Sample R code

Shuangshuang Liu

7/8/2019

This document contains sample R codes from typical analyses I have done for my past research projects. Specifically: (1) Analysis of plant traits (continuous and binomial traits using generalized linear mixed models) from a field trial. (2) Clustering analysis (unsupervised learning) on seed germination traits of 15 desert annual plant species. (3) Visualization of germination niche (Simulated seed germination percentages within a range of environmental conditions) (4) Mapping of introduction routes of invasive plant species worldwide

Load required libraries

```r
set.seed(1234)
# For data preparation
library(tidyr)
library(plyr)
library(stringr)
library(tidyverse)

# For visualization
library(ggplot2)
library(cowplot)
library(gplots)
library(gridExtra)
library(lattice) # for wireframe surface plot
library(maps)
library(mapplots)
library(scales)

# For analyses
library(lme4) # Mixed models
library(car) # Levene test
library(cluster) #clustering algorithms
library(factoextra) #clustering algorithms and visualization
library(flexclust) #weighted kmeans clustering
library(akima) #for interpolation of spaced data
library(network) # for constructing network objects
library(sna) # for social network analysis
```

1. Analysis of plant traits from a field trial

Experimental design

- This experiment followed a split-plot design with two irrigation levels as main plot treatments.
- Each irrigation treatment was applied to five completely randomized blocks containing the subplot treatments.
- Within each block, 72 rows of seeds from 12 populations (6 accessions per population) were sown in a randomized order. Each row contains 10 seeds.

- Therefore, a total of 2x5x72x10=7200 seeds were planted.
- For each survived individual plant, total biomass and seed biomass were measured (continuous traits)
- For each row of 10 plants, percentage of flowered plants,germination and survival were measured (binary traits).

- Dataframe `biomass` has been loaded, which includes log-transformed biomass data

```
head(biomass)
```

```
##   Water Block Pop LineID    whole      seed
## 1     N     2   1      1 2.009332 0.4304352
## 2     N     2   1      2 1.907108 0.6881939
## 3     N     2   1      3 2.145510 1.0495148
## 4     N     2   1      4 1.606594 0.2974002
## 5     N     2   1      5 2.082047 0.9549095
## 6     N     2   1      6 1.556021 0.2715046
```

```
pop.mean <- ddply(biomass,.(Water,Block,Pop),numcolwise(mean, na.rm=TRUE)) # Average accessions per population per block, resulting in 2x5x12=120 data points for both traits
```

ANOVA analysis of continuous traits (whole plant and seed biomass)

- Test anova assumptions

```
# Normalily
shapiro.test((pop.mean$whole)) # p<0.05
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  (pop.mean$whole)
## W = 0.97654, p-value = 0.03403
```

```
shapiro.test((pop.mean$seed)) # p<0.05
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  (pop.mean$seed)
## W = 0.97665, p-value = 0.03483
```

Results of Shapiro-willk test were significant. However, the large sample size(n>30) and high W value (>0.97) which suggested nearly normal distribution of data. We can also visualize the data distribution in Q-Q plot to confirm normality.

```
# qqnorm(pop.mean$whole)
# qqnorm(pop.mean$seed)
```

```
# Homogeneity of variance
leveneTest(whole~Pop,data=pop.mean) # Not significant (NS)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group  11  0.6357 0.7947
##       108
```

```
leveneTest(seed~Pop,data=pop.mean)# NS
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group  11  1.2456 0.2664
##       108
```

- **Full ANOVA model**

```
# Total biomass
fit.whole<-aov(whole~Pop*Water+Error(Block),data=pop.mean)
# Seed biomass
fit.seed<-aov(seed~Pop*Water+Error(Block),data=pop.mean)
summary(fit.seed)
```

```
##
## Error: Block
##           Df Sum Sq Mean Sq F value  Pr(>F)
## Water      1 10.207  10.207   37.58 0.00028 ***
## Residuals  8  2.173   0.272
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: Within
##           Df Sum Sq Mean Sq F value  Pr(>F)
## Pop       11  6.970  0.6336  18.044 < 2e-16 ***
## Pop:Water 11  0.959  0.0872   2.484 0.00921 **
## Residuals 88  3.090  0.0351
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Both total biomass and seed biomass show significant irrigation effects, population effects, and population-by-water interactions

- **Within-treatment ANOVA**

```
Y<-pop.mean[pop.mean[,"Water"]=="Y",] # Data--with irrigation
N<-pop.mean[pop.mean[,"Water"]=="N",] # Data--without irrigation

fit.whole.Y <- aov(whole~Pop+Error(Block),data=Y) # p< 0.05
fit.seed.Y <- aov(seed~Pop+Error(Block),data=Y) #p <0.05

fit.whole.N <- aov(whole~Pop+Error(Block),data=N) # p< 0.05
fit.seed.N <- aov(seed~Pop+Error(Block),data=N) # p< 0.05
```

For both with and without irrigation treatments, both total biomass and seed biomass were significantly affected by the origin of populations

Analysis of binary trait data

Mixed effects logistic regression models on binary traits –take flowering as an example

- Dataframe `BinaryTraits` has been loaded, which includes binary trait data.

```
head(BinaryTraits)
```

```
##   Water Block Pop LineID Flowered Survived DecGerm Planted
## 1     Y     1   1      1        0        8       2      10
## 2     Y     1   1      2        3        6       4      10
## 3     Y     1   1      3        6        7       6      10
## 4     Y     1   1      4        4       10       9      10
## 5     Y     1   1      5        4        6       5      10
## 6     Y     1   1      6        0        8       6      10
```

```
flower<-cbind(BinaryTraits$Flowered,BinaryTraits$Survived-BinaryTraits$Flowered)
head(flower)
```

```
##      [,1] [,2]
## [1,]    0    8
## [2,]    3    3
## [3,]    6    1
## [4,]    4    6
## [5,]    4    2
## [6,]    0    8
```

```
fit.flower <- glmer(flower~Water+Pop+Water:Pop+(1|Block),data=BinaryTraits,family=binomi
al,,control=glmerControl(optimizer="bobyqa"),nAGQ=10)
# Second model, removing interaction term
fit.flower2<-glmer(flower~Water+Pop+(1|Block),data=BinaryTraits,family=binomial,control=
glmerControl(optimizer="bobyqa"),nAGQ=10)
# Third model, removing water effects
fit.flower3<-glmer(flower~Pop+(1|Block),data=BinaryTraits,family=binomial,control=glmerC
ontrol(optimizer="bobyqa"),nAGQ=10)

anova(fit.flower,fit.flower2,test="LRT")
```

```
## Data: BinaryTraits
## Models:
## fit.flower2: flower ~ Water + Pop + (1 | Block)
## fit.flower: flower ~ Water + Pop + Water:Pop + (1 | Block)
##             Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## fit.flower2  4 3589.9 3608.2 -1791.0   3581.9
## fit.flower   5 3591.8 3614.7 -1790.9   3581.8 0.1347      1     0.7136
```

Insignificant results. Interaction terms in the first model can be removed

```
anova(fit.flower2,fit.flower3,test="LRT")
```

```
## Data: BinaryTraits
## Models:
## fit.flower3: flower ~ Pop + (1 | Block)
## fit.flower2: flower ~ Water + Pop + (1 | Block)
##             Df    AIC    BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## fit.flower3  3 3588.0 3601.7  -1791   3582.0
## fit.flower2  4 3589.9 3608.2  -1791   3581.9 0.0595      1     0.8073
```

Insignificant results. Water effects in the second model can be removed. The third model, fit.flower3 was retained as the final model

```
summary(fit.flower3)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 10) [glmerMod]
##  Family: binomial  ( logit )
## Formula: flower ~ Pop + (1 | Block)
##    Data: BinaryTraits
## Control: glmerControl(optimizer = "bobyqa")
## 
##      AIC      BIC   logLik deviance df.resid 
##   3588.0   3601.7  -1791.0   3582.0      717 
## 
## Scaled residuals: 
##     Min      1Q  Median      3Q     Max 
## -5.8021 -1.5201  0.4016  1.7241  3.8773 
## 
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Block  (Intercept) 0.1275   0.3571  
## Number of obs: 720, groups:  Block, 10
## 
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) 0.144834   0.125385   1.155    0.248    
## Pop         0.040939   0.004565   8.969   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Correlation of Fixed Effects:
##     (Intr)
## Pop -0.375
```

Visualize population-by-irrigation interactions

- Dataframe `biomass_raw` has been loaded, which contains the average biomass data (not log-transformed) for each population-by-irrigation treatment.

```
head(biomass_raw)
```

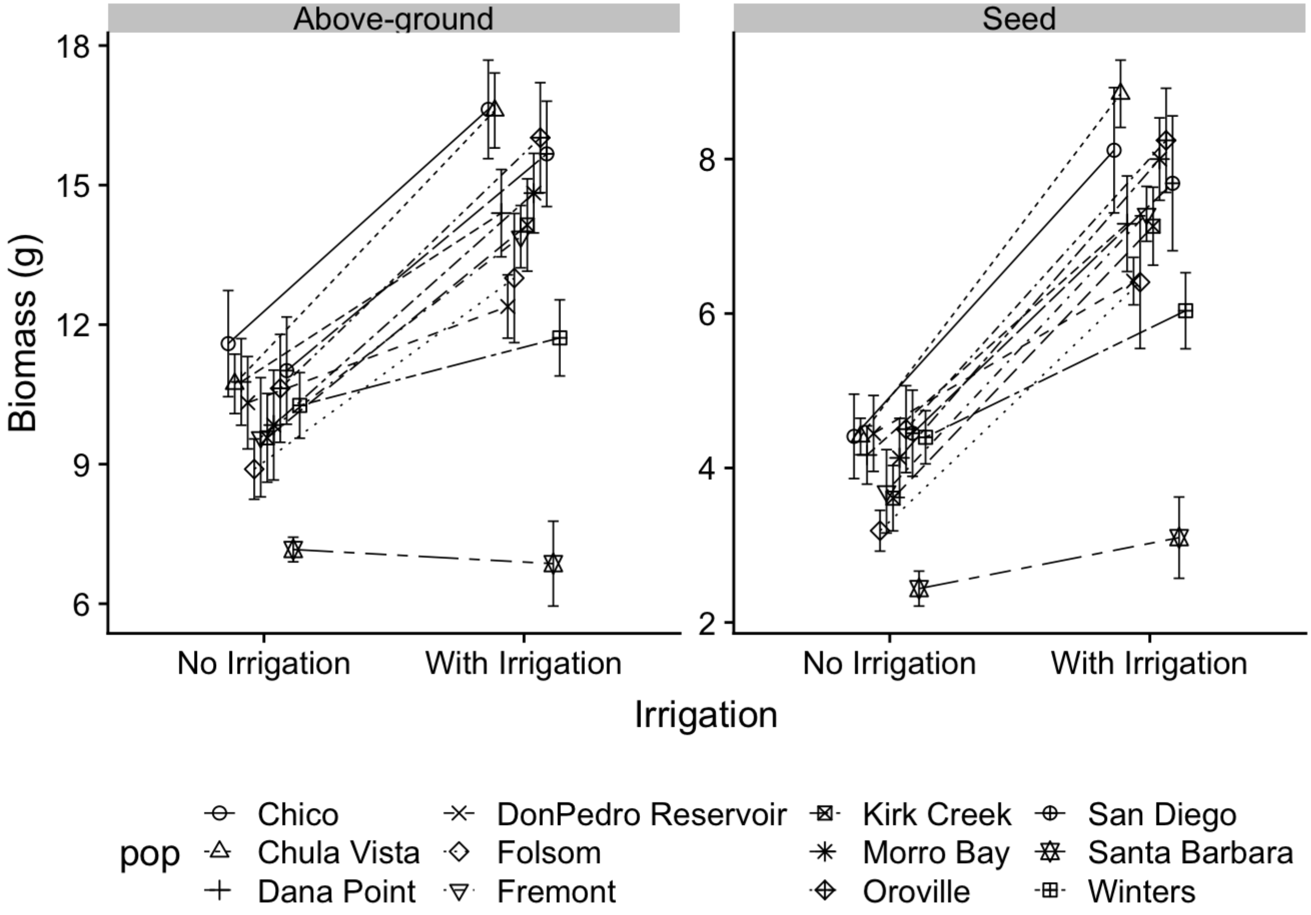
```
##   X Water Pop N      mean       sd        se          var    Irrigation
## 1 1     N   1 5  9.578597 2.866227 1.2818156 Above-ground No Irrigation
## 2 2     N   3 5  9.566220 2.130682 0.9528698 Above-ground No Irrigation
## 3 3     N   5 5  9.840923 2.641525 1.1813257 Above-ground No Irrigation
## 4 4     N   6 5  7.165202 0.592381 0.2649208 Above-ground No Irrigation
## 5 5     N   9 5 10.768596 2.080840 0.9305799 Above-ground No Irrigation
## 6 6     N  10 5 11.011915 2.578492 1.1531368 Above-ground No Irrigation
##             pop
## 1       Fremont
## 2    Kirk Creek
## 3     Morro Bay
## 4 Santa Barbara
## 5    Dana Point
## 6     San Diego
```

```
dim(biomass_raw)
```

```
## [1] 48 10
```

```
# Create interaction plot using ggplot
pd<-position_dodge(width=0.3)
bio<-ggplot(data=biomass_raw, aes(x=Irrigation, y=mean,group=pop,linetype=pop,shape=pop))+
  geom_errorbar(aes(ymin=mean-se,ymax=mean+se),linetype=1,width=0.5,size=0.3,position=pd)+
  geom_line(position=pd,size=0.3)+
  geom_point(position=pd,size=2)+
  scale_shape_manual(values=1:nlevels(biomass_raw$pop))+
  facet_wrap(~var,scales="free_y",ncol=2)+
  ylab("Biomass (g)")

bio+theme(legend.position="bottom")
```



Conclusions: (1) Within each irrigation treatment, origin of populations significantly affect plant biomass trait; Populations also respond to irrigation treatments differently (significant pop-by-water interactions). (2) Population effects were also significant for the binary trait, flowering. (3) We can thus continue to look into performance of specific populations and how other factors (such as environmental variables at origin locations) may play a role. (3) In agriculture, such trait differences among wild populations are potentially important for germplasm selection.

2. Clustering Analysis on germination traits of a group of annual species

- Dataframe `field_all` has been loaded, which includes modeled germination parameters for 15 annual plant species.

```
head(field_all)
```

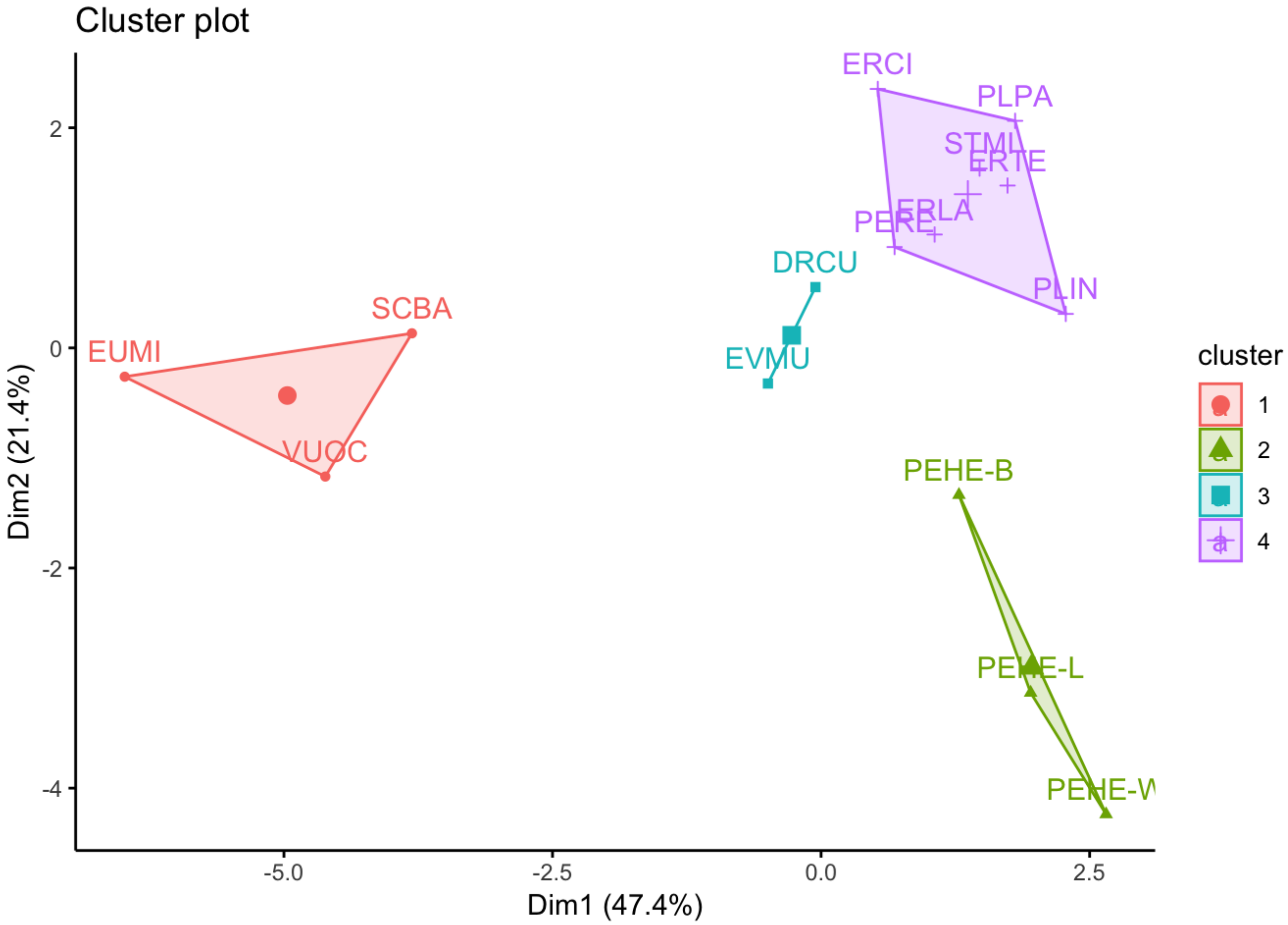
```
##      ThetaHT Psib_4.5.5 SigmaPsib_4.5.5   Tb   To   kT Psib_0 SigmaPsib_0
## VUOC     887       0.19            0.51 0.00  7.7 0.04  99.00       99.00
## EUMI     985       0.07            0.26 0.00 10.0 0.04  99.00       99.00
## DRCU    1230      -0.40            0.45 2.03 21.1 0.19  99.00       99.00
## EVMU    2153      -0.74            0.35 0.00 21.0 0.07  99.00       99.00
## SCBA     806      -0.53            0.28 0.00 15.3 0.14  99.00       99.00
## ERLA     447      -0.48            0.41 3.86 15.0 0.12   0.01        0.02
##      Psib_1 SigmaPsib_1 Psib_2 SigmaPsib_2 Psib_3 SigmaPsib_3 Psib_4
## VUOC  99.00       99.00  99.00       99.00   0.13        0.65   0.12
## EUMI  99.00       99.00  99.00       99.00  99.00       99.00   0.05
## DRCU  -0.09        0.22  -0.10        0.13  -0.63        0.33  -0.41
## EVMU  -0.20        0.34  -0.24        0.24  -0.54        0.27  -0.66
## SCBA  99.00       99.00  99.00       99.00   0.41        0.60  -0.53
## ERLA   0.29        0.48   0.05        0.40  -0.05        0.61  -0.45
##      SigmaPsib_4
## VUOC        0.48
## EUMI        0.35
## DRCU        0.30
## EVMU        0.27
## SCBA        0.22
## ERLA        0.37
```

```
#Scaling data for kmeans analyses
field_all_scale=scale(field_all)

#K mean. Try 4 cluseters initially, based on prior data inspection
k4 = kmeans(field_all_scale,centers=4, nstart = 25)
k4$cluster
```

```
##   VUOC   EUMI   DRCU   EVMU   SCBA   ERLA PEHE-B PEHE-L PEHE-W   PERE
##      1      1      3      3      1      4      2      2      2      4
##   STMI   PLPA   PLIN   ERCI   ERTE
##      4      4      4      4      4
```

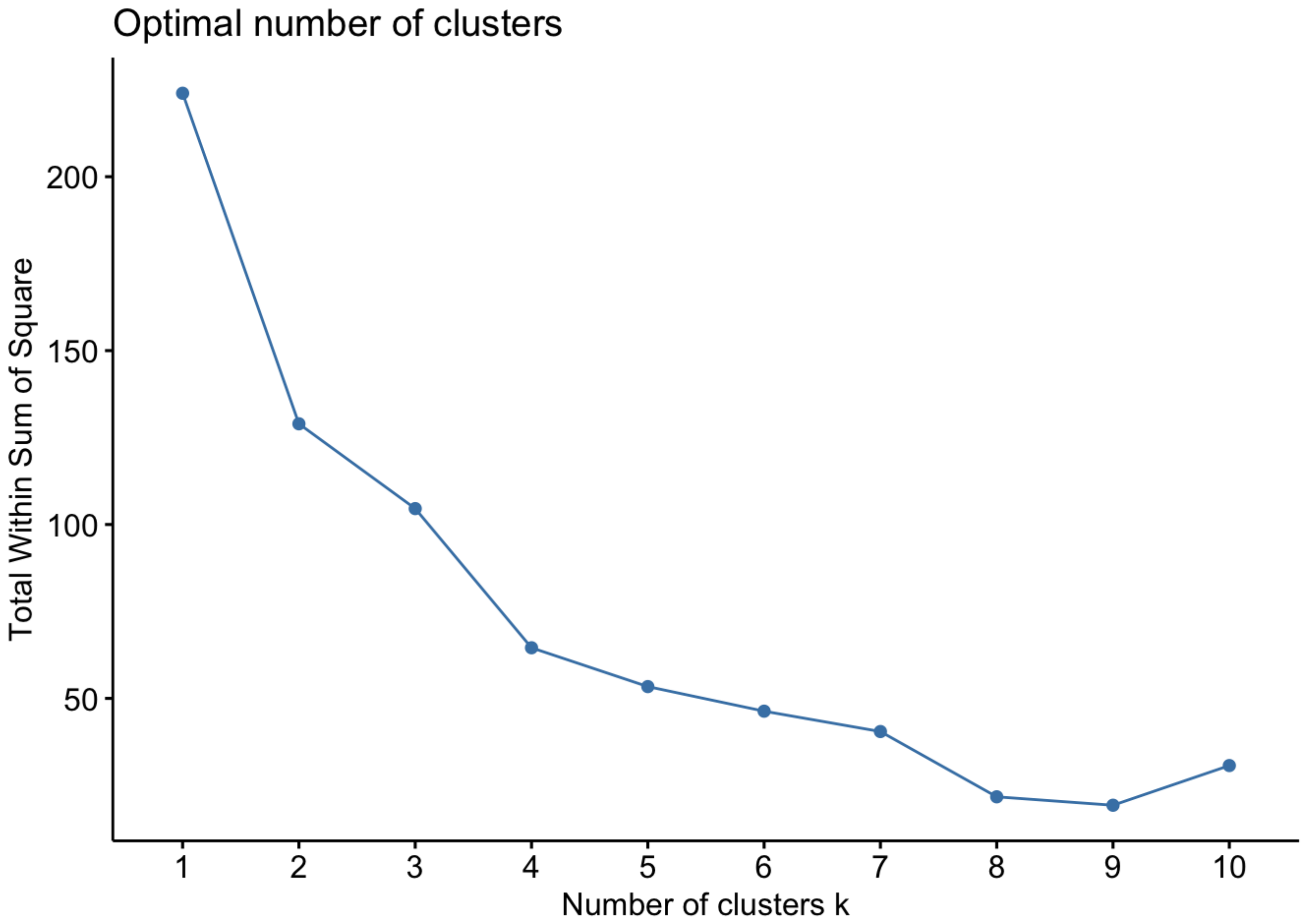
```
#Visualize 4 clusters
fviz_cluster(k4,data=field_all_scale,ggtheme = theme_classic())
```



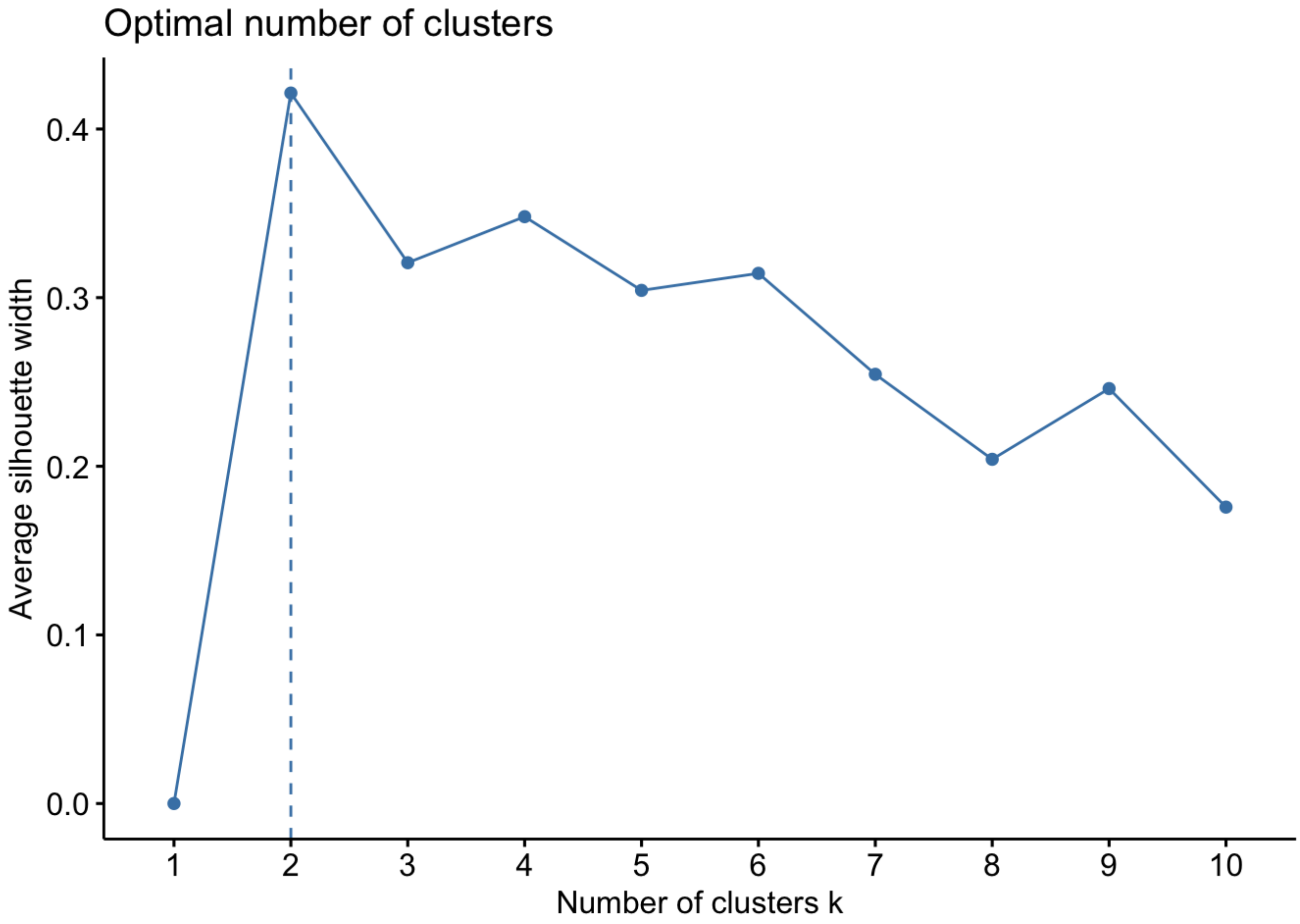
```
#Try different number of clusters
k2 <- kmeans(field_all_scale, centers = 2, nstart = 25)
k3 <- kmeans(field_all_scale, centers = 3, nstart = 25)
k5 <- kmeans(field_all_scale, centers = 5, nstart = 25)

# Compare plots of different number of clusters (plots not shown)
p2 <- fviz_cluster(k2, geom = "point", data = field_all_scale) + ggtitle("k = 2")
p3 <- fviz_cluster(k3, geom = "point",  data = field_all_scale) + ggtitle("k = 3")
p4 <- fviz_cluster(k4, geom = "point",  data = field_all_scale) + ggtitle("k = 4")
p5 <- fviz_cluster(k5, geom = "point",  data = field_all_scale) + ggtitle("k = 5")
# grid.arrange(p2, p3, p4, p5, nrow = 2)

# Use the "Elbow method" to find optimal number of clusters
set.seed(123)
fviz_nbclust(field_all_scale, kmeans, method = "wss") # suggests 4 is the optimal number
```



```
# Use the "Silhouette method" to find optimal number of clusters
# This method determines how well each object lies within its cluster
fviz_nbclust(field_all_scale,kmeans,method='silhouette') # suggests 4 is the optimal num
ber (2nd largest following the number 2)
```



```
#Compute summarizing stats for the variables
groupmeans= field_all %>%
  mutate(Cluster=k4$cluster) %>%
  group_by(Cluster) %>%
  summarise_all("mean")
```

Conclusions: The 15 species can be best grouped into 4 clusters. The first two principal components explained 47.4%+21.4%=68.8% of the variance. Characteristics of each group can be described according to the `groupmeans`.

3. Visualization of germination niche

- Example data includes simulated germination percentage of four plant species, across a range of temperatures and water potentials (lower value indicating higher drought stress)
- Bellow are sample codes for four species, from more dormant to less dormant: vuoc, drcu, erte, plin

```
#Check data, take erte as an example
head(erte)
```

```
##      T   WP Species SimG
## 2661 6 -0.1    ERTE 0.12
## 2662 6 -0.2    ERTE 0.07
## 2663 6 -0.3    ERTE 0.03
## 2664 6 -0.4    ERTE 0.01
## 2665 6 -0.5    ERTE 0.01
## 2666 6 -0.6    ERTE 0.00
```

Generating wireframe surface plot for each species

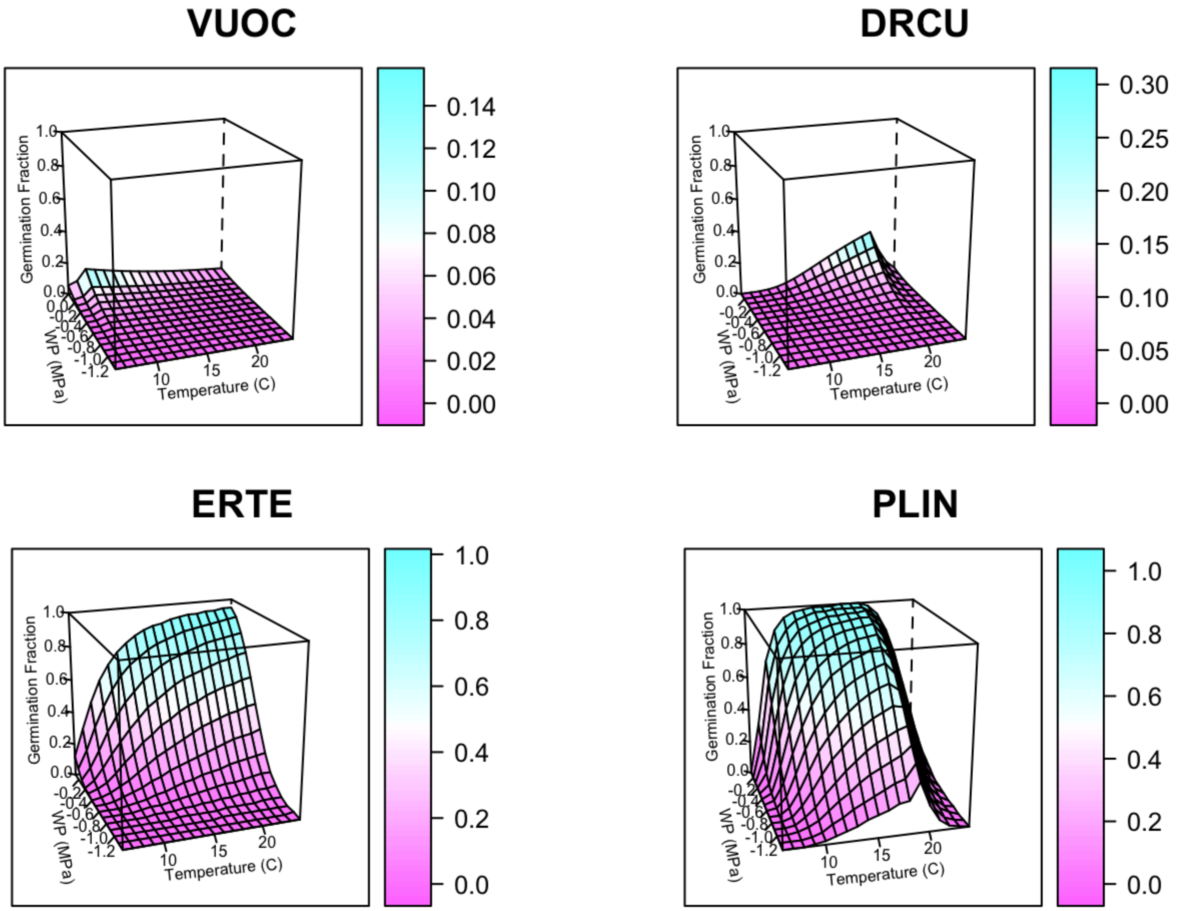
```
Wvuoc <-
  wireframe(vuoc[,4]~vuoc[,1]*vuoc[,2],main="VUOC",zlim=c(0,1),
          drape = TRUE,screen=list(z=20,x=-70,y=0),
          ylab=list("WP (MPa)",rot=280,cex=0.5),xlab=list("Temperature (C)",rot=6,cex=0.
5),zlab=list("Germination Fraction",rot=92,cex=0.5),
          scales=list(arrows=FALSE,y=list(distance=1.2),x=list(distance=0.8),z=list(dist
ance=1.2),cex=0.5))

Wdrcu <-
  wireframe(drcu[,4]~drcu[,1]*drcu[,2],main="DRCU",zlim=c(0,1),
          drape = TRUE,screen=list(z=20,x=-70,y=0),
          ylab=list("WP (MPa)",rot=280,cex=0.5),xlab=list("Temperature (C)",rot=6,cex=0.
5),zlab=list("Germination Fraction",rot=92,cex=0.5),
          scales=list(arrows=FALSE,y=list(distance=1.2),x=list(distance=0.8),z=list(dist
ance=1.2),cex=0.5))

Werte <-
  wireframe(erte[,4]~erte[,1]*erte[,2],main="ERTE",zlim=c(0,1),
          drape = TRUE,screen=list(z=20,x=-70,y=0),
          ylab=list("WP (MPa)",rot=280,cex=0.5),xlab=list("Temperature (C)",rot=6,cex=0.
5),zlab=list("Germination Fraction",rot=92,cex=0.5),
          scales=list(arrows=FALSE,y=list(distance=1.2),x=list(distance=0.8),z=list(dist
ance=1.2),cex=0.5))

Wplin <-
  wireframe(plin[,4]~plin[,1]*plin[,2],main="PLIN",zlim=c(0,1),
          drape = TRUE,screen=list(z=15,x=-70,y=0),
          ylab=list("WP (MPa)",rot=280,cex=0.5),xlab=list("Temperature (C)",rot=6,cex=0.
5),zlab=list("Germination Fraction",rot=92,cex=0.5),
          scales=list(arrows=FALSE,y=list(distance=1.2),x=list(distance=0.8),z=list(dist
ance=1.2),cex=0.5))

grid.arrange(Wvuoc,Wdrcu,Werte,Wplin,nrow=2)
```



4. Mapping of introduction routes of invasive plant species worldwide

The plot generated bellow shows how invasive plant species worldwide travel across countries.

- Dataframe `centroids` has been loaded, which includes the geographic coordinates of the centroids of each country, as well as the number of invasive and native species found in that country
- Dataframe `routes` has been loaded, which contains estimated frequencies of invasive species introduction between countries

```
head(centroids,3)
```

```
##   ISO     UNREGION1 Native_freq Alien_freq       LAT   LONG total
## 1 AFG Southern Asia          32          5  33.00000  66.00    37
## 2 AGO Middle Africa          13          4 -12.50000  18.50    17
## 3 AIA     Caribbean           1         24  18.21667 -63.05    25
##      radius perc_native perc_alien
## 1 1.2333333   0.8648649  0.1351351
## 2 0.5666667   0.7647059  0.2352941
## 3 0.8333333   0.0400000  0.9600000
```

```
head(routes,3)
```

```
##    X  N_to_A freq Native Alien
## 1  8 AFG-AUS   22    AFG   AUS
## 2 18 AFG-CAN   20    AFG   CAN
## 3 87 AFG-NZL   14    AFG   NZL
```

```
routes=routes[,c("Native","Alien","freq")]

# Select high frequency routes (freq>30)
routes_highfreq <- routes[routes$freq > 30,]
```

Create a network object

```
country_network<-network(routes_highfreq,
                 matrix.type='edgelist',
                 directed=FALSE,  # this will be an undirected network
                 ignore.eval=FALSE,
                 names.eval='freq'  # names for the edge weights
)

# attach the appropriate latitute and longitude coordinates
country_network%v%'LONG'<-sapply(network.vertex.names(country_network),function(name){
  centroids[centroids$ISO==name,]$LONG
})

country_network%v%'LAT'<-sapply(network.vertex.names(country_network),function(name){
  centroids[centroids$ISO==name,]$LAT
})
```

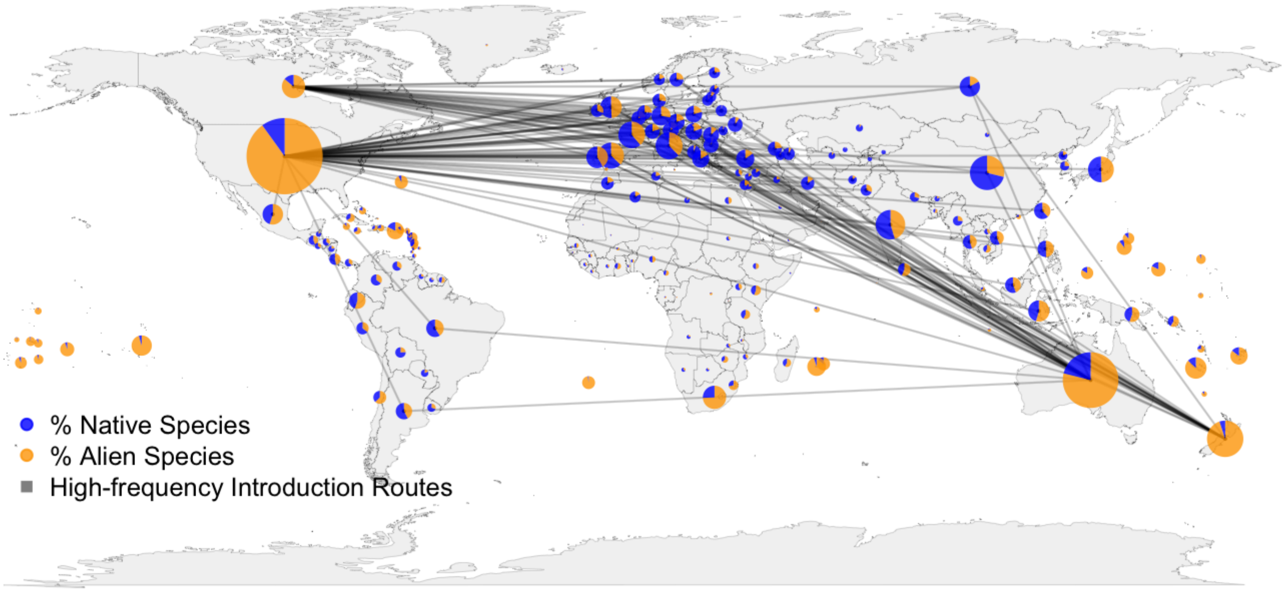
Plot the network using the country centroids coordinates

```
map('world',fill=TRUE,col='#f2f2f2',lwd=0.08,mar=c(1,1,1,0.1))
plot.network(country_network,
             new=FALSE,
             # get coordiantes from vertices and pass in as 2-col matrix
             coord=cbind(country_network%v%'LONG',country_network%v%'LAT'),
             # set a semi-transparent edge color
             edge.col=alpha("black",0.2),
             # specifiy an edge width scaled as fraction of total co-occurence
             edge.lwd=country_network%e%'freq'/150,
             # set the vertex size
             vertex.cex=0.1,
             usearrows=FALSE,
             arrowhead.cex=0.5,
             vertex.col=FALSE, #color of the connecting points
             jitter=FALSE)

# Add pies indicating the percentage of alien and native status
for (i in 1:nrow(centroids)) {
  add.pie(z=c(centroids[i,]$perc_alien,centroids[i,]$perc_native),
          x=centroids[i,]$LONG,y=centroids[i,]$LAT,
          radius =centroids[i,]$radius,col=c(alpha("orange",0.8),alpha("blue",0.8)),
          labels="",border=FALSE)
}

#Add legend
legend(-180,-20,title=" ",legend=c("% Native Species","% Alien Species"),
       col=c(alpha("blue",0.8),alpha("orange",0.8)),pch=19,cex=0.8,bty="n")

legend(-180,-38,title=" ",legend=c("High-frequency Introduction Routes"),
       col=c(alpha("black",0.5)),pch=15,cex=0.8, bty="n")
```



```
#dev.print('filename') # If needed to save to file
```