

Global ocean viral communities

Ford Fishman, Xiaotian Zhou

5/6/2021

Initialize

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

```
library('ggplot2')
```

```
library('vegan')
```

```
## Loading required package: permute
```

```
## Loading required package: lattice
```

```
## This is vegan 2.5-7
```

```
library("corrplot")
```

```
## corrplot 0.84 loaded
```

```
library("dplyr")
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##   filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##   intersect, setdiff, setequal, union
```

```
library("reshape2")
```

```
library("patchwork")
```

```
library("ggpmisc")
```

```
##
```

```
## Attaching package: 'ggpmisc'
```

```
## The following object is masked from 'package:ggplot2':
##
##      annotate

setwd("~/GitHub/QBVirus/")

df <- read.table("data/env-geo-div-class.txt", sep = "\t", header = TRUE, row.names = 1)
vcs.o <- read.csv("data/vcs.csv", row.names = 1)
vcs <- as.data.frame(t(vcs.o))

# remove outlier
vcs <- subset(vcs, ! rownames(vcs) %in% c("MSP114"))
vcs.o <- as.data.frame(t(vcs))
df <- subset(df, ! rownames(df) %in% c("MSP114"))

# numerical features
num_cols <- unlist(lapply(df, is.numeric))

# numerical df
df.num <- df[, num_cols]

# imputed data (will show this is done below)
env.data <- read.csv("data/metadata-imputed.csv", row.names = 1)
```

Plot of samples

```
library(oce)

## Loading required package: gsw

## Loading required package: testthat

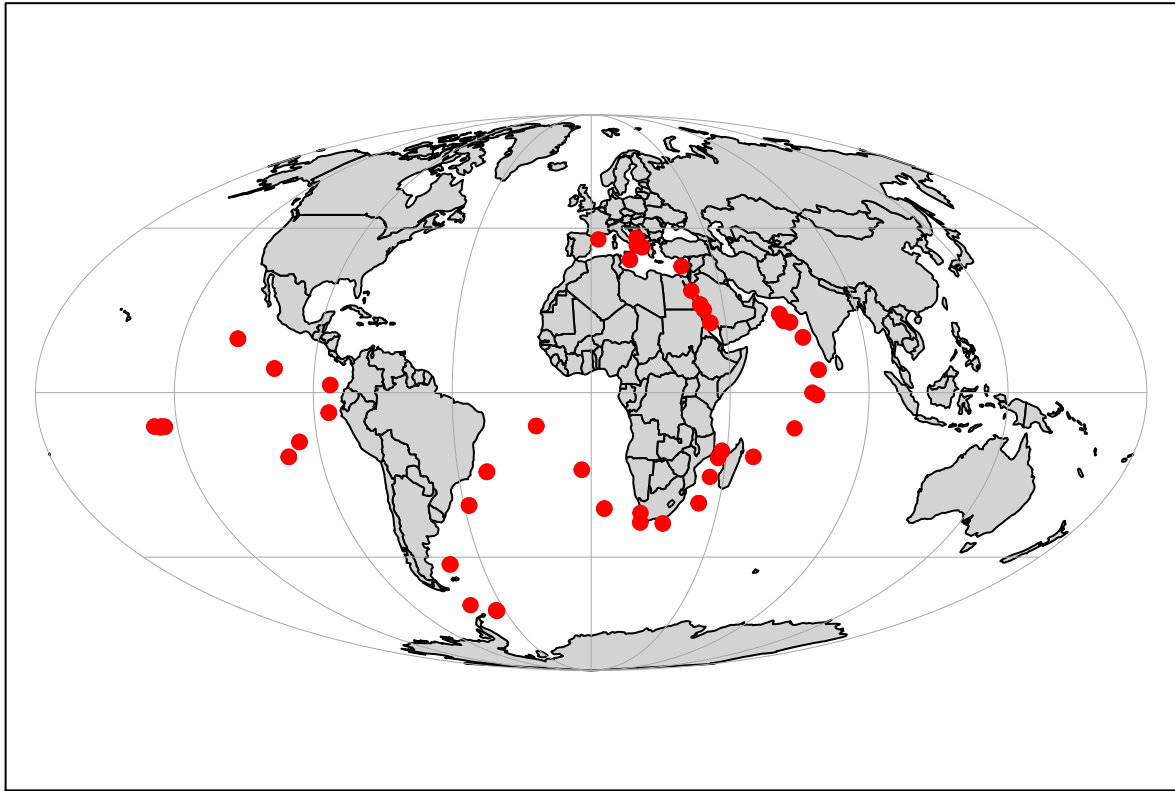
##
## Attaching package: 'testthat'

## The following object is masked from 'package:dplyr':
##
##      matches

## Loading required package: sf

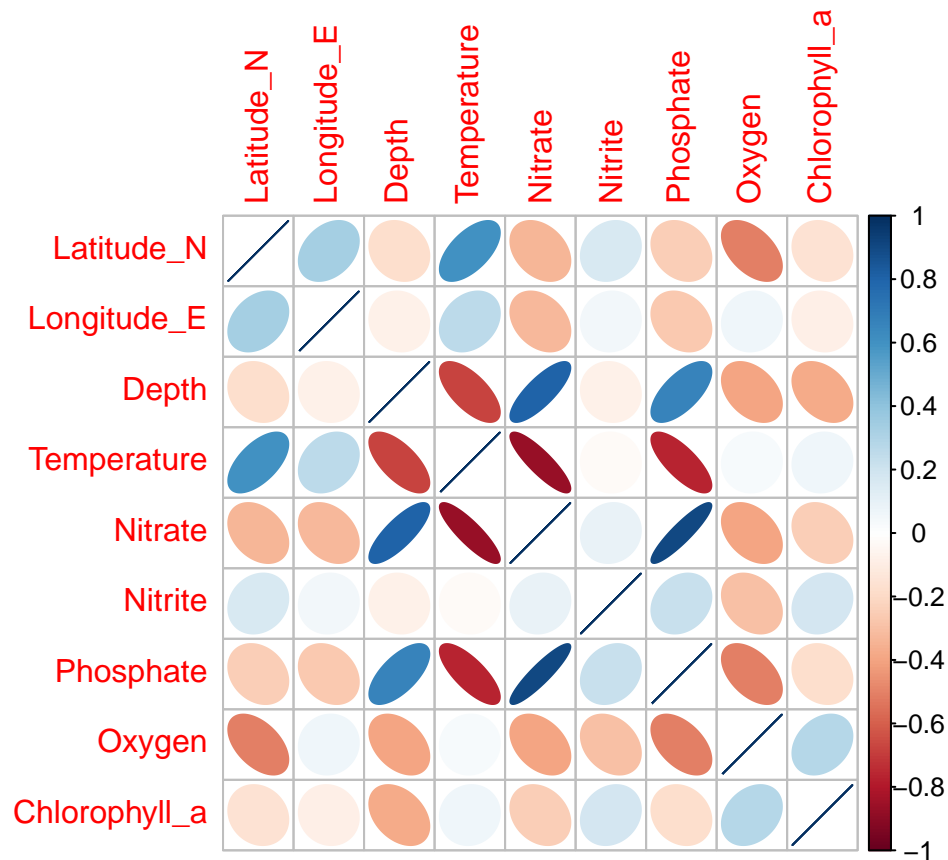
## Linking to GEOS 3.9.0, GDAL 3.2.1, PROJ 7.2.1

data(coastlineWorld)
par(mar=c(1.5, 1.5, 0.5, 0.5))
mapPlot(coastlineWorld, col="lightgrey")
mapPoints(df.num$Longitude_E, df.num$Latitude_N, pch=19, col="red")
```



Environmental info

```
cor1 <- cor(df.num[,1:9], use="complete.obs")  
corrplot(cor1, method = "ellipse")
```



Alpha diversity metrics

```
# richness
S.obs <- function( x = "" ){
  rowSums( x > 0 ) * 1
}

# evenness
Evar <- function(x){
  x <- as.vector(x[x > 0])
  1 - (2/pi)*atan(var(log(x)))
}

# ShannonH
ShanH <- function(x = ""){
  H = 0
  for (n_i in x){
    if(n_i > 0) {
      p = n_i / sum(x)
      H = H - p*log(p)
    }
  }
  return(H)
}

# calculating
vcs.S <- c()
```

```

vcs.E <- c()
vcs.H <- c()
for (i in 1:90) {
  vcs.S[i] = S.obs(vcs[i,])
  vcs.E[i] = Evar(vcs[i,])
  vcs.H[i] = ShanH(vcs[i,])
}
vcs.div <- cbind(vcs.S, vcs.E, vcs.H)

```

Stacked barplots with different grouping strategies

```

#convert to relative abundance table
vcs.rel <- vcs/rowSums(vcs)
# top20 VCs
top <- names(head(sort(colSums(vcs),decreasing = T),20))
top <- as.vector(top)
# recreate a table with the rest VCs classified into "Others"
vcs.rel2 <- vcs.rel
colnames(vcs.rel2)[!colnames(vcs.rel2)%in%top] <- "Others"
others <- rowSums(cbind(vcs.rel2[colnames(vcs.rel2)=="Others"]))
vcs.top <- cbind(vcs.rel2[colnames(vcs.rel2)!="Others"],others)
vcs.top <- vcs.top[,order(colSums(vcs.top))] # ranking
#load group data
env.class.geo.div <- df
vcs.top2 <- as.data.frame(cbind(vcs.top,env.class.geo.div$Layer,env.class.geo.div$Region))
vcs.top2$sample <- rownames(vcs.top2)
vcs.top20 <- melt(vcs.top2,ID="names")

```

Using env.class.geo.div\$Layer, env.class.geo.div\$Region, sample as id variables

```

colnames(vcs.top20)[names(vcs.top20)=="variable"]<-"Taxa"
vcs.top20$`env.class.geo.div$Layer` <- factor(vcs.top20$`env.class.geo.div$Layer`, levels = c("SUR","DCM","MIX","MES"))
# group by layer(zonation)
vcs.top.layer <- aggregate(vcs.top2[,1:21], by = list(Layer = vcs.top2$`env.class.geo.div$Layer`),FUN =
vcs.top.layer$sample <- rownames(vcs.top.layer)
vcs.group.layer <- melt(vcs.top.layer, ID="names")

```

Using Layer, sample as id variables

```

colnames(vcs.group.layer)[names(vcs.group.layer)=="variable"]<-"Taxa"
vcs.group.layer$Layer <- factor(vcs.group.layer$Layer, levels = c("SUR","DCM","MIX","MES")) #set order
# group by region
vcs.top.region <- aggregate(vcs.top2[,1:21], by = list(Layer = vcs.top2$`env.class.geo.div$Region`),FUN =
vcs.top.region$sample <- rownames(vcs.top.region)
vcs.group.region <- melt(vcs.top.region, ID="names")

```

Using Layer, sample as id variables

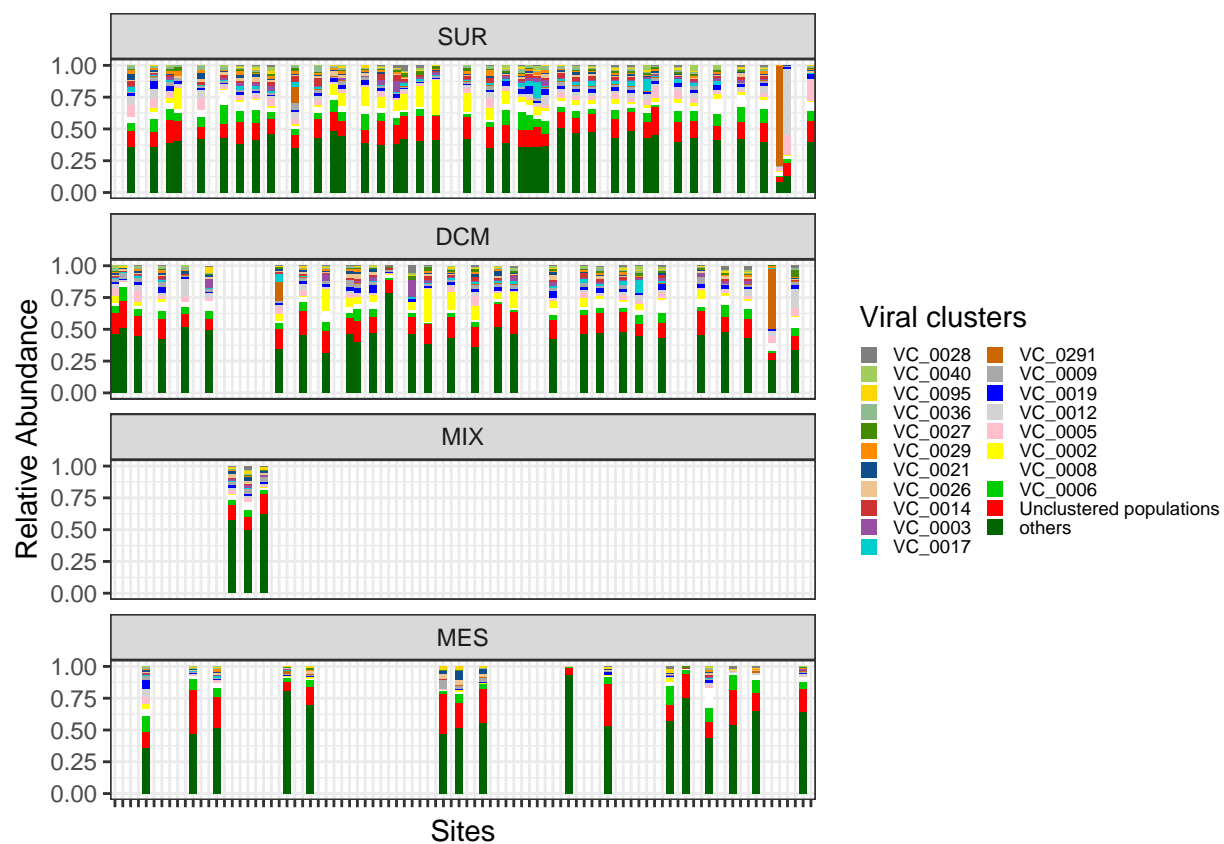
```
colnames(vcs.group.region)[names(vcs.group.region)=="variable"]<-"Taxa"
```

```
# plotting stacked barplot for top 20 viral clusters
```

```
colors<-c("grey50","darkolivegreen3","gold","darkseagreen","chartreuse4","darkorange","dodgerblue4","bu
```

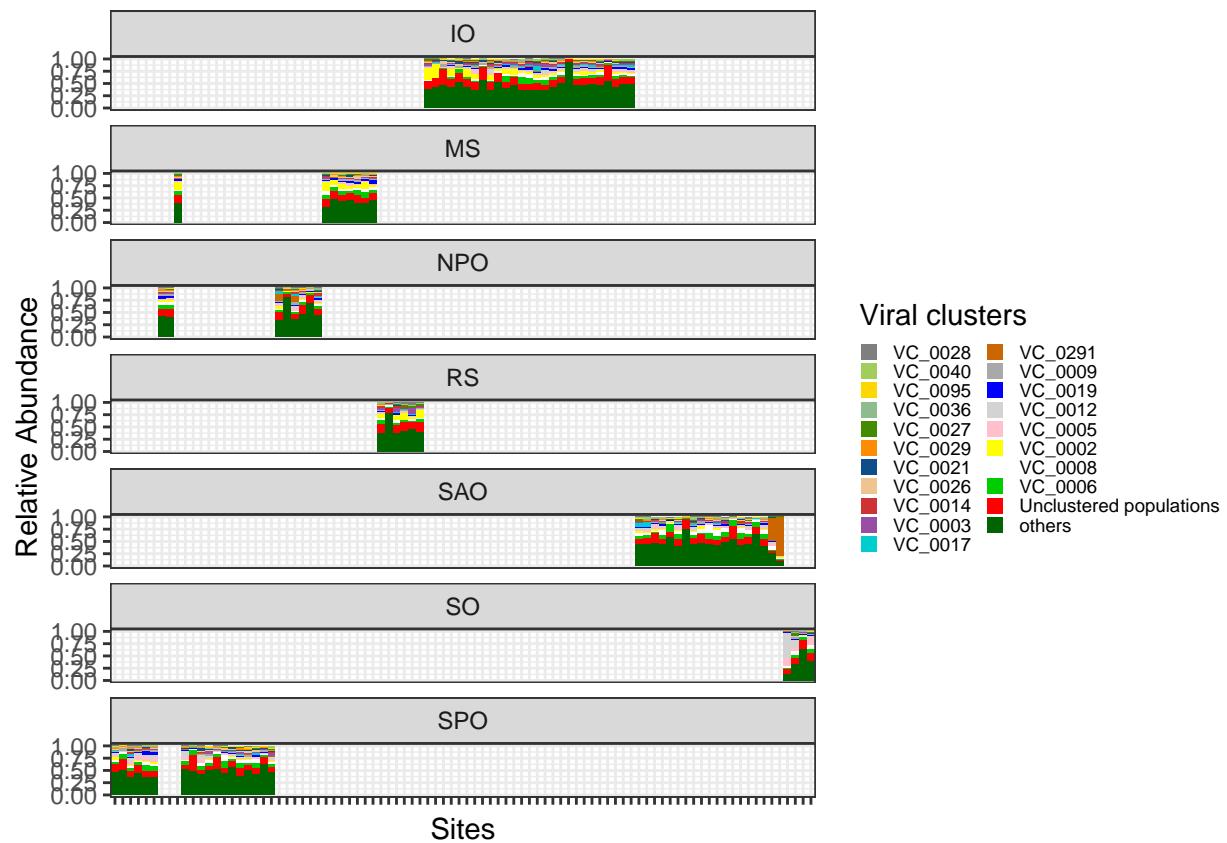
```
# water depth
```

```
ggplot(vcs.top20,aes(x = sample, y = value, fill = Taxa))+
  geom_bar(position = "fill", stat = "identity", width = 1)+
  theme_bw()+
  scale_fill_manual(values=colors)+
  facet_wrap(~'env.class.geo.div$Layer',nrow = 4)+
  labs(x = "Sites",y = "Relative Abundance", fill = "Viral clusters")+
  theme(axis.text.x = element_text(size = 0, color = "transparent"),
        legend.text = element_text(size = 7))+
  guides(fill = guide_legend(keywidth = 0.5, keyheight = 0.5))
```

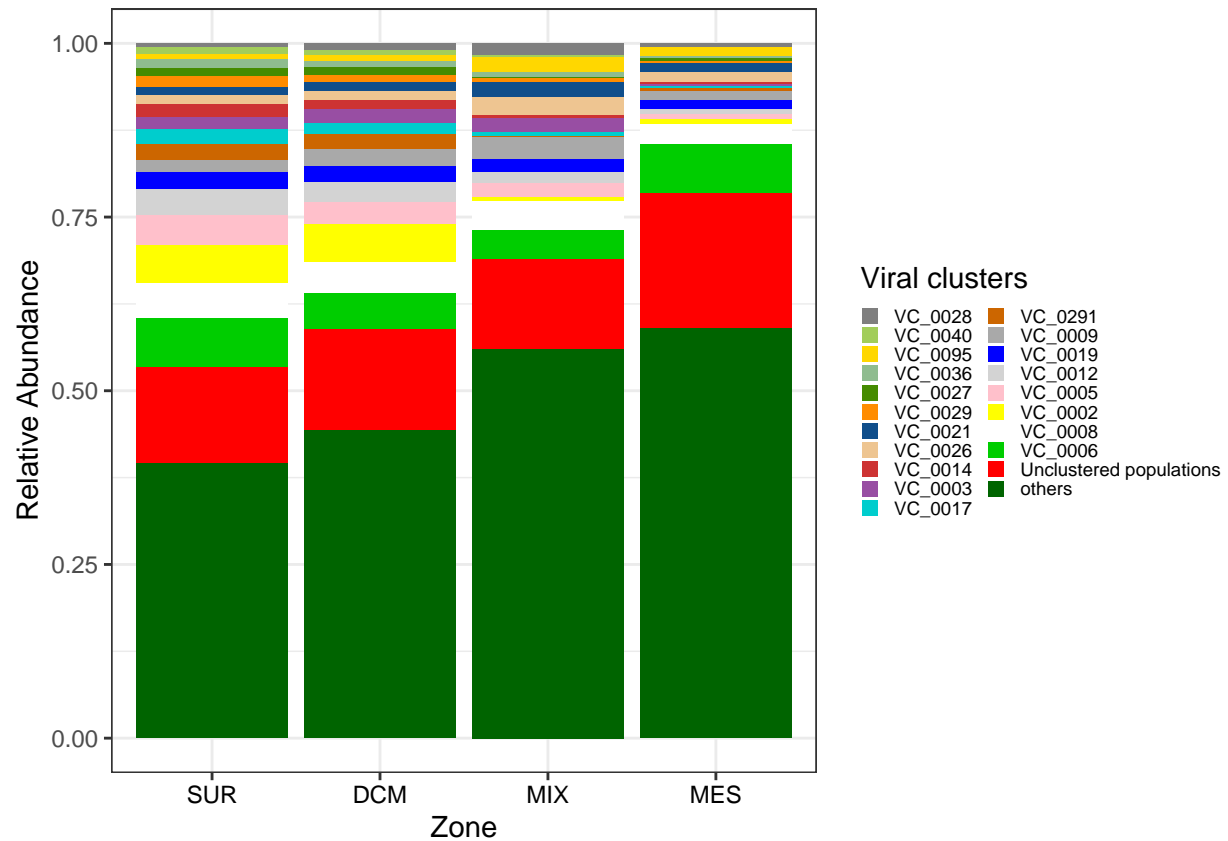


```
# region
```

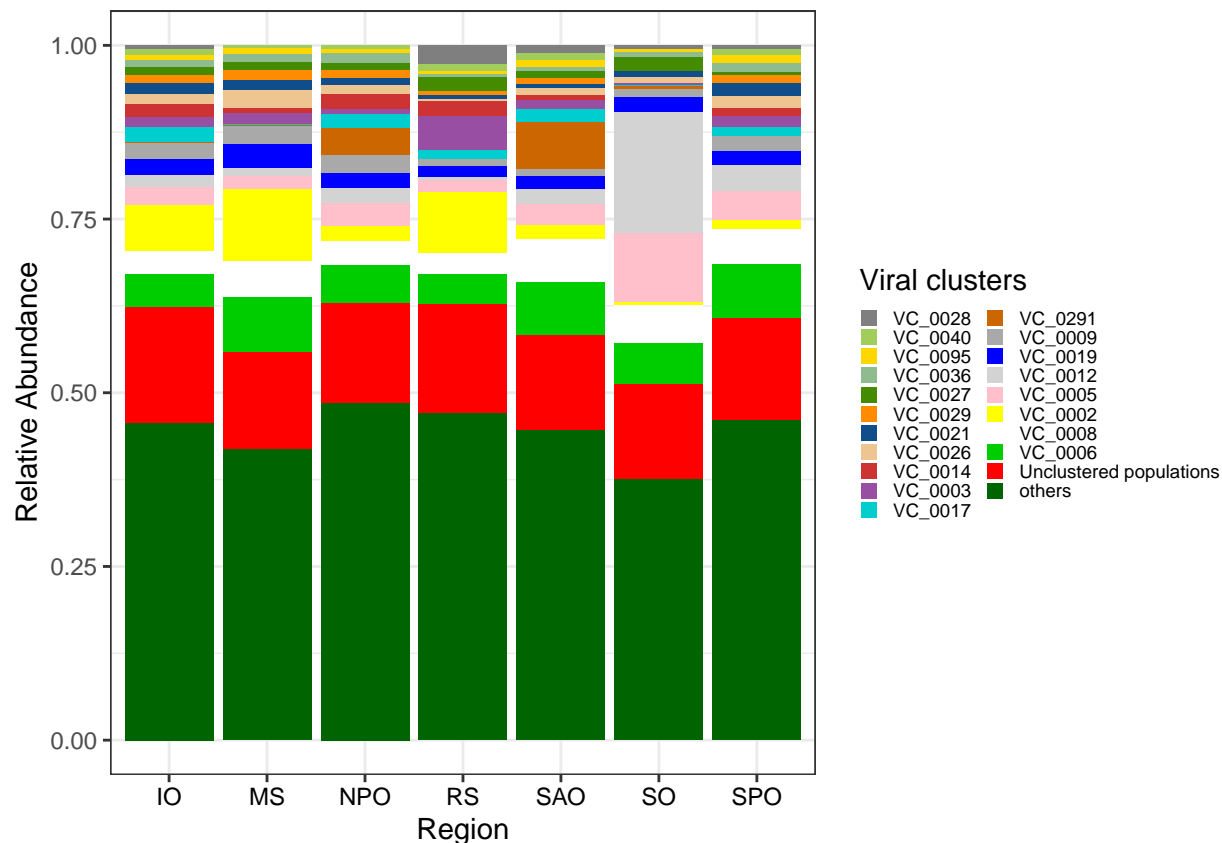
```
ggplot(vcs.top20,aes(x = sample, y = value, fill = Taxa))+
  geom_bar(position = "fill", stat = "identity", width = 1)+
  theme_bw()+
  scale_fill_manual(values=colors)+
  facet_wrap(~'env.class.geo.div$Region',nrow = 7)+
  labs(x = "Sites",y = "Relative Abundance", fill = "Viral clusters")+
  theme(axis.text.x = element_text(size = 0, color = "transparent"),
        legend.text = element_text(size = 7))+
  guides(fill = guide_legend(keywidth = 0.5, keyheight = 0.5))
```



```
# group by water layer
ggplot(vcs.group.layer, aes(x = Layer, y = value, fill = Taxa)) +
  geom_bar(position = "fill", stat = "identity") +
  theme_bw() +
  scale_fill_manual(values=colors) +
  labs(x = "Zone", y = "Relative Abundance", fill = "Viral clusters") +
  theme(axis.text.x = element_text(size = 9, color = "black"),
        legend.text = element_text(size = 7)) +
  guides(fill = guide_legend(keywidth = 0.5, keyheight = 0.5))
```



```
# group by region
ggplot(vcs.group.region, aes(x = Layer, y = value, fill = Taxa)) +
  geom_bar(position = "fill", stat = "identity") +
  theme_bw() +
  scale_fill_manual(values=colors) +
  labs(x = "Region", y = "Relative Abundance", fill = "Viral clusters") +
  theme(axis.text.x = element_text(size = 9, color = "black"),
        legend.text = element_text(size = 7)) +
  guides(fill = guide_legend(keywidth = 0.5, keyheight = 0.5))
```

Seeking relationships with diversity and water depth

```
# linear regression
summary(lm(env.class.geo.div$ShannonH ~ env.class.geo.div$Depth))

##
## Call:
## lm(formula = env.class.geo.div$ShannonH ~ env.class.geo.div$Depth)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.64947 -0.10602  0.09198  0.31019  0.86107
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.8305984   0.0673619   56.866 < 2e-16 ***
## env.class.geo.div$Depth -0.0008597   0.0002332  -3.687 0.000392 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5566 on 88 degrees of freedom
## Multiple R-squared:  0.1338, Adjusted R-squared:  0.1239
## F-statistic: 13.59 on 1 and 88 DF,  p-value: 0.0003923
```

```
summary(lm(env.class.geo.div$Richness ~ env.class.geo.div$Depth))
```

```
##
## Call:
## lm(formula = env.class.geo.div$Richness ~ env.class.geo.div$Depth)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -160.566  -37.782   -2.865   38.934  186.807
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      222.21629      8.47596  26.217 < 2e-16 ***
## env.class.geo.div$Depth  -0.13003      0.02934  -4.431 2.69e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 70.04 on 88 degrees of freedom
## Multiple R-squared:  0.1824, Adjusted R-squared:  0.1731
## F-statistic: 19.64 on 1 and 88 DF,  p-value: 2.687e-05
```

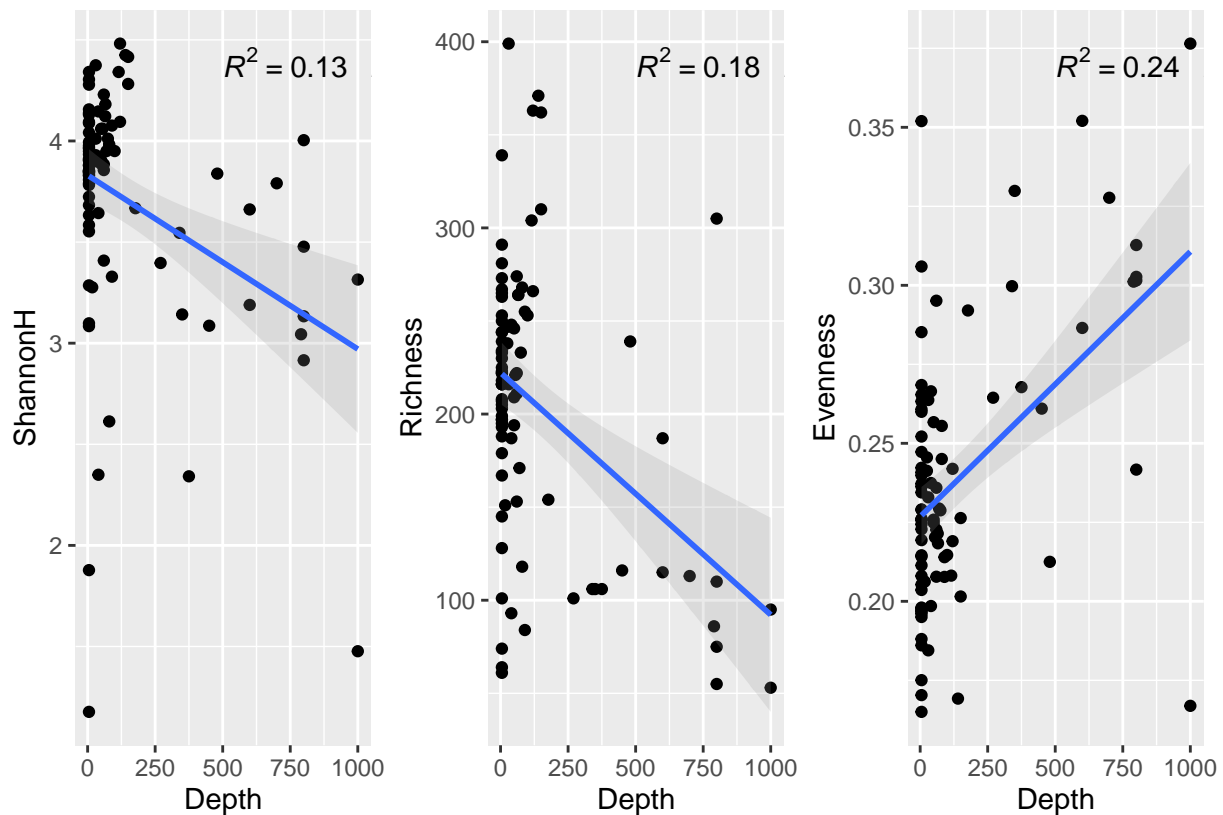
```
summary(lm(env.class.geo.div$Evenness ~ env.class.geo.div$Depth))
```

```
##
## Call:
## lm(formula = env.class.geo.div$Evenness ~ env.class.geo.div$Depth)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.143713 -0.021929 -0.001325  0.016281  0.124779
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.268e-01  4.568e-03  49.639 < 2e-16 ***
## env.class.geo.div$Depth  8.388e-05  1.581e-05   5.304 8.36e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03775 on 88 degrees of freedom
## Multiple R-squared:  0.2422, Adjusted R-squared:  0.2336
## F-statistic: 28.13 on 1 and 88 DF,  p-value: 8.357e-07
```

```
HD <- ggplot(env.class.geo.div, aes(x = Depth, y = ShannonH))+
  geom_point()+
  geom_smooth(method = "lm", alpha = 0.2)+
  stat_poly_eq(formula = y ~ x, aes(label = paste("~~~~~"),..rr.label...,..p.value.label..., sep
SD <- ggplot(env.class.geo.div, aes(x = Depth, y = Richness))+
  geom_point()+
  geom_smooth(method = "lm", alpha = 0.2)+
  stat_poly_eq(formula = y ~ x, aes(label = paste("~~~~~"),..rr.label...,..p.value.label..., sep
ED <- ggplot(env.class.geo.div, aes(x = Depth, y = Evenness))+
  geom_point()+
```

```
geom_smooth(method = "lm", alpha = 0.2)+
stat_poly_eq(formula = y ~ x, aes(label = paste("~~~~~"),..rr.label...,..p.value.label..., sep
HD + SD + ED
```

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



PCoA

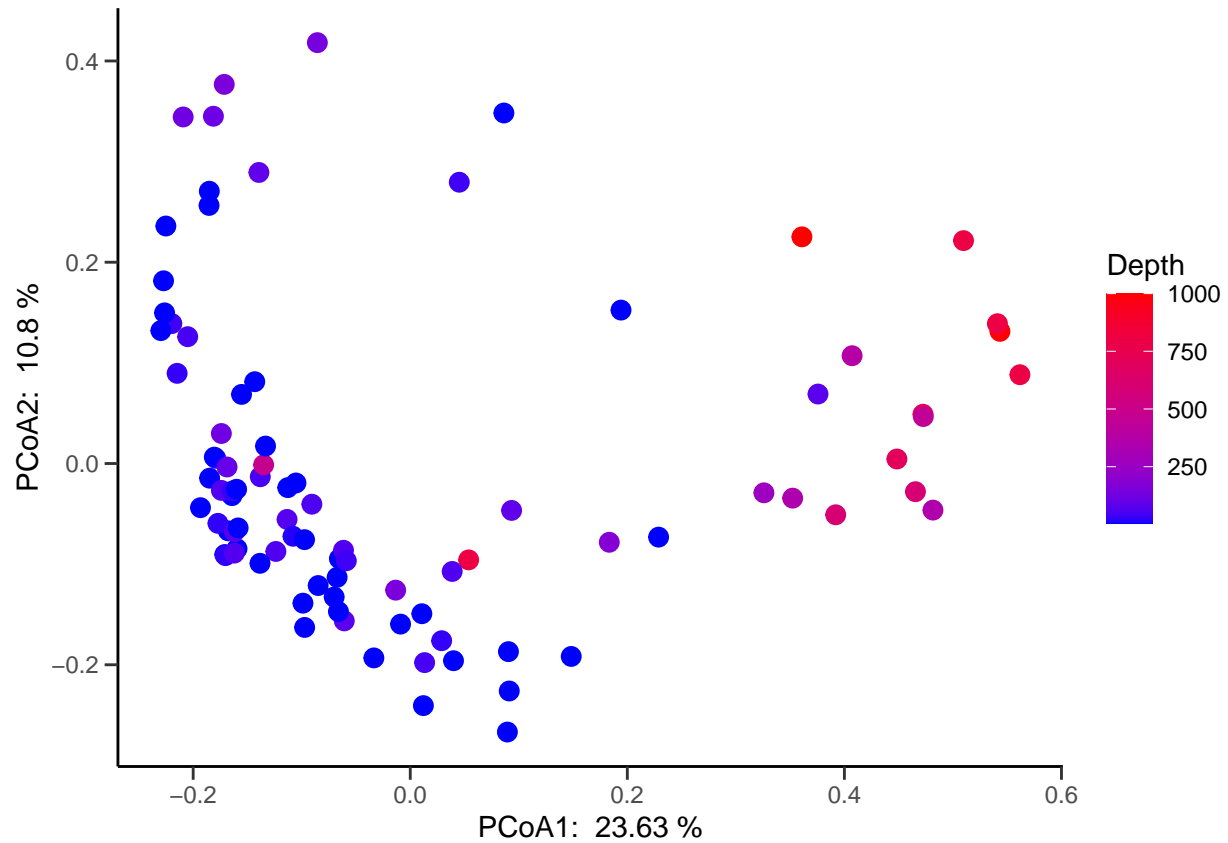
```
#pcoa with depth gradient
bray.vcs <- vegdist(vcs, method = "bray")
pcoa <- cmdscale(bray.vcs, k = 3, eig = TRUE)
pcoa_eig <- (pcoa$eig)[1:2]/sum(pcoa$eig)
site.pcoa <- data.frame(pcoa$points)[1:2]
site.pcoa <- cbind(site.pcoa, df$Depth)
colnames(site.pcoa) <- c("PCoA1", "PCoA2", "Depth")

ggplot(site.pcoa, aes(PCoA1, PCoA2))+
  geom_point(aes(color = Depth), size = 3)+
  scale_color_gradient(low = "blue", high = "red")+
  labs(x=paste("PCoA1: ", round(100*pcoa_eig[1],2), "%"), y = paste("PCoA2: ", round(100*pcoa_eig[2],2), "%"),
  theme(panel.grid.major = element_blank(),
```

```

panel.grid.minor = element_blank(),
panel.background = element_blank(),
axis.line = element_line(colour = "black"),
axis.ticks.length = unit(5, "pt"),

```



Imputation

```
require(mice)
```

```
## Loading required package: mice
```

```
##
```

```
## Attaching package: 'mice'
```

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

```
##
```

```
## filter
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## cbind, rbind
```

```
require(VIM)
```

```
## Loading required package: VIM
```

```
## Loading required package: colorspace
```

```
## Loading required package: grid
```

```
## VIM is ready to use.
```

```
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues
```

```
##
```

```
## Attaching package: 'VIM'
```

```
## The following object is masked from 'package:datasets':
```

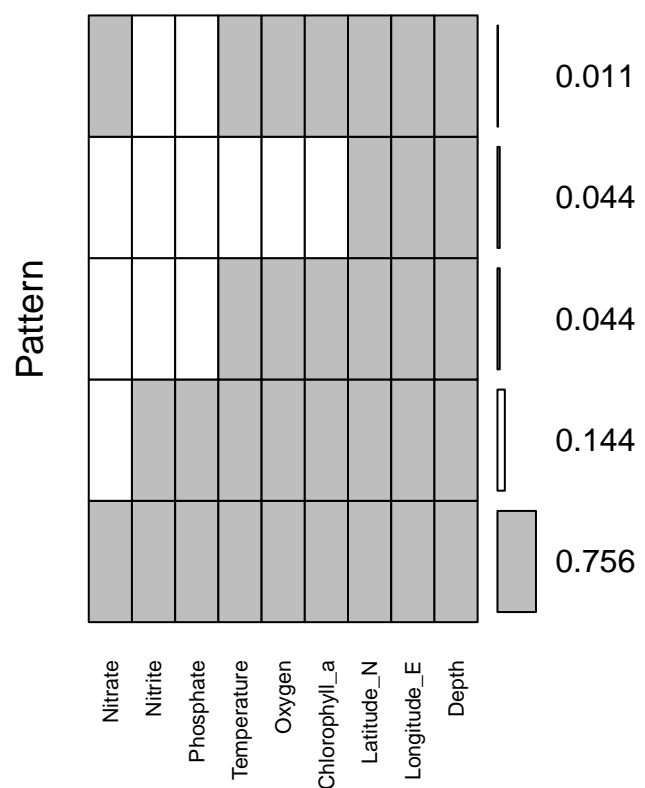
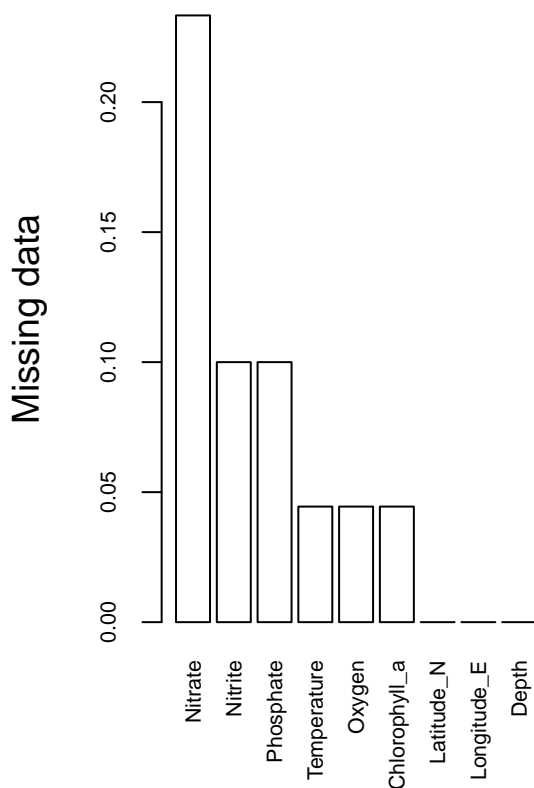
```
##
```

```
##      sleep
```

```
df.num1 <- subset(df.num, ! rownames(df.num) %in% c("MSP114"))
```

```
df.sparse <- subset(df.num1, select=-c(Richness,Evenness,ShannonH))
```

```
mice_plot <- aggr(df.sparse, col=c('grey','white'),
                  numbers=TRUE, sortVars=TRUE,
                  labels=names(df.sparse), cex.axis=.7,
                  gap=3, ylab=c("Missing data", "Pattern"))
```



```
##
## Variables sorted by number of missings:
##      Variable      Count
##      Nitrate 0.23333333
##      Nitrite 0.10000000
##      Phosphate 0.10000000
##      Temperature 0.04444444
##      Oxygen 0.04444444
##      Chlorophyll_a 0.04444444
##      Latitude_N 0.00000000
##      Longitude_E 0.00000000
##      Depth 0.00000000

df.sparse <- subset(df.sparse, select=-c(Nitrate))
imputed_Data <- mice(df.sparse, m=1, maxit = 100, method = 'pmm')
summary(imputed_Data)

completeData <- complete(imputed_Data,1)
# write.csv(completeData, file = "data/metadata-imputed.csv")
```

dbRDA

```
dbrda1 <- dbrda(bray.vcs ~ ., as.data.frame(env.data))
dbrda0 <- dbrda(bray.vcs ~ 1, as.data.frame(env.data))

dbrda <- ordiR2step(dbrda0, scope=formula(dbrda1), perm.max=200)

## Step: R2.adj= 0
## Call: bray.vcs ~ 1
##
##              R2.adjusted
## <All variables> 0.302408505
## + Depth        0.141441726
## + Temperature  0.119685498
## + Phosphate     0.106243382
## + Oxygen        0.046884733
## + Chlorophyll_a 0.043447490
## + Longitude_E  0.042543083
## + Latitude_N   0.037819787
## + Nitrite       0.003117142
## <none>          0.000000000
##
##      Df      AIC      F Pr(>F)
## + Depth  1 255.49 15.662 0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.1414417
## Call: bray.vcs ~ Depth
##
##              R2.adjusted
```

```

## <All variables>    0.3024085
## + Longitude_E      0.1858212
## + Temperature      0.1790677
## + Latitude_N       0.1742009
## + Phosphate        0.1714619
## + Oxygen           0.1691391
## + Chlorophyll_a    0.1585865
## + Nitrite          0.1472391
## <none>             0.1414417
##
##              Df      AIC      F Pr(>F)
## + Longitude_E  1 251.68 5.7967 0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.1858212
## Call: bray.vcs ~ Depth + Longitude_E
##
##              R2.adjusted
## <All variables>    0.3024085
## + Temperature      0.2246926
## + Phosphate        0.2178631
## + Oxygen           0.2156143
## + Latitude_N       0.2111922
## + Chlorophyll_a    0.1999347
## + Nitrite          0.1922520
## <none>             0.1858212
##
##              Df      AIC      F Pr(>F)
## + Temperature  1 248.24 5.3619 0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.2246926
## Call: bray.vcs ~ Depth + Longitude_E + Temperature
##
##              R2.adjusted
## <All variables>    0.3024085
## + Oxygen           0.2616048
## + Latitude_N       0.2501210
## + Chlorophyll_a    0.2437890
## + Phosphate        0.2433198
## + Nitrite          0.2305861
## <none>             0.2246926
##
##              Df      AIC      F Pr(>F)
## + Oxygen       1 244.8 5.2991 0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.2616048
## Call: bray.vcs ~ Depth + Longitude_E + Temperature + Oxygen
##
##              R2.adjusted

```

```

## <All variables>    0.3024085
## + Latitude_N      0.2783222
## + Chlorophyll_a    0.2782592
## + Phosphate        0.2683694
## + Nitrite          0.2641635
## <none>             0.2616048
##
##           Df      AIC      F Pr(>F)
## + Latitude_N  1 243.67 2.969 0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.2783222
## Call: bray.vcs ~ Depth + Longitude_E + Temperature + Oxygen + Latitude_N
##
##           R2.adjusted
## <All variables>    0.3024085
## + Chlorophyll_a    0.2958620
## + Phosphate        0.2847003
## + Nitrite          0.2806636
## <none>             0.2783222
##
##           Df      AIC      F Pr(>F)
## + Chlorophyll_a  1 242.38 3.0924 0.004 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.295862
## Call: bray.vcs ~ Depth + Longitude_E + Temperature + Oxygen + Latitude_N +      Chlorophyll_a
##
##           R2.adjusted
## + Phosphate        0.3024665
## <All variables>    0.3024085
## + Nitrite          0.2964497
## <none>             0.2958620

```

```
permutest(dbrda, permutations=999)
```

```

##
## Permutation test for dbrda under reduced model
##
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = bray.vcs ~ Depth + Longitude_E + Temperature +
## Oxygen + Latitude_N + Chlorophyll_a, data = as.data.frame(env.data))
## Permutation test for all constrained eigenvalues
##           Df Inertia      F Pr(>F)
## Model      6  6.6874 7.2326 0.001 ***
## Residual   83 12.7905
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```



```
envfit(dbrda, env.data[,c(4,6,7)], perm=999)
```

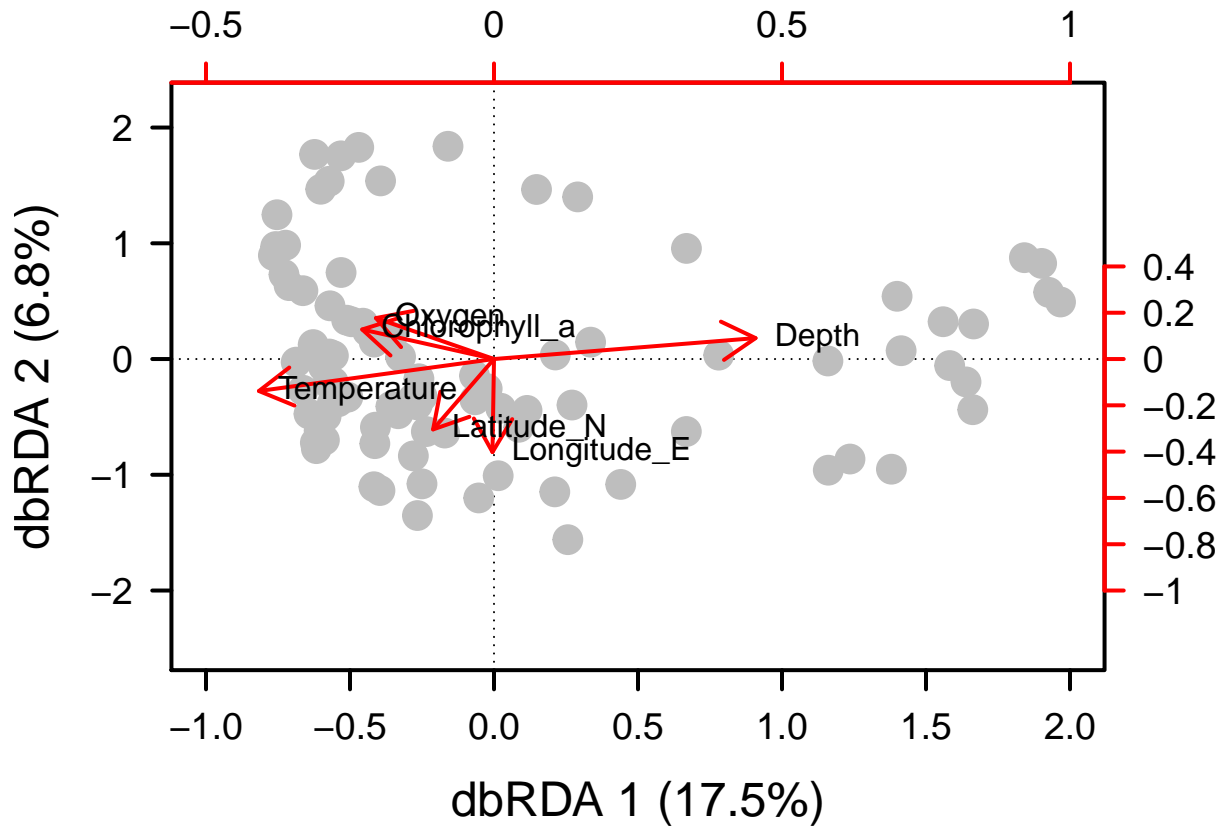
```
##
## ***VECTORS
##
##          dbrDA1    dbrDA2    r2 Pr(>r)
## Temperature -0.93932 -0.34304 0.5741  0.001 ***
## Phosphate    0.97569  0.21914 0.4749  0.001 ***
## Oxygen       -0.77516  0.63177 0.2075  0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
```

```
dbrda.explainvar1 <- round(dbrda$CCA$eig[1]/sum(c(dbrda$CCA$eig, dbrda$CA$eig)),3) * 100
dbrda.explainvar2 <- round(dbrda$CCA$eig[2]/sum(c(dbrda$CCA$eig, dbrda$CA$eig)),3) * 100
```

```
vals <- scores(dbrda, display="wa")
par(mar=c(5,4,2,4) + 0.1)
plot(vals,
      xlim=c(-1, 2), ylim=c(-2.5, 2.2),
      xlab=paste0("dbRDA 1 (", dbrda.explainvar1, "%)"),
      ylab=paste0("dbRDA 2 (", dbrda.explainvar2, "%)"),
      pch=16, cex=2.0, type="n", cex.lab=1.5, cex.axis=1.2, axes=F
)
axis(side=1, labels=T, lwd.ticks=2, cex.axis=1.2, las=1)
axis(side=2, labels=T, lwd.ticks=2, cex.axis=1.2, las=1)
abline(h=0, v=0, lty=3)
box(lwd=2)

points(vals, pch=19, cex=2, bg="grey", col="grey")

vectors <- scores(dbrda, display = "bp")
arrows(0, 0, vectors[,1], vectors[,2], lwd=2, lty=1, length = 0.2, col = "red")
text(vectors[,1], vectors[,2], pos=4, labels=row.names(vectors))
axis(side=3, lwd.ticks = 2, cex.axis=1.2, las=1, col="red", lwd=2.2,
      at = pretty(range(vectors[,1]))*2, labels = pretty(range(vectors[,1])))
axis(side=4, lwd.ticks = 2, cex.axis=1.2, las=1, col="red", lwd=2.2,
      at = pretty(range(vectors[,2]))*2, labels = pretty(range(vectors[,2])))
```



Remove deepest samples

```
vcs.shal <- subset(vcs, df$Layer!="MES")
env.shal <- subset(env.data, df$Layer!="MES")
layers <- subset(df, Layer!="MES", select="Layer")

vcs.shal.rel <- vcs.shal/rowSums(vcs.shal)
vcs.shal.db <- vegdist(vcs.shal.rel, method="bray")
```

Redo ordination without MES

```
vcs.shal.pcoa <- cmdscale(vcs.shal.db, eig=T, k=3)

dbrda1.shal <- dbrda(vcs.shal.db ~ ., as.data.frame(env.shal))
dbrda0.shal <- dbrda(vcs.shal.db ~ 1, as.data.frame(env.shal))

dbrda.shal <- ordiR2step(dbrda0.shal, scope=formula(dbrda1.shal), perm.max=200)
```

```
## Step: R2.adj= 0
## Call: vcs.shal.db ~ 1
##
```

```

##               R2.adjusted
## <All variables> 0.233382428
## + Latitude_N   0.068289468
## + Temperature  0.064867149
## + Longitude_E  0.058183387
## + Depth        0.039783583
## + Oxygen       0.037431878
## + Phosphate    0.035155619
## + Chlorophyll_a 0.020722010
## + Nitrite      0.003067425
## <none>         0.000000000
##
##               Df      AIC      F Pr(>F)
## + Latitude_N  1 169.44 6.3505 0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.06828947
## Call: vcs.shal.db ~ Latitude_N
##
##               R2.adjusted
## <All variables> 0.23338243
## + Temperature  0.12017417
## + Depth        0.10864227
## + Longitude_E  0.10722950
## + Phosphate    0.09186753
## + Oxygen       0.08604975
## + Chlorophyll_a 0.08539670
## + Nitrite      0.07174203
## <none>         0.06828947
##
##               Df      AIC      F Pr(>F)
## + Temperature  1 166.16 5.246 0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.1201742
## Call: vcs.shal.db ~ Latitude_N + Temperature
##
##               R2.adjusted
## <All variables> 0.2333824
## + Depth        0.1627800
## + Longitude_E  0.1592136
## + Chlorophyll_a 0.1367697
## + Oxygen       0.1327408
## + Phosphate    0.1275494
## + Nitrite      0.1243791
## <none>         0.1201742
##
##               Df      AIC      F Pr(>F)
## + Depth        1 163.44 4.6132 0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##

```

```

## Step: R2.adj= 0.16278
## Call: vcs.shal.db ~ Latitude_N + Temperature + Depth
##
##               R2.adjusted
## <All variables> 0.2333824
## + Longitude_E   0.1980321
## + Chlorophyll_a 0.1809458
## + Phosphate     0.1712814
## + Oxygen        0.1697774
## + Nitrite       0.1672336
## <none>          0.1627800
##
##           Df      AIC      F Pr(>F)
## + Longitude_E 1 161.19 4.077 0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.1980321
## Call: vcs.shal.db ~ Latitude_N + Temperature + Depth + Longitude_E
##
##               R2.adjusted
## <All variables> 0.2333824
## + Chlorophyll_a 0.2172276
## + Oxygen        0.2070224
## + Phosphate     0.2040181
## + Nitrite       0.2037865
## <none>          0.1980321
##
##           Df      AIC      F Pr(>F)
## + Chlorophyll_a 1 160.32 2.692 0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.2172276
## Call: vcs.shal.db ~ Latitude_N + Temperature + Depth + Longitude_E +      Chlorophyll_a
##
##               R2.adjusted
## <All variables> 0.2333824
## + Oxygen        0.2257895
## + Phosphate     0.2230164
## + Nitrite       0.2183960
## <none>          0.2172276
##
##           Df      AIC      F Pr(>F)
## + Oxygen 1 160.41 1.752 0.052 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

permutest(dbrda.shal, permutations=999)

##
## Permutation test for dbrda under reduced model
##
## Permutation: free

```

```
## Number of permutations: 999
##
## Model: dbrda(formula = vcs.shal.db ~ Latitude_N + Temperature + Depth +
## Longitude_E + Chlorophyll_a, data = as.data.frame(env.shal))
## Permutation test for all constrained eigenvalues
##      Df Inertia      F Pr(>F)
## Model      5  2.7941 5.0517  0.001 ***
## Residual 68  7.5224
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
envfit(dbrda.shal, env.shal, perm=999)
```

```
##
## ***VECTORS
##
##      dbRDA1  dbRDA2      r2 Pr(>r)
## Latitude_N  0.99868 -0.05143 0.5132  0.001 ***
## Longitude_E  0.96074 -0.27745 0.4470  0.001 ***
## Depth       -0.54679  0.83727 0.1248  0.011 *
## Temperature  0.69593  0.71811 0.6474  0.001 ***
## Nitrite     -0.64552 -0.76374 0.0181  0.511
## Phosphate   -0.79484 -0.60682 0.3209  0.001 ***
## Oxygen      -0.80235 -0.59685 0.2964  0.001 ***
## Chlorophyll_a -0.75721 -0.65317 0.0865  0.055 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
```

```
dbrda.explainvar1.shal <- round(dbrda.shal$CCA$eig[1]/sum(c(dbrda.shal$CCA$eig, dbrda.shal$CA$eig)),3)
dbrda.explainvar2.shal <- round(dbrda.shal$CCA$eig[2]/sum(c(dbrda.shal$CCA$eig, dbrda.shal$CA$eig)),3)
```

```
vals.shal <- scores(dbrda.shal, display="wa")
par(mar=c(5,4,2,4) + 0.1)
plot(vals.shal,
      xlim=c(-1.75, 1.75), ylim=c(-2.2, 2.2),
      xlab=paste0("dbRDA 1 (", dbrda.explainvar1.shal, "%)"),
      ylab=paste0("dbRDA 2 (", dbrda.explainvar2.shal, "%)"),
      pch=16, cex=2.0, type="n", cex.lab=1.5, cex.axis=1.2, axes=F
)
axis(side=1, labels=T, lwd.ticks=2, cex.axis=1.2, las=1)
axis(side=2, labels=T, lwd.ticks=2, cex.axis=1.2, las=1)
abline(h=0, v=0, lty=3)
box(lwd=2)

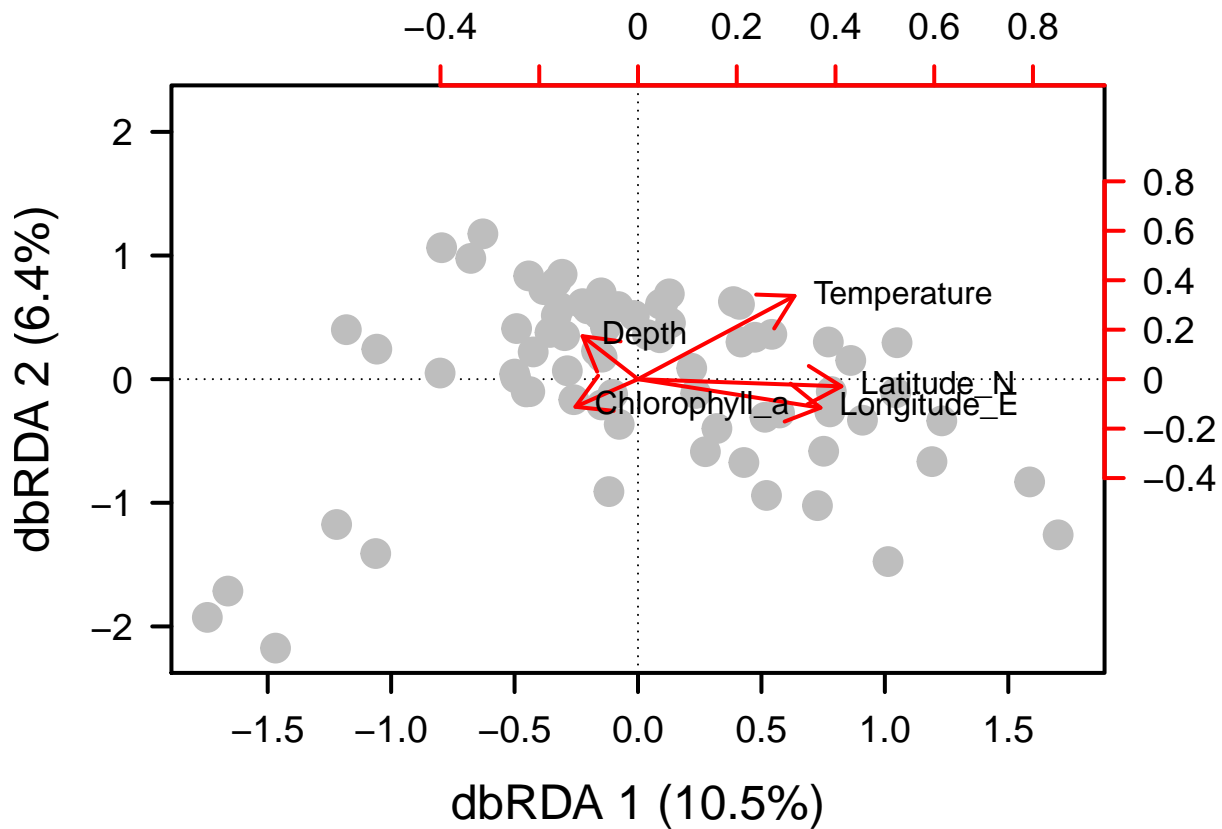
points(vals.shal, pch=19, cex=2, bg="grey", col="grey")

vectors.shal <- scores(dbrda.shal, display = "bp")
arrows(0, 0, vectors.shal[,1], vectors.shal[,2], lwd=2, lty=1, length = 0.2, col = "red")
text(vectors.shal[,1], vectors.shal[,2], pos=4, labels=row.names(vectors.shal))
axis(side=3, lwd.ticks = 2, cex.axis=1.2, las=1, col="red", lwd=2.2,
```

```

at = pretty(range(vectors.shal[,1]))*2, labels = pretty(range(vectors.shal[,1]))
axis(side=4, lwd.ticks = 2, cex.axis=1.2, las=1, col="red", lwd=2.2,
at = pretty(range(vectors.shal[,2]))*2, labels = pretty(range(vectors.shal[,2]))

```



Distance decay

```

library(geosphere)
library(ggplot2)
library(ggpmisc)

# geographical distance
vcs.geo <- cbind(df.num$Longitude_E, df.num$Latitude_N)
dist.vcs <- as.dist(distm(vcs.geo, fun = distHaversine))
dist_vcs <- as.data.frame(as.vector(dist.vcs))/1000 #transformed to kilometer
colnames(dist_vcs) <- "dist"

# depth distance
dep.m <- matrix(nrow = 90, ncol = 90)
for (i in 1:90){
  for (j in 1:90){
    dep.m[i,j] = abs(df$Depth[i]-df$Depth[j])
  }
}
depth_dis <- as.data.frame(as.vector(as.dist(dep.m)))

```

```

colnames(depth_dis) <- "depth"

# bray-curtis dissimilarity
bray.vcs <- vegdist(vcs, method = "bray")
bray_vcs <- as.data.frame(as.vector(bray.vcs))
colnames(bray_vcs) <- "bray"
data.dist <- data.frame(dist_vcs, depth_dis, bray_vcs)
data.dist$bray <- (1-data.dist$bray)*100 #convert to similarity percentage
summary(lm(data.dist$bray ~ data.dist$dist))

```

```

##
## Call:
## lm(formula = data.dist$bray ~ data.dist$dist)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -36.79 -14.53   2.93  14.10  50.27
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.030e+01  5.423e-01  74.301  <2e-16 ***
## data.dist$dist -4.549e-04  5.147e-05  -8.839  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.14 on 4003 degrees of freedom
## Multiple R-squared:  0.01914, Adjusted R-squared:  0.0189
## F-statistic: 78.12 on 1 and 4003 DF, p-value: < 2.2e-16

```

```

summary(lm(data.dist$bray ~ data.dist$depth))

```

```

##
## Call:
## lm(formula = data.dist$bray ~ data.dist$depth)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -38.135  -8.475   0.432   8.991  60.204
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   44.2382697  0.2667014  165.87  <2e-16 ***
## data.dist$depth -0.0377462  0.0007453  -50.64  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.51 on 4003 degrees of freedom
## Multiple R-squared:  0.3905, Adjusted R-squared:  0.3904
## F-statistic: 2565 on 1 and 4003 DF, p-value: < 2.2e-16

```

```

#plot
DDR.geo <- ggplot(data.dist, aes(x = dist, y = bray))+

```

```

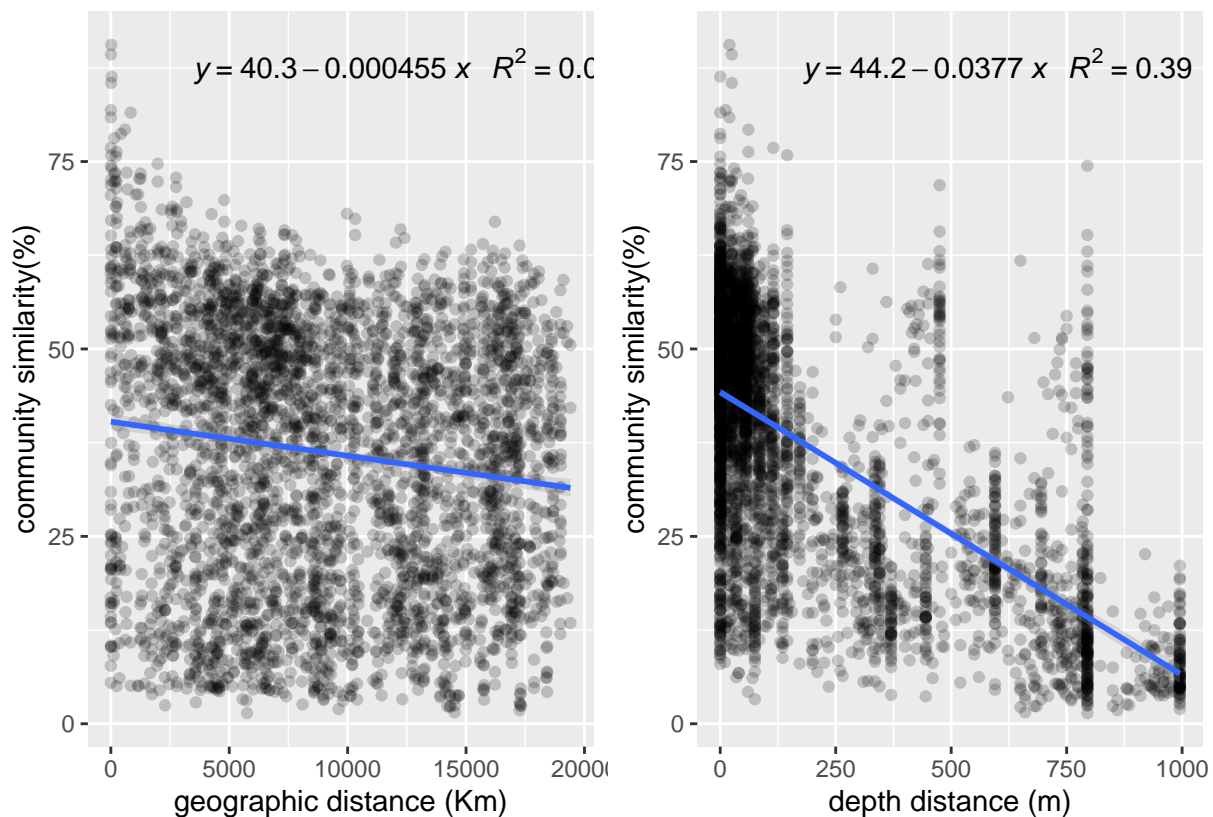
geom_point(alpha = 0.2)+
geom_smooth(method = "lm", alpha = 0.3)+
labs(x = "geographic distance (Km)", y = "community similarity(%)" )+
stat_poly_eq(formula = y ~ x, aes(label = paste("~~~~~",..eq.label.., ..rr.label.., ..p.value.label..)))
DDR.depth <- ggplot(data.dist, aes(x = depth, y = bray))+
geom_point(alpha = 0.2)+
geom_smooth(method = "lm", alpha = 0.3)+
labs(x = "depth distance (m)", y = "community similarity(%)" )+
stat_poly_eq(formula = y ~ x, aes(label = paste("~~~~~",..eq.label.., ..rr.label.., ..p.value.label..)))
DDR.geo + DDR.depth

```

```

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

```

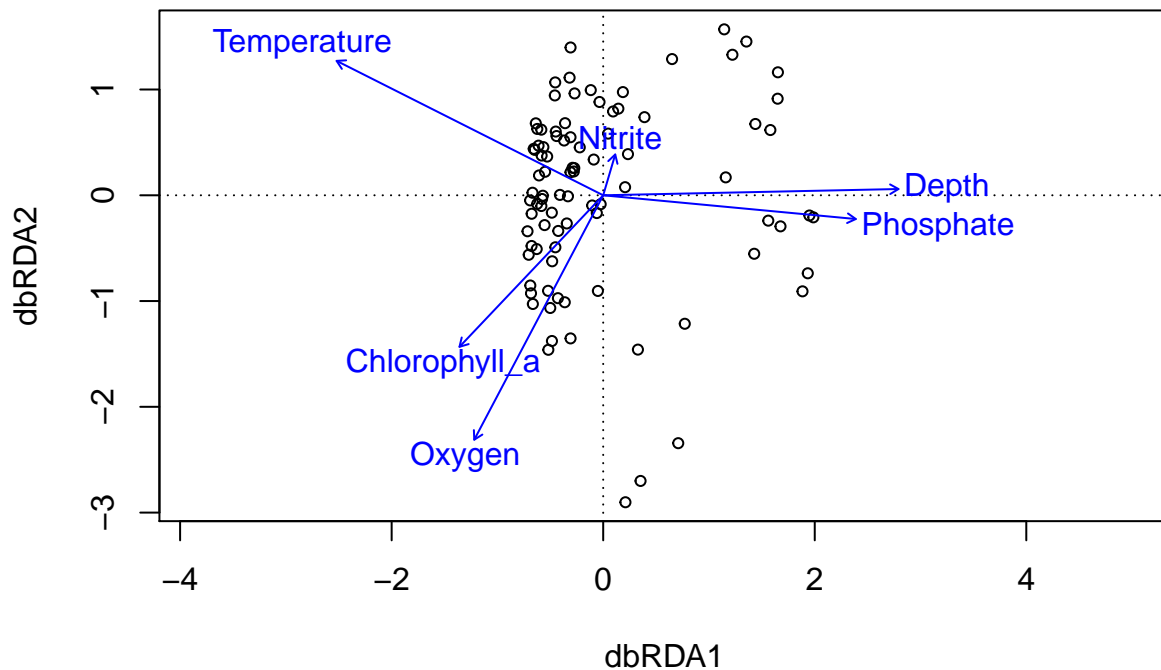


Variation partitioning (two-factor)

```

# vcs table without site #MSP114
vcs.db <- vegdist(vcs, method = "bray", upper = TRUE, diag = TRUE)
# env with depth
env <- as.matrix(env.data[,3:8])
#perform dbRDA
vcs.dbrda <- dbRDA(vcs.db ~ ., as.data.frame(env))
ordiplot(vcs.dbrda)

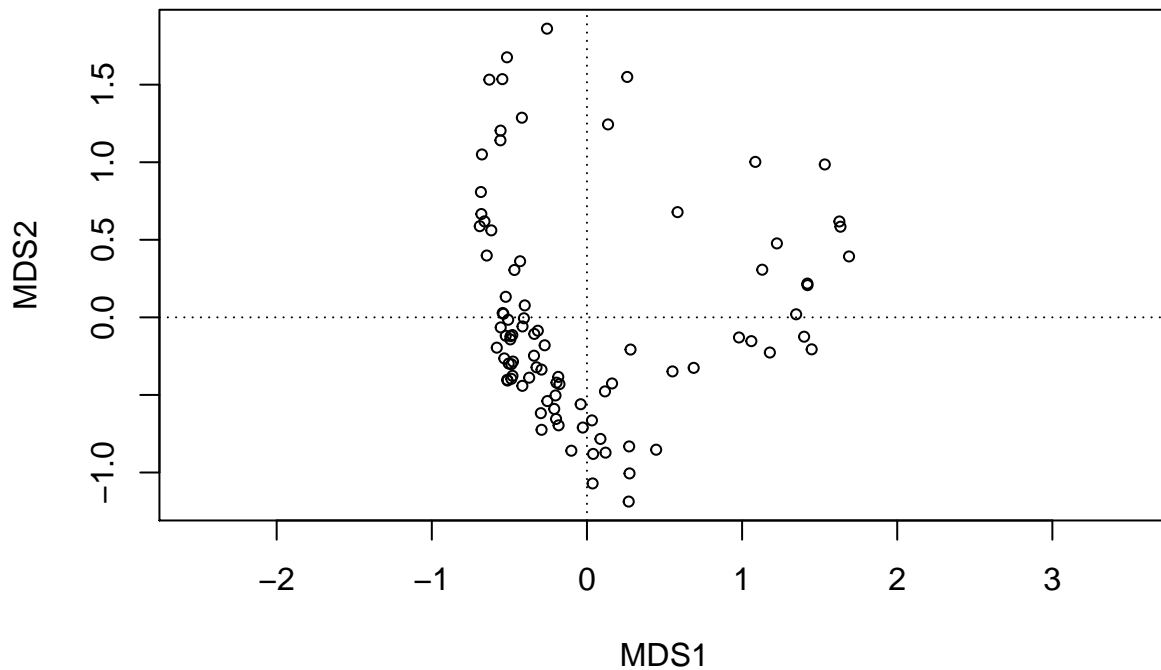
```

```
psych::corr.test(env)
```

```
## Call:psych::corr.test(x = env)
## Correlation matrix
##           Depth Temperature Nitrite Phosphate Oxygen Chlorophyll_a
## Depth      1.00      -0.66   -0.06     0.64   -0.37      -0.35
## Temperature -0.66       1.00   -0.02    -0.71    0.04       0.06
## Nitrite     -0.06     -0.02    1.00     0.23   -0.28       0.20
## Phosphate    0.64    -0.71    0.23     1.00   -0.55      -0.16
## Oxygen      -0.37     0.04   -0.28    -0.55    1.00       0.29
## Chlorophyll_a -0.35     0.06    0.20    -0.16    0.29       1.00
## Sample Size
## [1] 90
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##           Depth Temperature Nitrite Phosphate Oxygen Chlorophyll_a
## Depth      0.00      0.00    1.00     0.00    0.00       0.01
## Temperature 0.00      0.00    1.00     0.00    1.00       1.00
## Nitrite     0.56     0.83    0.00     0.22    0.06       0.39
## Phosphate    0.00      0.00    0.03     0.00    0.00       0.64
## Oxygen       0.00     0.73    0.01     0.00    0.00       0.05
## Chlorophyll_a 0.00     0.58    0.06     0.13    0.01       0.00
##
## To see confidence intervals of the correlations, print with the short=FALSE option
```

```
# Environmental variables are significantly correlated.
vcs.dbrda.mod0 <- dbrda(vcs.db ~ 1, as.data.frame(env))
ordiplot(vcs.dbrda.mod0)
```



```
vcs.dbrda.mod1 <- dbrda(vcs.db ~ ., as.data.frame(env))
vcs.dbrda <- ordiR2step(vcs.dbrda.mod0, vcs.dbrda.mod1, perm.max = 200)
```

```
## Step: R2.adj= 0
## Call: vcs.db ~ 1
##
##               R2.adjusted
## <All variables> 0.237389871
## + Depth        0.141441726
## + Temperature  0.119685498
## + Phosphate     0.106243382
## + Oxygen        0.046884733
## + Chlorophyll_a 0.043447490
## + Nitrite       0.003117142
## <none>          0.000000000
##
##      Df    AIC      F Pr(>F)
## + Depth 1 255.49 15.662 0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```

## Step: R2.adj= 0.1414417
## Call: vcs.db ~ Depth
##
##           R2.adjusted
## <All variables> 0.2373899
## + Temperature 0.1790677
## + Phosphate 0.1714619
## + Oxygen 0.1691391
## + Chlorophyll_a 0.1585865
## + Nitrite 0.1472391
## <none> 0.1414417
##
##           Df      AIC      F Pr(>F)
## + Temperature 1 252.43 5.0333 0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.1790677
## Call: vcs.db ~ Depth + Temperature
##
##           R2.adjusted
## <All variables> 0.2373899
## + Oxygen 0.2098507
## + Chlorophyll_a 0.1998654
## + Phosphate 0.1952861
## + Nitrite 0.1844586
## <none> 0.1790677
##
##           Df      AIC      F Pr(>F)
## + Oxygen 1 249.95 4.3894 0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.2098507
## Call: vcs.db ~ Depth + Temperature + Oxygen
##
##           R2.adjusted
## <All variables> 0.2373899
## + Chlorophyll_a 0.2286253
## + Phosphate 0.2196416
## + Nitrite 0.2103001
## <none> 0.2098507
##
##           Df      AIC      F Pr(>F)
## + Chlorophyll_a 1 248.73 3.0932 0.004 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.2286253
## Call: vcs.db ~ Depth + Temperature + Oxygen + Chlorophyll_a
##
##           R2.adjusted
## + Phosphate 0.2381854
## <All variables> 0.2373899

```

```
## <none>          0.2286253
## + Nitrite       0.2281003
```

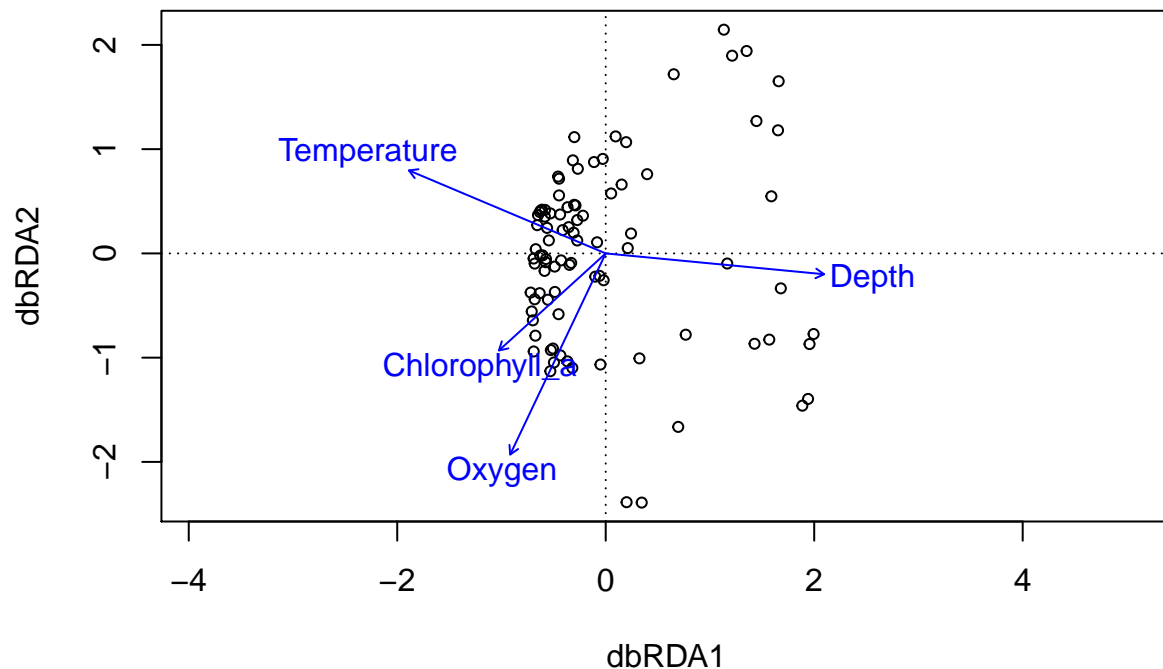
```
vcs.dbrda$call
```

```
## dbrda(formula = vcs.db ~ Depth + Temperature + Oxygen + Chlorophyll_a,
##       data = as.data.frame(env))
```

```
vcs.dbrda$anova
```

```
##           R2.adj Df      AIC      F Pr(>F)
## + Depth      0.14144 1 255.49 15.6622 0.002 **
## + Temperature 0.17907 1 252.43  5.0333 0.002 **
## + Oxygen      0.20985 1 249.95  4.3894 0.002 **
## + Chlorophyll_a 0.22862 1 248.73  3.0932 0.004 **
## <All variables> 0.23739
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ordiplot(vcs.dbrda)
```



```
env.mod <- model.matrix(~ Depth + Temperature + Oxygen + Chlorophyll_a, as.data.frame(env))[, -1]
```

```
# spatial model with PCNM
```

```
rs.vcs <- rowSums(vcs)/sum(vcs)
vcs.dis <- cbind(env.data$Latitude_N, env.data$Longitude_E)
vcs.pcnmw <- pcnm(dist(vcs.dis), w = rs.vcs, dist.ret = T)
vcs.pcnmw$values>0
```

```
## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [13] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [25] TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
#perform model selection for spatial matrix
```

```
vcs.space <- as.data.frame(scores(vcs.pcnmw))
vcs.pcnm.mod0 <- dbrda(vcs.db ~ 1, vcs.space)
vcs.pcnm.mod1 <- dbrda(vcs.db ~ ., vcs.space)
step.pcnm <- ordiR2step(vcs.pcnm.mod0, vcs.pcnm.mod1, perm.max = 200)
```

```
## Step: R2.adj= 0
```

```
## Call: vcs.db ~ 1
```

```
##
```

```
## R2.adjusted
```

```
## <All variables> 0.0962770053
```

```
## + PCNM1 0.0372899800
```

```
## + PCNM2 0.0337864325
```

```
## + PCNM15 0.0170560648
```

```
## + PCNM3 0.0119662318
```

```
## + PCNM4 0.0102935626
```

```
## + PCNM5 0.0090794575
```

```
## + PCNM14 0.0081380356
```

```
## + PCNM6 0.0032382404
```

```
## + PCNM19 0.0031318508
```

```
## + PCNM9 0.0013037913
```

```
## + PCNM17 0.0003197189
```

```
## <none> 0.0000000000
```

```
## + PCNM24 -0.0002517278
```

```
## + PCNM7 -0.0002650326
```

```
## + PCNM8 -0.0003707471
```

```
## + PCNM18 -0.0009237171
```

```
## + PCNM11 -0.0020593000
```

```
## + PCNM12 -0.0021780594
```

```
## + PCNM20 -0.0027678804
```

```
## + PCNM16 -0.0034954716
```

```
## + PCNM22 -0.0043393010
```

```
## + PCNM21 -0.0045196004
```

```
## + PCNM10 -0.0062164050
```

```
## + PCNM26 -0.0064536481
```

```
## + PCNM23 -0.0065543363
```

```
## + PCNM25 -0.0066384102
```

```
## + PCNM13 -0.0069773378
```

```
## + PCNM27 -0.0071189589
```

```
##
```

```
## Df AIC F Pr(>F)
```

```
## + PCNM1 1 265.79 4.4474 0.002 **
```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.03728998
## Call: vcs.db ~ PCNM1
##
##               R2.adjusted
## <All variables> 0.09627701
## + PCNM2        0.07129410
## + PCNM15       0.05497074
## + PCNM3        0.05061667
## + PCNM4        0.04927912
## + PCNM14       0.04856839
## + PCNM5        0.04715579
## + PCNM19       0.04097574
## + PCNM6        0.04055086
## + PCNM9        0.03913683
## + PCNM17       0.03796937
## + PCNM7        0.03746023
## + PCNM24       0.03733292
## <none>         0.03728998
## + PCNM8        0.03706036
## + PCNM18       0.03635466
## + PCNM11       0.03570246
## + PCNM12       0.03539870
## + PCNM16       0.03489007
## + PCNM20       0.03456254
## + PCNM22       0.03347677
## + PCNM21       0.03316719
## + PCNM10       0.03131661
## + PCNM26       0.03122488
## + PCNM25       0.03099457
## + PCNM23       0.03093995
## + PCNM27       0.03050025
## + PCNM13       0.03032424
##
##           Df      AIC      F Pr(>F)
## + PCNM2   1 263.53 4.2221 0.004 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.0712941
## Call: vcs.db ~ PCNM1 + PCNM2
##
##               R2.adjusted
## <All variables> 0.09627701
## + PCNM3        0.08535091
## + PCNM15       0.08492789
## + PCNM14       0.08326007
## + PCNM5        0.08287832
## + PCNM4        0.08174324
## + PCNM6        0.07589838
## + PCNM19       0.07532932
## + PCNM9        0.07249684

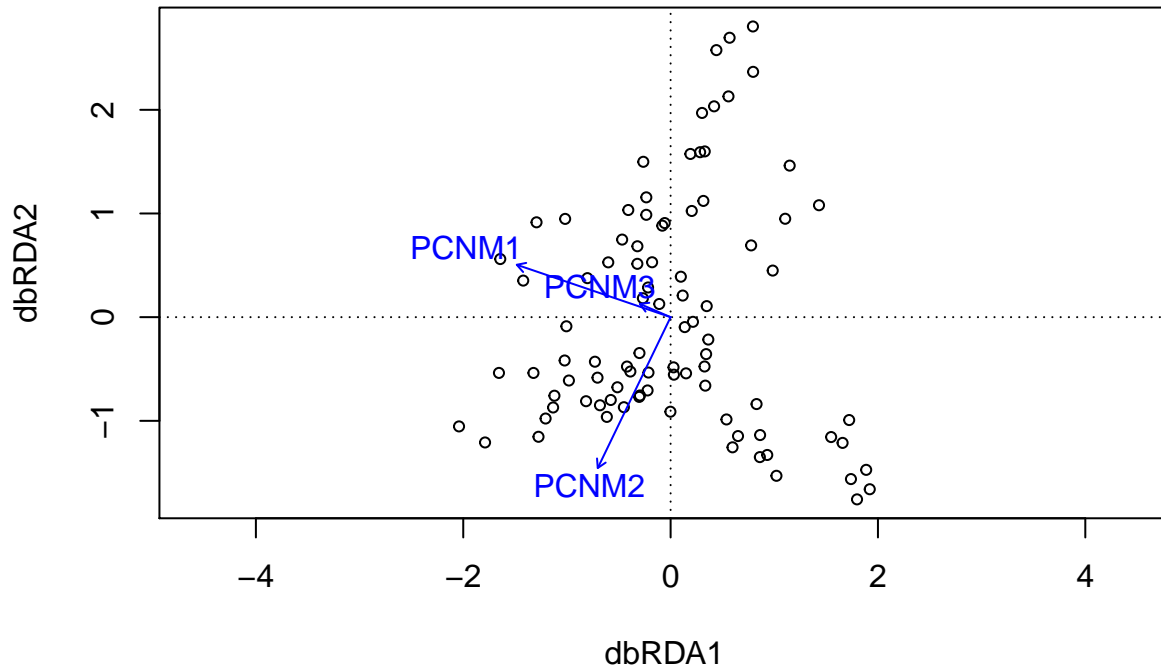
```

```

## <none> 0.07129410
## + PCNM8 0.07109266
## + PCNM24 0.07079342
## + PCNM18 0.07076584
## + PCNM7 0.07062606
## + PCNM17 0.07031123
## + PCNM12 0.06992536
## + PCNM16 0.06923239
## + PCNM11 0.06903507
## + PCNM20 0.06903058
## + PCNM22 0.06808286
## + PCNM21 0.06751725
## + PCNM10 0.06568101
## + PCNM23 0.06567405
## + PCNM26 0.06563057
## + PCNM25 0.06492882
## + PCNM13 0.06488942
## + PCNM27 0.06422087
##
##          Df      AIC      F Pr(>F)
## + PCNM3  1 263.11  2.3371  0.008 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.08535091
## Call: vcs.db ~ PCNM1 + PCNM2 + PCNM3
##
##          R2.adjusted
## + PCNM15 0.09818299
## + PCNM14 0.09765421
## + PCNM5 0.09722787
## <All variables> 0.09627701
## + PCNM4 0.09589324
## + PCNM6 0.09028626
## + PCNM19 0.08976652
## + PCNM9 0.08690710
## + PCNM8 0.08546337
## <none> 0.08535091
## + PCNM24 0.08490813
## + PCNM7 0.08454180
## + PCNM12 0.08445000
## + PCNM17 0.08424619
## + PCNM18 0.08366646
## + PCNM16 0.08340938
## + PCNM20 0.08324390
## + PCNM11 0.08320767
## + PCNM22 0.08225460
## + PCNM21 0.08146340
## + PCNM23 0.07994569
## + PCNM10 0.07980778
## + PCNM26 0.07948376
## + PCNM25 0.07913680
## + PCNM13 0.07906406
## + PCNM27 0.07843556

```

```
plot(step.pcnm)
```



```
step.pcnm$anova
```

```
##           R2.adj Df    AIC      F Pr(>F)
## + PCNM1      0.037290  1 265.79 4.4474 0.002 **
## + PCNM2      0.071294  1 263.53 4.2221 0.004 **
## + PCNM3      0.085351  1 263.11 2.3371 0.008 **
## <All variables> 0.096277
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
space.mod <- model.matrix(~ PCNM1 + PCNM2 + PCNM3, vcs.space)[-1]
# First conduct constrained ordinations
vcs.total.env <- dbrda(vcs.db ~ env.mod)
vcs.total.space <- dbrda(vcs.db ~ space.mod)
# Next construct partial constrained ordinations
vcs.env.cond.space <- dbrda(vcs.db ~ env.mod + Condition(space.mod))
vcs.space.cond.env <- dbrda(vcs.db ~ space.mod + Condition(env.mod))
# Next test for significance of the dbRDA fractions.
permutest(vcs.env.cond.space, permutations = 999)
```

```
##
## Permutation test for dbrda under reduced model
```



```
##
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = vcs.db ~ env.mod + Condition(space.mod))
## Permutation test for all constrained eigenvalues
##      Df Inertia      F Pr(>F)
## Model      4  4.7544 7.822  0.001 ***
## Residual 82 12.4605
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
permutest(vcs.space.cond.env, permutations = 999)
```

```
##
## Permutation test for dbrda under reduced model
##
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = vcs.db ~ space.mod + Condition(env.mod))
## Permutation test for all constrained eigenvalues
##      Df Inertia      F Pr(>F)
## Model      3  1.889 4.1437  0.001 ***
## Residual 82 12.460
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
permutest(vcs.total.env, permutations = 999)
```

```
##
## Permutation test for dbrda under reduced model
##
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = vcs.db ~ env.mod)
## Permutation test for all constrained eigenvalues
##      Df Inertia      F Pr(>F)
## Model      4  5.1284 7.5946  0.001 ***
## Residual 85 14.3495
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
permutest(vcs.total.space, permutations = 999)
```

```
##
## Permutation test for dbrda under reduced model
##
## Permutation: free
## Number of permutations: 999
##
```

```
## Model: dbrda(formula = vcs.db ~ space.mod)
## Permutation test for all constrained eigenvalues
##           Df Inertia      F Pr(>F)
## Model      3   2.263 3.7684 0.001 ***
## Residual 86  17.215
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

#Using the built-in varpart() function

```
vcs.varpart <- varpart(vcs.db, env.mod, space.mod, transfo = "hel")
```

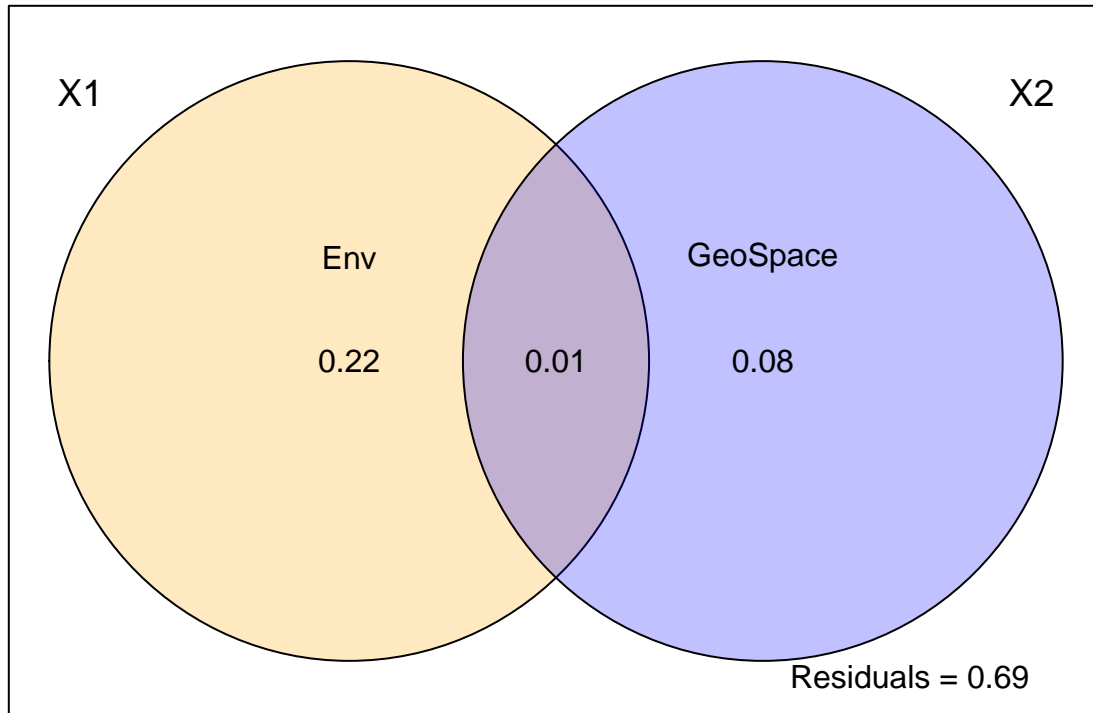
```
## arguments 'transfo' and 'scale' are ignored with distances
```

```
## arguments 'scale' and 'transfo' ignored: valid only in RDA
```

```
vcs.varpart
```

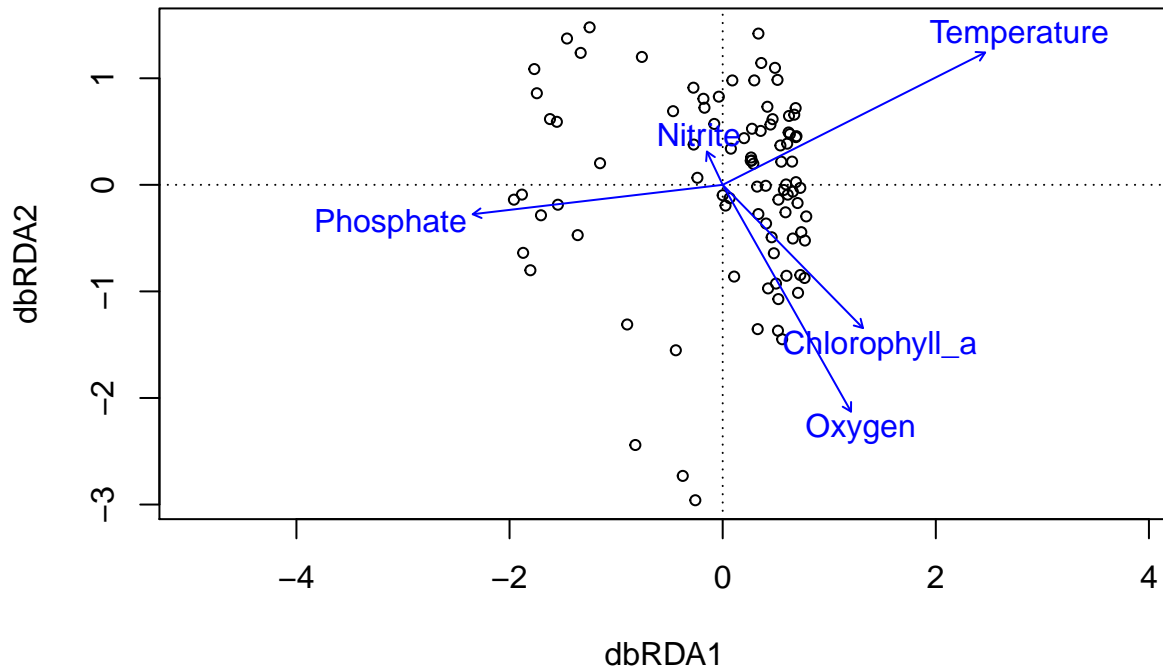
```
##
## Partition of squared Bray distance in dbRDA
##
## Call: varpart(Y = vcs.db, X = env.mod, space.mod, transfo = "hel")
##
## Explanatory tables:
## X1:  env.mod
## X2:  space.mod
##
## No. of explanatory tables: 2
## Total variation (SS): 19.478
## No. of observations: 90
##
## Partition table:
##
##      Df R.squared Adj.R.squared Testable
## [a+b] = X1      4  0.26329      0.22863    TRUE
## [b+c] = X2      3  0.11618      0.08535    TRUE
## [a+b+c] = X1+X2  7  0.36027      0.30566    TRUE
## Individual fractions
## [a] = X1|X2      4              0.22031    TRUE
## [b]              0              0.00831    FALSE
## [c] = X2|X1      3              0.07704    TRUE
## [d] = Residuals              0.69434    FALSE
## ---
## Use function 'dbrda' to test significance of fractions of interest
```

```
par(mar = c(2,2,2,2))
plot(vcs.varpart, bg = c("orange", "blue"))
text(1, 0.25, "GeoSpace")
text(0, 0.25, "Env")
```



Variation partitioning (Three-factor)

```
# env without depth
envpure <- as.matrix(env.data[,4:8])
#perform dbRDA
vcs.dbrda.p <- dbrda(vcs.db ~ ., as.data.frame(envpure))
ordiplot(vcs.dbrda.p)
```

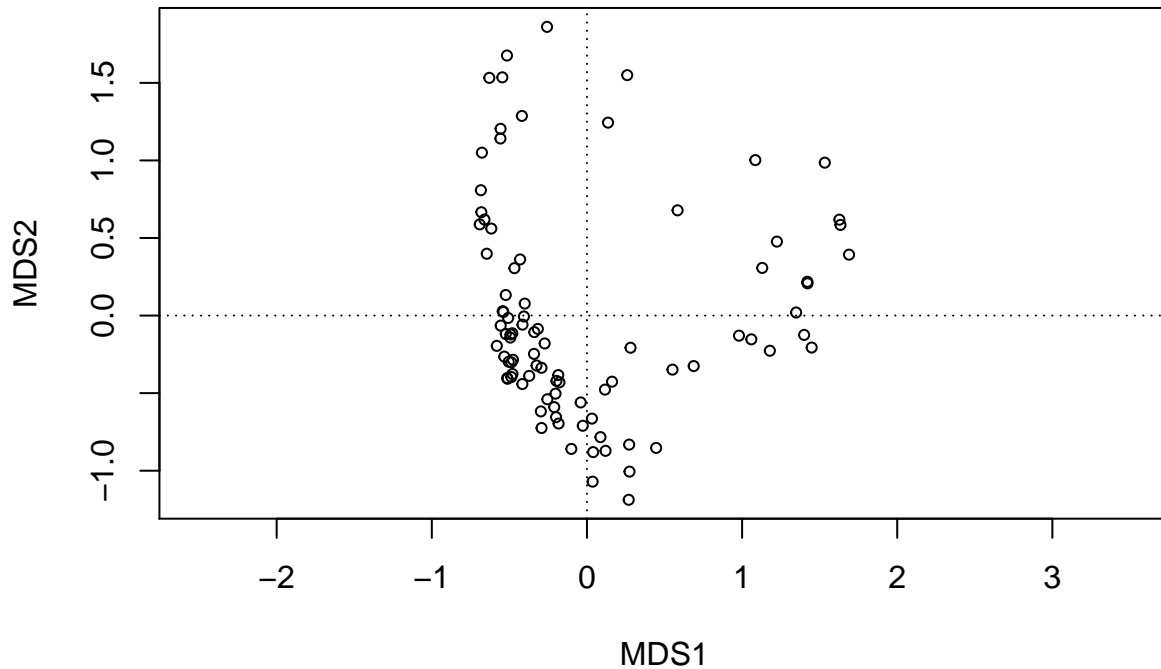


```
print(psych::corr.test(envpure),short = FALSE)
```

```
## Call:psych::corr.test(x = envpure)
## Correlation matrix
##           Temperature Nitrite Phosphate Oxygen Chlorophyll_a
## Temperature          1.00  -0.02   -0.71   0.04           0.06
## Nitrite              -0.02   1.00    0.23  -0.28           0.20
## Phosphate            -0.71   0.23    1.00  -0.55          -0.16
## Oxygen                0.04 -0.28   -0.55   1.00           0.29
## Chlorophyll_a         0.06  0.20   -0.16  0.29           1.00
## Sample Size
## [1] 90
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##           Temperature Nitrite Phosphate Oxygen Chlorophyll_a
## Temperature          0.00   1.00    0.00   1.00           1.00
## Nitrite              0.83   0.00    0.19  0.05           0.32
## Phosphate            0.00   0.03    0.00  0.00           0.51
## Oxygen               0.73   0.01    0.00  0.00           0.04
## Chlorophyll_a        0.58   0.06    0.13  0.01           0.00
##
## Confidence intervals based upon normal theory. To get bootstrapped values, try cor.ci
##           raw.lower raw.r raw.upper raw.p lower.adj upper.adj
## Tmprt-Nitrt      -0.23 -0.02    0.18  0.83    -0.23    0.18
## Tmprt-Phsph      -0.80 -0.71   -0.59  0.00   -0.83   -0.53
## Tmprt-Oxygn      -0.17  0.04    0.24  0.73   -0.20    0.27
```

```
## Tmpprt-Chlr_      -0.15  0.06      0.26  0.58      -0.19      0.31
## Nitrt-Phsph       0.02  0.23      0.41  0.03      -0.05      0.47
## Nitrt-Oxygn      -0.46 -0.28     -0.08  0.01     -0.52      0.00
## Nitrt-Chlr_      -0.01  0.20      0.39  0.06     -0.08      0.44
## Phsph-Oxygn      -0.68 -0.55     -0.39  0.00     -0.72     -0.31
## Phsph-Chlr_      -0.36 -0.16      0.05  0.13     -0.41      0.10
## Oxygn-Chlr_       0.09  0.29      0.47  0.01      0.01      0.53
```

```
# Temperature, Phosphate, Nitrite and Oxygen are significantly correlated.
vcs.dbrda.p.mod0 <- dbrda(vcs.db ~ 1, as.data.frame(envpure))
ordiplot(vcs.dbrda.p.mod0)
```



```
vcs.dbrda.p.mod1 <- dbrda(vcs.db ~ ., as.data.frame(envpure))
vcs.dbrda.p <- ordiR2step(vcs.dbrda.p.mod0, vcs.dbrda.p.mod1, perm.max = 200)
```

```
## Step: R2.adj= 0
## Call: vcs.db ~ 1
##
##           R2.adjusted
## <All variables> 0.200596042
## + Temperature  0.119685498
## + Phosphate     0.106243382
## + Oxygen        0.046884733
## + Chlorophyll_a 0.043447490
## + Nitrite       0.003117142
```

```

## <none>          0.000000000
##
##              Df      AIC      F Pr(>F)
## + Temperature  1 257.74 13.1  0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.1196855
## Call: vcs.db ~ Temperature
##
##              R2.adjusted
## <All variables>  0.2005960
## + Oxygen         0.1653014
## + Chlorophyll_a  0.1590395
## + Phosphate      0.1426166
## + Nitrite        0.1240118
## <none>           0.1196855
##
##              Df      AIC      F Pr(>F)
## + Oxygen  1 253.92 5.8092  0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.1653014
## Call: vcs.db ~ Temperature + Oxygen
##
##              R2.adjusted
## <All variables>  0.2005960
## + Chlorophyll_a  0.1919971
## + Phosphate      0.1747048
## + Nitrite        0.1689095
## <none>           0.1653014
##
##              Df      AIC      F Pr(>F)
## + Chlorophyll_a  1 251.96 3.8744  0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.1919971
## Call: vcs.db ~ Temperature + Oxygen + Chlorophyll_a
##
##              R2.adjusted
## + Phosphate      0.2010239
## <All variables>  0.2005960
## <none>           0.1919971
## + Nitrite        0.1918469

```

```
vcs.dbrda.p$call
```

```

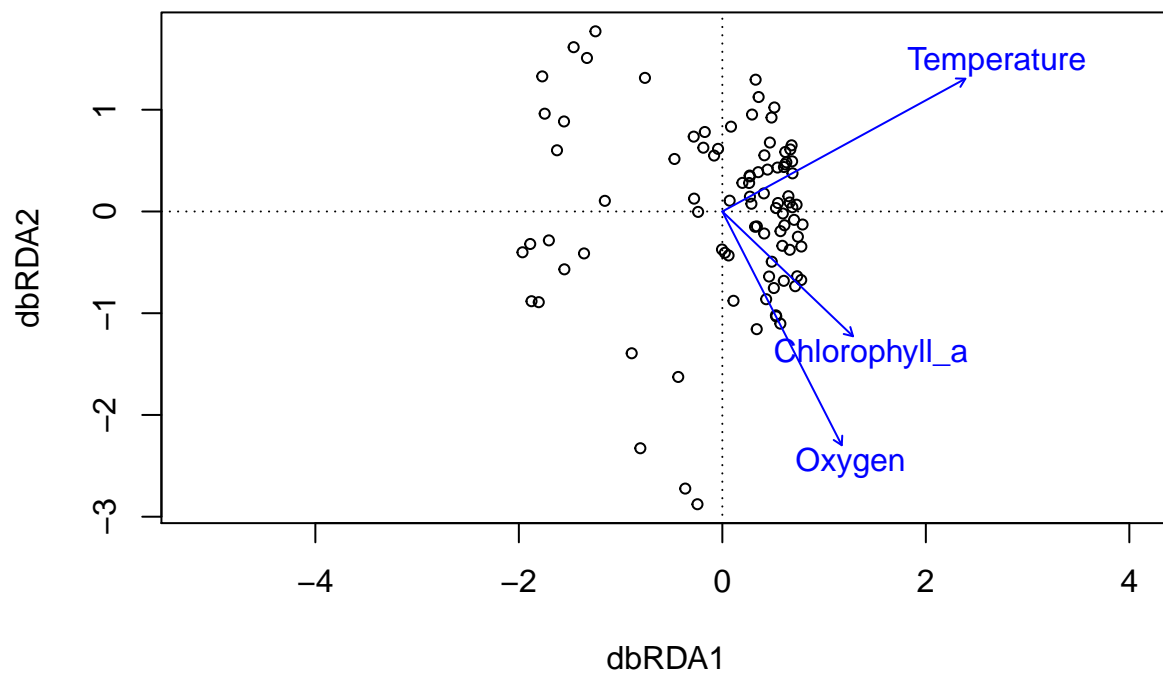
## dbrda(formula = vcs.db ~ Temperature + Oxygen + Chlorophyll_a,
##       data = as.data.frame(envpure))

```

```
vcs.dbrda.p$anova
```

```
##           R2.adj Df      AIC      F Pr(>F)
## + Temperature  0.11968 1 257.74 13.1002  0.002 **
## + Oxygen       0.16530 1 253.92  5.8092  0.002 **
## + Chlorophyll_a 0.19200 1 251.96  3.8744  0.002 **
## <All variables> 0.20060
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ordiplot(vcs.dbrda.p)
```



```
envpure.mod <- model.matrix(~ Temperature + Oxygen + Chlorophyll_a, as.data.frame(envpure))[, -1]
vcs.varpart.3 <- varpart(vcs.db, envpure.mod, space.mod, env.data[3], transfo = "hel")
```

```
## arguments 'transfo' and 'scale' are ignored with distances
```

```
## arguments 'scale' and 'transfo' ignored: valid only in RDA
```

```
vcs.varpart.3
```

```
##
```

```

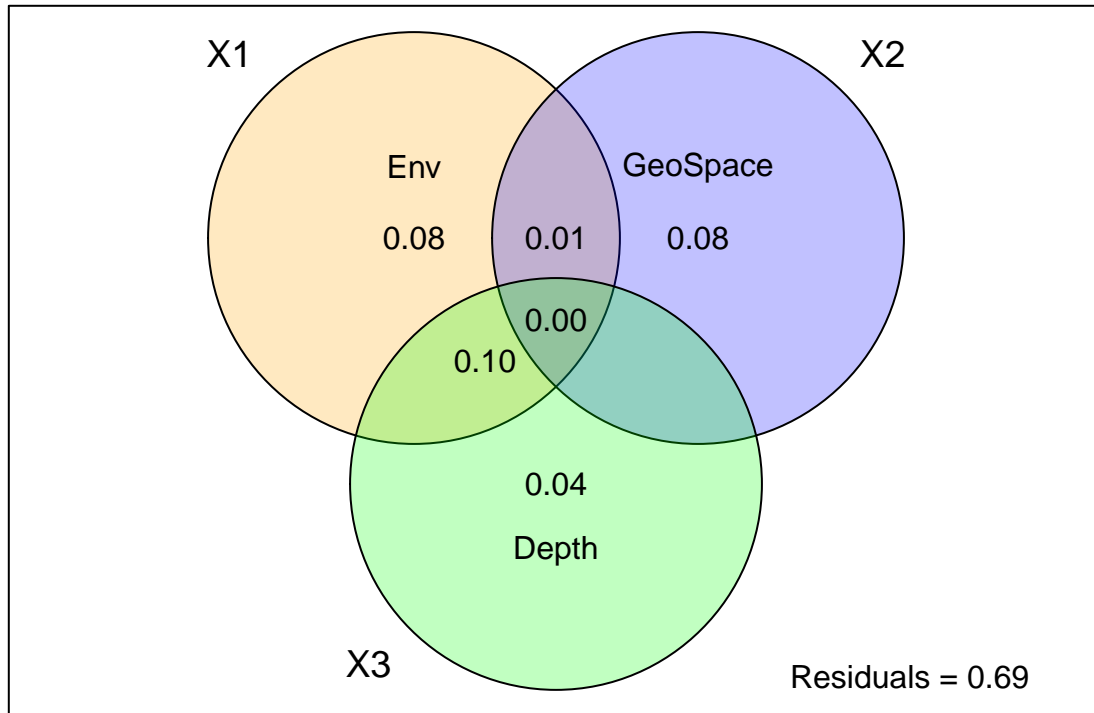
## Partition of squared Bray distance in dbRDA
##
## Call: varpart(Y = vcs.db, X = envpure.mod, space.mod, env.data[3],
## transfo = "hel")
##
## Explanatory tables:
## X1:  envpure.mod
## X2:  space.mod
## X3:  env.data[3]
##
## No. of explanatory tables: 3
## Total variation (SS): 19.478
## No. of observations: 90
##
## Partition table:
##
##      Df R.square Adj.R.square Testable
## [a+d+f+g] = X1      3  0.21923    0.19200    TRUE
## [b+d+e+g] = X2      3  0.11618    0.08535    TRUE
## [c+e+f+g] = X3      1  0.15109    0.14144    TRUE
## [a+b+d+e+f+g] = X1+X2  6  0.31697    0.26760    TRUE
## [a+c+d+e+f+g] = X1+X3  4  0.26329    0.22863    TRUE
## [b+c+d+e+f+g] = X2+X3  4  0.25996    0.22513    TRUE
## [a+b+c+d+e+f+g] = All  7  0.36027    0.30566    TRUE
## Individual fractions
## [a] = X1 | X2+X3      3              0.08053    TRUE
## [b] = X2 | X1+X3      3              0.07704    TRUE
## [c] = X3 | X1+X2      1              0.03806    TRUE
## [d]                  0              0.00665   FALSE
## [e]                  0             -0.00144   FALSE
## [f]                  0              0.10171   FALSE
## [g]                  0              0.00310   FALSE
## [h] = Residuals              0.69434   FALSE
## Controlling 1 table X
## [a+d] = X1 | X3      3              0.08718    TRUE
## [a+f] = X1 | X2      3              0.18225    TRUE
## [b+d] = X2 | X3      3              0.08369    TRUE
## [b+e] = X2 | X1      3              0.07560    TRUE
## [c+e] = X3 | X1      1              0.03663    TRUE
## [c+f] = X3 | X2      1              0.13978    TRUE
## ---
## Use function 'dbrda' to test significance of fractions of interest

```

```

par(mar = c(2,2,2,2))
plot(vcs.varpart.3, bg = c("orange", "blue", "green"))
text(0.5, -1.1, "Depth")
text(1, 0.25, "GeoSpace")
text(0, 0.25, "Env")

```

Values <0 not shown