

Informe de R en Markdown R

Pepito Perez

2024-03-06

Introducción

El estilo de Tufte handout es un estilo que Edward Tufte utiliza en sus libros y folletos. El estilo de Tufte se caracteriza por el uso extensivo de notas al margen, la integración ajustada de gráficos con texto y una tipografía bien establecida. Este estilo ha sido implementado en LaTeX y HTML/CSS¹, respectivamente. Hemos portado ambas implementaciones al **tufte** package. Si deseas una salida en LaTeX/PDF, puedes utilizar el formato **tufte_handout** para folletos y **tufte_book** para libros. Para la salida en HTML, utiliza **tufte_html**. Estos formatos se pueden especificar en el encabezado YAML al comienzo de un documento R Markdown (ver el ejemplo a continuación) o pasarlos a la función `rmarkdown::render()`. Consulta Allaire et al. [2023] para obtener más información sobre **rmarkdown**.

¹ Mire el repositorio `tufte-latex` and `tufte-css`

```
---
title: "An Example Using the Tufte Style"
author: "John Smith"
output:
  tufte::tufte_handout: default
  tufte::tufte_html: default
---
```

Este paquete tiene dos objetivos:

1. Producir tanto salida en PDF como en HTML con estilos similares desde el mismo documento R Markdown.
2. Proporcionar una sintaxis simple para escribir elementos del estilo Tufte, como notas al margen y figuras en el margen. Por ejemplo, si deseas una figura en el margen, todo lo que necesitas hacer es usar la opción del chunk `fig.margin = TRUE`, y nosotros nos encargaremos de los detalles para que no tengas que preocuparte por `\begin{marginfigure} \end{marginfigure}` o ` `; el código LaTeX y HTML subyacente puede ser complicado, pero nunca necesitas aprender o escribir dicho código.

Si tienes alguna solicitud de función o encuentras errores en **tufte**, no dudes en informarlos en <https://github.com/rstudio/tufte/issues>. Para preguntas generales, puedes preguntar en StackOverflow: <https://stackoverflow.com/tags/rmarkdown>.

Encabezados

Este estilo proporciona encabezados de primer y segundo nivel (es decir, # y ##), como se demuestra en la siguiente sección. Puedes obtener una salida inesperada si intentas usar ### y encabezados más pequeños.

EN SUS LIBROS MÁS RECIENTES², Tufte comienza cada sección con un poco de espacio vertical, un párrafo sin sangría y establece las primeras palabras de la oración en versalitas. Para lograr esto con este estilo, llama a la función `newthought()` en `tufte` en una *expresión R en línea* ``r`` como se muestra al principio de este párrafo.³

Figuras

Figuras al margen

Las imágenes y gráficos juegan un papel integral en el trabajo de Tufte. Para colocar figuras en el margen, puedes usar la opción de chunk `knitr fig.margin = TRUE`. Por ejemplo:

```
library(ggplot2)
mtcars2 <- mtcars
mtcars2$am <- factor(
  mtcars$am, labels = c('automatic', 'manual')
)
ggplot(mtcars2, aes(hp, mpg, color = am)) +
  geom_point() + geom_smooth() +
  theme(legend.position = 'bottom')
```

Observa el uso de la opción `fig.cap` para proporcionar un pie de figura. Puedes ajustar las proporciones de las figuras utilizando las opciones del chunk `fig.width` y `fig.height`. Estas se especifican en pulgadas y se escalarán automáticamente para ajustarse al margen del folleto.

Contenido arbitrio al margen

De hecho, puedes incluir cualquier cosa en el margen utilizando el motor `knitr` llamado `marginfigure`. A diferencia de los bloques de código R ```{r}``, puedes escribir un chunk empezando con ``{marginfigure} y luego colocas el contenido en el bloque. Mira un ejemplo sobre el primer teorema fundamental del cálculo a la derecha.`

Por razones de portabilidad entre LaTeX y HTML, debes mantener el contenido del margen lo más simple posible (syntax-wise) en los

² Beautiful Evidence

³ Ten en cuenta que no debes asumir que `tufte` ha sido instalado a tu sesión de R. Debes llamar `library(tufte)` en tu documento R Markdown antes de llamar a `newthought()`, o utilizar `tufte::newthought()`.

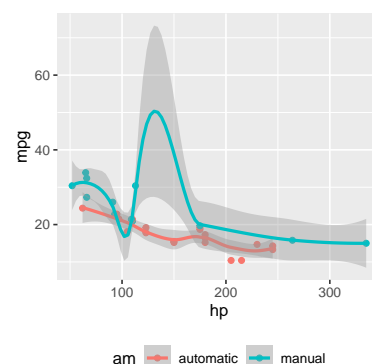


Figure 1: MPG vs horsepower, por tipo de transmisión.

Sabemos del *primer teorema fundamental del cálculo* que para x en $[a, b]$:

$$\frac{d}{dx} \left(\int_a^x f(u) du \right) = f(x).$$

bloques `marginfigure`. Puedes usar una sintaxis de Markdown simple como texto en *****negrita***** o *`_cursiva_`*, pero evita el uso de notas al pie, citas o elementos de nivel de bloque (por ejemplo, citas textuales y listas) allí.

Nota: si estableces `echo = FALSE` en tus opciones globales de chunk, deberás agregar `echo = TRUE` al chunk para mostrar una figura en el margen, por ejemplo ````{marginfigure, echo = TRUE}`.

Figuras de ancho completo

Puedes hacer que las figuras abarquen toda la página utilizando la opción del chunk `fig.fullwidth = TRUE`.

```
ggplot(diamonds, aes(carat, price)) + geom_smooth() +
  facet_grid(~ cut)
```

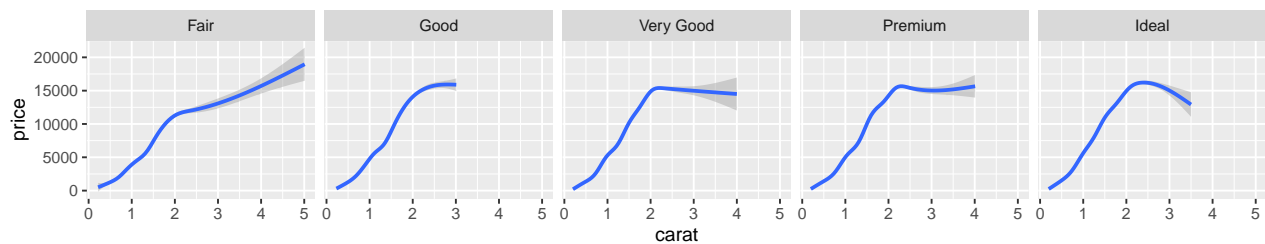


Figure 2: A full width figure.

Todavía se pueden usar otras opciones del chunk relacionadas con las figuras, como `fig.width`, `fig.cap`, `out.width`, y así sucesivamente. Para figuras de ancho completo, generalmente `fig.width` es grande y `fig.height` es pequeño. En el ejemplo anterior, el tamaño del gráfico es 10×2 .

Contenido arbitrario de ancho completo

Cualquier contenido puede abarcar todo el ancho de la página. Esta característica requiere Pandoc 2.0 o superior. Todo lo que necesitas hacer es colocar tu contenido en un Div cercado con la clase `fullwidth`, por ejemplo,

```
::: {.fullwidth}
Any _full width_ content here.
:::
```

A continuación un ejemplo:

R es software libre y se ofrece ABSOLUTAMENTE SIN GARANTÍA. Puedes redistribuirlo según los términos de las versiones 2 o 3 de la Licencia Pública General de GNU. Para obtener más información sobre estos asuntos, consulta <https://www.gnu.org/licenses/>.

Figuras en la columna principal

Además de las figuras en el margen y de ancho completo, también puedes incluir figuras en la columna principal. Este es el tipo predeterminado de figuras en la salida LaTeX/HTML

```
ggplot(diamonds, aes(cut, price)) + geom_boxplot()
```

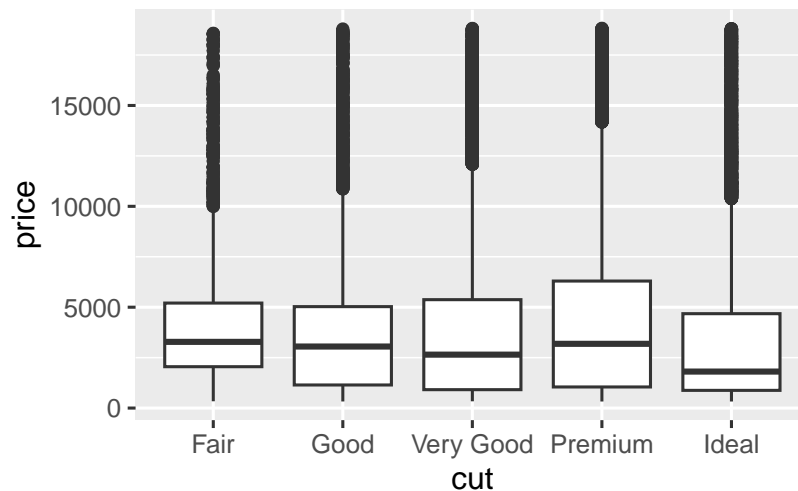


Figure 3: Una figura en la columna principal.

Notas al margen

Una de las características más prominentes y distintivas de este estilo es el uso extensivo de notas al margen. Hay un margen amplio para proporcionar un amplio espacio para notas al margen y pequeñas figuras. Cualquier uso de una nota al pie se convertirá automáticamente en una nota al margen.⁴

Si deseas colocar información adicional en el margen sin la marca de la nota al margen (el número sobrescrito), puedes usar la función `margin_note()` de **tufte** en una expresión R en línea. Esta función no procesa el texto con Pandoc, por lo que la sintaxis de Markdown no funcionará aquí. Si necesitas escribir algo en la sintaxis de Markdown, utiliza el bloque `marginfigure` descrito anteriormente.

⁴ Esta es una nota al margen que fue puesta al final del párrafo.

Esta es una nota al margen. Observa que no hay número antes de la nota.

Referencias

Las referencias se pueden mostrar como notas al margen para la salida HTML. Por ejemplo, podemos citar a R aquí [R Core Team, 2023]. Para habilitar esta función, debes establecer `link-citations: yes`

en el encabezado YAML y la versión de Pandoc debe ser al menos 2.11 o `pandoc-citeproc` debería estar disponible y ser al menos 0.7.2. Siempre puedes instalar tu propia versión de Pandoc desde <https://pandoc.org/installing.html> si la versión no es suficiente. Para verificar la versión de `pandoc-citeproc` en tu sistema, puedes ejecutar esto en R:

```
system2('pandoc-citeproc', '--version')
```

Si tu versión de `pandoc-citeproc` es demasiado baja, o no establece `link-citations: yes` en YAML, las referencias en la salida HTML se colocarán al final del documento de salida.

Tablas

Puedes utilizar la función `kable()` del paquete **knitr** para formatear tablas que se integran bien con el resto del estilo de Tufte handout. Los títulos de las tablas se colocan en el margen al igual que las figuras en la salida HTML.

```
knitr::kable(
  mtcars[1:6, 1:6], caption = 'A subconjunto de mtcars.'
)
```

Table 1: A subconjunto de mtcars.

	mpg	cyl	disp	hp	drat	wt
Mazda RX4	21.0	6	160	110	3.90	2.620
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875
Datsun 710	22.8	4	108	93	3.85	2.320
Hornet 4 Drive	21.4	6	258	110	3.08	3.215
Hornet Sportabout	18.7	8	360	175	3.15	3.440
Valiant	18.1	6	225	105	2.76	3.460

Citas en bloque

Sabemos por la sintaxis de Markdown que los párrafos que comienzan con `>` se convierten en citas en bloque. Si deseas agregar un pie de página alineado a la derecha para la cita, puedes usar la función `quote_footer()` de **tufte** en una expresión R en línea. Aquí tienes un ejemplo:

“Si no fuera por mi abogado, todavía estaría en prisión. Fue mucho más rápido con dos personas cavando.”

— Joe Martin sin usar `quote_footer()`, se vería así (la segunda línea es solo un párrafo normal):

“Las personas excelentes hablan de ideas, las personas promedio hablan de cosas y las personas pequeñas hablan de vino.”

— Fran Lebowitz # Responsiveness

La página HTML es receptiva en el sentido de que cuando el ancho de la página es menor que 760 px, las notas al margen y las notas al margen se ocultarán por defecto. Para las notas al margen, puedes hacer clic en sus números (los superíndices) para alternar su visibilidad. Para las notas al margen, puedes hacer clic en los signos de suma circulares para alternar su visibilidad.

Mas ejemplos

El resto de este documento consiste en algunos casos de prueba para asegurarnos de que todo siga funcionando bien en escenarios ligeramente más complicados. Primero generamos dos gráficos en un entorno de figura con la opción del chunk `fig.show = 'hold'`:

```
p <- ggplot(mtcars2, aes(hp, mpg, color = am)) +
  geom_point()
p
p + geom_smooth()
```

Luego, dos gráficos en entornos de figura separados (el código es idéntico al código anterior, pero la opción del chunk es la predeterminada `fig.show = 'asis'` ahora):

```
p <- ggplot(mtcars2, aes(hp, mpg, color = am)) +
  geom_point()
p

p + geom_smooth()
```

Puedes haber notado que las dos figuras tienen subtítulos diferentes, y eso es porque usamos un vector de caracteres de longitud 2 para la opción del chunk `fig.cap` (algo así como `fig.cap = c('primer gráfico', 'segundo gráfico')`).

A continuación, mostramos varios gráficos en figuras de margen. De manera similar, dos gráficos en el mismo entorno de figura en el margen:

```
p
p + geom_smooth(method = 'lm')

## `geom_smooth()` using formula = 'y ~ x'
```

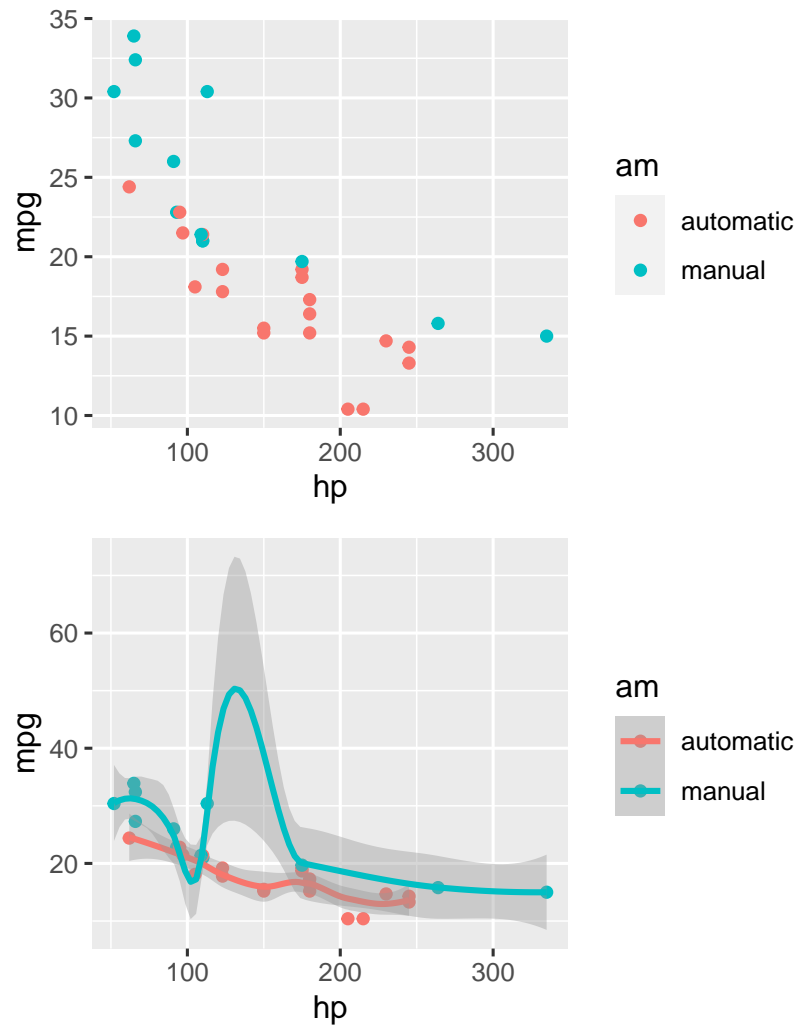


Figure 4: Dos figuras en la columna principal.

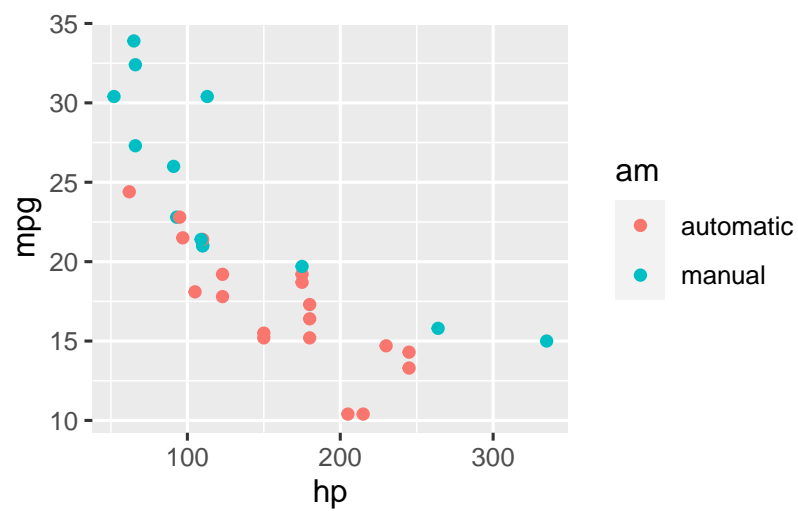


Figure 5: Dos graficos en entornos separados

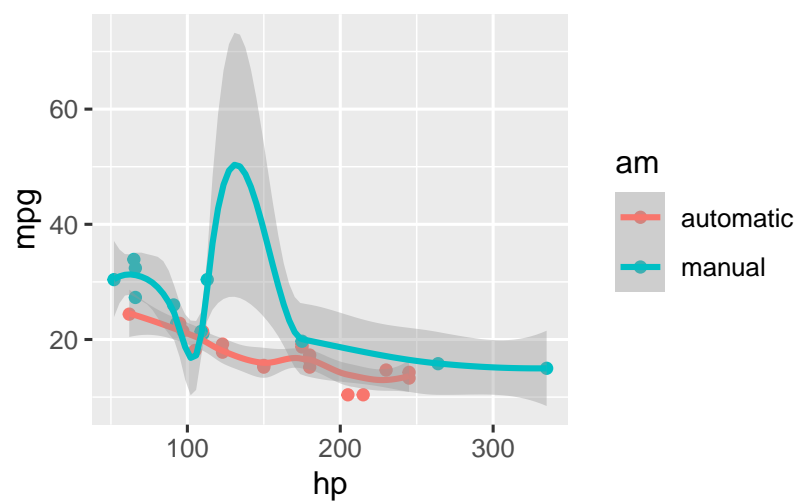


Figure 6: Dos graficos en entornos separados

Luego, dos gráficos del mismo bloque de código colocados en entornos de figura diferentes:

```
knitr::kable(head(iris, 15))
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa

p

```
knitr::kable(head(iris, 12))
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa

```
p + geom_smooth(method = 'lm')
```

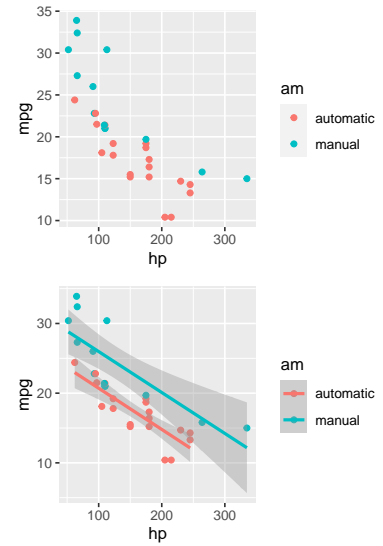


Figure 7: Dos gráficos en el mismo lugar al margen.

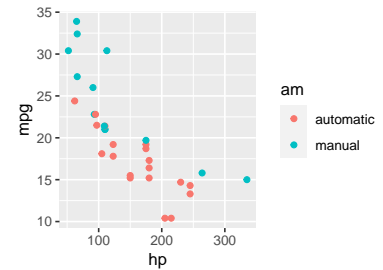


Figure 8: Dos gráficos separados al margen (El primer grafico).

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
knitr::kable(head(iris, 5))
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa

Mezclamos algunas tablas en el código anterior solo como *marcadores de posición* para asegurarnos de que haya suficiente espacio vertical entre las figuras en el margen, de lo contrario, se apilarán estrechamente juntas. Para un documento práctico, no deberías insertar demasiadas figuras en el margen consecutivamente y hacer que el margen esté abarrotado.

No es necesario asignar títulos a las figuras. Mostramos tres figuras sin títulos a continuación en el margen, en la columna principal y de ancho completo, respectivamente.

```
# un boxplot de weight vs transmission; la figura
# aparecerá en el margen
ggplot(mtcars2, aes(am, wt)) + geom_boxplot() +
  coord_flip()
```

```
# Una figura e la cloumna principal
p <- ggplot(mtcars, aes(wt, hp)) + geom_point()
p
```

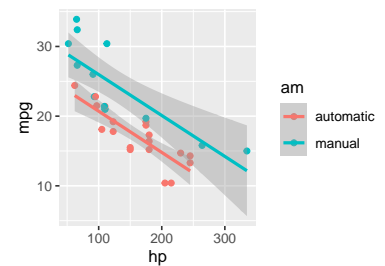
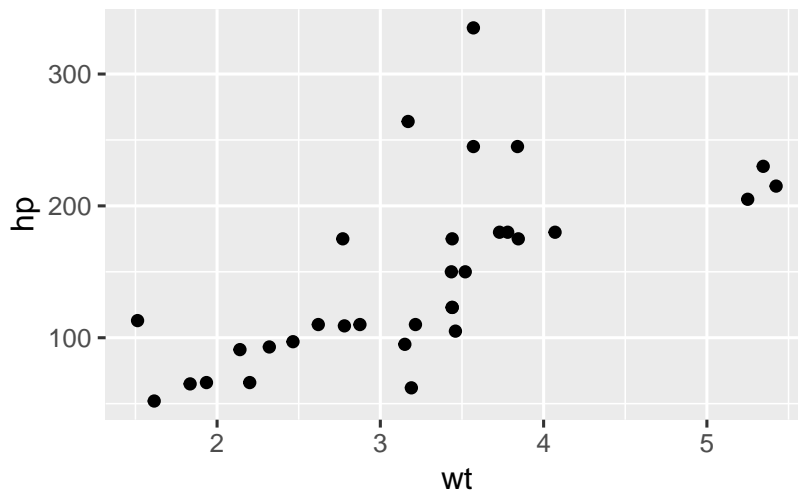
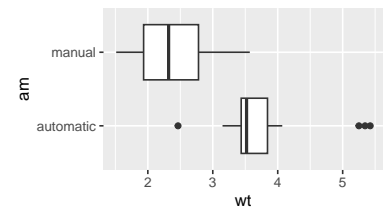
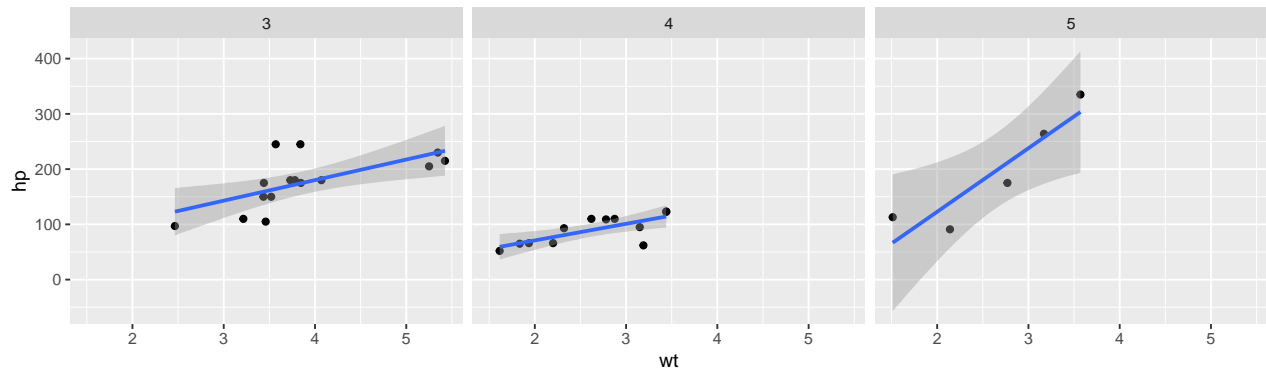


Figure 9: Dos gráficos separados al margen (El segundo grafico).



```
# Una figura con full tamaño
p + geom_smooth(method = 'lm') + facet_grid(~ gear)

## `geom_smooth()` using formula = 'y ~ x'
```



Algunas Notas sobre Tufte CSS

Hay algunas otras cosas en Tufte CSS que no hemos mencionado hasta ahora. Si prefieres las fuentes **sans-serif**, usa la función `sans_serif()` en **tufte**. Para epígrafes, puedes usar un par de guiones bajos para hacer que el párrafo aparezca en cursiva en una cita en bloque, por ejemplo.

`r tufte::quote_footer('— Dave Barry')` Esperamos que disfrutes de la simplicidad de R Markdown y este paquete de R, y agradecemos sinceramente a los autores de los proyectos Tufte-CSS y Tufte-LaTeX por desarrollar la hermosa CSS y clases LaTeX. Nuestro paquete tufte no hubiera sido posible sin su arduo trabajo.

Puedo ganar un argumento sobre cualquier tema, contra cualquier oponente. La gente lo sabe y me evita en las fiestas. A menudo, como muestra de su gran respeto, ni siquiera me invitan.

— Dave Barry Esperamos que disfrutes de la simplicidad de R Markdown y este paquete de R, y agradecemos sinceramente a los autores de los proyectos Tufte-CSS y Tufte-LaTeX por desarrollar la hermosa CSS y clases LaTeX. Nuestro paquete tufte no hubiera sido posible sin su arduo trabajo.

Puedes activar/desactivar algunas características del estilo Tufte en la salida HTML. Las características predeterminadas habilitadas son:

```
output:
tufte::tufte_html:
  tufte_features: ["fonts", "background", "italics"]
```

Si no quieres que el fondo de la página sea de color amarillo claro, puedes eliminar `background` de `tufte_features`. También puedes personalizar el estilo de la página HTML mediante un archivo CSS. Por ejemplo, si no quieres que el subtítulo esté en cursiva, puedes definir

```
h3.subtitle em {
  font-style: normal;
}
```

en un archivo CSS, digamos `my_style.css` (en el mismo directorio que tu documento Rmd), y aplicarlo a tu salida HTML mediante la opción `css`, por ejemplo,

```
output:
  tufte::tufte_html:
    tufte_features: ["fonts", "background"]
    css: "my_style.css"
```

También hay una variante del estilo Tufte en HTML/CSS llamada “Envisioned CSS”. Este estilo se puede usar especificando el argumento `tufte_variant = 'envisioned'` en `tufte_html()` ⁵, por ejemplo.

Puedes ver la fuente en HTML y CSS de todos los estilos Tufte mencionados anteriormente en tu consola de JavaScript, o puedes encontrar los estilos correspondientes en el código fuente de este paquete en el directorio `inst/rmarkdown/templates/tufte_html/resources/`. El CSS es bastante simple, y puedes aprender mucho sobre el diseño web simplemente mirando y modificando estos archivos.

```
output:
  tufte::tufte_html:
    tufte_variant: "envisioned"
```

Para mirar el R markdown de este documento, usted puede seguir [this link to Github](#), use RStudio IDE (File -> New File -> R Markdown -> From Template), o abra un archivo `rmd`:

```
file.edit(
  tufte::template_resources(
    'tufte_html', '..', 'skeleton', 'skeleton.Rmd'
  )
)
```

Este documento también está disponible en Chinese, y el `envisioned` style puede ser encontrado en [here](#).

⁵ El actual CSS de Envisioned no fue usado en el paquete **tufte**. Cambiamos solo fuentes, color, color de texto basado en el estilo por defecto de Tufte.

References

- JJ Allaire, Yihui Xie, Christophe Dervieux, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. *rmarkdown: Dynamic Documents for R*, 2023. URL <https://CRAN.R-project.org/package=rmarkdown>. R package version 2.22.
- R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2023. URL <https://www.R-project.org/>.