Módulo 3. Exploración, transformación y limpieza de datos para uso posterior

BARÇA INNOVATION HUB
Universitas

Exploración, transformación y limpieza de datos para su uso posterior

El extraer, cargar y transformar (ETL) alternativo

En analítica de datos, más específicamente en áreas de la analítica como el almacenamiento de datos o la creación de procesos automatizados de gestión y movimiento de datos, existe un proceso para la correcta integración y transformación en una base de datos específica. Este proceso es comúnmente conocido como ETL, correspondiente al acrónimo de los términos ingleses *extract, transform and load.*

A nivel general, este proceso consiste: en extraer la información de las distintas fuentes de datos y proveedores; hacer las transformaciones correspondientes para que encajen con el formato y los requerimientos de la base de datos a la que queremos enviar esa información, y finalmente, automatizar la carga de información a la base de datos para que pueda ser usada por los profesionales que requieran esa información. Estos procesos no suelen corresponder a las

responsabilidades de un *sport scientist,* ya que son tareas muy específicas y que requieren una formación y experiencia en el ámbito fuera del rango de habilidades de un *sport scientist*. Sin embargo, en nuestro dominio profesional también debemos llevar a cabo procesos que permitan estructurar nuestros datos de la mejor forma posible para que puedan utilizados de la manera más eficiente.

Por ejemplo, dentro del club en el que trabajas, el entrenador del equipo o un miembro de la directiva quiere conocer información sobre cómo los jugadores del equipo están evolucionando a lo largo de la temporada, ya que conocen que se están realizando test para la evaluación de su condición física. Realizan esta petición aproximadamente una vez cada dos semanas. Podrías ir a cada *software* de las herramientas que utilizaste para el registro de los resultados, descargarlos y analizarlos con herramientas como Excel o el mismo RStuido (Allaire, 2021), pero como hemos comentado, si esta petición se repite con frecuencia, debes organizar y guardar los datos de manera sistemática para que ese proceso de proveer información sea lo más eficiente posible.

No estamos hablando de utilizar bases de datos complejas si no tienes esa capacidad de almacenamiento. Un libro de Excel puede ser un recurso extremadamente valioso para poder tener la información disponible y con un formato que permita su uso posterior con garantías.

Para poder conseguir este objetivo, te enseñaremos a usar un proceso con las mismas siglas que el ETL descrito anteriormente, pero en este caso será específico a las necesidades y requerimientos de un *sport scientist*.

- Exploración de los datos: una parte de la exploración forma parte del análisis descriptivo que hemos desarrollado en el módulo anterior, pero en este caso hablamos del formato de nuestros datos (cómo se organiza la tabla de datos, qué tipo de variables tenemos, cómo detectar errores, etc.).

- Transformación de los datos: con base en el paso anterior, debemos cambiar el formato de los datos o la configuración de la tabla para que sea lo más eficiente posible.

- Limpieza de los datos: se trata de corregir los errores que hemos detectado, para que los análisis posteriores sean los correctos.

En este documento verás cuáles son las opciones más comunes de cada uno de los pasos del proceso, pero el abanico de posibilidades es muy extenso y sobre todo muy relacionado con el tipo de datos con los que trabajes (series temporales, datos, resumen, etc.). El seguimiento del material de video ofrecido durante los módulos es fundamental

para conocer las posibilidades y opciones que ofrece RStudio (Allaire, 2021) en este aspecto.

Selecciona las tres respuestas correctas. El proceso comúnmente conocido como ETL consiste en:

- [ ] Extraer la información de las distintas fuentes de datos y proveedores.
- [ ] Hacer las transformaciones correspondientes para que encajen con el formato y los requerimientos de la base de datos a la que queremos enviar esa información.
- [ ] Automatizar la carga de información a la base de datos para que pueda ser usada por los profesionales que requieran esa información.
- [ ] Presentar la información obtenida a los proveedores que precisan de la misma para su posterior análisis.

SUBMIT

Organización de los datos

Antes de profundizar en cada uno de los pasos mencionados anteriormente, vamos a enfatizar la importancia del registro de los datos. En este caso, no nos enfocaremos tanto en su consistencia y selección de herramientas que nos permitan confiar en la calidad del dato. Por el contrario, nos centraremos en la manera de guardar esos datos, la estructura que debe tener y la consistencia en el formato de registro. Las características que destacaremos a continuación son fácilmente aplicables a archivos de registro de datos básicos como Excel, pero los mismos principios se pueden aplicar a sistemas de almacenamiento más complejos.

Puedes encontrar el mismo tipo de información en distintos formatos. Por ejemplo, puedes tener un libro de Excel, donde en la primera hoja se encuentren almacenados los datos en formato «ancho» (imagen A de la figura 1), donde cada una de las variables tiene su columna correspondiente, o formato «largo» (imagen B de la figura 1), donde existe una columna con un identificador de la variable del cuestionario y otra columna con los valores de cada una de las respuestas.

Figura 1: Registros de datos básicos

A

Jugador	Fecha	Sueño	Fatiga	Dolor
A	1/1/24	3	5	3
B	1/1/24	4	2	2
C	1/1/24	3	5	1
D	1/1/24	5	4	1

B

Jugador	Fecha	Variable	Valor
A	1/1/24	Sueño	3
B	1/1/24	Sueño	4
C	1/1/24	Sueño	3
D	1/1/24	Sueño	5
A	1/1/24	Fatiga	5
B	1/1/24	Fatiga	2
C	1/1/24	Fatiga	5
D	1/1/24	Fatiga	4
A	1/1/24	Dolor	3
B	1/1/24	Dolor	2
C	1/1/24	Dolor	1
D	1/1/24	Dolor	1

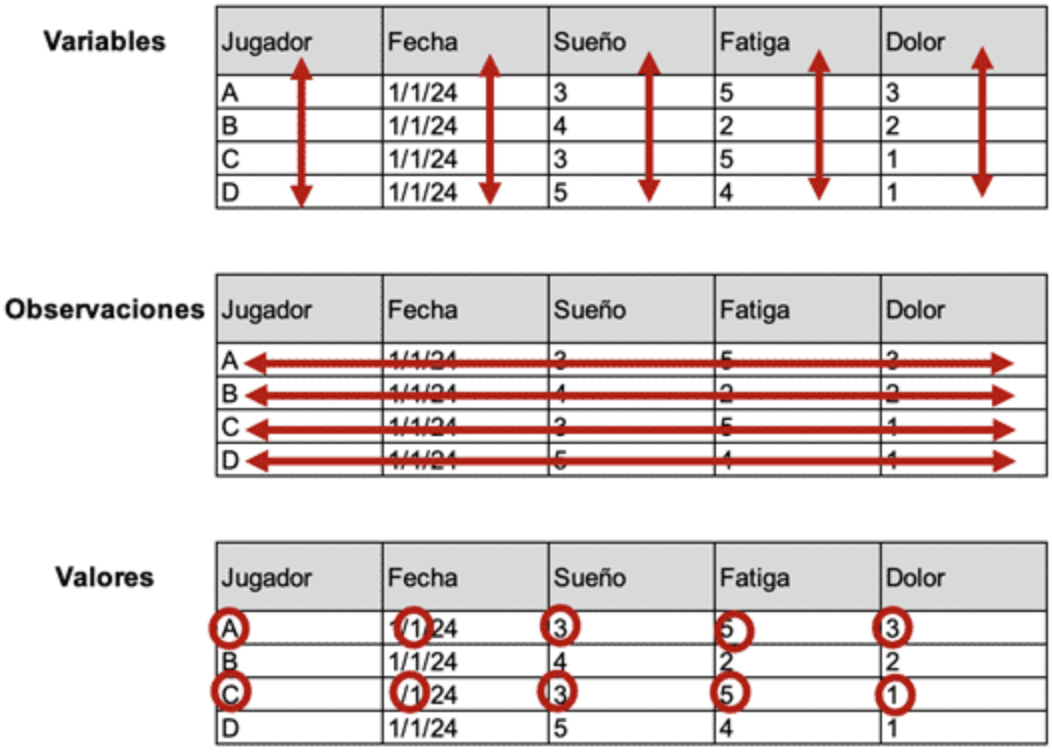
Fuente: adaptación propia con base en Grolemund y Wickham, 2017.

Según el tipo de análisis o visualización que necesites realizar, será conveniente un formato u otro, pero existen unas reglas determinadas para considerar una base de datos bien organizada (Grolemund y Wickham,2017).

- Cada variable debe tener su propia columna.
- Cada observación debe tener su propia fila.
- Cada valor debe tener su propia celda.

Con base en la imagen anterior, puedes ver a qué se refieren cada una de estas reglas con la siguiente imagen (figura 2).

Figura 2: Reglas para una base de datos organizada



Fuente: adaptación propia con base en Grolemund y Wickham, 2017.

Una vez que tienes estas primeras reglas cumplidas, hay una serie de consideraciones que debes tener en cuenta (Broman y Woo, 2018).

- Ser consistente: con esto nos referimos a recoger la información de la misma manera día tras día. Por ejemplo, si en el cuestionario que hemos mencionado anteriormente tienes una columna con la variable «Nombre» o «Jugador», debes verificar que el nombre que registres sea siempre el mismo. No debes usar abreviaturas ni cambios en el orden de nombre o apellido, etc.

- Cruce de datos: esta funcionalidad la abordaremos más adelante en otros cursos, pero ser consistente es fundamental para el funcionamiento eficiente si buscas utilizar información de distintas fuentes de datos. La práctica ideal es utilizar identificadores fijos para cada uno de los jugadores o atletas y que estos sean los mismos en cualquiera de los sistemas de registros de datos que utilices.

- Fechas: se trata de uno de los formatos con habituales problemas en su consistencia, ya que los distintos *software* de registro de datos utilizan variedad de formas en su registro. Ser consistente en ello te será de gran ayuda para evitar que la transformación y limpieza de datos sea muy costosa.

- Escoger nomenclatura correcta: debes evitar caracteres especiales y de poder ser, también espacios y letras mayúsculas. Esto debe aplicarse ya sea en el nombre de la variable/columna o la misma celda donde tienes los valores. Por ejemplo, si has registrado valores antropométricos y tienes una variable que da información sobre el porcentaje de masa muscular, en lugar de llamar a tu variable «% Masa Muscular», ya que contiene caracteres especiales y espacios, puedes optar por «pct_masa_muscular».

- Si las nomenclaturas se vuelven muy poco intuitivas, puedes crear documentos que sirvan de diccionarios con la explicación de dichas variables.

- Evitar celdas vacías: en nuestra profesión estamos acostumbrados a tratar con frecuencia con datos no registrados. Por ejemplo, en toda una observación para un jugador concreto a veces faltan datos de todas las variables, mientras que tenemos una sola variable o un día completo para todo el equipo. Debes decidir cómo especificar esos datos no registrados para poder identificarlos correctamente en los análisis posteriores.

    - Debes evitar también tipos de archivos como los que detallamos en el primer módulo, donde al inicio de la hoja Excel existen X filas con información y a continuación hay una tabla de datos. Este tipo de formato no es ideal para el análisis.

- No realizar cálculos en el archivo principal: debes mantener el archivo donde registras tus datos lo más limpio posible, es decir, sin procesar. En cualquiera de los cálculos que desees realizar será mejor opción utilizar RStudio (Allaire, 2021), de esta forma también evitarás que se pueda perder información.

Si respetas estas características, partirás de una buena estructura para desarrollar los siguientes pasos.

Que cada variable debe tener su propia columna, cada observación debe tener su propia fila y cada valor debe tener su propia celda son reglas determinadas para:

- ◯ Una base de datos bien organizada.
- ◯ Un registro de datos.
- ◯ Una exploración de datos.

SUBMIT

Exploración de los datos

La primera exploración de los datos que debes realizar es identificar su formato, ya sea si tratas con una tabla de datos y cada una de las

variables que contiene o si vas a trabajar con «objetos», vectores o listas.

- Función «class()»

Para conocer el tipo de datos con los que tratas, utilizarás la función *«class()»* esta función permite identificar la clase de los datos y cuál es su formato, para que lo puedas utilizar posteriormente de manera adecuada. Recuerda que las funciones necesitan «argumentos», es decir, un objeto al que aplicar las funciones y también una serie de argumentos que requiera cada una de las funciones. En los ejemplos posteriores, puedes ver los argumentos utilizados en estas funciones sencillas.

Por ejemplo, existen distintos tipos de formatos cuando hablamos de variables/columnas: estas pueden ser numéricas, «carácter», categóricas/factor, fechas, etc. Según el tipo de variable, se pueden realizar unos cálculos u otros, así como cada una de las funciones que utilices serán específicas al formato de datos que trates.

Figura 3: Función aplicada a diferentes objetos

Fuente: Elaboración propia

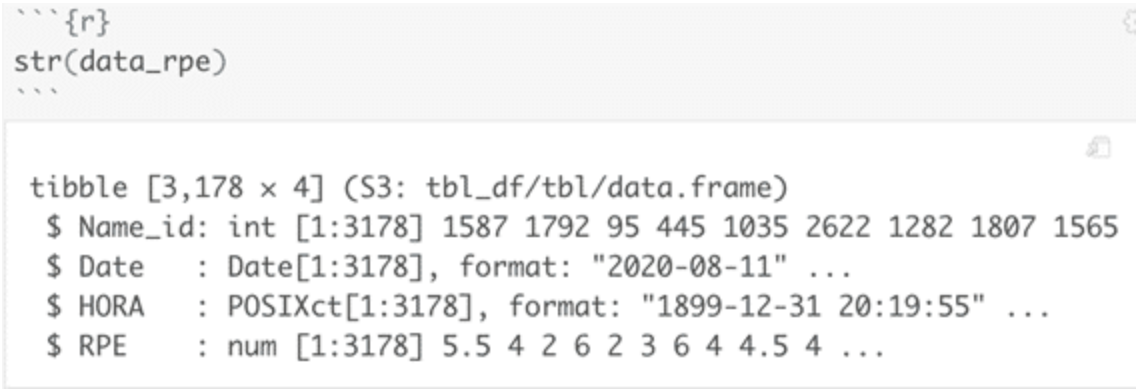
En la imagen anterior, puedes ver cómo se aplica la misma función a distintos objetos dentro del entorno de *«R»*. En primer lugar, se aplica al objeto «data_rpe». El resultado se muestra debajo del código escrito e indica que se trata de una «data_frame», uno de los formatos más comunes de tabla de datos en *«R»*. A continuación, se aplica de nuevo la función, pero en este caso a una de las variables de la tabla *«RPE»* que contiene las respuestas de los jugadores en la escala predeterminada. Por lo tanto, puedes observar que se trata de una columna que contiene valores numéricos.

En los dos últimos casos se muestra cómo se crean dos objetos. En el primero, se especifica un solo valor y se le aplica la función «ymd()», la cual verás durante el módulo, pero a grandes rasgos da formato de fecha al texto introducido de forma específica. Por su parte, se usa la función «class()» para constatar que efectivamente se trata del formato fecha, por lo que la conversión ha sido correcta. Podrías utilizar este objeto para tus informes diarios, especificando la fecha sobre la que quieres realizar el informe. Este objeto lo puedes utilizar para filtrar los datos en pasos posteriores.

Finalmente, el segundo objeto creado es un vector, ya que contiene más de un valor y sus valores son de formato «character». Aunque sean números, al escribirlos entre comillas obtienen el formato de «character». De acuerdo con el ejemplo anterior, si en tu informe no quieres que aparezcan los jugadores lesionados, podrías utilizar este vector para filtrar los jugadores que no quieras de tu fuente de datos. Para que este paso funcione correctamente, el formato de la columna de la base de datos que contiene los identificadores del jugador también debe ser «character». De no serlo, el filtro no funcionaría correctamente. De ahí la utilidad de conocer el formato de los datos.

- Función «str()»

Figura 4: Función «str()»



```
```{r}
str(data_rpe)
```
```

```
tibble [3,178 × 4] (S3: tbl_df/tbl/data.frame)
 $ Name_id: int [1:3178] 1587 1792 95 445 1035 2622 1282 1807 1565
 $ Date   : Date[1:3178], format: "2020-08-11" ...
 $ HORA   : POSIXct[1:3178], format: "1899-12-31 20:19:55" ...
 $ RPE    : num [1:3178] 5.5 4 2 6 2 3 6 4 4.5 4 ...
```

Fuente: Elaboración propia

En este extracto de código y su resultado puedes ver la utilidad de la función *«str()»* que viene a representar la palabra *structure*, es decir, busca conocer cómo está organizada la base de datos que desees utilizar. De esta forma, te dará información de si es una tabla u otro tipo de objeto, el número de filas y columnas (3178 filas y 4 columnas) y también las características de cada una de las variables. Además, verás las primeras observaciones de cada variable para poder tener una idea de los datos que aparecerán.

- Función *«ncol()»*: esta función permite conocer el número de columnas de la tabla de datos, por si no es necesario conocer toda la información que proporciona la función *«str()»*.

- Función *«nrow()»*: en este caso se trata de la misma funcionalidad que la anterior, pero en este caso obtendrías el número de filas.

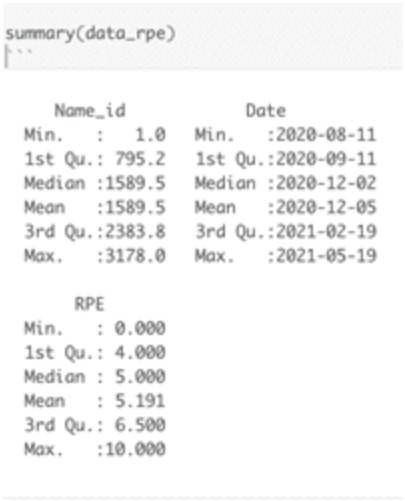
- Función *«dim()»*: es una combinación de las dos anteriores, te dará el número de filas y de columnas.

- Función *«colnames()»*

  Esta función permite conocer el nombre de las columnas de la tabla de datos si la utilizas únicamente con el argumento del objeto del cual quieres conocer los nombres de las columnas. Esta misma función se puede utilizar para renombrar las columnas, pero esto corresponde al apartado de transformación de los datos.

- Función *«summary()»*

Figura 5: Función *«summary()»*



Fuente: Elaboración propia

Como puedes ver en la imagen, la función permite tener un resumen numérico de las variables de la tabla de datos. La información puede ser útil en alguna de las variables, como el «RPE» en este caso, pero irrelevante en el caso de la columna «Name_id», ya que aunque se traten de números, son identificadores, por lo que los parámetros descriptivos como la media o el máximo carecen de valor informativo.

- Función *«head()»*

  Muestra las primeras 6 filas de la tabla de datos y permite obtener una primera vista de los datos con los que tratas.

- Función *«unique(data_rpe$Date)»*

Esta función permite, para una de las columnas de la tabla de datos o un objeto como un vector, conocer cuántos valores únicos existen en esa variable. Por ejemplo, en el caso de una tabla con información *«RPE»*, habrá multitud de observaciones con la misma fecha, ya que cada jugador que conteste el cuestionario el mismo día tendrá asociada la misma fecha. Si deseas ver en qué fechas (únicas) se contestó el cuestionario, debes utilizar la función *«unique()».*

- Función *«view()»*

  Esta función es de gran utilidad cuando te inicias en el uso de RStudio, puesto que te permite visualizar la tabla u objeto en un formato mucho más parecido al que puedes estar acostumbrado, en una ventana separada en la interfaz de RStudio (Allaire, 2021).

Transformación de los datos

Este punto en el proceso de análisis de datos puede ser tan complejo como avanzado sea el proyecto en el que te encuentres trabajando, sin embargo, conocer las funcionalidades básicas que te permitan modificar la estructura y contenido de tu fuente de datos te permitirá grandes resultados desde los primeros proyectos. A continuación, se

expondrá un listado y una breve descripción de las funciones más comunes de la transformación de datos. El contenido de video de este módulo va a ser fundamental para ver la aplicabilidad de estas funciones en datos del contexto del *sport scientist* y permitirte avanzar en el proceso.

Debemos recordar que de acuerdo con el tipo de función que quieras utilizar, como habrás visto en los videos, deberás instalar un paquete u otro.

- Transformación de las columnas
    - *«colnames()»:* mencionada con anterioridad, esta función permitirá cambiar el nombre de las columnas para su uso posterior.
    - *«as.factor()»/«as.numeric()»*: permite cambiar el formato de los valores de la columna al formato deseado.
    - *«arrange()»:* posibilita ordenar la columna según los parámetros que quieras.
- Reducción o selección de variables y observaciones
    - *«select()»:* permite seleccionar el número de columnas que desees.

- *«filter()»:* en este caso se filtrarán las observaciones que cumplan con los requisitos que busques.

- Creación de nuevas columnas

  - *«mutate()»:* es una de las funciones más utilizadas al empezar a trabajar con una base de datos para crear nuevas columnas o derivadas de columnas existentes.

- Cambio en la organización de los datos dentro de la tabla

  - *«pivot_longer()»:* permite realizar transformaciones de la organización de los datos como la que has visto en la primera imagen de este módulo, pasar de un formato «ancho» a un formato «largo», donde existen variables con identificadores y variables con los valores correspondientes a estos identificadores.

  - *«pivot_longer()»:* realiza lo opuesto a la variable anterior.

Es necesario volver a destacar que el tipo de transformación de datos que debes realizar está muy relacionado con el objetivo que persigas,

por lo que existirán multitud de posibilidades y funciones.

Limpieza de los datos

De la misma manera en la que hemos descrito el proceso de transformación de datos, cuando hablamos de limpieza de datos los mismos principios se aplican. Según el formato «final» que quieras dar a los datos y el uso específico, deberás utilizar los principios y funciones de limpieza de datos de una manera u otra.

Un ejemplo para ilustrarlo es cómo lidias con datos no registrados, por errores en la lectura de los datos de la tecnología que utilizas u otros motivos, como falta de respuesta en un cuestionario de uno de los jugadores. Pues bien, la pregunta inicial entonces es qué quieres realizar con los datos que dispones.

Caso 1: un jugador no tiene datos para la sesión MD-3. Aunque participó en la sesión de entrenamiento, su dispositivo GPS no registró actividad. Si al final de la semana quieres comparar la diferencia entre el volumen semanal actual y el del último mes, tener una sesión no registrada podría estar subestimando los valores de esta última semana, por lo que podrías optar por asignar valores a la sesión sin registro. Hay distintos razonamientos y opciones para usar un tipo de

asignación u otra, pero podrías decidir utilizar el promedio de sus otras sesiones MD-3 a lo largo de la temporada.

Caso 2: un jugador no tiene datos para la sesión MD-3. Aunque participó en la sesión de entrenamiento, su dispositivo GPS no registró actividad. Si al finalizar la sesión quieres tener un resumen grupal de la manifestación condicional del equipo en ciertas variables, omitirás que haya un jugador que no tenga datos, puesto que si a ese jugador le asignaras valores «0», los datos no estarían representando la realidad de la sesión de entrenamiento porque el jugador sí participó en la sesión.

Caso 3: un jugador no tiene datos para la sesión MD-3, ya que no participó por molestias físicas. Si quieres realizar el mismo análisis que en el caso 1, en este caso deberías optar por indicar que todos los valores para ese día son «0», a fin de representar con exactitud en los cálculos que ese día no hubo actividad.

Este ejemplo permite mostrar la especificidad de cada uno de los casos con los que puedas llegar a trabajar en algún momento, las funciones de RStudio (Allaire, 2021) únicamente te facilitarán obtener el resultado que deseas. Como en el paso anterior, estas son las principales funciones que se utilizan durante el proceso de limpieza de datos.

- Datos ausentes

  - Función *«na.omit()»*: en «R», si hay datos incompletos en alguna de las columnas, constan como «NA». Esta función permite eliminarlos. La función *«complete.cases()»* pretende lograr el mismo objetivo al seleccionar únicamente las columnas que tengan todas las observaciones completas.

  - Función *«is.na()»*: identifica las observaciones donde faltan datos.

- Datos extremos

  - Debes usar funciones que describan los datos para poder detectar los valores que estén fuera de rangos aceptables y filtrarlos consecuentemente.

- Limpieza de columnas con texto

  - Función *«clean_names()»*: asigna a las columnas nombres con un formato sencillo que permita su uso de manera más eficiente.

  - Función *«gsub()»*: detecta patrones en el texto de las observaciones y lo reemplaza con el formato deseado.

¿Qué formato determina la manera en que debes utilizar los principios y funciones de limpieza de datos?

- Según el formato «final» que quieras dar a los datos y el uso específico, deberás utilizar los principios y funciones de limpieza de datos de una manera u otra.
- Según el formato «provisorio» que quieras dar a los datos y el uso general, deberás utilizar los principios y funciones de limpieza de datos de una única manera en todos los casos.

SUBMIT

CONTINUAR

Referencias

Allaire, J. J. (2021). *RStudio* (09.0). [entorno de desarrollo integrado para el lenguaje de programación]. Posit. https://posit.co/products/open-source/rstudio-server/

Broman, K. W. y Woo, K. H. (2018). Data Organization in Spreadsheets. *The American Statistician,* 72(1), pp. 2-10. https://doi.org/10.1080/00031305.2017.1375989

Grolemund, G. y Wickham, H. (2017). R for Data Science. O'Reilly Media.

CONTINUAR