PCA for Environmental Traits

Jody Daniel

2020-11-03

Contents

Improving Explainability of Environmental Traits

As Julie described, the environmental traits are correlated. In past work, she found that they could be easily interpretable as PCs, each axis describing some element of the environment important for plants. As such, I will use the PCs in the ML model versus the raw traits. I will also make sure that the plan traits are not too correlated (>0.7).

Table of Contents

- Extracting Principle Components for Environmental Traits
- Safeguarding against multicollinearity in Plant Traits
- Conclusion

```r
library(corrplot)
library(RColorBrewer)
library(factoextra)
library(ggplot2)
require(ggrepel)
library(knitr)
library(kableExtra)
library(tidyverse)
library(dplyr)
library(here)
library(skimr)
library(reshape2)
library(tidymodels)
library(qdapTools)
library(rsample)
library(corrr)
library(broom)
library(vegan)
library(extrafont)
library(viridis)
library(car)
source(here("scripts/archive/1. functions.R"))
theme_set(theme_special())
```

Extracting Principle Components for Environmental Traits

```
rgr_msh_raw <- read_csv(here("data/RGR_MSH.csv"),
                        guess_max = 10000,
                        col_types = cols())

rgr_msh_na_raw <- read_csv(here("data/RGR_MSH_NA.csv"),
                        guess_max = 10000,
                        col_types = cols())
# now, let's remove columns that are either too correlated are would not be useful
labels_rgr_msh <- read_csv(here("data/labels.csv"),
                        guess_max = 10000,
                        col_types = cols())

# which traits are environmental versus plants?
environ_variables <- labels_rgr_msh$Feature[which(labels_rgr_msh$Class==2)]
environ_name <- which(colnames(rgr_msh_raw)%in%environ_variables)

plant_variables <- labels_rgr_msh$Feature[which(labels_rgr_msh$Class==1)]
plant_name <- which(colnames(rgr_msh_raw)%in%plant_variables)
```

Including Plots

You can also embed plots, for example:

```
rgr_na.pca <- prcomp(na.omit(rgr_msh_na_raw[,environ_name]),
                                center = TRUE, scale =TRUE)
# export eginvectors data
TW_G_Plot_NA<- data.frame(apply(data.frame(get_pca_ind(rgr_na.pca)$coord), 2, scale))
# save the column names as metric names
colnames(TW_G_Plot_NA) <- colnames(rgr_na.pca$x)
# export site coordinates
TW_G_Plot_VC_NA<- data.frame(get_pca_var(rgr_na.pca)$coord)
# save names
colnames(TW_G_Plot_VC_NA) <- colnames(rgr_na.pca$x)

# make the metric names a column
TW_G_Plot_VC_NA$Feature <- rownames(TW_G_Plot_VC_NA)
# order
TW_G_Plot_VC_NA$Order <- 1:nrow(TW_G_Plot_VC_NA)
# add metric labels for plotting
```

```
TW_G_Plot_VC_NA <- merge(TW_G_Plot_VC_NA,labels_rgr_msh,
                      by = "Feature")
# ensure we have the correct order
TW_G_Plot_VC_NA <- TW_G_Plot_VC_NA[order(TW_G_Plot_VC_NA$Order),]

rg_na.eigen <-  get_eigenvalue(rgr_na.pca)
```

```
rgr.pca <- prcomp(rgr_msh_raw[,environ_name],
                                center = TRUE, scale =TRUE)
# export eginvectors data
TW_G_Plot<- data.frame(apply(data.frame(get_pca_ind(rgr.pca)$coord), 2, scale))
# save the column names as metric names
colnames(TW_G_Plot) <- colnames(rgr.pca$x)
# export site coordinates
TW_G_Plot_VC<- data.frame(get_pca_var(rgr.pca)$coord)
# save names
colnames(TW_G_Plot_VC) <- colnames(rgr.pca$x)

# fliping axes to make sure the match the base PCA
TW_G_Plot_VC$PC6 <- TW_G_Plot_VC$PC6*-1
TW_G_Plot$PC6 <- TW_G_Plot$PC6*-1

TW_G_Plot_VC$PC5 <- TW_G_Plot_VC$PC5*-1
TW_G_Plot$PC5 <- TW_G_Plot$PC5*-1

TW_G_Plot_VC$PC4 <- TW_G_Plot_VC$PC4*-1
TW_G_Plot$PC4 <- TW_G_Plot$PC4*-1

# make the mertic names a column
TW_G_Plot_VC$Feature <- rownames(TW_G_Plot_VC)
# order
TW_G_Plot_VC$Order <- 1:nrow(TW_G_Plot_VC)
# add metric labels for plotting
TW_G_Plot_VC <- merge(TW_G_Plot_VC,labels_rgr_msh,
                      by = "Feature")
# ensure we have the correct order
TW_G_Plot_VC <- TW_G_Plot_VC[order(TW_G_Plot_VC$Order),]

rg.eigen <-  get_eigenvalue(rgr.pca)
```

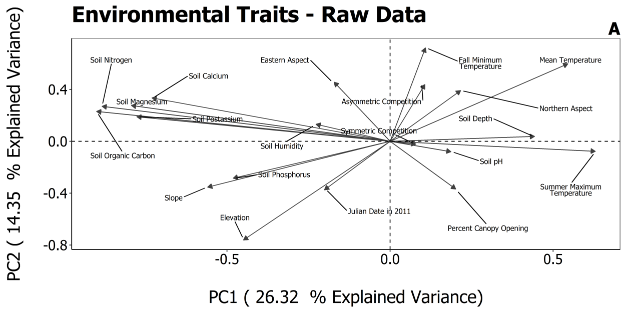
Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

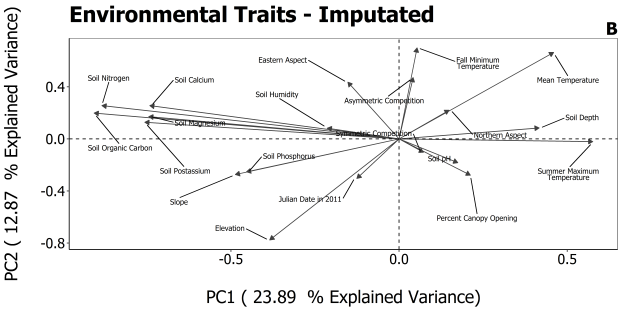
```
png(here("notebooks/figures/PCA_A.png"), width = 10 , height = 5, units = ’in’, res = 600)
ggplot(data = TW_G_Plot_NA, aes(x = PC1, y = PC2))+
geom_segment(data=TW_G_Plot_VC_NA,aes(x=0,xend = PC1, y=0, yend = PC2),
               arrow = arrow(length = unit(0.2, "cm"),
                             type="closed"),size = 0.5,color = "grey25",inherit.aes=TRUE)+
geom_text_repel(data=TW_G_Plot_VC_NA,
                  aes(x = PC1, y = PC2,label= stringr::str_wrap(Label,23)),
                lineheight = 0.7, size = 3,
                  box.padding = unit(1.5, "lines"),
                  point.padding = unit(0.5, "lines"), family = "Tahoma")+
geom_vline(xintercept = 0, linetype = "dashed")+
geom_hline(yintercept = 0, linetype = "dashed")+
labs(title = "Environmental Traits - Raw Data  ",
       subtitle  = LETTERS[1])+ #title
xlab(paste("PC1 (", round(rg_na.eigen$variance.percent[1], 2), " % Explained Variance)")) +
ylab(paste("PC2 (", round(rg_na.eigen$variance.percent[2], 2), " % Explained Variance)")) +
theme_special()
dev.off()
```

```
## pdf
##   2
```

```
png(here("notebooks/figures/PCA_B.png"), width = 10 , height = 5, units = 'in', res = 600)
ggplot(data = TW_G_Plot, aes(x = PC1, y = PC2))+
geom_segment(data=TW_G_Plot_VC,aes(x=0,xend = PC1, y=0, yend = PC2),
               arrow = arrow(length = unit(0.2, "cm"),
                             type="closed"),size = 0.5,color = "grey25",inherit.aes=TRUE)+
geom_text_repel(data=TW_G_Plot_VC,
                  aes(x = PC1, y = PC2,label= stringr::str_wrap(Label,23)), lineheight = 0.7, size = 3,
                  box.padding = unit(1.5, "lines"),
                  point.padding = unit(0.5, "lines"), family = "Tahoma")+
geom_vline(xintercept = 0, linetype = "dashed")+
geom_hline(yintercept = 0, linetype = "dashed")+
labs(title = "Environmental Traits - Imputated  ",
       subtitle  = LETTERS[2])+ #title
xlab(paste("PC1 (", round(rg.eigen$variance.percent[1], 2), " % Explained Variance)")) +
ylab(paste("PC2 (", round(rg.eigen$variance.percent[2], 2), " % Explained Variance)")) +
theme_special()
dev.off()
```

```
## pdf
##   2
```



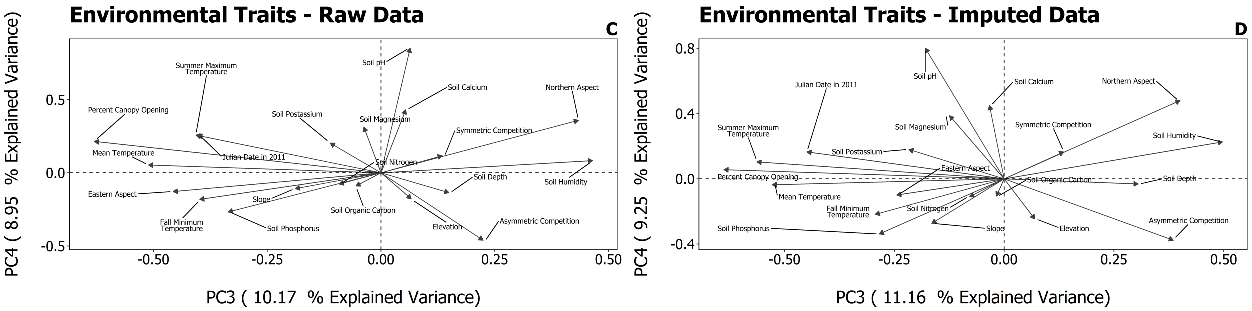


```
png(here("notebooks/figures/PCA_C.png"), width = 10 , height = 5, units = 'in', res = 600)
ggplot(data = TW_G_Plot_NA, aes(x = PC3, y = PC4))+
geom_segment(data=TW_G_Plot_VC_NA,aes(x=0,xend = PC3, y=0, yend = PC4),
               arrow = arrow(length = unit(0.2, "cm"),
                             type="closed"),size = 0.5,color = "grey25",inherit.aes=TRUE)+
geom_text_repel(data=TW_G_Plot_VC_NA,
                  aes(x = PC3, y = PC4,label= stringr::str_wrap(Label,23)), lineheight = 0.7, size = 3,
                  box.padding = unit(1.5, "lines"),
                   point.padding = unit(0.5, "lines"), family = "Tahoma")+
geom_vline(xintercept = 0, linetype = "dashed")+
geom_hline(yintercept = 0, linetype = "dashed")+
labs(title = "Environmental Traits - Raw Data ",
       subtitle  = LETTERS[3])+ #title
xlab(paste("PC3 (", round(rg_na.eigen$variance.percent[3], 2), " % Explained Variance)")) +
ylab(paste("PC4 (", round(rg_na.eigen$variance.percent[4], 2), " % Explained Variance)")) +
theme_special()
dev.off()
```

```
## pdf
##   2
```

```
png(here("notebooks/figures/PCA_D.png"), width = 10 , height = 5, units = 'in', res = 600)
ggplot(data = TW_G_Plot, aes(x = PC3, y = PC4))+
geom_segment(data=TW_G_Plot_VC,aes(x=0,xend = PC3, y=0, yend = PC4),
               arrow = arrow(length = unit(0.2, "cm"),
                             type="closed"),size = 0.5,color = "grey25",inherit.aes=TRUE)+
geom_text_repel(data=TW_G_Plot_VC,
                   aes(x = PC3, y = PC4,label= stringr::str_wrap(Label,23)), lineheight = 0.7, size = 3,
                   box.padding = unit(1.5, "lines"),
                   point.padding = unit(0.5, "lines"), family = "Tahoma")+
geom_vline(xintercept = 0, linetype = "dashed")+
geom_hline(yintercept = 0, linetype = "dashed")+
labs(title = "Environmental Traits - Imputed Data ",
       subtitle  = LETTERS[4])+ #title
xlab(paste("PC3 (", round(rg.eigen$variance.percent[3], 2), " % Explained Variance)")) +
ylab(paste("PC4 (", round(rg.eigen$variance.percent[4], 2), " % Explained Variance)")) +
theme_special()
dev.off()
```

```
## pdf
##   2
```

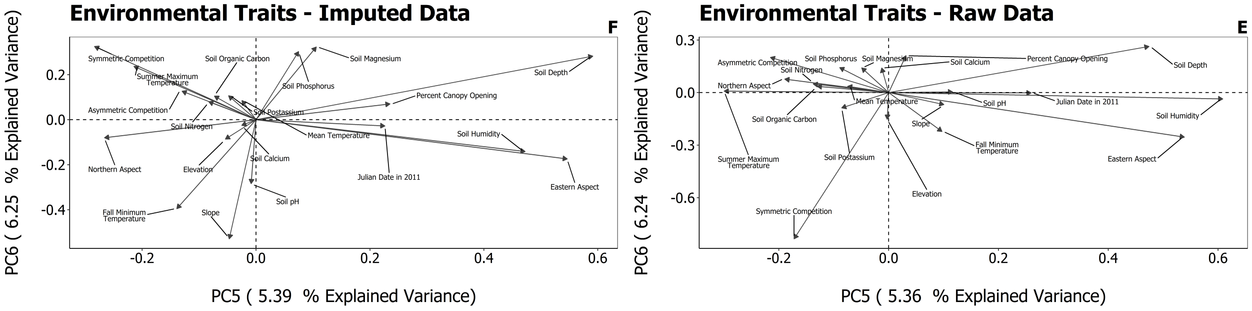


```
png(here("notebooks/figures/PCA_E.png"), width = 10 , height = 5, units = 'in', res = 600)
# plot PCA #
ggplot(data = TW_G_Plot, aes(x = PC5, y = PC6))+
geom_segment(data=TW_G_Plot_VC,aes(x=0,xend = PC5, y=0, yend = PC6),
               arrow = arrow(length = unit(0.2, "cm"),
                             type="closed"),size = 0.5,color = "grey25",inherit.aes=TRUE)+
geom_text_repel(data=TW_G_Plot_VC,
                   aes(x = PC5, y = PC6,label= stringr::str_wrap(Label,23)), lineheight = 0.7, size = 3,
                   box.padding = unit(1.5, "lines"),
                   point.padding = unit(0.5, "lines"), family = "Tahoma")+
geom_vline(xintercept = 0, linetype = "dashed")+
geom_hline(yintercept = 0, linetype = "dashed")+
labs(title = "Environmental Traits - Imputed Data ",
       subtitle  = LETTERS[6])+ #title
xlab(paste("PC5 (", round(rg.eigen$variance.percent[6], 2), " % Explained Variance)")) +
ylab(paste("PC6 (", round(rg.eigen$variance.percent[5], 2), " % Explained Variance)")) +
theme_special()
dev.off()
```

```
## pdf
##   2
```

```
png(here("notebooks/figures/PCA_F.png"), width = 10 , height = 5, units = 'in', res = 600)
ggplot(data = TW_G_Plot_NA, aes(x = PC5, y = PC6))+
geom_segment(data=TW_G_Plot_VC_NA,aes(x=0,xend = PC5, y=0, yend = PC6),
                arrow = arrow(length = unit(0.2, "cm"),
                              type="closed"),size = 0.5,color = "grey25",inherit.aes=TRUE)+
geom_text_repel(data=TW_G_Plot_VC_NA,
                   aes(x = PC5, y = PC6,label= stringr::str_wrap(Label,23)), lineheight = 0.7, size = 3,
                   box.padding = unit(1.5, "lines"),
                   point.padding = unit(0.5, "lines"), family = "Tahoma")+
geom_vline(xintercept = 0, linetype = "dashed")+
geom_hline(yintercept = 0, linetype = "dashed")+
labs(title = "Environmental Traits - Raw Data ",
       subtitle  = LETTERS[5])+ #title
xlab(paste("PC5 (", round(rg_na.eigen$variance.percent[6], 2), " % Explained Variance)")) +
ylab(paste("PC6 (", round(rg_na.eigen$variance.percent[5], 2), " % Explained Variance)")) +
theme_special()
dev.off()
```

```
## pdf
##   2
```



```
# now for correlation assessments, but need to rename PCs based on what they represent
TW_G_Plot_NA <- TW_G_Plot_NA[,1:6]
pca_env <- c( "Soil.Fertility", "Light", "Temperature",   "pH", "Soil.Humidity.Depth ", "Slope")
colnames(TW_G_Plot_NA) <- pca_env
```

```
TW_G_Plot <- TW_G_Plot[,1:6]
colnames(TW_G_Plot) <- pca_env

# which rows are ommitted from the PCA because there are NAs?
rgr_msh_na_raw_pca <- na.omit(rgr_msh_na_raw[,c(1, environ_name)])
which.g <- which(rgr_msh_na_raw$SampleID%in%rgr_msh_na_raw_pca$SampleID)
RGR_MSH_PCA <- data.frame(rgr_msh_raw[,-environ_name], TW_G_Plot)
RGR_MSH_NA_PCA <- data.frame(rgr_msh_na_raw[which.g,-environ_name], TW_G_Plot_NA)
```

Correlation on Plant Traits

I want to ensure that the plant traits are not correlated. Julie said that past work suggests that they are not easily represented using a PCA. So, I will not use the this feature reduction method.

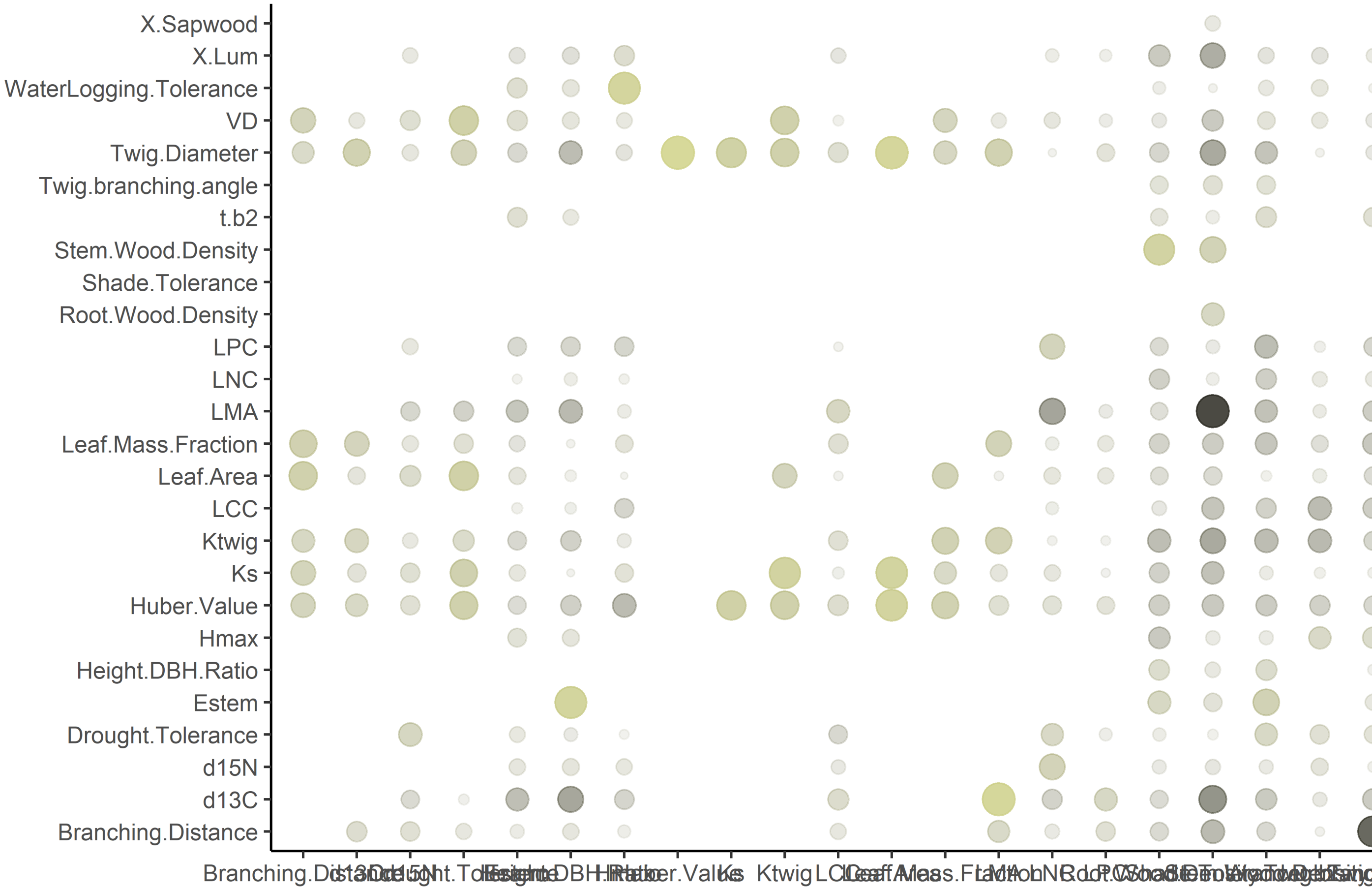
```
plant_name <- which(colnames(RGR_MSH_PCA)%in%plant_variables)
plant_name_na <- which(colnames(RGR_MSH_NA_PCA)%in%plant_variables)



png(here("notebooks/figures/CORR_A.png"), width = 10 , height = 5, units = 'in', res = 600)
RGR_MSH_PCA[,plant_name] %>%
  correlate() %>%
  # Re-arrange a correlation data frame
  # to group highly correlated variables closer together.
  rearrange(method = "MDS", absolute = FALSE) %>%
  shave() %>%
  rplot(shape = 19, colors = inferno(2))
dev.off()
```

```
## pdf
##   2
```

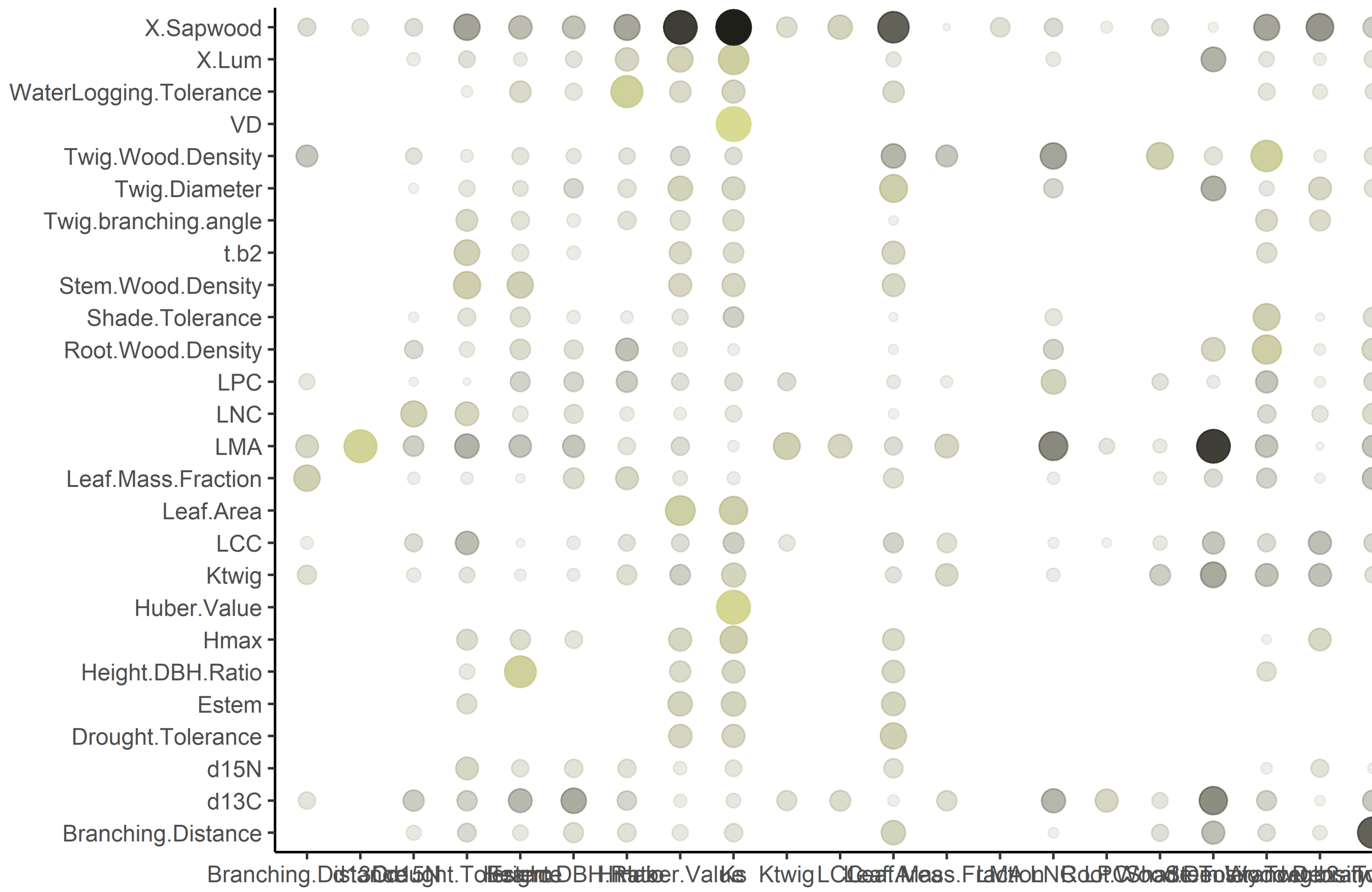
```
png(here("notebooks/figures/CORR_B.png"), width = 10 , height = 5, units = 'in', res = 600)
RGR_MSH_NA_PCA[,plant_name_na] %>%
  correlate() %>%
  # Re-arrange a correlation data frame
  # to group highly correlated variables closer together.
  rearrange(method = "MDS", absolute = FALSE) %>%
  shave() %>%
  rplot(shape = 19, colors = inferno(2))
dev.off()
```

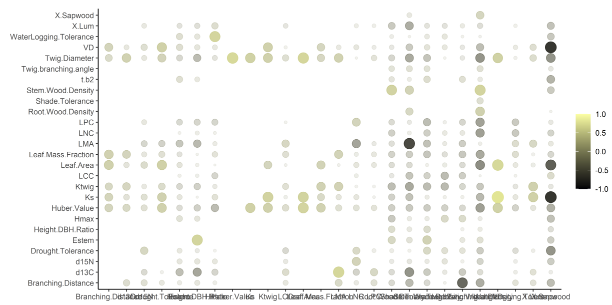
```
## pdf
##   2
```

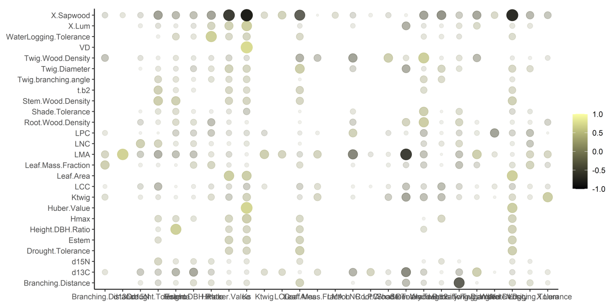
```
include_graphics(here("notebooks/figures/CORR_A.png"))
```



```r
include_graphics(here("notebooks/figures/CORR_B.png"))
```







I'll keep each of the plant traits. I should not have included the porosity traits anyway.

```
# now pulling what we need for final model building
environ_name <- which(colnames(RGR_MSH_PCA)%in%pca_env)
environ_name_na <- which(colnames(RGR_MSH_NA_PCA)%in%pca_env)

predictors_name <- which(colnames(rgr_msh_raw)%in% c("BAI_GR", "BIO_GR"))
predictors_name_na <- which(colnames(rgr_msh_na_raw)%in% c("BAI_GR", "BIO_GR"))


RGR_MSH_PCA_FINAL_NA <- data.frame(SampleID = RGR_MSH_NA_PCA$SampleID,
                                   rgr_msh_na_raw[which.g, predictors_name_na],
                                   RGR_MSH_NA_PCA[,plant_name_na],
                                    RGR_MSH_NA_PCA[,environ_name_na])

RGR_MSH_PCA_FINAL <- data.frame(SampleID = RGR_MSH_PCA$SampleID,
                                rgr_msh_raw[, predictors_name],
                                RGR_MSH_PCA[,plant_name],
                                RGR_MSH_PCA[,environ_name])
write_csv(RGR_MSH_PCA_FINAL,here("data/RGR_MSH_PCA.csv"))
write_csv(RGR_MSH_PCA_FINAL_NA,here("data/RGR_MSH_PCA_NA.csv"))
```