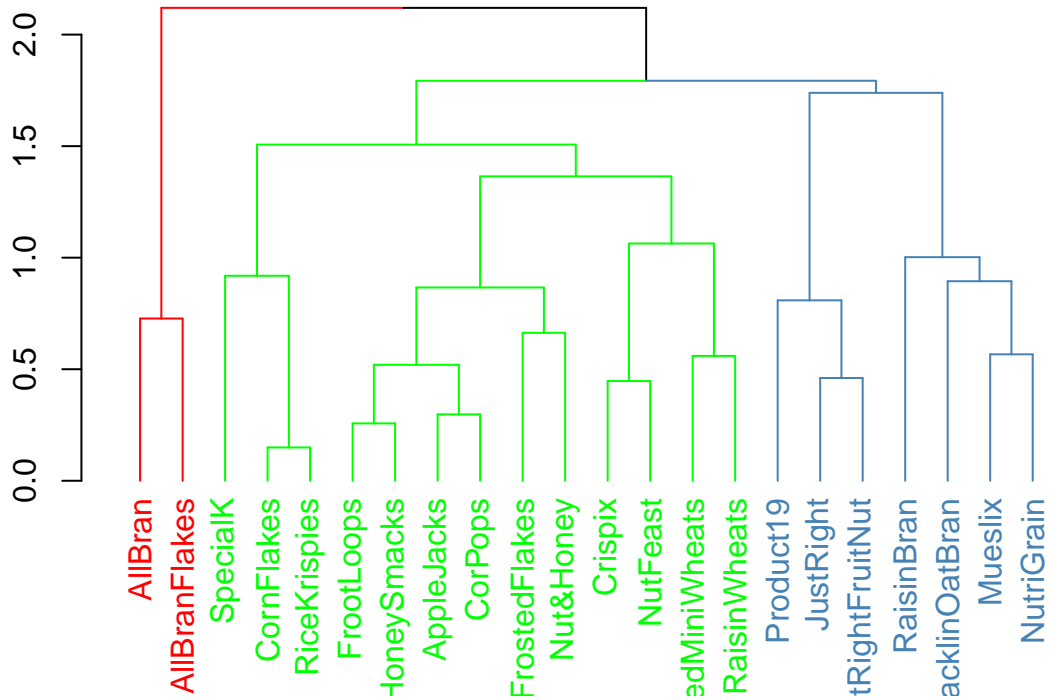
```
plot(colored_dend_3, main="Dendrogram\nEuclidean & Complete")
```
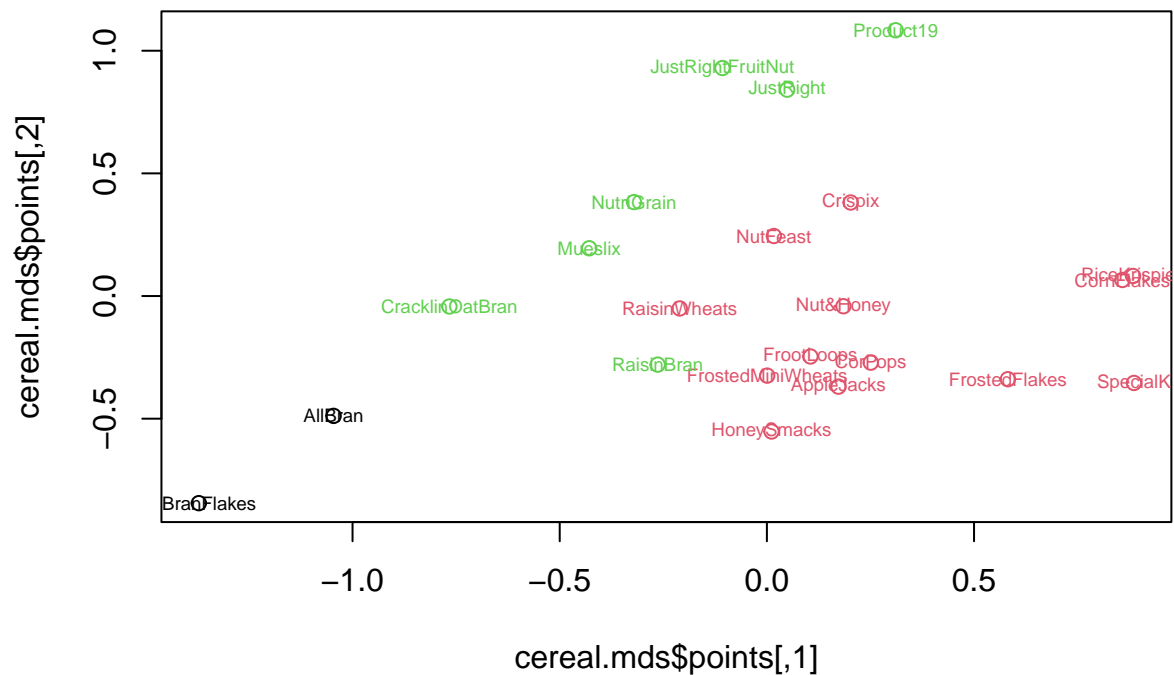
Dendrogram
Euclidean & Complete


```
# Use these clusters to color the MDS
plot(cereal.mds$points, col=CutclusterH)
text(cereal.mds$points, labels = row.names(cereal), col =CutclusterH, cex = .6)
```
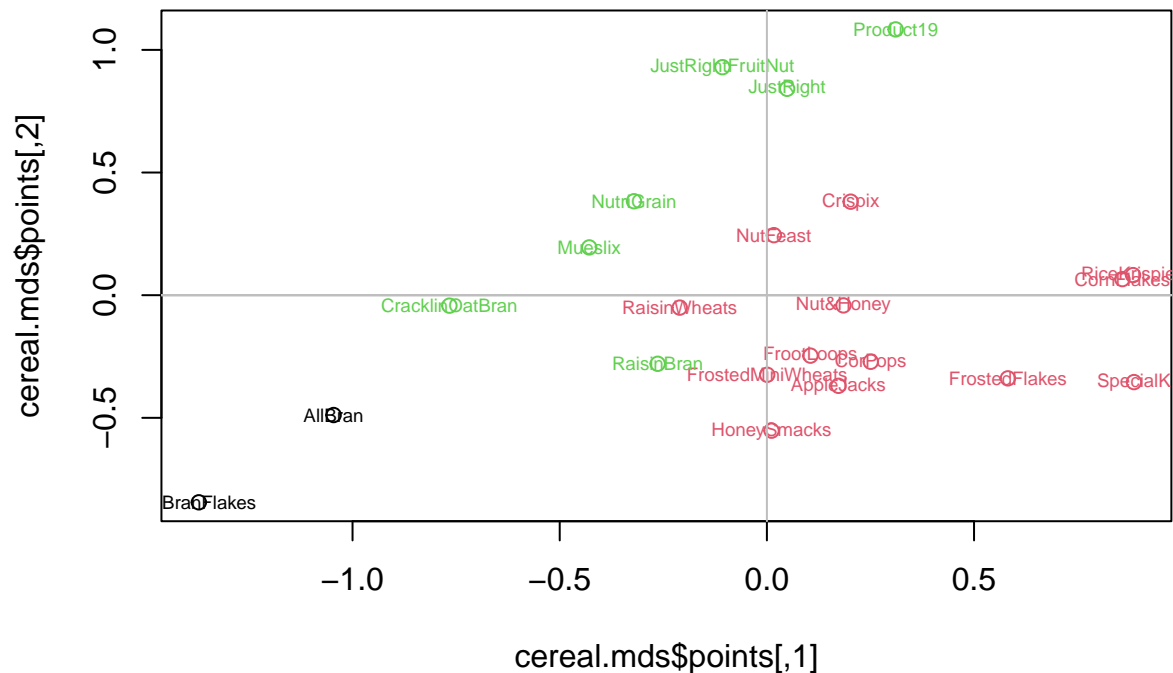
Problem 1 g)

Based on the position along the first dimension, cereals closer to the positive end are likely to have a sweeter taste profile due to higher sugar content. These cereals might appeal to individuals who prefer a sweeter flavor in their breakfast cereal. On the other hand, cereals closer to the opposing end have a milder or less sweet taste, which is suitable for those who prefer less sugary options or are looking for a more balanced flavor.

```
# Use these clusters to color the MDS
plot(cereal.mds$points, col=CutclusterH)
text(cereal.mds$points, labels = row.names(cereal), col =CutclusterH, cex = .6)
abline(0,0, col = "gray", h = 0)
abline(0,0, col = "gray", v = 0)
```


Problem 2

2) Problem 2 (Canonical Correlation Analysis – 10 points): Water, soil, and mosquito fish samples were collected at n = 165 sites/stations in the marshes of southern Florida.

The following water variables were measured:

MEHGSWB : Methyl Mercury in surface water, ng/L

TURB. : in situ surface water turbidity

DOCSWD. : Dissolved Organic Carbon in surface water, mg/L

SRPRSWFB: Soluble Reactive Phosphorus in surface water,mg/L or ug/L

THGFSFC : Total Mercury in mosquitofish (Gambusia affinis), average of 7 individuals, ug/kg

In addition, the following soil variables were measured:

THGSDFC : Total Mercury in soil, ng/g

TCSDFB : Total Carbon in soil, %

TPRSDFB : Total Phosphorus in soil, ug/g

Perform a canonical correlation analysis, describing the relationships between the soil and water variables using the data1 found in data_marsh_cleaned.csv.

Problem 2-1)

```r
# import data
marsh <- read.csv("data_marsh_cleaned.csv")

# display
head(marsh)
```

```
##    STATION MEHGSWB TURB DOCSWD    SRPRSWFB    THGFSFC THGSDFC TCSDFB TPRSDFB
## 1      12   0.703 12.3     22 0.002314096 164.28571      99  41.72     480
## 2      28   0.557 31.6     49 0.004913705 126.42857      89  26.13     320
## 3      30   0.233  6.6     19 0.003672657  17.17143      56  32.97     400
## 4      31   0.476 70.0     28 0.002796528 106.71429     150  43.53     450
## 5      33   0.442  1.6     26 0.012230508  70.57143      73  23.76     210
## 6      35   0.431 30.6     22 0.004016857 139.00000      55  19.00     180
```

```r
# summary
summary(marsh)
```

```
##     STATION         MEHGSWB            TURB            DOCSWD
##  Min.   : 12   Min.   :0.0349   Min.   :  0.000   Min.   : 4.60
##  1st Qu.:101   1st Qu.:0.1370   1st Qu.:  0.500   1st Qu.:13.00
##  Median :163   Median :0.2400   Median :  1.000   Median :17.00
##  Mean   :164   Mean   :0.3486   Mean   :  3.797   Mean   :18.99
##  3rd Qu.:214   3rd Qu.:0.3800   3rd Qu.:  2.500   3rd Qu.:22.00
##  Max.   :489   Max.   :3.8300   Max.   :110.000   Max.   :50.00
##     SRPRSWFB           THGFSFC          THGSDFC          TCSDFB
##  Min.   :0.001500   Min.   :  4.768   Min.   : 17.21   Min.   : 9.0
##  1st Qu.:0.003066   1st Qu.: 46.143   1st Qu.: 97.00   1st Qu.:31.0
##  Median :0.004500   Median : 75.286   Median :140.00   Median :44.0
##  Mean   :0.005982   Mean   : 89.578   Mean   :139.83   Mean   :38.1
##  3rd Qu.:0.006500   3rd Qu.:120.000   3rd Qu.:180.00   3rd Qu.:47.0
##  Max.   :0.065000   Max.   :320.000   Max.   :350.00   Max.   :52.0
##     TPRSDFB
##  Min.   : 110
##  1st Qu.: 310
##  Median : 400
##  Mean   : 426
##  3rd Qu.: 520
##  Max.   :1400
```

```r
# correlations
round(cor(marsh[,-1]), 2)
```

```
##          MEHGSWB  TURB DOCSWD SRPRSWFB THGFSFC THGSDFC TCSDFB TPRSDFB
## MEHGSWB     1.00  0.04   0.54    -0.06    0.05    0.16   0.20    0.02
## TURB        0.04  1.00   0.26    -0.03   -0.08   -0.05   0.04   -0.06
## DOCSWD      0.54  0.26   1.00     0.02   -0.20    0.12   0.32    0.22
## SRPRSWFB   -0.06 -0.03   0.02     1.00   -0.09   -0.10  -0.12    0.19
## THGFSFC     0.05 -0.08  -0.20    -0.09    1.00    0.25   0.04   -0.07
## THGSDFC     0.16 -0.05   0.12    -0.10    0.25    1.00   0.67    0.20
## TCSDFB      0.20  0.04   0.32    -0.12    0.04    0.67   1.00    0.32
## TPRSDFB     0.02 -0.06   0.22     0.19   -0.07    0.20   0.32    1.00
```

Problem 2-1 a)

The null hypothesis that the canonical correlations are all equal to zero was tested using the Wilks' Lambda test.

1. Canonical Correlation 1:

- Test Statistic (Wilks' Lambda): 0.70
- F-value: 4.05
- Degrees of Freedom (df1, df2): 15, 433.81
- p-value: 0.00

2. Canonical Correlation 2:

- Test Statistic (Wilks' Lambda): 0.82
- F-value: 4.18
- Degrees of Freedom (df1, df2): 8, 316.00
- p-value: 0.00

3. Canonical Correlation 3:

- Test Statistic (Wilks' Lambda): 0.93
- F-value: 4.09
- Degrees of Freedom (df1, df2): 3, 159.00
- p-value: 0.01

These results suggest strong evidence against the null hypothesis, indicating that the canonical correlations are not equal to zero. The low p-values indicate a significant relationship between the predictor variables and the response variables, as measured by the canonical correlations.

```r
# variables vector
waterCol <- c("MEHGSWB", "TURB", "DOCSWD", "SRPRSWFB", "THGFSFC")
soilCol  <- c("THGSDFC", "TCSDFB", "TPRSDFB")

# separate water and soil into respective vectors
water <- marsh %>% dplyr::select(all_of(waterCol))
soil  <- marsh %>% dplyr::select(all_of(soilCol))

# cannonical correlates
ccMarsh <-  cc(water, soil)

# hypothesis testing
wilksMarsh <-  ccaWilks(water, soil, ccMarsh)

# print
round(wilksMarsh, 2)
```

```
##      WilksL    F df1    df2    p
## [1,]   0.70 4.05  15 433.81 0.00
## [2,]   0.82 4.18   8 316.00 0.00
## [3,]   0.93 4.09   3 159.00 0.01
```

Problem 2-1 b)

The null hypothesis that the canonical correlations are all equal to zero was tested using the Wilks' Lambda test for the second and third anonical correlations equal zero.

2. Canonical Correlation 2:

- Test Statistic (Wilks' Lambda): 0.82
- F-value: 4.18
- Degrees of Freedom (df1, df2): 8, 316.00

- p-value: 0.00

3. Canonical Correlation 3:

- Test Statistic (Wilks' Lambda): 0.93
- F-value: 4.09
- Degrees of Freedom (df1, df2): 3, 159.00
- p-value: 0.01

These results suggest strong evidence against the null hypothesis, indicating that the canonical correlations for second and third are not equal to zero. The p-values less rthan 0.05 indicate a significant relationship between the predictor variables and the response variables, as measured by the canonical correlations.

Problem 2-1 c)

The null hypothesis that the canonical correlations are all equal to zero was tested using the Wilks' Lambda test for the third anonical correlations equal zero.

3. Canonical Correlation 3:

- Test Statistic (Wilks' Lambda): 0.93
- F-value: 4.09
- Degrees of Freedom (df1, df2): 3, 159.00
- p-value: 0.01

These results suggest strong evidence against the null hypothesis, indicating that the canonical correlations for the third are not equal to zero. The p-values less rthan 0.05 indicate a significant relationship between the predictor variables and the response variables, as measured by the canonical correlations.

Problem 2-1 d)

The canonical correlations measure the strength of the linear relationship between the predictor and response variables in a canonical correlation analysis. The first canonical variable has a canonical correlation of 0.39, indicating a moderate positive linear relationship between the predictor variables and the response variables represented by CV 1.

The second canonical variable has a canonical correlation of 0.34, suggesting a weaker positive linear relationship between the predictor variables and the response variables represented by CV 2 compared to CV 1.

Lastly, the third canonical variable has a canonical correlation of 0.27, indicating a weaker positive linear relationship between the predictor variables and the response variables represented by CV 3 compared to CV 1 and CV 2.

```
# library
library(yacca)
```

```
## yacca: Yet Another Canonical Correlation Analysis Package
## Version 1.4-2 created on 2022-03-08.
## copyright (c) 2008, Carter T. Butts, University of California-Irvine
##  For citation information, type citation("yacca").
##  Type help("yacca-package") to get started.
```

```
# canonical
cca.marsh.fit  <-  cca(water, soil)

# print
cca.marsh.fit
```

```
##
## Canonical Correlation Analysis
##
```

```
## Canonical Correlations:
##      CV 1      CV 2      CV 3
## 0.3855843 0.3449978 0.2675698
##
## X Coefficients:
##                CV 1         CV 2         CV 3
## MEHGSWB  -0.720571333   0.613310304   0.442819677
## TURB     -0.014902006  -0.003947628   0.046585662
## DOCSWD    0.122898091   0.045649299  -0.038307498
## SRPRSWFB 15.972715690 -77.864165952 -98.959103678
## THGFSFC  -0.004124619   0.009849176  -0.009493841
##
## Y Coefficients:
##                CV 1         CV 2         CV 3
## THGSDFC -0.011415578  0.010169482 -0.014106076
## TCSDFB   0.077556675  0.037720634  0.072787341
## TPRSDFB  0.002969355 -0.002268621 -0.004222605
##
## Structural Correlations (Loadings) - X Vars:
##                CV 1         CV 2         CV 3
## MEHGSWB    0.2138288   0.54424426   0.05580913
## TURB       0.1207027   0.03435814   0.49853147
## DOCSWD     0.8920181   0.39006177   0.02464817
## SRPRSWFB   0.1719363  -0.58138401  -0.63983875
## THGFSFC   -0.4914315   0.62009828  -0.52589688
##
## Structural Correlations (Loadings) - Y Vars:
##                CV 1        CV 2         CV 3
## THGSDFC 0.009505083  0.8836455 -0.46806012
## TCSDFB  0.639092107  0.7682559  0.03666214
## TPRSDFB 0.714065477 -0.1476683 -0.68432782
##
## Aggregate Redundancy Coefficients (Total Variance Explained):
##   X | Y: 0.07484764
##   Y | X: 0.1172123
```

```
# print three correlations
cca.marsh.fit$corr
```

```
##      CV 1      CV 2      CV 3
## 0.3855843 0.3449978 0.2675698
```

Problem 2-2 a)

water = - 0.72 * MEHGSWB - 0.01 * TURB + 0.12 * DOCSWD + 15.97 * SRPRSWFB - 0.00 * THGFSFC

water = - 0.72 * MEHGSWB - 0.01 * TURB + 0.12 * DOCSWD + 15.97 * SRPRSWFB

soil = - 0.01 * THGSDFC + 0.08 * TCSDFB + 0.00 * TPRSDFB

soil = - 0.01 * THGSDFC + 0.08 * TCSDFB

```
# x & y coefficient
cca.marsh.fit$xcoef
```

```
##                CV 1         CV 2         CV 3
## MEHGSWB  -0.720571333   0.613310304   0.442819677
## TURB     -0.014902006  -0.003947628   0.046585662
```

```
## DOCSWD    0.122898091    0.045649299   -0.038307498
## SRPRSWFB 15.972715690  -77.864165952  -98.959103678
## THGFSFC  -0.004124619    0.009849176   -0.009493841
```

```
cca.marsh.fit$ycoef
```

```
##                  CV 1          CV 2          CV 3
## THGSDFC -0.011415578   0.010169482  -0.014106076
## TCSDFB    0.077556675   0.037720634   0.072787341
## TPRSDFB   0.002969355  -0.002268621  -0.004222605
```

Problem 2-2 b)

CV1 captures a pattern where the soil variables TCSDFB (Total Carbon in soil) and TPRSDFB (Total Phosphorus in soil) have a somewhat moderately strong positive correlation, indicating that as CV1 increases, TCSDFB and TPRSDFB tend to increase as well. However, an insignificant positive correlation between CV1 and THGSDFC (Total Mercury in soil) suggests a weak or non-existent relationship between them.

For the water variables, CV1 exhibits a moderately weaker positive correlation with MEHGSWB (Methyl Mercury in surface water), TURB (in situ surface water turbidity), and SRPRSWFB (Soluble Reactive Phosphorus in surface water). These variables tend to increase together at a lower level as CV1 increases. However, DOCSWD (Dissolved Organic Carbon in surface water) has a more prominent positive influence. Additionally, CV1 has a moderate negative correlation with THGFSFC (Total Mercury in mosquitofish), indicating an inverse relationship or acting in the opposite direction.

In summary, examining the Helio plot, we can determine that Dissolved Organic Carbon in surface water, Total Carbon & Phosphorus in the soil acts positively. In contrast, Total Mercury in mosquitofish works opposite direction and has the most impact on CV1.

CV2 captures a pattern where the soil variable THGFSFC (Total Mercury in soil) has a moderately high positive correlation with TCSDFB (Total Carbon in soil), indicating that as CV2 increases, both variables tend to grow together. However, TPRSDFB (Total Phosphorus in soil) has a moderately low negative correlation with CV2, suggesting that as CV2 increases, TPRSDFB tends to decrease.

In the water variables, CV2 has a moderate positive correlation with HGFSFC (Total Mercury in soil) and TCSDFB (Total Carbon in soil) have moderaly high correlation but DOCSWD (Dissolved Organic Carbon in surface water) moderaly lower correlation, indicating that as CV2 increases, the concentrations of these variables tend to increase. On the other hand, SRPRSWFB (Soluble Reactive Phosphorus in surface water) has a moderate negative correlation with CV2, suggesting that as CV2 increases, the concentration of SRPRSWFB tends to decrease. Finally, the correlation between CV2 and TURB (in situ surface water turbidity) is nearly non-significant, indicating a weak or negligible relationship.

In summary, CV2 represents a pattern where Total Mercury in soil is positively associated with Total Carbon in soil, while Total Phosphorus in soil shows a weaker negative association. In water, CV2 is positively associated with Methyl Mercury and Dissolved Organic Carbon, but negatively associated with Soluble Reactive Phosphorus. The relationship with in situ surface water turbidity is weak or insignificant.

CV3 overall shows the most significant impact on TURB (in situ surface water turbidity) acting positively and SRPRSWFB (Soluble Reactive Phosphorus in surface water) & THGFSFC (Total Mercury in mosquitofish) in the opposite direction, with MEHGSWB & DOCSWD having the negligible opposite impact. On the soil, THGSDFC & TPRSDFB acts in the opposite direction with a negilible positive direction.

In summary, the highter the surface water turbidity all other variables thend to go oppoiste direction.

```
# correlation matrix
c <- matcor(water, soil)

# plot
img.matcor(c , type = 2)
```

X correlation

Y correlation

Cross−correlation
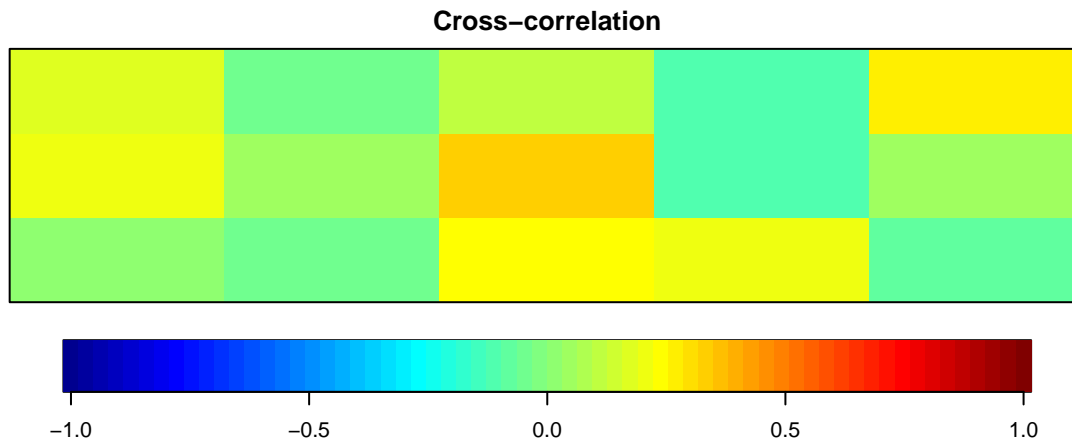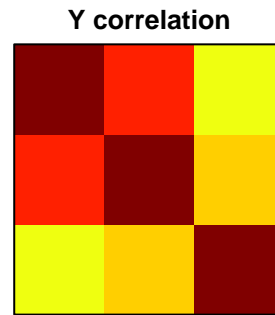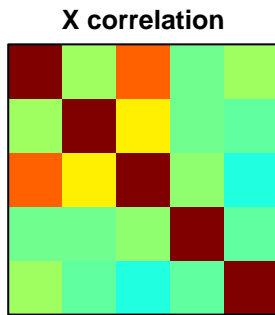

```r
# display loadings for x
cca.marsh.fit$xstructcorr
```

```
##                 CV 1        CV 2        CV 3
## MEHGSWB    0.2138288   0.54424426   0.05580913
## TURB       0.1207027   0.03435814   0.49853147
## DOCSWD     0.8920181   0.39006177   0.02464817
## SRPRSWFB   0.1719363  -0.58138401  -0.63983875
## THGFSFC   -0.4914315   0.62009828  -0.52589688
```
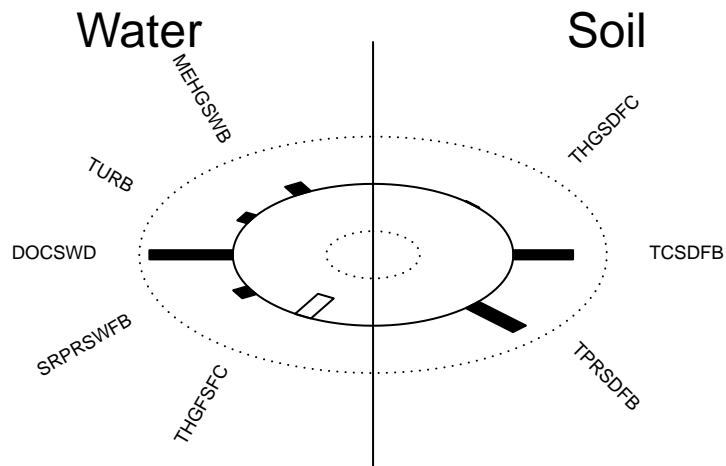
```r
# display loadings for y
cca.marsh.fit$ystructcorr
```

```
##                 CV 1        CV 2        CV 3
## THGSDFC 0.009505083   0.8836455  -0.46806012
## TCSDFB  0.639092107   0.7682559   0.03666214
## TPRSDFB 0.714065477  -0.1476683  -0.68432782
```

```r
# display loadings on first canonical variate
helio.plot(cca.marsh.fit, cv = 1, x.name="Water",
           y.name="Soil", lab.cex = .6)
```
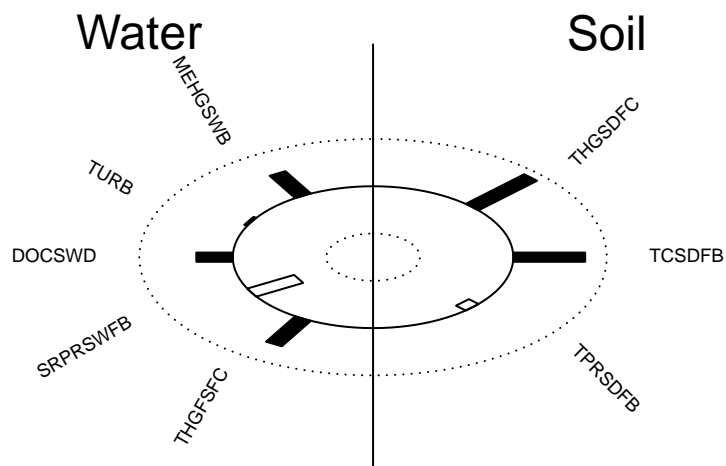
Helio Plot

Water   Soil

Canonical Variate1

```r
# display loadings on second canonical variate
helio.plot(cca.marsh.fit, cv = 2, x.name="Water",
           y.name="Soil", lab.cex = .6)
```
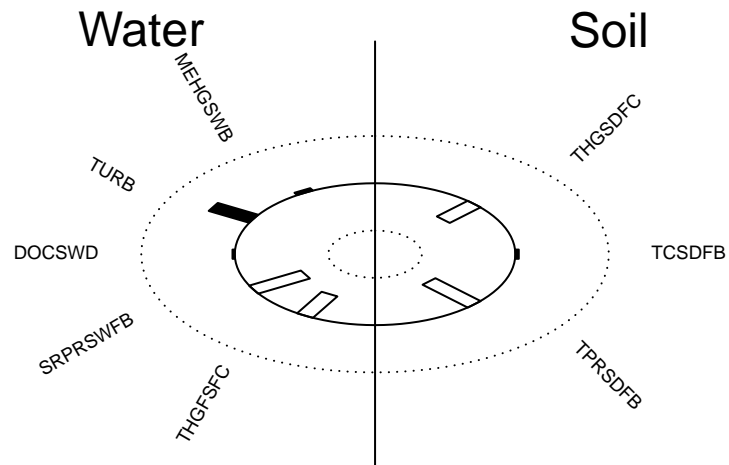
Helio Plot

Water   Soil

Canonical Variate2

```
helio.plot(cca.marsh.fit, cv = 3, x.name="Water",
           y.name="Soil", lab.cex = .6)
```

Helio Plot

Canonical Variate3