Rstudio y WORD

Xavi Barber

11/2/2019

# Título principal

Muy importante aprender el significado de los “comnandos” del chunk.

Yo suelo utilizar al principio del documento:

knitr::opts\_chunk$set(echo = TRUE,message=FALSE, warning=FALSE)

## Subtítulo

Aquí vamos a ver el primer ejemplo de tabla en WORD:

library(knitr)
library(rmarkdown)
library(fpp2)
data(elecdemand)
kable(elecdemand[1:5,])

|  |  |  |
| --- | --- | --- |
| Demand | WorkDay | Temperature |
| 3.914647 | 0 | 18.2 |
| 3.672550 | 0 | 17.9 |
| 3.497539 | 0 | 17.6 |
| 3.339145 | 0 | 16.8 |
| 3.204313 | 0 | 16.3 |

Esto es solo el principio (aquí he insertado un “bigskip”).

Es muy importante para un buen funcionamiento que nos creemos una plantilla en WORD y así podamos enviarle “información” al documento, como por ejemplo saltos de página o formato de lso encabezados, etc.

Os recomiendo visitar esta página:

[enlace](https://rmarkdown.rstudio.com/articles_docx.html)

Y ahora voy a insertar un salto de página:

<br>

# Aquí empeiza una nueva sección

## Y aqui la subsección 2

Por lo que puedo seguir escribiendo

## Tabla de descriptivos

* Existen diferentes opciones de realizar una tabla de descriptivos.
* Paquetes como o funcionana muy bien en HTML o LaTeX, pero no para WORD

# MUY importante, hemos puesto en chunk: results='asis'
library(stargazer)
stargazer(data.frame(elecdemand), header=FALSE)

Como veis no hay salida!!!!!!

Otras opciones que funcionan:

library(dplyr)
temp<-data.frame(elecdemand)
temp$WorkDay<-factor(temp$WorkDay, labels=c("Laboral","Festivo"))
dt <- summarise(group\_by( temp,WorkDay),
 Media=mean(Demand, na.rm=TRUE),
 Desv=sd(Demand, na.rm=TRUE),
 Maximo=max(Demand, na.rm=TRUE))
kable(dt)

|  |  |  |  |
| --- | --- | --- | --- |
| WorkDay | Media | Desv | Maximo |
| Laboral | 4.104919 | 0.6587216 | 7.819034 |
| Festivo | 4.839323 | 0.8688780 | 9.345004 |

## Mejorando la tabla

* para mejorar la tabla obtenidad tenemos en primer ñlugar las opciones del comando
* Y como segunda opción la utilizacion del paquete

library(officer)
library(flextable)
myft2<-regulartable(dt)
tbl<-autofit(myft2)

| WorkDay | Media | Desv | Maximo |
| --- | --- | --- | --- |
| Laboral | 4.105 | 0.659 | 7.819 |
| Festivo | 4.839 | 0.869 | 9.345 |

Que como podeis ver… va mejorando cadfa vez más.

 myft <- flextable(
 data.frame(dt),
 col\_keys = c("WorkDay", "Media" , "Desv","Maximo" ))
myft <- italic(myft, j = 1)
myft <- bg(myft, bg = "#C90000", part = "header")
myft <- color(myft, color = "white", part = "header")
myft <- color(myft, ~ Maximo > 8, ~ Maximo, color = "red")
myft <- bold(myft, ~ Maximo > 8, ~ Maximo, bold = TRUE)
myft <- autofit(myft)

| WorkDay | Media | Desv | Maximo |
| --- | --- | --- | --- |
| *Laboral* | 4.105 | 0.659 | 7.819 |
| *Festivo* | 4.839 | 0.869 | **9.345** |

mola ehhh!!!

# Graficos: ggplot2

## ggplot2

* Aunque inicialmente su sintaxis se aleja de los comandos habituales del , es cierto que al final se pueden conseguir gráficos muy “porfesionales”.

ggplot2. Elegant Graphics for Data Analysis. Hadley Wickham (2016). [enalce](https://link-springer-com.publicaciones.umh.es/book/10.1007/978-3-319-24277-4)

Learn ggplot2 Using Shiny App. Keon-Woong Moon (2016).

# install.packages("ggplot2")
# load package and data
options(scipen=999) # turn-off scientific notation like 1e+48
library(ggplot2)
theme\_set(theme\_bw()) # pre-set the bw theme.
data("midwest", package = "ggplot2")
# midwest <- read.csv("http://goo.gl/G1K41K") # bkup data source

# Scatterplot
gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
 geom\_point(aes(col=state, size=popdensity)) +
 geom\_smooth(method="loess", se=F) +
 xlim(c(0, 0.1)) +
 ylim(c(0, 500000)) +
 labs(subtitle="Area Vs Population",
 y="Population",
 x="Area",
 title="Scatterplot",
 caption = "Source: midwest")

plot(gg)

