Analisis de datos

```
#Lectura de datos
d.env <- read.csv(
  paste0(
    'http://geografiafisica.org/sem201801/geo112/',
    'datos_campo_201802/datos/unified_geology_sem_201802_coord.csv'
  )
)
d <- read.csv(
  paste0(
    'http://geografiafisica.org/sem201801/geo112/',
    'datos_campo_201802/datos/unified_geology_sem_201802_samples.csv'
  )
)
#Datos de Kesia Ferreras
d.kf <- d[grep('kesia', d$responsable, ignore.case = T),]

#Supuesto de normalidad
(normalidad <- sapply(d.kf[,c('a','b','c')], shapiro.test))#Ninguna proviene de dist. normal
```

```
##           a                           b
## statistic 0.9762096                   0.9533123
## p.value   0.001353913                 2.63928e-06
## method    "Shapiro-Wilk normality test" "Shapiro-Wilk normality test"
## data.name "X[[i]]"                    "X[[i]]"
##           c
## statistic 0.7236341
## p.value   2.508532e-18
## method    "Shapiro-Wilk normality test"
## data.name "X[[i]]"
```

```
#Correlacion
(correl <- sapply(
  c("pearson", "kendall", "spearman"),
  function(x)
  cor(
    d.kf[,c('a','b','c')],
    use = 'pairwise.complete.obs',
    method = x
  ),
  simplify = F
))
```

```
## $pearson
##           a         b         c
## a 1.0000000 0.7070731 0.4191695
## b 0.7070731 1.0000000 0.4635331
## c 0.4191695 0.4635331 1.0000000
##
## $kendall
##           a         b         c
## a 1.0000000 0.5589305 0.4818879
```

```
## b 0.5589305 1.0000000 0.5165635
## c 0.4818879 0.5165635 1.0000000
##
## $spearman
##           a         b         c
## a 1.0000000 0.7449944 0.6611871
## b 0.7449944 1.0000000 0.6975123
## c 0.6611871 0.6975123 1.0000000
```

```
#Prueba de asociacion (paquete psych)
(pruebasocpsych <- psych::corr.test(
  d.kf[,c('a','b','c')],
  use = 'complete'
))
```

```
## Call:psych::corr.test(x = d.kf[, c("a", "b", "c")], use = "complete")
## Correlation matrix
##      a    b    c
## a 1.00 0.71 0.42
## b 0.71 1.00 0.46
## c 0.42 0.46 1.00
## Sample Size
## [1] 208
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##   a b c
## a 0 0 0
## b 0 0 0
## c 0 0 0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

```
print(pruebasocpsych, digits = 10)
```

```
## Call:psych::corr.test(x = d.kf[, c("a", "b", "c")], use = "complete")
## Correlation matrix
##           a         b         c
## a 1.0000000 0.7070731 0.4191695
## b 0.7070731 1.0000000 0.4635331
## c 0.4191695 0.4635331 1.0000000
## Sample Size
## [1] 208
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##       a b     c
## a 0e+00 0 3e-10
## b 0e+00 0 0e+00
## c 3e-10 0 0e+00
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

```
#Panel de correlaciones
devtools::source_url(
  paste0('https://raw.githubusercontent.com/JoeyBernhardt/',
         'NumericalEcology/master/panelutils.R'
  )
)
```

```r
invisible(
  sapply(
    c(
      "pearson",
      "kendall",
      "spearman"
    ),
    function(x) {
      dev.new()
      p <- pairs(na.omit(d.kf[,c('a','b','c')]),
            cex.labels = 2,
            lower.panel = panel.smooth,
            upper.panel = panel.cor,
            diag.panel = panel.hist,
            method = x,
            main = paste0(
              toupper(substr(x, 1, 1)),
              substr(x, 2, nchar(x)),
              " Correlation Matrix"))
      print(p)
    }
  )
)
```

```
## NULL
```

```
## NULL
```

```
## NULL
```

```r
#Prueba de asociacion (paquete stats)
combinaciones <- as.data.frame(
  t(
    combn(c('a','b','c'), 2)
  )
)

combinaciones$nombre <- paste0(
  combinaciones[,1],
  ' y ',
  combinaciones[,2]
)

(pruebasocstats <- sapply(
  combinaciones$nombre,
  function(x) {
    cor.test(
      d.kf[,as.character(combinaciones[combinaciones$nombre==x, 1])],
      d.kf[,as.character(combinaciones[combinaciones$nombre==x, 2])]
    )
  },
  simplify = F
))
```

```
## $`a y b`
##
```

```
##  Pearson's product-moment correlation
##
## data:  d.kf[, as.character(combinaciones[combinaciones$nombre == x,  and d.kf[, as.character(combina
## t = 14.351, df = 206, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6318060 0.7691308
## sample estimates:
##       cor
## 0.7070731
##
##
## $`a y c`
##
##  Pearson's product-moment correlation
##
## data:  d.kf[, as.character(combinaciones[combinaciones$nombre == x,  and d.kf[, as.character(combina
## t = 6.6265, df = 206, p-value = 2.947e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3002499 0.5252582
## sample estimates:
##       cor
## 0.4191695
##
##
## $`b y c`
##
##  Pearson's product-moment correlation
##
## data:  d.kf[, as.character(combinaciones[combinaciones$nombre == x,  and d.kf[, as.character(combina
## t = 7.5083, df = 206, p-value = 1.78e-12
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3495334 0.5640081
## sample estimates:
##       cor
## 0.4635331
```