

Mon rapport en Quarto

École nationale de la statistique et de l'analyse de l'information

Étudiant·e·s : Blanche Neige, Peter Pan, Mickey Mouse
 et Elsa d'Arendelle

Invalid Date

Introduction

This document demonstrates the use of a number of advanced page layout features to produce an attractive and usable document inspired by the Tufte handout style and the use of Tufte’s styles in RMarkdown documents [xie2018]. The Tufte handout style is a style that Edward Tufte uses in his books and handouts. Tufte’s style is known for its extensive use of sidenotes, tight integration of graphics with text, and well-set typography. Quarto¹ supports most of the layout techniques that are used in the Tufte handout style for both HTML and LaTeX/PDF output.

```
---
title: "An Example Using the Tufte Style"
author: "John Smith"
format:
  html:
    grid:
      margin-width: 350px
  pdf: default
reference-location: margin
citation-location: margin
---
```

①

②

- ① Increases the width of the margin to make more room for sidenotes and margin figures (HTML only).
- ② Places footnotes and cited sources in the margin. Other layout options (for example placing a figure in the margin) will be set per element in examples below.

These layout features are designed with two important goals in mind:

1. To produce both PDF and HTML output with similar styles from the same Quarto document;
2. To provide simple syntax to write elements of the Tufte style such as side notes and margin figures. If you’d like a figure placed in the margin, just set the option `fig-column: margin` for your code chunk, and we will take care of the details for you².

If you have any feature requests or find bugs in these capabilities, please do not hesitate to file them to <https://github.com/quarto-dev/quarto-cli/issues>.

¹To learn more, you can read more about [Quarto](#) or visit [Quarto’s Github repository](#).

²You never need to think about `\begin{marginfigure}` or ``; the LaTeX and HTML code under the hood may be complicated, but you never need to learn or write such code.

Figures

Margin Figures

Images and graphics play an integral role in Tufte's work. To place figures in the margin you can use the **Quarto** chunk option `column: margin`. For example:

```
```{r}
#| label: fig-margin
#| fig-cap: "MPG vs horsepower, colored by transmission."
#| column: margin
#| message: false
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')
```
```

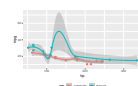


Figure 1:
MPG vs
horsepower,
colored by
transmission.

Note the use of the `fig-cap` chunk option to provide a figure caption. You can adjust the proportions of figures using the `fig-width` and `fig-height` chunk options. These are specified in inches, and will be automatically scaled down to fit within the handout margin.

Arbitrary Margin Content

You can include anything in the margin by placing the class `.column-margin` on the element. See an example on the right about the first fundamental theorem of calculus.

Full Width Figures

You can arrange for figures to span across the entire page by using the chunk option `fig-column: page-right`.

```
```{r}
#| label: fig-fullwidth
#| fig-cap: "A full width figure."
#| fig-width: 11
#| fig-height: 3
#| fig-column: page-right
#| warning: false
ggplot(diamonds, aes(carat, price)) + geom_smooth() +
 facet_grid(~ cut)
```
```

We know
from *the first
fundamental
theorem of
calculus* that
for x in
 $[a, b]$:

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

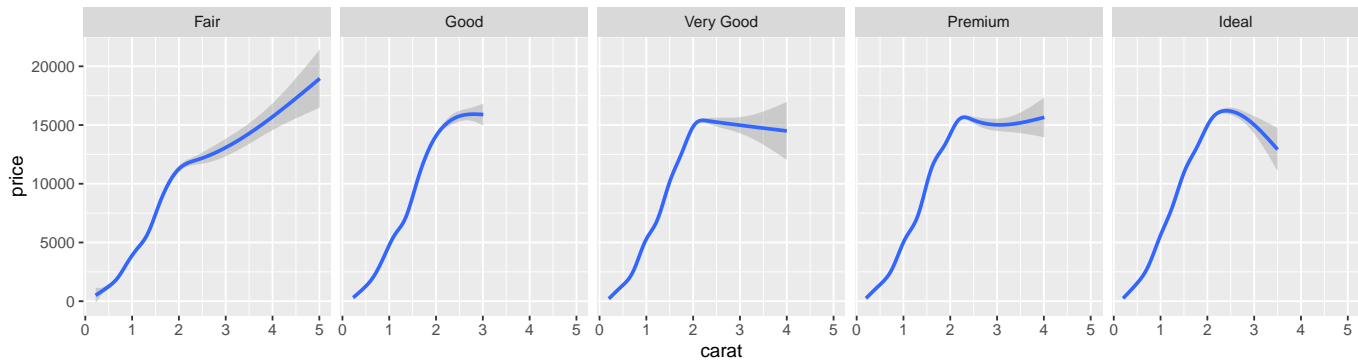


Figure 2: A full width figure.

Other chunk options related to figures can still be used, such as `fig-width`, `fig-cap`, and so on. For full width figures, usually `fig-width` is large and `fig-height` is small. In the above example, the plot size is 11×3 .

Arbitrary Full Width Content

Any content can span to the full width of the page, simply place the element in a `div` and add the class `column-page-right`. For example, the following code will display its contents as full width.

```

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

```

Below is an example:

R is free software and comes with ABSOLUTELY NO WARRANTY. You are welcome to redistribute it under the terms of the GNU General Public License versions 2 or 3. For more information about these matters see <https://www.gnu.org/licenses/>.

Main Column Figures

Besides margin and full width figures, you can of course also include figures constrained to the main column. This is the default type of figures in the LaTeX/HTML output.

```

```{r}
#| label: fig-main
#| fig-cap: "A figure in the main column."
ggplot(diamonds, aes(cut, price)) + geom_boxplot()
```

```

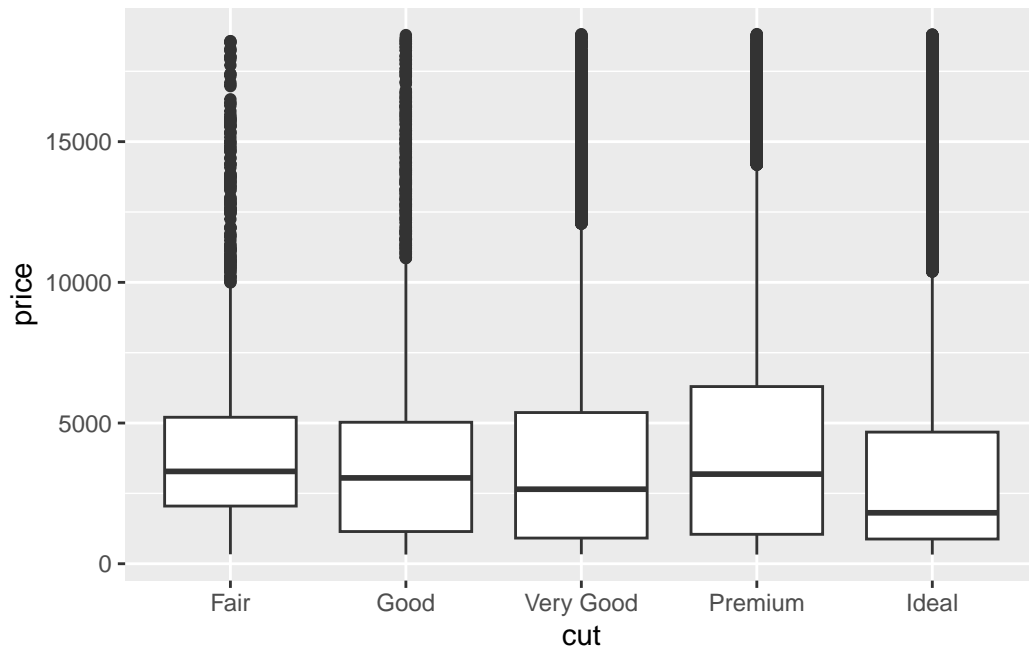


Figure 3: A figure in the main column.

Margin Captions

When you include a figure constrained to the main column, you can choose to place the figure's caption in the margin by using the `cap-location` chunk option. For example:

```
```{r}
#| label: fig-main-margin-cap
#| fig-cap: "A figure with a longer caption. The figure appears in the main column, but the
#| cap-location: margin
ggplot(diamonds, aes(cut, price)) + geom_boxplot()
```
```

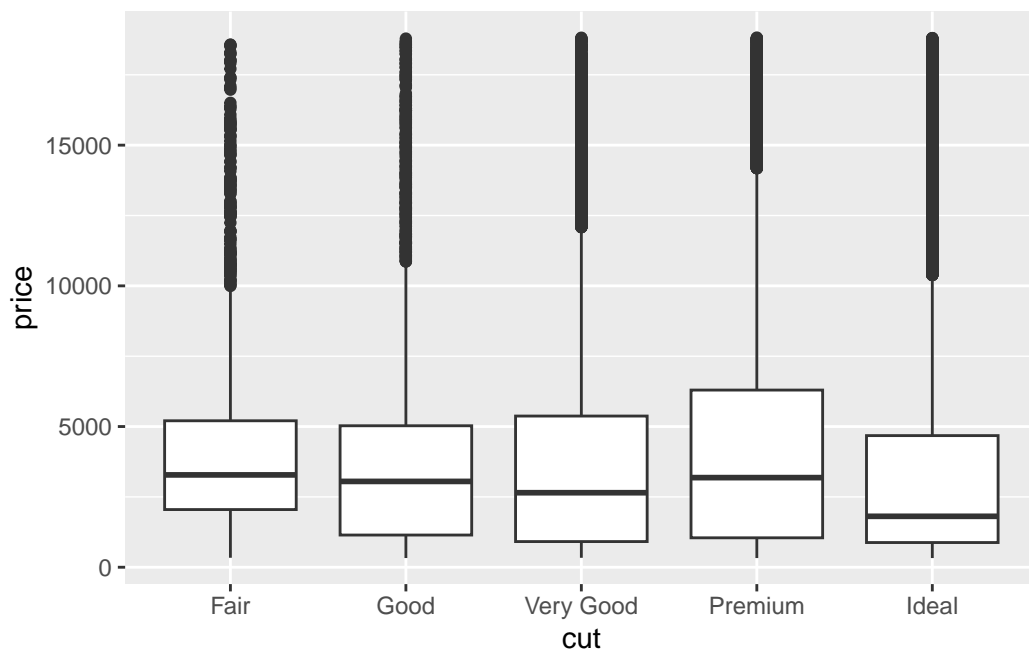


Figure 4: A figure with a longer caption. The figure appears in the main column, but the caption is placed in the margin. Captions can even contain elements like a citation such as @xie2018.

Sidenotes

One of the most prominent and distinctive features of this style is the extensive use of sidenotes. There is a wide margin to provide ample room for sidenotes and small figures. Any use of a footnote will automatically be converted to a sidenote.

If you'd like to place ancillary information in the margin without the sidenote mark (the superscript number), you can use apply the `column-margin` class to the element.

This is a span
that has the
class
`column-margin`
which places
it in the
margin
without the
sidenote
mark.

References

References can be displayed as margin notes for HTML output. For example, we can cite R here [[@R-base](#)].

i This feature depends upon `link-citations` to locate and place references in the margin. This is enabled by default, but if you disable `link-citations` then references in the HTML output will be placed at the end of the output document as they normally are.

Tables

You can use the `kable()` function from the **knitr** package to format tables that integrate well with the rest of the Tufte handout style. The table captions are placed in the margin like figures in the HTML output.

```
```{r}
#| tbl-cap-location: margin
knitr::kable(
 mtcars[1:6, 1:6], caption = 'A subset of 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 |

Table 1: A subset of mtcars.

Responsiveness

The HTML page layout is responsive- as the page width shrinks, elements will automatically adjust their position. Elements that appear in the margins will move inline with the content and elements that span the body and margin will automatically span only the body.

More Examples

The rest of this document consists of a few test cases to make sure everything still works well in slightly more complicated scenarios. First we generate two plots in one figure environment with the chunk option `fig-show: hold`:

```
```{r}
#| label: fig-two-together
#| fig-cap: "Two plots in one figure environment."
#| fig-show: hold
#| warning: false
#| cap-location: margin
p <- ggplot(mtcars2, aes(hp, mpg, color = am)) +
 geom_point()
p
p + geom_smooth()
```
```

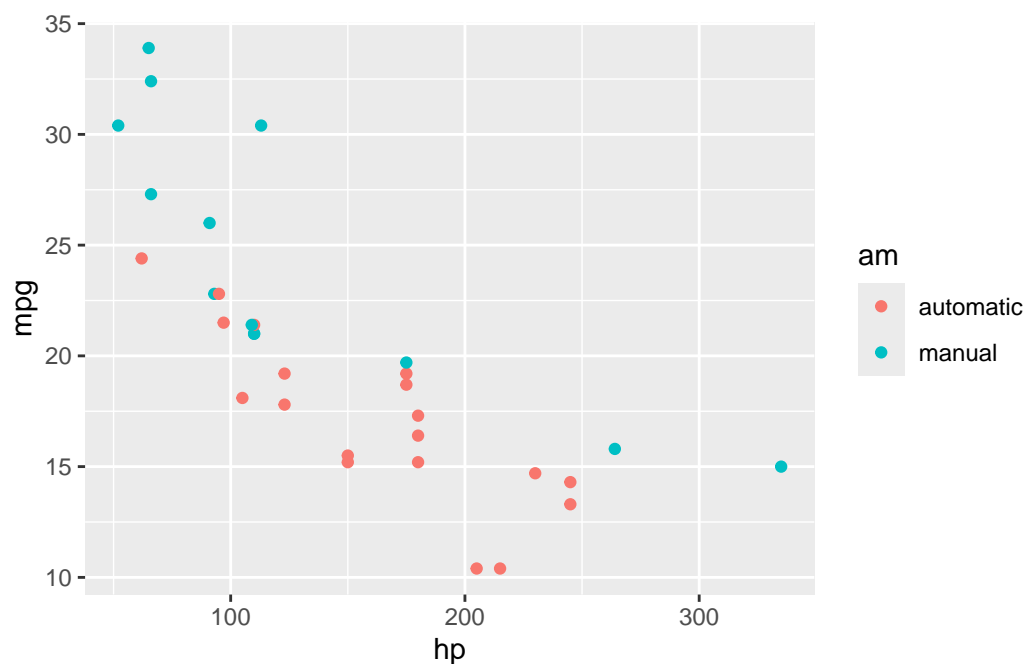


Figure 5: Two plots in one figure environment.

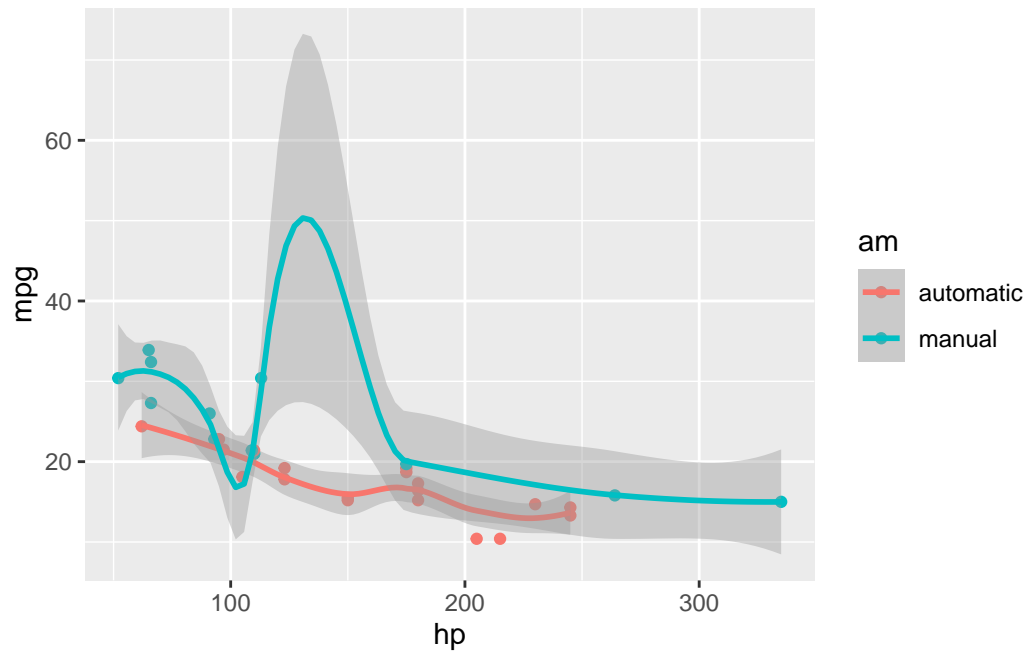


Figure 6: Two plots in one figure environment.

Then two plots in separate figure environments (the code is identical to the previous code chunk, but the chunk option is the default `fig-show: asis` now):

```
```{r fig-two-separate, ref.label='fig-two-together', fig.cap=sprintf("Two plots in separate
#| cap-location: margin
p <- ggplot(mtcars2, aes(hp, mpg, color = am)) +
 geom_point()
p
p + geom_smooth()
```
```

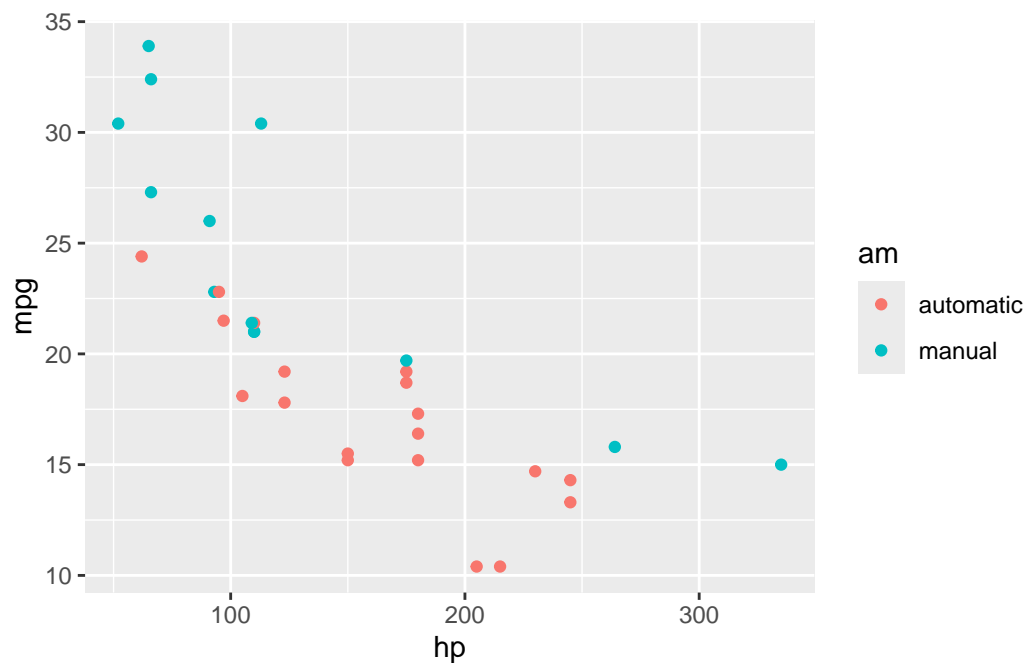


Figure 7: Two plots in separate figure environments (the first plot).

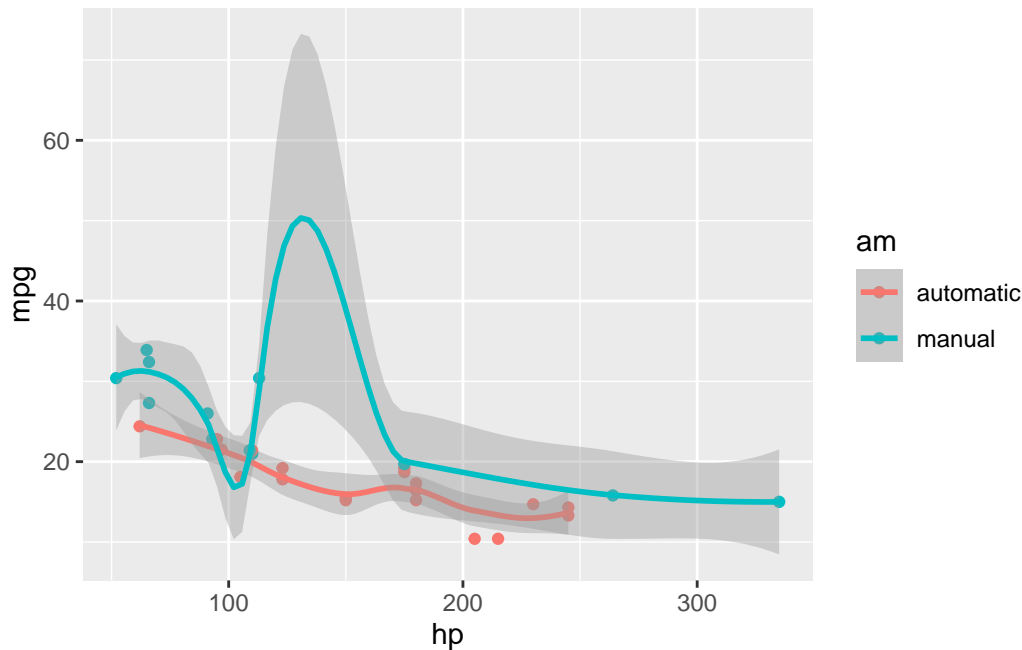


Figure 8: Two plots in separate figure environments (the second plot).

You may have noticed that the two figures have different captions, and that is because we used a character vector of length 2 for the chunk option `fig.cap` (something like `fig.cap = c('first plot', 'second plot')`).

💡 Using R within Chunk Options

If you wish to use raw R expressions as part of the chunk options (like above), then you need to define those in the `tag=value` format within the curly brackets `{r label, tag=value}` instead of the `tag: value` YAML syntax on a new line starting with the hashpipe `#|`. The former approach is documented on [knitr's website](#) while the latter is explained in [Quarto's documentation](#).

Next we show multiple plots in margin figures. Similarly, two plots in the same figure environment in the margin:

Then two plots from the same code chunk placed in different figure environments:

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

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width |
|--------------|-------------|--------------|-------------|
| 5.1 | 3.5 | 1.4 | 0.2 |
| 4.9 | 3.0 | 1.4 | 0.2 |

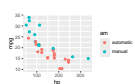


Figure 9: Two plots in one figure environment in the margin.

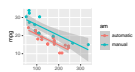


Figure 10: Two plots in one figure environment in the margin.



Figure 11: Two plots in separate figure environments in the margin

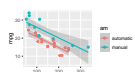


Figure 12: Two plots

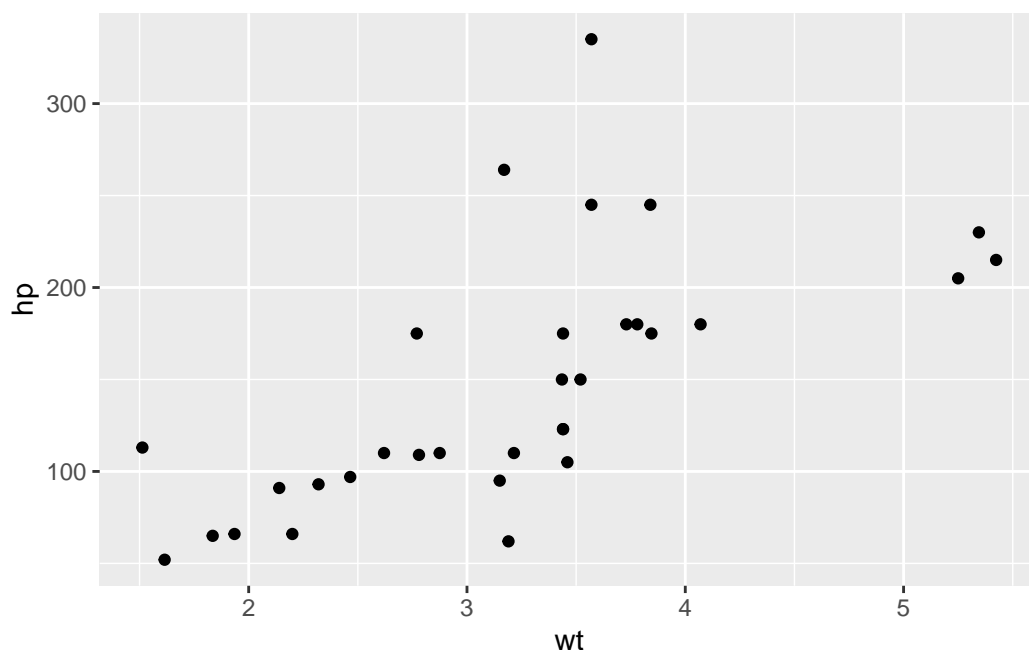
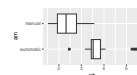
| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width |
|--------------|-------------|--------------|-------------|
| 4.7 | 3.2 | 1.3 | 0.2 |
| 4.6 | 3.1 | 1.5 | 0.2 |
| 5.0 | 3.6 | 1.4 | 0.2 |
| 5.4 | 3.9 | 1.7 | 0.4 |
| 4.6 | 3.4 | 1.4 | 0.3 |
| 5.0 | 3.4 | 1.5 | 0.2 |
| 4.4 | 2.9 | 1.4 | 0.2 |
| 4.9 | 3.1 | 1.5 | 0.1 |
| 5.4 | 3.7 | 1.5 | 0.2 |

We blended some tables in the above code chunk only as *placeholders* to make sure there is enough vertical space among the margin figures, otherwise they will be stacked tightly together. For a practical document, you should not insert too many margin figures consecutively and make the margin crowded.

You do not have to assign captions to figures. We show three figures with no captions below in the margin, in the main column, and in full width, respectively.

```
```{r}
#| fig-width: 3.5
#| fig-height: 2
#| column: margin
a boxplot of weight vs transmission; this figure
will be placed in the margin
ggplot(mtcars2, aes(am, wt)) + geom_boxplot() +
 coord_flip()
```
```

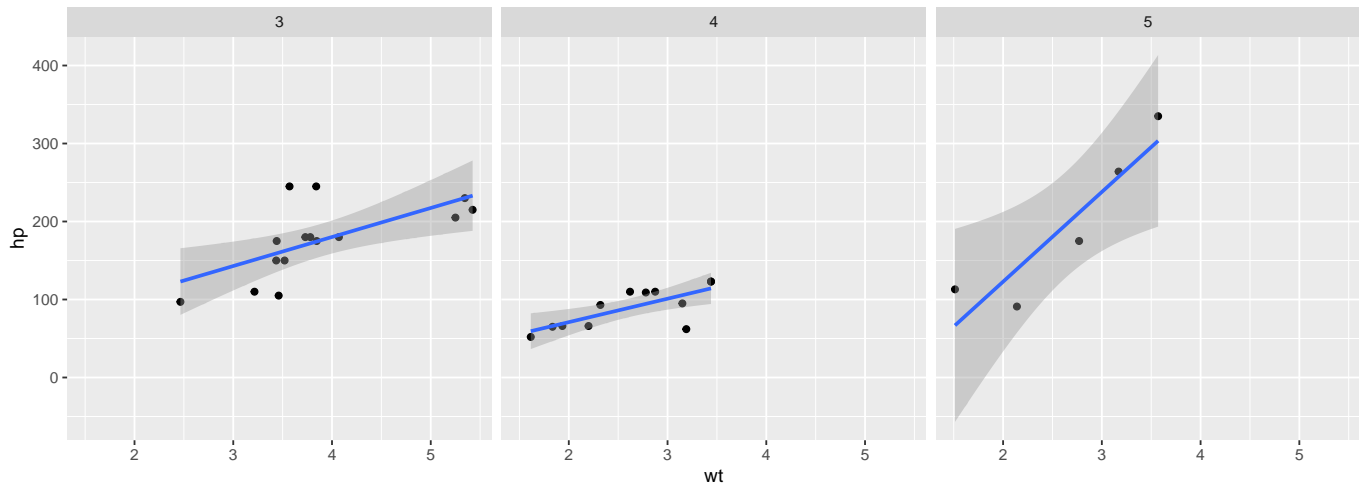
```
```{r}
#| warning: false
a figure in the main column
p <- ggplot(mtcars, aes(wt, hp)) + geom_point()
p
```
```



```

```{r}
#| fig-width: 11
#| fig-height: 4
#| column: page-right
#| warning: false
a fullwidth figure
p + geom_smooth(method = 'lm') + facet_grid(~ gear)
```

```



Some Notes on Page Layout

To see the Quarto markdown source of this example document, you may follow [this link to Github](#).