

# Vitamin D Microbiome Analysis

Aneesa Valentine

5/8/21

## Contents

R Markdown . . . . .	2
Install latest version of rlang and LaTeX for PDF output knit . . . . .	2
Clear global environment . . . . .	2
Setting up R to acknowledge microbiome dataset . . . . .	2
Read this dataset excel file . . . . .	2
Organize dataset . . . . .	2
Omit NA from tables . . . . .	3
Aggregate tables i.e sort mouse samples by bacteria category . . . . .	3
Remove 0's from pre-melted data . . . . .	3
Log transform data from aggregated tables for ggplot function . . . . .	4
Proportion table using pre-melted data . . . . .	4
Melt proportion table . . . . .	6
Subset melted prop table to show values >0.5% . . . . .	7
Proportion plot . . . . .	7
T-test using prop tables . . . . .	7
Species violin plot . . . . .	9
Principal component analysis for species . . . . .	11
Subset mdata to look for enrichment overall, rather than within individual taxa . . . . .	15
Overall taxonomic abundance histogram . . . . .	15

```
library(ggplot2)
library(ggpubr)
library(ggfortify)
#install.packages("pdfcrop")
#install.packages("ghostscript")
```

## R Markdown

Install latest version of rlang and LaTeX for PDF output knit

Clear global environment

```
rm(list=ls())
```

Setting up R to acknowledge microbiome dataset

```
#Identify path
```

```
getwd()
```

```
## [1] "C:/Users/neesy/OneDrive/Desktop"
```

```
# Set path
```

```
setwd("C:/Users/neesy/OneDrive/Desktop")
```

Read this dataset excel file

```
library(readxl)
```

```
# Read and View dataset
```

```
One_Codex_Results_2021_04_05_18_27_ <- read_excel("C:/Users/neesy/OneDrive/Desktop/One Codex Results (2021-04-05 18:27:00).xlsx",  
  sheet = "All Taxa")
```

```
View(One_Codex_Results_2021_04_05_18_27_)
```

```
# Identify the dataset as "mdata," meaning "microbiome data"
```

```
mdata = One_Codex_Results_2021_04_05_18_27_
```

```
#tinytex::install_tinytex()
```

Organize dataset

```
# Pull Phylum (and other independent categories i.e Class etc.) columns and mouse samples only from "mdata"
```

```
#install.packages("GLDEX")
```

```
Phylum = cbind(mdata[,6],mdata[,13:24])
```

```
Class = cbind(mdata[,7],mdata[,13:24])
```

```
Order = cbind(mdata[,8],mdata[,13:24])
```

```
Family = cbind(mdata[,9],mdata[,13:24])
```

```
Genus = cbind(mdata[,10],mdata[,13:24])
```

```
Species = cbind(mdata[,11],mdata[,13:24])
```

## Omit NA from tables

```
# NA removed from the tables
Phylum2 <- na.omit(Phylum)
Class2 <- na.omit(Class)
Order2 <- na.omit(Order)
Family2 <-na.omit(Family)
Genus2 <-na.omit(Genus)
Species2 <-na.omit(Species)
```

```
paste("Original Phylum file dimension")
```

```
## [1] "Original Phylum file dimension"
```

```
dim(Phylum)
```

```
## [1] 3405 13
```

```
paste ("na removed file dimentstion")
```

```
## [1] "na removed file dimentstion"
```

```
dim(Phylum2)
```

```
## [1] 3315 13
```

## Aggregate tables i.e sort mouse samples by bacteria category

```
# Get list of Phylum (and other categories) only

phy <-list(Phylum2$Phylum)
ord <-list(Order2$Order)
clas <-list(Class2$Class)
fam <-list(Family2$Family)
gen <-list(Genus2$Genus)

# Aggregate by list
dtphylum <-aggregate(Phylum2[,2:13],by = phy, FUN="sum")
dtorder <-aggregate(Order2[,2:13],by = ord, FUN="sum")
dtclass <-aggregate(Class2[,2:13],by = clas, FUN="sum")
dtfamily <-aggregate(Family2[,2:13],by = fam, FUN="sum")
dtgenus <-aggregate(Genus2[,2:13],by = gen, FUN="sum")
```

## Remove 0's from pre-melted data

```

phylum0 <-dtphylum[which(rowSums(dtphylum[,2:dim(dtphylum)[2]])>1),]
order0 <-dtorder[which(rowSums(dtorder[,2:dim(dtorder)[2]])>1),]
class0 <-dtclass[which(rowSums(dtclass[,2:dim(dtclass)[2]])>1),]
family0 <-dtfamily[which(rowSums(dtfamily[,2:dim(dtfamily)[2]])>1),]
genus0 <-dtgenus[which(rowSums(dtgenus[,2:dim(dtgenus)[2]])>1),]
species0 <-Species2[which(rowSums(Species2[,2:dim(Species2)[2]])>1),]

```

## Log transform data from aggregated tables for ggplot function

### Proportion table using pre-melted data

```

# X = list of the read, Y: colSums(sample 1), x[sample 1]/Y

```

```

# Phylum prop table

```

```

col_sum <-colSums(phylum0[,2:dim(phylum0)[2]]) # Calculate sum of each column, starting from col 2
head(col_sum)

```

```

## 2bd332ec71534b09 c52273f17b8148ac 5d247f45be1c4258 ab61a1c60b33433a
##          9704118          6710434          7859927          6990508
## c76d8968674f44a5 c34442318bf4443e
##          9905087          11146673

```

```

phylumprop <-matrix(nrow=dim(phylum0)[1], ncol=dim(phylum0)[2]) # Make an empty table with the same dim
rownames(phylumprop) <-phylum0[,1] # Change row names of new table to names in col 1 of phylum0
for (i in 1:dim(phylum0)[1]){

```

```

  for (c in 2:dim(phylum0)[2]){
    phylumprop[i,c] <-(phylum0[i,c])/(col_sum[c-1])} # C-1 tells R to divide by colsums-1st column
  }

```

```

# Remove col 1 from prop table

```

```

phylumprop <-phylumprop[, -1]

```

```

# Change col names

```

```

colnames(phylumprop) <-(c("VDS1", "VDD1", "VDD2", "VDD3", "VDS2", "VDD4", "VDS3", "VDD5", "VDD6", "VDS4", "VDD7",

```

```

# Class prop table

```

```

col_sum <-colSums(class0[,2:dim(class0)[2]])
head(col_sum)

```

```

## 2bd332ec71534b09 c52273f17b8148ac 5d247f45be1c4258 ab61a1c60b33433a
##          9605433          6549373          7734551          6867419
## c76d8968674f44a5 c34442318bf4443e
##          9703594          11038873

```

```

classprop <-matrix(nrow=dim(class0)[1], ncol=dim(class0)[2])
rownames(classprop) <-class0[,1]
for (i in 1:dim(class0)[1]){

```



```
## 2bd332ec71534b09 c52273f17b8148ac 5d247f45be1c4258 ab61a1c60b33433a
##          8938018          6065813          7068254          6261363
## c76d8968674f44a5 c34442318bf4443e
##          8799021          10449811
```

```
familyprop <-matrix(nrow=dim(family0)[1], ncol=dim(family0)[2])
rownames(familyprop) <-family0[,1]
for (i in 1:dim(family0)[1]){

  for (c in 2:dim(family0)[2]){
    familyprop[i,c] <-(family0[i,c])/(col_sum[c-1])}
}

familyprop <-familyprop[, -1]
colnames(familyprop) <-(c("VDS1", "VDD1", "VDD2", "VDD3", "VDS2", "VDD4", "VDS3", "VDD5", "VDD6", "VDS4", "VDD7",

# Species prop table
col_sum <-colSums(species0[,2:dim(species0)[2]])
head(col_sum)
```

```
## 2bd332ec71534b09 c52273f17b8148ac 5d247f45be1c4258 ab61a1c60b33433a
##          3118984          3008476          3235931          2861068
## c76d8968674f44a5 c34442318bf4443e
##          4394002          4487706
```

```
speciesprop <-matrix(nrow=dim(species0)[1], ncol=dim(species0)[2])
rownames(speciesprop) <-species0[,1]
for (i in 1:dim(species0)[1]){

  for (c in 2:dim(species0)[2]){
    speciesprop[i,c] <-(species0[i,c])/(col_sum[c-1])}
}

speciesprop <-speciesprop[, -1]
colnames(speciesprop) <-(c("VDS1", "VDD1", "VDD2", "VDD3", "VDS2", "VDD4", "VDS3", "VDD5", "VDD6", "VDS4", "VDD7"
```

## Melt proportion table

```
library(reshape2)

phylummelt <-melt(phylumprop)
classmelt <-melt(classprop)
ordermelt <-melt(orderprop)
familymelt <-melt(familyprop)
genusmelt <-melt(genusprop)
speciesmelt <-melt(speciesprop)
speciesmelt[,4] <-substr(speciesmelt[,2],1,3)
```

## Subset melted prop table to show values >0.5%

```
#install.packages("dplyr")

subsetphylum <-subset(phylummelt, phylummelt[,3]>0.01)
subsetclass <-subset(classmelt, classmelt[,3]>0.01)
subsetorder <-subset(ordermelt, ordermelt[,3]>0.01)
subsetgenus <-subset(genusmelt, genusmelt[,3]>0.01)
subsetfamily <-subset(familymelt, familymelt[,3]>0.01)
subsetspecies <-subset(speciesmelt, speciesmelt[,3]>0.05) #Most specific classification so increased su
```

## Proportion plot

```
library(ggplot2)
pdf(file="/Users/neesy/Documents/Phylumprop3.pdf",width = 24,height = 36)
ggplot(subsetspecies, aes(fill=Var1, y=value,x=Var2))+
  geom_bar(position="fill", stat="identity") + theme(legend.position = "bottom")+scale_x_discrete(limit=
dev.off()
```

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

## T-test using prop tables

```
#install.packages("ggpubr")

# Sample columns 1,5,7 and 11 = VDS. Columns 2,3,4,6,8,9,11 and 12 = VDD.

# Phylum t test (independent groups)

phylumtt <-matrix(nrow=dim(phylumprop)[1], ncol=5)
colnames(phylumtt) <-(c("P", "Mean VDS", "Mean VDD", "Difference", "P adj"))
rownames(phylumtt) <-phylum0[,1]
for (i in 1:dim(phylumprop)[1]){
  phylumtt[i,1] <-wilcox.test(phylumprop[i,c(2,6,8,11)], phylumprop[i,c(3,4,5,7,9,10,12)])$p.value ##
  phylumtt[i,2] <-mean(phylumprop[i,c(2,6,8,11)]) #mean of VDS
  phylumtt[i,3] <-mean(phylumprop[i,c(3,4,5,7,9,10,12)]) # mean of VDD
  phylumtt[i,4] <- mean(phylumprop[i,c(3,4,5,7,9,10,12)]) - mean(phylumprop[i,c(2,6,8,11)]) # Diff (V
}

# Adjust p value using Bonferroni
phylumtt[,5] <-p.adjust(phylumtt[,1], method="BH")

# Class t test

classtt <-matrix(nrow=dim(classprop)[1], ncol=5)
```

```

colnames(classtt) <- (c("P", "Mean VDS", "Mean VDD", "Difference", "P adj"))
rownames(classtt) <- class0[,1]
for (i in 1:dim(classprop)[1]){
  classtt[i,1] <- wilcox.test(classprop[i,c(2,6,8,11)], classprop[i,c(3,4,5,7,9,10,12)])$p.value ## p-value
  classtt[i,2] <- mean(classprop[i,c(2,6,8,11)]) #mean of VDS
  classtt[i,3] <- mean(classprop[i,c(3,4,5,7,9,10,12)]) # mean of VDD
  classtt[i,4] <- mean(classprop[i,c(3,4,5,7,9,10,12)]) - mean(classprop[i,c(2,6,8,11)]) # Diff (VDD-VDS)
}

classtt[,5] <- p.adjust(classtt[,1], method="BH")

# Order t test

ordertt <- matrix(nrow=dim(orderprop)[1], ncol=5)
colnames(ordertt) <- (c("P", "Mean VDS", "Mean VDD", "Difference", "P adj"))
rownames(ordertt) <- order0[,1]
for (i in 1:dim(orderprop)[1]){
  ordertt[i,1] <- wilcox.test(orderprop[i,c(2,6,8,11)], orderprop[i,c(3,4,5,7,9,10,12)])$p.value ## p-value
  ordertt[i,2] <- mean(orderprop[i,c(2,6,8,11)]) #mean of VDS
  ordertt[i,3] <- mean(orderprop[i,c(3,4,5,7,9,10,12)]) # mean of VDD
  ordertt[i,4] <- mean(orderprop[i,c(3,4,5,7,9,10,12)]) - mean(orderprop[i,c(2,6,8,11)]) # Diff (VDD-VDS)
}

ordertt[,5] <- p.adjust(ordertt[,1], method="BH")

# Family t test

familytt <- matrix(nrow=dim(familyprop)[1], ncol=5)
colnames(familytt) <- (c("P", "Mean VDS", "Mean VDD", "Difference", "P adj"))
rownames(familytt) <- family0[,1]
for (i in 1:dim(familyprop)[1]){
  familytt[i,1] <- wilcox.test(familyprop[i,c(2,6,8,11)], familyprop[i,c(3,4,5,7,9,10,12)])$p.value ## p-value
  familytt[i,2] <- mean(familyprop[i,c(2,6,8,11)]) #mean of VDS
  familytt[i,3] <- mean(familyprop[i,c(3,4,5,7,9,10,12)]) # mean of VDD
  familytt[i,4] <- mean(familyprop[i,c(3,4,5,7,9,10,12)]) - mean(familyprop[i,c(2,6,8,11)]) # Diff (VDD-VDS)
}

familytt[,5] <- p.adjust(familytt[,1], method="BH")

# Genus t test

genustt <- matrix(nrow=dim(genusprop)[1], ncol=5)
colnames(genustt) <- (c("P", "Mean VDS", "Mean VDD", "Difference", "P adj"))
rownames(genustt) <- genus0[,1]
for (i in 1:dim(genusprop)[1]){
  genustt[i,1] <- wilcox.test(genusprop[i,c(2,6,8,11)], genusprop[i,c(3,4,5,7,9,10,12)])$p.value ## p-value
  genustt[i,2] <- mean(genusprop[i,c(2,6,8,11)]) #mean of VDS
  genustt[i,3] <- mean(genusprop[i,c(3,4,5,7,9,10,12)]) # mean of VDD
  genustt[i,4] <- mean(genusprop[i,c(3,4,5,7,9,10,12)]) - mean(genusprop[i,c(2,6,8,11)]) # Diff (VDD-VDS)
}

```



```

genustt[,5] <-p.adjust(genustt[,1], method="BH")

# Species t test

speciestt <-matrix(nrow=dim(speciesprop)[1], ncol=5)
colnames(speciestt) <-(c("P", "Mean VDS", "Mean VDD", "Difference", "P adj"))
rownames(speciestt) <-species0[,1]
for (i in 1:dim(speciesprop)[1]){
  speciestt[i,1] <-wilcox.test(speciesprop[i,c(2,6,8,11)], speciesprop[i,c(3,4,5,7,9,10,12)], alternative="two.sided")
  speciestt[i,2] <-mean(speciesprop[i,c(2,6,8,11)]) #mean of VDS
  speciestt[i,3] <-mean(speciesprop[i,c(3,4,5,7,9,10,12)]) # mean of VDD
  speciestt[i,4] <- mean(speciesprop[i,c(3,4,5,7,9,10,12)]) - mean(speciesprop[i,c(2,6,8,11)]) # Difference
}

speciestt[,5] <-p.adjust(speciestt[,1], method="BH")

```

## Species violin plot

```

#install.packages("ggpubr")
library(ggpubr)
#install.packages("ggplot2")

# Look in species prop for the row names that match speciestt that p equals <0.05
sigspecies <-speciesprop[which(rownames(speciesprop)%in%rownames(speciestt[which(speciestt[,1]<0.05),]))]

siggenus <-genusprop[which(rownames(genusprop)%in%rownames(genustt[which(genustt[,1]<0.05),])),]

# Melt sig species + add column VDS/VDD variable
sigspeciesmelt <-melt(sigspecies)
sigspeciesmelt[,4] <-substr(sigspeciesmelt[,2],1,3)

siggenusmelt <-melt(siggenus)
siggenusmelt[,4] <-substr(siggenusmelt[,2],1,3)

# Plot 7 sig species
my_comparisons2 <- list(c("VDS", "VDD"))
ggviolin(data = sigspeciesmelt, x = "V4", y = "value", fill="V4", palette = c("lightblue", "orchid2"),

         add = "boxplot", add.params = list(fill = "blue"))+

  facet_wrap(~Var1, scales="free") + stat_compare_means(comparisons = my_comparisons2, method="wilcox.test") +
  labs(title = "Genus taxa", y="Proportion (%)", x="") + theme_bw()

## Warning in wilcox.test.default(c(0, 0, 0, 7.56094178145712e-06), c(0, 0, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(0, 0, 4.80070019475683e-06,

```

```
## 7.08838292011605e-06: cannot compute exact p-value with ties

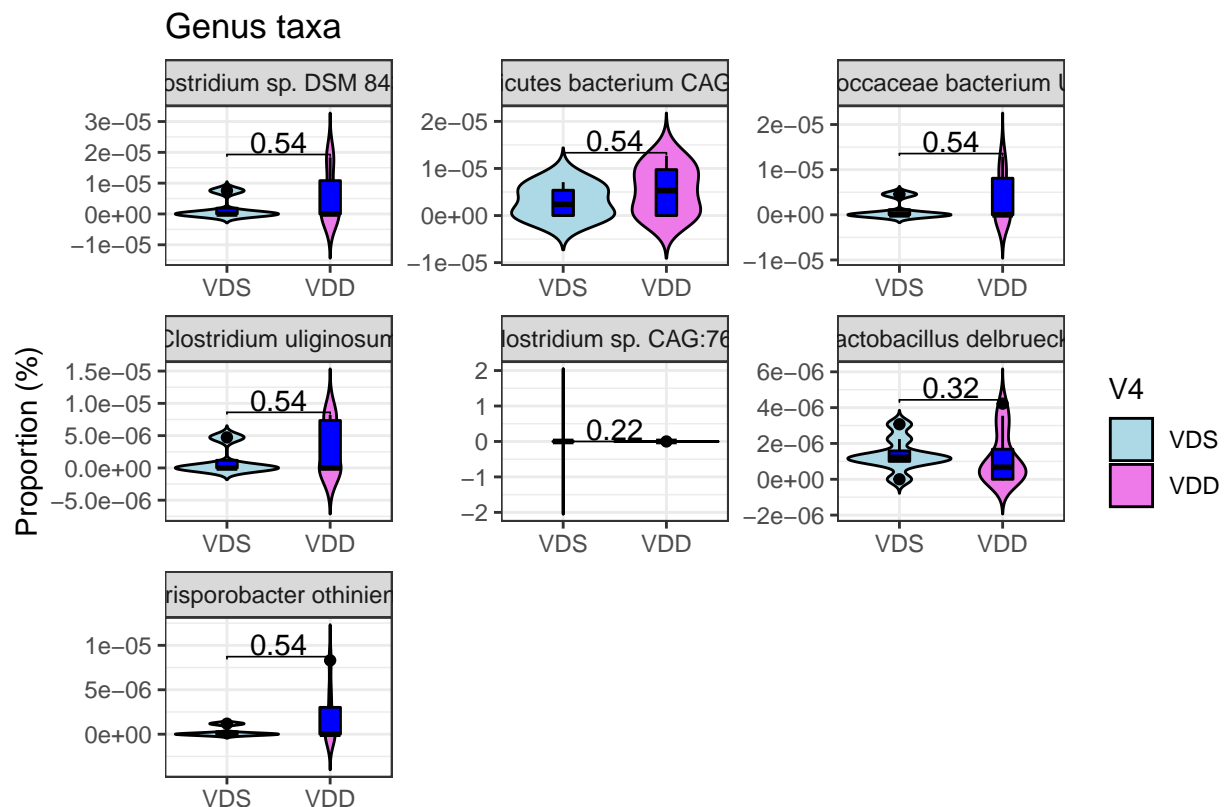
## Warning in wilcox.test.default(c(0, 4.55165928463392e-06, 0, 0), c(0, 0, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(0, 0, 0, 4.7255886134107e-06), c(0, 0, : cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(c(0, 0, 0, 0), c(1.79492872803373e-05, 0, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(2.24432058644738e-06, 9.61851679906021e-07, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(0, 0, 0, 1.18139715335268e-06),
## c(8.30985522237837e-06, : cannot compute exact p-value with ties
```



```
# Plot 7 sig genus
my_comparisons2 <- list(c("VDS","VDD"))
ggviolin(data = siggenusmelt, x = "V4", y = "value", fill="V4", palette = c("lightblue","orchid2"),

          add = "boxplot", add.params = list(fill = "blue"))+

facet_wrap(~Var1,scales="free") + stat_compare_means(comparisons = my_comparisons2,method="wilcox.test")
labs(title = "Genus taxa",y="Proportion (%)",x="") + theme_bw()
```

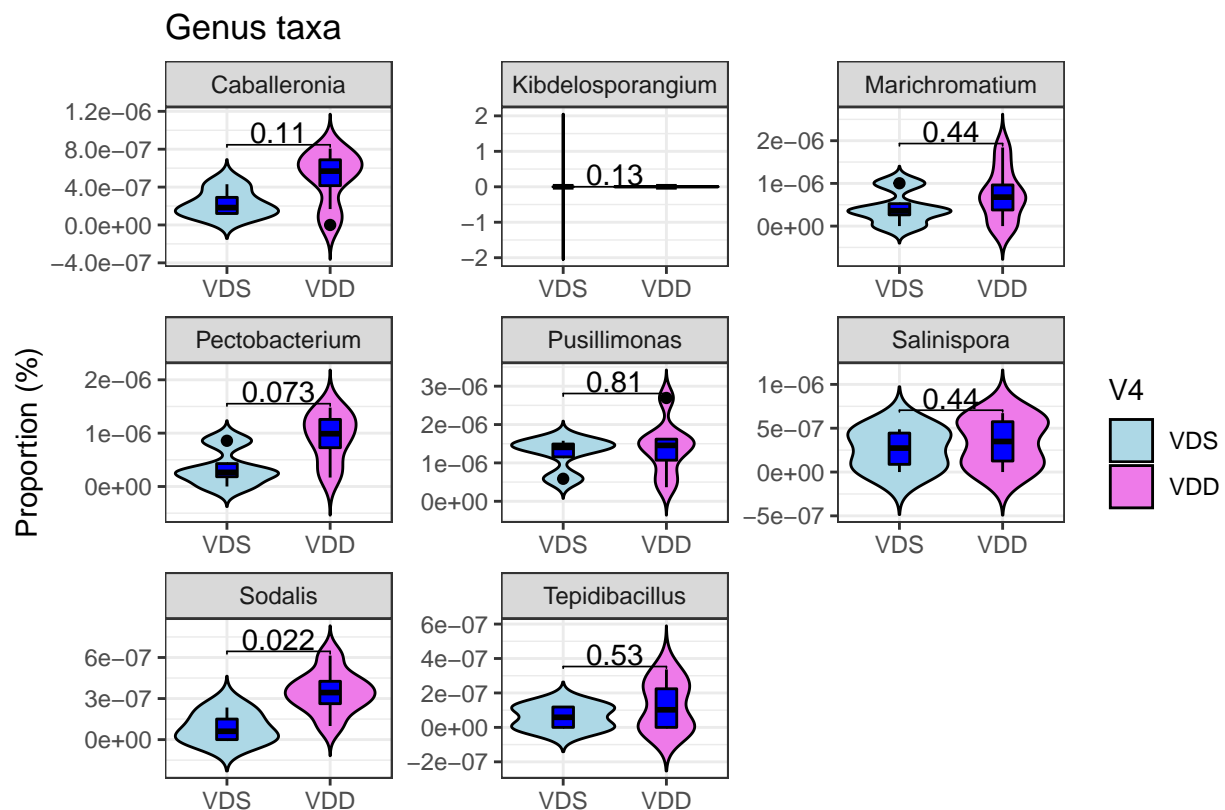
```
## Warning in wilcox.test.default(c(0, 0, 0, 0), c(2.04626314493866e-07, 0, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(0, 3.65924121242371e-07,
## 3.51469099832572e-07, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(0, 4.87898828323161e-07,
## 1.17156366610857e-07, : cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(0, 1.2197470708079e-07, 2.34312733221715e-07, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(0, 1.2197470708079e-07, 1.17156366610857e-07, :
## cannot compute exact p-value with ties
```



## Principal component analysis for species

```
# PC analysis & plot

#install.packages("ggfortify")
library(ggfortify)
#install.packages("FactoMineR")
library("FactoMineR")
library("factoextra")
```

## Welcome! Want to learn more? See two factoextra-related books at <https://goo.gl/ve3WBa>

```
#install.packages("kassambara")
```

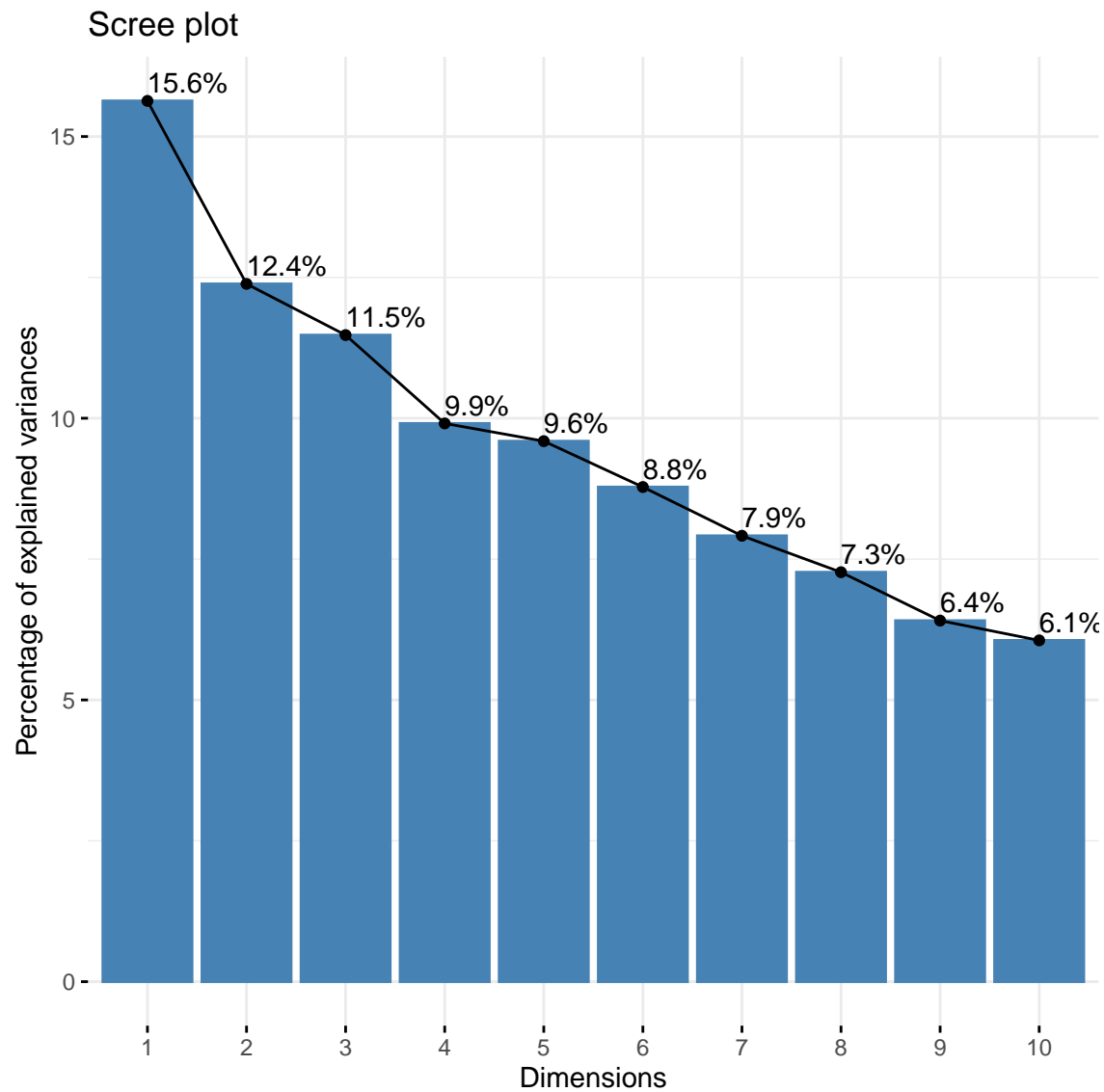
```
# Compute PCA
```

```
speciespca <- prcomp(t(speciesprop), scale = TRUE)
```

```
genuspca <- prcomp(t(genusprop), scale = TRUE)
```

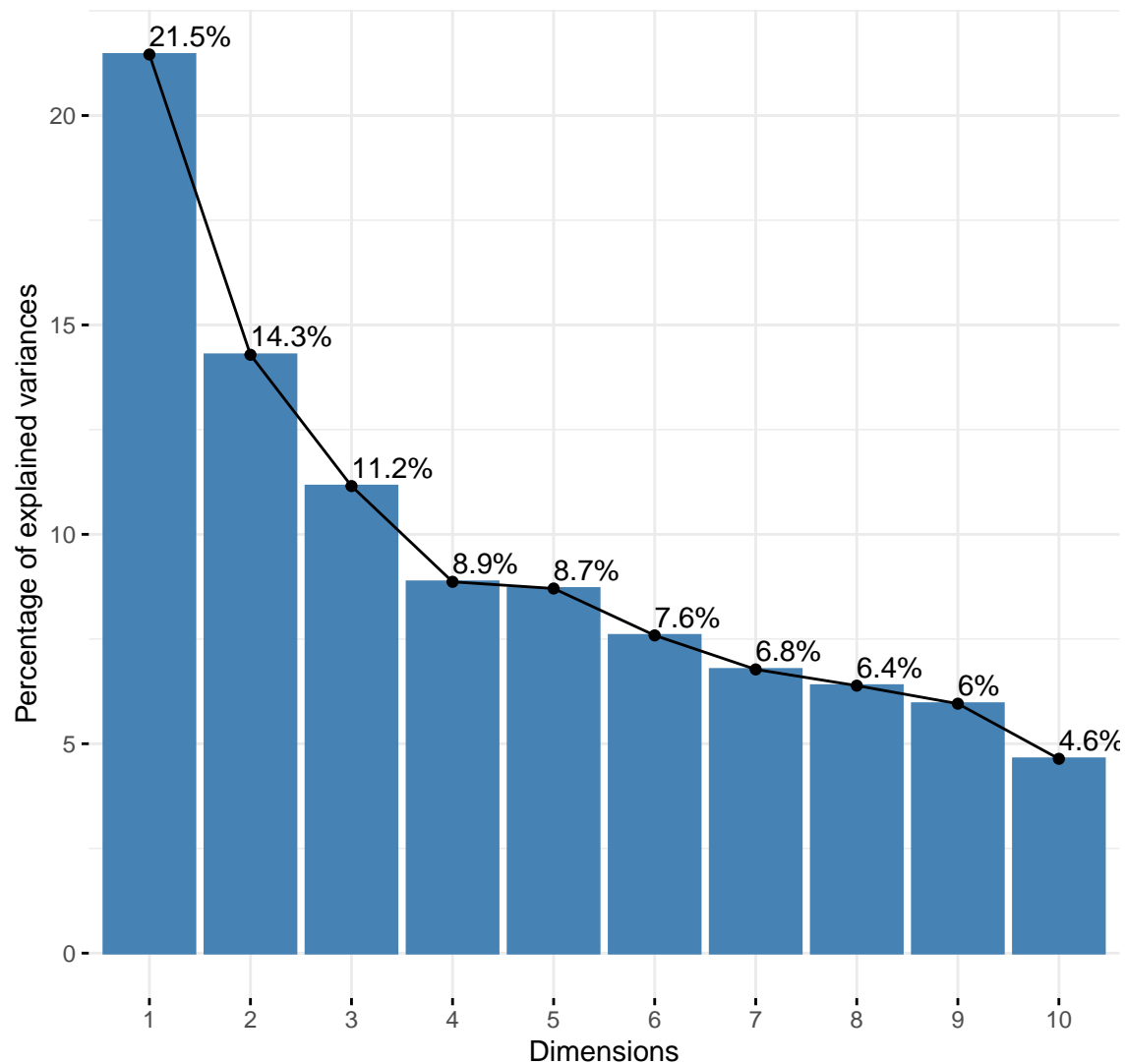
```
# Visualize eigenvalues in scree plot
```

```
fviz_eig(speciespca, addlabels=TRUE)
```



```
fviz_eig(genuspca, addlabels=TRUE)
```

Scree plot



*# See summary values*

```
summary(speciespca)
```

```
## Importance of components:
```

```
##          PC1      PC2      PC3      PC4      PC5      PC6
## Standard deviation 13.1964 11.7458 11.3068 10.50590 10.33780 9.88926
## Proportion of Variance 0.1563 0.1239 0.1148 0.09908 0.09593 0.08779
## Cumulative Proportion 0.1563 0.2802 0.3949 0.49401 0.58994 0.67773
##          PC7      PC8      PC9      PC10     PC11     PC12
## Standard deviation  9.38943 8.99756 8.44916 8.21454 7.1429 7.637e-14
## Proportion of Variance 0.07914 0.07267 0.06408 0.06057 0.0458 0.000e+00
## Cumulative Proportion 0.75687 0.82954 0.89363 0.95420 1.0000 1.000e+00
```

```
summary (genuspca)
```

```
## Importance of components:
```

```
##          PC1      PC2      PC3      PC4      PC5      PC6      PC7
## Standard deviation 13.9043 11.3457 10.0240 8.93778 8.85698 8.26798 7.81248
## Proportion of Variance 0.2146 0.1429 0.1115 0.08866 0.08707 0.07587 0.06774
## Cumulative Proportion 0.2146 0.3574 0.4690 0.55762 0.64469 0.72056 0.78830
##          PC8      PC9      PC10     PC11     PC12
## Standard deviation 7.58610 7.32571 6.4660 6.14145 9.853e-15
## Proportion of Variance 0.06387 0.05956 0.0464 0.04186 0.000e+00
## Cumulative Proportion 0.85217 0.91174 0.9581 1.00000 1.000e+00
```

```
# Get eigenvalues
get_eig(speciespca)
```

```
##          eigenvalue variance.percent cumulative.variance.percent
## Dim.1 1.741442e+02      1.563233e+01          15.63233
## Dim.2 1.379648e+02      1.238463e+01          28.01696
## Dim.3 1.278438e+02      1.147611e+01          39.49306
## Dim.4 1.103740e+02      9.907901e+00          49.40096
## Dim.5 1.068701e+02      9.593362e+00          58.99433
## Dim.6 9.779748e+01      8.778948e+00          67.77327
## Dim.7 8.816135e+01      7.913946e+00          75.68722
## Dim.8 8.095601e+01      7.267146e+00          82.95437
## Dim.9 7.138827e+01      6.408283e+00          89.36265
## Dim.10 6.747867e+01      6.057331e+00          95.41998
## Dim.11 5.102141e+01      4.580019e+00         100.00000
## Dim.12 5.832852e-27      5.235953e-28         100.00000
```

```
get_eig(genuspca)
```

```
##          eigenvalue variance.percent cumulative.variance.percent
## Dim.1 1.933292e+02      2.145718e+01          21.45718
## Dim.2 1.287254e+02      1.428695e+01          35.74413
## Dim.3 1.004799e+02      1.115204e+01          46.89618
## Dim.4 7.988400e+01      8.866148e+00          55.76233
## Dim.5 7.844601e+01      8.706550e+00          64.46888
## Dim.6 6.835951e+01      7.587071e+00          72.05595
## Dim.7 6.103484e+01      6.774123e+00          78.83007
## Dim.8 5.754898e+01      6.387234e+00          85.21730
## Dim.9 5.366602e+01      5.956273e+00          91.17358
## Dim.10 4.180868e+01      4.640253e+00          95.81383
## Dim.11 3.771741e+01      4.186172e+00         100.00000
## Dim.12 9.708092e-29      1.077480e-29         100.00000
```

```
# Individual points pca plot by VDS/VDD
```

```
tt<-speciespca$x[,1:2]
a<-substr(row.names(tt),1,3)
tt<-cbind(tt,a)
tt<-data.frame(tt)

bb<-genuspca$x[,1:2]
c<-substr(row.names(tt),1,3)
bb<-cbind(bb,c)
```

```
bb<-data.frame(bb)
```

```
dev.off()
```

```
## null device  
##      1
```

```
fviz_pca_ind(genuspca,  
             col.ind = a, # color by groups  
             palette = c("#00AFBB", "#FC4E07"),  
             addEllipses = TRUE, # Concentration ellipses  
             ellipse.type = "confidence",  
             legend.title = "Groups",  
             repel = TRUE  
            )  
  
# Individual points pca plot by quality of representation  
fviz_pca_ind(speciespca,  
             col.ind = "cos2",  
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),  
             repel = TRUE # Avoid text overlapping  
            )
```

Subset mdata to look for enrichment overall, rather than within individual taxa

```
# Remove na's from mdata  
mdata_na <-na.omit(mdata)  
  
#install.packages("tidyverse")  
library(tibble)  
  
# Add column with row ID  
mdataID <-rowid_to_column(mdata_na, "ID")  
  
# Subset to include row ID, taxonomic groups and samples only and turn into matrix  
mdata_sub <-subset(mdataID, select=(c(1,7:12,14:25)))  
mdata_sub[,1] <-NULL
```

Overall taxonomic abundance histogram