Diana Lu
Kayla Klaus
Kshitiji Jain
Gabe Palomarez

Q1

Diana's Code

Q1: The data country.csv

Download country.csv

provides information on aggregate personal savings (sr), %of population under age 15 (pop15), %of population over 75(pop75), per-capita disposable income (dpi), and % of growth rate of per-capita disposable income in 50 different countries.

```
install.packages("corrr")

library('corrr')

install.packages("ggcorrplot")

library(ggcorrplot)

install.packages("FactoMineR")

library("FactoMineR")

install.packages("factoextra")#used for scree plots

library("factoextra")

#Import Country data

country <- read.csv("C:/Users/diana/OneDrive - Texas A&M University/Desktop/Data_Analytics/Fall_2024/ANLY608/Assignment/Assignment3/country.csv", header = T)

str(country)
```

Q1a. Conduct a principal component analysis and provide your interpretation of the results.

#data normalization

#removing country as PCA can only be used on numeric data

numerical_country <- country[,2:6]

head(numerical_country)

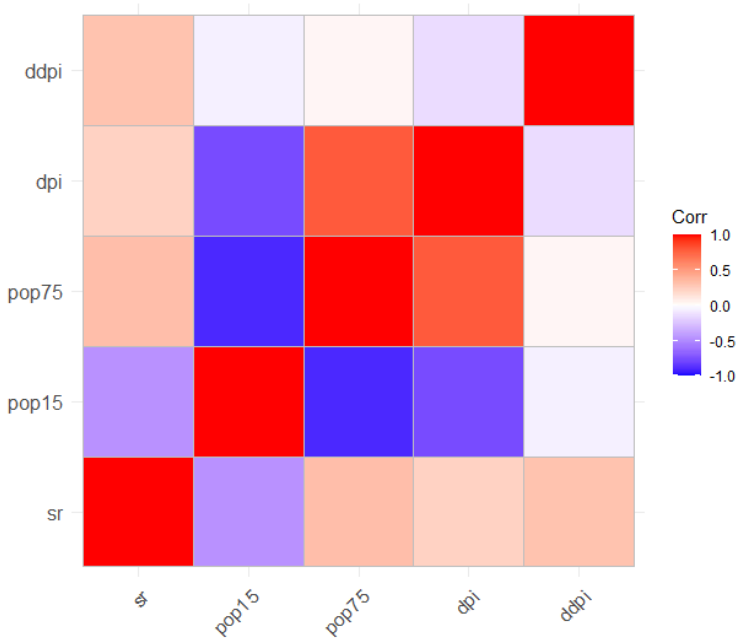
country_normalized <- scale(numerical_country) #normalization

head(country_normalized)

#Correlation matrix

corr_country <- cor(country_normalized)

ggcorrplot(corr_country)



#PCA-Princomp() computes the PCA and summary() function shows results

country.pca <- princomp(corr_country)

summary(country.pca)

```
Importance of components:
                          Comp.1    Comp.2     Comp.3     Comp.4 Comp.5
Standard deviation     1.1729886 0.4936428 0.26886258 0.08828136      0
Proportion of Variance 0.8095132 0.1433712 0.04253017 0.00458537      0
Cumulative Proportion  0.8095132 0.9528845 0.99541463 1.00000000      1
```

#Component 1 explains the majority of the variance (about 81.0%), indicating that it captures the most significant underlying structure of your data.

#Component 2 adds about 14.3% of variance

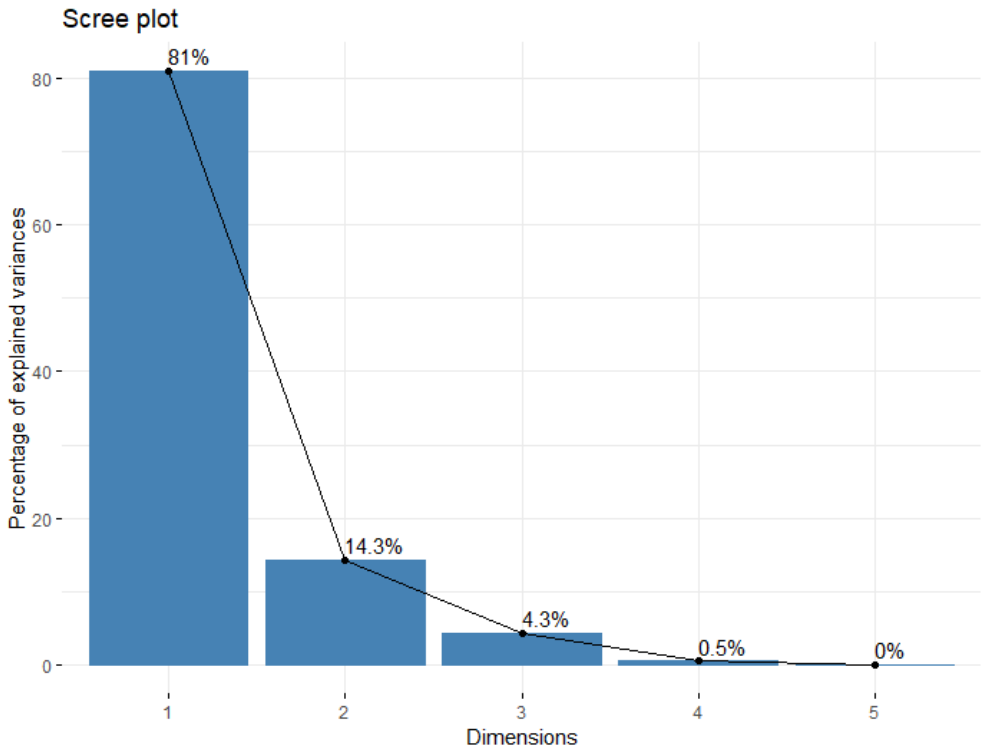
#Component 3 has about 4.3%.

#Component 4 and 5 have percentages less than 1%.

#Based on findings, it would be best to use PC1 and 2.

Q1b. Create a Scree plot and provide your interpretation of the same.

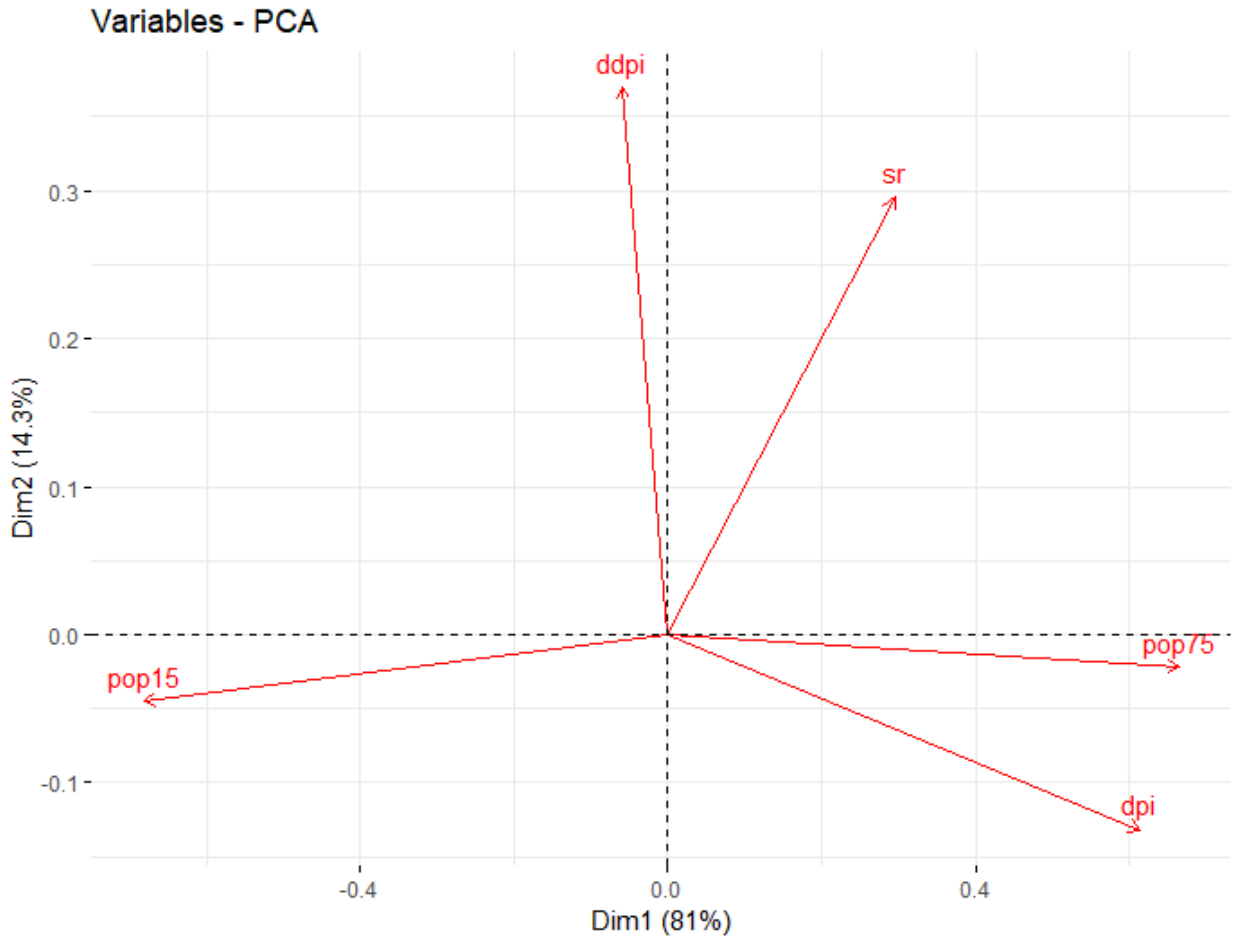
fviz_screeplot(country.pca, addlabels = TRUE)



Scree plot shows Component 1 explains most of the variance while there is a sharp decrease from Component 2 to 5.

Q1c. Create a Biplot of Attributes and provide your interpretation of the visual.

fviz_pca_var(country.pca, col.var = 'red')



Representation of angle degrees between two variables :

- 90 degree, there is no correlation between two variable
- less than 90 degree, there is positive correlation between two variable
- more than 90 degree, there is negative correlation between two variable

Representation of vector line length :

vector line length represent the variance level of variable

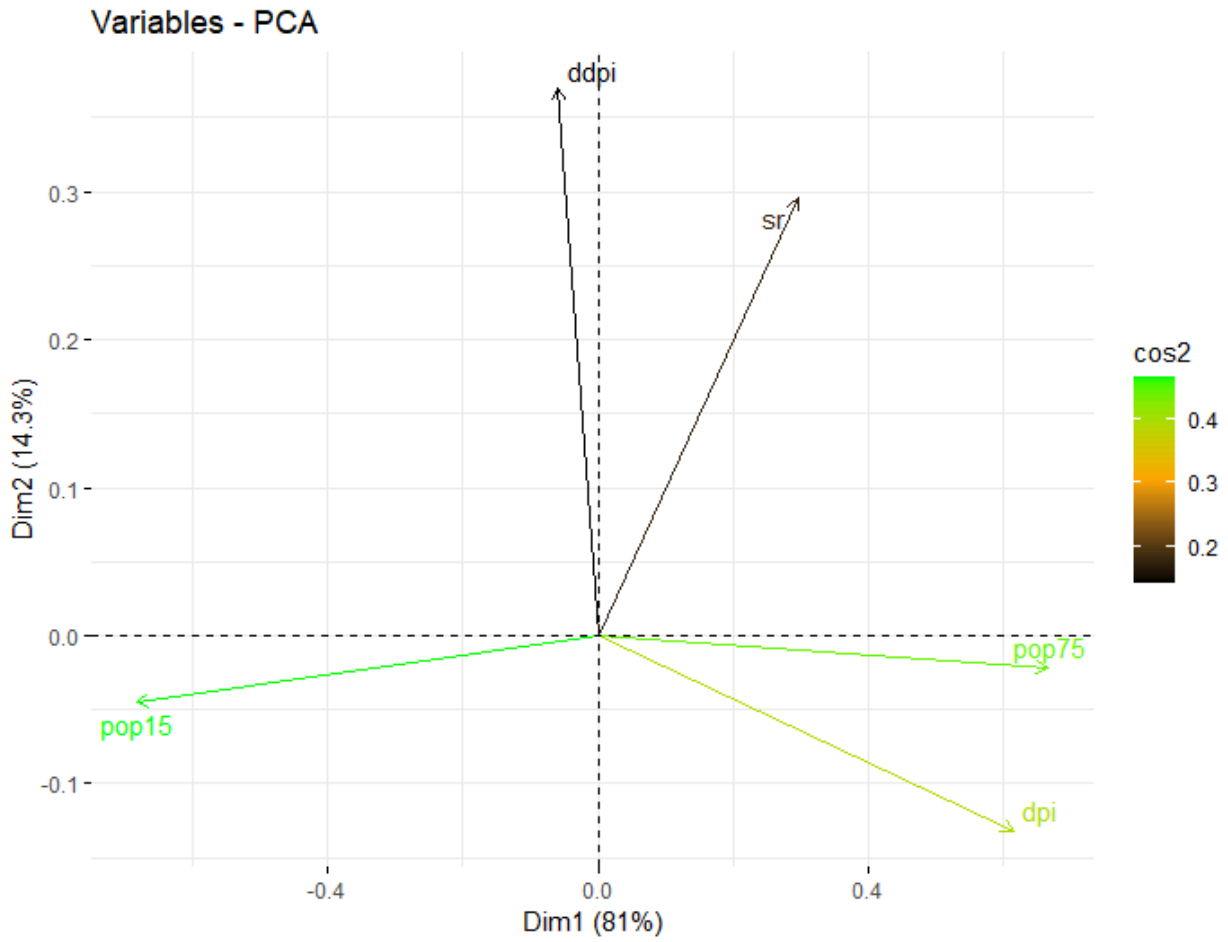
- the longer the vector line, the greater the variance
- the shorter the line, the less the variance

Ddpi and sr have a positive contribution PCA2 while the others have a negative contribution. Ddpi have a stronger variance than sr due to the vector line length. Pop75, sr, and dpi have a positive contribution to PCA1 while pop15 has a strong negative contribution.

Pop75 and dpi have a strong positive correlation between the two. While pop75 and pop15 have a strong negative correlation.

Q1d. How much of each variable is represented in each component? Provide your interpretation.

```
fviz_pca_var(country.pca, col.var = "cos2",
             gradient.cols = c("black", "orange", "green"),
             repel = TRUE)
```



##High cos2 attributes are colored in green: pop15, pop75, dpi.

#Mid cos2 attributes have an orange color: none

#Low cos2 attributes have a black color: ddpi and sr

##A low value means that the variable is not perfectly represented by that component. – ddpi and sr in this case.

#A high value means a good representation of the variable on that component. - pop14, pop75, dpi in this case.

Q1e. Conduct a PCR where DV is pop15 and provide the summary of results, along with your interpretation.

```
install.packages("pls")

library(pls)


#we will fit a principal components regression (PCR) model using hp as the response variable and the

  #following variables as the predictor variables:

#fit model

#make this example reproducible

  set.seed(1)


#fit PCR model

country.model <- pcr(pop15~ddpi+sr+pop75+dpi, data=numerical_country, scale=TRUE, validation="CV")

summary(country.model)
```

```
Data:   X dimension: 50 4
        Y dimension: 50 1
Fit method: svdpc
Number of components considered: 4

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps
CV           9.245    4.159    4.087    4.352    3.978
adjCV        9.245    4.096    4.072    4.321    3.943

TRAINING: % variance explained
       1 comps  2 comps  3 comps  4 comps
X        48.58    79.97    95.07   100.00
pop15    82.22    82.30    82.38    86.26
>
```

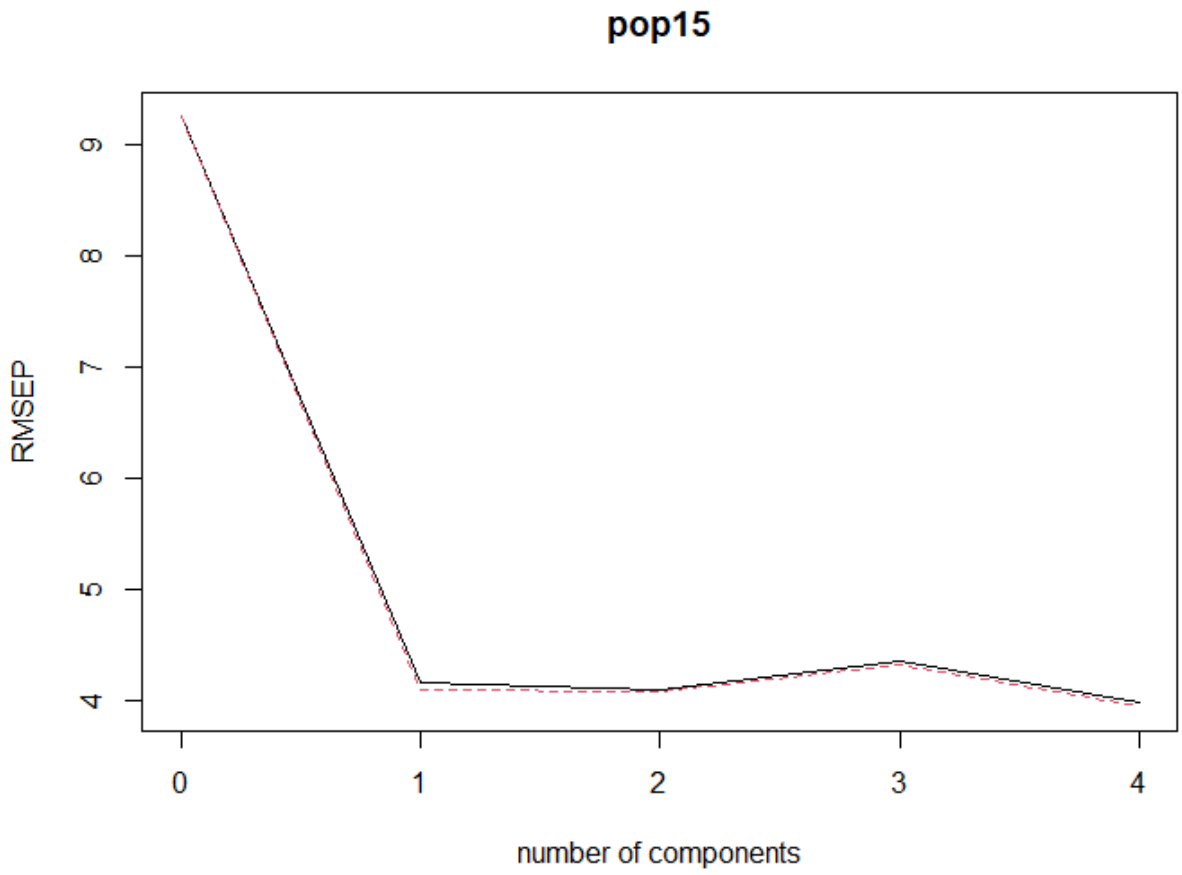
#Interpretation
#Table: Validation RMSEP tells us the test RMSE calculated by the k-fold cross validation.
#We can see the following:
#If we only use the intercept term in the model, the test RMSE is 9.245.
#If we add in the first principal component, the test RMSE drops to 4.096
#If we add in the second principal component, the test RMSE drops to 4.072.
#If we add in the third principal component, the test RMSE increases to 4.321 but
#decreases to 3.943 after adding the fourth principal component.
#DECISION: Four components would be ideal to use.

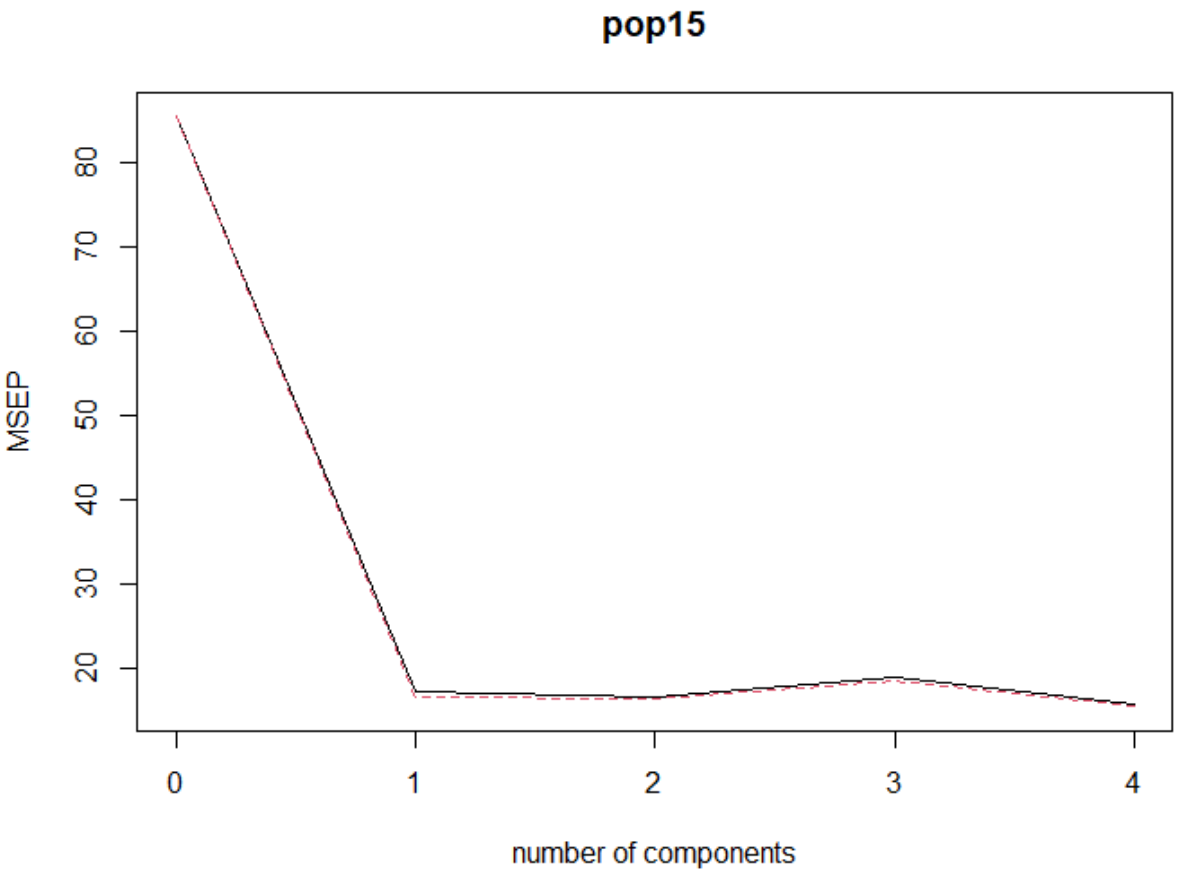
#For pop15, variance explained increases with additional components but reaches only 86.26% with 4 components, indicating that while more components improve fit, there may be diminishing returns or complexity in capturing variance.

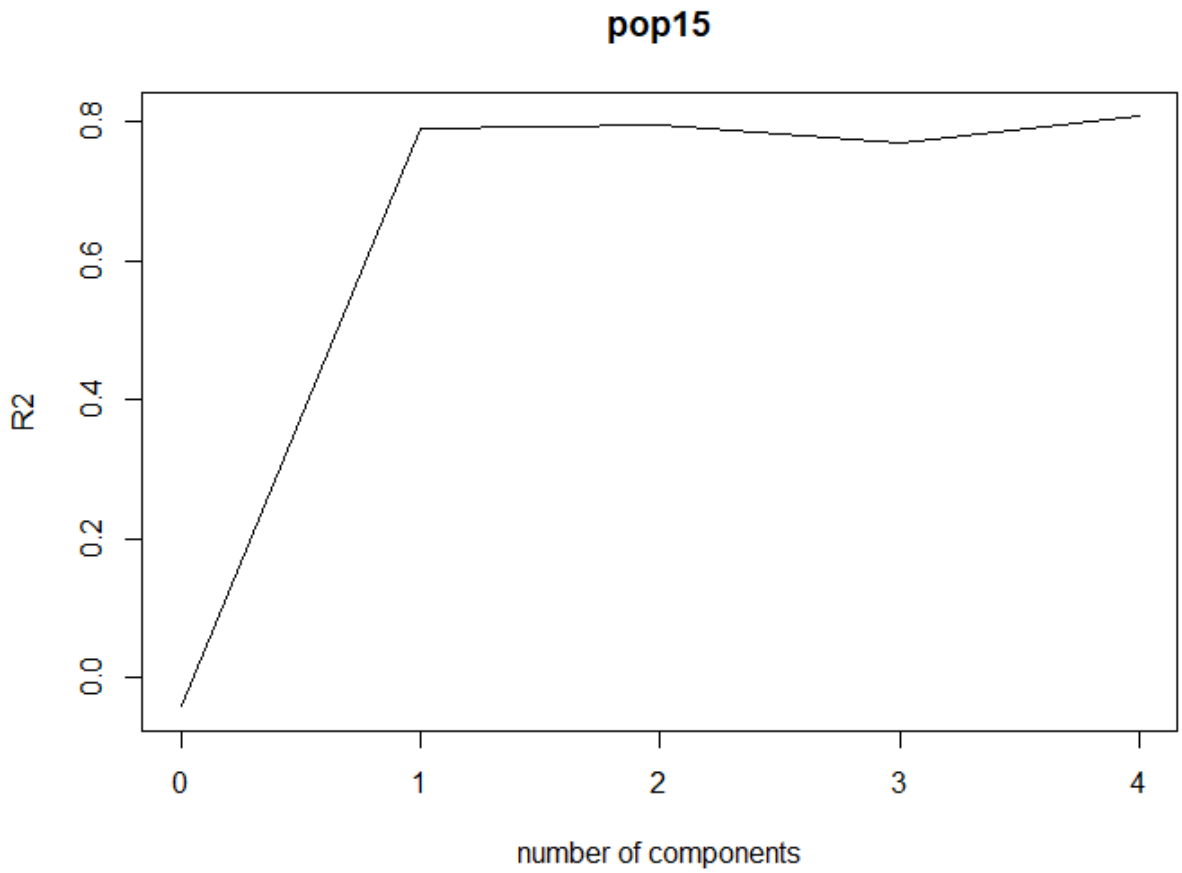
Q1f. Provide visual representation of RMSE, along with your interpretation.

##Visual representation of RMSE
#visualize cross-validation plots

validationplot(country.model)
validationplot(country.model, val.type="MSEP")
validationplot(country.model, val.type="R2")


pop15
RMSEP
9
8
7
6
5
4
0
1
2
3
4
number of components


pop15
MSEP
20
30
40
50
60
70
80
0
1
2
3
4
number of components



Q1g. Find the average deviation between the predicted value for pop15 and the observed value for pop15 for the observations in the testing set. Here, you would need to create two datasets [include first 25 obs. in train data and rest in the test data]. Find this for number of components=3.

```
train <- numerical_country[1:25, c("pop15", "ddpi", "sr", "pop75", "dpi")]
y_test <- numerical_country[26:nrow(numerical_country), c("pop15")]
test <- numerical_country[26:nrow(numerical_country), c("ddpi", "sr", "pop75", "dpi")]
#use model to make predictions on a test set
model <- pcr(pop15~ddpi+sr+pop75+dpi, data=train, scale=TRUE, validation="CV")
pcr_pred <- predict(model, test, ncomp=4)

#Calculate RSME
sqrt(mean((pcr_pred - y_test)^2))
```

```
> #Calculate RSME
> sqrt(mean((pcr_pred - y_test)^2))
[1] 5.385717
```

#Interpretation: The Test RSME for Component 3 is 4.321 and Calculated RSME is 5.386. The lower Test RMSE suggests that the model may be overfitting.

Kayla's Code:
#Question 1

```
#Q1a. Conduct a principal component analysis and provide your interpretation of the results.
install.packages("corrr")
library('corrr')
install.packages("ggcorrplot")
library(ggcorrplot)
install.packages("FactoMineR")
library("FactoMineR")
install.packages("factoextra")
library("factoextra")

country <- read.csv("C:/Users/kayla/OneDrive/Desktop/MSA608_2024/Assignment 3/country.csv",header=T)
str(country)
View(country)

colSums(is.na(country))

numerical_data <- country[,2:6]  #getting read of country as PCA can only be used on numeric data

head(numerical_data)
data_normalized <- scale(numerical_data) #normalization
head(data_normalized)

corr_matrix <- cor(data_normalized)
ggcorrplot(corr_matrix)

data.pca <- princomp(corr_matrix)
summary(data.pca)

data.pca$loadings[, 1:3]

#Q1b
#Create a Scree plot and provide your interpretation of the same.

fviz_screeplot(data.pca, addlabels = TRUE)

#Q1c
```

```
#Create a Biplot of Attributes and provide your interpretation of the visual.

fviz_pca_var(data.pca, col.var = "green")

#Q1d
#How much of each variable is represented in each component? Provide your interpretation.

fviz_cos2(data.pca, choice = "var", axes = 1:2)
fviz_pca_var(data.pca, col.var = "cos2", gradient.cols = c("black", "orange", "green"), repel =
TRUE)

#Q1e
#Conduct a PCR where DV is pop15 and provide the summary of results, along with your
interpretation.

install.packages("pls")
library(pls)

set.seed(1)

model <- pcr(pop15~sr+pop75+dpi+ddpi, data=numerical_data, scale=TRUE, validation="CV")
summary(model)

#Interpretation: The lowest RSMEP values is after adding 4 components. The highest R2 from
the Training: % variance explained is after 4 components are added, at 86.26% of the variation
explained.

#Q1f
#Provide visual representation of RMSE, along with your interpretation.
validationplot(model)
validationplot(model, val.type="MSEP")
validationplot(model, val.type="R2")

#Interpretation: From the first two charts, we can see that MSEP has the lowest value at 4
components, and the last graph shows that R2 has the highest value at 4 components.
#Therefore we should use all 4 components.

#Q1g
#Find the average deviation between the predicted value for pop15 and the observed value for
pop15 for the observations in the testing set.
#Here, you would need to create two datasets [include first 25 obs. in train data and rest in the
test data]. Find this for number of components=3.

train <- numerical_data[1:25, c("pop15", "sr", "pop75", "dpi", "ddpi")]
```

```
y_test <- numerical_data[26:nrow(numerical_data), c("pop15")]
test <- numerical_data[26:nrow(numerical_data), c("pop15", "sr", "pop75", "dpi", "ddpi")]

#use model to make predictions on a test set

model1 <- pcr(pop15~sr+pop75+dpi+ddpi, data=train, scale=TRUE, validation="CV")
pcr_pred <- predict(model1, test, ncomp=3)

#calculate RMSE
sqrt(mean((pcr_pred - y_test)^2))
#Result: 5.385717
```

Q2

Stone Leiker

```
/* Step 1: Import the dataset */
proc import datafile="C:\Users\leiker-s\Desktop\msa608_2024\Assignment 3\psurvey.csv"
    out=psurvey
    dbms=csv
    replace;
    getnames=yes;
run;

/* Q2a: Split the dataset into id=1 (train) and id=2 (test), then drop the id column */
data train (drop=id);
    set psurvey;
    if id=1;
run;

data test (drop=id);
    set psurvey;
    if id=2;
run;

/* Q2a: Create pairwise relationships among v1-v9 using scatter plots */
proc sgscatter data=train;
    matrix v1-v9 / diagonal=(histogram kernel);
    title "Pairwise Scatterplot Matrix of v1 to v9 for id=1 (Train Data)";
run;

/* Q2b: Perform parallel analysis for number of factors and show Scree plot */
proc factor data=train method=principal priors=smc scree nfactors=9;
    title "Scree Plot for Principal Components";
```

```
run;


/* Q2b: Create parallel analysis (in practice, this is commonly done by examining the scree plot
above) */

/* Q2c: Perform factor analysis with n=3 and rotation=oblimin */
proc factor data=train method=principal rotate=oblimin nfactors=3 outstat=fact_out score;
   var v1-v9;
   title "Factor Analysis with Oblimin Rotation (n=3)";
run;

/* Q2c: Create factor scores and calculate correlation among factors */
proc score data=train score=fact_out out=factor_scores type=score;
   var v1-v9;
run;

proc corr data=factor_scores;
   var factor1 factor2 factor3;
   title "Correlation among Factor Scores";
run;

/* Q2d: Interpretation */
/* Provide a manual interpretation based on the factor loadings and factor scores from the
output.
  This step cannot be automated within the code itself, but you can use the printed results
  from the factor analysis and correlation to interpret the relationships between variables. */
```

Kayla's Code:
#Question 2

```
ps <- read.csv("C:/Users/kayla/OneDrive/Desktop/MSA608_2024/Assignment
3/psurvey.csv",header=TRUE)
summary(ps)
View(ps)

install.packages ("corrplot")
install.packages ("psych")
install.packages("ggplot2")
install.packages("car")
library(corrplot)
library(psych)
library(ggplot2)
library(car)
```

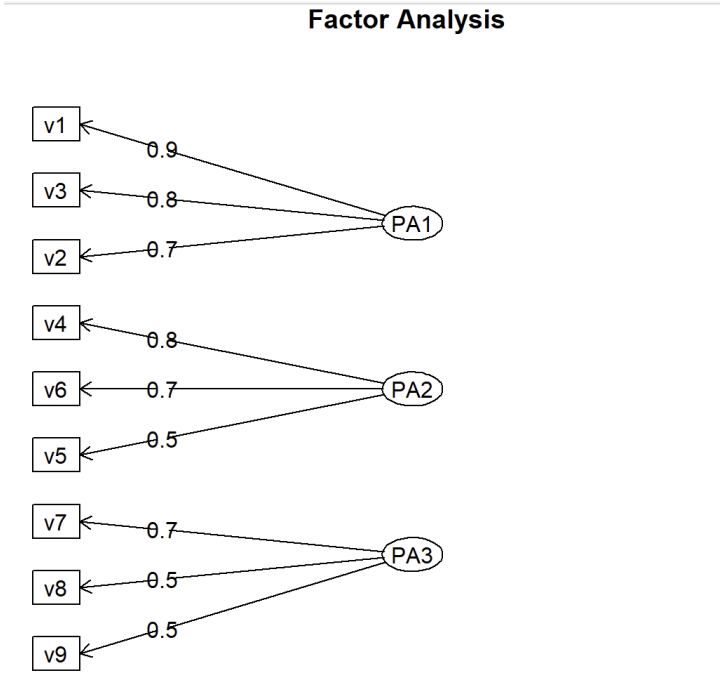
```
#Q2a
#Split the dataset into two groups, one for id=1, and other for id=2. Now drop the column "id". Treat the data for id=1 as train data.
ps1 <- ps[c(1:100),]
ps11 <- ps1[2:10]

ps2 <- ps[c(101:200),]
ps22 <- ps2[2:10]

#Now create a visual that shows pairwise relationships among v1-v9.
datamatrix1 <- cor(ps11)
corrplot(datamatrix1, method="number")

datamatrix2 <- cor(ps22)
corrplot(datamatrix2, method="number")

#Q2b
#Create a parallel analysis for number of factors and show the analysis along with scree plot in one visual.

parallel <- fa.parallel(ps11)

#Scree Plot
fafitfree <- fa(ps11, nfactors = ncol(ps11), rotate = "none")

n_factors <- length(fafitfree$e.values)

scree <- data.frame(
  Factor_n =  as.factor(1:n_factors),
  Eigenvalue = fafitfree$e.values)
ggplot(scree, aes(x = Factor_n, y = Eigenvalue, group = 1)) +
  geom_point() + geom_line() + xlab("Number of factors") +
  ylab("Initial eigenvalue") +
  labs( title = "Scree Plot",
        subtitle = "(Based on the unreduced correlation matrix)")

#Q2c
#Create a factor analysis (n=3) using rotate=oblimin. #Now create the factor scores. Now show the correlation among the factors using a visual.

fa.none <- fa(r=ps11, nfactors = 3, covar = FALSE, SMC = TRUE, fm="pa",  max.iter=100, rotate="oblimin")
print(fa.none)
```

#Scores:
print(fa.none$scores)

#Visual:
fa.diagram(fa.none)

#Q2d
#Provide your interpretation of the factor scores and loadings.

```
Standardized loadings (pattern matrix) based upon correlation matrix
     PA1   PA2   PA3   h2   u2 com
v1  0.90 -0.04  0.06 0.79 0.21 1.0
v2  0.74  0.05  0.00 0.56 0.44 1.0
v3  0.80  0.02 -0.10 0.66 0.34 1.0
v4  0.00  0.78  0.04 0.62 0.38 1.0
v5  0.09  0.54  0.15 0.38 0.62 1.2
v6 -0.03  0.70 -0.10 0.46 0.54 1.0
v7 -0.03  0.01  0.73 0.54 0.46 1.0
v8  0.02  0.04  0.51 0.27 0.73 1.0
v9  0.03 -0.03  0.45 0.20 0.80 1.0
```



Factor 1 has a strong influence on the variables V1, V3, and V2, with loadings of 0.9, 0.8, and 0.7. This means that factor 1 is able to explain the variance in these variables by that measure. The remaining factors follow suit with this as well, in having a strong relationship to the variables and are able to explain the variance in the variables. Factor 2 has loading values of 0.8, 0.7, and 0.5 on variables V4, V6, and V5. Factor 3 has loading values of 0.7, 0.5, and 0.5 on variables V7, V8, and V9.

```
> print(fa.none$scores)
              PA1         PA2         PA3
1  -0.783420678 -1.15067371  0.59837826
2  -0.262502699  2.10048854  1.82115169
3   0.433115912 -1.72262085  0.39988624
```

The factor scores are values for each observation in the dataset that represent how strongly each observation is associated with an underlying factor. For example, row or observation 1 has a -0.78 value of factor 1, a -1.15 value of factor 2, and a 0.6 value of factor 3.

Q3

Kayla's Code:

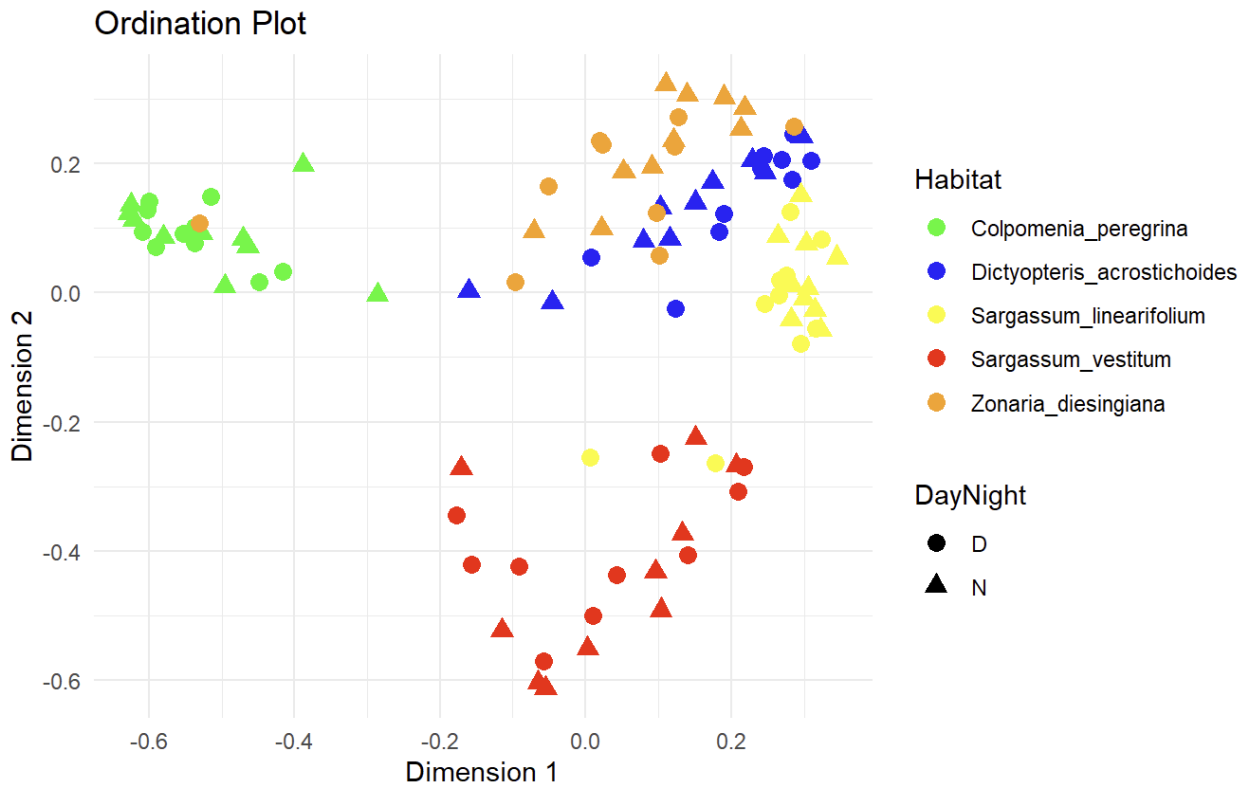
```
#Q3a
#Create a data frame using the data (column 5-column 11).

species <- read.csv("C:/Users/kayla/OneDrive/Desktop/MSA608_2024/Assignment 3/species.csv",header=TRUE)
summary(species)
View(species)

species1 <- species[,5:11]
View(species1)

#Q3b
#Create a dissimilarity matrix with and distance= “bray” [bray= Bray-Curtis distance measure] that measures the similarity between every pair of samples.

species_mds <- cmdscale(dist(species1))

#Q3c
#Create an ordination plot from that dissimilarity matrix. Please plot samples from each habitat type as a different color. Also, add habitat and day vs night factors to the visual.

plot(species_mds[, 1], species_mds[, 2],
    type = "n", xlab = "MDS Dimension 1",
    ylab = "MDS Dimension 2")

points(species_mds[, 1], species_mds[, 2],
      pch = 21, bg = "lightblue")
text(species_mds[, 1], species_mds[, 2],
    pos = 3, cex = 0.8)
```

DIANA's CODE

```
#Q3
```

```
#Q3: The dataset species.csv Download species.csvrepresents habitat of various species.
#Column 1: specifies the habitat (categorical),
#column 2: informs when the specimen was collected (day/night),
#Column 2: replicate number per combination of habitat and day/night,
#Column 4: biomass of the habitat sampled.
#The rest of the columns represent the counts of each herbivore species in each sample.

#Import data
species <- read.csv("C:/Users/diana/OneDrive - Texas A&M
University/Desktop/Data_Analytics/Fall_2024/ANLY608/Assignment/Assignment3/species.csv",
header = T)

#Q3a. Create a data frame using the data (column 5-column 11).
df_species <-data.frame(species[, 5:11])
df_species

#Q3b. Create a dissimilarity matrix with and distance= “bray” [bray= Bray-Curtis distance
measure]
  #that measures the similarity between every pair of samples.
install.packages("vegan")
library(vegan)

species_bray_curtis <- vegdist(df_species, method = 'bray')
#This dissimilarity measure ranges between 0 and 1, where 0 indicates complete similarity
  #(i.e., the two samples have exactly the same species composition),
  #and 1 indicates complete dissimilarity (i.e., the two samples share no species in common).

print(species_bray_curtis)

mds_species <- cmdscale(species_bray_curtis, k = 2)
mds_df<- as.data.frame(mds_species)
colnames(mds_df) <- c("Dim1", "Dim2")

print(mds_df)

#Q3c. Create an ordination plot from that dissimilarity matrix.
  #Please plot samples from each habitat type as a different color.
  #Also, add habitat and day vs night factors to the visual.

#adding habitat and DayNight columnns

mds_df$Habitat <- species$Habitat
mds_df$DayNight <- species$DayNight
```

```
print(mds_df)

ggplot(mds_df, aes(x = Dim1, y = Dim2, color = Habitat, shape = DayNight)) +
  geom_point(size = 3) +
  labs(title = "Ordination Plot", x = "Dimension 1", y = "Dimension 2") +
  theme_minimal() +
  scale_color_manual(values = c("Colpomenia_peregrina" = "green",
"Dictyopteris_acrostichoides" = "blue", "Sargassum_linearifolium" = "yellow",
"Sargassum_vestitum" = 'red', 'Zonaria_diesingiana' = 'orange')) +
  scale_shape_manual(values = c("D" = 16, "N" = 17)) +
  theme(legend.position = "right")
```



KSHITIJ'S CODE

Q3

#Load the dataset

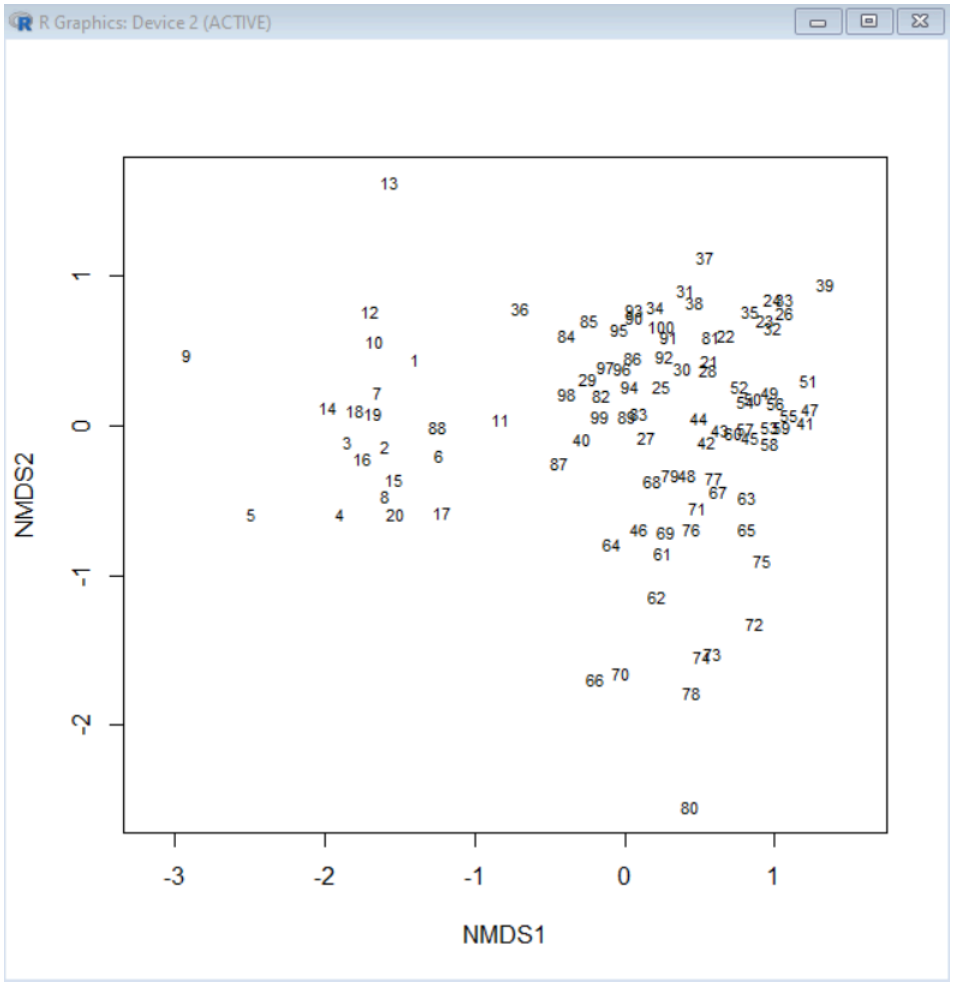
```
install.packages("vegan")
# Load necessary libraries
library(tidyverse)
library(vegan) # For ecological analysis and ordination methods
# Load the data
species_data <- read.csv("C:/Users/jaink/Downloads/Class Work/ANLY608/species.csv")
```

```
# View the data structure
head(species_data)
```
Q3A)
```
# Subset the data frame to include only columns 5 to 11
herbivore_data <- species_data[, 5:11]

# View the subsetted data
head(herbivore_data)
```

Q3B)
```
# Calculate Bray-Curtis dissimilarity matrix
dissimilarity_matrix <- vegdist(herbivore_data, method = "bray")

# View the dissimilarity matrix
dissimilarity_matrix
```

Q3C)
```
# Run NMDS using Bray-Curtis dissimilarity
nmds <- metaMDS(dissimilarity_matrix, distance = "bray", trymax = 100)

# Extract habitat and day/night columns for plotting
habitat <- species_data$Column1  # Replace with actual column name
day_night <- species_data$Column2  # Replace with actual column name

# Create the ordination plot
plot(nmds, type = "n")  # 'n' for no points initially

# Add points colored by habitat
points(nmds, display = "sites", col = as.factor(habitat), pch = 16)

# Add labels for habitat and day/night
ordiplot(nmds, type = "t", display = "sites")
with(species_data, ordiellipse(nmds, habitat, col = as.factor(habitat)))
with(species_data, ordihull(nmds, as.factor(day_night), draw = "polygon", col = "blue"))

# Add legend
legend("topright", legend = unique(habitat), col = unique(as.factor(habitat)), pch = 16)
```

Output


R Graphics: Device 2 (ACTIVE)
NMDS1
NMDS2
-3
-2
-1
0
1
-2
-1
0
1