PCA

Carol

2023-01-22

Working directory, packages, and data

1.Set working directory

```
setwd("/Users/Carol's PC/Documents/KIRAoccModel/PCA")
```

2. Packages

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## v purrr   1.0.1
```

```
## Warning: package 'tibble' was built under R version 4.2.2
```

```
## Warning: package 'purrr' was built under R version 4.2.2
```

```
## Warning: package 'dplyr' was built under R version 4.2.2
```

```
## Warning: package 'stringr' was built under R version 4.2.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(DataExplorer)
```

```
## Warning: package 'DataExplorer' was built under R version 4.2.2
```

```
library(devtools)
```

```
## Warning: package 'devtools' was built under R version 4.2.2
```

```
## Loading required package: usethis
```

```
## Warning: package 'usethis' was built under R version 4.2.2
```

```
library(ggbiplot)
```

```
## Loading required package: plyr
```

```
## Warning: package 'plyr' was built under R version 4.2.2
```

```
## ------------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## ------------------------------------------------------------------------------
##
## Attaching package: 'plyr'
##
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
##
## The following object is masked from 'package:purrr':
##
##     compact
##
## Loading required package: scales
```

```
## Warning: package 'scales' was built under R version 4.2.2
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
##
## Loading required package: grid
```

3. Data

```r
data<-read.csv("PCAdata.csv", header = TRUE)

head(data) # Make sure data looks correct
```

```
##   Location MaxDetections Avg_ht.m. Canopy m_toOpenWater Juncus Typha Phrag
## 1  BackBay             3    1.6764      5             0     15    50    30
## 2  BackBay             0    1.3716     18             0      0    38     5
## 3  BackBay             1    1.6764     17             0     23     0     9
## 4  BackBay             1    1.3716      6            15      7    15    10
## 5  BackBay             3    1.6764      0             0     10    25    60
## 6  BackBay             2    1.2954      9             0      5    12    45
##   Grasses Schoenoplectus Trees.shrubs MixedEmergents Management0_1
## 1       0              0            5              0             1
## 2       4             33           16              4             1
## 3      13             15           22             18             1
## 4      53              0            7              8             1
## 5       0              0            1              4             1
## 6      26              2            8              2             1
```

Exploratory data analysis

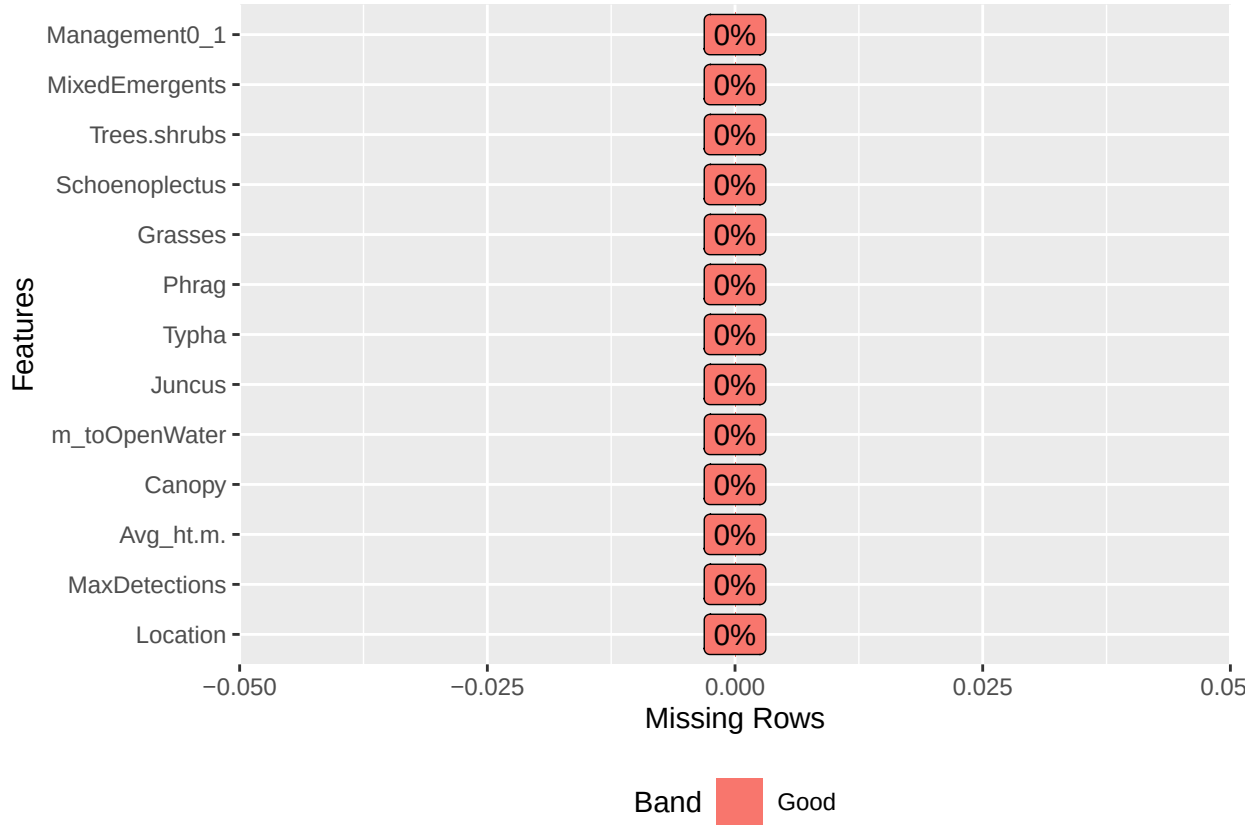
1. Ensure data is formatted correctly

```r
data %>% glimpse() # Basic overview of data. This helps me catch mistakes early.
```

```
## Rows: 58
## Columns: 13
## $ Location       <chr> "BackBay", "BackBay", "BackBay", "BackBay", "BackBay", ~
## $ MaxDetections  <int> 3, 0, 1, 1, 3, 2, 3, 4, 3, 1, 0, 4, 2, 1, 0, 3, 5, 5, 0~
## $ Avg_ht.m.      <dbl> 1.6764, 1.3716, 1.6764, 1.3716, 1.6764, 1.2954, 0.9144,~
## $ Canopy         <int> 5, 18, 17, 6, 0, 9, 0, 7, 8, 15, 3, 3, 0, 5, 15, 1, 2, ~
## $ m_toOpenWater  <int> 0, 0, 0, 15, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ Juncus         <int> 15, 0, 23, 7, 10, 5, 5, 30, 20, 20, 25, 4, 0, 9, 12, 24~
## $ Typha          <int> 50, 38, 0, 15, 25, 12, 10, 20, 10, 35, 0, 27, 45, 0, 0,~
## $ Phrag          <int> 30, 5, 9, 10, 60, 45, 40, 15, 55, 20, 17, 21, 15, 50, 2~
## $ Grasses        <dbl> 0.00, 4.00, 13.00, 53.00, 0.00, 26.00, 25.00, 12.00, 3.~
## $ Schoenoplectus <int> 0, 33, 15, 0, 0, 2, 0, 0, 2, 0, 38, 40, 0, 0, 2, 0, 2, ~
## $ Trees.shrubs   <dbl> 5.0, 16.0, 22.0, 7.0, 1.0, 8.0, 0.0, 7.0, 7.0, 15.0, 3.~
## $ MixedEmergents <dbl> 0.00, 4.00, 18.00, 8.00, 4.00, 2.00, 20.00, 16.00, 3.00~
## $ Management0_1  <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0~
```

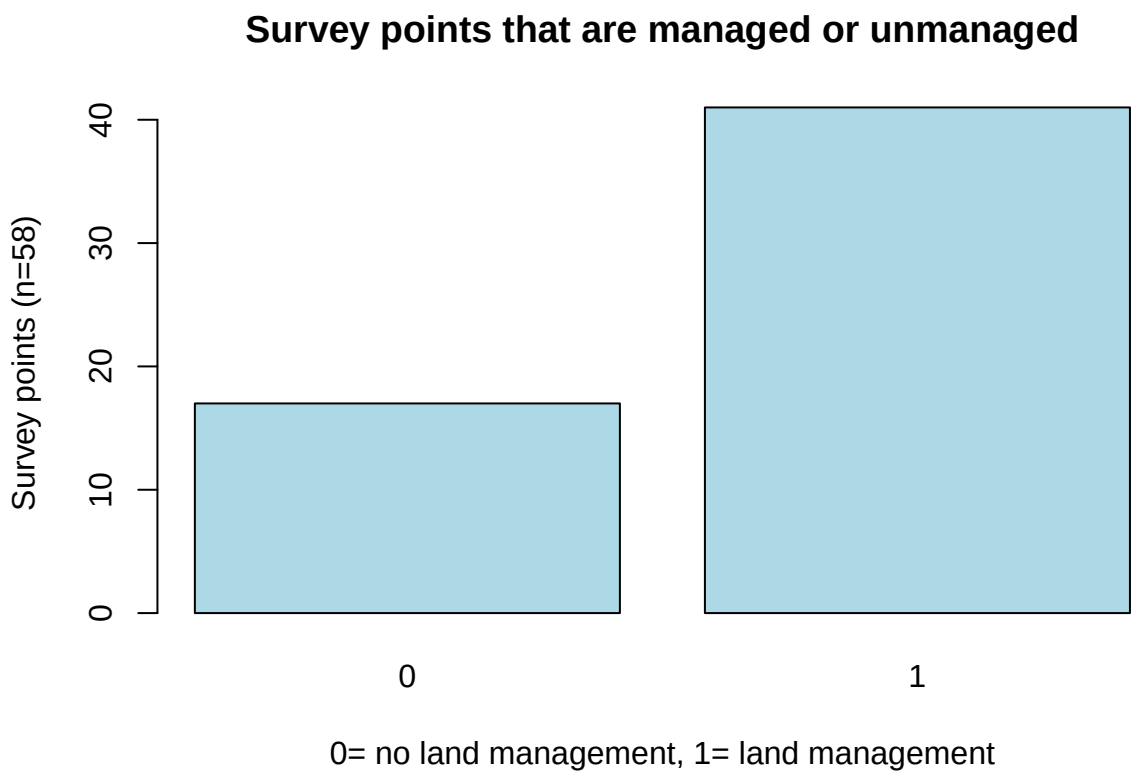
```r
data %>% introduce()
```

```
##   rows columns discrete_columns continuous_columns all_missing_columns
## 1   58      13                1                  12                   0
##   total_missing_values complete_rows total_observations memory_usage
## 1                    0            58                754         8064
```



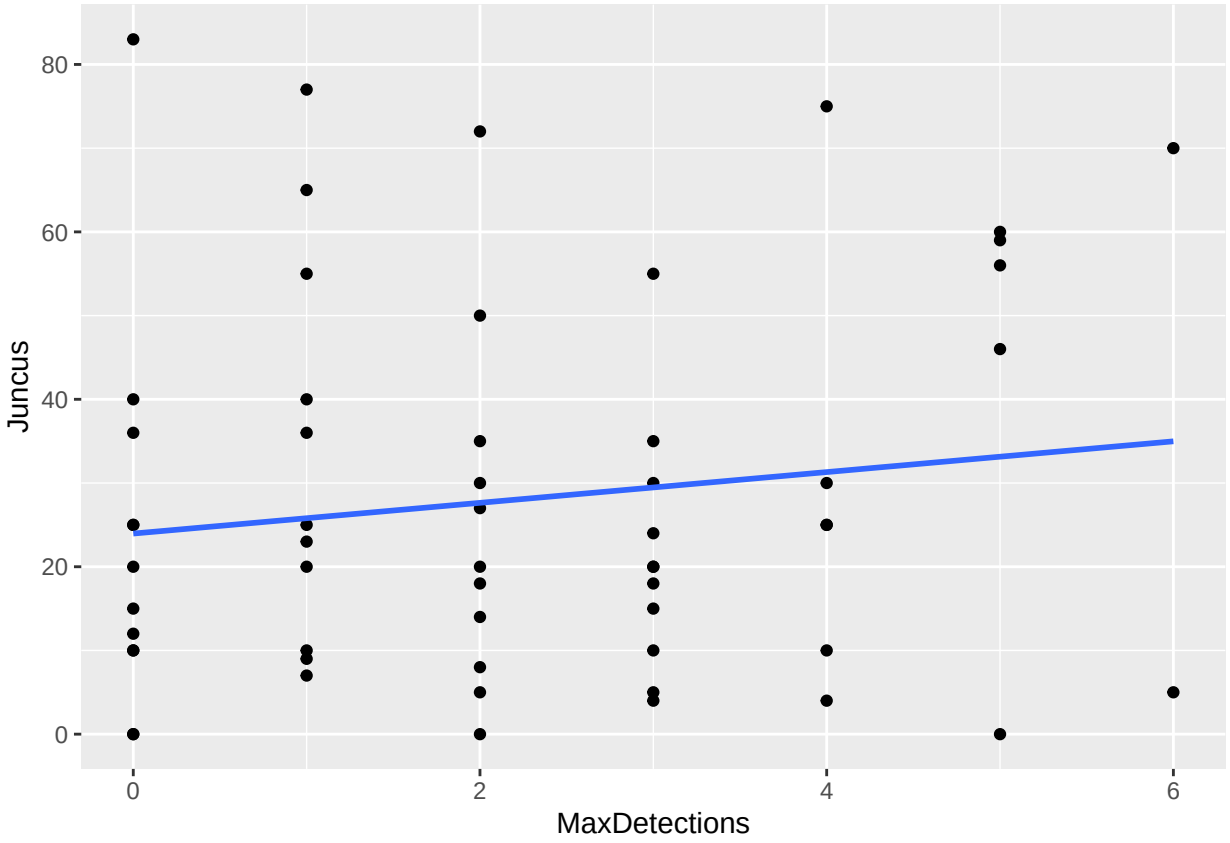
2. Managment as a binary variable

```
Management <- table(data$Management0_1)

barplot(Management,
     main="Survey points that are managed or unmanaged",
     xlab="0= no land management, 1= land management",
     ylab="Survey points (n=58)",
     col="lightblue"
)
```



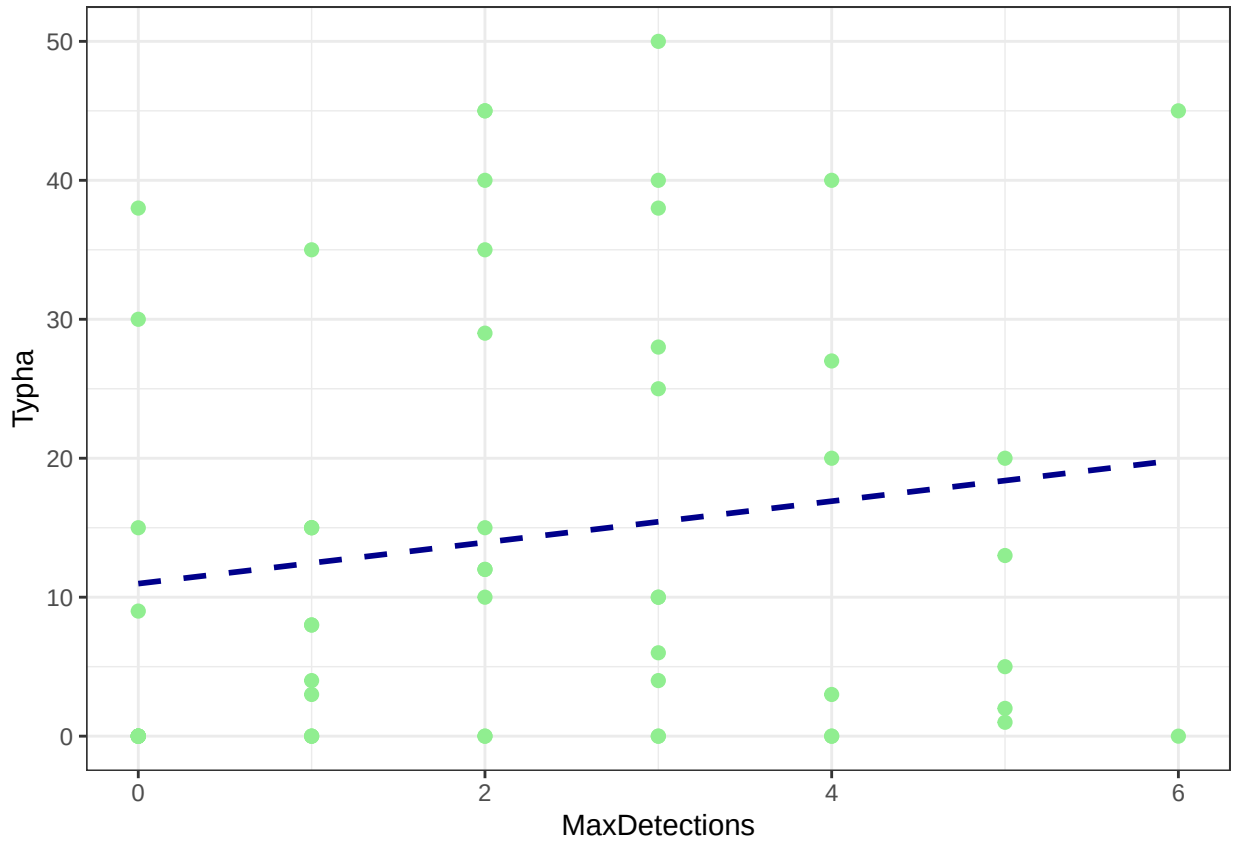

```
ggplot(data, aes(x=MaxDetections, y=Juncus)) +
    geom_point() +
    geom_smooth(method=lm, se=FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



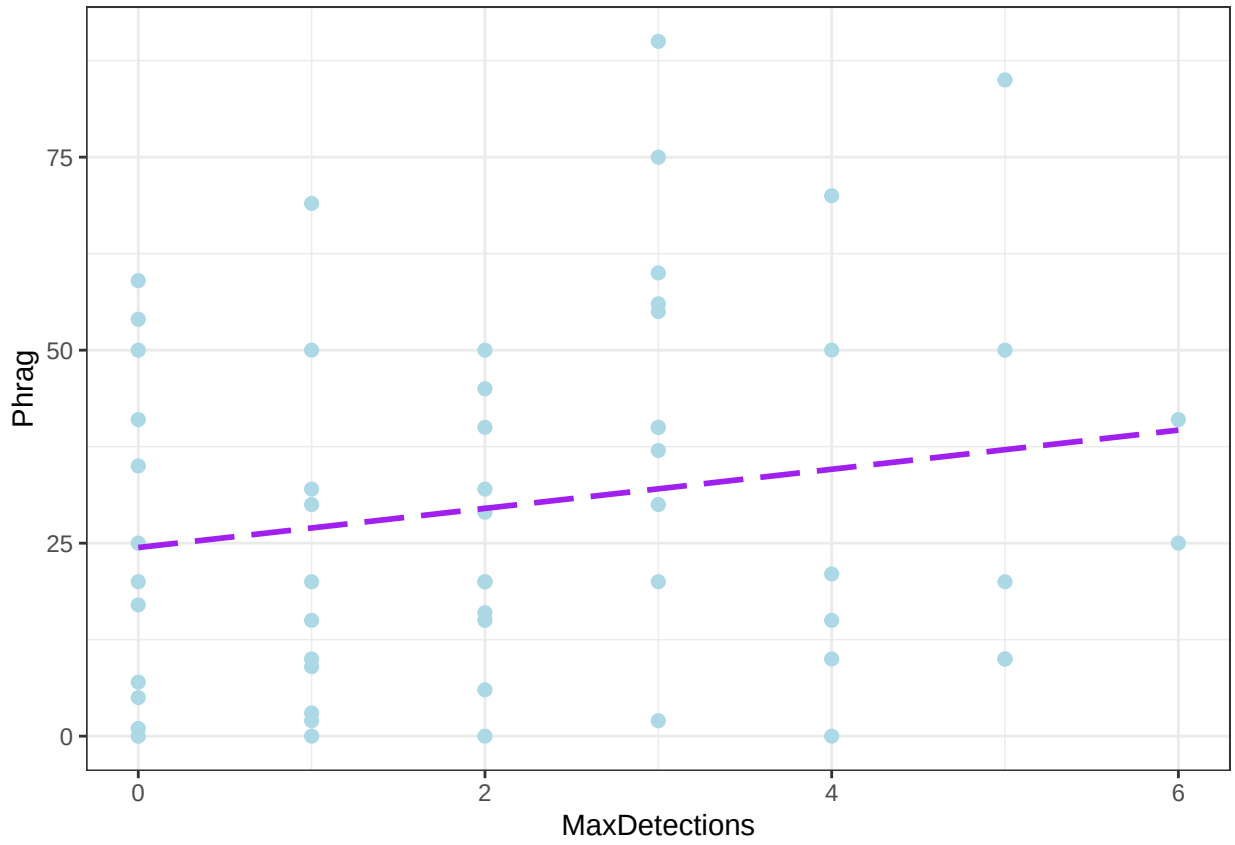

```
ggplot(data, aes(x=MaxDetections, y=Typha)) +
   geom_point(col='lightgreen', size=2) +
    geom_smooth(method=lm, se=FALSE, col='darkblue', linetype='dashed') +
    theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



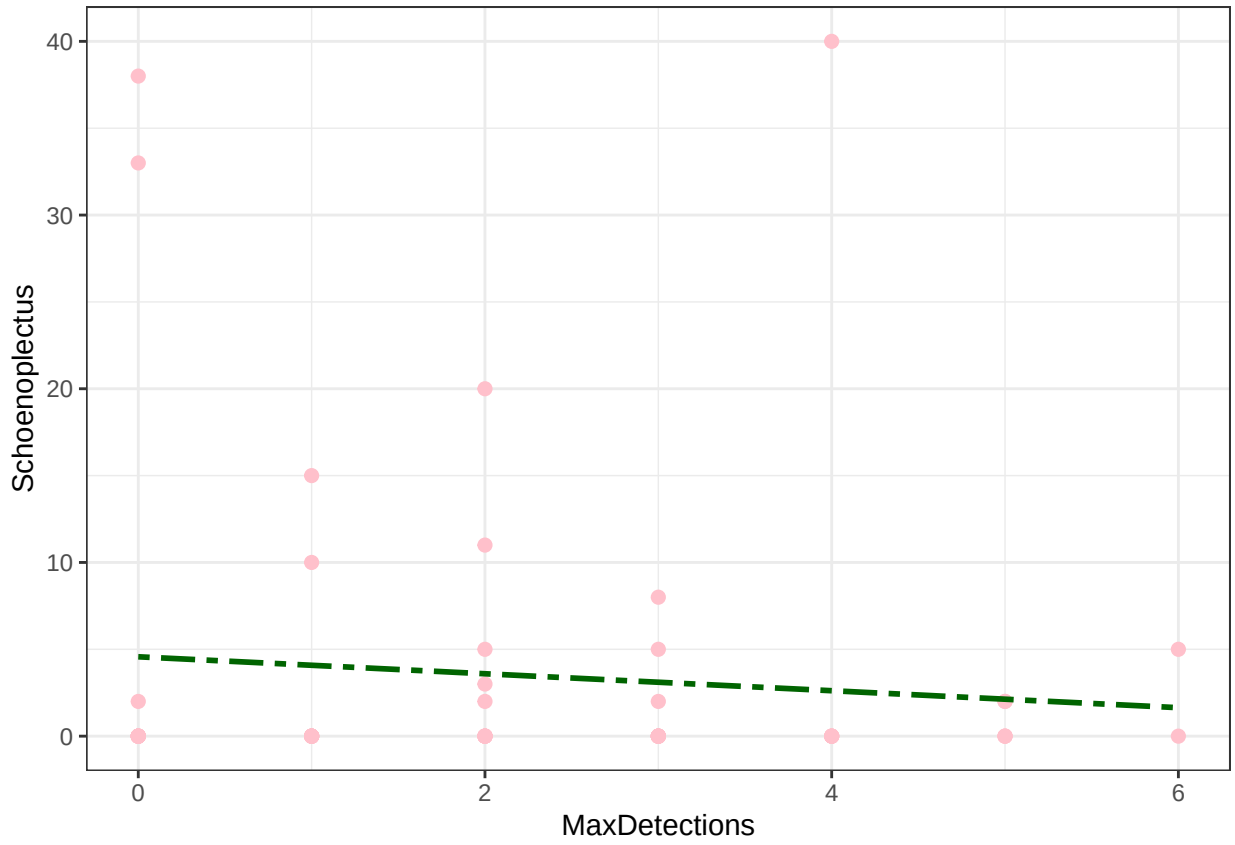

```
ggplot(data, aes(x=MaxDetections, y=Phrag)) +
   geom_point(col='lightblue', size=2) +
    geom_smooth(method=lm, se=FALSE, col='purple', linetype=5 ) +
    theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



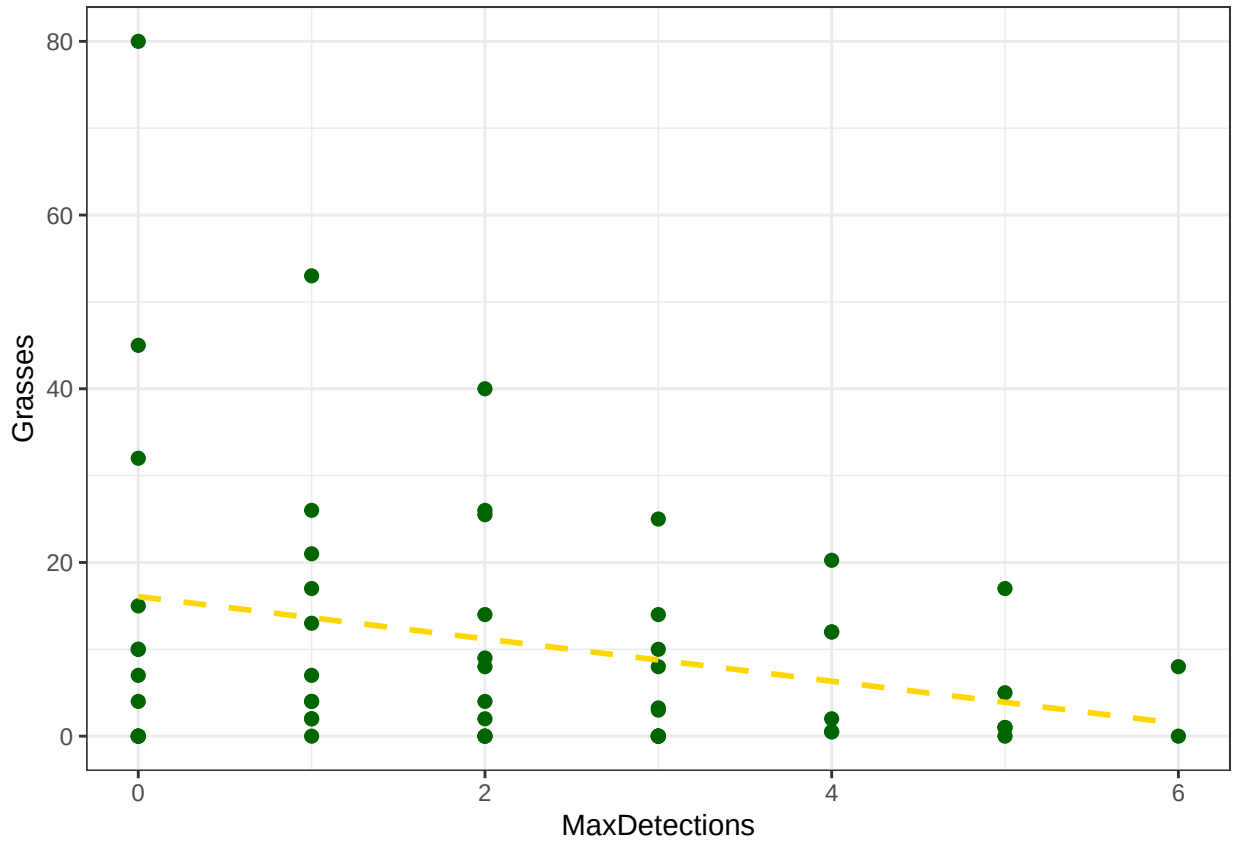

```
ggplot(data, aes(x=MaxDetections, y=Schoenoplectus)) +
   geom_point(col='pink', size=2) +
    geom_smooth(method=lm, se=FALSE, col='darkgreen', linetype=6 ) +
    theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



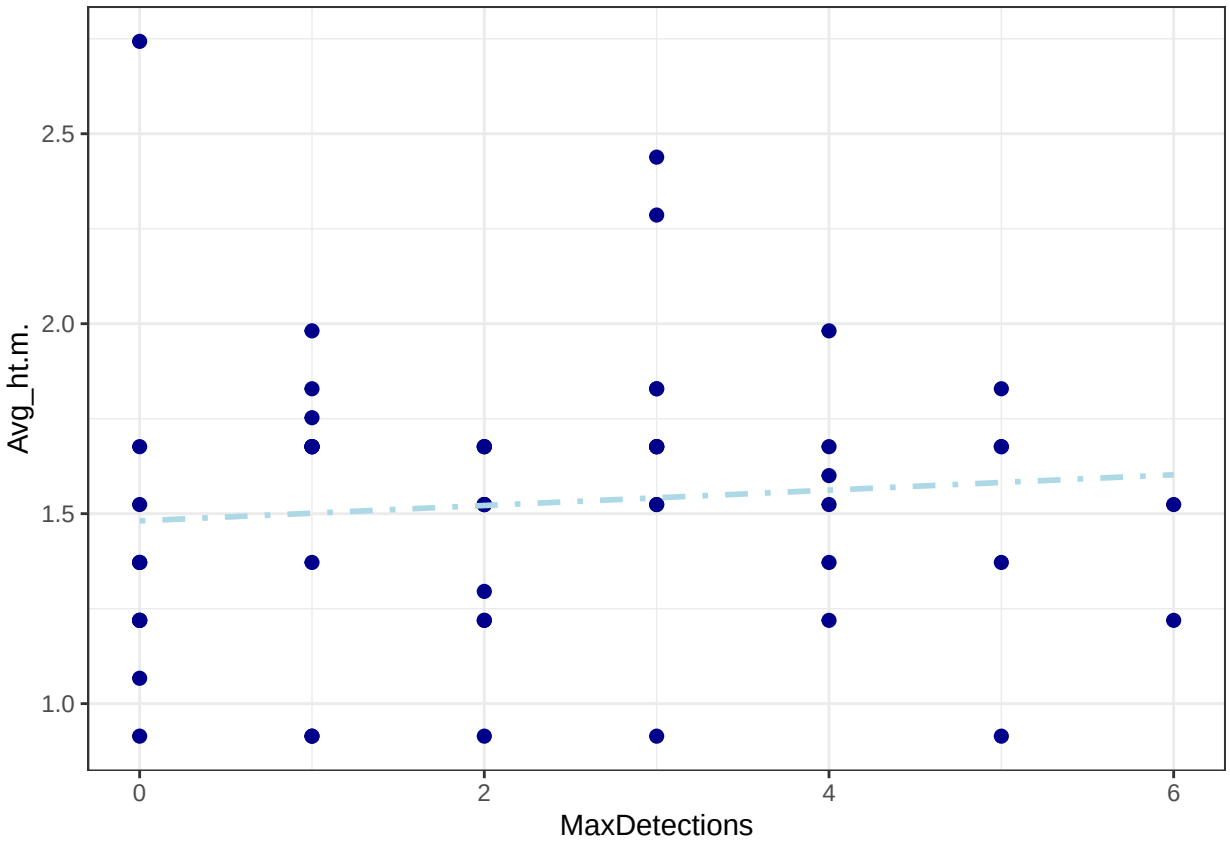

```
ggplot(data, aes(x=MaxDetections, y=Grasses)) +
   geom_point(col='darkgreen', size=2) +
    geom_smooth(method=lm, se=FALSE, col='gold', linetype=2 ) +
    theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



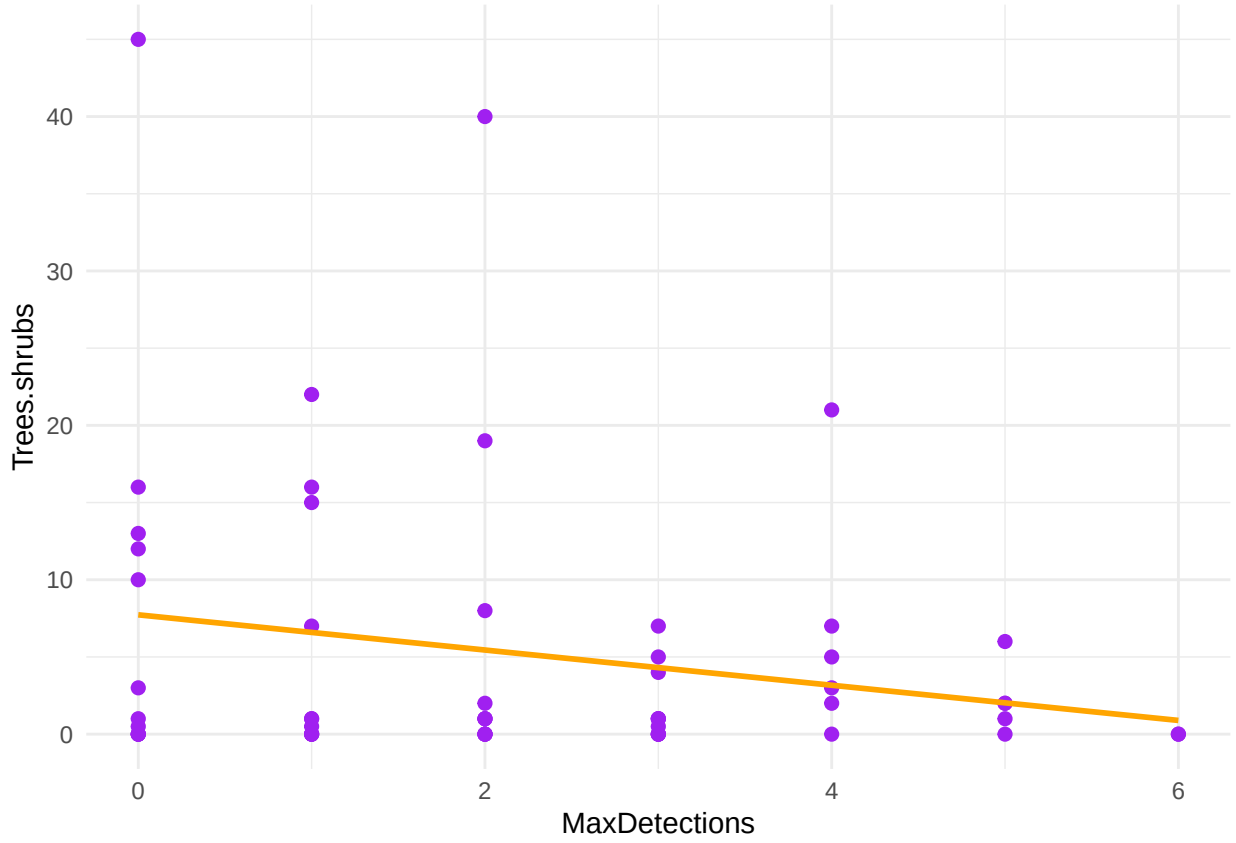

```
ggplot(data, aes(x=MaxDetections, y=Avg_ht.m., title(main = "Detections x Avg. veg. height (m)"))) +
   geom_point(col='darkblue', size=2) +
    geom_smooth(method=lm, se=FALSE, col='lightblue', linetype=4 ) +
    theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



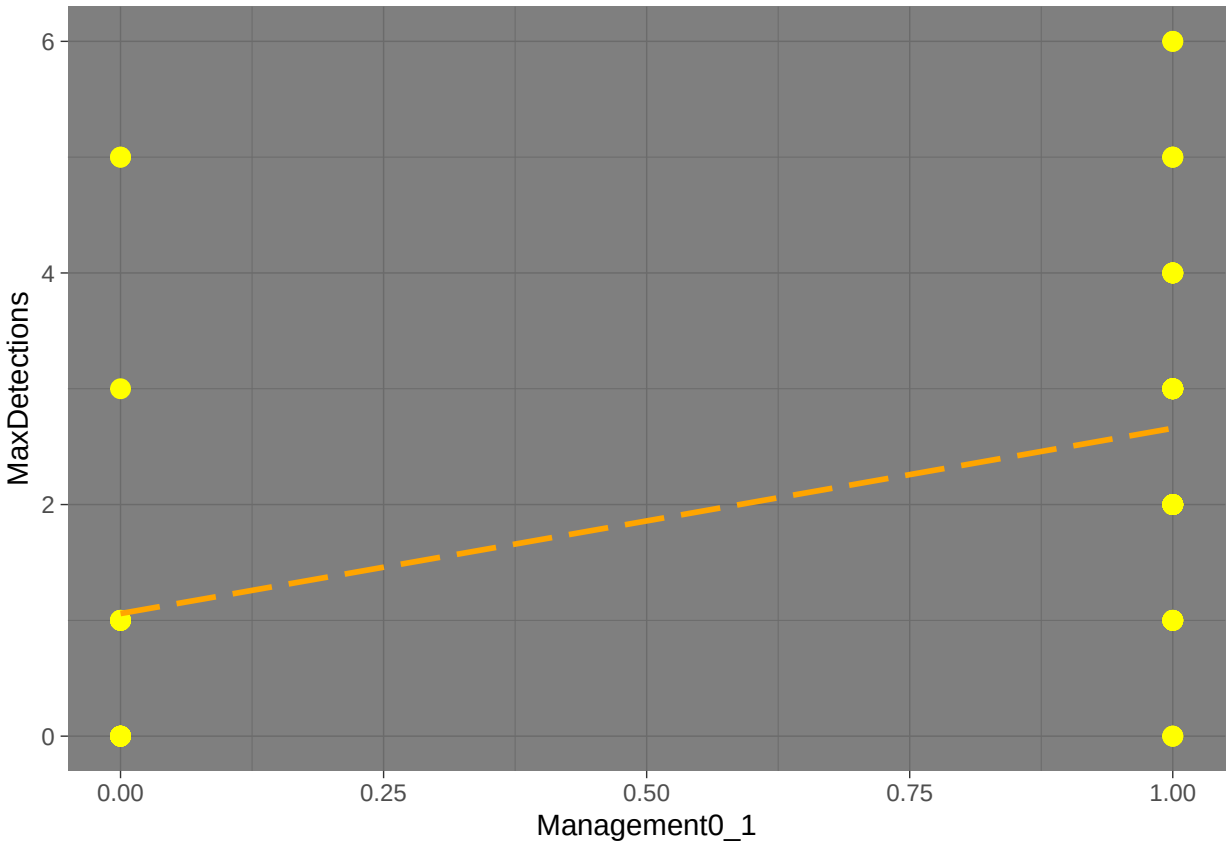

```
ggplot(data, aes(x=MaxDetections, y=Trees.shrubs, title(main = "Detections x Trees"))) +
   geom_point(col='purple', size=2) +
    geom_smooth(method=lm, se=FALSE, col='orange', linetype=1 ) +
    theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(data, aes(x=Management0_1, y=MaxDetections, title(main = "Detections x management as a binary va
   geom_point(col='yellow', size=3) +
    geom_smooth(method=lm, se=FALSE, col='orange', linetype=5 ) +
    theme_dark()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



PCA

1. Scale

```
data <- data[,-1] # Remove location names column
# Scale for normalization

data$MaxDetections<-scale(data$MaxDetections)
data$Avg_ht.m.<-scale(data$Avg_ht.m.)
data$Juncus<-scale(data$Juncus)
data$Canopy<-scale(data$Canopy)
data$m_toOpenWater<-scale(data$m_toOpenWater)
data$Typha<-scale(data$Typha)
data$Phrag<-scale(data$Phrag)
data$Grasses<-scale(data$Grasses)
data$Schoenoplectus<-scale(data$Trees.shrubs)
data$Trees.shrubs<-scale(data$Schoenoplectus)
data$MixedEmergents<-scale(data$MixedEmergents)
data$Management0_1<-scale(data$Management0_1)

# Save scaled data for future time saving

# write.csv(data, "data_scaled.csv")
```

2. Calculate the Principal Components

```
# Calculate principal components
PC <- prcomp(data, scale = TRUE)

# Reverse the signs. Note:eigenvectors in R point in the negative direction by default, so multiply by
PC$rotation <- -1*PC$rotation

# Display principal components
PC$rotation
```

```
##                        PC1          PC2         PC3         PC4          PC5
## MaxDetections   0.08315233 -0.419665889  0.27137109 -0.14288058  0.140232998
## Avg_ht.m.      -0.28000300 -0.403940579 -0.27167277  0.07029344 -0.079527545
## Canopy         -0.43859287  0.203547694  0.09656550  0.03649314  0.045690862
## m_toOpenWater  -0.37091202 -0.050401975 -0.06640052 -0.07056252 -0.008198351
## Juncus          0.11035250  0.008628325  0.05726087 -0.77934753  0.106877659
## Typha           0.11209040 -0.130571608  0.52968456  0.41593172 -0.223042868
## Phrag          -0.10654058 -0.407253654 -0.47382547  0.22231290 -0.141498539
## Grasses         0.08062371  0.315366774 -0.11729723  0.35217088  0.739189236
## Schoenoplectus -0.49594053  0.179834638  0.11368313 -0.05171193 -0.041283101
## Trees.shrubs   -0.49594053  0.179834638  0.11368313 -0.05171193 -0.041283101
## MixedEmergents  0.16472065  0.419507856  0.04167737  0.07892444 -0.553706675
## Management0_1  -0.15200215 -0.306002546  0.53769809  0.05409910  0.185100024
##                        PC6         PC7         PC8         PC9         PC10
## MaxDetections  -0.388529385  0.58114963 -0.41942292  0.17323996  0.068963983
## Avg_ht.m.       0.246119052 -0.10472149  0.09071655  0.73243227  0.239007455
## Canopy         -0.403726835 -0.18904645 -0.06224725 -0.13874054  0.725355658
## m_toOpenWater   0.670502250  0.39526468 -0.24598298 -0.38699774  0.184991679
## Juncus          0.111371552 -0.16965765 -0.02898276  0.07449941  0.061571940
## Typha           0.214654501 -0.32479019 -0.41191448  0.03607367 -0.005329362
## Phrag          -0.279029750  0.07317320  0.07164325 -0.33845078 -0.140532455
## Grasses         0.094765336  0.19200510 -0.03602352  0.20190616  0.014352012
## Schoenoplectus -0.111800993  0.03210783 -0.07429791  0.13461038 -0.402259246
## Trees.shrubs   -0.111800993  0.03210783 -0.07429791  0.13461038 -0.402259246
## MixedEmergents -0.002506844  0.51431204  0.22379250  0.23107704  0.173735393
## Management0_1   0.074687681  0.12749281  0.71652295 -0.11274615  0.013145590
##                       PC11          PC12
## MaxDetections   0.062210302  0.000000e+00
## Avg_ht.m.      -0.002485018  2.042002e-16
## Canopy         -0.073904875 -6.495400e-16
## m_toOpenWater   0.011152030 -4.006627e-16
## Juncus         -0.560620237 -4.268452e-16
## Typha          -0.380138478 -3.226213e-16
## Phrag          -0.553349036 -4.694208e-16
## Grasses        -0.348917935 -1.363427e-16
## Schoenoplectus -0.073601705 -7.071068e-01
## Trees.shrubs   -0.073601705  7.071068e-01
## MixedEmergents -0.290062549 -6.651136e-17
## Management0_1  -0.093954813  1.502887e-17
```

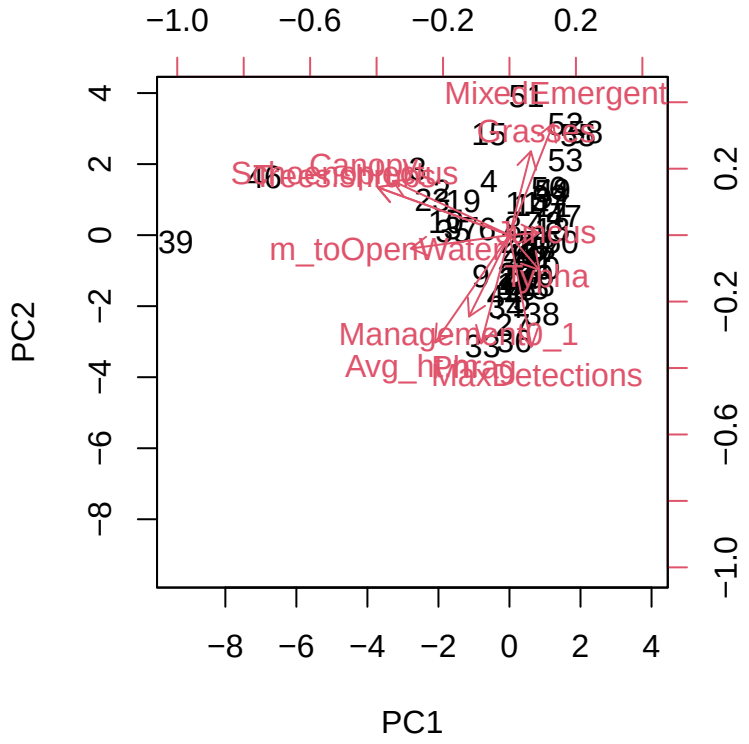
```
#reverse the signs of the scores
PC$x <- -1*PC$x
```

```r
# Make sure everything looks okay
head(PC$x)
```

```
##              PC1        PC2          PC3        PC4        PC5         PC6
## [1,] -0.09391944 -1.3822714  1.616800318  1.1217278 -0.5052485  0.08382228
## [2,] -1.89681532  1.2384841  1.858452868  1.3363069 -0.4623626 -0.31380795
## [3,] -2.58808243  1.8538633  0.594538588 -0.2263980 -0.1613421 -0.82900511
## [4,] -0.57029309  1.5323717  0.314980655  1.5509237  2.1313657  1.26642133
## [5,]  0.34257677 -1.8359648 -0.005822724  0.9684059 -0.5553836 -0.28686226
## [6,] -0.62331075  0.1789508  0.052296463  1.2151484  1.0625170 -0.82478848
##              PC7          PC8        PC9        PC10        PC11          PC12
## [1,] -1.01415683 -0.711776554  0.1181397  0.01298805 -0.11461157 -2.231577e-16
## [2,] -1.65628097 -0.007551378 -0.2863091  0.14644417  0.54477612  1.475822e-16
## [3,]  0.07352351  0.992886628  0.9751343  0.05543190  0.15965897  6.159743e-16
## [4,]  0.75284692  0.209152839 -0.1669889  0.21625214 -0.11371539 -1.825061e-16
## [5,] -0.07335222  0.236526162 -0.3353927 -0.23433814 -0.09278445 -1.895121e-17
## [6,]  0.02036508  0.398558577 -0.6881327 -0.20977230 -0.06846409 -1.730962e-16
```

3. Visualize results with a bi-plot

```r
biplot(PC, scale = 0) # scale = 0 ensures that the arrows in the plot are scaled to represent the loadi
```



4. Find the variance explained by each principal component

```
PC$sdev^2 / sum(PC$sdev^2)
```

```
##  [1] 2.919519e-01 2.007901e-01 1.343040e-01 1.235788e-01 7.425391e-02
##  [6] 6.317112e-02 4.987040e-02 2.840509e-02 2.157761e-02 8.855705e-03
## [11] 3.241417e-03 4.991809e-34
```

Try the PCA with fewer variables

1. Data management

```
scaled.data<-read.csv("data_scaled.csv", header = TRUE) # read in the scaled data

# remove columns of less predictive variables

scaled.data<-scaled.data[,-c(1,3,5,9,10,11,12)]
```

2. Calculate the Principal Components

```
# Calculate principal components
PC2 <- prcomp(scaled.data, scale = TRUE)

# Reverse the signs. Note:eigenvectors in R point in the negative direction by default, so multiply by
PC2$rotation <- -1*PC2$rotation

# Display principal components
PC2$rotation
```

```
##                       PC1         PC2         PC3         PC4         PC5
## MaxDetections -0.49894130 -0.27866937  0.44924992  0.25803259 -0.59914887
## Canopy         0.10930708  0.42311810 -0.39119051  0.67845969 -0.37370169
## Juncus         0.24329868 -0.71360603  0.05836168  0.28343073  0.08623574
## Typha         -0.56128084  0.02096763 -0.37622376 -0.44703957 -0.21241727
## Phrag         -0.06417655  0.48061135  0.69260162  0.03954317  0.18783603
## Management0_1 -0.60063340 -0.05151670 -0.14316804  0.43745928  0.64306112
##                       PC6
## MaxDetections  0.2145117
## Canopy        -0.2367119
## Juncus        -0.5834351
## Typha         -0.5459069
## Phrag         -0.4983483
## Management0_1  0.1057839
```

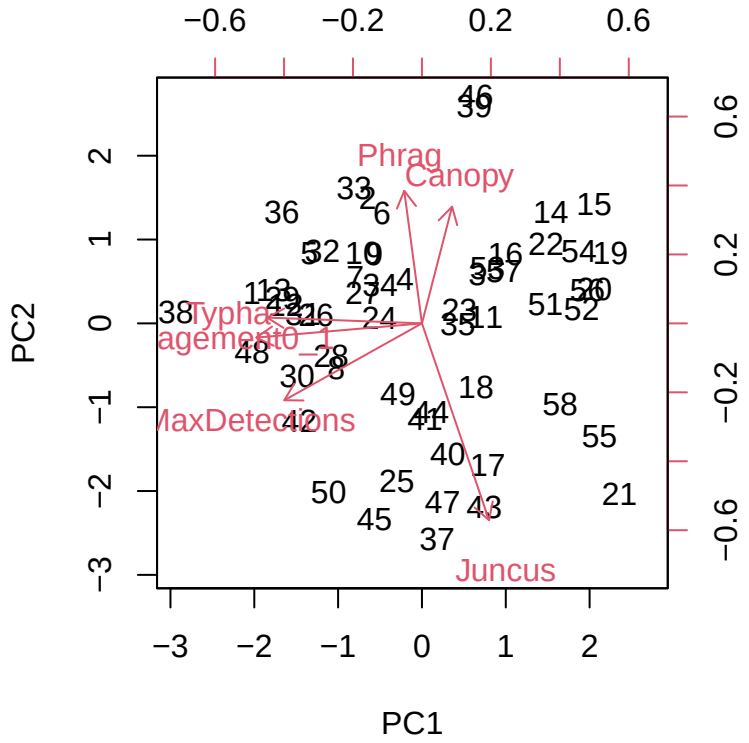
```
#reverse the signs of the scores
PC2$x <- -1*PC2$x

# Make sure everything looks okay
head(PC2$x)
```

```
##             PC1       PC2         PC3         PC4        PC5        PC6
## [1,] -2.0271086 0.3609094 -0.83500302 -0.69918420 -0.4538347 -0.7794026
```

```
## [2,] -0.6485624 1.5021642 -2.76793843  0.12229488 -0.1595342 -0.1985658
## [3,]  0.6627238 0.5856202 -1.36646991  1.56765633  0.1898897  0.5959097
## [4,] -0.2031127 0.5346931 -1.18532448 -0.04399826  0.4794701  0.8045966
## [5,] -1.3431096 0.8301397  0.91455898 -0.43496976  0.3473049 -0.2620987
## [6,] -0.4745510 1.3181142  0.04917724  0.49939364  0.2858240  0.2457213
```

3. Visualize results with a bi-plot

```
biplot(PC2, scale = 0)
```



4. Find the variance explained by each principal component

```
PC2$sdev^2 / sum(PC2$sdev^2)
```

```
## [1] 0.28552395 0.24062947 0.20500219 0.16939951 0.06285630 0.03658858
```