

PRACTICA_11_S3

Nelson de Jesus Magaña Godinez

19/8/2022

Ejemplo:

El tiempo que tarda un sistema informático en red en ejecutar una instrucción depende del número de usuarios conectados a él. Si no hay usuarios el tiempo es cero. Se tienen registrados los siguientes datos:

No. usuarios	tiempo de ejecucion
10	1.0
15	1.2
20	2.0
20	2.1
25	2.2
30	2.0
30	1.9

REALICE UN ANÁLISIS ESTADÍSTICO.

- 1) Activa tu directorio de trabajo

```
getwd()
```

```
## [1] "C:/Users/pc 1/Desktop/PAQUETE R/PRACTICA_S3"
```

- 2) Crea un nuevo script y llámale “Script11-DatosBivariados4”

- 3) Crea los dos vectores para las dos variables

```
# Número de usuarios = Variable explicativa o independiente
usuarios <- c(10, 15, 20, 20, 25, 30, 30);
usuarios
```

```
## [1] 10 15 20 20 25 30 30
```

```
tiempo <-c(1.0, 1.2, 2.0, 2.1, 2.2, 2.0, 1.9)
tiempo
```

```
## [1] 1.0 1.2 2.0 2.1 2.2 2.0 1.9
```

4) Crea una hoja de datos que tenga como componentes o columnas los dos vectores.

```
Sistema <- data.frame(usuarios = usuarios, tiempo = tiempo)
Sistema
```

```
##   usuarios tiempo
## 1      10    1.0
## 2      15    1.2
## 3      20    2.0
## 4      20    2.1
## 5      25    2.2
## 6      30    2.0
## 7      30    1.9
```

```
# Para editar o ampliar los datos puede utilizar la función fix()
fix(Sistema)
```

5) Guarda la hoja de datos en un archivo.

```
write.table(Sistema, file = "Sistema.txt", append = FALSE, quote = TRUE,
            sep = " ", na = "NA", col.names = TRUE)
```

6) Elimina los objetos almacenados en el área de trabajo (Workspace).

```
ls(); rm(list=ls(all=TRUE)); ls()
```

```
ls()
```

```
## [1] "Sistema" "tiempo"  "usuarios"
```

```
#rm(list = ls(all = TRUE))
ls
```

```
## function (name, pos = -1L, envir = as.environment(pos), all.names = FALSE,
##   pattern, sorted = TRUE)
## {
##   if (!missing(name)) {
##     pos <- tryCatch(name, error = function(e) e)
##     if (inherits(pos, "error")) {
##       name <- substitute(name)
##       if (!is.character(name))
##         name <- deparse(name)
##       warning(gettextf("%s converted to character string",
##         sQuote(name)), domain = NA)
##       pos <- name
##     }
##   }
##   all.names <- .Internal(ls(envir, all.names, sorted))
```

```
##   if (!missing(pattern)) {
##       if ((ll <- length(grep("[", pattern, fixed = TRUE))) &&
##           ll != length(grep("]", pattern, fixed = TRUE))) {
##           if (pattern == "[") {
##               pattern <- "\\["
##               warning("replaced regular expression pattern '[' by '\\\\['")
##           }
##           else if (length(grep("[^\\\\]\\\\[<-", pattern))) {
##               pattern <- sub("\\[<-", "\\\\\\\\[<-", pattern)
##               warning("replaced '['<' by '\\\\\\\\[<' in regular expression pattern")
##           }
##       }
##       grep(pattern, all.names, value = TRUE)
##   }
##   else all.names
## }
## <bytecode: 0x000001ebb6cc3970>
## <environment: namespace:base>
```

7) Recupera la hoja de datos.

```
sistema <-read.table("sistema.txt", header = TRUE);
sistema
```

```
##   usuarios tiempo
## 1         10    1.0
## 2         15    1.2
## 3         20    2.0
## 4         20    2.1
## 5         25    2.2
## 6         30    2.0
## 7         30    1.9
```

8) Conecta la hoja de datos a la segunda ruta o lista de búsqueda.

```
attach(Sistema, pos = 2)
```

```
## The following objects are masked _by_ .GlobalEnv:
##
##   tiempo, usuarios
```

```
search()
```

```
## [1] ".GlobalEnv"      "Sistema"          "package:stats"
## [4] "package:graphics" "package:grDevices" "package:utils"
## [7] "package:datasets" "package:methods"   "Autoloads"
## [10] "package:base"
```

9) Muestra un resumen de principales estadísticos de las variables.

```
summary(Sistema)
```

```
##      usuarios      tiempo
## Min.   :10.00   Min.   :1.000
## 1st Qu.:17.50   1st Qu.:1.550
## Median :20.00   Median :2.000
## Mean   :21.43   Mean   :1.771
## 3rd Qu.:27.50   3rd Qu.:2.050
## Max.   :30.00   Max.   :2.200
```

```
cov(Sistema) # Matriz de covarianza
```

```
##      usuarios      tiempo
## usuarios 55.952381 2.714286
## tiempo   2.714286 0.222381
```

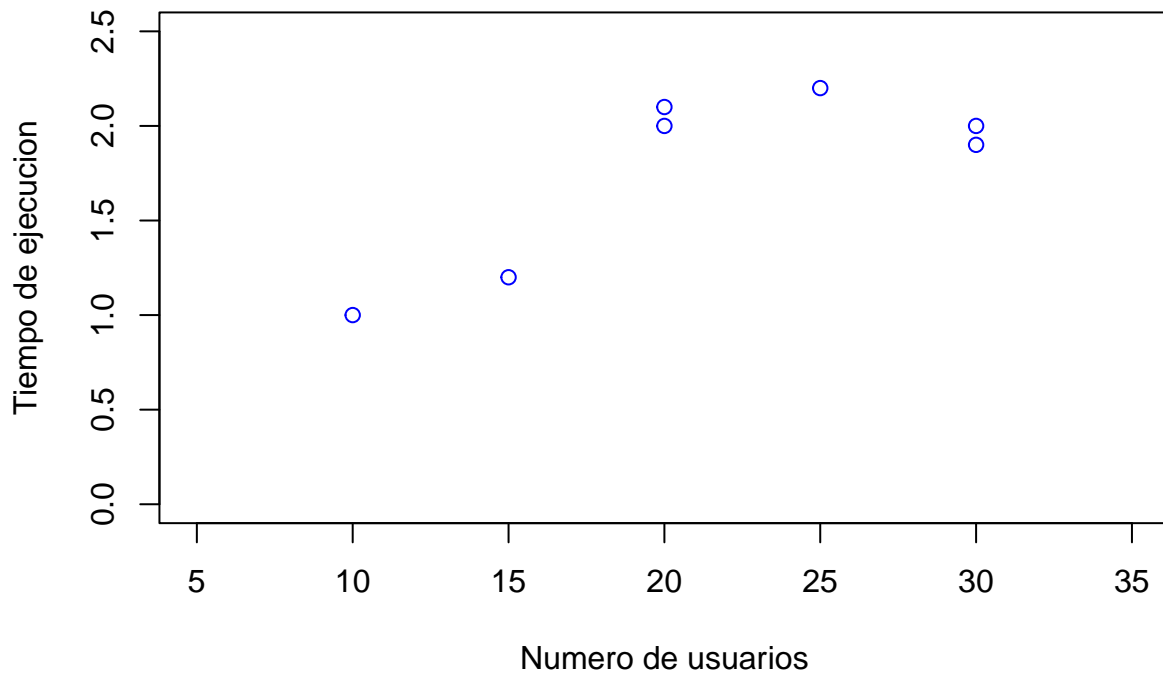
```
cor(Sistema, use = "all.obs", method = "pearson") # Matriz de correlaciones
```

```
##      usuarios      tiempo
## usuarios 1.0000000 0.7694803
## tiempo   0.7694803 1.0000000
```

10) Elabora un gráfico de dispersión para analizar alguna relación entre las variables.

```
plot(usuarios, tiempo, xlim = c(5, 35), ylim = c(0.0, 2.5), type = "p",
     pch = 1, col = "blue",
     main = "Grafico de dispersion (Usuarios, Tiempo)",
     xlab = "Numero de usuarios", ylab = "Tiempo de ejecucion")
```

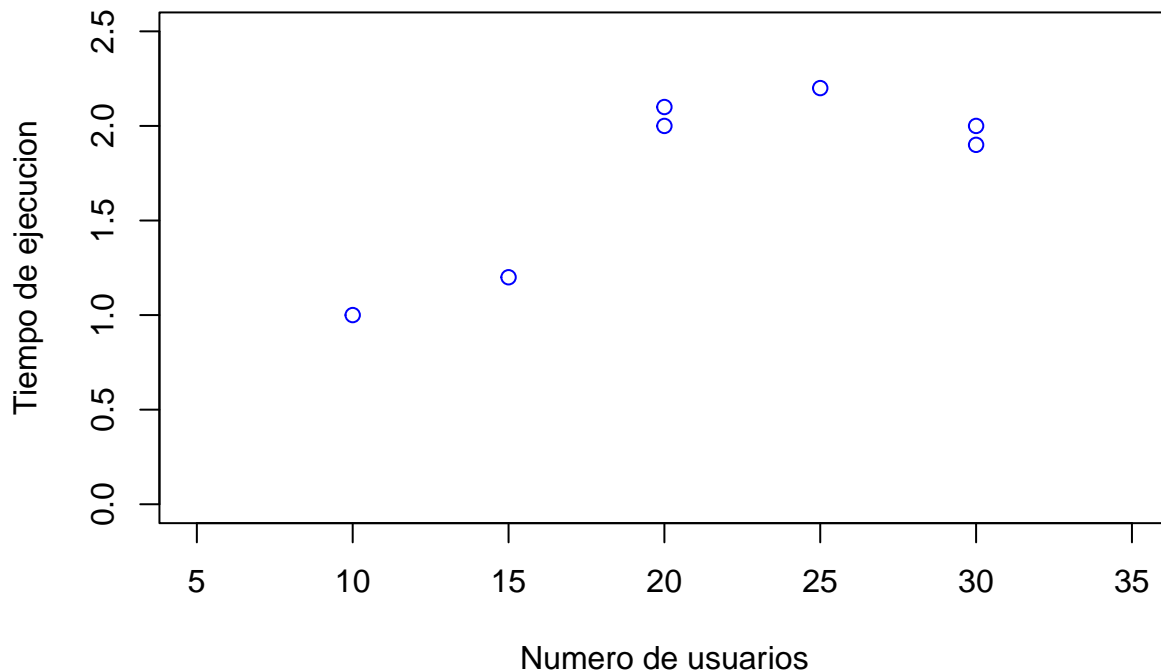
Grafico de dispersion (Usuarios, Tiempo)



11) Para identificar un punto arbitrario, se procede de la siguiente manera:

```
plot(usuarios, tiempo, xlim = c(5, 35), ylim = c(0.0, 2.5), type = "p",  
     pch = 1, col = "blue",  
     main = "Grafico de dispersion (Usuarios, Tiempo)",  
     xlab = "Numero de usuarios", ylab = "Tiempo de ejecucion")  
#Sin cerrar la ventana del gráfico anterior, ejecuta la siguiente instrucción  
identify(usuarios, tiempo, n=1) # n=1 indica que solamente será un punto seleccionado
```

Grafico de dispersion (Usuarios, Tiempo)



```
## integer(0)
```

*# Y luego selecciona un punto en el gráfico haciendo clic con el ratón.
Esto es útil para identificar puntos que podrían ser atípicos.
Deberá aparecer en la R-Console el índice que corresponde a este punto.*

12) Aplica la función `lm()` para encontrar el modelo lineal que se ajusta a los datos.

```
reg.Y.X <- lm(tiempo~ -1+usuarios, Sistema, na.action = NULL,  
             method = "qr", model = TRUE)  
#-1 indica que no se toma en cuenta la constante en el modelo.  
summary(reg.Y.X)
```

```
##  
## Call:  
## lm(formula = tiempo ~ -1 + usuarios, data = Sistema, na.action = NULL,  
##     method = "qr", model = TRUE)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -0.4831 -0.1873  0.2056  0.3127  0.5113   
##  
## Coefficients:
```

```
##           Estimate Std. Error t value Pr(>|t|)
## usuarios 0.079437    0.006496  12.23 1.82e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3871 on 6 degrees of freedom
## Multiple R-squared:  0.9614, Adjusted R-squared:  0.955
## F-statistic: 149.5 on 1 and 6 DF, p-value: 1.821e-05
```

Note que es necesaria la instrucción anterior para poder visualizar los resultados más sobresalientes de la regresión encontrada. Nos muestra la estimación de los parámetros junto con su significancia, el coeficiente de determinación.

13) Agrega la recta de regresión al gráfico de dispersión.

```
#abline(reg.Y.X)
# Observación: Alternativamente si quiere una recta más "exacta" use:
line(usuarios, 0.09437*usuarios)
```

```
##
## Call:
## line(usuarios, 0.09437 * usuarios)
##
## Coefficients:
## [1] 4.441e-16 9.437e-02
```

14) Efectúa un análisis de variabilidad del modelo o descomposición de la varianza.

```
reg.anova <- anova(reg.Y.X)
reg.anova
```

```
## Analysis of Variance Table
##
## Response: tiempo
##           Df Sum Sq Mean Sq F value    Pr(>F)
## usuarios   1 22.4011 22.4011  149.53 1.821e-05 ***
## Residuals   6  0.8989  0.1498
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Se obtiene un P-valor menor que 0.05 por lo tanto la varianzas no presentan homogeneidad entre los grupos.