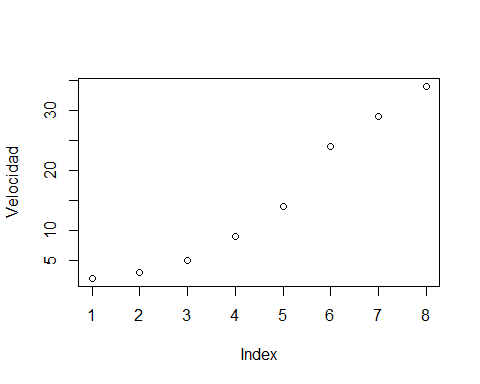
Laboratorio-8.R

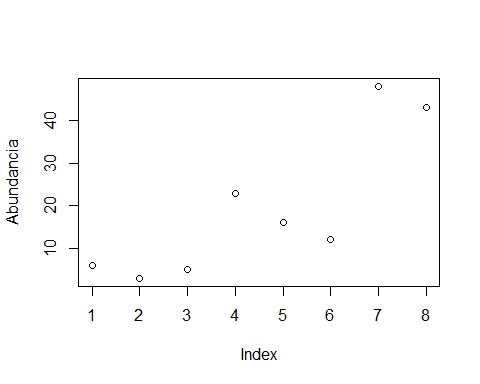
Usuario

2025-10-02

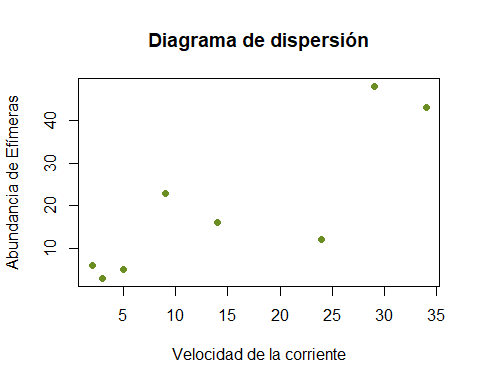
##############################################################################  
  
#Laboratorio 8   
#Tarea: HW\_05   
#02/10/25   
#Evelyn Sofia Ruiz Galarza   
#Dr.Marco Aurelio González Tagle   
  
##############################################################################  
  
# E J E R C I C I O N.1   
  
resp <- data.frame(   
 Velocidad <- c(2,3,5,9,14,24,29,34),  
 Abundancia <- c(6,3,5,23,16,12,48,43)  
)  
  
resp$Rango\_velocidad <- rank(resp$Velocidad,  
 ties.method = "first")  
resp$Rango\_Abundancia <- rank(resp$Abundancia,   
 ties.method = "first")  
plot(Velocidad)



plot(Abundancia)



plot(resp$Velocidad,  
 resp$Abundancia,  
 main = "Diagrama de dispersión",  
 xlab = "Velocidad de la corriente",  
 ylab = "Abundancia de Efímeras",  
 pch = 19,   
 col = "olivedrab")



#Es estadisticamente significativa la correlación?   
#Se rechaza la H0 o la H1?   
#H0=No existe una correlación entre la velocidad del arroyo   
#y la abundancia de efímeras  
#H1=“Existe una correlación positiva entre la velocidad de los arroyos  
#y la abundancia de efímeras   
  
resp$dif<- resp$Rango\_velocidad -  
 resp$Rango\_Abundancia  
resp$dif2<- resp$dif^2  
sum(resp$dif2)

## [1] 16

length(resp$Rango\_velocidad)

## [1] 8

length(resp$Rango\_Abundancia)

## [1] 8

#Para ver si la distribución de mis datos es normal   
  
shapiro.test(resp$Velocidad....c.2..3..5..9..14..24..29..34.)

##   
## Shapiro-Wilk normality test  
##   
## data: resp$Velocidad....c.2..3..5..9..14..24..29..34.  
## W = 0.89444, p-value = 0.2572

shapiro.test(resp$Abundancia....c.6..3..5..23..16..12..48..43.)

##   
## Shapiro-Wilk normality test  
##   
## data: resp$Abundancia....c.6..3..5..23..16..12..48..43.  
## W = 0.85403, p-value = 0.1046

#Los datos no son normales, ya que son   
#mayores a 0.05 es por eso que se utiliza el método  
#de Spearman   
  
  
#Función de correlación   
  
cor.test(resp$Rango\_velocidad,  
 resp$Rango\_Abundancia,  
 method = "spearman")

##   
## Spearman's rank correlation rho  
##   
## data: resp$Rango\_velocidad and resp$Rango\_Abundancia  
## S = 16, p-value = 0.02178  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.8095238

cor.test(resp$Velocidad....c.2..3..5..9..14..24..29..34.,   
 resp$Abundancia....c.6..3..5..23..16..12..48..43.,  
 method = "spearman")

##   
## Spearman's rank correlation rho  
##   
## data: resp$Velocidad....c.2..3..5..9..14..24..29..34. and resp$Abundancia....c.6..3..5..23..16..12..48..43.  
## S = 16, p-value = 0.02178  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.8095238

#p=VALUE: 0.02178   
#R= 0.80 Nos indica una correlación muy alta   
#Ambas variables se explican una a la otra   
#Se rechaza la Hipótesis nula   
  
##############################################################################  
  
#E J E R C I C I O N.2   
  
datos\_suelo <- read.csv("suelo.csv")  
  
library(Hmisc)

##   
## Adjuntando el paquete: 'Hmisc'

## The following objects are masked from 'package:base':  
##   
## format.pval, units

library(reshape2)  
  
#Para empezar a trabajar con la base de datos   
#se selecciona primero todos los datos númericos   
  
variables <- datos\_suelo[, c("pH", "N", "Dens", "P", "Ca", "Mg","K",  
 "Na", "Conduc")]  
  
#Se calculan correlaciones y p-values con Pearson   
  
resultados <- rcorr(as.matrix(variables),type = "pearson")  
  
#Extraer resultados de correlaciones y p-values   
  
cor\_matrix <- resultados$r  
p\_matrix <- resultados$P  
  
#Upper.tri para convertir los datos a una tabla sin   
#duplicados ni diagonal  
  
get\_corr\_table <- function(cor\_matrix, p\_matrix) {  
 ut <- upper.tri(cor\_matrix)  
 data.frame(  
 Var1=rownames(cor\_matrix)[row(cor\_matrix)[ut]],  
 Var2=colnames(cor\_matrix)[col(cor\_matrix)[ut]],  
 Correlation = cor\_matrix[ut],  
 p\_value= p\_matrix[ut]  
 )  
}  
  
results <- get\_corr\_table(cor\_matrix, p\_matrix)  
  
print(results)

## Var1 Var2 Correlation p\_value  
## 1 pH N 0.38811455 0.3895987485  
## 2 pH Dens -0.77369126 0.0412492796  
## 3 N Dens -0.79266282 0.0335058897  
## 4 pH P 0.42061197 0.3473966472  
## 5 N P 0.94101589 0.0015719124  
## 6 Dens P -0.78657314 0.0358942516  
## 7 pH Ca 0.56848734 0.1829719669  
## 8 N Ca 0.69412870 0.0835908337  
## 9 Dens Ca -0.79809646 0.0314518872  
## 10 P Ca 0.57439198 0.1774320223  
## 11 pH Mg -0.61115331 0.1448352643  
## 12 N Mg -0.43103915 0.3342844506  
## 13 Dens Mg 0.45828088 0.3010542750  
## 14 P Mg -0.45099416 0.3097948199  
## 15 Ca Mg -0.01009406 0.9828646629  
## 16 pH K 0.37094191 0.4126847718  
## 17 N K 0.18594583 0.6897574314  
## 18 Dens K -0.49128624 0.2628616908  
## 19 P K 0.43976248 0.3234801257  
## 20 Ca K 0.18456449 0.6919827944  
## 21 Mg K -0.01344459 0.9771778187  
## 22 pH Na -0.71143799 0.0730110649  
## 23 N Na -0.85248154 0.0148032129  
## 24 Dens Na 0.89502105 0.0064764582  
## 25 P Na -0.93224595 0.0022122973  
## 26 Ca Na -0.65215650 0.1124057616  
## 27 Mg Na 0.55987093 0.1912050716  
## 28 K Na -0.51761397 0.2340925564  
## 29 pH Conduc -0.80139013 0.0302420518  
## 30 N Conduc -0.78881244 0.0350053599  
## 31 Dens Conduc 0.95770170 0.0006907938  
## 32 P Conduc -0.80028840 0.0306437882  
## 33 Ca Conduc -0.84959432 0.0155129011  
## 34 Mg Conduc 0.39241421 0.3839018130  
## 35 K Conduc -0.50660743 0.2459325415  
## 36 Na Conduc 0.92307132 0.0030236919

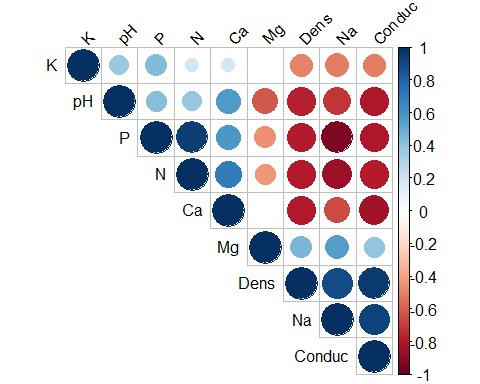
#Se debe adaptar la información en un cuadro como el   
#Cuadro ejemplo 3   
  
#Crear una columna de "Conjunto"  
results$Conjunto <- paste(results$Var1, "-", results$Var2)  
  
#Se reordenan las columnas   
Tabla\_final <- results[, c("Conjunto","Correlation","p\_value")]  
  
#Renombrar   
colnames(Tabla\_final) <- c("Conjunto","r","valor de P")  
print(Tabla\_final)

## Conjunto r valor de P  
## 1 pH - N 0.38811455 0.3895987485  
## 2 pH - Dens -0.77369126 0.0412492796  
## 3 N - Dens -0.79266282 0.0335058897  
## 4 pH - P 0.42061197 0.3473966472  
## 5 N - P 0.94101589 0.0015719124  
## 6 Dens - P -0.78657314 0.0358942516  
## 7 pH - Ca 0.56848734 0.1829719669  
## 8 N - Ca 0.69412870 0.0835908337  
## 9 Dens - Ca -0.79809646 0.0314518872  
## 10 P - Ca 0.57439198 0.1774320223  
## 11 pH - Mg -0.61115331 0.1448352643  
## 12 N - Mg -0.43103915 0.3342844506  
## 13 Dens - Mg 0.45828088 0.3010542750  
## 14 P - Mg -0.45099416 0.3097948199  
## 15 Ca - Mg -0.01009406 0.9828646629  
## 16 pH - K 0.37094191 0.4126847718  
## 17 N - K 0.18594583 0.6897574314  
## 18 Dens - K -0.49128624 0.2628616908  
## 19 P - K 0.43976248 0.3234801257  
## 20 Ca - K 0.18456449 0.6919827944  
## 21 Mg - K -0.01344459 0.9771778187  
## 22 pH - Na -0.71143799 0.0730110649  
## 23 N - Na -0.85248154 0.0148032129  
## 24 Dens - Na 0.89502105 0.0064764582  
## 25 P - Na -0.93224595 0.0022122973  
## 26 Ca - Na -0.65215650 0.1124057616  
## 27 Mg - Na 0.55987093 0.1912050716  
## 28 K - Na -0.51761397 0.2340925564  
## 29 pH - Conduc -0.80139013 0.0302420518  
## 30 N - Conduc -0.78881244 0.0350053599  
## 31 Dens - Conduc 0.95770170 0.0006907938  
## 32 P - Conduc -0.80028840 0.0306437882  
## 33 Ca - Conduc -0.84959432 0.0155129011  
## 34 Mg - Conduc 0.39241421 0.3839018130  
## 35 K - Conduc -0.50660743 0.2459325415  
## 36 Na - Conduc 0.92307132 0.0030236919

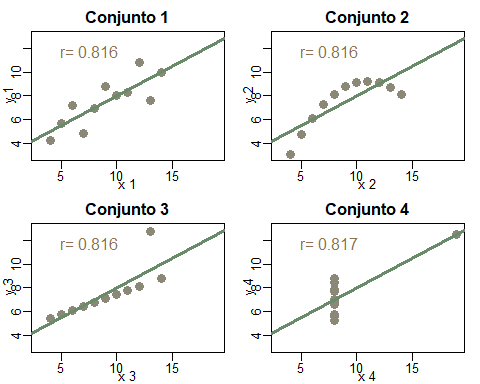
library(corrplot)

## corrplot 0.95 loaded

orden <- c("K", "pH","P","N","Ca","Mg","Dens","Na","Conduc")  
cor\_matrix2 <- cor\_matrix[orden, orden]  
  
corrplot(cor\_matrix2,   
 method = "circle",   
 type="upper",  
 order="original",  
 tl.col= "black",  
 tl.srt= 45,  
 cl.cex =1,  
 cl.pos="r",  
 )



##############################################################################  
  
# E J E R C I C I O N.3   
  
#Cargar base de datos   
data("anscombe")  
  
#Ajustar márgenes mínimos: abajo,   
#izquierda, arrib, derecha   
par(mar=c(2,2,2,1))  
#Establecer una cuadrícula de 2x2   
par(mfrow= c(2,2))  
#Reducir el espacio entre ejes y reducir etiquetas   
par(mgp=c(1, 0.5, 0))  
  
#Configurar área de gráficos con 2 filas, 2 columnas  
par(mfrow=c(2,2))  
  
#Crear los 4 gráficos solicitados   
for(i in 1:4) {  
x <- anscombe[,i] #Columnas x1, x2, x3, x4   
y <- anscombe [, i + 4 ] #Columnas y1,y2,y3,y4  
  
#Gráficos   
  
plot(x, y,   
 main= paste("Conjunto", i),   
 xlab=paste("x", i),  
 ylab=paste("y", i),  
 pch= 19,   
 col= "cornsilk4",   
 cex=1.5,  
 xlim=c(3,19),  
 ylim=c(3,13))  
  
#Línea de regresión   
abline(lm(y ~ x ), col= "darkseagreen4", lwd= 3)  
  
#Estadísticas   
  
r <- round(cor(x,y),3)  
legend("topleft",   
 legend = paste("r=",r),  
 bty="n",  
 text.col="burlywood4",  
 cex=1.2)  
}



#Para restaurarlo a configuración normal   
par(mfrow = c(1,1))  
  
##################################################################  
  
# E J E R C I C I O N.3   
  
x1 <- c(10.0, 8.0, 13.0, 9.0, 11.0, 14.0, 6.0, 4.0, 12.0, 7.0, 5.0)  
y1 <- c(8.04, 6.95, 7.58, 8.81, 8.33, 9.96, 7.24, 4.26, 10.84, 4.82, 5.68)  
  
x2 <- c(10.0, 8.0, 13.0, 9.0, 11.0, 14.0, 6.0, 4.0, 12.0, 7.0, 5.0)  
y2 <- c(9.14, 8.14, 8.74, 8.77, 9.26, 8.10, 6.13, 3.10, 9.13, 7.26, 4.74)  
  
x3 <- c(10.0, 8.0, 13.0, 9.0, 11.0, 14.0, 6.0, 4.0, 12.0, 7.0, 5.0)  
y3 <- c(7.46, 6.77, 12.74, 7.11, 7.81, 8.84, 6.08, 5.39, 8.15, 6.42, 5.73)  
  
x4 <- c(8.0, 8.0, 8.0, 8.0, 8.0, 8.0, 8.0,19.0, 8.0, 8.0, 8.0)  
y4 <- c(6.58, 5.76, 7.71, 8.84, 8.47, 7.04, 5.25, 12.50, 5.56, 7.91, 6.8)  
  
calculo\_prop<- function(x, y){  
 modelo <- lm(y ~ x)  
 list(  
 media\_x = mean(x),  
 var\_x = var(x),  
 media\_y = mean(y),  
 var\_y = var(y),  
 correlacion = cor(x,y),  
 regresion = coef(modelo),  
 R2 = summary(modelo)$r.squared  
 )  
}  
#Resultados para cada conjunto  
result1 <- calculo\_prop(x1, y1)  
result2 <- calculo\_prop(x2, y2)  
result3 <- calculo\_prop(x3, y3)  
result4 <- calculo\_prop(x4, y4)  
  
#Mostrar resultados   
result1

## $media\_x  
## [1] 9  
##   
## $var\_x  
## [1] 11  
##   
## $media\_y  
## [1] 7.500909  
##   
## $var\_y  
## [1] 4.127269  
##   
## $correlacion  
## [1] 0.8164205  
##   
## $regresion  
## (Intercept) x   
## 3.0000909 0.5000909   
##   
## $R2  
## [1] 0.6665425

result2

## $media\_x  
## [1] 9  
##   
## $var\_x  
## [1] 11  
##   
## $media\_y  
## [1] 7.500909  
##   
## $var\_y  
## [1] 4.127629  
##   
## $correlacion  
## [1] 0.8162365  
##   
## $regresion  
## (Intercept) x   
## 3.000909 0.500000   
##   
## $R2  
## [1] 0.666242

result3

## $media\_x  
## [1] 9  
##   
## $var\_x  
## [1] 11  
##   
## $media\_y  
## [1] 7.5  
##   
## $var\_y  
## [1] 4.12262  
##   
## $correlacion  
## [1] 0.8162867  
##   
## $regresion  
## (Intercept) x   
## 3.0024545 0.4997273   
##   
## $R2  
## [1] 0.666324

result4

## $media\_x  
## [1] 9  
##   
## $var\_x  
## [1] 11  
##   
## $media\_y  
## [1] 7.492727  
##   
## $var\_y  
## [1] 4.134982  
##   
## $correlacion  
## [1] 0.8166967  
##   
## $regresion  
## (Intercept) x   
## 2.9861818 0.5007273   
##   
## $R2  
## [1] 0.6669935

#Para mostrar los resultados en un formato tabla   
calc\_prop\_df <- function(x, y){  
 modelo <- lm(y ~ x)  
 data.frame(  
 `Media de x` = mean(x),  
 `Varianza de x` = var(x),  
 `Media de y` = mean(y),  
 `Varianza de y` = var(y),  
 `Correlación x-y` = cor(x,y),  
 `Intersección` = coef(modelo)[1],  
 `Pendiente` = coef(modelo)[2],  
 `R2` = summary(modelo)$r.squared  
 )  
}  
  
#Resultados para cada conjunto  
df1 <- calc\_prop\_df(x1, y1)  
df2 <- calc\_prop\_df(x2, y2)  
df3 <- calc\_prop\_df(x3, y3)  
df4 <- calc\_prop\_df(x4, y4)  
  
#Se une todo en una sola tabla   
  
cuadro4 <- rbind(df1, df2, df3, df4)  
rownames(cuadro4) <- c("Conjunto I", "Conjunto II", "Conjunto III", "Conjunto IV")  
  
print(cuadro4)

## Media.de.x Varianza.de.x Media.de.y Varianza.de.y Correlación.x.y  
## Conjunto I 9 11 7.500909 4.127269 0.8164205  
## Conjunto II 9 11 7.500909 4.127629 0.8162365  
## Conjunto III 9 11 7.500000 4.122620 0.8162867  
## Conjunto IV 9 11 7.492727 4.134982 0.8166967  
## Intersección Pendiente R2  
## Conjunto I 3.000091 0.5000909 0.6665425  
## Conjunto II 3.000909 0.5000000 0.6662420  
## Conjunto III 3.002455 0.4997273 0.6663240  
## Conjunto IV 2.986182 0.5007273 0.6669935

#Para generar el cuadro 5   
  
cuadro5 <- data.frame(  
 I\_x =x1, I\_y= y1,   
 II\_x= x2, II\_y = y2,  
 III\_x = x3, III\_y = y3,  
 IV\_x = x4, IV\_y = y4  
)  
  
print(cuadro5)

## I\_x I\_y II\_x II\_y III\_x III\_y IV\_x IV\_y  
## 1 10 8.04 10 9.14 10 7.46 8 6.58  
## 2 8 6.95 8 8.14 8 6.77 8 5.76  
## 3 13 7.58 13 8.74 13 12.74 8 7.71  
## 4 9 8.81 9 8.77 9 7.11 8 8.84  
## 5 11 8.33 11 9.26 11 7.81 8 8.47  
## 6 14 9.96 14 8.10 14 8.84 8 7.04  
## 7 6 7.24 6 6.13 6 6.08 8 5.25  
## 8 4 4.26 4 3.10 4 5.39 19 12.50  
## 9 12 10.84 12 9.13 12 8.15 8 5.56  
## 10 7 4.82 7 7.26 7 6.42 8 7.91  
## 11 5 5.68 5 4.74 5 5.73 8 6.80

############################################################################