Bioestadística Aplicada con R - Manual Práctico

Christian Jair Cerazo Huapaya

2026-10-01

Table of contents

Prefacio

Este es el prefacio del documento quarto book.

1 Introducción

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

```
1 + 1
```

```
[1] 2
```

2 Resumen

Este manual ofrece una introducción práctica a la bioestadística con R, orientado a estudiantes e investigadores en ciencias biológicas. A lo largo de sus capítulos, se exploran conceptos clave como estadística descriptiva, pruebas de hipótesis, regresión y visualización de datos, todo con ejemplos reproducibles en Quarto.

El enfoque está centrado en la aplicación real, con datos biológicos y herramientas modernas como ggplot2, dplyr y Quarto para documentación científica.

```
1 + 1
```

```
[1] 2
```

Referencias

Knuth, D. E. (1984). Literate programming. *Comput. J.*, *27*(2), 97–111. https://doi.org/10.1093/comjnl/27.2.97

3 prueba ejercicio

```
#| standalone: true
#| viewerHeight: 600
library(shiny)
library(bslib)

# Define UI for app that draws a histogram ----
ui <- page_sidebar(
  sidebar = sidebar(open = "open",
    numericInput("n", "Sample count", 100),
    checkboxInput("pause", "Pause", FALSE),
  ),
  plotOutput("plot", width=1100)
)

server <- function(input, output, session) {
  data <- reactive({
    input$resample
    if (!isTRUE(input$pause)) {
      invalidateLater(1000)
    }
    rnorm(input$n)
  })

  output$plot <- renderPlot({
    hist(data(),
      breaks = 40,
      xlim = c(-2, 2),
      ylim = c(0, 1),
      lty = "blank",
      xlab = "value",
      freq = FALSE,
      main = ""
    )
```

```
    x <- seq(from = -2, to = 2, length.out = 500)
    y <- dnorm(x)
    lines(x, y, lwd=1.5)

    lwd <- 5
    abline(v=0, col="red", lwd=lwd, lty=2)
    abline(v=mean(data()), col="blue", lwd=lwd, lty=1)

    legend(legend = c("Normal", "Mean", "Sample mean"),
      col = c("black", "red", "blue"),
      lty = c(1, 2, 1),
      lwd = c(1, lwd, lwd),
      x = 1,
      y = 0.9
    )
  }, res=140)
}

# Create Shiny app ----
shinyApp(ui = ui, server = server)
```

4 Códigos R para análisis estadístico

4.1 Funciones estadísticas básicas

4.1.1 Aritméticos - básicos

- `help()` o `?función` *Ayuda sobre una función*
- `sum()` *Suma de elementos*
- `min()` *Mínimo valor*
- `max()` *Máximo valor*
- `leigth()` *Conteo de elementos*
- `read.table()` *Lee archivos y los esquematiza en una tabla*
- `sample()` *Genera números aleatorios*

4.1.2 Medidas de tendencia central

- `mean()` *Media*
- `median()` *Mediana*
- `quantile()` *Cuartiles/percentiles*

4.1.3 Medidas de dispersión

- `sd()` *Desviación estándar*
- `var()` *Varianza*
- `IQR()` *Rango intercuartílico*
- `range()` *Rango total*
- `fivenum()` *Mínimo, primer cuartil, mediana, tercer cuartil y máximo*

4.1.4 Distribuciones

- `dnorm()` *Densidad normal*
- `ppois()` *Probabilidad acumulada Poisson*
- `rbinom()` *Generador binomial*
- ‘

4.1.5 Pruebas de hipótesis

- `t.test()` *T-Student*
- `cor.test()` *Correlación*
- `chisq.test()` *Chi-cuadrado*
- `aov()` *ANOVA*

4.1.6 Modelización

- `lm()` *Regresión lineal*
- `glm()` *Modelos lineales generalizados*
- `nls()` *Mínimos cuadrados no lineales*

4.1.7 Gráficos

- `boxplot()` *Diagrama de caja y bigotes*
- `hist()` *Histograma*

4.2 Diseño muestral

Se sabe que el consumo de alcohol durante el embarazo puede dañar al feto. Para estudiar este fenómeno, se asignarán 20 ratones preñados a tres grupos de tratamiento. El grupo 1 (diez ratones) no recibirá alcohol y el grupo 2 (cinco ratones) recibirá una dosis elevada. Prepara una asignación completamente aleatoria. Use R para etiquetar y aleatorizar.

Colocar un `set.seed()` para que la aleatorización no se realiza cada vez que se ejecuta el código

```
set.seed(122)
ratones <- c(1:20)
trats <- c(rep("control",10), rep("dosis media", 5), rep("dosis alta", 5))
diseño1 <- data.frame("individuos"= sample(ratones), "tratamientos"= sample(trats))
diseño1 <- diseño1[order(diseño1$tratamientos),]
diseño1
```

```
   individuos tratamientos
1          16      control
2           8      control
3           9      control
6           7      control
```

```
7          18      control
8           2      control
9          20      control
10          3      control
15          4      control
17         19      control
4          15   dosis alta
16         11   dosis alta
18          6   dosis alta
19         17   dosis alta
20          1   dosis alta
5          12  dosis media
11         10  dosis media
12          5  dosis media
13         13  dosis media
14         14  dosis media
```

4.3 Analisis de varianza ANOVA

los datos de la tabla lo planteamos en forma de codigo, son tres columnas (3 muestras), la primera tiene 4 datos, la segunda tiene 3, y la tercera 4

	Muestra		
	1	**2**	**3**
	48	40	39
	39	48	30
	42	44	32
	43		35
Promedio	43,00	44,00	34,00
DE	3,74	4,00	3,92

Para resolverlo primero llamamos a `library(car)` para usar la función `leveneTest()`, crearemos un objeto (**data1**) donde crearemos nuestro **data frame**. La función `factor()` para generar variables explicativas continuas y las repeticiones que tendrán por grupo, el cual se asignará con otro vector

```r
library(car)
```

```
Cargando paquete requerido: carData
```

```r
data1 <- data.frame("muestras" = factor(rep(c(1, 2, 3), c(4, 3, 4))),
                    "medidas" = c(48, 39, 42, 43, 40, 48, 44, 39, 30, 32, 35))
data1
```

```
   muestras medidas
1         1      48
2         1      39
3         1      42
4         1      43
5         2      40
6         2      48
7         2      44
8         3      39
9         3      30
10        3      32
11        3      35
```

luego de tener los datos, hacemos las pruebas previas de normalidad (Shapiro) y homocedasticidad (Levene test)

```r
tapply(data1$medidas, data1$muestras, shapiro.test)
```

```
$`1`

    Shapiro-Wilk normality test

data:  X[[i]]
W = 0.96058, p-value = 0.7825


$`2`

    Shapiro-Wilk normality test

data:  X[[i]]
W = 1, p-value = 1
```

```
$`3`

    Shapiro-Wilk normality test

data:  X[[i]]
W = 0.97133, p-value = 0.8497
```

los valores de la prueba shapiro tienen un P>0.05 por lo tanto no rechazan hipótesis nula.

prueba de homocedasticidad

```
#el simbolo "~" se hace con "Ctrl + Alt + +"
leveneTest(medidas ~ muestras, data = data1)
```

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  2  0.0519 0.9497
       8
```

la prueba no rechaza la hipotesis nula, por lo tanto si hay homocedasticidad y podemos proceder con el analisis conveniente: en este caso ANOVA de un factor.

Prueba ANOVA

```
#primero creamos un modelo y luego la función anova hacia ese modelo
#"~" se hace con "Ctrl + Alt + +"
modelo1 <- lm(medidas ~ muestras, data = data1)
modelo1
```

```
Call:
lm(formula = medidas ~ muestras, data = data1)

Coefficients:
(Intercept)    muestras2    muestras3
         43            1           -9
```

```
anova(modelo1)
```

```
Analysis of Variance Table

Response: medidas
          Df Sum Sq Mean Sq F value  Pr(>F)
muestras   2    228     114     7.6 0.01414 *
Residuals  8    120      15
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

el anova indica diferencia significativa de las medias de los grupos debido a que P=0.014 < 0.05

como prueba Post-hoc realizaremos la prueba Tukey

```
#primero cambiaremos la estructura del objeto "modelo1"
anova1 <- aov(modelo1)
anova1
```

```
Call:
   aov(formula = modelo1)

Terms:
                muestras Residuals
Sum of Squares       228       120
Deg. of Freedom        2         8

Residual standard error: 3.872983
Estimated effects may be unbalanced
```

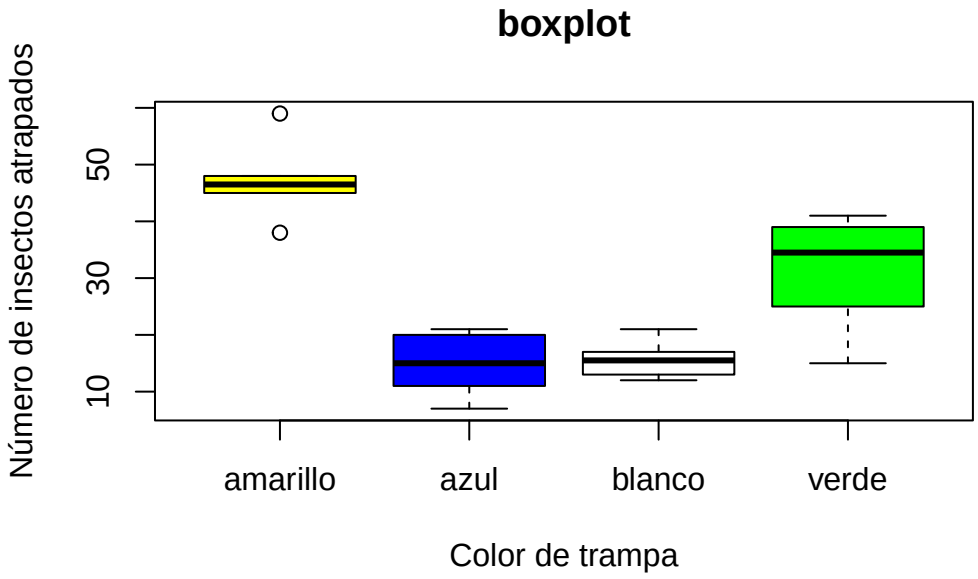
```
TukeyHSD(anova1)
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = modelo1)

$muestras
    diff        lwr       upr     p adj
2-1    1  -7.452434  9.452434 0.9394135
3-1   -9 -16.825433 -1.174567 0.0267449
3-2  -10 -18.452434 -1.547566 0.0233496
```

en las comparaciones 3-1 y 3-2 si hay diferencias significativas, en la comparacion 2-1 no podemos afirmar diferencias significativas

4.4 Prueba ANOVA variable explicativa cualitativa:

```
library(car)
cualitativo <- data.frame("color" = as.factor(c(rep(c("azul", "verde", "blanco", "amarillo")
boxplot(atrapado ~ color, data = cualitativo, col = c("yellow", "blue", "white", "green"),yla
```




```
tapply(cualitativo$atrapado, cualitativo$color, shapiro.test)
```

```
$amarillo

    Shapiro-Wilk normality test

data:  X[[i]]
W = 0.90241, p-value = 0.3883


$azul
```

```
    Shapiro-Wilk normality test

data:  X[[i]]
W = 0.95913, p-value = 0.813


$blanco

    Shapiro-Wilk normality test

data:  X[[i]]
W = 0.9302, p-value = 0.5817


$verde

    Shapiro-Wilk normality test

data:  X[[i]]
W = 0.90483, p-value = 0.4033
```

```r
leveneTest(atrapado ~ color, data = cualitativo)
```

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  3  1.2875 0.3059
      20
```

```r
modelo3 <- lm(atrapado ~ color, data = cualitativo)
summary(modelo3)
```

```

Call:
lm(formula = atrapado ~ color, data = cualitativo)

Residuals:
     Min       1Q   Median       3Q      Max
-16.5000  -2.9167   0.1667   5.2083  11.8333

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   47.167      2.770  17.030 2.27e-13 ***
colorazul    -32.333      3.917  -8.255 7.16e-08 ***
colorblanco  -31.500      3.917  -8.042 1.07e-07 ***
colorverde   -15.667      3.917  -4.000 0.000704 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.784 on 20 degrees of freedom
Multiple R-squared:  0.8209,    Adjusted R-squared:  0.794
F-statistic: 30.55 on 3 and 20 DF,  p-value: 1.151e-07
```

```
anova(modelo3)
```

```
Analysis of Variance Table

Response: atrapado
          Df Sum Sq Mean Sq F value    Pr(>F)
color      3 4218.5 1406.15  30.552 1.151e-07 ***
Residuals 20  920.5   46.03
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova3 <- aov(modelo3)
TukeyHSD(anova3)
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = modelo3)

$color
                       diff        lwr       upr     p adj
azul-amarillo   -32.3333333 -43.296330 -21.37034 0.0000004
blanco-amarillo -31.5000000 -42.462996 -20.53700 0.0000006
verde-amarillo  -15.6666667 -26.629663  -4.70367 0.0036170
blanco-azul       0.8333333 -10.129663  11.79633 0.9964823
verde-azul       16.6666667   5.703670  27.62966 0.0020222
verde-blanco     15.8333333   4.870337  26.79633 0.0032835
```

4.5 Regresion lineal

Ejercicio: En un estudio sobre la síntesis de proteínas en el ovocito (célula huevo en desarrollo) de la rana Xenopus laevis, un biólogo inyectó leucina radiactiva en ovocitos individuales. En distintos momentos tras la inyección, realizó mediciones de radiactividad y calculó la cantidad de leucinaque se había incorporado a la proteína. Los resultados se muestran en la tabla adjunta; cada valor de leucina es el contenido de leucina marcada en dos ovocitos. Todos los ovocitos procedían de la misma hembra.

	Tiempo	Leucina
	0	0.02
	10	0.25
	20	0.54
	30	0.69
	40	1.07
	50	1.50
	60	1.74
Promedio	30.00	0.830
DE	21.60	0.637
SS(resid)=0.035225		

```
#creamos la data
data2 <- data.frame("tiempo" = c(0, 10, 20, 30, 40, 50, 60), "leucina" = c(0.02, 0.25, 0.54,
data2
```

```
  tiempo leucina
1      0    0.02
2     10    0.25
3     20    0.54
4     30    0.69
5     40    1.07
6     50    1.50
7     60    1.74
```

añadimos columnas

```
# "^" = Alt Gr + {}
data2$"xi-xm" <- data2$tiempo-mean(data2$tiempo)
data2$"yi-ym" <- data2$leucina-mean(data2$leucina)
data2$"(xi-xm)^2" <- data2$`xi-xm`^2
data2$"(yi-ym)^2" <- data2$`yi-ym`^2
data2$"(xi-xm)(yi-ym)" <- data2$`xi-xm`*data2$`yi-ym`
data2
```

```
  tiempo leucina xi-xm yi-ym (xi-xm)^2 (yi-ym)^2 (xi-xm)(yi-ym)
1      0    0.02   -30 -0.81       900    0.6561           24.3
2     10    0.25   -20 -0.58       400    0.3364           11.6
3     20    0.54   -10 -0.29       100    0.0841            2.9
4     30    0.69     0 -0.14         0    0.0196            0.0
5     40    1.07    10  0.24       100    0.0576            2.4
6     50    1.50    20  0.67       400    0.4489           13.4
7     60    1.74    30  0.91       900    0.8281           27.3
```

valores total para la tabla

```
sum(data2$`(yi-ym)(xi-xm)`)
```

```
[1] 0
```

```
sum(data2$`yi-ym`)
```

```
[1] 3.330669e-16
```

```
sum(data2$`xi-xm`)
```

```
[1] 0
```

```
sum(data2$`(xi-xm)^2`)
```

```
[1] 2800
```

```
sum(data2$`(yi-ym)^2`)
```

```
[1] 2.4308
```

creamos el modelo de regresion lineal

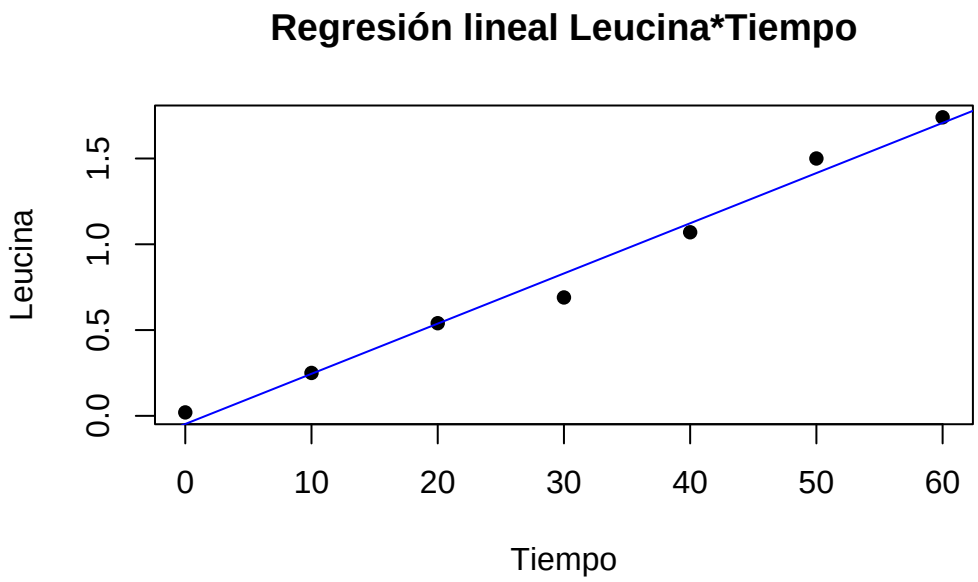
```
modelo2 <- lm(leucina ~ tiempo, data = data2)
modelo2
```

```
Call:
lm(formula = leucina ~ tiempo, data = data2)

Coefficients:
(Intercept)       tiempo
   -0.04750      0.02925
```

se observa el intercepto: -0.048 y el coeficiente de x: 0.029 siendo el modelo: y=0.029x - 0.048 (y:leucina; x:tiempo)

graficamos el modelo

```
plot(data2$tiempo, data2$leucina, pch= 16, xlab = "Tiempo", ylab = "Leucina", main = "Regresi
abline(modelo2, col = "blue")
```



Evaluamos mas parámetros con la función `summary()`

```r
summary(modelo2)
```

```
Call:
lm(formula = leucina ~ tiempo, data = data2)

Residuals:
      1       2       3       4       5       6       7
 0.0675  0.0050  0.0025 -0.1400 -0.0525  0.0850  0.0325

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.047500   0.057192  -0.831    0.444
tiempo       0.029250   0.001586  18.440 8.63e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.08393 on 5 degrees of freedom
Multiple R-squared:  0.9855,    Adjusted R-squared:  0.9826
F-statistic:   340 on 1 and 5 DF,  p-value: 8.628e-06
```

los residuos son las diferencias entre los valores observados y los valores predichos por el modelo. Los coeficientes: intercepto es -0.048, y para el tiempo es 0.029, significancia se muestra en la ultima columna. siendo P<0.05. Multiple R-squared: 0.9855, nos indica el coeficiente de determinacion, nos indica cuánta variabilidad en la variable dependiente se explica por las variables independientes.

4.6 Correlación

empleando los datos del ejercicio de regresión:

```r
#manualmente podemos hallar r (coerficiente de correlacion):
r <- sum(data2$`(xi-xm)(yi-ym)`)/sqrt(sum(data2$`(xi-xm)^2`)*sum(data2$`(yi-ym)^2`))
r
```

```
[1] 0.992728
```

```r
#usando funcion:
cor(data2$leucina, data2$tiempo)
```

```
[1] 0.992728
```

```
#o
cor(data2$tiempo, data2$leucina)
```

```
[1] 0.992728
```

grafica: añadimos la correlación

```
plot(data2$tiempo, data2$leucina, pch= 16, xlab = "Tiempo", ylab = "Leucina", main = "Regresi
abline(modelo2, col = "blue")
text(10, 1, "r = 0.992728")
```