Hoja de código R — y dónde pedir ayuda

Taller para docentes · Trujillo, 21 de agosto de 2026

Cada línea se ejecuta con **Ctrl + Enter**, con el cursor puesto en ella; todo esto está también en `taller.R`, así que no hay que teclearlo. `datos$area` = la columna *area* de la tabla *datos* · <− = «guardá esto con este nombre» · lo que va después de # es un comentario.

1. Traer los datos — se usa UNA de las tres

```
# sep = ";" porque así separa Kobo las columnas.
# Sin eso, R carga UNA sola columna ancha.

# A) Un CSV que ya está en esta computadora — el de su propia encuesta.
#    file.choose() abre el buscador: se hace clic en el archivo, sin
#    escribir rutas (en Windows llevan \ y R da error).
datos <- read.csv(file.choose(), sep = ";", fileEncoding = "UTF-8")

# B) Los datos de hoy, en vivo (la dirección ya viene escrita en taller.R)
# datos <- read.csv("https://kf.kobotoolbox.org/.../data.csv",
#                   sep = ";", fileEncoding = "UTF-8")

# C) Datos de ejemplo, sin conexión
# datos <- read.csv("datos_ejemplo.csv", sep = ";", fileEncoding = "UTF-8")
```

¿Prefieren el ratón? **Environment » Import Dataset » From Text (base)…** y poner **Separator: Semicolon**. RStudio escribe el código por ustedes.

2. Mirar antes de calcular

```
head(datos)     # primeras filas        nrow(datos)   # cuántas respuestas
names(datos)    # nombres de columnas   str(datos)    # tipo de cada columna
```

3. Quedarse con las respuestas

```
# Kobo agrega columnas técnicas (_id, _uuid, _submission_time...)
respuestas <- c("instituto", "anios_experiencia", "area",
                "estudiantes_aula", "minutos_traslado", "uso_celular")
datos <- datos[, respuestas]

# ¿Agregamos preguntas hoy? Sus columnas no están en ese vector.
# Para conservarlas: respuestas <- c(respuestas, "nombre_de_la_pregunta")
```

4. Contar, promediar, comparar

```
table(datos$area)                 # contar por familia profesional
mean(datos$anios_experiencia)     # promedio
summary(datos$minutos_traslado)   # mínimo, mediana, máximo

table(datos$area, datos$uso_celular)              # cruzar dos preguntas
tapply(datos$minutos_traslado, datos$area, mean)  # promedio por grupo
```

5. Gráficas

```
par(mar = c(5, 9, 4, 2))          # más margen: los nombres son largos
barplot(table(datos$area), horiz = TRUE, las = 1, col = "steelblue")
```

```r
hist(datos$anios_experiencia, breaks = 8, col = "steelblue")

par(mar = c(9, 5, 4, 2))
boxplot(minutos_traslado ~ area, data = datos, las = 2,
        ylab = "Minutos de traslado")
```

La **virgulilla** ~ se lee «según»: *minutos de traslado **según** el área*. En el teclado latinoamericano está a la derecha de la P, o con `AltGr` + 4.

6. Guardar una gráfica

```r
png("mi_grafico.png", width = 1000, height = 700)
barplot(table(datos$area), horiz = TRUE, col = "steelblue")
dev.off()                          # cierra el archivo: sin esto queda vacío
getwd()                            # dónde quedó el archivo
```

Vocabulario de RStudio (la interfaz está en inglés)

File / Upload = Archivo / Subir · *Console* = donde corre el código · *Environment* = los datos cargados · *Import Dataset* = traer un archivo de datos · *Plots » Export* = guardar la imagen · *Run* (**Ctrl + Enter**) = ejecutar la línea donde está el cursor

Cómo se cuenta un hallazgo

Una cifra · una gráfica · una frase que proponga algo — y siempre al lado, **cuántas respuestas** hay detrás.

> «De 27 docentes, los de agropecuaria demoran el doble en llegar. *(gráfica)* Propongo revisar quién tiene clase a primera hora.»

La cifra dice cuánto, la gráfica lo hace ver, y la frase dice qué hacer. Sin esa última, un hallazgo se queda en una gráfica bonita.

Dos cautelas, y valen siempre:

- Son las respuestas de **quienes contestaron**, no de todos: quien no contestó puede no parecerse a quien sí
- Con pocos datos, una diferencia pequeña puede ser **casualidad**. Fíjense si es **grande** y si **tiene sentido**

Si se atascan

Primero, sin internet — R trae su ayuda dentro:

```r
?barplot           # la ayuda de una función
help(read.csv)     # lo mismo, escrito de otra forma
example(boxplot)   # ejemplos que se ejecutan solos
```

Cuando algo falle, **copiar el mensaje de error tal cual** y buscarlo: casi siempre alguien ya lo preguntó.

Dónde preguntar — Kobo: `support.kobotoolbox.org` (manual) y `community.kobotoolbox.org` (foro) · R: `stackoverflow.com` y `es.stackoverflow.com` · Hojas de referencia: `posit.co/resources/cheatsheets`

Preguntarle a una IA (ChatGPT, Claude, Gemini — gratuitas y responden en español). Lo que convierte una respuesta inútil en una útil:

1. Decir el contexto: «**en R base, sin instalar paquetes**, soy principiante». Sin esa frase contestan con `ggplot2` y `install.packages(...)`, que en la computadora del laboratorio no se puede instalar
2. Pegar el **mensaje de error completo**, no un resumen
3. Pegar la salida de `str(datos)`, para que sepa cómo son sus datos
4. Pedir que **explique**, no sólo que arregle

Dos advertencias:

- **La IA se equivoca con seguridad** e inventa funciones que no existen. No pasa nada: se ejecuta y R avisa. **Nunca copiar sin correr.**
- **No pegar datos personales de estudiantes** — nombres, DNI, notas. Lo que se pega en un servicio ajeno sale del instituto. Para pedir ayuda basta con `str(datos)` o dos filas inventadas.