

Scripts

#Packages load

```
pacman::p_load(RVAideMemoire, car, rstatix, effectsize,  
               ggplot2, factoextra, corrplot, qgraph)
```

#Data.frame load

```
data <- read.csv2("data.csv")
```

#Normality test -> Shapiro Wilk-test by groups -> RVAideMemoire package

```
byf.shapiro(dependent variable ~ independent variable,  
data.frame)
```

#Homogeneity of variance -> Levene test by groups -> car package

```
leveneTest(dependent variable ~ independent variable,  
data.frame, center=median)
```

#Outliers verification -> rstatix package

```
outliers <- data.frame %>%  
group_by(group) %>%  
identify_outliers(value)
```

#Mean test -> t-test by groups -> native stats package

```
t.test(dependent variable ~ independent variable,  
data.frame, var.equal=T)
```

Obs. For data that do not exhibit homogeneity of variance, the argument `var.equal = FALSE` should be used to adjust for variance through the modification of degrees of freedom.

#Effect Size -> Cohen d -> effectsize package

```
data_dependente variable <- data.frame(value.1 =
c(x,y,...,z),value.2 = c(x,y,...,z))

effectsize::hedges_g(data.frame$dependente variable,
date.frame$dependente variable, paired = TRUE)

effectsize::interpret_g(value hedges_g, rules =
"cohen1988")
```

#Graphic effectsize -> ggplot package

```
ggplot(morpho, aes(x = ind, y = value, fill = area))+
geom_line(linetype = 2)+
geom_hline(yintercept = 408.07, size = 1, linetype = 1,
color = "blue")+
geom_hline(yintercept = 345.26, size = 1, linetype = 1,
color = "red")+
geom_point(shape = 21, size = 3.5)+
scale_fill_manual(values = c("blue", "red"))+
annotate("rect",
ymin = 345.26, ymax = 408.07,
xmin = -Inf, xmax = Inf,
alpha = 0.2, fill = "gray")+
xlab("title axis x")+
ylab("title axis y")+
geom_text(data = morpho, aes(x = 3, y = 380,
label = "Large", fontface = "italic"),
size = 5)+
geom_text(data = morpho, aes(x = 3, y = 370,
label = "Hedges g = 2.61",
fontface = "italic"), size = 4)+
geom_text(data = data, aes(x = 2, y = 423,
label = "t(8) = 7.48", fontface = "italic"),
size = 3.5)+
geom_text(data = data, aes(x = 2, y = 415,
label = "p < 0.001", fontface = "italic"),
size = 3.5)+
geom_text(data = data, aes(x = 5, y = 425, label = "A",
fontface = "bold"), size = 5.5)+
theme_classic()+
theme(axis.title.x = element_blank(),
axis.text.x = element_text(size = 10,
color = "black"),
axis.title.y = element_text(size = 14,
color = "black",
face = "bold"),
```

```

axis.text.y = element_text(size = 10,
                             color = "black"),
legend.title = element_blank(),
legend.position = "center",
legend.text = element_text(size = 12,
                             color = "black"))

```

#Principal Component Analysis (PCA)

#Renaming lines -> stats function

```

data_pca=row.names(data)<-
c("S1","S2","S3","S4","S5",
  "F1","F2","F3","F4","F5")
print(data_pca)

```

#Determination of covariance -> stats function

```

pca = prcomp(data[,c(3:25)], center = T,
              scale. = T)
print(pca)

```

#Determining eigenvalues and variance -> factoextra package

```

eig.val = get_eigenvalue(pca)
eig.val

```

#Scree plot -> factoextra package

```

tiff("Variance.tiff", width = 4, height = 4,
     units = 'in', res = 300)
fviz_eig(pca, addlabels = T, ylim = c(0,50))
dev.off()

```

#Correlation matrix -> corrplot package

```

var=get_pca_var(pca)

tiff("Cor_PCA.tiff", width = 4, height = 4,
     units = 'in', res = 300)
corrplot(var$cos2, is.corr = F)
dev.off()

```

```
#Selection of the top 10 traits most correlated with
dimensions 1 and 2 -> factoextra package
```

```
a = fviz_contrib(pca, choice = "var", axes = 1,
                 top = 5)
b = fviz_contrib(pca, choice = "var", axes = 2,
                 top = 5)
```

Obs. To choose more than five variables per dimension, change the argument top = "desired value".

```
tiff("Contri_Dim.tiff", width = 4, height = 4,
     units = 'in', res = 300)
grid.arrange(a,b, ncol = 1)
dev.off()
```

```
#PCA Biplot
```

```
#Selection of data.frame columns -> stat function
```

```
pca3=prcomp(data[,c(8,10,11,14,6,13,12,18,1
5,9)], center = T, scale. = T)
```

```
#PCA Biplot construction -> factoextra package
```

```
fviz_pca_biplot(pca3, repel = T,
                 geom = c("point"),
                 col.var = "blue",
                 col.ind = as.factor(data$area),
                 palette = c("#E7B800", "#FC4E07"),
                 addEllipses = T,
                 ellipse.level = 0.95,
                 legend.title = "Environments",
                 ggtheme = theme_classic())
```

```
ggsave("PCA3.tiff")
```

```
#Interaction networks -> qgraph package
```

```
qgraph(cor(M),
        graph = 'cor',
        threshold = 0.05,
        shape = "circle",
        posCol = c("#009900", "brown"),
        negCol = c("#BF0000", "blue"),
        negDashed = T,
        curveAll = T,
```

```

curveDefault = 0.9,
curveShape = -1,
curveScale = T,
minimum = 0.65,
esize= 10,
layout = 'spring',
title = "Interactions network of functional
traits",
title.cex = 3,
filetype = "tiff",
filename = "con_pca_traits",
height = 14,
width = 14,
normalize = T)

```

Obs. To increase the minimum cutoff value, change the argument minimum = desired value.

#Integration network parameters

#Number of Nodes

```

num_nodes <- ncol(M)

```

#Remove the main diagonal to avoid counting correlations of a variable with itself

```

M[lower.tri(M, diag = TRUE)]

```

#Setting the minimum value

```

value_minimum <- 0.65

```

Count positive correlations above threshold

```

num_correlation_positive <- sum(M >
value_minimum, na.rm = TRUE)

```

Count negative correlations above threshold

```

num_correlation_negatives <- sum(M < -
value_minimum, na.rm = TRUE)

```

#Viewing results

```

cat("num_nodes:", ncol(M_cor), "\n")
cat("num_correlation_positive(>|0.65|):",
num_correlation_positives, "\n")

```

```
cat("num_correlation_negative(<-0.65):",
num_correlation_negatives, "\n")
```

#Linear Regression Leaf Economic Spectrum -> corrplot package

```
sle <- select(data, LMA, THIC, Pal, Subep, C13)
M <- cor(sle)
p.mat <- cor.mtest(sle)$p

jpeg('leaf_spectrum.jpg', width = 2000, height = 2000,
res = 600)
corrplot(M,
  method = "pie",
  type = "lower
  tl.cex = 1.2,
  tl.col = "black",
  cl.cex = 0.7,
  diag = F,
  p.mat = p.mat,
  sig.level = 0.05,
  insig = "pch",
  addCoef.col = "black")
dev.off()
```

#Linear Regression between Variation and Covariation

#Fitting Linear Models -> stats package

```
traits_df = lm[c(6:12), c(1,2,3,4)]

print(traits_df)
```

#Regression Graph -> ggplot2 package

```
ggplot(anatomy_df, aes(x = var, y = covar))+
  geom_point(size = 4, shape = 21, fill =
'lightgreen')+
  geom_smooth(method = 'lm', alpha = 0.3,
fill = 'lightblue')+
  xlab('')+
  ylab('')+
  labs(title = 'Anatomy Traits')+
  stat_poly_eq(aes(label
paste(..eq.label...,
..adj.rr.label...,
..p.value.label...,
```

```

                                                    sep      =
"*plain(\"\",\")~~")) ,
                                label.x = 0.96, label.y =
0.02,
                                parse = TRUE, coef.digits =
5,
                                formula = y ~ x,
                                color = "dark blue", size =
3)+
  theme_bw()+
  theme(axis.title.x = element_text(size =
14, color = 'black', face = 'bold'),
        axis.text.x = element_text(size =
10, color = "black"),
        axis.title.y = element_text(size =
14, color = "black", face = "bold"),
        axis.text.y = element_text(size =
10, color = "black"),
        title = element_text(size = 14,
color = 'black', face = 'bold'),
        panel.background
                                =
element_rect(fill = "white"),
        panel.grid = element_line(size =
0.0))

```