

Módulo 4. Informes automatizados para la comunicación de resultados



☰ Informes automatizados para la comunicación de resultados

☰ Referencias

Informes automatizados para la comunicación de resultados

Hemos repasado, en cursos anteriores, cuál es el proceso de trabajos en el análisis de datos de rendimiento físico, cómo profundizar en cada uno de ellos y escoger cuáles son las mejores herramientas o funciones para la importación, exploración, transformación, análisis y visualización de datos.

RStudio se ha presentado como una herramienta de gran versatilidad para responder a cada una de las necesidades planteadas, por lo tanto, es una herramienta integral que permite ejecutar cada uno de los pasos. Esta herramienta también presenta gran reproducibilidad. Con esto nos referimos a que, si pretendemos repetir un mismo proceso de análisis en múltiples ocasiones, únicamente deberemos asegurarnos de que los nuevos datos que analicemos tengan la misma estructura, y si nuestro código está bien organizado, el nuevo análisis se podrá hacer en cuestión de segundos.

Ya hablamos, en módulos anteriores, sobre las estrategias de comunicación de los resultados, cómo debemos ser lo más eficientes

posibles en la comunicación de los resultados y escoger correctamente qué información transmitir según el receptor de los análisis que realicemos. Una vez solventadas estas preguntas, debemos escoger qué herramienta permitirá comunicar los objetivos de la forma deseada.

En ocasiones, puede tratarse de una única visualización, un gráfico, que represente aquello que queremos evaluar y vaya directo al punto que queremos demostrar. En otro caso, será necesario desarrollar sistemas con mayor interacción del usuario (también conocidos como *dashboards*). Estos elementos permiten mayor flexibilidad en la exploración de los resultados y serán altamente útiles si ese es nuestro objetivo. Hemos visto, en módulos anteriores, cómo realizar visualizaciones sencillas y veremos, en módulos posteriores, cómo elaborar esos *dashboards* con interacción para los usuarios finales.

En este módulo, trataremos ese punto intermedio, aquellos informes o análisis que queremos repetir múltiples veces.

El director de la academia en la que trabajamos como Sport Scientist nos pide cierto informe sobre jugadores específicos de las distintas categorías, donde vemos los resultados de los *test* físicos a lo largo de la temporada, en donde se sitúa al jugador respecto al resto de jugadores y podemos observar cuál es su evolución. Otra situación puede ser cuando queremos ver los resultados de los *test* mensuales que realizamos y queremos compartir esa información con el cuerpo

técnico. Finalmente, un caso más técnico en el que desarrollamos un modelo estadístico para realizar estimaciones del desarrollo físico de nuestros jugadores y queremos enviar los resultados y también el código utilizado para que lo revisen y así recibir la opinión del resto de los compañeros del departamento de ciencias del deporte.

Estos tipos de documentos contienen múltiples tipos de información, hay texto y descripción de cada uno de los objetivos que perseguimos. Mostramos imágenes, visualizaciones, gráficos, pero en ocasiones, también nos interesa compartir los métodos que utilizamos, como el tipo de datos que observamos y el código.

Finalmente, con estos documentos lo que perseguimos es poder convertirlos en un formato que sea útil para compartir. Un documento PDF que podamos imprimir o un documento Word para añadir comentarios, etc.

¿Cuál de las siguientes afirmaciones sobre RStudio es correcta?

RStudio es una herramienta integral que permite ejecutar cada uno de los pasos del análisis.

RStudio no es un adecuado para procesos que

requieren alta reproductibilidad.

- RStudio no permite la organización del código de manera eficiente.
- Para repetir un mismo proceso de análisis en RStudio, los nuevos datos deben tener la misma estructura.
- RStudio no es versátil y no puede responder a diferentes necesidades de análisis.

SUBMIT

Como hemos destacado con anterioridad, Excel puede ser una herramienta excelente para objetivos de hasta cierto nivel de complejidad. Si tenemos nuestra base de datos organizada y disponemos de hojas de Excel con ciertos filtros o macros, podríamos automatizar este proceso hasta cierto punto. Lo mismo con herramientas más enfocadas a la visualización como Power BI o Tableau.

Sin embargo, en este curso hemos mostrado el potencial de utilizar una herramienta como RStudio para el análisis de datos: su gran

flexibilidad durante todos los pasos del proceso; la superioridad, respecto a las otras herramientas, para manejar grandes cantidades de datos, distintos tipos y orígenes de tablas; su capacidad de análisis y visualización posterior.

¿Cuál de las siguientes afirmaciones describe correctamente situaciones en las que se requiere la creación de informes en una academia de deportes?

- Compartir los resultados de los test físicos con los jugadores.
- Crear informes sobre la evaluación física de los jugadores específicos a lo largo de la temporada.
- Crear informes exclusivamente sobre las finanzas de la academia.
- Desarrolla un modelo estadístico para estimar el desarrollo físico de los jugadores y enviar los resultados junto con el código utilizado.
- Ver los resultados de los test mensuales y compartir la información con el cuerpo técnico.

SUBMIT

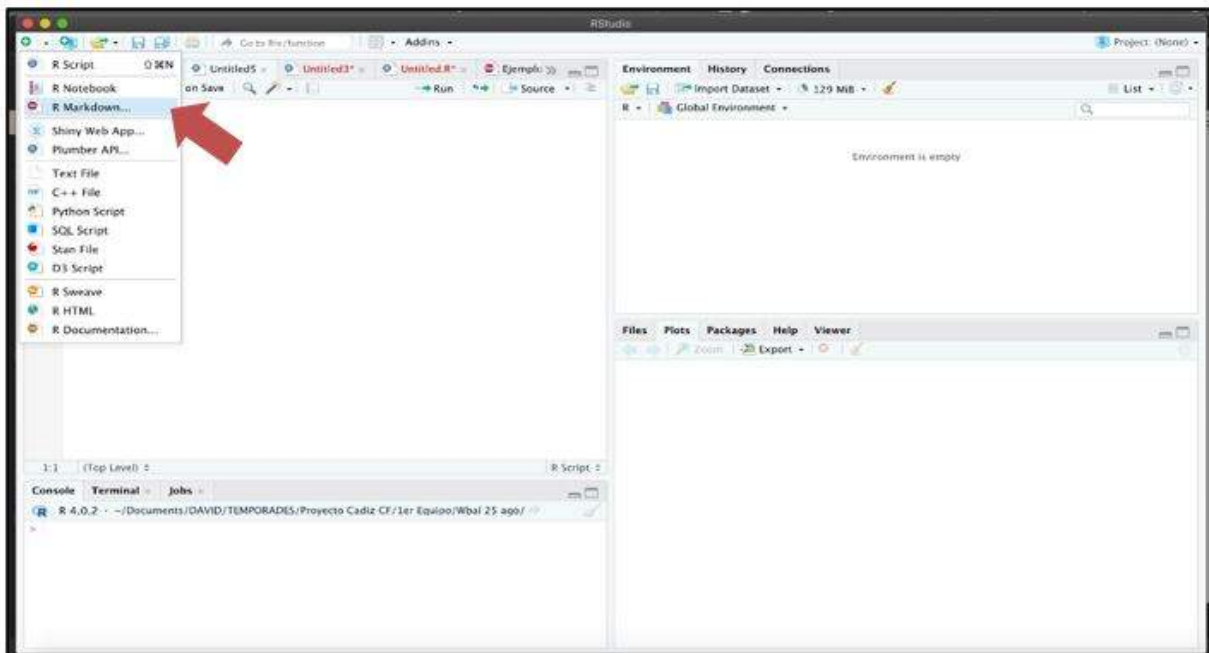
Si el análisis que queremos comunicar solo pudiera realizarse con RStudio; ¿de qué manera podríamos compartir esos informes?

Hasta ahora, hemos mostrado únicamente cómo trabajar con archivos R simples, los llamados R Script. Si utilizáramos estos archivos para los ejemplos que hemos descrito anteriormente, el proceso de compartir un informe sería muy costoso. El análisis estaría automatizado y los resultados serían obtenidos de manera eficiente, pero, ¿qué hacemos con la información? Probablemente, deberíamos escribir el texto que quisiéramos reflejar en un documento Word, descargar las visualizaciones que hemos producido en RStudio y añadirlas manualmente al documento Word y lo mismo con las tablas que produjéramos. Si es un proceso que repetimos semanalmente, estaríamos ante un proceso muy poco eficiente.

Para resolver este problema, RStudio dispone de otro tipo de archivos, llamados RMarkdown.

RMarkdown permite crear archivos donde se combina texto, imagen, código y visualización de tablas, entre otros.

Figura 1: RMarkdown



Fuente: captura de pantalla de RStudio.

La principal particularidad del archivo, cuando lo utilizamos en RStudio, es que hay ciertas zonas del archivo destinadas para cada propósito.

- Encabezado (YAML) está dentro de ---.
- Permite establecer el título del documento y el formato en el que queremos que se convierta el archivo final.

- bloque de código dentro de "".
- Todo el código que escribamos debe de estar dentro de estos bloques, también llamados «chucks». Si no está dentro, no se ejecutará.
- Texto escrito.
- Fuera del encabezado y de los bloques de código. En este documento veremos cómo cambiar el formato del texto para que tenga la forma que deseemos.

Figura 2: Zonas del archivo

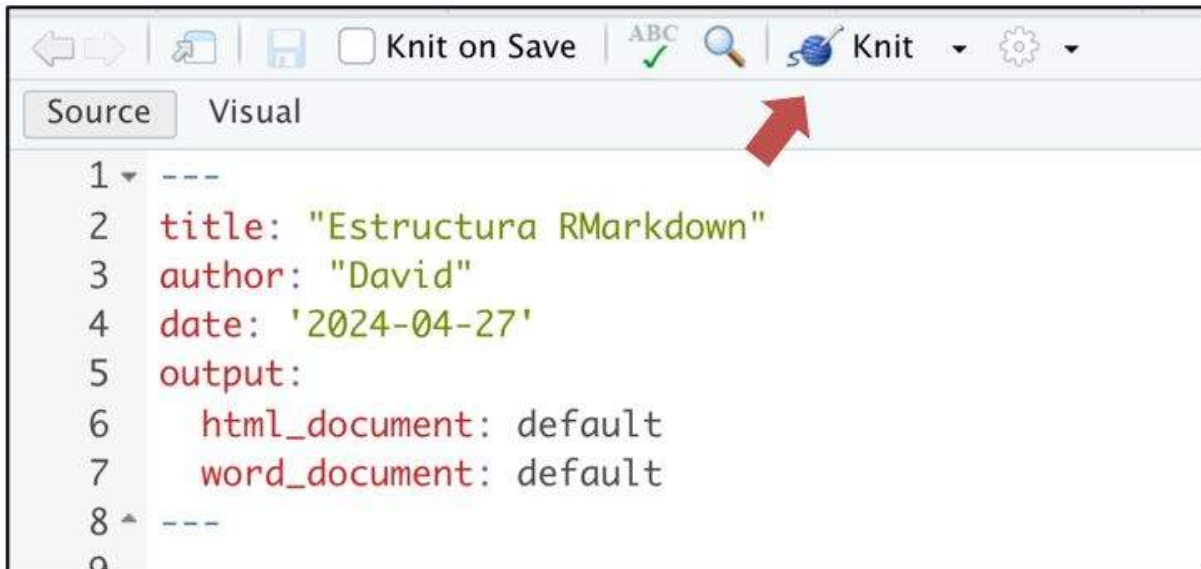


Fuente: elaboración propia.

Como vemos en la imagen anterior, en el caso de los documentos RMarkdown, los resultados de cada línea de código los vemos en la misma ventana que tenemos el archivo, en lugar de verlos en los paneles de gráficos o de la consola. Este elemento puede ser de gran

utilidad para que queden reflejados de manera organizada cada uno de los pasos que realizamos.

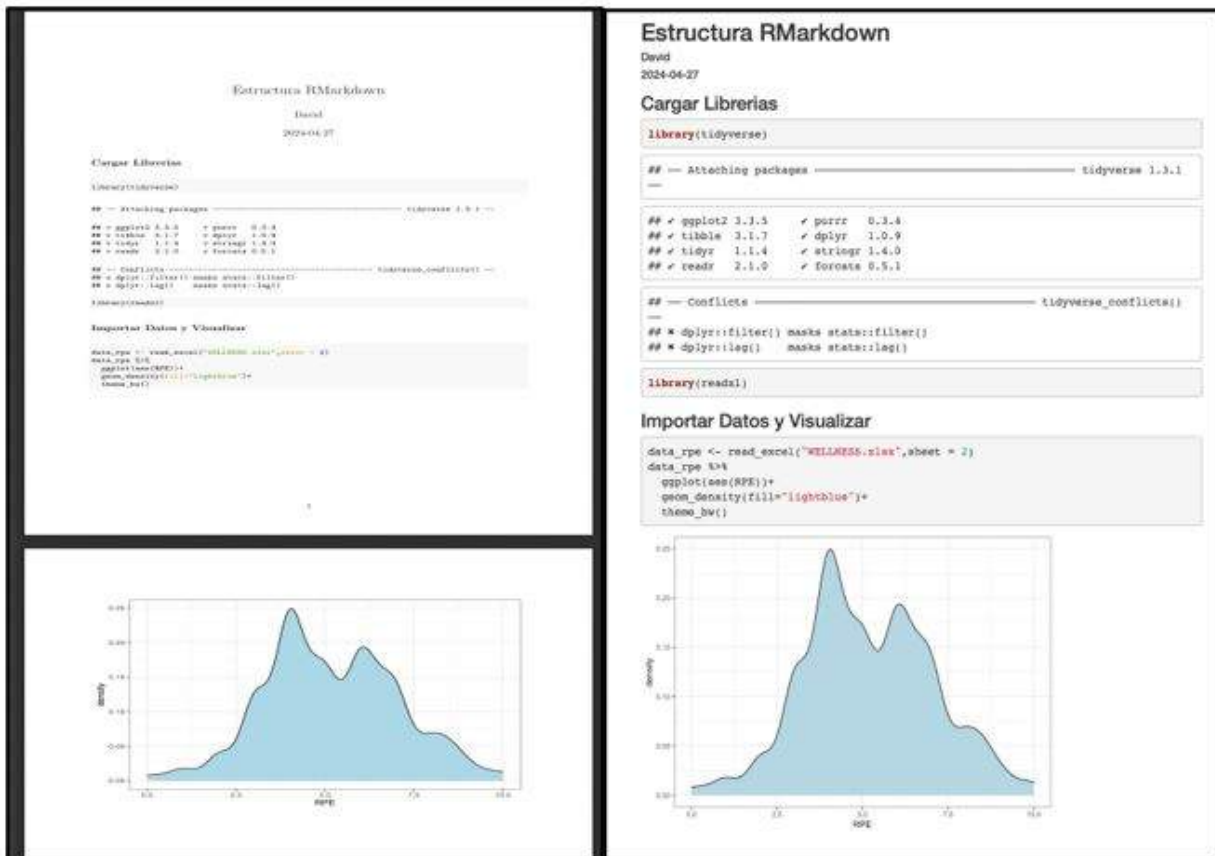
Figura 3: Knit



Fuente: elaboración propia.

Aunque a continuación destacaremos las características de cada una de las zonas del archivo, el paso final es el de la conversión del archivo al formato deseado. Una vez hayamos escrito todo el código y estructura que queremos comunicar, únicamente deberemos dar «click» al botón «knit». Este botón ejecutará el código entero y nos devolverá el archivo en el formato que queramos. A continuación, los ejemplos en PDF y HTML (archivo *web*).

Figura 4: Ejemplos en PDF y HTML



Fuente: elaboración propia.

Utilizando un archivo RMarkdown, hemos convertido un proceso que era ineficiente al “click” de un botón que realiza todos los pasos de manera automática y nos produce un documento listo para ser compartido.

Existen libros publicados escritos y diseñados exclusivamente utilizando RMarkdown, incluso tesis doctorales o páginas *web*. En

muchos procesos de selección, donde se requiere habilidades técnicas como saber utilizar lenguaje de programación, los evaluadores de la solicitud de trabajo requieren presentar un archivo RMarkdown para evaluar la capacidad del candidato para estructurar y escribir código para conseguir los resultados deseados, además de la comprensión de este al compartirlo con otros profesionales. Por lo tanto, nos encontramos ante una herramienta de gran utilidad en nuestro proceso de desarrollo en análisis de datos de rendimiento físico.

Características del bloque de código —

Como hemos descrito anteriormente, este apartado siempre deberá estar entre ````` para que se pueda ejecutar. Sin embargo, podemos determinar parámetros que cambien el resultado final según nuestro objetivo, como hemos destacado en el ejemplo. En ocasiones, podríamos elegir compartir el código que utilizamos, si es para compañeros de nuestro departamento, pero en otras ocasiones, el objetivo será presentar documentos sin ningún tipo de código, ya que el receptor final no necesita esa información. Ese formato puede ser escogido modificando ciertos elementos dentro de `{ }`.

- `include= FALSE`. Esto hará que el código no se muestre en el documento final, aunque sí lo ejecutará, pero no se mostrarán los resultados.
- `echo=FALSE`. Mostrará los resultados, pero no mostrará el código.
- `warning=FALSE`. En ocasiones R muestra advertencias en la ejecución del código. Si no queremos que se muestre en el documento final, debemos añadirlo.
- `results='hide'`. No muestra los resultados.

- `fig.show='hide'`. No muestra los gráficos.

Características del formato de texto

De la misma manera que cambiamos el formato en los archivos Word o Excel para destacar o estructurar nuestro documento (títulos, índices, palabras destacadas, etc.), lo podremos realizar en el documento RMarkdown. Para conseguirlo, debemos escribir el texto de cierta manera a fin de conseguir el objetivo deseado en el documento producido. En las siguientes imágenes, se encuentran ejemplos de la manera en la que debemos escribir el texto en nuestro documento RMarkdown (área sombreada gris) y la manera en la que se mostrará en el documento final (área en blanco debajo).

Figura 5: Cómo escribir el texto en nuestro documento RMarkdown

Tamaño de los títulos

```
# Título 1  
## Título 2  
### Título 3  
#### Título 4
```

Título 1

Título 2

Título 3

Título 4

Fuente: elaboración propia.

Para títulos, deberemos añadir # y un espacio antes del nombre de nuestro título, cuantos más #, menor será el tamaño del título.

Figura 6: Ejemplo de estructura del texto

Estructura del texto

```
línea primera\  
segunda línea  
  
nuevo párrafo
```

línea primera
segunda línea
nuevo párrafo

Fuente: elaboración propia.

Para hacer separaciones entre líneas añadimos \. De no ser así, aunque el texto aparezca en dos líneas en el archivo RMarkdown, aparecerá en la misma línea en el documento creado.

Figura 7: Ejemplo de formato del texto

Formato del texto

```
palabra/s en *cursiva*

palabra en **negrita**

1. índice o lista
2. segundo elemento
3. tercer punto de la lista

Muestras de `código` por ejemplo la funcion `mutate()`
```

palabra/s en *cursiva*

palabra/s en **negrita**

1. índice o lista
2. segundo elemento
3. tercer punto de la lista

Muestras de `código` por ejemplo la funcion `mutate()`

Fuente: elaboración propia.

Listas indexadas de la manera que se muestra en la imagen y utilizaremos un * o 2 según si queremos resaltar el texto en cursiva o negrita. Para mostrar el texto en formato de código utilizaremos `

Figura 8: Ejemplo de enlaces

Enlaces

```
[Enlace a la app](https://davidpm.shinyapps.io/SubMIP/)
```

[Enlace a la app](https://davidpm.shinyapps.io/SubMIP/)

Fuente: elaboración propia.

Para añadir enlaces, debemos escribir el texto que queremos que se muestre entre [] y el enlace deseado entre paréntesis.

Figura 9: Ejemplo para incluir imágenes

Imágenes

```
Incluir Imagen:  
![alt text](https://upload.wikimedia.org/wikipedia/en/4/47/FC_Barcelona_%28crest%29.svg)
```



Fuente: elaboración propia.

Si queremos añadir imágenes desde un enlace, este es el formato a seguir.

Figura 10: Ejemplo para mostrar tablas de texto

Tablas de texto

```
| Player | Age |  
| :-----: | :-----: |  
| Michael | 23 |  
| Charles | 28 |  
| Bob | 18 |
```

Player	Age
Michael	23
Charles	28
Bob	18

Fuente: elaboración propia.

Si queremos mostrar tablas de texto (hay otras opciones para mostrar tablas que obtengamos mediante el código analizado, como veremos en los videos), la estructura a seguir es la indicada en el ejemplo.

¿Cuál de las siguientes afirmaciones sobre la edición de documentos en RMarkdown es correcta?

- Cuantos más # se añadan antes del nombre del título, mayor será el tamaño del título.
- Para hacer separaciones entre líneas en RMarkdown, se debe añadir .
- Para crear títulos en RMarkdown, se debe añadir # y un espacio antes del nombre del título.
- Si no se añade \ entre líneas, el texto aparecerá en la misma línea en el documento creado.
- Los documentos RMarkdown no permiten la edición de formato como en Word o Excel.

SUBMIT

CONTINUAR

Referencias

Aughrey, R. (2011). Applications of GPS Technologies to Field Sports. *International Journal of Sports Physiology and Performance*, 6, 295-310. PMID: 21911856

Barnes, C., Archer, D. T., Hogg, B., Bush, M., & Bradley, P. S. (2014). The Evolution of Physical and Technical Performance Parameters in the English Premier League. *International Journal of Sports Medicine*, 35(13), 1095-1100. doi: <http://dx.doi.org/10.1055/s-0034-1375695>

Gabbett, T. (2013). Influence of playing standard on the physical demands of professional rugby league. *Journal of Sports Sciences*, 31(10), 1125-1138. doi: 10.1080/02640414.2013.773401

CONTINUAR