

**MBA
USP
ESALQ**

Data Wrangling

Prof. Dr. Wilson Tarantin Jr.

Preparación de Datos en R

Luiz Rodriguez Farfani 005.374.619-81

Data wrangling

- Utilizaremos, principalmente, el dplyr
 - El dplyr es un paquete incluido en el tidyverse
 - Contiene funciones útiles para la manipulación/preparación de bases de datos
 - Material para referencia.
 - <https://dplyr.tidyverse.org/>
 - <https://github.com/rstudio/cheatsheets/blob/master/data-transformation.pdf>
 - Wickham, H. & Grolemund, G. **R for Data Science**: <https://r4ds.had.co.nz/index.html>

Data wrangling

- **Pipe:** encadenamiento de diversas funciones en secuencia
- **Rename:** alteración de nombres de variables
- **Mutate:** alteración de contenido de las variables y creación de nuevas variables
- **Filter:** selección de observaciones con base en criterios lógicos
- **Select:** selección de variables
- **Summarise:** creación de tablas con medidas resumen (estadísticas descriptivas)
- **Group by:** agrupación de las observaciones con base en criterios
- **Join:** unión (*merge*) de bases de datos

Creación de Projects y Scripts R Markdown

Luiz Rodriguez Fantini 005.374.619-81

R Markdown

- **Introducción al R Markdown**
- **Formateo básico del texto**
- **Inserción de fórmulas**
- **Chunks**
- **Generando outputs (HTML; PDF, DOC)**
- **Material para referencia.**
 - <https://rmarkdown.rstudio.com/index.html>

Proyectos de Data Science & Analytics en el GitHub

Luiz Rodriguez Fartini 005.374.619-81

Git

- Software útil para el control de versiones
- Registra los cambios realizados en los archivos
- Vamos a utilizarlo en conjunto con el Github
- Instalar el Git en la computadora (<https://git-scm.com/downloads>)
 - Sólo precisa avanzar todas las etapas en las configuraciones sugeridas

Github

- Sitio utilizado para alojar los archivos
 - <https://github.com/>
- Organizado en repositorios (carpetas) que pueden ser compartidas, inclusive, pueden ser publicadas
 - Útil para almacenar y compartir su portfolio de proyectos
- Los archivos de la computadora pueden ser enviados al Github (por el Git)

Git y Github

- Add y Commit
 - Cree una carpeta en el escritorio de su computadora
 - En RStudio, cree un nuevo scrip y escriba apenas # Versão 1
 - Guarde este archivo en la carpeta con el nombre Versão Exemplo.R
 - Dentro de la carpeta, haga clic con el botón derecho del mouse y elija Git Bash Here
- En Git, escriba **git init** (inicializa el Git en la carpeta seleccionada)
- Escriba **git add "Versão Exemplo.R"** (añade el archivo para el índice)
- Para generar versiones utilice el comando **git commit -m "título"** (son las versiones)

El nombre del commit, ejemplo: "Primeira Versão"

Git y Github

- Push

- En su Github cree un nuevo repositorio y nomínelo como preferir.
- Copie el enlace del repositorio creado
- En Git, escriba **git remote add origin(enlace de su carpeta).....**
- Finalmente, digite **git push -u origin master** (envía el archivo para el repositorio, quedando en la ramificación principal)
 - En la primera vez que sea realizado, solicitará login en el Github
- ¡Después de actualizar, es posible verificar que el archivo ya está en su Github!

Git y Github

- Creando y comparando versiones
 - Abra el archivo Versão Exemplo y escriba una línea más: # Versão 2
 - Después de guardar, cierre y con el botón derecho abra el Git Bash Here en la carpeta
 - Utilice los mismos procedimientos:
 - **git add “Versão Exemplo.R”**
 - **git commit -m “Segunda Versão”**
 - **git push –u origin master**
- ¡En Github, la nueva versión ya está disponible y podemos compararlas!

Note que no fue
necesario informar de
nuevo la dirección

Git y Github

- Creando ramificaciones en el repositorio
 - En los comandos anteriores, alteramos la ramificación principal del repositorio
 - Podríamos crear ramificaciones nuevas en el Github
 - **git checkout -b “nome da nova branch”**
 - En Git, ya existe la indicación de cambio de la “master” para la “nova”
 - Los mismos procedimientos de add y commit
 - **git push -u origin “nome da nova branch”**

Git y Github

- Importando repositorios (Clone y Pull)
 - Puede ser útil traer para su computadora archivos que están en el Github
 - Una forma de “descargar” tales archivos es por medio de la función clon
 - Cree una carpeta en su computadora
 - Dentro de la carpeta, con el botón derecho del mouse, abra el Git Bash Here
 - En Github, en el repositorio de interés, haga clic en **code** y copie el enlace
 - En Git, digite **git clone(enlace del repositorio).....**
 - Para descargar nuevamente, después de alteraciones en el Github, indique **cd “repositorio”**
 - En secuencia, digite **git pull** (el archivo fue actualizado en la computadora)

Git y Github

- Copiando repositorios públicos (Fork)
 - Es posible copiar repositorios que están publicados en el Github
 - Busque algún tema de interés
 - Accese al repositorio
 - En la esquina superior derecha, existe el botón **Fork**
 - Después de hacer clic, podrá ver el repositorio en su lista (en su perfil)

Git, Github y RStudio

- Es posible integrar el Git, Github y RStudio
- En RStudio, haga clic en File → New Project → Version Control → Git
 - En “Repository URL” simplemente indique el enlace del repositorio en el Github
- Después de crear un documento (R Script, R Markdown), haga clic en Git y haga el **commit** y, a continuación, el **push**
 - También es posible hacer el **pull** de los archivos del repositorio que fue indicado

Funciones e Iteraciones con Paquete Purr

Luiz Rodriguez Fartini 005.374.619-81

Functions, Purrr

- **Creando funciones en R**
- **Atribuyendo condiciones (“IF”)**
- **Iteraciones con Purrr (funciones map)**
- **Material para referencia.**
 - Wickham, H. & Grolemund, G. **R for Data Science**: <https://r4ds.had.co.nz/index.html>
 - <https://github.com/rstudio/cheatsheets/blob/master/purrr.pdf>