

# Curso de Rstudio

## 1. Flujo de trabajo

- A)
  - 1. Reportar la investigación
  - 2. Reportar de forma que sea reproducible
  - 3. La comunicación y la ciencia acumulativa
  - 4. ¿Cómo ayudan las matemáticas en "replication crisis"?
  - 5. Excel no es la solución

## 2. Preguntas

- B)
  - 1. ¿Quien es Rstudio ?
  - 2. ¿Para que .Rmd format ?
    - 2.1  $R + md = Rmd$
  - 3. ¿Que es RMarkdown notebook ?

## 3. Rstudio

- C)
  - 1. ¿Que es Rstudio ?
    - 1.1  $Rstudio = Rmd\ editor$
    - 1.2 Compila y visualiza
    - 1.3 Platilla
      - 1.31 Encabezamiento
      - 1.32 Texto
      - 1.33 Código
    - 1.4 Salva y corre

## 4. Escribir un documento en markdown

- D)
  - 1. Markdown sintaxis
    - 1.1 Referencia rápida de markdown dentro Rstudio
    - 1.2 `* cursiva *` y `** negrita **`
    - 1.3 Seis encabezados
    - 1.4 Listas sin y con orden
    - 1.5 `[link](www.rstudio.com)`
    - 1.6 Escribir ecuaciones
      - 1.61 En linea
      - 1.62 En bloque
      - 1.63 Matemática  $\text{\LaTeX}$



## 5. R notebook VS Jupyter notebook

- E)
  - 1. Crisis de replicación de resultados
    - 1.1 Replicación es importante para los negocios
    - 1.2 La crisis de replicación es una crisis de credibilidad
      - 1.21 Nadie cree nada
      - 1.22 Credibilidad en base a estudios no reproducibles
  - 2. Necesidad de Mapas
  - 3. Ventajas de R notebook
    - 3.1 RStudio esta conectado a GitHub
    - 3.2 Código en línea
    - 3.3 Metadatos y Temáticas
    - 3.4 Tabla de contenido

## 6. Escribir código en Rmarkdown

- F)
  - 1. Opciones de código de Rmarkdown
    - 1.1 echo=FALSE
    - 1.2 eval=FALSE
    - 1.3 message=False
    - 1.4 warning=False
    - 1.5 fig,width=3, fig,height=5
    - 1.6 engin=python

## 7. Tipos de documentos de salida

- G)
  - 1. HTML por defecto
    - 1.1 PDF
    - 1.2 Word
    - 1.3 Beamer

## 8. Parámetros

- H)
  - 1. Re usar el documento
    - 1.1 Agregar parámetros en el encabezado
    - 1.2 Llamar valores de parámetros



## 9. Interfaces del ambiente integrado de Rstudio

- I)
  - 1. Editor de scripts
    - 1.1 Elija el formato de salida
    - 1.2 Insertar fragmento de código
    - 1.3 Navegar sección / fragmento de código
    - 1.4 Correr el documento
  - 1. Consola
    - 1.1 Como una calculadora R
    - 1.2 Terminal bash
  - 1. Objetos en el espacio de trabajo
    - 1.1 Importar un dataset
  - 1. Administrador de archivos y vista de gráficos
    - 1.1 Administrador de archivos
    - 1.2 Gráficos
    - 1.3 Paquetes
    - 1.4 Ayuda
  - 1. Menú principal
    - 1.1 Abre muchos tipos de archivos
    - 1.2 Opciones

## 10. R-studio directorios

- J)
  - 1.
    - 1.1
    - 1.2
    - 1.3

## 11. Lenguaje de ciencia de datos R

### 11.1. Leer archivos planos en R

- K)
  - 1. Formatos de archivo comunes como .xlsx y .csv
    - 1.1 `read.table( )`
    - 1.2 `read.csv()`
    - 1.3 Botón de importar de Rstudio



## 11.2. Comandos para explorar

- L) 1. Comandos de base R
  - 1.1 View()
  - 1.2 ncol()
  - 1.3 nrow()
  - 1.4 names()

## 11.3. Organizar la Data con tidyr

- M) 1. 1.1 ¿Qué son los datos ordenados?
  - 1.2 gather() y spread()
  - 1.3 separate()

## 11.4. Descubrir información con dplyr

- N) 1. 1.1 Conjunto de verbos para de datos
  - 1.2 select() y filter()
  - 1.3 mutate() y arrange()
  - 1.4 summarise()
  - 1.5 Operador pipe %>%
  - 1.6 data %>% select(VARIABLES) %>% filter(VARIABLES)



## 11.5. Gráficos con ggplot2

- Ñ) 1. ggplot2 vs gráficos base de R
  - 1.1 ¿Por qué elegir ggplot2?
- 2. El gráfico esta compuesto por elementos separados
  - 2.1 Lo primero es la data
  - 2.2 `summary(object)`
  - 2.3 Agregar una de las 29 `geom_`
  - 2.4 Ahora si hay un grafico
- 3. Atributos geométricos y estéticos
  - 3.1 puntos, líneas, barras
  - 3.2 color, forma, tamaño
- 4. Otros elementos
  - 4.1 Curvas separadas por condiciones
  - 4.2 Transformación estadística
  - 4.3 Sistema de coordenadas

## 12. Organización de ejemplos

- O) 1. 1.1
- 1.2
- 1.3

