

Sesión 01: Introducción a R

Richard Pérez Palma Ponce

2022



Facultad de
Ingeniería
Económica,
Estadística y
Ciencias
Sociales

Centro de
Formación
Continua

Programa de Especialización en
ECONOMETRÍA APLICADA

- 1 Entorno de R y R Studio
- 2 Creación de Objetos
- 3 Bucles y Condicionales
- 4 Creación de Funciones
- 5 Manejo de datos
- 6 Gráficos en R
- 7 Manejo de Paquetes en R

- R 4.0.2 for Windows:
<https://cran.r-project.org/bin/windows/base/>.
- R Studio
<https://www.rstudio.com/products/rstudio/download/>.

- Lista de objetos en R

```
ls()
```

- Eliminar objetos

```
rm()
```

- Ver el directorio actual

```
getwd()
```

- Cambiar directorio actual

```
setwd("C:/Users/rpere/Desktop")
```

- Lista de archivos

```
list.files()
```

Se pueden identificar los siguientes tipos de datos en R:

```
condicion <- TRUE  
class(condicion)
```

```
## [1] "logical"
```

```
edad <- 23L  
class(edad)
```

```
## [1] "integer"
```

```
talla <- 1.76  
class(talla)
```

```
## [1] "numeric"
```

```
nombre <- 'Manuel'  
class(nombre)
```

```
## [1] "character"
```

Los nombres de los objetos presentan las siguientes características:

- Los nombres pueden ser combinaciones de letras, números y el punto.
- No pueden empezar con un número.
- Hace diferencias entre mayúsculas y minúsculas. (Case Sensitive)
- Hay nombres reservados. (function, for, if,...)

```
pbi <- rnorm(100, mean = 2, sd = 2)
grupo.profesion<- sample(c("A","B","C"), 10,
                        replace = TRUE)
```

- Utilizando operador igual (usado para asignar dentro de funciones)

```
fecha1 = 20180901
```

- Utilizando operador menor (usado en la creación de objetos)

```
fecha2 <- 20180901
```

- utilizando operador mayor

```
20180901 -> fecha3
```

Outline

- 1 Entorno de R y R Studio
- 2 Creación de Objetos
- 3 Bucles y Condicionales
- 4 Creación de Funciones
- 5 Manejo de datos
- 6 Gráficos en R
- 7 Manejo de Paquetes en R

Objetos en R: Vectores y Matrices

- Creación de Vectores

```
v_datos <- c("Juan","Jose")  
print(v_datos)  
## [1] "Juan" "Jose"
```

- Creación de Matrices

```
m_squares <- matrix(c(1,4,9,16),nrow = 2,ncol = 2)  
print(m_squares)  
##      [,1] [,2]  
## [1,]    1    9  
## [2,]    4   16
```

Objetos en R: Arrays y Factors

- Creación de Arrays

```
ar_colores <- array(data = c("azul","verde"),dim = c(1,3,2))
print(ar_colores)
## , , 1
##
##      [,1] [,2] [,3]
## [1,] "azul" "verde" "azul"
##
## , , 2
##
##      [,1] [,2] [,3]
## [1,] "verde" "azul" "verde"
```

- Creación de Factores

```
v_profesion <- sample(x = c("Economista","Abogado","Contador"),
                      size = 10, replace = TRUE)
v_profesion_factor <- factor(v_profesion)
```

```
print(v_profesion_factor)
```

```
## [1] Economista Contador Contador Contador Abogado Economista
## [7] Economista Contador Economista Contador
## Levels: Abogado Contador Economista
```

Objetos en R: Data Frame

- Creación de Data Frame

```
v_nom  <- c("Jesus","Juan","Manuel")
v_edad <- c(25,30,40)
v_prof <- c("Economista","Abogado","Contador")

df_pobl <- data.frame(v_nom,v_edad,v_prof)
print(df_pobl)

##      v_nom v_edad      v_prof
## 1  Jesus      25 Economista
## 2   Juan      30   Abogado
## 3 Manuel      40   Contador

#LLamar variables dentro del df
df_pobl$v_prof
## [1] "Economista" "Abogado"      "Contador"
```

- Creación de Lista con múltiples objetos:

```
l_objetos <- list(nombres = v_nom,
                 matriz = m_squares,
                 dataframe = df_pobl)

print(l_objetos)

## $nombres
## [1] "Jesus"  "Juan"   "Manuel"
##
## $matriz
##      [,1] [,2]
## [1,]    1    9
## [2,]    4   16
##
## $dataframe
##      v_nom v_edad    v_prof
## 1  Jesus     25 Economista
## 2   Juan     30  Abogado
## 3 Manuel     40  Contador
```

Operaciones con Matrices

```
# Operaciones con matrices
v_a <- c(10,22,30)
v_b <- c(3,4,7)
v_c <- c(8,9,10)

# Producto y Division
v_ab_prod <- v_a*v_b
v_ab_div <- v_a%%v_b
v_ab_resto <- v_a%%v_b
print(rbind(v_ab_div,v_ab_resto))

##           [,1] [,2] [,3]
## v_ab_div      3     5     4
## v_ab_resto     1     2     2

# Determinante
v_d <- cbind(v_a, v_b, v_c)
det(v_d) #Determinante

## [1] 192
```

- 1 Entorno de R y R Studio
- 2 Creación de Objetos
- 3 Bucles y Condicionales**
- 4 Creación de Funciones
- 5 Manejo de datos
- 6 Gráficos en R
- 7 Manejo de Paquetes en R

Bucles y Condicionales

- Bucles: For

```
for (i in 1:3) {  
  print(paste0("Bienvenidos al curso de R el alumno ",i))  
}  
## [1] "Bienvenidos al curso de R el alumno 1"  
## [1] "Bienvenidos al curso de R el alumno 2"  
## [1] "Bienvenidos al curso de R el alumno 3"
```

- Condicionales: If

```
for (i in 1:4) {  
  if ( i%%2 == 0) {  
    print(paste0("El numero ",i, " es par"))  
  } else {  
    print(paste0("El numero ",i, " es impar"))  
  }  
}  
## [1] "El numero 1 es impar"  
## [1] "El numero 2 es par"  
## [1] "El numero 3 es impar"  
## [1] "El numero 4 es par"
```

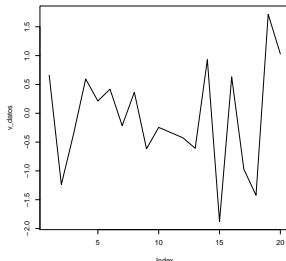
- 1 Entorno de R y R Studio
- 2 Creación de Objetos
- 3 Bucles y Condicionales
- 4 Creación de Funciones**
- 5 Manejo de datos
- 6 Gráficos en R
- 7 Manejo de Paquetes en R

Creación de una función

- Se requiere de una función que almacene n datos aleatorios normales y la gráfica respectiva:

```
CrearAleatorioGrafico <- function(n) {  
  v_datos <- rnorm(n,mean=0,sd=1)  
  g_datos <- plot(v_datos,type="l")  
  l_datos <- list(v_datos,g_datos)  
  return(l_datos)  
}
```

```
l_modelo <- CrearAleatorioGrafico(20)
```



- Sea el número áureo:

$$\phi = \frac{(1 + \sqrt{5})}{2} \quad (1)$$

- Se puede calcular el número Fibonacci de orden n

$$Fib(n) = \frac{\phi^n - (1 - \phi)^n}{\sqrt{5}} \quad (2)$$

- Crear una función en R que permita calcular el número Fibonacci de orden n .
- Crear otra función que permita determinar si un número es Fibonacci o no. TIP: Un número es fibonacci si al menos uno de las siguientes expresiones es cuadrado perfecto:

$$5 * Fib^2 \pm 4$$

- 1 Entorno de R y R Studio
- 2 Creación de Objetos
- 3 Bucles y Condicionales
- 4 Creación de Funciones
- 5 Manejo de datos**
- 6 Gráficos en R
- 7 Manejo de Paquetes en R

Tratamiento con Strings

```
#Concatenar
software <- 'R'
curso <- 'Aplicado a Finanzas'
paste(software,curso,sep = " , ")

## [1] "R , Aplicado a Finanzas"

#Mayuscula o Minuscula
v_nombres <- c("Manuel","Juan","Jose")
nombre <- "jose"

nombre %in% v_nombres; toupper(nombre) %in% toupper(v_nombres)

## [1] FALSE
## [1] TRUE

#Extraer partes del texto
v_fecha <- c("20180810","20190810")
substring(v_fecha,1,4)

## [1] "2018" "2019"
```

- Archivos con formato delimitado por comas (CSV):

```
archivo = paste0(ruta,"fallecidos_covid.csv")
df_fallecidos <- read.csv(archivo,header = T,sep = ",")
colnames(df_fallecidos)
```

## [1]	"FECHA_CORTE"	"UUID"	"FECHA_FALLECIMIENTO"
## [4]	"EDAD_DECLARADA"	"SEXO"	"FECHA_NAC"
## [7]	"DEPARTAMENTO"	"PROVINCIA"	"DISTRITO"

- En el caso se desee importar datos desde un archivo txt, se utiliza el comando: *read.table*.
- Si se desea importar otros formatos se puede hacer uso de los siguientes packages: *foreign*, *Hmisc* o *xlsx*,

Manipulación de datos

- Filtrar por una condición

```
df_fallecidos[df_fallecidos$DEPARTAMENTO == "LIMA",]
```

- Seleccionar más de una columna

```
df_fallecidos[,c("FECHA_FALLECIMIENTO", "EDAD_DECLARADA")]
```

- Seleccionar los 150 primeros registros

```
df_fallecidos[1:150, c("FECHA_FALLECIMIENTO", "SEXO")]  
head(df_fallecidos[, c("FECHA_FALLECIMIENTO", "SEXO")], n = 150)
```

- Seleccionar los 150 últimos registros

```
tail(df_fallecidos[, c("FECHA_FALLECIMIENTO", "SEXO")], n = 150)
```

- Creación de Variables

```
df_fallecidos$MES <- substring(df_fallecidos$FECHA_FALLECIMIENTO, 1, 6)
```

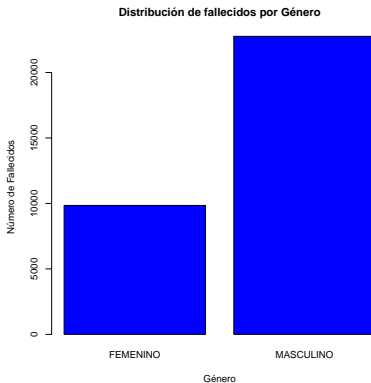
- Agregación de variables

```
df_datos <- aggregate(UUID ~ SEXO + MES, data = df_fallecidos ,  
                      FUN = length)
```

- 1 Entorno de R y R Studio
- 2 Creación de Objetos
- 3 Bucles y Condicionales
- 4 Creación de Funciones
- 5 Manejo de datos
- 6 Gráficos en R**
- 7 Manejo de Paquetes en R

Gráficos en R

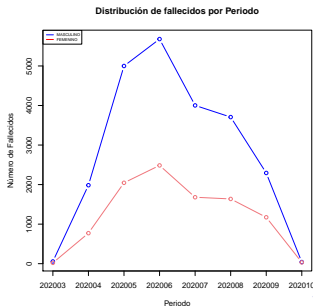
```
#Graficar el Producto Bruto Interno  
plot(as.factor(df_fallecidos$SEXO),  
     main = "Distribución de fallecidos por Género",  
     ylab = "Número de Fallecidos",  
     xlab = "Género",  
     col = "blue")
```



Gráficos en R

#Graficar de Múltiples Líneas + Leyenda

```
plot(df_datos[df_datos$SEXO == "MASCULINO",]$MES,  
     df_datos[df_datos$SEXO == "MASCULINO",]$UUID,  
     main = "Distribución de fallecidos por Periodo",  
     ylab = "Número de Fallecidos", xlab = "Periodo",  
     col = "blue", type = "b")  
lines(df_datos[df_datos$SEXO == "FEMENINO",]$MES,  
      df_datos[df_datos$SEXO == "FEMENINO",]$UUID,  
      type = "b", col = rgb(240,118,124,max=255))  
legend("topleft", legend = c("MASCULINO","FEMENINO"),  
      col = c("blue","red"), lty = 1, cex = 0.5)
```



- 1 Entorno de R y R Studio
- 2 Creación de Objetos
- 3 Bucles y Condicionales
- 4 Creación de Funciones
- 5 Manejo de datos
- 6 Gráficos en R
- 7 Manejo de Paquetes en R

- Instalar paquetes en R: `install.packages(packages)`
Por lo general se descargan del siguiente repositorio:
repos = "http://cran.us.r-project.org"
Instalar las siguientes librerías
 - *data.table*: Manejo de Base de Datos
 - *ggplot2*: Gráficos personalizados
- Cargar paquete en R

```
library(data.table)  
library(ggplot2)
```

Package: data.table

- Importar datos

```
dt_positivos <- fread(paste0(ruta,"positivos_covid.csv"),sep = ",")
```

- Filtrar y seleccionar columnas

```
dt_positivos[SEXO == "MASCULINO" & EDAD < 18, c("DEPARTAMENTO", "PROVINCIA")]
```

##	DEPARTAMENTO	PROVINCIA	DISTRITO
## 1:	CUSCO	CUSCO	WANCHAQ
## 2:	LIMA	EN INVESTIGACIÓN	EN INVESTIGACIÓN
## 3:	CUSCO	CUSCO	SAN JERONIMO
## 4:	LIMA	LIMA	SAN JUAN DE MIRAFLORES
## 5:	LIMA	LIMA	RIMAC
## ---			
## 24037:	CUSCO	LA CONVENCION	SANTA ANA
## 24038:	AREQUIPA	AREQUIPA	SOCABAYA
## 24039:	AREQUIPA	AREQUIPA	YANAHUARA
## 24040:	CUSCO	CUSCO	SAN SEBASTIAN
## 24041:	CUSCO	CUSCO	SANTIAGO

- Crear variables

```
dt_positivos[, PERIODO := substring(FECHA_RESULTADO, 1, 6)]
```

- Agrupar Variables

```
dt_r <- dt_positivos[, list(MeanEdad = mean(EDAD, na.rm = T)),  
  by = c("PERIODO", "SEXO")]
```

Package: ggplot2

- Importar datos

```
dt_casos <- dt_positivos[,list(N = .N),by = c("PERIODO","SEXO")]  
ggplot(data = dt_casos)+  
  geom_line(aes(x = PERIODO, y = N,group = SEXO,col= SEXO)) +  
  ggtitle("Número de Casos por Género") +  
  xlab("Periodo")+  
  ylab("Número de Casos")+  
  theme(axis.text.x = element_text(size = 12, angle = 90)) +  
  theme(title = element_text(size = 18)) +  
  theme_bw()
```

