



MTXXXX: Some stats

L. Scott-Hayward and JJ Valletta

21 July 2020

Contents

Welcome	5
Prerequisites	5
Learning outcomes	5
Recommended reading	5
Data files	5
Assessment	6
Lateness policy	6
Work submitted late for good reason	6
Acknowledgements	6
1 Introduction	7
1.1 Figures and Tables	7
1.2 Equations	8
1.3 Examples	11
1.4 Shiny apps and HTML widgets	12
2 Bells and Whistles	13
2.1 Task and solution block	13
2.2 Different task and solution titles	15
2.3 Two-tabbed solution	16
2.4 Multi-tabbed options	17
2.5 Task with held solutions	17
3 Final Words	19
3.1 What are all these files for?	19
3.2 Publish book on Moodle	23
A Answers	25

Welcome

Welcome to MTXXXX: Some stats!

An introductory-course in the field of statistical modelling in R. The focus will be on how to fit statistical models in R. The target audience is anyone who wants to learn how to fit linear models in R. The progression will be linear models, generalised linear models and linear mixed effects models.

Prerequisites

- Programming basics in R
- MTXXXX

Learning outcomes

- Understand the key concepts and terminology used in statistical modelling
- Use R to fit linear, generalised linear and mixed effect models in R
- Recognise practical issues with fitting these models
- Checking model fit
- Perform model comparisons

Recommended reading

I highly recommend the following books:

- [Statistics: An Introduction using R](#)
- [Linear models with R](#)

Data files

All data files can be found on Moodle.

Assessment

80% written exam and 20% **individual** coursework

Lateness policy

The School has a lateness [policy](#). The standard policy is an initial penalty of 15% of the maximum available mark, then a further 5% per 8-hour period, or part thereof for work submitted late without good reason.

Work submitted late for good reason

If students have a justified reason for submitting work late, then the various University's policies relating to extenuating circumstances apply. In these circumstances, students must as soon as possible submit a self-certificate of absence and contact the relevant member of School (usually the module coordinator). You will then be advised whether further documentation is required and what format this documentation will take.

Acknowledgements

We are indebted to all the statisticians who made some stats possible.

Chapter 1

Introduction

The `bookdown` package can be installed from CRAN or Github:

```
install.packages("bookdown")  
# or the development version  
# devtools::install_github("rstudio/bookdown")
```

`bookdown` is a tool for combining multiple R Markdown documents into a single book.

Each `.Rmd` file contains one and only one chapter, and a chapter is defined by the first-level heading `#` (e.g. `# Introduction`). To learn more about R Markdown read [R Markdown: The Definitive Guide](#).

To compile to PDF, you need to have LaTeX installed on your machine. `xelatex` and `lualatex` are recommended as they support Unicode better (see [here](#) and [here](#) for details). For a lightweight LaTeX installation one can use `TinyTeX` (which includes `xelatex`).

You can label chapter and section titles using the `{#label}` syntax. For example `# Introduction {#intro}` assigns the label `intro` to the introductory chapter. We can then reference the chapter using the `\@ref(label)` syntax. For example, Chapter `\@ref(intro)` renders as Chapter [1](#).

1.1 Figures and Tables

Figures and tables with captions are placed in `figure` and `table` environments, respectively.

Reference a figure by its code chunk label with the `fig:` prefix. For example, Figure `\@ref(fig:nice-fig)` renders as Figure [1.1](#).

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

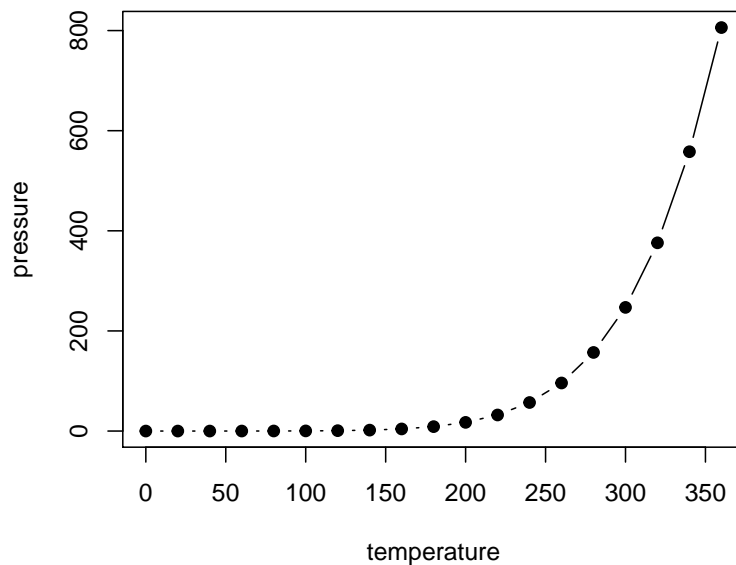


Figure 1.1: Here is a nice figure!

Similarly, tables generated by `knitr::kable()` can be referenced by their code chunk label with the `tab:` prefix. For example, Table \@ref(tab:nice-tab) renders as Table 1.1.

```
knitr::kable(
  head(iris, 10), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

You can also do a text reference for a figure caption which is useful if you want to include a mathematical expression and/or caption is very long. The syntax is “(ref:label) Some text which can include equations”.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19, col=2)
```

You can write citations, too. For example, we are using the **bookdown** package (Xie, 2020) in this sample book, which was built on top of R Markdown and **knitr** (Xie, 2015).

1.2 Equations

```
\begin{equation}
```


Table 1.1: Here is a nice table!

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

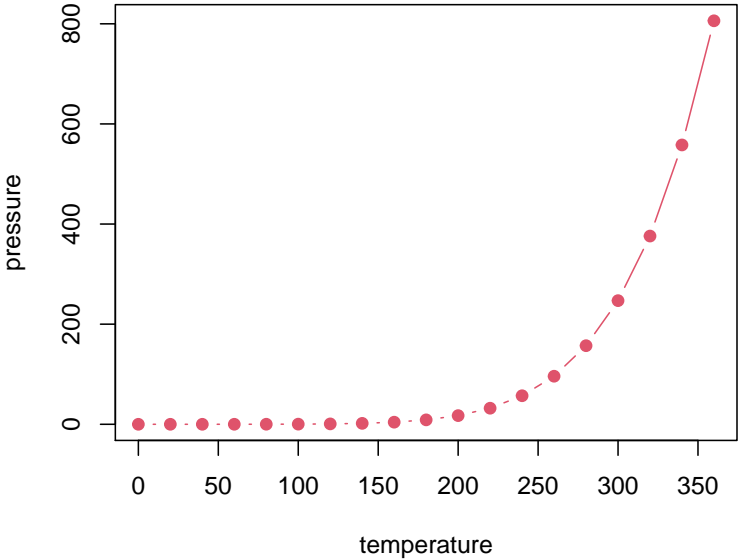


Figure 1.2: A text reference with a mathematical expression $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$ and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$.

```
f\left(k\right) = \binom{n}{k} p^k\left(1-p\right)^{n-k}
\label{eq:binom}
\end{equation}
```

renders as:

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (1.1)$$

You may refer to it using `\@ref{eq:binom}` which renders as (1.1).

Please make sure equations without labels are not numbered by either using the `equation*` environment or adding `\nonumber` or `\notag` to your equations. The same rules apply to other math environments. For aligning multiple equations, `align` is the preferred environment. For [example: /index%7Balign%7D](#)

```
\begin{align}
g(X_{\{n\}}) \&= g(\theta)+g'(\tilde{\theta})(X_{\{n\}}-\theta) \notag \\
\sqrt{n}[g(X_{\{n\}})-g(\theta)] \&= g'(\tilde{\theta})\sqrt{n}[X_{\{n\}}-\theta] \\
\sqrt{n}[X_{\{n\}}-\theta] \label{eq:align} \\
\end{align}
```

renders as

$$g(X_n) = g(\theta) + g'(\tilde{\theta})(X_n - \theta) \quad (1.2)$$

$$\sqrt{n}[g(X_n) - g(\theta)] = g'(\tilde{\theta}) \sqrt{n}[X_n - \theta] \quad (1.3)$$

Use `split` to have multiple lines with one equation reference. For example:

```
\begin{equation}
\begin{split}
\mathrm{Var}(\hat{\beta}) \&= \mathrm{Var}((X'X)^{-1}X'y) \\
&= (X'X)^{-1}X' \mathrm{Var}(y) (X'X)^{-1}X' \\
&= (X'X)^{-1}X' \mathrm{Var}(y) X (X'X)^{-1} \\
&= (X'X)^{-1}X' \sigma^2 I X (X'X)^{-1} \\
&= (X'X)^{-1} \sigma^2
\end{split}
\end{equation}
\label{eq:var-beta}
\end{equation}
```

renders as

$$\begin{aligned}
\text{Var}(\hat{\beta}) &= \text{Var}((X'X)^{-1}X'y) \\
&= (X'X)^{-1}X'\text{Var}(y)((X'X)^{-1}X')' \\
&= (X'X)^{-1}X'\text{Var}(y)X(X'X)^{-1} \\
&= (X'X)^{-1}X'\sigma^2IX(X'X)^{-1} \\
&= (X'X)^{-1}\sigma^2
\end{aligned} \tag{1.4}$$

1.3 Examples

Using the `example` environment. For example:

```

{example, eg1, name='Confidence intervals using the Normal approximation.'}
\begin{alignat*}{2}
-z_{\alpha/2} \leq {} & \text{; ; ; } Z \leq {} & z_{\alpha/2} \\
-z_{\alpha/2} \leq {} & \frac{\hat{\mu}-\mu}{\frac{\sigma}{\sqrt{n}}} \leq {} & z_{\alpha/2} \\
\hat{\mu} - z_{\alpha/2}\frac{\sigma}{\sqrt{n}} \leq {} & \mu \leq {} & \hat{\mu} + z_{\alpha/2}\frac{\sigma}{\sqrt{n}}
\end{alignat*}


```

renders as

Example 1.1 (Confidence intervals using the Normal approximation.).

$$\begin{aligned}
-z_{\alpha/2} &\leq Z \leq z_{\alpha/2} \\
-z_{\alpha/2} &\leq \frac{\hat{\mu}-\mu}{\frac{\sigma}{\sqrt{n}}} \leq z_{\alpha/2} \\
\hat{\mu} - z_{\alpha/2}\frac{\sigma}{\sqrt{n}} &\leq \mu \leq \hat{\mu} + z_{\alpha/2}\frac{\sigma}{\sqrt{n}}
\end{aligned}$$

We can reference the example by its code chunk label with the `exm:` prefix. For example, Example `\@ref{exm:eg1}` renders as Example 1.1.

There are also `theorem`, `lemma`, etc. environments. Full details can be found [here](#).

You can also create blocks of text using the simple `>` without the need for a chunk. For example,

```

> Aside: using the model to predict outside of the range of the observed data is called ext

```

renders as

Aside: using the model to predict outside of the range of the observed data is called **extrapolating**.

1.4 Shiny apps and HTML widgets

You can add [Shiny apps](#) and [HTML widgets](#) to your book.

For example, using the `knitr::include_app(...)` function we can embed a Shiny app that is hosted somewhere else. Note that this app renders well in the web version of the notes but the screenshot needs resizing in the PDF version.

An example of a shiny app. You can see a live version [here](#)

Chapter 2

Bells and Whistles

Chapter 1 briefly introduces the basic features of R Markdown and bookdown. For full details please refer to:

- [R Markdown: The Definitive Guide](#)
- [bookdown: Authoring Books and Technical Documents with R Markdown](#)

In this Chapter we are going to introduce *extra* features/environments exclusive to the St Andrews template. Big thanks to [TJ McKinley](#) for sharing with us the code for these features.

2.1 Task and solution block

The `task` block can be used to set exercises for the students. The `solution` block reveals the answer to the student (if enabled; more on that later). For the gitbook/HTML version there is a toggle button **Show Solution** that reveals the answer. In the PDF version, there is a hyperlink to take the reader to the solutions, which are in the appendix. There is also a link to get back to the place in the text from the appendix.

The `task` block is used as follows. For example:

```
```{task}
Here is a task written in markdown.
```
```

which renders as:

| Task 1 |
|---|
| Here is a task written in markdown . |

You can include chunks within the `task` chunk, but you need to use double backticks *within* the chunk, and leave carriage returns around the internal chunk. For example:

```
```{task}

``{r}
x <- 2 + 2
x
``

```
```

which renders as:

Task 2

```
x <- 2 + 2
x

## [1] 4
```

Be careful to have suitable carriage returns around e.g. `enumerate` or `itemize` environments inside the chunk also. For example:

```
```{task}
Here is a list:
1. item 1
2. item 2
```
```

will not render nicely. But

```
```{task}
Here is a list:

1. item 1
2. item 2

```
```

will:

Task 3

Here is a list:

1. item 1
2. item 2

The `solution` chunk works in the same way, and the numbers will follow the previous `task` chunk (so you can set tasks without solutions). For example:

```
```{task}
Add 2 and 2 together
```
```

```
```{solution}
```

```
``{r}
2 + 2
``
```

```
```
```

gives:

Task 4

Add 2 and 2 together

Show Solution on P25

2.2 Different task and solution titles

`task` and `solution` boxes can be given different names using the `title` option (these can be set globally if preferred). For example:

```
```{task, title = "Question"}
Produce a scatterplot of `mpg` against `hp`. What does the relationship look like?
```
```

```
```{solution, title = "Answer"}
```

```
``{r}
plot(hp ~ mpg, data = mtcars,
 pch=19, col='darkgrey')
```

```

..
...

```

renders as:

#### Question 5

Produce a scatterplot of `mpg` against `hp`. What does the relationship look like?

Show Answer on P25

### 2.3 Two-tabbed solution

You can have a task with **two** different solutions side-by-side, using the `multCode = T` option to the solution box. For example, you may want to show a solution using both base R and `tidyverse`. Here the two tabs are separated by four consecutive hashes: `####`, and the `titles` option gives the tab titles (these can be set globally if preferred). For example:

```

```{task}
Produce a scatterplot of `mpg` against `hp`. What does the relationship look like?
```

```{solution, multCode=T, titles = c("Base R", "tidyverse")}

``{r, fig.height=6, fig.width=6, out.width = "60%"}
plot(hp ~ mpg, data = mtcars,
      pch=19, col='darkgrey')
..

```

The plot suggests that a linear relationship might exist between the two variables. So we can proceed by fitting a linear model in R.

####

```

``{r, fig.height=6, fig.width=6, out.width = "60%"}
ggplot(mtcars) +
  geom_point(aes(x = mpg, y = hp))
..

```

The plot suggests that a linear relationship might exist between the two variables. So we can proceed by fitting a linear model in R.


```
```
```

will render as:

| Task 6                                                                         |
|--------------------------------------------------------------------------------|
| Produce a scatterplot of mpg against hp. What does the relationship look like? |

Show Solution on P27

## 2.4 Multi-tabbed options

You can also have just the multicode part (not embedded within the solution panel.). These appear side-by-side in the PDF document. Note that currently you can only have **two** tabs. For example:

```
```{multCode, titles=c('Part A', 'Part B')}
```

Two options:

```
* Option 1 - This is some text for part A
```

```
####
```

Two options:

```
* Option 2 - This is some text for part B
```

```
```
```

will typeset to:

| Part A                                                                                                 | Part B                                                                                                 |
|--------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| Two options: <ul style="list-style-type: none"> <li>Option 1 - This is some text for part A</li> </ul> | Two options: <ul style="list-style-type: none"> <li>Option 2 - This is some text for part B</li> </ul> |

## 2.5 Task with held solutions

In the solution chunk header, if `renderSol=FALSE` then the solutions are not rendered as part of the book. For example:

```
```{task, title='Task (solution hidden)'}
Produce a scatterplot of `mpg` against `hp`. What does the relationship look like?
```
```

```
```{solution, renderSol=FALSE}
```

```
``{r}
plot(hp ~ mpg, data = mtcars,
     pch=19, col='darkgrey')
``
```

This is my solution which you will only see if `renderSol` is set to `TRUE`.

```
```
```

will render as:

Task (solution hidden) 7

Produce a scatterplot of mpg against hp. What does the relationship look like?

By default, in `_setup.Rmd`, `renderSol` is set to `TRUE`. If one of your chapters is a tutorial/practical, and you want to release the answers later on in the course, it can become tedious having to set `renderSol` to `FALSE` for every question. Instead, you can override this default at the beginning of each chapter, so you can turn on/off the solutions, as follows:

```
``{r, include=F}
opts_chunk$set(renderSol=FALSE)
``
```

**Note** that the chunk above changes everything after the chunk, so later chapters will retain this change unless you reset it.

## Chapter 3

# Final Words

This template contains several files with different extensions and may look daunting at first. However, unless you want to customise specific aspects of this template, you *only* need to edit/add individual R Markdown files corresponding to each chapter of your course notes. Nevertheless, here's a brief description of each file in this template stratified by type.

### 3.1 What are all these files for?

#### Markdown files

- `index.Rmd`: The *only* non-optional file.
  1. Starts with a YAML configuration file to set the title, author, date and other build options.

```

title: |
 {width=0.4in} MTXXXX: Some stats
description: "This is an example gitbook with some St Andrews styling."
author: "[L. Scott-Hayward](lass@st-andrews.ac.uk) and [JJ Valletta](jjv1@st-andrews.ac.uk)"
date: '21 July 2020'
site: bookdown::bookdown_site
documentclass: book
bibliography: [book.bib, packages.bib]
biblio-style: apalike
link-citations: yes
colorlinks: yes

```

2. Sets the global `knitr` options. These options can be overridden at the start of each chapter if one wants each chapter to have its own

default options.

```
``{r setup, echo=FALSE}
library(knitr)
set knitr global options
opts_chunk$set(cache=F, tidy.opts=list(width.cutoff=55), tidy=F,
 fig.align="center", fig.width=5, fig.height=5,
 multCode=F, renderTask=T, renderSol=T)
``
```

3. Add `_setup.Rmd` to the book.

```
``{r, child = "_setup.Rmd", include = F, purl = F, cache = F}
``
```

4. Normal R Markdown. Typically contains the preface of your course notes, as shown in the template. Although it can be written as a separate `.Rmd` file for more modularity.

```
Welcome {-}
```

```
Welcome to MTXXXX: Some stats!
```

```
...
```

5. The last chunk of code in `index.Rmd` creates a BibTeX file to reference the packages used, see `packages.bib` for more details. This is optional.

- `_setup.Rmd`: This file specifies features/environments exclusive to this template; the task/solution block, multi-tabbed solutions, etc (see Chapter 2). Do not edit this file!
- `01-intro.Rmd` - `03-summary.Rmd`: These files contain all the content of your course notes. Each `.Rmd` file is a chapter and must begin with a single `#` level header (e.g. `# Linear regression`). You can have as many files as you want. `.Rmd` files are added to the book in alphabetical order. It is therefore recommended to number the files in the order you'd like them to appear. Alternatively, use `_bookdown.yml` to specify the precise order (more on that later).
- `04-references.Rmd`: Appends references at the end of the git-book/HTML book. You can edit the name of the file, but don't edit what's in it.
- `05-ch_appendix.Rmd`: Appends solutions at the end of the PDF file. You can edit the name of the file, but don't edit what's in it.
- `README.md`: Primarily intended as a landing page for the [Github repo](#) where this template is currently residing. This file can be safely deleted. Alternatively, if you don't use version control (like git), this file can be used to keep track of any major changes done to your course notes.

## YAML files

YAML files are human-readable files typically used for configuration files.

- `_bookdown.yml`: Overarching configuration file for the book; set book file name, output directory, etc. It allows you to choose which chapters to include under which build (gitbook/html or pdf/latex). This is particularly useful when you're working on a large number of chapters. Through the `rmd_files` option you can choose to only compile the chapter you're currently working on, which reduces compilation time significantly.

**Note:** The `latex` option must have the `appendix.Rmd` file as its last entry (name of file is irrelevant). This ensures that the appendix is properly formatted for the solutions to the problems. This is *only* needed for the PDF output; for gitbook/HTML, the task/solution block uses a toggle button instead.

```
book_filename: "StAndrewsTemplate"
language:
 ui:
 chapter_name: "Chapter "
delete_merged_file: true
rmd_files:
 html: ["index.Rmd", "01-intro.Rmd", "02-bellsandwhistles.Rmd", "03-summary.Rmd", "04-references", "05-appendix.Rmd"]
 latex: ["index.Rmd", "01-intro.Rmd", "02-bellsandwhistles.Rmd", "03-summary.Rmd", "04-references", "05-appendix.Rmd"]
output_dir: "docs"
```

- `_output.yml`: Configuration file to set individual options of *each* output format. For example, which TeX typesetting engine to use, which style sheet to use for gitbook/HTML, whether to split the book by chapter or sections, etc. This information can be alternatively included in the YAML portion of `index.Rmd` (under the `output:` option). However, by having it as a separate file, it makes `index.Rmd` less cluttered. No need to edit this file unless you really want to.

```
bookdown::gitbook:
 lib_dir: "book_assets"
 config:
 sharing: null
 edit: null
 download: ["pdf"]
 split_by: chapter
 highlight: pygments
 css: 'style.css'
 includes:
 in_header: '_toggle.html'

bookdown::pdf_book:
 keep_tex: TRUE
```

```

citation_package: natbib
latex_engine: xelatex
includes:
 in_header: preamble.tex
 after_body: afterbody.tex

```

## LaTeX files

- **preamble.tex**: LaTeX preamble; load packages, define new commands, etc. It also defines the task/solution boxes for the PDF.
- **afterbody.tex**: This file specifies what to include at the *end* of the PDF. Currently it simply calls `\printindex` to generate a linked index. You can add words to the index using the `\index{}` command, e.g. `\index{least-squares}`.

## Bibliography files

- **book.bib**: BibTeX file with your references. This file can be generated by most reference management software like [Mendeley](#), [Zotero](#), etc.
- **packages.bib**: This BibTeX file is created from the last chunk of code in `index.Rmd` file. It compiles the citations of the specified packages.

```

``{r include=FALSE}
automatically create a bib database for R packages
knitr::write_bib(c(
 .packages(), 'bookdown', 'knitr', 'rmarkdown', 'ggplot2'
), 'packages.bib')
``

```

## Web files

- **style.css**: CSS files describe the “look” of a webpage (colours, font sizes, etc.). Here, it describes the styling of the gitbook and includes the style of the task/solution blocks, the colour of the figure captions etc. The St Andrews colours used can be found [here](#). Edit at your own risk!
- **toggle.html**. This is a javascript file for rendering the show/hide solution toggle button. Do not edit.

## Hidden files

- **.gitignore/.gitattributes**: Ignore/delete unless you use git for version control.

## 3.2 Publish book on Moodle

Once you have finished compiling your course notes into a nice book, it is time to share it with your students on Moodle by following these steps:

1. Compress the folder containing the compiled course notes as a `.zip` file<sup>1</sup>. If you do not use version control, it is recommended that you rename this zip file by adding a timestamp (e.g. `docs_20200721.zip`), in case you want to archive older versions.
2. Log into your course page on Moodle and turn editing on.
3. Navigate to the section/topic where you want to upload the course notes and press “+ Add an activity or resource”. Select “File” rather than “Folder” and upload the `.zip` file.
4. Click on the zipped file and select “unzip”.
5. Click on the unzipped folder, select `index.html` and “Set main file”.
6. Save and return to course.

---

<sup>1</sup>The name of this folder is set in `output_dir` in `_bookdown.yml`, e.g. `docs`





## Appendix A

## Answers

### Solution 4

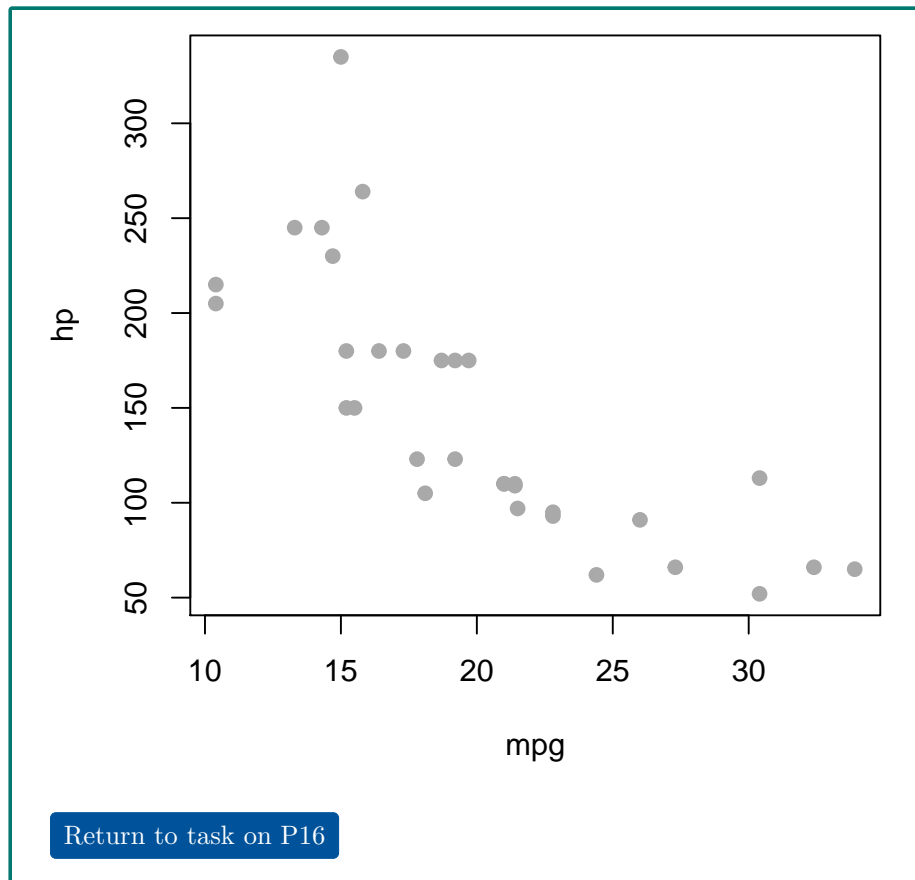
```
2 + 2
```

```
[1] 4
```

[Return to task on P15](#)

### Answer 5

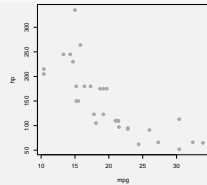
```
plot(hp ~ mpg, data = mtcars,
 pch=19, col='darkgrey')
```



## Solution 6

## Base R

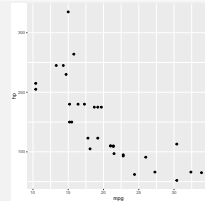
```
plot(hp ~ mpg, data = mtcars,
 pch=19, col='darkgrey')
```



The plot suggests that a linear relationship might exist between the two variables. So we can proceed by fitting a linear model in R.

## tidyverse

```
ggplot(mtcars) +
 geom_point(aes(x = mpg, y = hp))
```



The plot suggests that a linear relationship might exist between the two variables. So we can proceed by fitting a linear model in R.

[Return to task on P17](#)



# Bibliography

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2020). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.20.

# Index

confidence intervals, [11](#)

equations, [10](#)