

Programación y Uso de Librerías en R: Herramientas de Análisis y Visualización de Datos en la Enseñanza y la Investigación Científica

Juan Luis Peñaloza Figueroa
Universidad Complutense de Madrid

Milagros Dones Tacero
Universidad Autónoma de Madrid

Carmen Gladys Vargas Pérez
Universidad Complutense de Madrid

AÑO: 2025

SCRIPT_2: CAPÍTULO IV. VARIABLES, MANEJO Y LIMPIEZA DE DATOS Y BASES DE DATOS

❑ VARIABLE NOMINAL

```
> factor_nominal <- factor(rep(c("Ford", "Seat", "Renault"), 10))
> levels(factor_nominal)
> factor_sexo <- factor(sexo)
> factor_sexo
```

❑ Variables categóricas ordinales.

```
> pct <- c("alto", "bajo", "bajo", "medio")
> fpct = factor(pct)
> table(fpct)
```

IV.5. GENERACIÓN DE DATOS, MUESTRAS Y MUESTREO

Función secuencia – seq()

```
> seq(from=1, to=1, by, length.out)
```

Función repetir – rep()

```
> rep(x, times=1, length.out=NA, each=1)
```

Función sort()

```
> x <- c(2, 3, 6, 4, 9, 5)
> sort(x)
```

Generación de muestras

#install.packages("dplyr")

```
> library(dplyr)
```

```
> sample_n(mtcars, 5)
```

```
> muestra3 <- crime %>% sample_n(size=n, replace=F)
```

```
> head(muestra3)
```

Muestreo Aleatorio Simple sin Repetición

```
> data(iris)
> indice <- sample(1:nrow(iris), 60)
> iris.muestral <- iris[indice,]
> head(iris.muestral, 4)
```

Muestreo Aleatorio Simple con Repetición

```
> indice1 <- sample(1:nrow(iris), 60, replace = TRUE)
```

```

> iris.muestra2<-iris[indice1,]
> head(iris.muestra2,4)

# Muestreo Estratificado con o sin Reemplazo.
> install.packages("sampling")
> library(sampling)
> stratos<-strata(iris, stratanames=c("Species"), size=
c(20,20,20),method="srswor")
> iris.muestra3<-getdata(iris,stratos)
> head(iris.muestra3)
> summary(iris.muestra3)

#Muestreo Sistemático
> sys.sample = function(N, n) {
k = ceiling(N/n) r = sample(1:k,
1)
sys.samp = seq(r, r + k * (n - 1), k)
}
> systematic.index <- sys.sample(nrow(iris), nrow(iris) * 0.75)
> summary(iris[systematic.index, ])

# Muestreo de bola de nieve
> personas_iniciales <- c("Persona1", "Persona2")
> bola_de_nieve <- function(personas, num_iteraciones)
{
  muestra <- personas
  for (i in 1:num_iteraciones)
  {
    nuevos_participantes <- sapply(muestra, function(x) paste(x, "_Ref",
i, sep = ""))
    muestra <- unique(c(muestra, nuevos_participantes))
  }
  return(muestra)
}

```

IV.6. TAMAÑOS DE MUESTRA

```

> Z=1.645; p=0.5; q=1-p; N=10000; e=0.04
> n=(Z^2*N*p*q) / (e^2*(N-1)+Z^2*p*q)
> n

```

IV.9. CÁLCULO DE INTERVALOS DE CONFIANZA

```

> install.packages("DescTools")
> library(DescTools)

#Intervalos de confianza al 95% de confianza
> ICTamHog<- MeanCI(x=muestra$TamHog, trim = 0,conf.level =
0.95, na.rm = FALSE)
> ICTamHog

#Media del total de la encuesta
> mean(df$TamHog)

#Diferencia relativa
> difR<-paste0(abs(round((ICTamHog[1]-mean(df$TamHog))/mean
(df$TamHog),3))*100,"%")
>difR

# Modificación de un data frame
> dat

```

```
> dat[3, 3] <- "Granada"
> dat
```

```
#Añadir o eliminar nuevas columna
> dat$Peso <- c(55.2, 72.1, 85)
> dat
```

```
#Añadir una fila
t <- data.frame(Nombre = "Marta", Edad = 18, Ciudad = "Jaén", Peso = 67)
dat <- rbind(dat, t)
dat
```

IV.13. DISCRETIZACIÓN DE VECTORES

```
# Creamos una muestra de edades
> sample(x, size, replace = FALSE, prob = NULL)
> set.seed(1)
> edades<- sample(1:30,40,T)
> edades

#Discretización mediante un gráfico de barras
> barplot(table(edades),cex.names = 0.8,main = "Personas por etapas de vida",col=terrain.colors(4, alpha=0.8), font=2)
```

IV. DETECCIÓN Y TRATAMIENTO DE OUTLIERS

```
#Detección de outliers
función boxplot() para detectar outliers
>read.csv(https://faculty.washington.edu/heagerty/Books/Biostatistics/DATA/ozone.csv)
> head(ozonol,4) .
> str(ozonol)
>g_caja<-boxplot(ozonol$ozone_reading,col="skyblue", frame.plot =FALSE)
```

```
# Cómo eliminar los outliers
> ozonol<-ozonol[!(ozonol$ozone_reading%in%g_caja$out),]
```

```
#Representación gráfica de la normalización y outliers
> pch_site<-as.numeric(abs(z)>3)
> plot(x, pch=pch_site, col = gray((1:4)/6)[abs(z)])
> valcol <- (x + abs(min(x)))/max(x + abs(min(x)))
> plot(x, pch = 16, col = rgb(0, 0, 1))
```

```
#Prueba de Tukey para Identificar Outliers
```

```
> x<-sample(5:16,100,replace=T)
> x[c(10,21,33)]<- c(21,31,40)
> x
> q1<-quantile(x, 0.25)
> q1

# Prueba de tukey
> p_tukey<-x<(q1-1.5*IQR(x)) |x>(q3+1.5*IQR(x))
```

```
#Distancia de Cooks e Identificación de Outliers
```

```
> mod1<-lm(x~1) > mod1
> dcooks<-cooks.distance(mod1)
> plot(dcooks, pch=8, cex=1)
```

```

> abline(h=4*mean(dcooks,na.rm=T),col="red")
> text(x=1:length(dcooks)+1,y=dcooks,labels=ifelse(dcooks>4*mean(dcooks,na.rm=T),names(dcooks),""),col="red")

#Tests Estadísticos para Detectar Valores aAtípicos
> install.packages("ggplot2")
> library(ggplot2)
> data(mpg)
> head(mpg)

#Test de Grubbs para el valor más alto (por defecto)
> Install.packages("outliers")
> library(outliers)
> test <- grubbs.test(mpg$hwy)
> test

#Tests de Grubbs para el valor más bajo (opposite = T)
> test <- grubbs.test(mpg$hwy, opposite = TRUE)
> test

#Test de Dixon para el valor más bajo del subconjunto de datos "subdat"
> test <- dixon.test(subdat$hwy)
> test

#Test de Dixon para el valor más alto del conjunto de datos
> test <- dixon.test(subdat$hwy, opposite = TRUE)
> test

#Test de Rosner
> install.packages("EnvStats")
> library(EnvStats)
> testr <- rosnerTest(mpg$hwy,k = 3)
> testr

#Utilizar is.na() para identificar valores perdidos
> vector_with_na<-c(4, 5, NA, 7)
> is.na(vector_with_na)

# Imputación por medio de los estadísticos Media, Mediana o Moda
> data(iris)
> dNa<-iris
> install.packages("missForest")
> library(missForest)
# Generamos datos perdidos en la base de datos "dNa"
> dfA<-prodNA(dNa,noNA=0.1)
> summary(dfA)
> dfA2<-dfA %>% select(-Species)
> head(dfA2)
> dfA2$Sepal.Length[is.na(dfA2$Sepal.Length)]<-mean(dfA2$Sepal.Length,na.rm=T)
> dfA2$Sepal.Width[is.na(dfA2$Sepal.Width)]<-median(dfA2$Sepal.Width,na.rm=T)
> dfA2$Petal.Length[is.na(dfA2$Petal.Length)]<-median(dfA2$Petal.Length,na.rm=T)
> dfA2$Petal.Width[is.na(dfA2$Petal.Width)]<-mean(dfA2$Petal.Width,na.rm=T)
> summary(dfA2)

```

Tabla de frecuencia de una vía

```

# Generación del vector fuma y sexo
> fuma <- c('Frecuente', 'Nunca', 'A veces', 'A veces', 'A veces',
'Nunca', 'Frecuente', NA, 'Frecuente', NA, 'Hola', 'Nunca', 'Hola',
'Frecuente', 'Nunca')
> dfu <- table(fuma)
> dfu
> sexo <- c('Hombre', 'Hombre', 'Hombre', NA, 'Mujer', NA, 'Mujer',
'Mujer', 'Mujer', 'Hombre', 'Mujer', 'Hombre', NA, 'Mujer', 'Mujer')
> tabla3<-table(sexo,fuma)
> tabla3

#Función prop.table (Frecuencias relativas)
> tabla4<-prop.table(tabla3)
> tabla4
> tabla6<-prop.table(tabla3, margin=1)
> tabla6
> addmargins(tabla4)

```

-----000-----