

Fastbook 09

Programación en R

Lleva tus informes al siguiente nivel



09. Lleva tus informes al siguiente nivel



En el anterior fastbook tuvimos nuestro primer contacto con los ficheros R Markdown. En él aprendimos la estructura y conceptos básicos que habilitan su uso como medio para la presentación de resultados.

Este último nos lleva un paso más allá. Así, vamos a abordar aspectos avanzados que permiten elevar el nivel de calidad y profesionalidad de nuestros informes.

¡A por ello!

Autor: Juan Jiménez García

[Tablas y gráficos](#)

[Índice de contenido y customización](#)

[Conclusiones](#)

Tablas y gráficos

X Edix Educación

Como ya sabéis, las tablas y los gráficos son elementos esenciales en el mundo de analytics. Los utilizamos a diario para compartir y presentar información, y en función del objetivo, será más recomendable el uso de uno otro.

En esta primera sección vamos a entender cómo introducir tablas y gráficos de alta calidad en nuestros documentos R Markdown. Empecemos con las tablas.

Con lo que hemos aprendido hasta ahora, sabemos cómo mostrar tablas en nuestro RMarkdown (simplemente imprimiéndolas en los chunks de código R). Sin embargo, tal y como se muestra en la siguiente imagen, esa forma de hacerlo no tiene una apariencia demasiado agradable.

AnalisisVentaDiaria

Juan Jiménez

18/4/2021

En el presente RMarkdown vamos a analizar el dato de venta de las tiendas 24512 y 14526 entre los años 2017 y 2020

Vamos a comenzar leyendo y tratando el dato

```
df <- read.csv("data/venta_diaria.csv")

#Ordenamos por fecha
df <- df[order(df$DIA),]

#Inicializacion de los indices
row.names(df) <- NULL

df <- tibble(df)

#Eliminamos los NAs y construimos la variable importe
df <- df %>%
  mutate(UNIDADES = replace(UNIDADES, is.na(UNIDADES), mean(UNIDADES, na.rm=T))) %>%
  mutate(PRECIO = replace(PRECIO, is.na(PRECIO), mean(PRECIO, na.rm=T))) %>%
  mutate(IMPORTE = UNIDADES * PRECIO)
```

Tras ello, ejecutamos las funciones head y summary para entender la estructura

```
## # A tibble: 6 x 6
##   PRODUCTO      TIENDA DIA      UNIDADES PRECIO IMPORTE
##   <chr>        <int> <chr>      <dbl>   <dbl>   <dbl>
## 1 Manzana a granel 14526 2017-07-31     823.    2.99   2461.
## 2 Manzana a granel 24512 2017-07-31     742     2.99   2219.
## 3 Manzana a granel 14526 2017-08-01     743.    2.99   2222.
## 4 Manzana a granel 24512 2017-08-01     696     2.99   2081.
## 5 Manzana a granel 14526 2017-08-02     769     2.99   2299.
## 6 Manzana a granel 24512 2017-08-02     678     2.99   2027.
```

A continuación, vamos a formarnos en el uso de dos paquetes R que nos permiten mejorar la presentación de nuestras tablas cuando creamos documentos R Markdown. Empezaremos con la librería **kableExtra** y continuaremos con **DT**.

kableExtra

El uso de tablas kable está altamente recomendado cuando trabajamos con la visualización de pequeñas tablas, es decir, que no poseen demasiadas filas.

Antes de nada, tendremos que realizar la instalación ejecutando el comando
`install.packages("kableExtra")`.

La primera función que debemos aprender es `tbl()`, la cual recibe un dataframe de R y lo convierte en una tabla HTML/latex sin formato. Aunque su apariencia siempre va a resultar básica, en sus parámetros ya podemos configurar ciertos aspectos relacionados con la presentación. A continuación, se listan los más importantes.

Función <code>tbl()</code>		
Parámetro	Clase	Definición
<code>x</code>	Dataframe	Dataframe que se quiere visualizar y transformar a formato tabla kable.
<code>digits</code>	Vector	Vector con el número de decimales a mostrar en cada columna. Si se introduce un solo valor, se aplicará a todas las columnas.
<code>row.names</code>	Boolean	Indica si mostrar o no en la tabla los nombres de las filas.
<code>col.names</code>	Vector	Permite modificar los nombres de las columnas en caso de que no queramos visualizar los que vienen en el dataframe.
<code>caption</code>	Character	Permite introducir una leyenda que acompañe a la tabla.

El siguiente carrusel de imágenes muestra el resultado que obtenemos al utilizar los anteriores parámetros.

```
head(df) %>% kbl()
```

PRODUCTO	TIENDA	DIA	UNIDADES	PRECIO	IMPORTE
Manzana a granel	14526	2017-07-31	823.14	2.99	2461.19
Manzana a granel	245122	2017-07-31	742.00	2.99	2218.58
Manzana a granel	14526	2017-08-01	743.26	2.99	2222.35
Manzana a granel	245122	2017-08-01	696.00	2.99	2081.04
Manzana a granel	14526	2017-08-02	769.00	2.99	2299.31
Manzana a granel	245122	2017-08-02	678.00	2.99	2027.22

```
head(df) %>% kbl(digits = c(0,0,0,0,2,0))
```

PRODUCTO	TIENDA	DIA	UNIDADES	PRECIO	IMPORTE
Manzana a granel	14526	2017-07-31	823	2.99	2461
Manzana a granel	245122	2017-07-31	742	2.99	2219
Manzana a granel	14526	2017-08-01	743	2.99	2222
Manzana a granel	245122	2017-08-01	696	2.99	2081
Manzana a granel	14526	2017-08-02	769	2.99	2299
Manzana a granel	245122	2017-08-02	678	2.99	2027

```
head(df) %>% kbl(digits = c(0,0,0,0,2,0),  
row.names=TRUE)
```

PRODUCTO	TIENDA	DIA	UNIDADES	PRECIO	IMPORTE
1Manzana a granel	14526	2017-07-31	823	2.99	2461
2Manzana a granel	245122	2017-07-31	742	2.99	2219
3Manzana a granel	14526	2017-08-01	743	2.99	2222
4Manzana a granel	245122	2017-08-01	696	2.99	2081
5Manzana a granel	14526	2017-08-02	769	2.99	2299
6Manzana a granel	245122	2017-08-02	678	2.99	2027

```
head(df) %>% kbl(digits = c(0,0,0,0,2,0),  
row.names = FALSE,  
col.names = c("Producto","Tienda","Dia","Unidades","Precio","Importe"))
```

Producto	Tienda	Dia	Unidades	Precio	Importe
Manzana a granel	14526	2017-07-31	823	2.99	2461
Manzana a granel	245122	2017-07-31	742	2.99	2219
Manzana a granel	14526	2017-08-01	743	2.99	2222
Manzana a granel	245122	2017-08-01	696	2.99	2081
Manzana a granel	14526	2017-08-02	769	2.99	2299
Manzana a granel	245122	2017-08-02	678	2.99	2027

```
head(df) %>% kbl(digits = c(0,0,0,0,2,0),
                    row.names = FALSE,
                    col.names = c("Producto","Tienda","Dia","Unidades","Precio","Importe"),
                    caption = "Previsualización de tabla")
```

Previsualización de tabla

Producto	Tienda	Dia	Unidades	Precio	Importe
Manzana a granel	14526	2017-07-31	823	2.99	2461
Manzana a granel	24512	2017-07-31	742	2.99	2219
Manzana a granel	14526	2017-08-01	743	2.99	2222
Manzana a granel	24512	2017-08-01	696	2.99	2081
Manzana a granel	14526	2017-08-02	769	2.99	2299
Manzana a granel	24512	2017-08-02	678	2.99	2027

Una vez hemos obtenido la tabla en formato kable básico, podemos usar la función `kable_styling()` para mejorar su presentación.

En la siguiente imagen se muestra el resultado cuando la aplicamos de forma directa.

```
kable_table <- head(df) %>% kbl(digits = c(0,0,0,0,2,0),
                                    row.names = FALSE,
                                    col.names = c("Producto","Tienda","Dia","Unidades","Precio","Importe"),
                                    caption = "Previsualización de tabla")
```

```
kable_table %>% kable_styling()
```

Previsualización de tabla

Producto	Tienda	Dia	Unidades	Precio	Importe
Manzana a granel	14526	2017-07-31	823	2.99	2461
Manzana a granel	24512	2017-07-31	742	2.99	2219
Manzana a granel	14526	2017-08-01	743	2.99	2222
Manzana a granel	24512	2017-08-01	696	2.99	2081
Manzana a granel	14526	2017-08-02	769	2.99	2299
Manzana a granel	24512	2017-08-02	678	2.99	2027

Además, tal y como se desglosa en la siguiente tabla, existen ciertos parámetros que nos permiten personalizar la apariencia.

Función kable_styling()

Parámetro	Clase	Definición
x	Dataframe	Tabla kable cuyo estilo quiere ser modificado.
bootstrap_options	Vector	Permite activar ciertos aspectos relacionados con la apariencia de la tabla. Los valores más importantes son: basic, striped, bordered y hover. Son combinables entre ellos.
full_width	Boolean	Indica si la tabla debe ocupar o no todo el ancho de la pantalla.
position	Vector	Permite indicar en qué posición de la página se ha de introducir la tabla. Los posibles valores son: left, center, right, float_right y float_left.
font_size	Numeric	Permite aumentar y disminuir el tamaño de letra de las tablas.
html_font	Character	Permite modificar la fuente de letra usada en las tablas.

Antes de acabar con las tablas kable, tenemos que hablar de los kable themes.

El paquete kableExtra posee un conjunto de funciones que aplican de la misma forma que kable_styling (tienen hasta los mismos parámetros), además de un tema. Es decir, añaden ciertas customizaciones predefinidas de visualización y apariencia.

Los nombres de estas funciones son:

- *kable_paper()*
- *kable_classic()*
- *kable_classic_2()*
- *kable_minimal()*
- *kable_material()*
- *kable_material_dark()*

DT

Debido a su capacidad para realizar filtros y búsquedas, el uso del paquete DT presenta grandes ventajas cuando trabajamos con la visualización de grandes tablas, es decir, que poseen una importante cantidad de filas.

Antes de nada, al igual que con kableExtra, tendremos que realizar la instalación ejecutando el comando *install.packages("DT")*.

En este caso, la función que nos permite transformar los dataframes R en una tabla con calidad visual es ***datatable()***. En la siguiente imagen se muestra su resultado cuando la aplicamos de forma directa.

df %>% DT::datatable()						
			Search: <input type="text"/>			
Show	10	▼ entries	TIENDA	DIA	UNIDADES	PRECIO
1	Manzana a granel		14526	2017-07-31	823.14	2.99
2	Manzana a granel		24512	2017-07-31	742	2.99
3	Manzana a granel		14526	2017-08-01	743.26	2.99
4	Manzana a granel		24512	2017-08-01	696	2.99
5	Manzana a granel		14526	2017-08-02	769	2.99
6	Manzana a granel		24512	2017-08-02	678	2.99
7	Manzana a granel		14526	2017-08-03	745	2.99
8	Manzana a granel		24512	2017-08-03	630	2.99
9	Manzana a granel		14526	2017-08-04	785	2.99
10	Manzana a granel		24512	2017-08-04	558	2.99

Showing 1 to 10 of 1,899 entries

Previous 1 2 3 4 5 ... 190 Next

Además, tal y como se desglosa a continuación, existen ciertos parámetros que nos permiten personalizar la apariencia.

Función datatable()		
Parámetro	Clase	Definición
data	Dataframe	Dataframe que se quiere visualizar y transformar a formato tabla kable.

Función datatable()

class	Vector	Permite activar ciertos aspectos relacionados con la apariencia de la tabla. Los valores más importantes son: hover, row-bordered, stripe y cell-bordered. Son combinables entre ellos.
rownames	Boolean	Indica si mostrar o no los nombres de las filas.
colnames	Vector	Permite modificar los nombres de las columnas en caso de que no queramos visualizar los que vienen definidos en el dataframe.
caption	Character	Permite introducir una leyenda que acompañe a la tabla.
filter	Character	Añade la capacidad de filtrar en base a los valores de cada columna. Asignando el valor top, los filtros se sitúan en la parte superior; mientras que asignando el valor bottom, lo hacen en la inferior.

Además de `datatable()`, es interesante conocer la función `formatRound()`, la cual nos permite asignar el número de decimales de cada una de nuestras columnas. La siguiente tabla muestra su funcionamiento.

Función formatRound()		
Parámetro	Clase	Definición
table	DT::datatable	Resultado de la función datatable() sobre el que queremos realizar el formateado.
columns	Vector	Vector con el nombre de columnas que deseamos formatear.
digits	Numeric	Número de decimales que se mostrarán en las columnas seleccionadas.

Gráficos

Como ya hemos comentado anteriormente, los gráficos son una parte esencial de la ciencia de datos. Se utilizan para presentar la información de una forma atractiva y entendible, ayudando a que la comunicación resulte más eficiente. Dado que el objetivo de los informes R Markdown es el de transmitir los resultados de análisis y modelos, la presencia de gráficos en ellos está más que justificada.

Para introducir gráficos en un fichero R Markdown basta con introducir el código R que los genera dentro de un chunk.

Índice de contenido y customización

X Edix Educación

En esta segunda parte del fastbook vamos a abordar elementos sumamente interesantes que darán a nuestros informes un aspecto muy profesional. Aprenderemos a introducir un índice de contenido dinámico y navegable, a añadir imágenes o logos como seña de marca y a customizar los colores y la apariencia global del documento para poder adaptarla a nuestras necesidades y casos de uso.

Índice de contenido

Para activar el índice de contenido de R Markdown tendremos que irnos al YAML Header e introducir ciertos parámetros en el lugar donde se define el output de salida (`html_document`). La siguiente tabla muestra el nombre de cada parámetro y su cometido.

Parámetro	Definición
<code>toc</code>	Con un valor booleano, indica si se debe introducir o no el índice de contenido.

Parámetro	Definición
toc_depth	Las secciones y subsecciones se detectan automáticamente con el uso de encabezados (h1, h2, h3, h4...), lo que en Markdown sería (#, ##, ###, #####...). El toc_depth indica cuántos niveles de profundidad debe considerar el índice de contenido.
number_sections	Por medio de un booleano sirve para activar la numeración automática de secciones y subsecciones.
toc_float	Solo es válido para HTML. Cuando se activa por medio de un booleano, el índice de contenido pasa a situarse a la izquierda de la pantalla, apareciendo en todo momento y haciendo que la navegación resulte más interactiva y cómoda.

La siguiente imagen ejemplo muestra cómo estos parámetros deben ser introducidos:

```
1 ---  
2 title: "Indice y Customización"  
3 author: "Juan Jiménez"  
4 date: "18/4/2021"  
5 always_allow_html: true  
6 output:  
7   html_document:  
8     toc: TRUE  
9     toc_depth: 2  
10    number_sections: TRUE  
11    toc_float: TRUE
```

Customización

Para empezar, resulta interesante comentar algunos aspectos muy básicos relacionados con el mundo del desarrollo web para aquellos que no os sintáis muy familiarizados con él.

HTML

CSS

HTML (*Hypertext Markup Language*) es el componente básico y principal de cualquier web. **Define su estructura y contenido.** Como ya sabéis, su visualización se realiza por medio de navegadores.

```
<!DOCTYPE html>
<html>
<body>

<h1>Encabezado</h1>

<p>Parrafo</p>

</body>
</html>
```

HTML

CSS

CSS (*Cascading Style Sheets*) es un lenguaje de diseño gráfico que nos permite crear y modificar la **presentación y apariencia de los documentos HTML**.

```
h1{  
    color:#B23412;  
    font_weight:bold;  
}  
  
p{  
    background-color:#000000;  
}
```

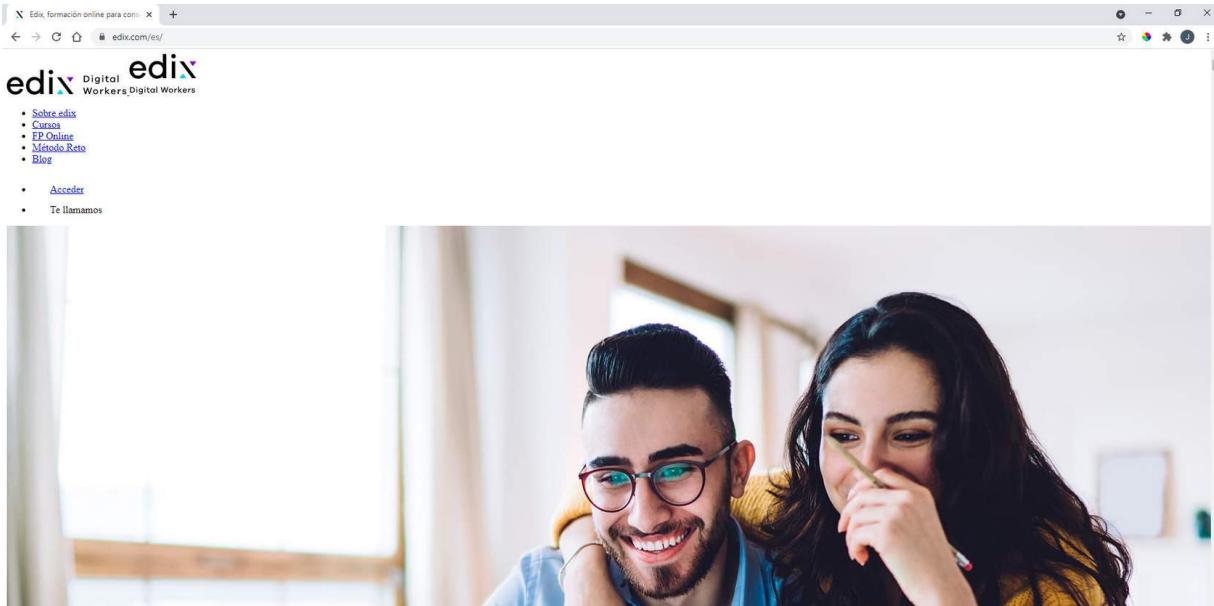
Dicho de una forma básica, HTML es el esqueleto, mientras que CSS es la piel y la ropa que lo envuelve.

Las páginas web modernas también usan otros lenguajes para poder acometer tareas de mayor complejidad, sin embargo, HTML y CSS siempre van a estar presentes.

A continuación, se muestra una página con y sin CSS para que podáis entender cuál es la aportación de cada lenguaje.

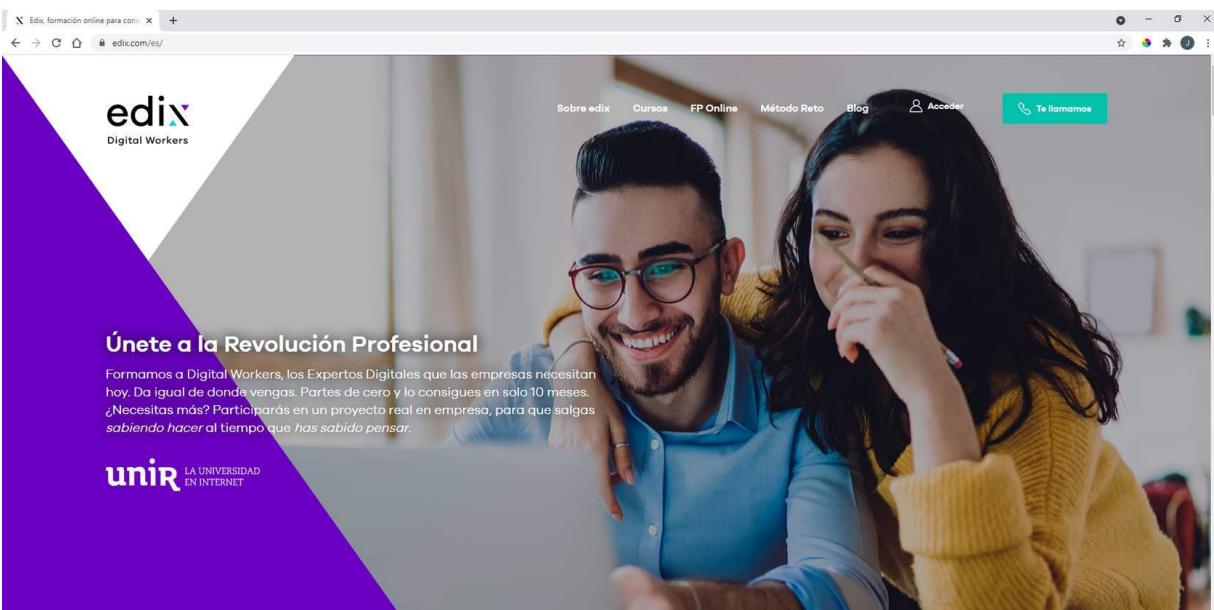
SIN CSS (SOLO HTML)

CON CSS



SIN CSS (SOLO HTML)

CON CSS



Es posible que os preguntéis cuál es la razón por la que estamos hablando de lenguajes de desarrollo web. Pues bien, como ya sabéis, estamos exportando nuestros ficheros R Markdown a formato HTML. Para poder introducir modificaciones que afecten a su apariencia y estilo necesitamos hacer uso de CSS. Para ello, tenemos dos opciones.

1

Temas

Podemos usar un conjunto de temas R Markdown que ya vienen predefinidos y que realizan modificaciones CSS de forma transparente para nosotros: default, cerulean, journal, flatly, darkly, readable, spacelab, united, cosmo, lumen, paper, sandstone, simplex y yeti.

En la página www.bootstrapcdn.com podéis visualizarlos si os resulta necesario.

Su asignación se realiza con el parámetro *theme* tal y como muestra la imagen.

```
1 ---  
2 title: "Indice y Customización"  
3 author: "Juan Jiménez"  
4 date: "18/4/2021"  
5 always_allow_html: true  
6 output:  
7   html_document:  
8     toc: TRUE  
9     toc_depth: 2  
10    number_sections: TRUE  
11    toc_float: TRUE  
12    theme: united
```

2

Ficheros CSS propios

La segunda opción consiste en crear nuestro propio CSS y linkarlo al R Markdown para que adquiera su definición de apariencia. Dicho linkado se realiza tal y como aparece en la imagen.

```
1 * ---  
2 title: "Indice y Customización"  
3 author: "Juan Jiménez"  
4 date: "18/4/2021"  
5 always_allow_html: true  
6 output:  
7   html_document:  
8     toc: TRUE  
9     toc_depth: 2  
10    number_sections: TRUE  
11    toc_float: TRUE  
12    theme: united  
13    css: customizacion.css
```

El objetivo no es ni mucho menos que aprendáis CSS si nunca lo habéis trabajado. Sin embargo, es una opción a valorar en el caso de que ya conozcáis dicho lenguaje o en el futuro os intereséis por él.

Conclusiones

X Edix Educación

En este fastbook hemos extendido los conceptos de R Markdown que fueron construidos a lo largo del anterior. Y, ahora, somos capaces de añadir gráficos, tablas, índice y customización de apariencia, todo ello con el objetivo de que nuestros reportings presenten mayor calidad y eficiencia en la transmisión de conocimiento.

Os dejo tres enlaces de interés que os pueden ayudar a alcanzar un mayor nivel de detalle por si en algún momento lo necesitáis.

- [*R Markdown: The Definitive Guide*](#), de Yihui Xie, J. J. Allaire, Garrett Grolemund.
- [*Create Awesome HTML Table with knitr::kable and kableExtra*](#)
- [*DT: An R interface to the DataTables library*](#)

Por otra parte, vamos a echar la vista atrás para recapitular todo lo que hemos visto en la asignatura.

Parte 1

Los primeros cuatro fastbooks nos introdujeron a la programación en R. Abordamos su instalación e IDE (RStudio), el tratamiento y transformación de variables tanto básicas como estructurales, el uso de sentencias para controlar el flujo del programa, la escritura y lectura de ficheros, así como la construcción de nuestras propias funciones.

Parte 2

En los fastbooks 06, 07, 08 y 09 nos hemos centrado al 100% en la práctica de analytics. Hemos aprendido a explotar y transformar la información contenida en dataframes usando diferentes paquetes, así como a construir reportes que nos ayuden a compartir nuestros resultados.

Llegados a este punto, y después del recorrido realizado, solo me queda decir ¡enhorabuena! Os deseo un próspero camino como data scientists y programadores.

¡Enhorabuena! Fastbook superado

edix

Creamos Digital Workers