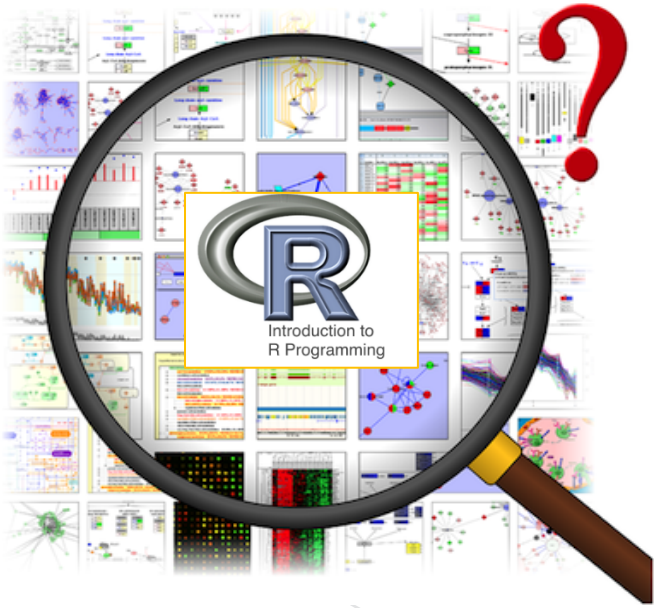
Introducción al R

Victor Manuel Hoyos Valencia – victor.hoyos@idata.com.co

Qué requerimos saber para trabajar con R?

- Herramientas de trabajo
- Tipos de datos
- Carga de datos
- Tratamiento de datos ausentes
- Análisis exploratorio

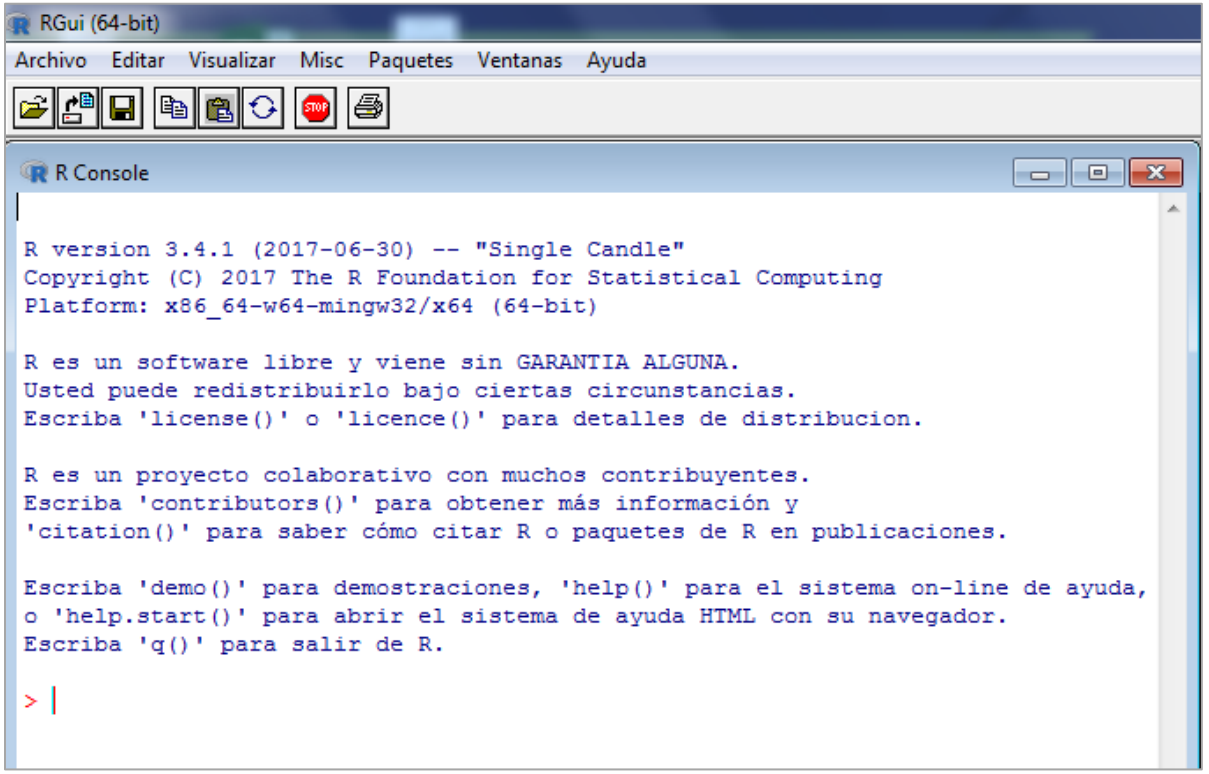


Cuáles son las Diferencias entre R y R Studio?

- R (programming language) is a programming language and environment, "made by statistician"
- **RStudio** is a Integrated development environment dedicated for R development.

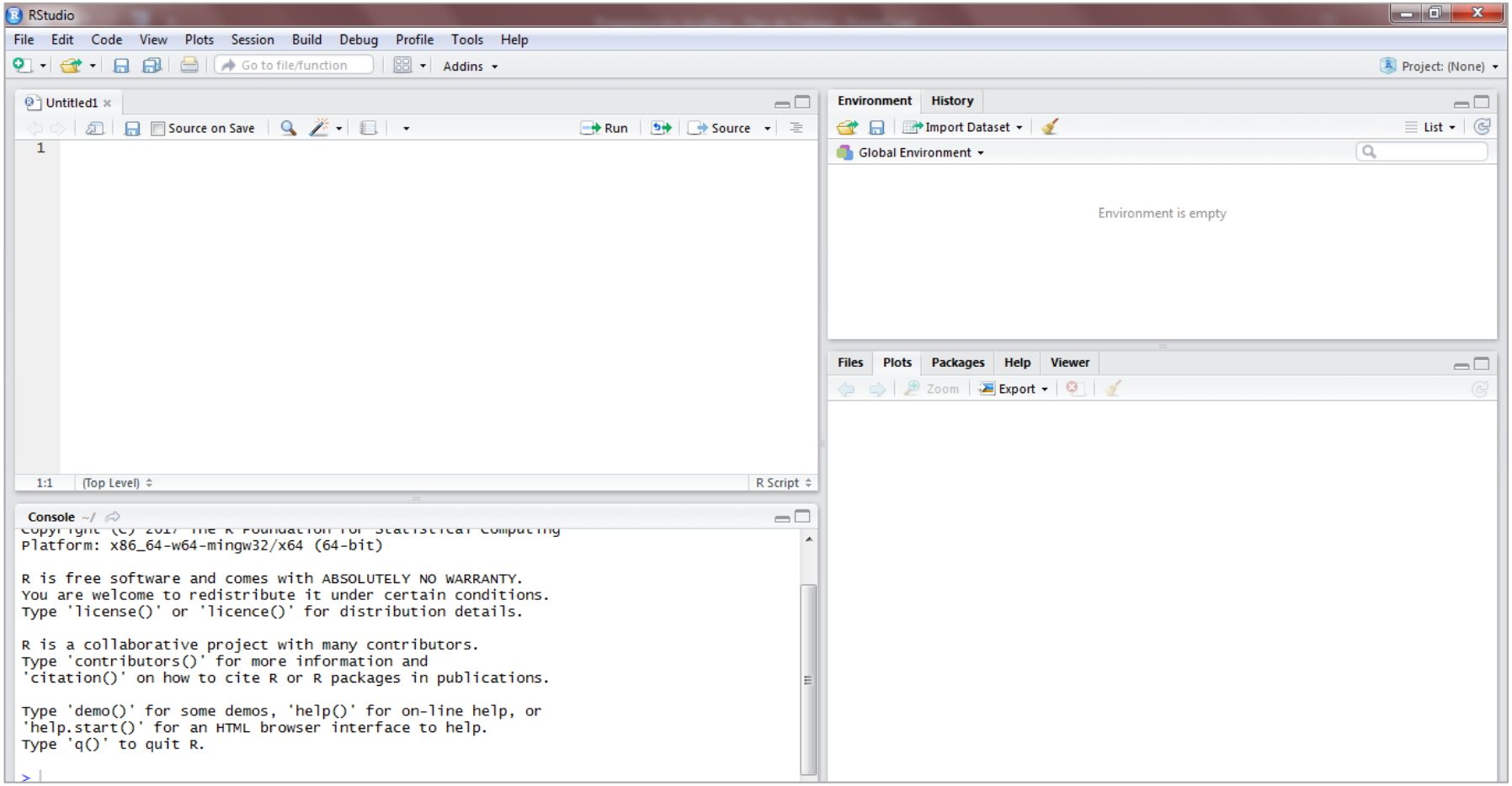
Nuevas versiones de R aparecen con el paso del tiempo

- Trabajo interactivo mediante línea de comandos



Cuáles son las Diferencias entre R y R Studio?

- Descarga desde http://www.rstudio.com/



- RStudio Server, accesible desde un navegador

```
>> instalaPaquetes<<
```

Herramientas de trabajo ► `>> tareasHabituales.R <<`

- Acceso a la documentación
  - `help('source'), vignette('grid'), demo('image')`
- Ruta de trabajo
  - `getwd(), setwd()`
- Espacio de trabajo
  - `save(), save.image(), load()`
- Instalación de paquetes
  - `install.packages(), library()`

Tipos de datos

- Simples
- Vectores
- Matrices
- Factors

>> tiposDatos.R <<

- Data Frames
- Listas

>> tiposDatosII.R <<

Tipos de datos ► Simples

- numeric
  - Enteros :: `1024, -3`
  - Punto flotante :: `3.1415927`
  - Notación exponencial :: `3.85e6`
  - Otros :: `Inf, NaN`
- integer :: `as.integer(numeric)`
- complex :: `1+2i`
- character :: `'R', "Hola"`
- logical :: `TRUE, FALSE, NA`

Tipos de datos ► Variables

- Asignación
  - `a = 1024 | a <- 1024 | 1024 -> a`
- Obtención de clase y tipo
  - `class(a) # numeric | typeof(a) # double`
- Comprobación de tipo
  - `is.numeric(a), is.character(a), is.integer(a), is.infinite(a), is.na(a)`
- Objetos en el espacio de trabajo
  - `ls(), rm(var), str(var), save(var,file = arch), save.image(), load()`

Tipos de datos ► Vectores

- Definición
  - `diasMes <- c(31,29,31,30,31,30,31,31,30,31,30,31)`
  - `dias <- c('Lun','Mar','Mié','Jue','Vie','Sáb','Dom')`
  - `quincena <- 16:30`
  - `semanas <- seq(1, 365, 7)`
  - `rep(T, 5)`
- Obtención número de elementos
  - `length(dias)`
- Acceso a elementos
  - `dias[2]` `# Segundo elemento del vector`
  - `dias[-2]` `# Todos los elementos menos el segundo`
  - `días[c(3, 7)]` `# Elementos 3 y 7`

- Generación de valores aleatorios
  - Establecimiento de la semilla: `set.seed(4242)`
  - Distribución normal
    - `rnorm(100, mean = 10, sd = 3)`
  - Distribución uniforme
    - `runif(6, min = 1, max = 49)`
  - Otras distribuciones
    - `rbinom()`, `rlogis()`, `rpois()`, etc.
- Operaciones sobre vectores
  - No necesaria la iteración por los elementos
  - Posibilidades de paralelización

Tipos de datos ► Matrices

- Definición
  - `mes <- matrix(1:35, nrow = 5)`
  - `mes <- matrix(1:35, ncol = 7, byrow = T)`
- Obtención número de elementos
  - `length(mes)` | `nrow(mes)` | `ncol(mes)`
- Acceso a elementos
  - `mes[2, ]` `# Segunda fila completa`
  - `mes[ ,2]` `# Segunda columna completa`
  - `mes[2, 5]` `# Quinta columna de la segunda fila`
  - `fix(mes)` `# Edición de elementos en la matriz`

Tipos de datos ► Factors

- Definición
  - `herramientas <- factor('Consola','RStudio')`
  - `fdias <- factor(días)`
  - `tam <- ordered(c('Ligero','Medio','Pesado'))`
- Obtención niveles
  - `nlevels(fdias)`
  - `levels(días)`
- Relación de orden (factors ordenados)
  - `tam[2] < tam[1]  # FALSE`

Tipos de datos ▶ Data Frames

- Definición
  - `df <- data.frame(vect1, …, vectN)`
  - `df <- data.frame(matrix)`
  - `df <- data.frame(col1 = tipo(N), …, colN = tipo(N)`
- Ejemplo
  - ```
    df <- data.frame(Dia = fdias,
                     Estimado = rep(c(T, F), 7),
                     Lectura = rnorm(14, 5))
    ```
- Obtención número de elementos
  - `nrow(mes)`
  - `ncol(mes)`
    ```

Tipos de datos ► Data Frames

- Selección y proyección de datos
  - `df[5, ]` `# 5ª fila`
  - `df[ ,3]` `# 3ª columna`
  - `df[c(-3,-6), ]` `# Menos 3ª y 6ª fila`
  - `df$Lectura` `# 3ª columna`
  - `df$Lectura[5]` `# 5ª fila de 3ª col.`
  - `df[, c('Dia','Lectura')]` `# Columnas 1 y 3`
  - `df[df$Estimado == F, ]` `# Filas condición`
  - `# Selección de filas y columnas`
    `df[df$Estimado == F & df$Lectura > 3,`
    `c('Dia', 'Lectura')]`

Tipos de datos ► Data Frames

- Agregar nuevas filas
  - `df[15, 1] <- 'Vie'`
  - `df$Dia[15] <- 'Vie'`
  - `rbind(df, data.frame(Dia=fdias[1], Est=F, Lect=5))`
  - `rbind(df[1:9, ], nuevaFila, df[10:14, ])`
- Agregar nuevas columnas
  - `df$Ajustado <- df$Lectura + rnorm(15, 2)`
  - `cbind(df, Ajustado = df$Lectura + rnorm(15, 2))`
  - `cbind(df[ ,c(1,3)], nuevaCol, df$Estimado)`
- Nombres de filas y columnas
  - `names(df)  # Vector con nombres de columnas`
  - `rownames(df)  # Vector con nombres de filas`

Tipos de datos ► Listas

- Definición
  - `lst <- list(3.1415927, 'Hola', TRUE, fdias[4])`
  - `lst <- list(fdias, mes, df)`
- Información sobre la lista
  - `length(lst)`
  - `names(lst)`
- Acceso a los elementos
  - `lst[[1]]`
  - `lst[['PI']]`
  - `lst$PI`

Carga de datos ► >> cargaDatos.R <<

- Lectura de archivos CSV
- Importación de hojas de cálculo Excel
- Carga de datasets en formato ARFF
- Obtención de datos de otras fuentes

Carga de datos ▶ csv

```
datosCSV <- read.table(
    file         = "miArchivo.csv",
    header       = T,
    sep          = ",",
    dec          = ".",
    quote        = "\"")
```

```
read.csv("miArchivo.csv")  # sep=",", dec="."
```

```
read.csv2("miArchivo.csv") # sep=";", dec=","
```

Carga de datos ▶ Excel

- Múltiples posibilidades
  - Exportar desde Excel a CSV
  - Copiar datos al portapapeles
  - Leer archivo Excel desde R
- Paquetes R para trabajar con archivos Excel
  - XLConnect
    - `datos <- readWorksheetFromFile('archivo.xls', sheet=n)`
  - xlsx
    - `datos <- read.xlsx('archivo.xlsx',sheetName = n, rango)`
- `vignette(paquete)` # Abrir el manual asociado

Carga de datos ► ARFF

- Paquete `foreign`
  - Funciones para leer múltiples formatos de archivo
  - `read.arff('dataset.arff')`
- Paquete RWeka
  - Interfaz completa entre R y Weka
    - Leer y escribir archivos ARFF
    - Acceso a algoritmos de clasificación, agrupamiento, etc.
  - `read.arff('dataset.arff')`

Carga de datos ► Otras fuentes

- Portapapeles
  - `read.delim('clipboard')`
  - `write.table(datos, 'clipboard')`
- Desde URL
  - `conn <-getURL('http://url/datos')  # Conexión abierta`
  - `datos <- read(conn)`
  - `conn <- getURL('https://url/datos')  # Conexión cifrada`
  - `datos <- read(textConnection(conn))`
- Datasets integrados
  - `data() # Lista de todos los datasets integrados`
  - `summary(iris)`

Tratamiento de datos ausentes ► >> tratamientoNulos.R <<

- Problemática
  - Datos ausentes (*missing values*) dificultan múltiples operaciones
- Detectar existencia de valores ausentes
  - `is.na(variable)`
  - `na.fail(variable)`
- Eliminar valores ausentes
  - `na.omit(variable)`
  - `complete.cases(variable)`
  - `variable[is.na(variable)] <- valor`
- Operar con presencia de valores ausentes
  - `mean(variable, na.rm = T)`
  - `lm(x ~ y, variable, na.action = na.omit)`

Análisis exploratorio ► Información general >> analisisExploratorio.R <<

- Estructura interna de la variable
  - `str(variable)`
- Resumen del contenido
  - `summary(variable)`
- Exploración del contenido
  - `head(variable)` | `tail(variable)`
  - `variable[filas, columnas]`
  - `variable$columna`
  - `variable$columna[which(condición)]`
    - `iris$Sepal.Length[which(iris$Species == 'versicolor')]`

Análisis exploratorio ► Estadística descriptiva

- Funciones básicas (operan sobre vectores)
  - `mean` `# media`
  - `median` `# mediana`
  - `var` `# varianza`
  - `sd` `# desviación estándar`
  - `max` `# máximo valor`
  - `min` `# mínimo valor`
  - `range` `# rango de valores`
  - `quantile` `# quartiles`
- Para estructuras complejas
  - `lapply(iris[ ,1:4], mean) # Aplicar a cada columna`
  - `describe(variable) # Paquete Hmisc`

Análisis exploratorio ► Agrupamiento de datos

- Tabla de contingencia con número de combinaciones
  - Longitud de sépalo según especie
    ```
    table(iris$Sepal.Length, iris$Species)
    ```
  - Valoración de vendedores según moneda
    ```
    table(ebay$sellerRating, ebay$currency)
    ```
- Agrupamiento y selección
  - Separar los casos por especie de flor
    ```
    split(iris, iris$Species)
    ```
  - Obtener elevación, pendiente y clase de filas que cumplan condición
    ```
    subset(covertype, slope > 45 & soil_type == '1',
           select = c(elevation, slope, class))
    ```

Análisis exploratorio ► Ordenación de datos

- Ordenar un vector obteniendo otro
  - `sort(valores)`
- Obtener la posición para cada valor
  - `order(valores)`
- Generar un ranking a partir de los valores
  - `rank(valores)`
  - `rank(valores, ties.method = 'average')`

Análisis exploratorio ► Particionamiento de datos

- Tomando el orden en que aparecen en el dataset
  - División entre entrenamiento (75%) y prueba (25%)

```
nTraining  <- as.integer(nrow(iris)    * .75)
training   <- iris[1:nTraining, ]
test       <- iris[(nTraining+1):nrow(iris), ]
```

- Tomando un subconjunto aleatorio
  - Misma proporción anterior

```
set.seed(4242)          # Asegurar reproducibilidad

indices   <- sample(1:nrow(iris), nTraining)
training  <- iris[indices, ]
test      <- iris[-indices, ]
```

Cibergrafía

- http://sci2s.ugr.es/otherCourses/CienciaDatosBigData
- https://www.r-bloggers.com/why-you-should-learn-r-first-for-data-science/
- https://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf

Consultar si se desea más información sobre generalidades de R, los manuales de **cran.r-project.org:**

- https://cran.r-project.org/doc/manuals/r-release/R-admin.html
- https://cran.r-project.org/doc/manuals/r-release/R-intro.html