# Evaluation of Machine learning algorithms for delineating Site-Specific Management Units

Huma and Yilma

12/14/2020

### Introduction

The whole concept of precision agriculture revolves around the idea of managing infield spatial and temporal variability to both enhance the yield returns and lower the environmental footprint caused by the excess application of resources. This issue becomes more crucial by the fact that agriculture is one of the major culprits causing human induced climate change. This sector contributes to about 13.5% of global greenhouse emissions. It is responsible for the production of 70% of nitrous oxide, 50% of methane and 25% of carbon dioxide emissions globally ˆ*1.*This challenge also provides us with the opportunity.

Different precision management strategies can be used to estimate variable rates of inputs for a particular field. One of those are management zones of high, low, and medium productivity which are delineated by using bare-soil imagery, topography, and farmer's knowledge as data layers, and then inputs are applied accordingly. Management zone can be defined as the subregion of the field with homogenous soil properties which allows us to apply single rate of any agricultural supplies in that subregion. There are several methods discussed in literature for the delineation of Management zones ˆ*2.* In this project we evaluated the performance of different Machine learning algorithms to delineate the management zones.

### Methodology

## Site description

The experimental site is the field 3100 at Colorado State University's Agricultural Research Development and Educational Center, in Fort Collins Colorado, USA. (40°40' 38.24" N, 104° 58' 44.76" W).

## Soil data

Soil sample for 84 points were collected and analyzed.
• The 84 soil samples were interpolated, and 678 points were extracted from the spatial image in respective to each pixel of the study area.
• Soil parameter, like Nitrate-Nitrogen, Phosphorus-Olsen, Potassium, Calcium, Magnesium, Sodium, Sulfur, Copper, Iron, Manganese, Zinc Soil Ph, EC, and Soluble Salt are available for each 678 points.

## Vegetation indices

• Vegetation indices calculated from multi-spectral imagery for the whole period of crop growing season, every 4 to 5 days (May to October 2020). • Vegetation indices (SAVI, NDVI, OSAVI, and GNDVI) of 678 points are available in respective to each pixel of the study area.

## Electro-conductivity data

- ECa shallow and deep depth data for 678 points

## DEM

- Digital elevation modeling points for 678 points in respective to each pixel.

### Libraries used

Prior to start our analysis, we have uploaded raster, rgdal, pls, ggplot2 and ggfortify libraries for the purpose of successfully executing this specific r code.

```
set.seed(445)
Soil_data_WithoutG = read.csv("Data_4_machine learning_wzout_G.csv")
head(Soil_data_WithoutG)
```

```
##        PH      OM Nitrate_N Olsen_P       K      Ca      Mg    Sand    Silt
## 1 8.16776 1.15009   6.88318 6.86953 205.586 4854.23 597.541 47.9392 19.2798
## 2 8.16500 1.14643   7.31058 7.15690 205.965 4881.00 593.286 48.0056 19.3037
## 3 8.14588 1.17351   8.36935 7.67319 206.672 4894.07 590.770 48.2484 18.9421
## 4 8.14352 1.20639   8.92212 7.83998 213.062 4875.77 590.337 48.2449 18.8502
## 5 8.13997 1.25939   9.57739 8.80722 215.910 4883.96 590.333 48.0932 18.8358
## 6 8.13500 1.33714  10.77630 9.29141 219.447 4953.85 594.457 48.0438 18.7761
##      Clay
## 1 33.0182
## 2 32.9836
## 3 33.1589
## 4 33.3486
## 5 33.4074
## 6 33.4619
```

```
str(Soil_data_WithoutG)
```

```
## 'data.frame':    678 obs. of  10 variables:
##  $ PH       : num  8.17 8.16 8.15 8.14 8.14 ...
##  $ OM       : num  1.15 1.15 1.17 1.21 1.26 ...
##  $ Nitrate_N: num  6.88 7.31 8.37 8.92 9.58 ...
##  $ Olsen_P  : num  6.87 7.16 7.67 7.84 8.81 ...
##  $ K        : num  206 206 207 213 216 ...
##  $ Ca       : num  4854 4881 4894 4876 4884 ...
##  $ Mg       : num  598 593 591 590 590 ...
##  $ Sand     : num  47.9 48 48.2 48.2 48.1 ...
##  $ Silt     : num  19.3 19.3 18.9 18.9 18.8 ...
##  $ Clay     : num  33 33 33.2 33.3 33.4 ...
```

```
#install.packages("rgdal")
#install.packages ("raster")
#install.packages("ggplot2")
#install.packages("ggfortify")
```

```r
#install.packages ("autoplotly")
library(rgdal)
```

```
## Loading required package: sp

## rgdal: version: 1.5-18, (SVN revision 1082)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 2.2.2, released 2017/09/15
## Path to GDAL shared files: /usr/share/gdal/2.2
## GDAL binary built with GEOS: TRUE
## Loaded PROJ runtime: Rel. 4.9.2, 08 September 2015, [PJ_VERSION: 492]
## Path to PROJ shared files: (autodetected)
## Linking to sp version:1.4-4
```

```r
library(raster)
library(ggplot2)
library(ggfortify)
library(autoplotly)
```

## Principal component analysis

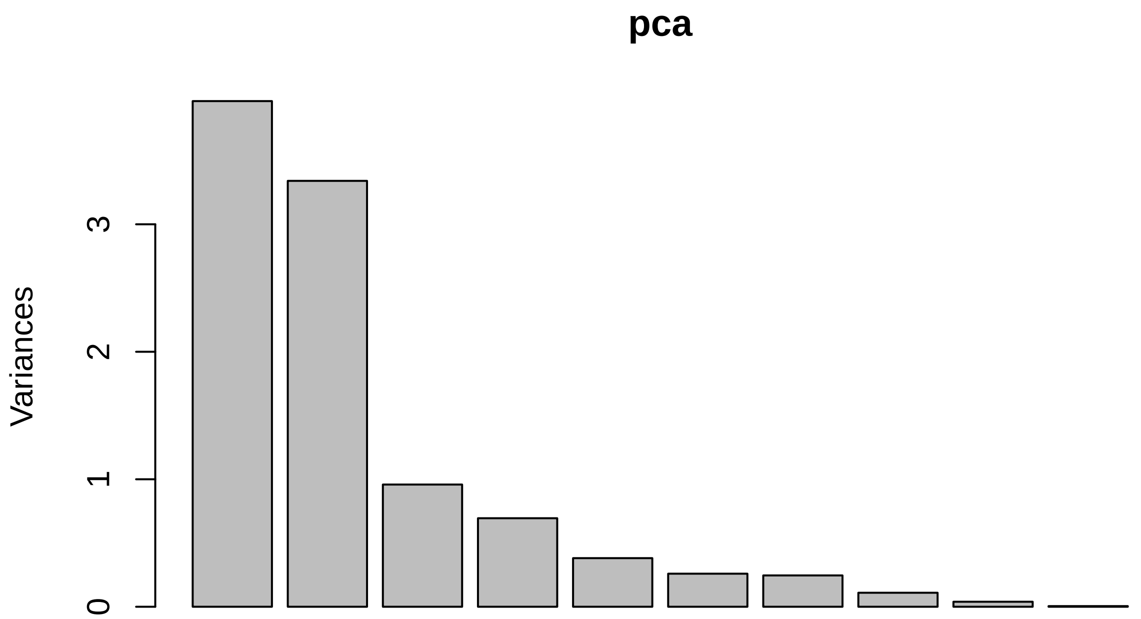
For this project two types of principal component analysis performed, the first PCA is without considering the spatial perspective of the data (normal data clustering) and the second clustering is with a perception that this data is spatial data and clustering need to take into account the spatial distribution of the data.

As we have used the variety of soil data including multiple predictors, hence, our first objective is to choose best predictors for the delineation of management zones from almost a dozen predictors. For that purpose, we have used Principal component analysis to compute principal components, by using "princomp" function in R statistical software on the scaled dataset.

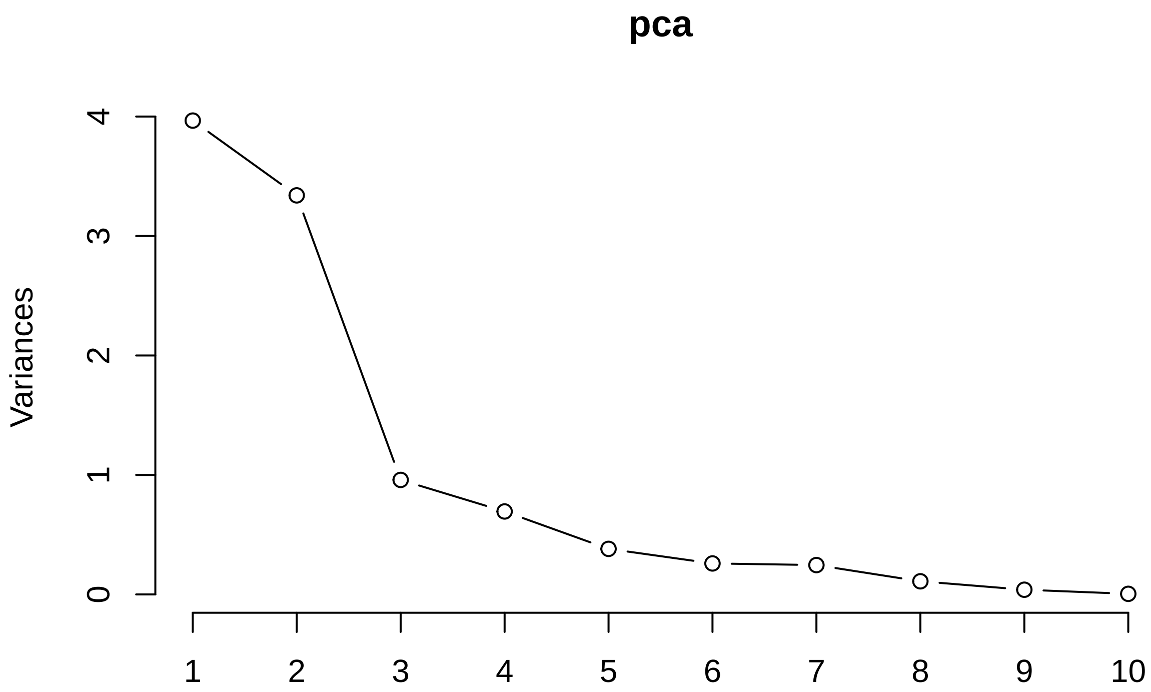
```r
#Principal component analysis



pca = prcomp(Soil_data_WithoutG, center = TRUE, scale = TRUE)
pca = prcomp(Soil_data_WithoutG[], center = TRUE, scale = TRUE)


summary(pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.9915 1.8276 0.97907 0.83333 0.61751 0.50929 0.49577
## Proportion of Variance 0.3966 0.3340 0.09586 0.06944 0.03813 0.02594 0.02458
## Cumulative Proportion  0.3966 0.7306 0.82649 0.89593 0.93406 0.96000 0.98458
##                            PC8     PC9    PC10
## Standard deviation     0.33145 0.19835 0.07077
## Proportion of Variance 0.01099 0.00393 0.00050
## Cumulative Proportion  0.99556 0.99950 1.00000
```

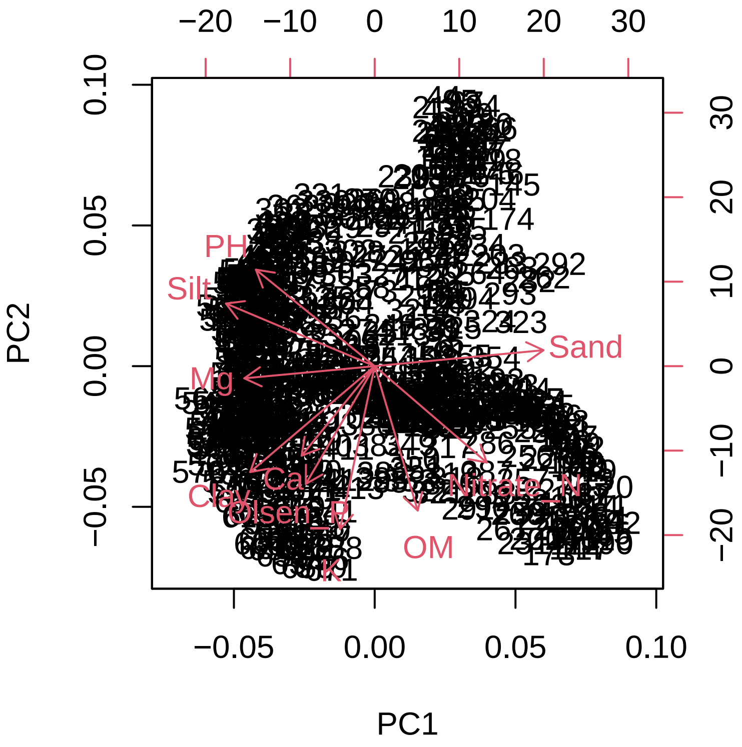
```r
plot(pca)
```



```r
plot(pca, type = 'l')
```



```r
biplot(pca)
```



```
attributes(pca)
```

```
## $names
## [1] "sdev"     "rotation" "center"   "scale"    "x"
##
## $class
## [1] "prcomp"
```

```
#pca$sdev
pca$rotation
```

```
##                   PC1         PC2          PC3         PC4         PC5
## PH        -0.33871543  0.29997378  0.247419753 -0.28691801  0.47738727
## OM         0.12274858 -0.44894535  0.244216351  0.34590721 -0.05740304
## Nitrate_N  0.31961468 -0.29977260 -0.183255629 -0.41166182 -0.19677477
## Olsen_P   -0.19639503 -0.36621778  0.524324342 -0.28576853  0.34548934
## K         -0.09904314 -0.50656730  0.158688480  0.20529023  0.03125272
## Ca        -0.20846493 -0.27849575 -0.651068629 -0.30845533  0.31643526
## Mg        -0.37194498 -0.03787905 -0.306232503  0.60935136  0.28342645
## Sand       0.48141101  0.04957907 -0.006787629  0.11008827  0.39628158
## Silt      -0.42421194  0.19468929  0.126615944 -0.01992369 -0.45745627
## Clay      -0.35518660 -0.32942008 -0.108234885 -0.16057063 -0.25095657
##                   PC6         PC7         PC8         PC9         PC10
## PH        -0.01943190  0.08855029 -0.57686143 -0.27861751  0.072352811
## OM        -0.63030197  0.07558311 -0.43361814  0.09827744  0.010760811
## Nitrate_N  0.24859337  0.65766405 -0.26314113 -0.05142266 -0.002500667
## Olsen_P    0.02850002  0.18467633  0.41867369  0.37133602 -0.057230101
## K          0.30140261 -0.12335723  0.12106717 -0.73460176  0.070140089
## Ca        -0.44884286 -0.07149366  0.18592534 -0.13103376  0.038817901
## Mg         0.26519710  0.44095413 -0.05008377  0.21667650 -0.024020781
## Sand       0.04453609 -0.00274806  0.08211782  0.05236565  0.764921075
## Silt      -0.28280131  0.37492682  0.24555619 -0.16569631  0.498142252
## Clay       0.30844435 -0.40555163 -0.34044123  0.37178403  0.388734259
```

```r
#pca$center
#pca$scale
#pca$x
pca$prcomp
```

```
## NULL
```

```r
#install.packages('ggplot2')
#library(ggplot2)

#ggplot(Soil_data_WithoutG, aes(PC1, PC2, col =))
```

```r
Soil_data_WithG = read.csv("Data_4_machine learning_with_G.csv")

head(Soil_data_WithG)
```

```
##   ID   POINT_X  POINT_Y      PH      OM Nitrate_N Olsen_P       K      Ca
## 1  1 -104.9992 40.66694 8.16776 1.15009   6.88318 6.86953 205.586 4854.23
## 2  2 -104.9991 40.66694 8.16500 1.14643   7.31058 7.15690 205.965 4881.00
## 3  3 -104.9990 40.66694 8.14588 1.17351   8.36935 7.67319 206.672 4894.07
## 4  4 -104.9989 40.66694 8.14352 1.20639   8.92212 7.83998 213.062 4875.77
## 5  5 -104.9988 40.66694 8.13997 1.25939   9.57739 8.80722 215.910 4883.96
## 6  6 -104.9986 40.66694 8.13500 1.33714  10.77630 9.29141 219.447 4953.85
##        Mg    Sand    Silt    Clay
## 1 597.541 47.9392 19.2798 33.0182
## 2 593.286 48.0056 19.3037 32.9836
## 3 590.770 48.2484 18.9421 33.1589
## 4 590.337 48.2449 18.8502 33.3486
## 5 590.333 48.0932 18.8358 33.4074
## 6 594.457 48.0438 18.7761 33.4619
```

```r
str(Soil_data_WithG)
```

```
## 'data.frame':    678 obs. of  13 variables:
##  $ ID       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ POINT_X  : num  -105 -105 -105 -105 -105 ...
##  $ POINT_Y  : num  40.7 40.7 40.7 40.7 40.7 ...
##  $ PH       : num  8.17 8.16 8.15 8.14 8.14 ...
##  $ OM       : num  1.15 1.15 1.17 1.21 1.26 ...
##  $ Nitrate_N: num  6.88 7.31 8.37 8.92 9.58 ...
##  $ Olsen_P  : num  6.87 7.16 7.67 7.84 8.81 ...
##  $ K        : num  206 206 207 213 216 ...
##  $ Ca       : num  4854 4881 4894 4876 4884 ...
##  $ Mg       : num  598 593 591 590 590 ...
##  $ Sand     : num  47.9 48 48.2 48.2 48.1 ...
##  $ Silt     : num  19.3 19.3 18.9 18.9 18.8 ...
##  $ Clay     : num  33 33 33.2 33.3 33.4 ...
```

```r
colnames(Soil_data_WithG)
```

```
##  [1] "ID"        "POINT_X"   "POINT_Y"   "PH"        "OM"        "Nitrate_N"
##  [7] "Olsen_P"   "K"         "Ca"        "Mg"        "Sand"      "Silt"
## [13] "Clay"
```

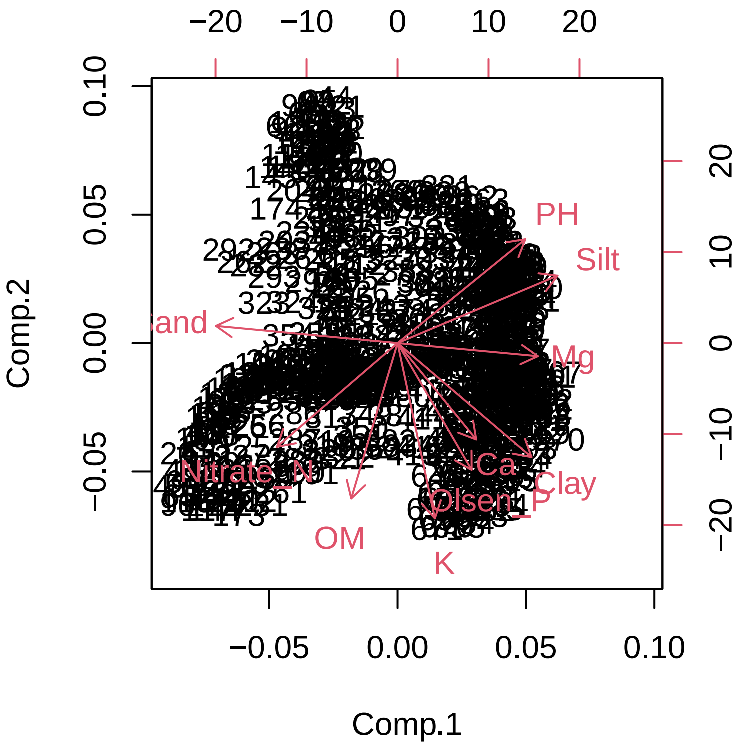
```
Soil_data_WithG.scaled = scale(as.matrix(Soil_data_WithG[4:13]))

#princomp performs a principal components analysis on the given numeric data matrix and returns the res

pca = princomp(Soil_data_WithG.scaled,cor=F,scores=T) # use covariance matrix to match the following...

attributes(pca)
```

```
## $names
## [1] "sdev"     "loadings" "center"   "scale"    "n.obs"    "scores"   "call"
##
## $class
## [1] "princomp"
```

```
pca$loadings
```

```
##
## Loadings:
##           Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
## PH         0.339  0.300  0.247  0.287  0.477                0.577  0.279
## OM        -0.123 -0.449  0.244 -0.346         0.630         0.434
## Nitrate_N -0.320 -0.300 -0.183  0.412 -0.197 -0.249  0.658  0.263
## Olsen_P    0.196 -0.366  0.524  0.286  0.345         0.185 -0.419 -0.371
## K                -0.507  0.159 -0.205        -0.301 -0.123 -0.121  0.735
## Ca         0.208 -0.278 -0.651  0.308  0.316  0.449        -0.186  0.131
## Mg         0.372        -0.306 -0.609  0.283 -0.265  0.441        -0.217
## Sand      -0.481               -0.110  0.396
## Silt       0.424  0.195  0.127        -0.457  0.283  0.375 -0.246  0.166
## Clay       0.355 -0.329 -0.108  0.161 -0.251 -0.308 -0.406  0.340 -0.372
##           Comp.10
## PH
## OM
## Nitrate_N
## Olsen_P
## K
## Ca
## Mg
## Sand       0.765
## Silt       0.498
## Clay       0.389
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
## SS loadings       1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0
## Proportion Var    0.1    0.1    0.1    0.1    0.1    0.1    0.1    0.1    0.1
## Cumulative Var    0.1    0.2    0.3    0.4    0.5    0.6    0.7    0.8    0.9
##                Comp.10
## SS loadings        1.0
## Proportion Var     0.1
## Cumulative Var     1.0
```

```
biplot(pca)
```



## Study area

The PCA indicates what the individual weightings are for the predictors involved, however it does not consider the actual geography. For that purpose, we have imported the ESRI shape file of the experimental site. To be able to use the study area map for the later visualizations, we have defined a function the "backdrop ()" to bring up our study area map later in the analysis. We have mapped the PC1 and PC2 by indicating the points greater than zero as yellow and less than zero as brown, respectively.

```
library(rgdal)
Study_area= readOGR("Study_area.shp")
```

```
## Warning in ogrInfo(dsn = dsn, layer = layer, encoding = encoding, use_iconv =
## use_iconv, : ogrInfo: /cloud/project/Study_area.dbf not found

## OGR data source with driver: ESRI Shapefile
## Source: "/cloud/project/Study_area.shp", layer: "Study_area"
## with 1 features
## It has 0 fields

## Warning in readOGR("Study_area.shp"): Z-dimension discarded
```

```
#crs(Study_area)

plot(Study_area, asp=1,type='l',xaxt='n',yaxt='n',xlab='',ylab='',bty='n',col='grey')
```

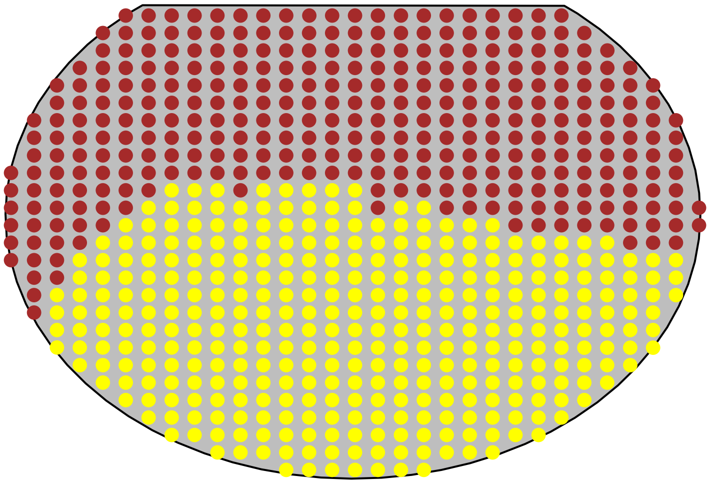
```
## Warning in title(...): graphical parameter "type" is obsolete

## Warning in polypath(x = mcrds[, 1], y = mcrds[, 2], border = border, col =
## col, : graphical parameter "type" is obsolete
```

```
backdrop = function()
```

```
plot(Study_area,asp=1,type='l',xaxt='n',yaxt='n',xlab='',ylab='',bty='n',col='grey')

pc1 = pca$scores[,1]
backdrop()
```

```
## Warning in title(...): graphical parameter "type" is obsolete

## Warning in title(...): graphical parameter "type" is obsolete
```

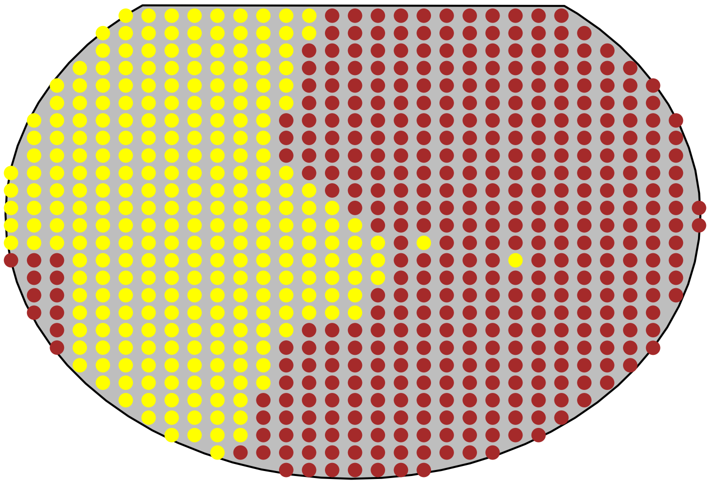
```
points(Soil_data_WithG$POINT_X[pc1>0],Soil_data_WithG$POINT_Y[pc1>0],pch=16,col='yellow')
points(Soil_data_WithG$POINT_X[pc1<0],Soil_data_WithG$POINT_Y[pc1<0],pch=16,col='brown')
```



```
pc2 = pca$scores[,2]
backdrop()
```

```
## Warning in title(...): graphical parameter "type" is obsolete

## Warning in title(...): graphical parameter "type" is obsolete
```

```
points(Soil_data_WithG$POINT_X[pc2>0],Soil_data_WithG$POINT_Y[pc2>0],pch=16,col='yellow')
points(Soil_data_WithG$POINT_X[pc2<0],Soil_data_WithG$POINT_Y[pc2<0],pch=16,col='brown')
```



# Discussion on PCA output:

Summary output is giving us three values for each principal component including Standard deviation along specific PC, proportion of variation of the original data explained by each PC and cumulative proportion of variation explained by PCs. It indicates that PC1 and PC2 are representatives of most variability (~72%) present in the data set. As shown below in the PCA plot, it shows PC1 and PC2 on X and Y axis, respectively. The plot in general reveals the overall pattern of the data with respect to respective axes. For instance, PC1 seems to have placed equal weight on PH and silt while PC2 placing same weight on clay, potassium and Phosphorous.

Output figures indicating a spatial trend of PC1 and PC2 respectively.

## K-means clustering based on soil and vegetation indices (satellite imagery based)

K-means clustering is one of most commonly used unsupervised learning for a successful grouping according to numerous researchesˆ*3,4*. Delineating site–specific management zone is one of the proven technique, accepted worldwide to practice a precision agriculture for a spatially variable fieldˆ*2*.
Pre-season and in-season management zone required to understand the soil and crop status variability, respectively. This project used k-means clustering algorithms to represent zonal clustering on homogeneity of soil properties and bare soil indicesˆ*5*. Two k-mean clustering performed, the first clustering is with combination of all the soil data and the second clustering is with a combination of bare NDVI, SAVI, OSAVI and GNDVI. Both clustering considered spatial characteristics of the data and clustering was conducted with perspective of the spatial distribution of the data.
All the data for both clustering are quantitative, a total of 678 observation and 9 variables used for the clustering based on soil information. Data indexed from column 4 to 13 specifically and used as data input for executing the k-means function in r. For vegetation indices-based clustering, a combination of 4 vegetation indices used.
The vegetation indices-based clustering is assumed for in-season period even if this specific data were at the beginning of the season. K-means cluster with k-means function method used for both cases to perform soil fertility status grouping. We have determined the number of clusters to be 3 based on the assumption that we will have three soil fertility status (poor, medium and good). For vegetation indices-based clustering, we have used vegetation as indicative of variability of soil properties to form cluster using k-means function and Lloyd method.
Lloyd algorithm with maximum iteration of 500 used to do clustering for both cases. Lloyd uses the selected k data points as center, assign the data to closest center each time and update cluster during each iteration.

```r
#k-means clustering based on sOil data

#Data
Data_4_Kmeans = read.csv("Data_4_machine learning_with_G.csv")

#View(Data_4_Kmeans)

#Prepare Data
K_means_cluster  = kmeans(Data_4_Kmeans[,4:13], centers = 3, iter.max = 500, nstart = 3, algorithm="Llo

attributes(K_means_cluster)
```

```
## $names
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
##
## $class
## [1] "kmeans"
```

```r
str(K_means_cluster)
```

```
## List of 9
##  $ cluster     : int [1:678] 3 3 3 3 3 3 2 2 2 2 ...
##  $ centers     : num [1:3, 1:10] 8.12 8.12 8.13 1.38 1.37 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3] "1" "2" "3"
##   .. ..$ : chr [1:10] "PH" "OM" "Nitrate_N" "Olsen_P" ...
##  $ totss       : num 11736123
##  $ withinss    : num [1:3] 795617 1032670 740005
##  $ tot.withinss: num 2568291
##  $ betweenss   : num 9167831
##  $ size        : int [1:3] 250 323 105
##  $ iter        : int 9
##  $ ifault      : NULL
##  - attr(*, "class")= chr "kmeans"
```

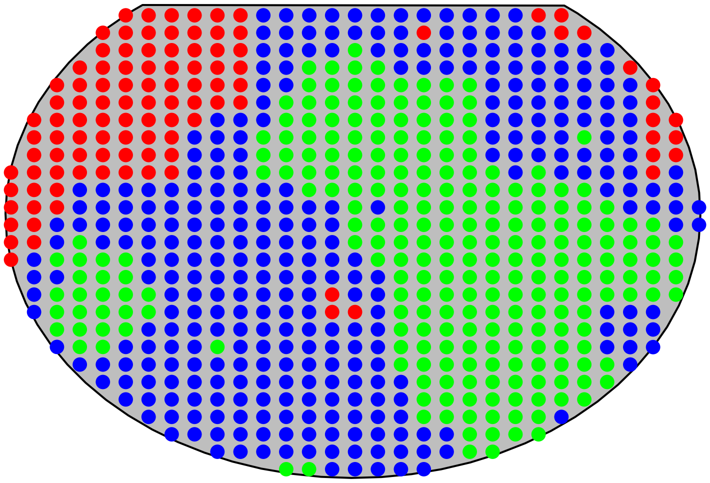
```r
cluster_4_plot = K_means_cluster$cluster

#Study area
Study_area = readOGR("Study_area.shp")
```

```
## Warning in ogrInfo(dsn = dsn, layer = layer, encoding = encoding, use_iconv =
## use_iconv, : ogrInfo: /cloud/project/Study_area.dbf not found

## OGR data source with driver: ESRI Shapefile
## Source: "/cloud/project/Study_area.shp", layer: "Study_area"
## with 1 features
## It has 0 fields

## Warning in readOGR("Study_area.shp"): Z-dimension discarded
```

```r
#crs(Study_area)

#plot
plot(Study_area, asp=1,type='l',xaxt='n',yaxt='n',xlab='',ylab='',bty='n',col='grey')
```

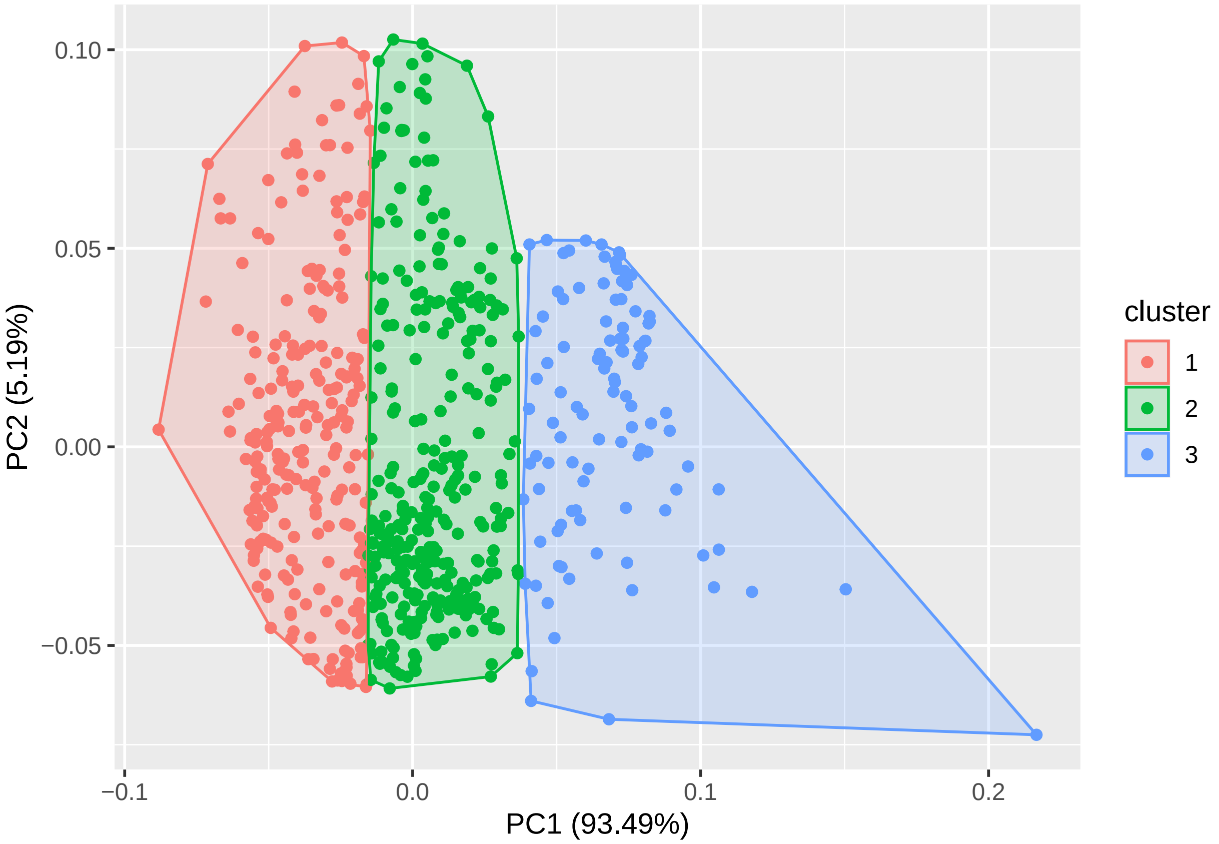
```
## Warning in title(...): graphical parameter "type" is obsolete

## Warning in polypath(x = mcrds[, 1], y = mcrds[, 2], border = border, col =
## col, : graphical parameter "type" is obsolete
```

```r
backdrop_1 = function()
plot(Study_area, asp=1,type='l',xaxt='n',yaxt='n',xlab='',ylab='',bty='n',col='grey')

points(Data_4_Kmeans$POINT_X[cluster_4_plot==1], Data_4_Kmeans$POINT_Y[cluster_4_plot==1], pch=16, col=
points(Data_4_Kmeans$POINT_X[cluster_4_plot==2], Data_4_Kmeans$POINT_Y[cluster_4_plot==2], pch=16, col=
points(Data_4_Kmeans$POINT_X[cluster_4_plot==3], Data_4_Kmeans$POINT_Y[cluster_4_plot==3], pch=16, col=
```



```r
library(autoplotly)
library(ggplot2)
autoplot(K_means_cluster, Data_4_Kmeans[,4:13], frame=TRUE)
```

```
## Warning: `select_()` is deprecated as of dplyr 0.7.0.
## Please use `select()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.

## Warning: `group_by_()` is deprecated as of dplyr 0.7.0.
## Please use `group_by()` instead.
## See vignette('programming') for more help
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```



## Soil data based k-means clustering results

We have performed the k-means clustering for both cases. The output indicates that k-means clustering has performed well during both conditions and the geographically weighted vegetation indices cluster map were able to be compared to vegetation indices map, which is based on satellite data. As show in the above figures, these clusters look distinct and without any overlapping, we can conclude that the clustering analysis was successfully executed.

```
#k-means clustering based on NDVI data

#Data
NDVI_Data_4_Kmeans = read.csv("Data_4_machine learning_bare_NDVI_SAVI.csv")

#Prepare Data
NDVI_K_means_cluster = kmeans(NDVI_Data_4_Kmeans[,4:7], centers = 3, iter.max = 500, nstart = 3, algori

attributes(NDVI_K_means_cluster)
```

```
## $names
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"      
##
## $class
## [1] "kmeans"
```

```r
str(NDVI_K_means_cluster)
```

```
## List of 9
##  $ cluster     : int [1:678] 3 3 3 3 3 3 3 3 3 3 ...
##  $ centers     : num [1:3, 1:4] 0.0718 0.0793 0.1278 0.101 0.1159 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3] "1" "2" "3"
##   .. ..$ : chr [1:4] "Bare_SAVI" "Bare_NDVI" "BARE_GNDVI" "Bare_OSAVI"
##  $ totss       : num 0.615
##  $ withinss    : num [1:3] 0.0521 0.052 0.0336
##  $ tot.withinss: num 0.138
##  $ betweenss   : num 0.477
##  $ size        : int [1:3] 356 302 20
##  $ iter        : int 18
##  $ ifault      : NULL
##  - attr(*, "class")= chr "kmeans"
```

```r
cluster_4_NDVI_plot = NDVI_K_means_cluster$cluster

#Study area
Study_area = readOGR("Study_area.shp")
```

```
## Warning in ogrInfo(dsn = dsn, layer = layer, encoding = encoding, use_iconv =
## use_iconv, : ogrInfo: /cloud/project/Study_area.dbf not found
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/cloud/project/Study_area.shp", layer: "Study_area"
## with 1 features
## It has 0 fields
```

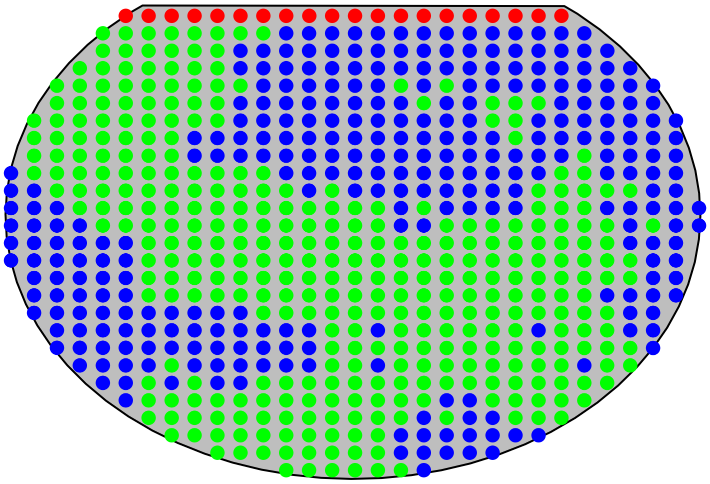
```
## Warning in readOGR("Study_area.shp"): Z-dimension discarded
```

```r
#crs(Study_area)

#plot
plot(Study_area, asp=1,type='l',xaxt='n',yaxt='n',xlab='',ylab='',bty='n',col='grey')
```

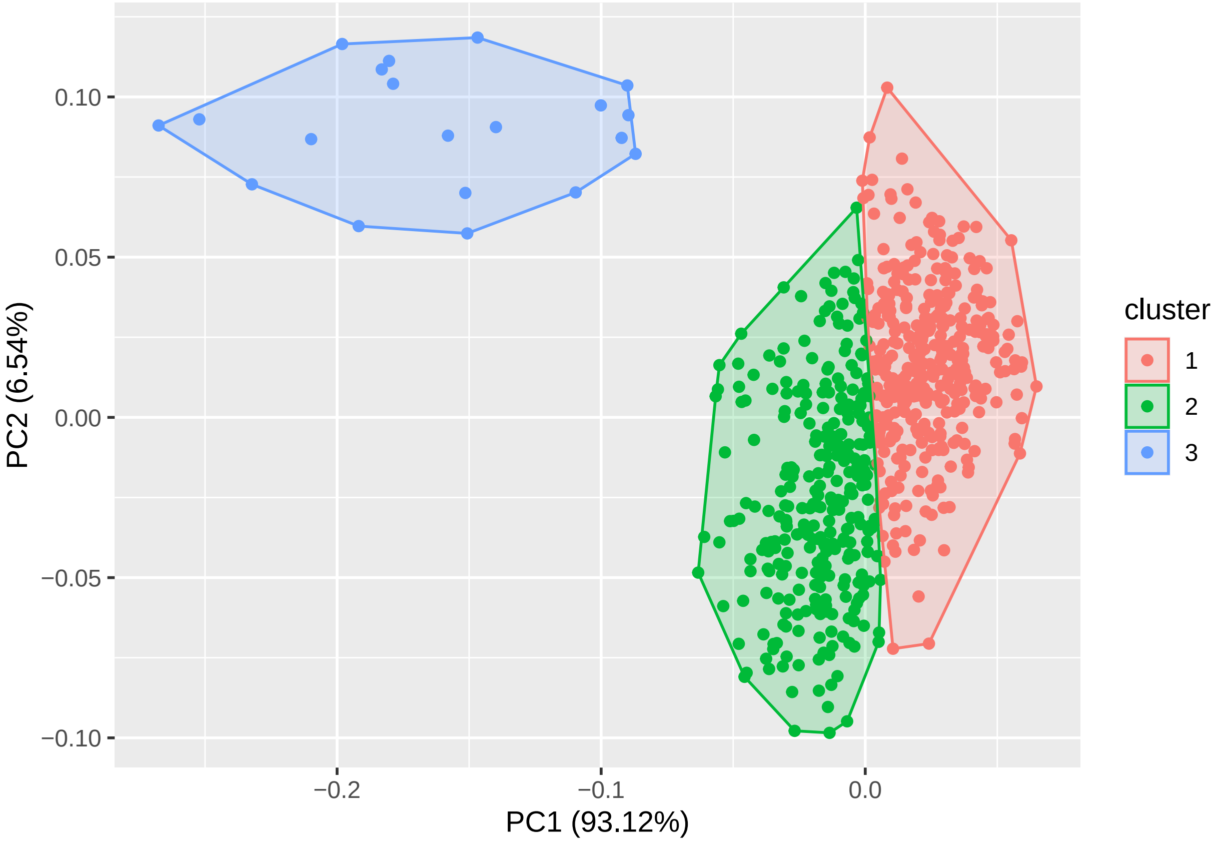
```
## Warning in title(...): graphical parameter "type" is obsolete
```

```
## Warning in polypath(x = mcrds[, 1], y = mcrds[, 2], border = border, col =
## col, : graphical parameter "type" is obsolete
```

```r
backdrop_1 = function()
plot(Study_area, asp=1,type='l',xaxt='n',yaxt='n',xlab='',ylab='',bty='n',col='grey')

points(NDVI_Data_4_Kmeans$POINT_X[cluster_4_NDVI_plot==1], NDVI_Data_4_Kmeans$POINT_Y[cluster_4_NDVI_pl
points(NDVI_Data_4_Kmeans$POINT_X[cluster_4_NDVI_plot==2], NDVI_Data_4_Kmeans$POINT_Y[cluster_4_NDVI_pl
points(NDVI_Data_4_Kmeans$POINT_X[cluster_4_NDVI_plot==3], NDVI_Data_4_Kmeans$POINT_Y[cluster_4_NDVI_pl
```



```
autoplot(NDVI_K_means_cluster, NDVI_Data_4_Kmeans[,4:7], frame=TRUE)
```



## Vegetation indices based k-means clustering results

Vegetation indices based in-season clustering looks distinct and perfectly clustered. As shown in figures above, the spatial distribution comparison between k-means clustered vegetation indices and NDVI look similar. Specifically, at the northern part of the field, on both maps it clearly shows that there is a different kind of vegetation covers, which on ground is true.

## Nitrogen and organic matter prediction prediction

Predicting nitrogen and organic matter with reflectance data as a predictor is common and numerous studies successfully predicted soil nutrient and crop nutrient status mainly from hyperspectral informationˆ*6.* For this project, only partial least square regression (PSLR) method used to predict nitrogen and organic matter and assess how well it perform compared to the observed variable. A model is developed to be help us predict nitrogen and organic matter separately from using combination of multi-spectral based vegetation indices, electro-conductivity data and digital elevation model. For training the PSLR model, the data was divided the full data into training and test to train and test the model respectively. Scaling of the dataset were applied and cross validation was used as a validation approach.

```r
set.seed(445)
#Nitrogen prediction based on ECa and VI based
N_prediction = read.csv("Data_4_machine learning_PLSR.csv")
str(N_prediction)
```

```
## 'data.frame':    678 obs. of  10 variables:
##  $ OBJECTID_1 : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ POINT_X    : num  -105 -105 -105 -105 -105 ...
##  $ POINT_Y    : num  40.7 40.7 40.7 40.7 40.7 ...
##  $ OM         : num  1.15 1.15 1.17 1.21 1.26 ...
##  $ Nitrate_N  : num  6.88 7.31 8.37 8.92 9.58 ...
##  $ Eca_deep   : num  40.5 40.2 40.3 40.8 41.5 ...
##  $ Eca_shallow: num  14.1 14.2 14.4 14.7 14.8 ...
##  $ Bare_SAVI  : num  0.109 0.11 0.108 0.113 0.109 ...
##  $ Bare_NDVI  : num  0.152 0.153 0.15 0.156 0.153 ...
##  $ DEM        : int  1563 1563 1563 1563 1563 1563 1563 1563 1563 1562 ...
```

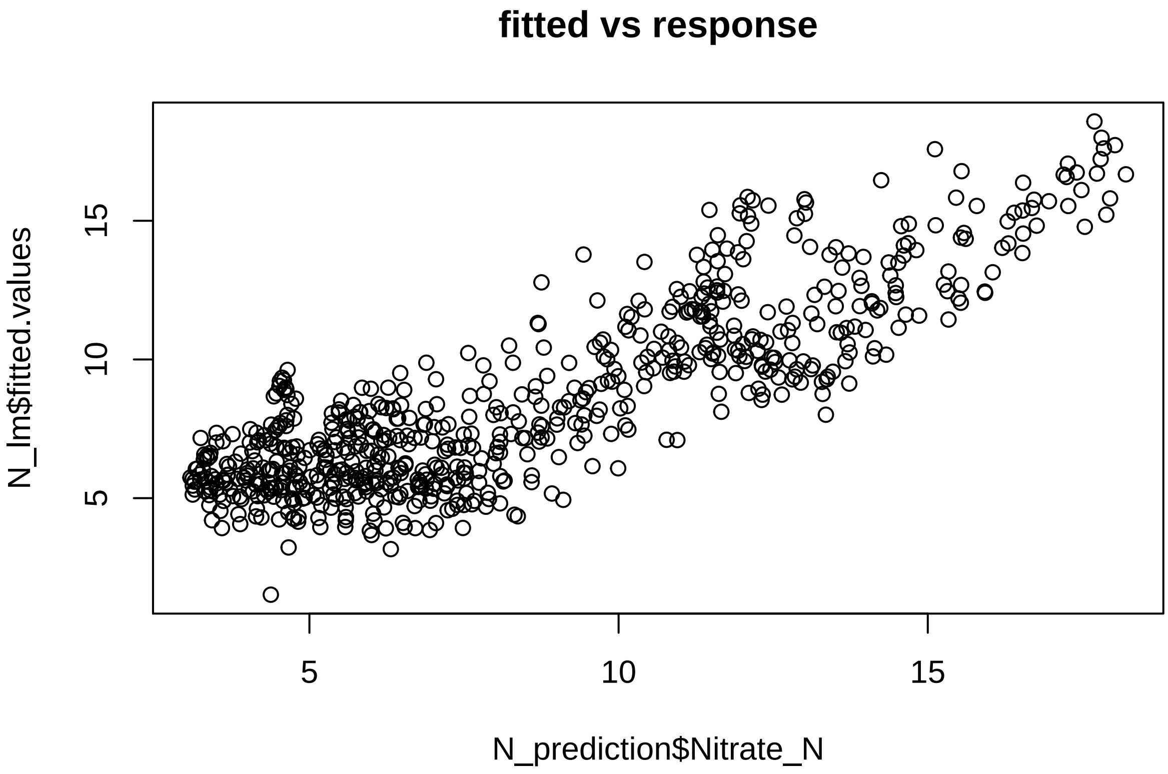
```r
N_lm = lm(Nitrate_N ~ Eca_deep + Eca_shallow + Bare_SAVI + Bare_NDVI + DEM , data = N_prediction)
summary(N_lm)
```

```
##
## Call:
## lm(formula = Nitrate_N ~ Eca_deep + Eca_shallow + Bare_SAVI +
##     Bare_NDVI + DEM, data = N_prediction)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9755 -1.5251 -0.0132  1.4344  5.3443
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1782.99391  136.24179  13.087  < 2e-16 ***
## Eca_deep       0.10784    0.09972   1.081 0.279891
## Eca_shallow    0.52108    0.13691   3.806 0.000154 ***
## Bare_SAVI   -723.01386   32.36161 -22.342  < 2e-16 ***
## Bare_NDVI    544.72083   22.63566  24.065  < 2e-16 ***
## DEM           -1.14749    0.08624 -13.306  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.022 on 672 degrees of freedom
## Multiple R-squared:  0.7291, Adjusted R-squared:  0.7271
## F-statistic: 361.8 on 5 and 672 DF,  p-value: < 2.2e-16
```

```r
summary(N_prediction)
```

```
##    OBJECTID_1       POINT_X         POINT_Y            OM        
##  Min.   :  1.0   Min.   :-105   Min.   :40.66   Min.   :0.9322  
##  1st Qu.:170.2   1st Qu.:-105   1st Qu.:40.67   1st Qu.:1.2161  
##  Median :339.5   Median :-105   Median :40.67   Median :1.3492  
##  Mean   :339.5   Mean   :-105   Mean   :40.67   Mean   :1.3583  
##  3rd Qu.:508.8   3rd Qu.:-105   3rd Qu.:40.67   3rd Qu.:1.4856  
##  Max.   :678.0   Max.   :-105   Max.   :40.67   Max.   :1.7893  
##    Nitrate_N         Eca_deep       Eca_shallow        Bare_SAVI      
##  Min.   : 3.076   Min.   :39.46   Min.   :14.04   Min.   :0.05885  
##  1st Qu.: 5.290   1st Qu.:42.20   1st Qu.:15.19   1st Qu.:0.07137  
##  Median : 7.463   Median :43.10   Median :15.93   Median :0.07536  
##  Mean   : 8.496   Mean   :43.59   Mean   :16.27   Mean   :0.07681  
##  3rd Qu.:11.593   3rd Qu.:44.57   3rd Qu.:16.98   3rd Qu.:0.07964  
##  Max.   :18.204   Max.   :51.60   Max.   :22.14   Max.   :0.15603  
##    Bare_NDVI            DEM      
##  Min.   :0.08476   Min.   :1559  
##  1st Qu.:0.10078   1st Qu.:1561  
##  Median :0.10790   Median :1562  
##  Mean   :0.11003   Mean   :1562  
##  3rd Qu.:0.11478   3rd Qu.:1563  
##  Max.   :0.22755   Max.   :1565  
```

```r
#Many of these predictors are significant
#The r square is 0.7291 for all 5 regresses

#plot the fitted variables against their response variables >> #To check the scatter and if it's noisy.

plot(N_lm$fitted.values ~ N_prediction$Nitrate_N, main = "fitted vs response")
```



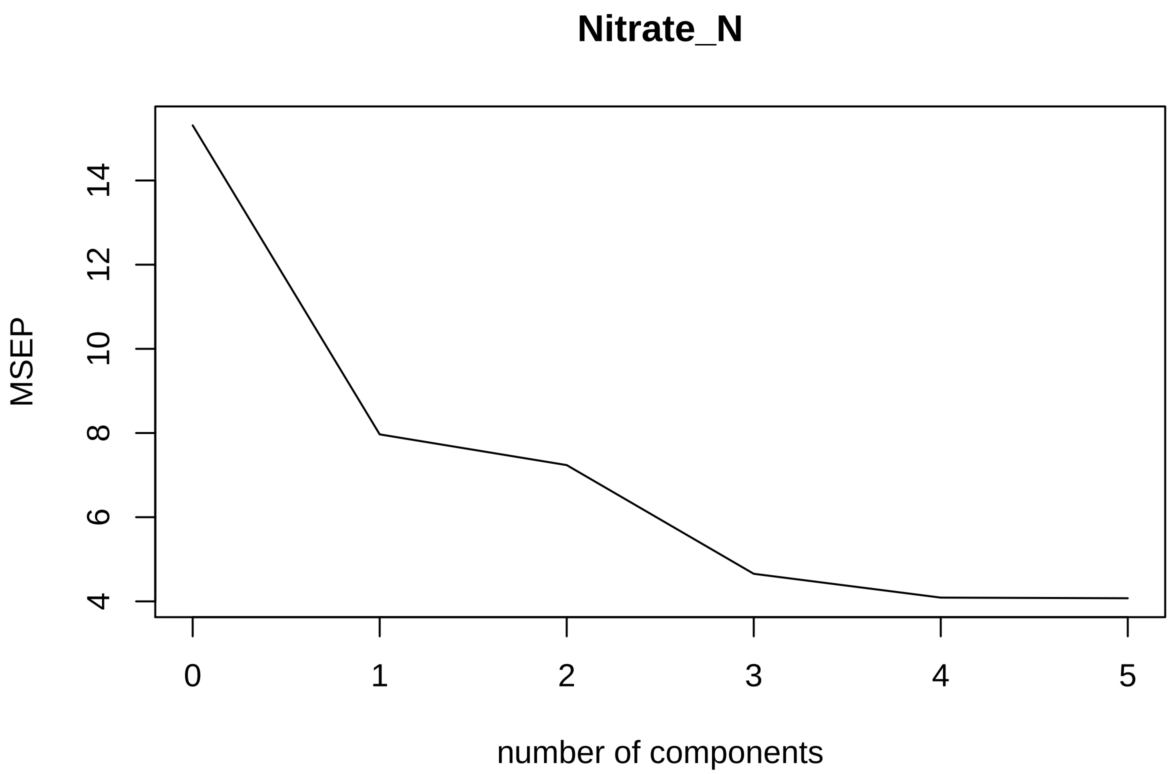
```
library (pls)
```

```
##
## Attaching package: 'pls'

## The following object is masked from 'package:stats':
##
##     loadings
```

```
#partial least square
#N_plsr = plsr(Nitrate_N ~ Eca_deep + Eca_shallow + Bare_SAVI + Bare_NDVI + DEM , #data = N_prediction,
#validation procedure - standard cross validation
#summary(N_plsr)
#shows us the percentage variance explained in our Nitrate N
#plsr mapping a linear combination of x variables and use that to optimize predicting against y variabl
#plot(RMSEP(N_plsr))



#subsetting
#training and test data
#Split the data into training (60%) and "test" (40%) set randomly.

plsr_data_OM = sample(nrow(N_prediction), nrow(N_prediction)*.6)
plsr_training = N_prediction[plsr_data_OM,]
plsr_test = N_prediction[-plsr_data_OM,]

#number of component chosen is 5
#training
N_plsr_training = plsr(Nitrate_N ~ Eca_deep + Eca_shallow + Bare_SAVI + Bare_NDVI + DEM , data = plsr_t

#validation procedure - standard cross validation
summary(N_plsr_training)
```

```
## Data:    X dimension: 406 5
##  Y dimension: 406 1
## Fit method: kernelpls
## Number of components considered: 5
## TRAINING: % variance explained
##            1 comps  2 comps  3 comps  4 comps  5 comps
## X            49.60    59.11    67.27    98.67   100.00
## Nitrate_N    47.96    52.73    69.60    73.29    73.39
```

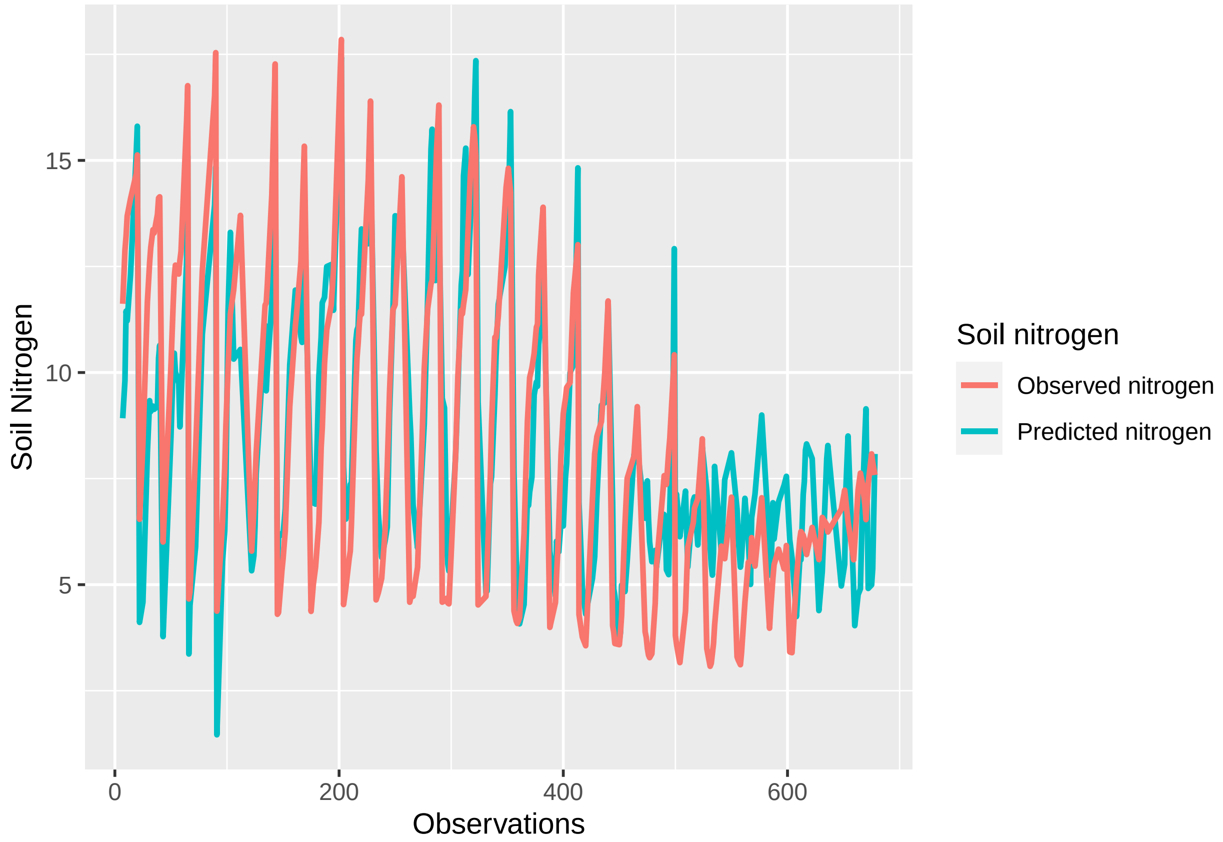
```
validationplot(N_plsr_training, val.type = "MSEP")
```



```
#test >>> predict
plsr_test$plsr_N_fitted = predict(N_plsr_training, newdata=plsr_test, ncomp= 4)

average_test = mean(plsr_test$Nitrate_N)

1-mean((plsr_test$plsr_N_fitted- plsr_test$Nitrate_N)^2)/mean((average_test - plsr_test$Nitrate_N)^2)
```

```
## [1] 0.715134
```

```
ggplot()+
  geom_line(data = plsr_test, aes(y=plsr_N_fitted, x=OBJECTID_1, colour = "red"), size =1) +
  geom_line(data = plsr_test, aes(y=Nitrate_N, x=OBJECTID_1,  colour="blue"), size =1) +
  xlab("Observations")+
  ylab("Soil Nitrogen")+
  scale_color_discrete(name = "Soil nitrogen", labels=c("Observed nitrogen", "Predicted nitrogen"))
```



```
set.seed(445)
#OM prediction based on ECa and VI based
OM_prediction = read.csv("Data_4_machine learning_PLSR.csv")
str(OM_prediction)
```

```
## 'data.frame':    678 obs. of  10 variables:
##  $ OBJECTID_1 : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ POINT_X    : num  -105 -105 -105 -105 -105 ...
##  $ POINT_Y    : num  40.7 40.7 40.7 40.7 40.7 ...
##  $ OM         : num  1.15 1.15 1.17 1.21 1.26 ...
##  $ Nitrate_N  : num  6.88 7.31 8.37 8.92 9.58 ...
##  $ Eca_deep   : num  40.5 40.2 40.3 40.8 41.5 ...
##  $ Eca_shallow: num  14.1 14.2 14.4 14.7 14.8 ...
##  $ Bare_SAVI  : num  0.109 0.11 0.108 0.113 0.109 ...
##  $ Bare_NDVI  : num  0.152 0.153 0.15 0.156 0.153 ...
##  $ DEM        : int  1563 1563 1563 1563 1563 1563 1563 1563 1563 1562 ...
```

```
OM_lm = lm(OM ~ Eca_deep + Eca_shallow + Bare_SAVI + Bare_NDVI + DEM , data = OM_prediction)
summary(OM_lm)
```

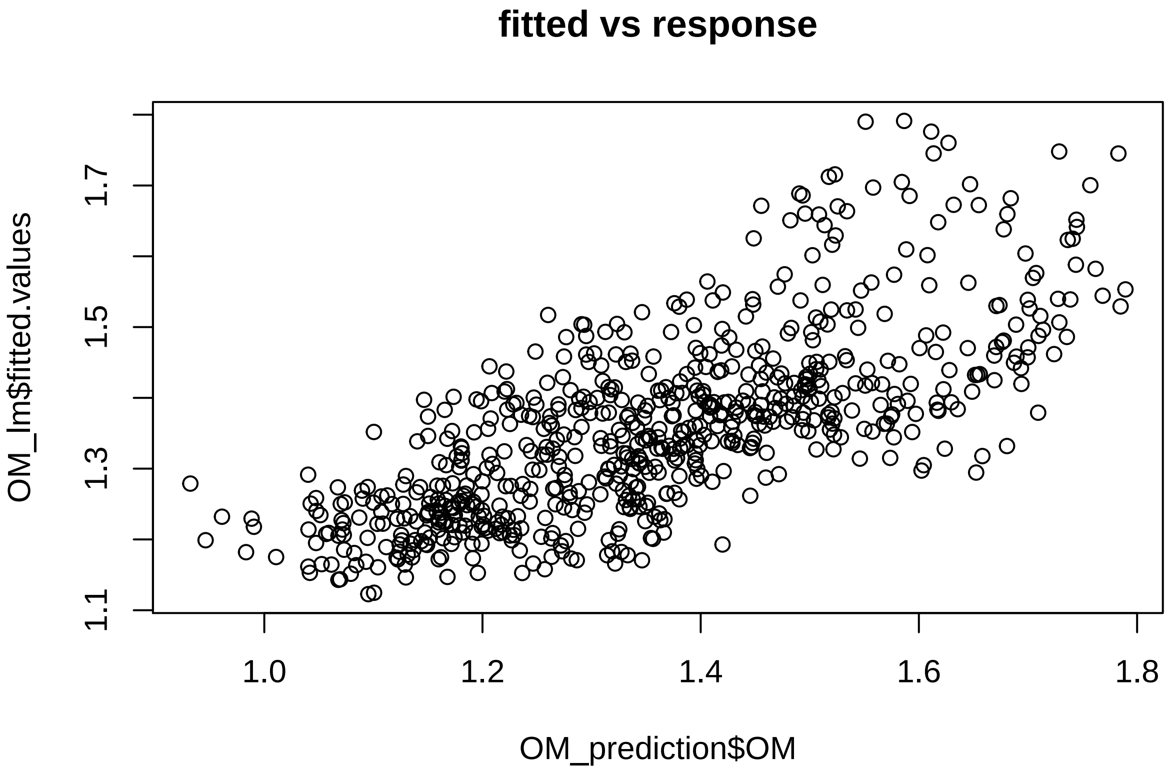
```
##
## Call:
## lm(formula = OM ~ Eca_deep + Eca_shallow + Bare_SAVI + Bare_NDVI +
##     DEM, data = OM_prediction)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.34669 -0.09261 -0.00488  0.08372  0.35801
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.426449   8.294231   1.016  0.31002
## Eca_deep    -0.051096   0.006071  -8.417 2.32e-16 ***
## Eca_shallow  0.133503   0.008335  16.017  < 2e-16 ***
## Bare_SAVI   -0.721800   1.970135  -0.366  0.71420
## Bare_NDVI    4.068788   1.378031   2.953  0.00326 **
## DEM         -0.004742   0.005250  -0.903  0.36677
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1231 on 672 degrees of freedom
## Multiple R-squared:  0.5268, Adjusted R-squared:  0.5233
## F-statistic: 149.6 on 5 and 672 DF,  p-value: < 2.2e-16
```

```r
summary(OM_prediction)
```

```
##    OBJECTID_1       POINT_X        POINT_Y            OM
##  Min.   :  1.0   Min.   :-105   Min.   :40.66   Min.   :0.9322
##  1st Qu.:170.2   1st Qu.:-105   1st Qu.:40.67   1st Qu.:1.2161
##  Median :339.5   Median :-105   Median :40.67   Median :1.3492
##  Mean   :339.5   Mean   :-105   Mean   :40.67   Mean   :1.3583
##  3rd Qu.:508.8   3rd Qu.:-105   3rd Qu.:40.67   3rd Qu.:1.4856
##  Max.   :678.0   Max.   :-105   Max.   :40.67   Max.   :1.7893
##    Nitrate_N         Eca_deep      Eca_shallow      Bare_SAVI
##  Min.   : 3.076   Min.   :39.46   Min.   :14.04   Min.   :0.05885
##  1st Qu.: 5.290   1st Qu.:42.20   1st Qu.:15.19   1st Qu.:0.07137
##  Median : 7.463   Median :43.10   Median :15.93   Median :0.07536
##  Mean   : 8.496   Mean   :43.59   Mean   :16.27   Mean   :0.07681
##  3rd Qu.:11.593   3rd Qu.:44.57   3rd Qu.:16.98   3rd Qu.:0.07964
##  Max.   :18.204   Max.   :51.60   Max.   :22.14   Max.   :0.15603
##    Bare_NDVI            DEM
##  Min.   :0.08476   Min.   :1559
##  1st Qu.:0.10078   1st Qu.:1561
##  Median :0.10790   Median :1562
##  Mean   :0.11003   Mean   :1562
##  3rd Qu.:0.11478   3rd Qu.:1563
##  Max.   :0.22755   Max.   :1565
```

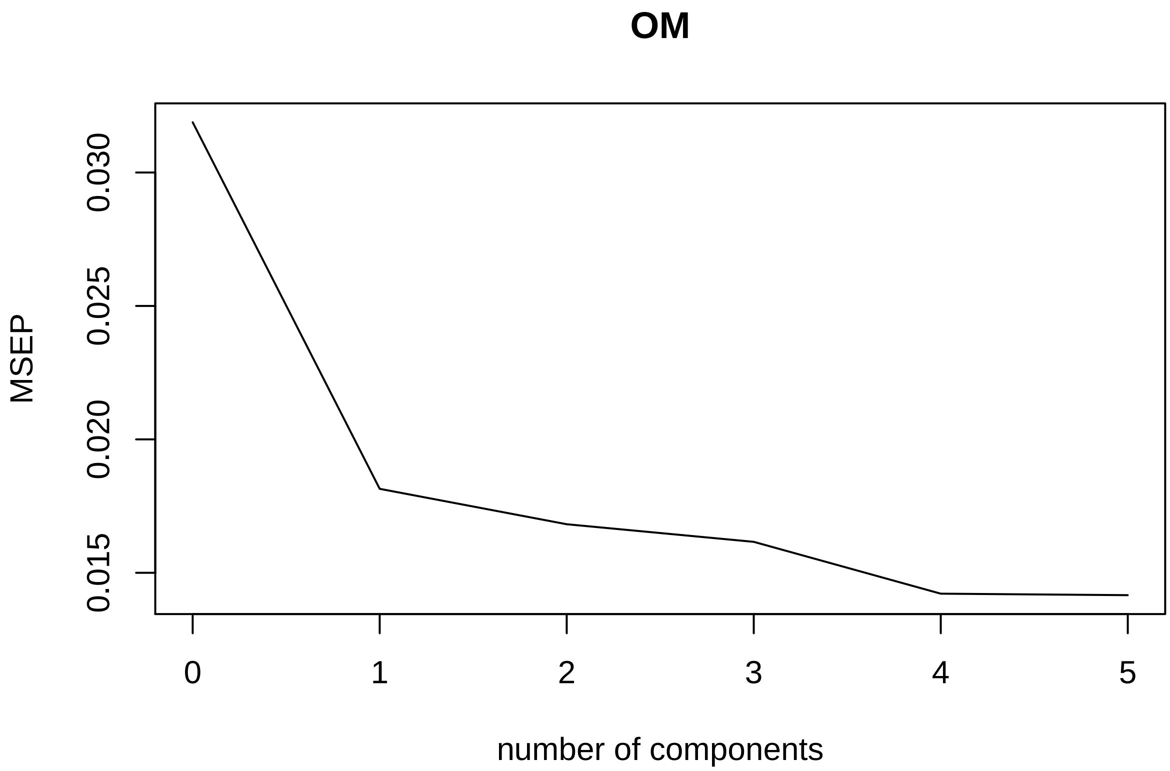
```r
#Many of these predictors are significant
#The r square is 0.7291 for all 5 regresses

#plot the fitted variables against their response variables >> #To check the scatter and if it's noisy.

plot(OM_lm$fitted.values ~ OM_prediction$OM, main = "fitted vs response")
```



```
library (pls)

#partial least square
#OM_plsr = plsr(OM ~ Eca_deep + Eca_shallow + Bare_SAVI + Bare_NDVI + DEM , data = #OM_prediction, scal
#validation procedure - standard cross validation
#summary(OM_plsr)
#shows us the percentage variance explained in our Nitrate N
#plsr mapping a linear combination of x variables and use that to optimize predicting against y variabl
#plot(RMSEP(OM_plsr))



#subsetting
#training and test data
#Split the data into training (60%) and "test" (40%) set randomly.

plsr_data_OM= sample(nrow(OM_prediction), nrow(OM_prediction)*.6)
plsr_training_OM_data = OM_prediction[plsr_data_OM,]
plsr_test_OM_data = OM_prediction[-plsr_data_OM,]

#number of component chosen is 5
#training
OM_plsr_training = plsr(OM ~ Eca_deep + Eca_shallow + Bare_SAVI + Bare_NDVI + DEM , data = plsr_trainin

#validation procedure - standard cross validation
summary(OM_plsr_training)
```

```
## Data:    X dimension: 406 5
##  Y dimension: 406 1
## Fit method: kernelpls
## Number of components considered: 5
## TRAINING: % variance explained
##      1 comps  2 comps  3 comps  4 comps  5 comps
## X      49.17    69.31    96.92    99.48   100.00
## OM     43.08    47.25    49.31    55.41    55.58
```

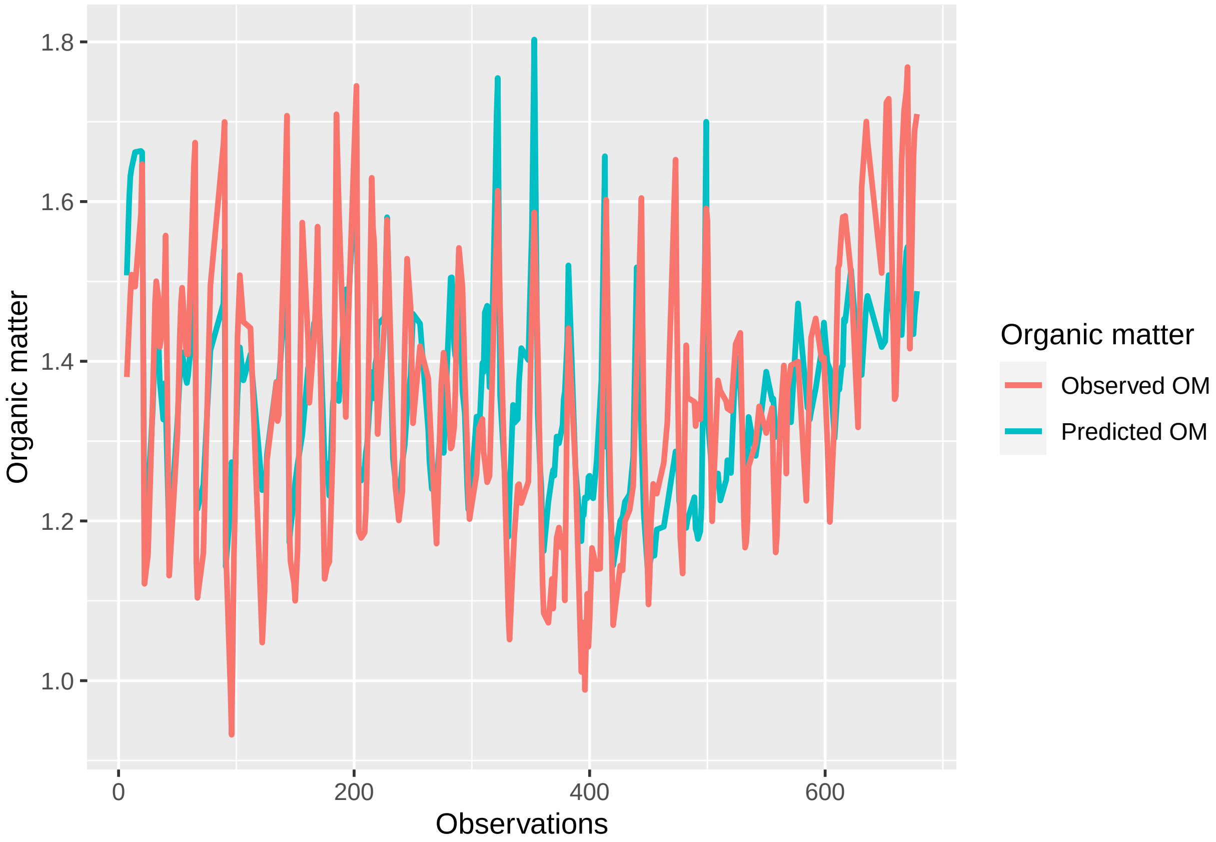
```
validationplot(OM_plsr_training, val.type = "MSEP")
```



```
#test >>> predict
plsr_test_OM_data$plsr_OM_fitted = predict(OM_plsr_training, newdata=plsr_test_OM_data, ncomp= 4)

average_test = mean(plsr_test_OM_data$OM)

1- mean((plsr_test_OM_data$plsr_OM_fitted - plsr_test_OM_data$OM)^2)/mean((average_test - plsr_test_OM_
```

```
## [1] 0.4801307
```

```
ggplot()+
  geom_line(data = plsr_test_OM_data, aes(y=plsr_OM_fitted, x=OBJECTID_1, colour = "red"), size =1) +
  geom_line(data = plsr_test_OM_data, aes(y=OM, x=OBJECTID_1,  colour="blue"), size =1) +
  xlab("Observations")+
  ylab("Organic matter")+
  scale_color_discrete(name = "Organic matter", labels=c("Observed OM", "Predicted OM"))
```



## Nitrogen and organic matter prediction results

1. A linear regression was done to understand the data from a linear perspective for both nitrogen and OM as a response variable.
2. The multiple r-square of the linear regression are 0.729 and 0.5268 respectively for nitrogen and OM.
3. Nitrogen was 71.51% accurately predicted, nitrogen prediction shows that the trend between observed and predicted nitrogen is promising.
4. Organic matter was 48.01% accurately predicted, based on the visual assessment of Organic matter prediction figure, the trend between observed and predicted organic matter is not promising.
5. It’s possible to conclude that PLSR worked better for nitrogen prediction than organic matter based on combination of vegetation indices (NDVI and SAVI), electro-conductivity data and elevation information.

## References:

1. Montzka SA, Dlugokencky EJ, Butler JH. Non-CO 2 greenhouse gases and. Nature. Published online 2011:0-7. doi:10.1038/nature10322
2. Nawar S, Corstanje R, Halcro G, Mulla D, Mouazen AM. Delineation of Soil Management Zones for Variable-Rate Fertilization: A Review. Vol 143. 1st ed. Elsevier Inc.; 2017. doi:10.1016/bs.agron.2017.01.003
3. Galambošová J, Rataj V, Prokeinová R, Prešinská J. Determining the management zones with hierarchic and non-hierarchic clustering methods. Res Agric Eng. 2014;60(2000):S44-S51. doi:10.17221/34/2013-rae
4. Saifuzzamna M, Adamchuk V, Huang H, Ji W. Data Clustering Tools for Understanding Spatial Heterogeneity in Crop Production by Integrating Proximal Soil Sensing and Remote Sensing Data. Int Conf Precis Agric. Published online 2018:1-14.

5. Gavioli A, de Souza EG, Bazzi CL, Schenatto K, Betzek NM. Identification of management zones in precision agriculture: An evaluation of alternative cluster analysis methods. Biosyst Eng. 2019;181:86-102. doi:10.1016/j.biosystemseng.2019.02.019
6. Gmur S, Vogt D, Zabowski D, Monika Moskal L. Hyperspectral analysis of soil nitrogen, carbon, carbonate, and organic matter using regression trees. Sensors (Switzerland). 2012;12(8):10639-10658. doi:10.3390/s120810639

### Code concept support from the following site:

https://rpubs.com/chrisbrunsdon/99675

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

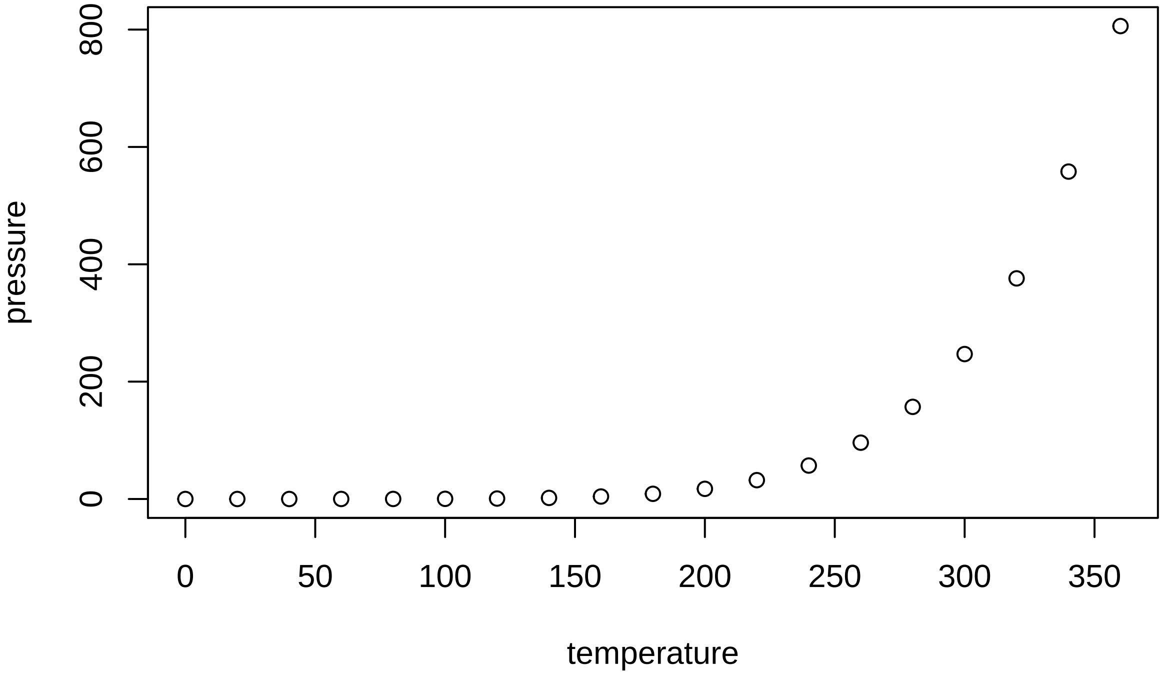
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```r
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

## Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.