EYP1113 - Probabilidad y Estadística

Laboratorio 1

Profesoras: Natalia Venegas y Pilar Tello

Facultad de Matemáticas
Departamento de Estadística
Pontificia Universidad Católica de Chile

Segundo Semestre 2019

Contenido I

Introducción a R

R es un conjunto integrado de programas para manipulación de datos, calculo y gráficos.

R es un software estadístico de libre acceso el cual puede ser utilizado en diferentes sistemas operativos como Windows, MacOS y Linux.

R es un lenguaje de programación en el que se introducen códigos para posteriormente ser ejecutados.

Una de las grandes ventajas de R es que es un programa de código abierto en el que miles de personas de todo el mundo colaboran en el desarrollo de nuevas metodologías, de manera que se pueden acceder a los paquetes descargándolos como también compartir los propios paquetes con otros.

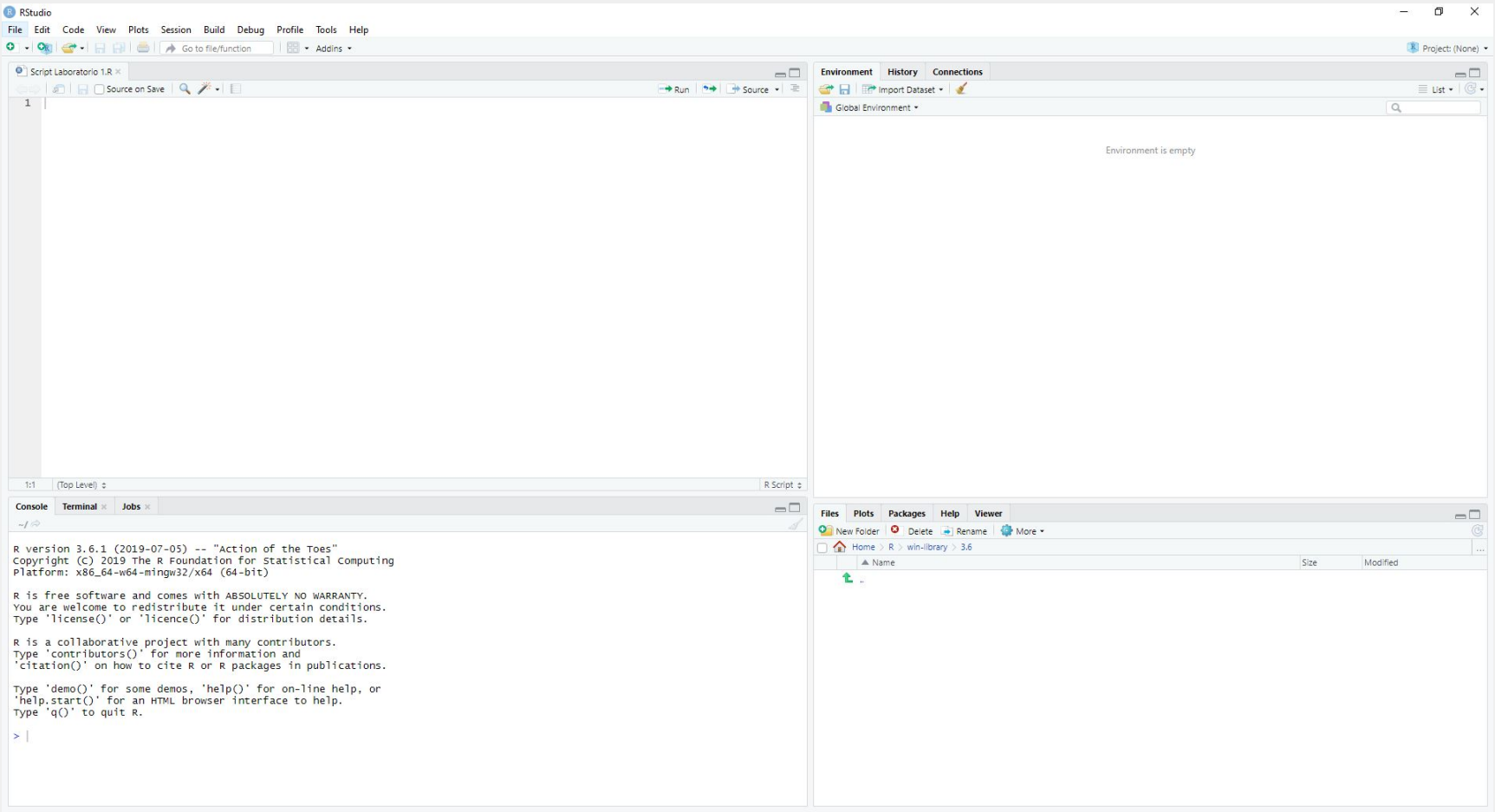
La descarga del archivo de instalación de R se realiza desde uno de los links de abajo dependiendo del sistema operativo:

- Microsoft Windows: http://cran.r-project.org/bin/windows/base/
- OSX: http://cran.r-project.org/bin/macosx/
- Linux: http://cran.r-pro ject.org/bin/linux/

Una vez instalado R, se puede instalar R Studio desde:

- https://www.rstudio.com/products/rstudio/download/

Introducción a R



En la consola de R, se puede escribir directamente el código de R , y pulsar **enter** para ver el valor. Sin embargo, esta no es la manera más eficiente de trabajar en R.

Si se quiere guardar el trabajo, corregirlo, repetirlo, etc., es más conveniente usar el editor de R. Se debe seleccionar archivo, nuevo Script (documento en blanco del editor) en el cual se puede escribir los programas y guardar.

En sistema Windows, para mayor comodidad, se puede seleccionar la opción “Dividir verticalmente” del menú Ventana de la consola.

La ejecución del código desde el script se hace desde cualquier posición de la línea o seleccionando

Windows: Tecla F5 o Control + R, MacOS: comand + enter.

Se puede incluir comentarios que R no leerá si utilizamos el símbolo # al comienzo de la línea.

En R es posible llevar a cabo distintas operaciones matemáticas usando operadores básicos tales como $+, -, /, **, *$.

Ejercicio 1: Calcule en la consola de R la siguiente suma $11 + 8 + 2016$ y presione ENTER. Observe el resultado. Ahora, calcule la siguiente operación usando los operadores mostrados anteriormente:

$$3 - (2 - (1 - (15 : 5 \times 3)) - 3^2)$$

Los resultados obtenidos de cualquier operación, o de aplicar una función, van a apareciendo en color azul, antecedido por el símbolo $>$, mientras que cualquier código o sentencia que escribamos aparecerá en color rojo.

A continuación se presentan distintos comandos de utilidad usados
comúnmente en R.

Funciones matemáticas:

Nombre de la función	Descripción
`sqrt`	Raíz cuadrada
`log, log2, log10`	Logaritmos
`exp`	Función exponencial
`abs`	Valor absoluto
`sign`	Signo
`cos, sin, tan`	Funciones trigonométricas
`acos, asin, atan`	Funciones trigonométricas inversas
`%%`	Resto de una división
`factorial, lfactorial`	Factorial y su logaritmo

Ejercicio 2 (Tarea): Calcule la siguientes cantidades:

a) $\sqrt{1082016}$

b) $\log 235$

c) $\exp\{0\}$

d) $85!$

e) $\frac{\sin \pi}{\cos 2\pi}$

R es un software que permite la creación de objetos de varios tipos: alfanuméricos, escalares, vectores, matrices, etc. Los valores que se asignen a los objetos quedan guardados en la memoria de R mientras dure la sesión de trabajo. Los objetos pueden definirse de la siguiente forma:

```
objeto <- expresión
```

donde `objeto` es el nombre que se le asigna al objeto, y `expresión` puede ser una fórmula, un vector, una palabra, etc.

Ejercicio 3 (Tarea): Definir en R los objetos **a**, **b** y **c** como:

```
a <- "Laboratorio EYP113"
b <- 3*6+9/7
d <- sqrt(9)/log(10)
```

Ahora, escriba en la consola **a** y presione ENTER. Repetir para **b** y **d**.
Escribir en la consola de R las siguientes expresiones $a + b, a * d, b/d$ y $d - b$ y observe los resultados. Comente.

- Para ver los objetos creados en la sesión de trabajo, debe usarse el comando `ls()` o la sentencia `objects()`.
- También, si se quiere eliminar algún objeto en especial, se puede usar `rm(objeto)`. Si se quiere borrar todos los objetos creados en la sesión usar la sentencia `rm(list=ls())`.
- Si se quiere guardar todos los objetos del espacio de trabajo en un archivo usamos la sentencia `save.image(file="nombre")`, donde `nombre` es el nombre que se quiere dar al archivo guardado.
- Si se quiere guardar algunos objetos, usamos el comando `save(x, file="nombre")`. Para cargar un espacio de trabajo anterior, usamos el comando `load(file="nombre")`.

Ejercicio 4 (Tarea):

a) Muestre en la consola de R todos los objetos que se han creado en la sesión actual.

b) Elimine el objeto **a** y verificar que el objeto no existe, escribiendo a en la consola y presionando ENTER.

c) Guardar los objetos **b** y **d** en un archivo.

d) Eliminar todos los objetos de la sesión.

e) Cargar los objetos guardados en el archivo creado en la parte c), y verificar que ahora existen.

Funciones en R

Ya vimos que en R hay una gran variedad de funciones para utilizar. Sin embargo, también es posible crear nuestras propias funciones, entregando los *inputs* que nosotros deseemos.
Ejemplos de funciones:

- la función `seq(from=a,to=b,by=c)`
- la función `rep(x,each=a,...)`

Vectores en R

En R también podremos crear vectores, usando las funciones `c()`, `rep()` y `seq()`.

Ejemplos:

- Una variable de una base de datos importada en R, corresponde a un vector
- Use las funciones mencionadas anteriormente para definir el objeto `z` como el vector
  `(-3,-1,1,3,5,...,79,81,79,81,79,81,79,81)`
  Solución: `z <- seq(from=-3, to=81, by=2)`

Para obtener el tamaño de un vector, i.e. el número de elementos o componentes, usaremos la función `length()`, y si queremos saber cuál es el elemento del vector en la posición `i` usamos `vector[i]`. En el ejemplo anterior:

```
# Largo de z:  largo.z <- length(z)
# Elemento 2 del vector:  elemento2 <- z[2]
```

Vectores en R

Podemos crear vectores numéricos cuyo incremento sea de una unidad, usando la expresión `a:b`.
Ejemplo: el vector `(1,2,3,...,7,8)` se crea usando `1:8`

Ejercicio 5: Del vector definido en el apartado anterior, definir el subvector y correspondiente a las primeras 3 componentes.

En R también es posible llevar a cabo operaciones aritméticas, usando las operaciones básicas y también las funciones preestablecidas.

Ejercicio 6 (Tarea): ¿Qué sucede cuando operamos con vectores de distintos largo, como por ejemplo `c(1,2,3)+c(6,8,9,7)`?

Expresiones lógicas

En R podemos representar expresiones lógicas con los operadores de comparación: $<$ (menor), $>$ (mayor), $<=$ (menor o igual), $>=$ (mayor o igual), $==$ (es igual) y $!=$ (no es igual). Y con los operadores lógicos & (y), | (o), y ! (no).

Ejemplo: evaluar las siguientes expresiones lógicas:

- $9 \leq 6$ (`9<=6`)
- `c(2,5,8)` $>$ `c(9,8,7)`

Matrices en R

Existe una función en R para definir matrices como objetos:

```
matrix(data, ncol=, nrow=, byrow=FALSE)
```

Donde `data` puede ser un vector o un escalar (fijarse que el largo del vector sea igual al producto del número de filas multiplicado por el número de columnas de la matriz), `ncol=` el número de columnas de la matriz, `nrow=` el número de filas y `byrow=` un argumento lógico que indica si la matriz se completa por filas (`TRUE`) o por columnas (`FALSE`).
Si se quiere obtener un elemento específico de una matriz usamos `[ , ]`. Por ejemplo, sea la matriz $A_{n \times k}$:

- La primera columna de la matriz A: `A[ , 1]`
- La primera fila de la matriz A: `A[1 , ]`
- La celda (i,j) de la matriz A (intersección de la fila i con la columna j) `A[i , j]`

Funciones asociadas a matrices

Algunas funciones asociadas a matrices en R:

Función	Descripción
`diag()`	Diagonal de una matriz
`*`	Prod elemento a elemento
`%*%`	Prod matricial
`dim(), ncol(), nrow()`	Dimensiones de una matriz
`t(), det(), solve()`	Transpuesta, determinante e inversa

Funciones asociadas a matrices

Ejercicio 7: Defina dos matrices cuadradas de 2×2 e implemente las funciones mostradas anteriormente.

También podemos definir matrices usando las funciones `cbind()`, que concatena vectores columnas, y `rbind()`, que concatena vectores filas.

Si se desea saber si un objeto de R es una matriz podemos usar la función `is.matrix()` y para definir un objeto como una matriz usamos el comando `as.matrix()`.

R contiene dos tipos de paquetes, los del tipo Base los cuales están incorporados automáticamente en la instalación de R, y los paquetes de contribución los cuales se deben descargar para su instalación.

Ejecutando el comando getOpcion() en la consola se obtiene las aplicaciones que contiene el paquete base

```
getOption("defaultPackages")
"datasets"  "utils"     "grDevices" "graphics"  "stats"     "methods"
library(help = "datasets")
```

Existe una gran variedad de paquetes de contribución que son aporte de personas a lo largo del mundo (los cuales son gratuitos). Se requiere conexión a internet para descargarlo e instalarlo y se debe ejecutar

```
install.packages("Nombre")
```

Una vez instalado el paquete se carga con el comando

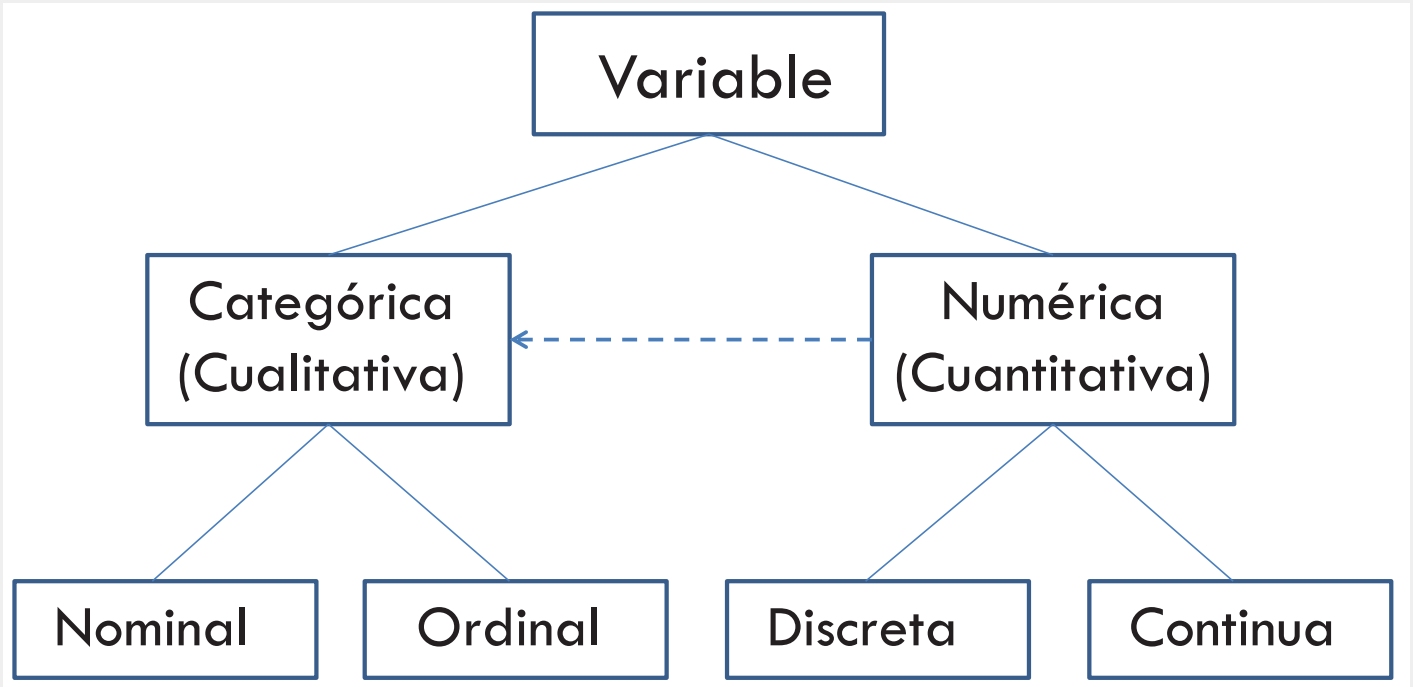
```
library(Nombre)
```

Estadística Descriptiva con R

Antes de comenzar cualquier análisis de datos es importante saber:

- Formato en que se encuentra la información: TXT, DAT, XLS, XLSX, CSV, SPSS, SAS, SQL, ACCES, etc.
- Indicador de dato faltante: “na”, “NA”, “,”, “;”, “ ”, etc.
- Números especiales: "88", "888", "99", "999", etc.

En resumen, tener el llamado libro de variables a mano.


Variable
Categórica
(Cualitativa)
Numérica
(Cuantitativa)
Nominal
Ordinal
Discreta
Continua

La mayoría de las veces la información que se recolecta se presenta de la siguiente manera:

	Variables						
Observación	X_1	X_2	X_3	$\cdots$	X_j	$\cdots$	X_K
1	x_{11}	x_{12}	x_{13}	$\cdots$	x_{1j}	$\cdots$	x_{1K}
2	x_{21}	x_{22}	x_{23}	$\cdots$	x_{2j}	$\cdots$	x_{2K}
3	x_{31}	x_{32}	x_{33}	$\cdots$	x_{3j}	$\cdots$	x_{3K}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$
i	x_{i1}	x_{i2}	x_{i3}	$\cdots$	x_{ij}	$\cdots$	x_{iK}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$
N	x_{N1}	x_{N2}	x_{N3}	$\cdots$	x_{Nj}	$\cdots$	x_{NK}

donde x_{ij} corresponde a los valores (o nombres) de toman las variables para una i-ésima observación.

Para importar bases de datos en formato TXT, DAT y Excel, en la consola de **R** se utilizan las siguiente funciones:

- **read.table()**: Importa Base de Datos en formato TXT, DAT y CSV
- **read.csv()**: Importa Base de Datos en formato CSV
- **readXL()**: Importa Base de Datos en formato XLS y XLSX. (paquete "readxl")
- **scan()**: Importar Vector de Datos

Lectura de datos

Lectura de datos en R

Las condiciones climáticas son muy importantes para practicar deportes. La base de datos `Tenis.txt` contiene información diaria que se ha recolectado sobre condiciones climáticas y la decisión final de un jugador profesional de Tenis para practicar el deporte:

Variable	Descripción
`Dia`	identificador del día evaluado
`Pronostico`	pronóstico del día: Soleado, Nublado o Lluvioso
`Temperatura`	temperatura pronosticada del día: Calido, Moderado o Frio
`Temperatura Maxima`	temperatura máxima pronosticada en grados Celsius
`Temperatura Minima`	temperatura mínima pronosticada en grados Celsius
`Humedad`	humedad pronosticada del día: Alta o Normal
`Viento`	intensidad del viento pronosticada del día: Alta o Debil
`Juega_Tenis`	indica si el jugador juega finalmente tenis o no. 1: Sí 0: No

Una vez que los datos ya están leídos, a veces es de utilidad usar el comando `attach()`, el cual permitirá trabajar directamente con ellos. Además, con la función `names()` se podrán obtener los nombres de las variables correspondientes.

Lectura de datos en R

Las dos formas más comunes de leer una base de datos son:

- `data<-read.table(file.choose(),header=TRUE)`
- `data<-read.table("..../BaseLaboratorio02.txt",h=T`

donde la función `file.choose()` permite seleccionar directamente un archivo de la unidad de trabajo, sea cual sea la extensión. La opción `header` hace referencia al encabezado de las variables, donde e `T` o `TRUE` si existe encabezado, o `F` o `FALSE` si no tiene.

Lectura de datos en R

En caso donde la lectura es través del directorio, una sentencia muy conveniente es:

- `getwd()`
- `setwd()`

Ejemplo:

```
setwd("../EYP113/Laboratorio02/")
data<-read.table("BaseLaboratorio02.txt", header=TRU
```

Lectura de datos en R

Se puede obtener la clase de cada columna de la base mediante la sentencia:

- `class()`

Ejemplo:

```
class(data$Temperatura_Maxima)
class(Temperatura_Maxima)
class(data$Juega_Tenis)
class(Juega_Tenis)
```

Lectura de datos en R

Observemos que la `Temperatura_Maxima` se ha leído como un factor, es decir, una variable categórica. Para corregir este error al abrir la base de datos debemos indicarle a la función `read.table` que en nuestra base de datos los decimales vienen delimitados con una coma (,) con el argumento `dec=","`.
Ejemplo:

```
data<-read.table(file.choose(),header=TRUE,dec=",")
```

Lectura de datos en R

La variable `Juega_Tenis` se ha considerado una variable numérica, cuando debería ser una variable categórica. Esto se puede modificar con la sentencia:

- `as.factor()`

En el caso contrario se podría modificar con la sentencia:

- `as.numeric()`

Ejemplo:

```
as.factor(data$Juega_Tenis)
```

Las medidas de resumen más comunes para variables numéricas se pueden clasificar de la siguiente manera:

- Tendencia Central: Media, Moda, Mediana.
- Posición: Percentil, Mínimo, Máximo.
- Dispersion: Varianza, Desviación Estándar, Coeficiente de Variación, Rango, Rango Intercuantil.
- Forma: Coeficiente de Asimetría, Kurtosis.

Mientras que en las variables no numéricas solo se pueden trabajar como tablas de frecuencias

A continuación se presentan distintos comandos de las principales medidas de resumen en R.

Nombre de la función	Descripción
`mean`	media
`var`	varianza
`sd`	desviación estándar
`summary`	Resumen de un vector numérico
`quantile`	Cuantiles de una muestra
`min, max, range`	Mínimo y máximo de una muestra
`median`	Mediana de una muestra

Ejericio 8:

a) Importe la base de datos a R, adjunte los datos y reconozca los nombres de las variables.
b) ¿Cuál es el pronóstico de `Temperatura` más frecuente?
c) ¿Cuál es el día con el menor pronóstico de `Temperatura_Minima`?
d) Obtenga estadísticas de resumen para las variables `Temperatura_Maxima` y `Temperatura_Minima`.
e) ¿Cuál es la mediana para la `Temperatura_Maxima`?