

Creando un archivo en RMarkdown

R Ladies

Taller de RMarkdonw - R Ladies Buenos Aires

Te animás a convertir este scrip en un archivo de rmarkdown? Éste código está armado a partir del material de encuentros anteriores, seguramente te va a resultar familiar. La idea es tener algunos gráficos, tablas y otros resultados para ver como quedarían en un archivo final luego de compilarlo con Knitr. Abajo te dejamos algunos tips que tal vez te ayuden.

1. Cuando elijas el tipo de archivo final, recordá que si no tenés instalado Latex no funcionará la opción PDF. El tipo de archivo final se puede cambiar en cualquier momento!
2. Recordá que el código debe ir dentro de uno o más chunks, podés crearlos desde la opción Insertar, escribiendo “{r}” “o con el atajo **Ctrl+Alt+i**. Para correr el código podés usar el atajo **Ctrl+Shift+Enter**.
3. El texto plano es todo lo que queda por fuera de los chunks, si te animás podés usar algo de markdown para darle estilo!
4. Importante: una de las herramientas útiles de este formato es poder insertar código de R en el texto plano para que al compilar aparezca el resultado directamente. Se hace usando código que aparece al compilar, ¿te animás a probar?
5. En este link hay una hoja de referencia con más información: [hoja de referencia](#) pero Google también es una gran opción.
6. Para compilarlo podés usar el atajo **Ctrl+Shift+k** o el botón que dice Knit. Está bueno hacer esto seguido para que los errores no aparezcan todos juntos al final!

Ahora si, el código

Vamos a usar la base de datos “Diamonds” que contiene información sobre 54.000 diamantes incluyendo el precio, el color, el corte y otros.

Al cargar la librería **ggplot2** tendremos disponible la base de datos.

Veamos que pinta tiene esta base de datos:

```
head(diamonds)
```

```
## # A tibble: 6 × 10
##   carat      cut color clarity depth table price      x      y      z
##   <dbl>    <ord> <ord>   <ord> <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23    Ideal     E     SI2   61.5   55   326   3.95   3.98   2.43
## 2  0.21   Premium     E     SI1   59.8   61   326   3.89   3.84   2.31
## 3  0.23     Good     E     VS1   56.9   65   327   4.05   4.07   2.31
## 4  0.29   Premium     I     VS2   62.4   58   334   4.20   4.23   2.63
## 5  0.31     Good     J     SI2   63.3   58   335   4.34   4.35   2.75
## 6  0.24 Very Good     J    VVS2   62.8   57   336   3.94   3.96   2.48
```

¿Cómo queda esta tabla al compilar?

*Tip: hay muchas opciones para darle formato a las tablas. Una posibilidad usar la función **kable** del paquete **knitr**!*

¿Vamos de nuevo?

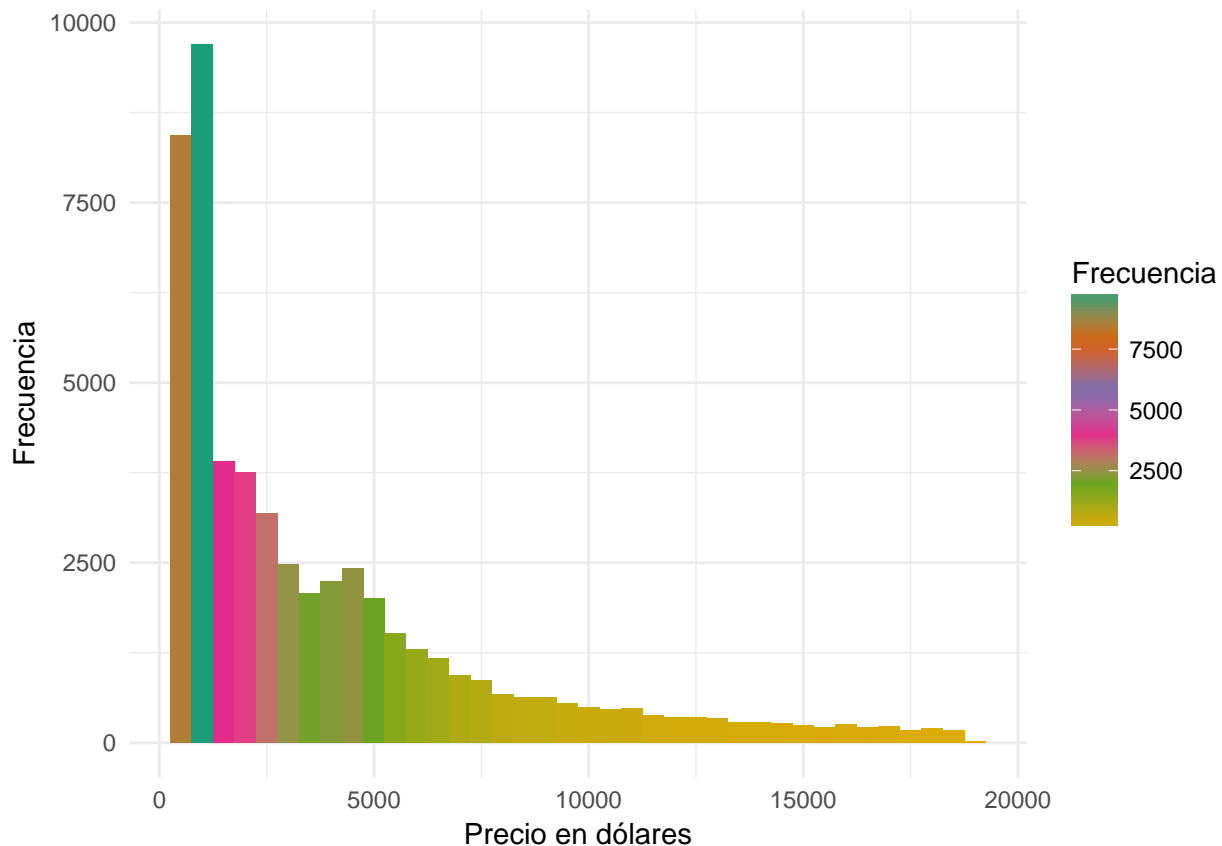
```
knitr::kable(head(diamonds))
```

carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
0.24	Very Good	J	VVS2	62.8	57	336	3.94	3.96	2.48

Podría ser interesante ver como varía el precio de los diamantes para las diferentes características.

*# Los chunks funcionan igual que un script y por eso podés poner comentarios.
Nunca te olvides # de comentar tu código! Es un regalo para tu yo futuro.*

```
ggplot(diamonds, aes(price)) +
  geom_histogram(binwidth = 500, aes(fill = ..count..)) +
  scale_fill_distiller(name = "Frecuencia", palette = "Dark2") +
  xlab("Precio en dólares") +
  ylab("Frecuencia") +
  theme_minimal()
```



Ojo con los histogramas! Son tramposos cuando cambias el binwidth.

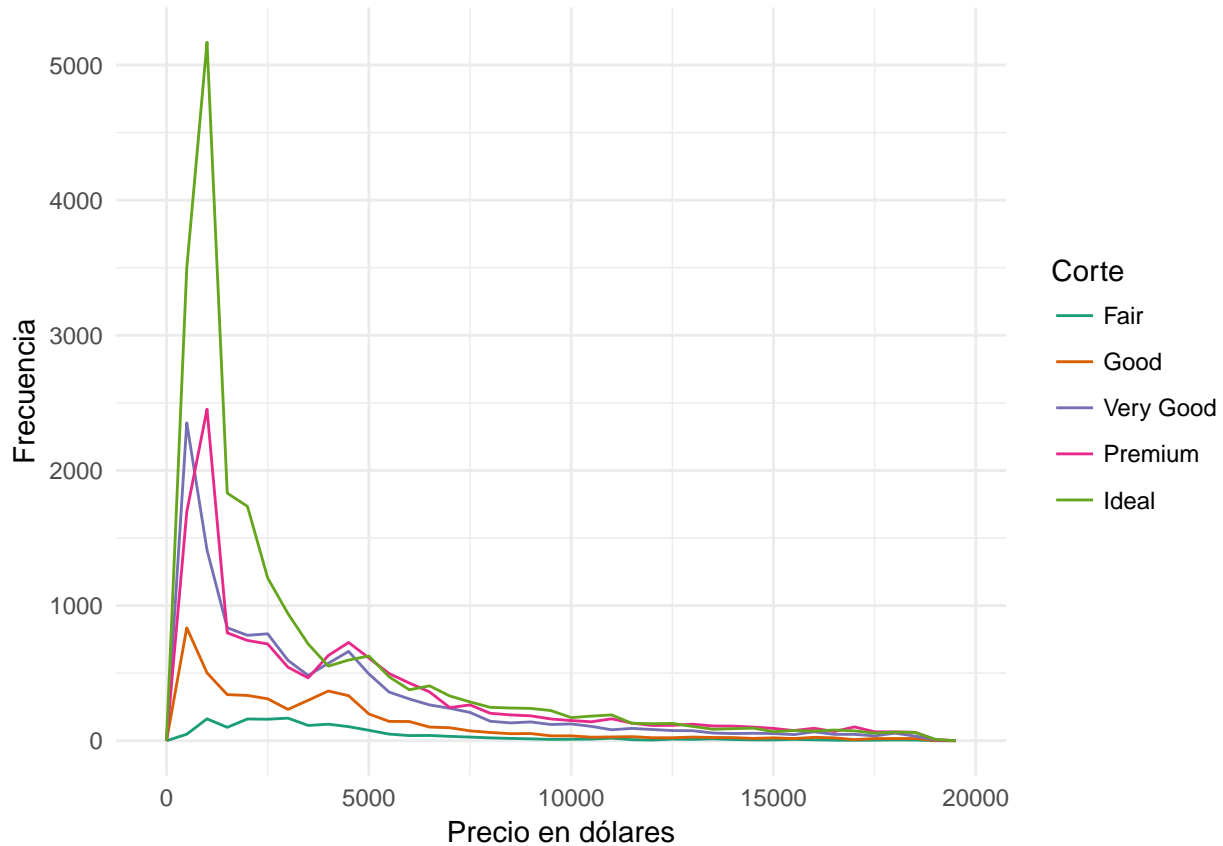
Sería interesante saber cual es el diamante más caro en la base de datos. ¿Te animás a obtener esa información y ponerla dentro del texto? Algo como “El diamante más caro es ...”

Tip: para esto puede servirte la función max()

Por ejemplo, analizando el costo de los diamantes podemos ver que el más caro tiene un precio de 18823 dolares!.

¿Cómo varía el precio para cada calidad de corte (variable cut)? ¿Realmente el precio depende de esa variable?

```
ggplot(diamonds, aes(price, color = cut)) +  
  geom_freqpoly(binwidth = 500) +  
  scale_color_brewer(name = "Corte", palette = "Dark2") +  
  xlab("Precio en dólares") +  
  ylab("Frecuencia") +  
  theme_minimal()
```



Bueno, todavía no descubrimos de que depende el precio de los diamantes, pero pueden seguir explorando la base de datos ustedes! (Pueden revisar las presentaciones de visualización de datos en github [aquí](#)).

Si tienen ganas de seguir complejizando el asunto pueden probar agregando epígrafes a las figuras, cambiarles el tamaño, agregarle más formato al texto o cambiar el tipo de archivo de salida, las opciones son infinitas!