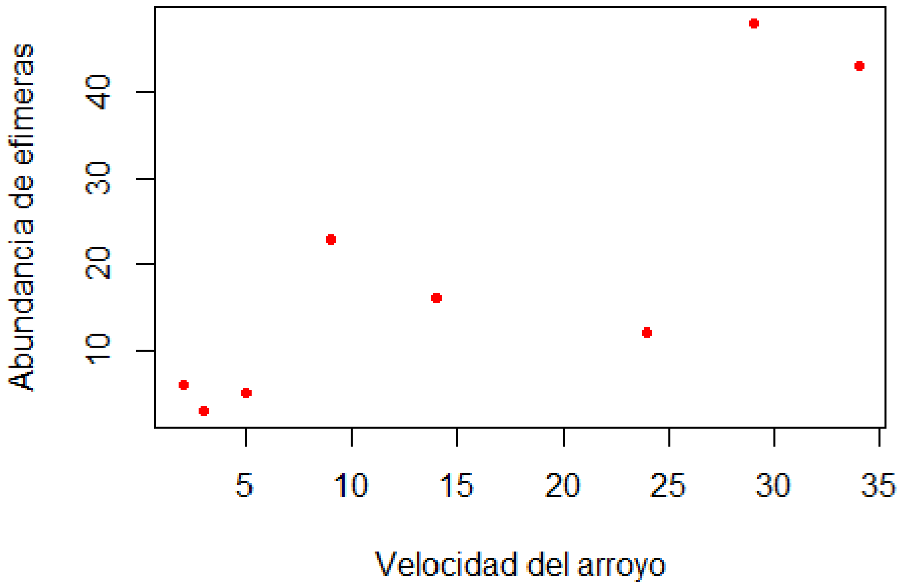
Tarea_Correlacion.R

2025-11-27

```r
#================================
#################################
#Asignación 5: Correlación
#Tarea
#Juan Carlos Delgado Grarza
# Curso 2025 - Dr. Marco Aurelio González Tagle
#01/10/2025
#################################
#================================
# Ejercicio 1
# Efimeras y la velocidad de un arroyo
efimeras<-data.frame(speed = c(2,3,5,9,14,24,29,34),
                     abundance = c(6,3,5,23,16,12,48,43))

# datos por separado en dos variables
speed <-c(2,3,5,9,14,24,29,34)
abundance <-c(6,3,5,23,16,12,48,43)

# gráfico para ver si hay relación entre velocidad del arroyo y cantidad
de efímeras
plot(speed,abundance,
     xlab="Velocidad del arroyo",
     ylab="Abundancia de efimeras",
     col= "red",
     pch =20)
```




```
# Prueba de correlación de Pearson
# Esto va a calcula si existe una relación lineal entre velocidad y
abundancia
cor.test(speed, abundance, method = "pearson")

##
##  Pearson's product-moment correlation
##
## data:  speed and abundance
## t = 3.8568, df = 6, p-value = 0.008393
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3442317 0.9711386
## sample estimates:
##       cor
## 0.8441408

#================================
# Ejercicio 2
# Corelacion entre variables de suelo

# Variables de pH, N, Dencidad,P, Ca, Mg, K, Na

suelo <- data.frame(
  pH = c(5.40, 5.65, 5.14, 5.14, 5.14, 5.10, 4.70),
  N = c(0.188, 0.165, 0.260, 0.169, 0.164, 0.094, 0.100),
  Dens = c(0.92, 1.04, 0.95, 1.10, 1.12, 1.22, 1.52),
```

```r
  P = c(215, 208, 300, 248, 174, 129, 117),
  Ca = c(16.35, 12.25, 13.02, 11.92, 14.17, 8.55, 8.74),
  Mg = c(7.65, 5.15, 5.68, 7.88, 8.12, 6.92, 8.16),
  K = c(0.72, 0.71, 0.68, 1.09, 0.70, 0.81, 0.39),
  Na = c(1.14, 0.94, 0.60, 1.01, 2.17, 2.67, 3.32)
)

# La matriz de correlaciones entre todas las variables del suelo
# Esto nos da los valores de r entre cada par de variables
cor_matrix <- cor(suelo, method = "pearson")
cor_matrix
```

```
##              pH          N       Dens          P          Ca
Mg
## pH    1.0000000  0.3881145 -0.7736913  0.4206120  0.56848734 -
0.61115331
## N     0.3881145  1.0000000 -0.7926628  0.9410159  0.69412870 -
0.43103915
## Dens -0.7736913 -0.7926628  1.0000000 -0.7865731 -0.79809646
0.45828088
## P     0.4206120  0.9410159 -0.7865731  1.0000000  0.57439198 -
0.45099416
## Ca    0.5684873  0.6941287 -0.7980965  0.5743920  1.00000000 -
0.01009406
## Mg   -0.6111533 -0.4310391  0.4582809 -0.4509942 -0.01009406
1.00000000
## K     0.3709419  0.1859458 -0.4912862  0.4397625  0.18456449 -
0.01344459
## Na   -0.7114380 -0.8524815  0.8950210 -0.9322460 -0.65215650
0.55987093
##                K         Na
## pH    0.37094191 -0.7114380
## N     0.18594583 -0.8524815
## Dens -0.49128624  0.8950210
## P     0.43976248 -0.9322460
## Ca    0.18456449 -0.6521565
## Mg   -0.01344459  0.5598709
## K     1.00000000 -0.5176140
## Na   -0.51761397  1.0000000
```

```r
library(Hmisc)
```

```
##
## Adjuntando el paquete: 'Hmisc'

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

```r
res <- rcorr(as.matrix(suelo))
res$r
```

```
##              pH         N       Dens          P          Ca
Mg
## pH    1.0000000  0.3881145 -0.7736913  0.4206120  0.56848734 -
0.61115331
## N     0.3881145  1.0000000 -0.7926628  0.9410159  0.69412870 -
0.43103915
## Dens -0.7736913 -0.7926628  1.0000000 -0.7865731 -0.79809646
0.45828088
## P     0.4206120  0.9410159 -0.7865731  1.0000000  0.57439198 -
0.45099416
## Ca    0.5684873  0.6941287 -0.7980965  0.5743920  1.00000000 -
0.01009406
## Mg   -0.6111533 -0.4310391  0.4582809 -0.4509942 -0.01009406
1.00000000
## K     0.3709419  0.1859458 -0.4912862  0.4397625  0.18456449 -
0.01344459
## Na   -0.7114380 -0.8524815  0.8950210 -0.9322460 -0.65215650
0.55987093
##                K          Na
## pH    0.37094191 -0.7114380
## N     0.18594583 -0.8524815
## Dens -0.49128624  0.8950210
## P     0.43976248 -0.9322460
## Ca    0.18456449 -0.6521565
## Mg   -0.01344459  0.5598709
## K     1.00000000 -0.5176140
## Na   -0.51761397  1.0000000
```

```
res$P
```

```
##              pH           N        Dens           P          Ca
Mg
## pH           NA 0.389598749 0.041249280 0.347396647 0.18297197
0.1448353
## N    0.38959875          NA 0.033505890 0.001571912 0.08359083
0.3342845
## Dens 0.04124928 0.033505890          NA 0.035894252 0.03145189
0.3010543
## P    0.34739665 0.001571912 0.035894252          NA 0.17743202
0.3097948
## Ca   0.18297197 0.083590834 0.031451887 0.177432022          NA
0.9828647
## Mg   0.14483526 0.334284451 0.301054275 0.309794820 0.98286466
NA
## K    0.41268477 0.689757431 0.262861691 0.323480126 0.69198279
0.9771778
## Na   0.07301106 0.014803213 0.006476458 0.002212297 0.11240576
0.1912051
##             K          Na
## pH   0.4126848 0.073011065
```

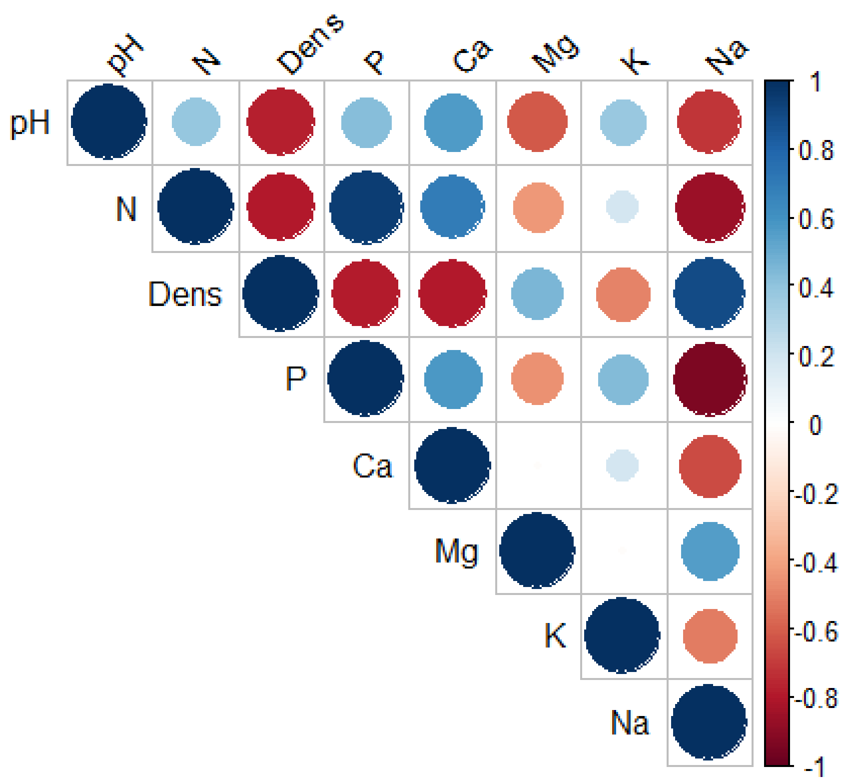
```
## N    0.6897574 0.014803213
## Dens 0.2628617 0.006476458
## P    0.3234801 0.002212297
## Ca   0.6919828 0.112405762
## Mg   0.9771778 0.191205072
## K           NA 0.234092556
## Na   0.2340926          NA
```

```r
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```r
#Creamos el gráfico de correlaciones (Figura 2)

corrplot(cor_matrix, method = "circle", type = "upper",
         tl.col = "black", tl.srt = 45)
```



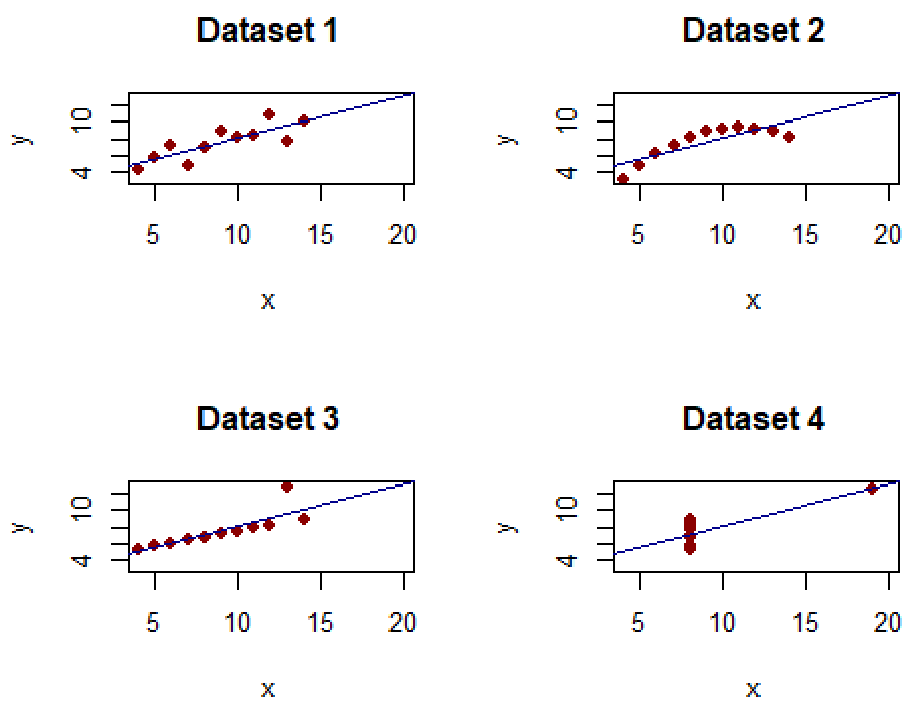

```r
#========================================
#Ejercicio 3
# Cargamos el conjunto de datos 'anscombe'
data(anscombe)

# Ver los datos
anscombe
```

```
##     x1 x2 x3 x4    y1   y2   y3    y4
## 1   10 10 10  8  8.04 9.14  7.46  6.58
## 2    8  8  8  8  6.95 8.14  6.77  5.76
```

```
## 3   13 13 13  8  7.58 8.74 12.74  7.71
## 4    9  9  9  8  8.81 8.77  7.11  8.84
## 5   11 11 11  8  8.33 9.26  7.81  8.47
## 6   14 14 14  8  9.96 8.10  8.84  7.04
## 7    6  6  6  8  7.24 6.13  6.08  5.25
## 8    4  4  4 19  4.26 3.10  5.39 12.50
## 9   12 12 12  8 10.84 9.13  8.15  5.56
## 10   7  7  7  8  4.82 7.26  6.42  7.91
## 11   5  5  5  8  5.68 4.74  5.73  6.89

# Gráficar los 4 conjuntos de datos
par(mfrow=c(2,2)) #
for(i in 1:4){
  x <- anscombe[,i]
  y <- anscombe[,i+4]
  # Configurar el gráfico de dispersión con una línea de regresión
  plot(x, y, main=paste("Dataset", i),
       xlim = c(4,20), ylim = c(3, 13),
       pch = 19, col = "darkred")
  abline(lm(y ~ x), col = "darkblue")
}
```




```
# Los números de resumen (como media y correlación) son casi iguales en
los cuatro casos,
# al ver los gráficos, cada conjunto muestra una relación diferente entre
sus variables.
```

```r
# Calculamos la correlación de Pearson para cada conjunto
for(i in 1:4)
{

  print(cor.test(anscombe[,i],
                 anscombe[,i+4]))
}

##
##  Pearson's product-moment correlation
##
## data:  anscombe[, i] and anscombe[, i + 4]
## t = 4.2415, df = 9, p-value = 0.00217
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4243912 0.9506933
## sample estimates:
##       cor
## 0.8164205
##
##
##  Pearson's product-moment correlation
##
## data:  anscombe[, i] and anscombe[, i + 4]
## t = 4.2386, df = 9, p-value = 0.002179
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4239389 0.9506402
## sample estimates:
##       cor
## 0.8162365
##
##
##  Pearson's product-moment correlation
##
## data:  anscombe[, i] and anscombe[, i + 4]
## t = 4.2394, df = 9, p-value = 0.002176
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4240623 0.9506547
## sample estimates:
##       cor
## 0.8162867
##
##
##  Pearson's product-moment correlation
##
## data:  anscombe[, i] and anscombe[, i + 4]
## t = 4.243, df = 9, p-value = 0.002165
## alternative hypothesis: true correlation is not equal to 0
```

```
## 95 percent confidence interval:
##  0.4246394 0.9507224
## sample estimates:
##       cor
## 0.8165214
```