The Science and Art of Data

Hui Lin and Ming Li

2017-07-04

LEARNING OUTCOMES

After taking the CE course, participants will:

- Understand what data scientists "in the wild" are doing.
- Understand data science is the process of defining the problem with business knowledge, gathering needed data from various sources, developing models, extracting insight and recommend actions.
- Get familiar with the cloud-based big data platform (Hadoop/Hive/Spark/GPU etc.) that are widely used in the development and production setting for industry and know how to transit from academic environment to enterprise environment quickly.
- Get familiar with data extraction, transformation and load from various database systems such that participates can be self-sufficient to get needed data in the enterprise environment.
- Understand how to leverage interactive dashboard to present insight and results and communicate efficiently with business partners and customers
- Learn how to encode a real problem to a data science problem, search for the right data, preprocess data and deploy of analytical results through a case study
- Get familiar with how to achieve high-performance computing for standard statistical procedures with big data infrastructure

The art of data science

Data science and data scientist have become buzz words. Allow me to reiterate what you may have already heard a million times in the media: **data scientists are in demand and demand continues to grow**. A study by the McKinsey Global Institute concludes,

> "a shortage of the analytical and managerial talent necessary to make the most of Big Data is a significant and pressing challenge (for the U.S.)."

You may expect that statisticians and graduate students from traditional statistics departments are great data scientist candidates. But the situation is that the majority of current data scientists do not have a statistical background. As David Donoho pointed out:

> "statistics is being marginalized here; the implicit message is that statistics is a part of what goes on in data science but not a very big part." (from "50 years of Data Science").

What is wrong? The activities that preoccupied statistics over centuries are now in the limelight, but those activities are claimed to belong to a new discipline and are practiced by professionals from various backgrounds. Various professional statistics organizations are reacting to this confusing situation. (Page 5-7, "50 Years of Data Science") From those discussions, Donoho summarizes the main recurring "Memes" about data sciences:

1. The 'Big Data' Meme
2. The 'Skills' Meme
3. The 'Jobs' Meme

The first two are linked together which leads to statisticians' current position on data science. I assume everyone has heard the 3V (volume, variety and velocity) definition of big data. The media hasn't taken a minute break from touting "big" data. Data science trainees now need the skills to cope with such big data sets. What are those skills? You may hear about: Hadoop, system using Map/Reduce to process large data sets distributed across a cluster of computers. The new skills are for dealing with organizational artifacts of large-scale cluster computing but not for better solving the real problem. A lot of data on its own is worthless. It isn't the size of the data that's important. It's what you do with it. The big data skills that so many are touting today are not skills for better solving the real problem of inference from data.

Some media think they sense the trends in hiring and government funding. We are transiting to universal connectivity with a deluge of data filling telecom servers. But these facts don't immediately create a science. The statisticians have been laying the groundwork of data science for at least 50 years. Today's data science is an enlargement of traditional academic statistics rather than a brand new discipline.

Our goal is to help you enlarge your background to become a data scientist in US enterprise environments. We will use case studies to cover how to leverage big data distributed platforms (Hadoop / Hive / Spark), data wrangling, modeling, dynamic reporting (R markdown) and interactive dashboards (R-Shiny) to tackle real-world data science problems. One typical skill gap for statisticians is data ETL (extraction, transformation and load) in production environments, and we will cover this topic as well. Data science is a combination of science and art with data as the foundation. We will also cover the "art" part to guide participants to learn soft skills to define data science problems and to effectively communicate with business stakeholders. The

prerequisite knowledge is MS level education in statistics and entry level knowledge of R-Studio.

Introduction

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. This style has been implemented in LaTeX and HTML/CSS[1], respectively. We have ported both implementations into the **tufte** package. If you want LaTeX/PDF output, you may use the `tufte_handout` format for handouts, and `tufte_book` for books. For HTML output, use `tufte_html`. These formats can be either specified in the YAML metadata at the beginning of an R Markdown document (see an example below), or passed to the `rmarkdown::render()` function. See **?** more information about **rmarkdown**.

[1] See Github repositories tufte-latex and tufte-css

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

There are two goals of this package:

1. To produce both PDF and HTML output with similar styles from the same R Markdown document;
2. To provide simple syntax to write elements of the Tufte style such as side notes and margin figures, e.g. when you want a margin figure, all you need to do is the chunk option `fig.margin = TRUE`, and we will take care of the deails for you, so you never need to think about `\begin{marginfigure} \end{marginfigure}` or `<span class="marginfigure"> </span>`; the LaTeX and HTML code under the hood may be complicated, but you never need to learn or write such code.

If you have any feature requests or find bugs in **tufte**, please do not hesitate to file them to `https://github.com/rstudio/tufte/issues`. For general questions, you may ask them on StackOverflow: `http://stackoverflow.com/tags/rmarkdown`.

Headings

This style provides first and second-level headings (that is, `#` and `##`), demonstrated in the next section. You may get unexpected output if

you try to use `###` and smaller headings.

In his later books[2], Tufte starts each section with a bit of vertical space, a non-indented paragraph, and sets the first few words of the sentence in small caps. To accomplish this using this style, call the `newthought()` function in **tufte** in an *inline R expression* `` `r ` `` as demonstrated at the beginning of this paragraph.[3]

[2] Beautiful Evidence

[3] Note you should not assume **tufte** has been attached to your R session. You should either `library(tufte)` in your R Markdown document before you call `newthought()`, or use `tufte::newthought()`.

Figures

Margin Figures

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

```
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')
```

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

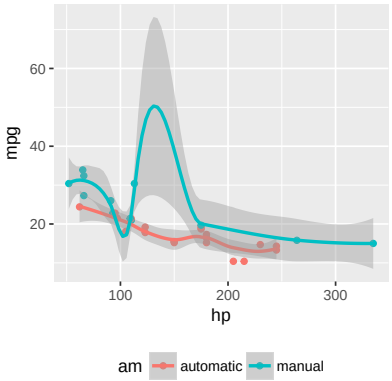


Figure 1: MPG vs horsepower, colored by transmission.

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

In fact, you can include anything in the margin using the **knitr** engine named `marginfigure`. Unlike R code chunks `` ```{r} ``, you write a chunk starting with `` ```{marginfigure} `` instead, then put the content in the chunk. See an example on the right about the first fundamental theorem of calculus.

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

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

For the sake of portability between LaTeX and HTML, you should keep the margin content as simple as possible (syntax-wise) in the `marginefigure` blocks. You may use simple Markdown syntax like `**bold**` and `_italic_` text, but please refrain from using footnotes, citations, or block-level elements (e.g. blockquotes and lists) there.

Full Width Figures

You can arrange for figures to span across the entire page by using the chunk option `fig.fullwidth = TRUE`.

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

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

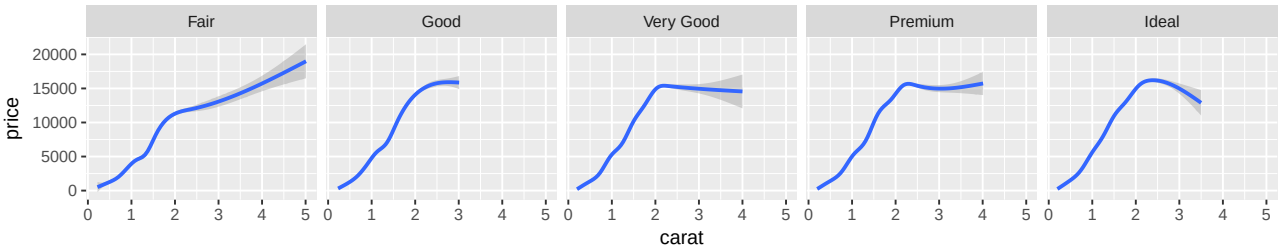


Figure 2: A full width figure.

Other chunk options related to figures can still be used, such as `fig.width`, `fig.cap`, `out.width`, 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 10×2.

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.

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

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. [4]

[4] This is a sidenote that was entered using a footnote.

If you'd like to place ancillary information in the margin without the sidenote mark (the superscript number), you can use the `margin_note()` function from **tufte** in an inline R expression. This function does not process the text with Pandoc, so Markdown syntax will not work here. If you need to write anything in Markdown syntax, please use the `marginfigure` block described previously.

This is a margin note. Notice that there is no number preceding the note.

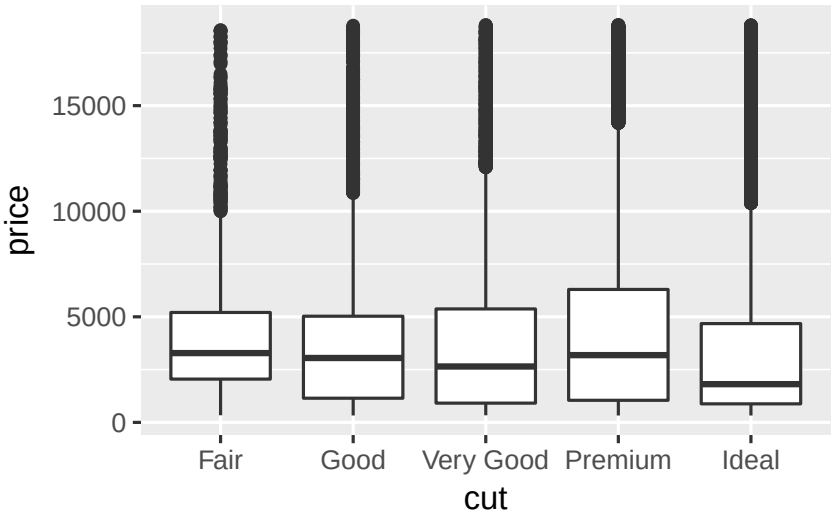


Figure 3: A figure in the main column.

References

References can be displayed as margin notes for HTML output. For example, we can cite R here [**?**]. To enable this feature, you must set `link-citations: yes` in the YAML metadata, and the version of `pandoc-citeproc` should be at least 0.7.2. You can always install your own version of Pandoc from `http://pandoc.org/installing.html` if the version is not sufficient. To check the version of `pandoc-citeproc` in your system, you may run this in R:

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

If your version of `pandoc-citeproc` is too low, or you did not set `link-citations: yes` in YAML, references in the HTML output will be placed at the end of the output document.

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.

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

Table 1: 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

Block Quotes

We know from the Markdown syntax that paragraphs that start with `>` are converted to block quotes. If you want to add a right-aligned footer for the quote, you may use the function `quote_footer()` from **tufte** in an inline R expression. Here is an example:

> "If it weren't for my lawyer, I'd still be in prison. It went a lot faster with two people digging."
>
> — Joe Martin

Without using `quote_footer()`, it looks like this (the second line is just a normal paragraph):

> "Great people talk about ideas, average people talk about things, and small people talk about wine."
>
> — Fran Lebowitz

Responsiveness

The HTML page is responsive in the sense that when the page width is smaller than 760px, sidenotes and margin notes will be hidden by default. For sidenotes, you can click their numbers (the superscripts) to toggle their visibility. For margin notes, you may click the circled plus signs to toggle visibility.

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'`:

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

```
p
p + geom_smooth()
```

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

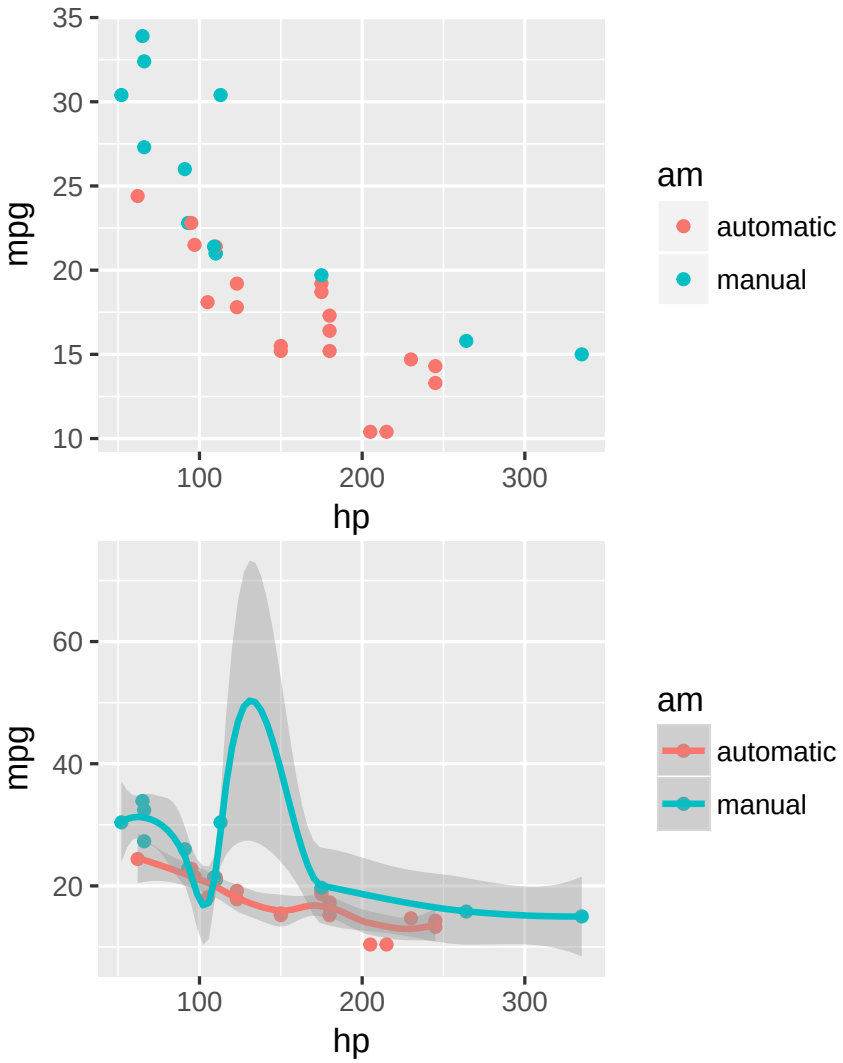


Figure 4: Two plots in one figure environment.

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):

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

p + geom_smooth()
```

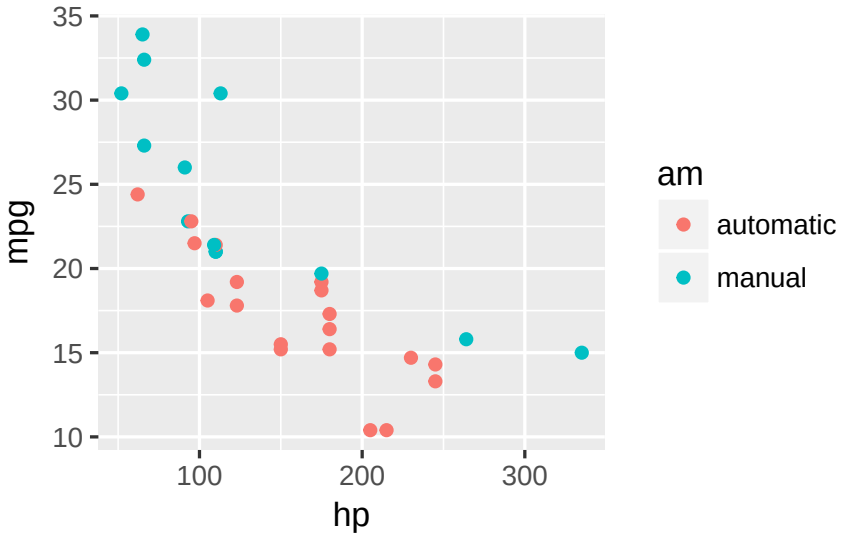


Figure 5: Two plots in separate figure environments (the first plot).

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

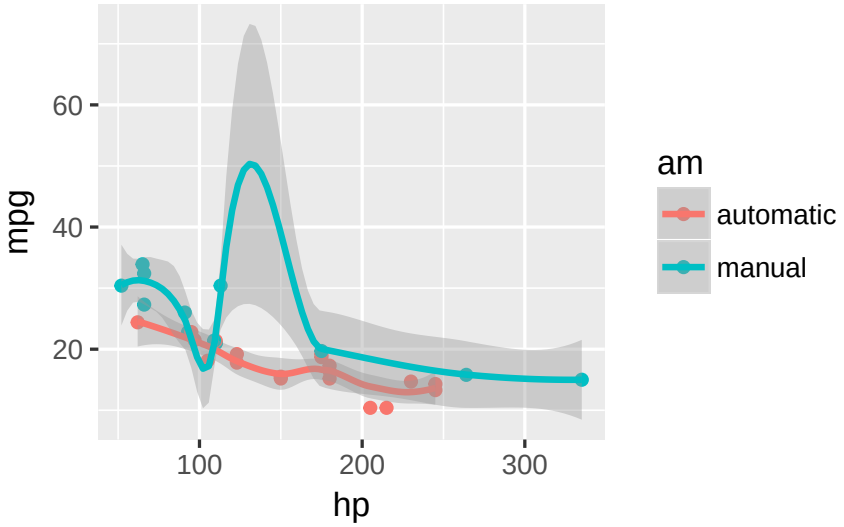


Figure 6: Two plots in separate figure environments (the second plot).

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')`).

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

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

THE

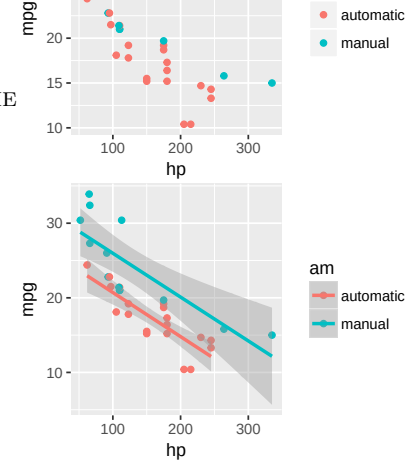


Figure 7: Two plots in one figure environment in the margin.

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

```
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
```

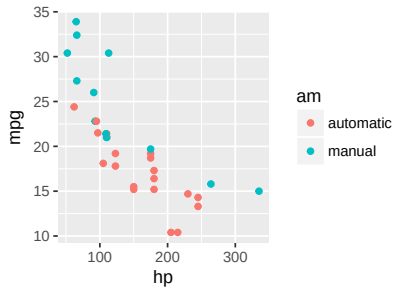


Figure 8: Two plots in separate figure environments in the margin (the first plot).

```
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')
```

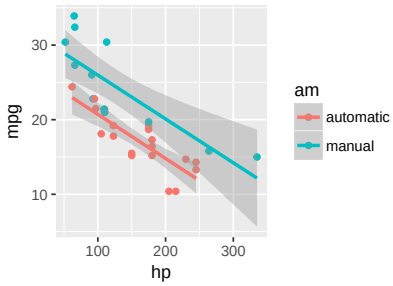


Figure 9: Two plots in separate figure environments in the margin (the second plot).

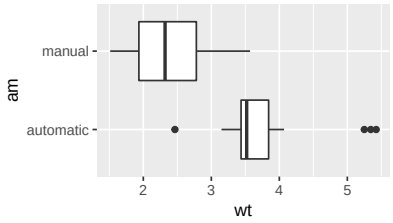
```
knitr::kable(head(iris, 5))
```

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

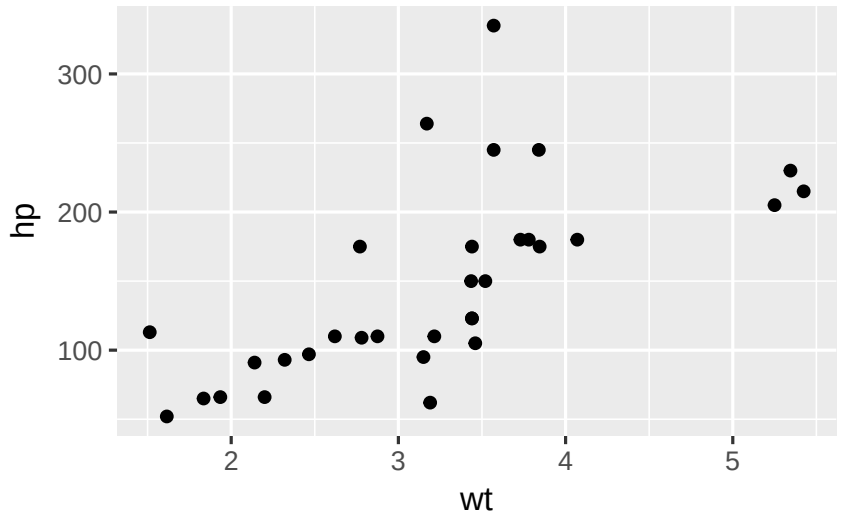
We blended some tables in the above code chunk only as *placeholders* to make sure there is enough vertical space among the margin figures, otherwise they will be stacked tightly together. For a practical document, you should not insert too many margin figures consecutively and make the margin crowded.

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.

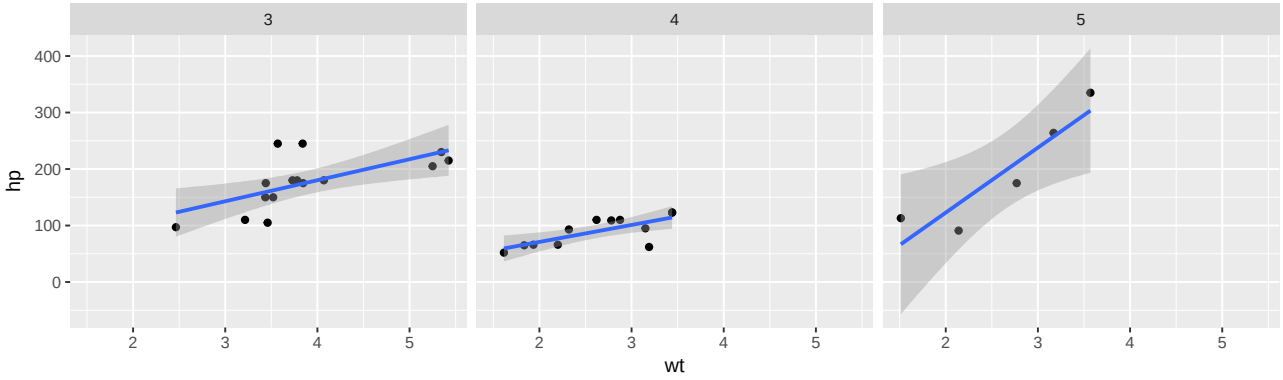
```
# a boxplot of weight vs transmission; this figure
# will be placed in the margin
ggplot(mtcars2, aes(am, wt)) + geom_boxplot() +
  coord_flip()
```




```
# a figure in the main column
p <- ggplot(mtcars, aes(wt, hp)) + geom_point()
p
```



```
# a fullwidth figure
p + geom_smooth(method = 'lm') + facet_grid(~ gear)
```



Some Notes on Tufte CSS

There are a few other things in Tufte CSS that we have not mentioned so far. If you prefer sans-serif fonts, use the function `sans_serif()` in **tufte**. For epigraphs, you may use a pair of underscores to make the paragraph italic in a block quote, e.g.

> *I can win an argument on any topic, against any opponent. People know this, and steer clear of me at parties. Often, as a sign of their great respect, they don't even invite me.*
>
> — Dave Barry

We hope you will enjoy the simplicity of R Markdown and this R package, and we sincerely thank the authors of the Tufte-CSS and Tufte-LaTeX projects for developing the beautiful CSS and LaTeX classes. Our **tufte** package would not have been possible without their heavy lifting.

To see the R Markdown source of this example document, you may follow this link to Github, use the wizard in RStudio IDE (`File -> New File -> R Markdown -> From Template`), or open the Rmd file in the package:

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

References