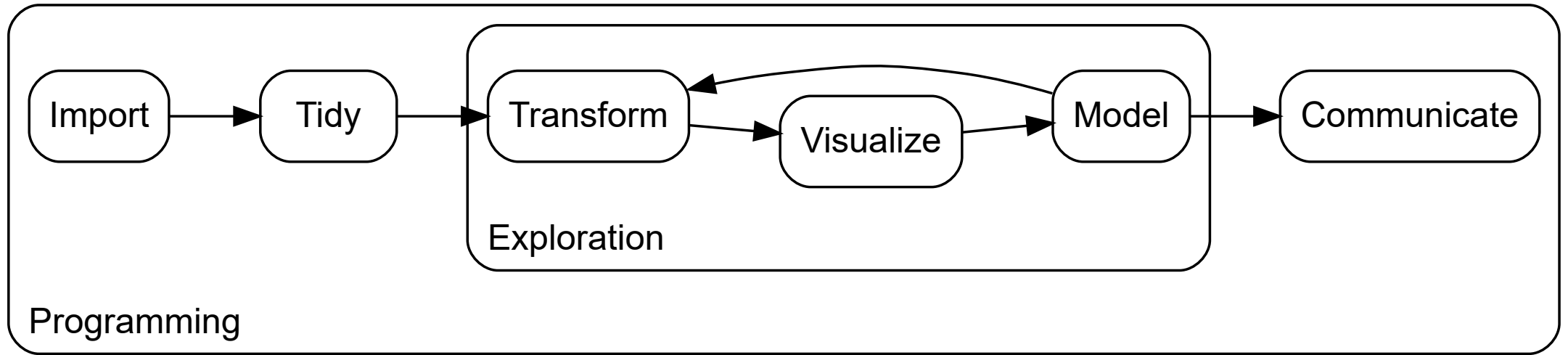


Data Analysis using R

Reporting

Sven Werenbeck-Ueding

18.11.2024



Source: [Wickham and Grolemund \(2016\)](#)

Introduction

It is quite common to perform data analyses and communicate the results in a separate document that...

... has to be updated manually whenever results change.

... is usually not version-controlled.

... is, unfortunately, oftentimes a rather poorly formatted MS Office document.

Quarto

Quarto is an open-source tool developed by Posit (the developers behind, among others, RStudio and the tidyverse) that builds on the popular universal document converter Pandoc and allows for:

- Creating dynamic documents and web-content with R (and Python, Julia and Observable)
- Writing reports as plain text markdown
- Compiling reports to PDF, HTML, MS Word and more
- Scientific notations, display of complex equations, automatically formatted citations, cross-referencing and more

To get started, download the newest Quarto version [here](#).

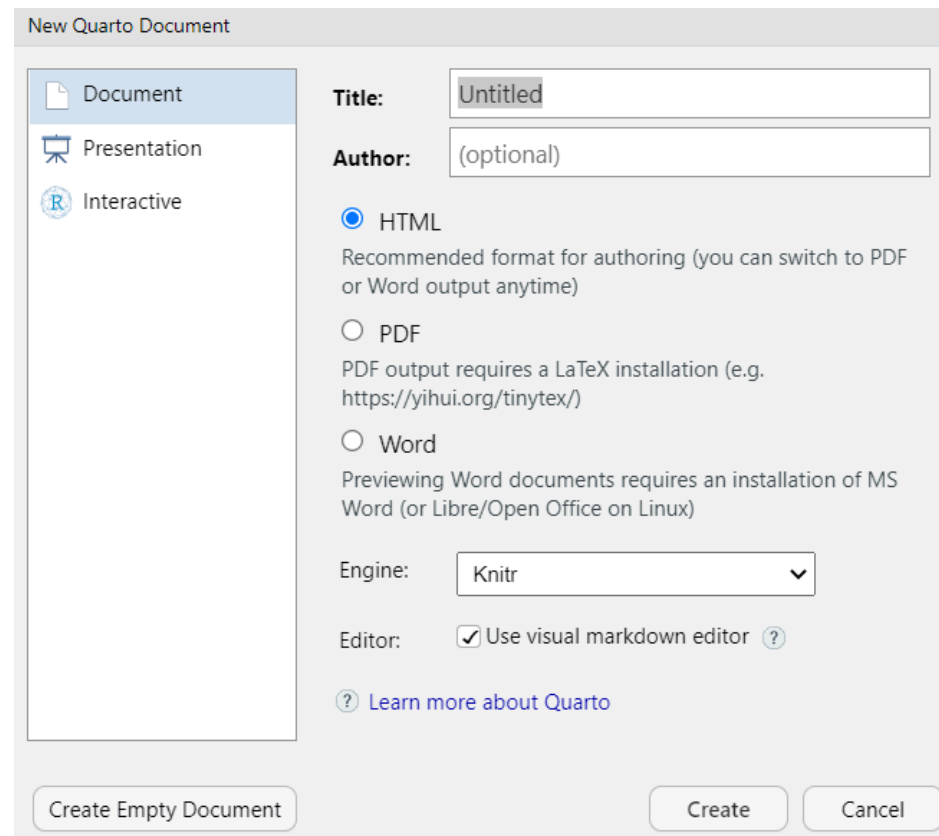
Installing TinyTeX

- For compiling reports to PDF, a TeX distribution is needed on your device
- It is recommended to use [TinyTeX](#) that is targeted primarily at R users
- Quarto offers a built-in function to install TinyTeX to your system
- Type the following command in the RStudio terminal:

```
quarto install tinytex
```

Creating a Quarto Document

- Click on File > New File > Quarto Document... to create a Quarto document
- Provide document title, author and output format now (or define it later)
- Uncheck the box for the visual editor



The screenshot shows the 'New Quarto Document' dialog box. On the left, there is a sidebar with three options: 'Document' (selected with a blue highlight), 'Presentation', and 'Interactive'. The main area on the right contains the following fields and options:

- Title:** A text input field containing 'Untitled'.
- Author:** A text input field containing '(optional)'.
- Output Format:** Three radio button options:
 - ☒ **HTML**: Recommended format for authoring (you can switch to PDF or Word output anytime)
 - ☐ **PDF**: PDF output requires a LaTeX installation (e.g. <https://yihui.org/tinytex/>)
 - ☐ **Word**: Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux)
- Engine:** A dropdown menu showing 'Knitr'.
- Editor:** A checkbox labeled 'Use visual markdown editor' which is checked.
- A link: [? Learn more about Quarto](#)

At the bottom, there are three buttons: 'Create Empty Document' (disabled), 'Create', and 'Cancel'.

Structure of Quarto Documents

A Quarto document consists of...

... a YAML header

```
---  
title: "Border Fences and Migration"  
subtitle: "Replication of Feigenberg (2020)"  
author: "Sven Werenbeck-Ueding"  
format: beamer  
---
```

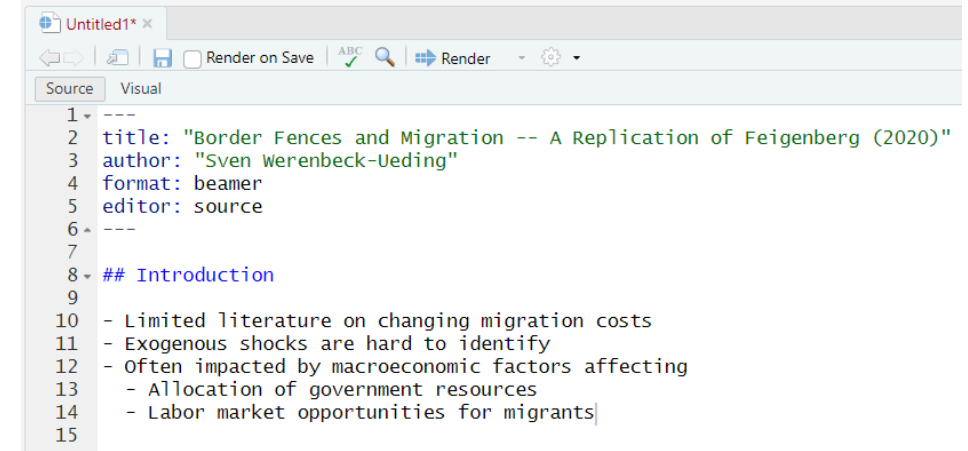
... a document body

```
## Introduction
```

- Limited literature on changing migration costs
- Exogenous shocks are hard to identify
- Often impacted by macroeconomic factors affecting
 - Allocation of government resources
 - Labor market opportunities for migrants

Rendering Quarto Documents

- Click on the blue arrow labeled "Render"
- Or use the keyboard shortcut `Ctrl+Shift+K`
- File needs to be saved before being able to render



```
1 ---  
2 title: "Border Fences and Migration -- A Replication of Feigenberg (2020)"  
3 author: "Sven Werenbeck-Ueding"  
4 format: beamer  
5 editor: source  
6 ---  
7  
8 ## Introduction  
9  
10 - Limited literature on changing migration costs  
11 - Exogenous shocks are hard to identify  
12 - often impacted by macroeconomic factors affecting  
13   - Allocation of government resources  
14   - Labor market opportunities for migrants  
15
```


How does it work?

When clicking on rendering (or using the keyboard shortcut `Ctrl+Shift+K`), `knitr` executes the code contained in the file and generates a markdown document (.md) that includes the code and its output along with the markdown text in the Quarto document.

Pandoc then processes the markdown file to the format defined in the YAML header.



Quarto (2023)

YAML Header

- Placed at the top and specifies metadata and customization options of the document
- Begins and ends with three dashes (---)
- Given by key-value pairs in the format key: value
- The output format can be defined with the format key
 - format: pdf for PDF documents
 - format: beamer for LaTeX-based PDF presentations
 - We will focus on creating beamer presentations
- Setting editor to source let's RStudio open the file as raw source text
- For a full list of YAML fields for beamer presentations, see [here](#)

```
---  
title: "Border Fences and Migration"  
subtitle: "Replication of Feigenberg (2020)"  
author: "Sven Werenbeck-Ueding"  
date: "`r format(Sys.Date(), '%d.%m.%Y')`"  
aspectratio: 169  
format: beamer  
editor: source  
---
```

Code Chunks

- The huge benefit of using Quarto is to be able to compile code when rendering the document
- R code chunks begin with three backticks (``) and {r} and ends with three backticks
- Optional chunk options can be specified using the YAML key-value pair format key: value beginning with #| in the line after the beginning of the chunk
- Code inside the chunk is executed when rendering the document

Markdown

Output

```
```${r}
#| label: mpg_summary

library(tidyverse)

mtcars %>%
 group_by(cyl) %>%
 summarize(mean_mpg = round(mean(mpg), 2))
```
```

Code Chunks

- The huge benefit of using Quarto is to be able to compile code when rendering the document
- R code chunks begin with three backticks (```) and `{r}` and ends with three backticks
- Optional chunk options can be specified using the YAML key-value pair format `key: value` beginning with `#|` in the line after the beginning of the chunk
- Code inside the chunk is executed when rendering the document

Markdown

Output

```
library(tidyverse)

mtcars %>%
  group_by(cyl) %>%
  summarize(mean_mpg = round(mean(mpg), 2))
```

```
## # A tibble: 3 × 2
##   cyl mean_mpg
##   <dbl>   <dbl>
## 1     4    26.7
## 2     6    19.7
## 3     8    15.1
```

Useful Chunk Options

| Key | Values | Description |
|---------|------------|--|
| eval | true/false | Controls whether or not the code should be compiled. |
| echo | true/false | Controls whether or not the source code should be included in the rendered document. |
| message | true/false | Controls whether or not code messages should be included in the rendered document. |
| warning | true/false | Controls whether or not code warnings should be included in the rendered document. |



- Chunk options for customizing figures can be found [here](#)
- Chunk options for customizing tables can be found [here](#)
- Default chunk options can be set for the whole document in the YAML header

Creating Beamer Slides

- Slide titles are defined by two hashtags (##)
- To create slides without a title, separate the slides with three dashes (---)
- Comment your Quarto document with %
- Slides can be divided into sections using a single hashtag (#)

Note: There should be no text, code chunk or comment between section title and slide title!

```
---  
title: "Some Title for a Research Project"  
format: beamer  
editor: source  
---  
  
# First section  
  
## Title of the first slide  
  
% This is some comment that is not rendered  
  
This is the slide body  
  
---  
  
This is a slide without title  
but with code chunk  
  
```${r}```  
plot(mtcars$hp ~ mtcars$mpg)
```
```

Writing Text

- Text in Quarto documents is written in markdown syntax
- Markdown is a lightweight syntax used for adding formatting of plain texts
- When format is pdf or beamer, LaTeX expressions can be used as if one would write a LaTeX document
- Markdown expressions can be used to define section headers, italic text, bold text, bullet points, enumerations, and more
- Different from e. g. MS Word, where text formatting has to be added by clicking on the respective elements in the user interface

Emphasize Text

- *Italic*: Surround text with single asterisks *
- **Bold**: Surround text with two asterisks **
- ~~Strikethrough~~: Surround text with two tildes ~~
- ***Combine emphasis*** of text by surrounding it with three asterisks ***

```
---  
title: "Some Title for a Research Project"  
format: beamer  
editor: source  
---  
  
## Some slide title  
  
This is *italic*  
  
This is **bold**  
  
This is ~~striked through~~  
  
This is ***bold and italic***
```


Lists

- Lists with Bullet point can be created by beginning a line with a dash (–)
- Enumerations can be created by beginning a line with a digit followed by a dot (e. g. 1.)
- To create nested lists, indent the bullet points

Note: Lists have to be separated from paragraphs by an empty line!

```
---
title: "Some Title for a Research Project"
format: beamer
editor: source
---
```

Some slide title

Be sure to separate paragraph and list by an empty line!

- First bullet point
 - Bullet point with indent
 - Another bullet point
- Another first-level bullet point
- ...

1. First element of an ordered list
2. Second element
3. ...

Equations

- Equations in **inline math mode** can be written by surrounding the equation by dollar signs (\$), e. g. `$y_i=X_i\beta+\varepsilon_i$` renders to $y_i = X_i\beta + \varepsilon_i$
- Equations in **display math mode** can be written by surrounding the equation by two dollar signs (\$\$), e. g. surrounding the equation above with two dollar signs on each side renders to

$$y_i = X_i\beta + \varepsilon_i$$

- For more information on how to write equations, see the documentation on mathematical expressions in LaTeX on [Overleaf](#)

Figures

- Figures can be included using

```
! [<fig-caption>] (<fig-path>) {<attributes>}
```

- Resizing is done by either providing width or height in the figure attributes
 - Add width=50% in the curly brackets to resize the width to 50% of the original figure width
 - Can also specify pixels or inches
 - Height is then chosen automatically
 - Could also provide only height or both
- Alignment can be changed using the fig-align attribute ("left", "center", "right")
- Multiple attributes are separated by a white space only (no comma!)

Code	Output
<pre>! [] (../.. /resources/assets/example.jpg)</pre>	

Figures

- Figures can be included using

```
! [<fig-caption>] (<fig-path>) {<attributes>}
```

- Resizing is done by either providing width or height in the figure attributes
 - Add `width=50%` in the curly brackets to resize the width to 50% of the original figure width
 - Can also specify pixels or inches
 - Height is then chosen automatically
 - Could also provide only height or both
- Alignment can be changed using the `fig-align` attribute ("left", "center", "right")
- Multiple attributes are separated by a white space only (no comma!)

Code

Output



Figures

- Figures can be included using

```
![<fig-caption>](<fig-path>){<attributes>}
```

- Resizing is done by either providing width or height in the figure attributes
 - Add `width=50%` in the curly brackets to resize the width to 50% of the original figure width
 - Can also specify pixels or inches
 - Height is then chosen automatically
 - Could also provide only height or both
- Alignment can be changed using the `fig-align` attribute ("left", "center", "right")
- Multiple attributes are separated by a white space only (no comma!)

Code

Output

```
![A not at all staged photo...](../../resources,
```

Figures

- Figures can be included using

```
![<fig-caption>](<fig-path>){<attributes>}
```

- Resizing is done by either providing width or height in the figure attributes
 - Add `width=50%` in the curly brackets to resize the width to 50% of the original figure width
 - Can also specify pixels or inches
 - Height is then chosen automatically
 - Could also provide only height or both
- Alignment can be changed using the `fig-align` attribute ("left", "center", "right")
- Multiple attributes are separated by a white space only (no comma!)

Code

Output



A not at all staged photo...

Figures

- Figures can be included using

`![<fig-caption>](<fig-path>){<attributes>}`

- Resizing is done by either providing width or height in the figure attributes
 - Add `width=50%` in the curly brackets to resize the width to 50% of the original figure width
 - Can also specify pixels or inches
 - Height is then chosen automatically
 - Could also provide only height or both
- Alignment can be changed using the `fig-align` attribute ("left", "center", "right")
- Multiple attributes are separated by a white space only (no comma!)

Code	Output
<code>![A not at all staged photo...](../../resources,</code>	

Figures

- Figures can be included using

```
![<fig-caption>](<fig-path>){<attributes>}
```

- Resizing is done by either providing width or height in the figure attributes
 - Add `width=50%` in the curly brackets to resize the width to 50% of the original figure width
 - Can also specify pixels or inches
 - Height is then chosen automatically
 - Could also provide only height or both
- Alignment can be changed using the `fig-align` attribute ("left", "center", "right")
- Multiple attributes are separated by a white space only (no comma!)

Code

Output



A not at all staged photo...

Cross-Referencing Figures

- Use `#fig-<reference_ID>` in the attributes to add a cross-reference ID to figures
- Throughout the document, you can simply write `@fig-<reference_ID>` to reference the corresponding figure
- Be sure to add the prefix `fig-`!

Figures from URL

We can also include a figure from an URL by giving the link to the figure in the parentheses:

```

```



Linked Figures

To link figures, simply enclose the figure within a link, i. e. `[](<url>):`

```
[

# Figures from Code

- Figures from code chunks are automatically inserted
- To add a reference ID, caption and/or link, specify the chunk options `label`, `fig-cap` or `fig-link`
- The size of the figure can be controlled via `fig-width/fig-height`
- See the [Quarto documentation](#) for more options
- If your code creates multiple plots, these can be shown in a subfigure environment with a panel layout (see the Quarto documentation [here](#))

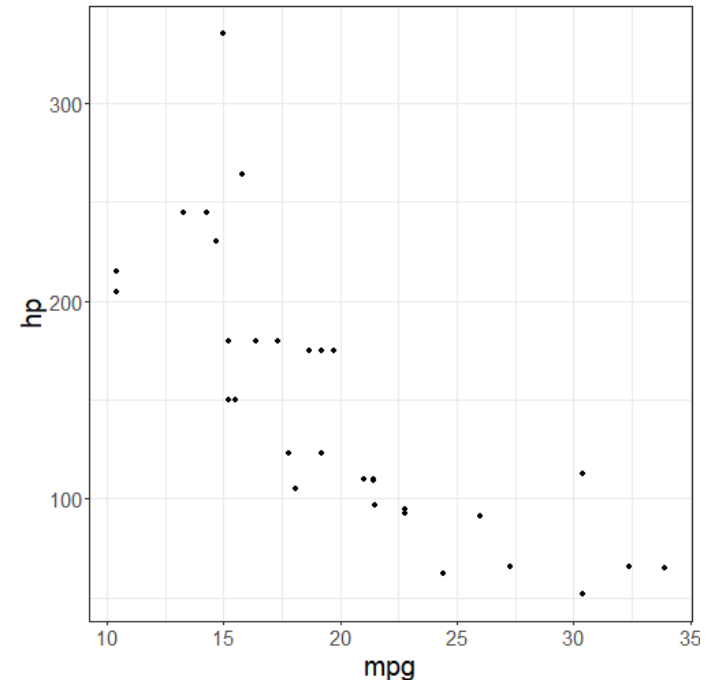
| Code                                                                                                                                                                                                                                                                                                                           | Output |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| <pre>`` `{r} #  echo: false #  label: mtcars #  fig-cap: "Horse Power vs. Miles per Gallon" #  fig-width: "75%" #  fig-align: "center"  mtcars %&gt;%   ggplot(aes(x = mpg, y = hp)) +   geom_point() +   theme_bw() +   theme(     axis.title = element_text(size = 20),     axis.text = element_text(size = 15)   ) ``</pre> |        |

# Figures from Code

- Figures from code chunks are automatically inserted
- To add a reference ID, caption and/or link, specify the chunk options `label`, `fig-cap` or `fig-link`
- The size of the figure can be controlled via `fig-width/fig-height`
- See the [Quarto documentation](#) for more options
- If your code creates multiple plots, these can be shown in a subfigure environment with a panel layout (see the Quarto documentation [here](#))

Code

Output



Horse Power vs. Miles per Gallon

# Tables in R

- Many packages offer functions for exporting tables in different formats
- An easy to use function resides in the knitr package: `kable()`

The `kable()` function in knitr is a very simple table generator, and is simple by design. It only generates tables for strictly rectangular data such as matrices and data frames. You cannot heavily format the table cells or merge cells. However, this function does have a large number of arguments for you to customize the appearance of tables.

[Xie, Dervieux, and Riederer \(2023\)](#)

```
kable(
 x, # matrix or data.frame
 format, # e. g. "latex" or "html"
 digits = getOption("digits"),
 row.names = NA,
 col.names = NA,
 align, # Column alignment
 caption = NULL,
 label = NULL, # reference label
 format.args = list(),
 escape = TRUE,
 ...
)
```

- If format is left out, the appropriate format will be chosen by knitr's global option `knitr.table.format`, when rendering the document

# Example: Summary Statistics

| Task | Code | Output |
|------|------|--------|
|------|------|--------|

For the `mtcars` data set, calculate the mean and standard deviation of

- horse power (`hp`),
- gas consumption (`mpg`), and
- acceleration (`qsec`)

for each category of cylinders (`cyl`).

# Example: Summary Statistics

| Task | Code | Output |
|------|------|--------|
|------|------|--------|

```
df_sum_stats <- mtcars %>%
 group_by(cyl) %>%
 summarise(
 across(c(hp, mpg, qsec),
 .fns = list(mean = mean, sd = sd))
) %>%
 pivot_longer(
 c(ends_with("mean"), ends_with("sd")),
 names_to = c("metric", ".value"),
 names_pattern = c("(.*)_(.*)")
) %>%
 pivot_wider(
 names_from = cyl,
 values_from = c(mean, sd),
 names_glue = "{cyl}_{.value}",
 names_vary = "slowest"
)
df_sum_stats
```



# Example: Summary Statistics

| Task | Code                                                                                                                                                                                                                                                                                                                                                              | Output |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
|      | <pre>## # A tibble: 3 × 7 ##   metric `4_mean` `4_sd` `6_mean` `6_sd` `8_mean` `8_sd` ##   &lt;chr&gt;    &lt;dbl&gt; &lt;dbl&gt;    &lt;dbl&gt; &lt;dbl&gt;    &lt;dbl&gt; &lt;dbl&gt; ## 1 hp      82.6  20.9    122.   24.3    209.   51.0 ## 2 mpg     26.7   4.51    19.7   1.45    15.1   2.56 ## 3 qsec    19.1   1.68    18.0   1.71    16.8   1.20</pre> |        |



Already in a tabular format ready for display

# Example: Create a Table

| Task | Code | Output |
|------|------|--------|
|------|------|--------|

Use the tabular summary statistics to create a table using `knitr's kable()` function. Provide a

- caption,
- reasonable column names, and
- a label for cross-referencing.

Choose an appropriate formatting for the values.

# Example: Create a Table

| Task | Code | Output |
|------|------|--------|
|------|------|--------|

```
df_sum_stats %>%
 mutate(
 metric = case_when(
 metric == "hp" ~ "Horse Power",
 metric == "mpg" ~ "Miles per Gallon",
 metric == "qsec" ~ "1/4 Mile in sec."
)
) %>%
 knitr::kable(
 col.names = c("", rep(c("Mean", "Std. Dev."), 3)),
 caption = "Summary Statistics",
 label = "sum_stats",
 format.args = list(digits = 2, nsmall = 2)
)
```

# Example: Create a Table

| Task | Code | Output |
|------|------|--------|
|------|------|--------|

Table: Summary Statistics

|                  | Mean  | Std. Dev. | Mean   | Std. Dev. | Mean   | Std. Dev. |
|------------------|-------|-----------|--------|-----------|--------|-----------|
| Horse Power      | 82.64 | 20.93     | 122.29 | 24.26     | 209.21 | 50.98     |
| Miles per Gallon | 26.66 | 4.51      | 19.74  | 1.45      | 15.10  | 2.56      |
| 1/4 Mile in sec. | 19.14 | 1.68      | 17.98  | 1.71      | 16.77  | 1.20      |



No further customization (grouped header, footnote, ...) without considerable effort, i. e. including raw LaTeX (or HTML) code

# kableExtra

[...] the ultimate simplicity of `kable()` also brought troubles [...]. It is not rare to see people including experienced users asking questions like how to center/left-align a table on Stack Overflow. [...] For LaTeX, it's even worse [...]

- Use default base `kable()` [...] for all simple tables
- Use `kable()` with `kableExtra` to generate 90 % of complex/advanced/self-customized/beautiful tables in either HTML or LaTeX
- Only have to mess with raw HTML/LaTeX in the last 10% cases where `kableExtra` cannot solve the problem

Zhu (2023)

# Package Features

```
install.packages("kableExtra")
library(kableExtra)
```

- kableExtra does not change the way how tables are generated using kable
- Adds convenient functions for customizing a table
  - group\_rows()
  - footnote()
  - add\_header\_above()
  - kable\_styling()
  - ...
- Functions can be piped and usually support both HTML and PDF format
- Replaced kable() with kbl() to generate the table (better documentation and format detection)
  - Includes the booktab argument which enables the booktab format from the corresponding LaTeX package (highly recommend)
- See the [documentation](#) for more information

# Example: Customized Summary Statistics Table

| Task | Code | Output |
|------|------|--------|
|------|------|--------|

Add further customization to the table created before using the `kableExtra` package. For this, add

- a header to group the columns by their respective cylinder category, and
- footnote below the table.

*Hint:* Use `kableExtra`'s `add_header_above()` and `footnote()` functions.

# Example: Customized Summary Statistics Table

| Task | Code | Output |
|------|------|--------|
|------|------|--------|

```
df_sum_stats %>%
 mutate(
 metric = case_when(
 metric == "hp" ~ "Horse Power",
 metric == "mpg" ~ "Miles per Gallon",
 metric == "qsec" ~ "1/4 Mile in sec."
)
) %>%
 kbl(
 col.names = c("", rep(c("Mean", "Std. Dev."), 3)),
 caption = "Summary Statistics",
 label = "sum_stats",
 format.args = list(digits = 2, nsmall = 2),
 booktabs = T
) %>%
 add_header_above(c("", "4 Cylinders" = 2, "6 Cylinders" = 2, "8 Cylinders" = 2)) %>%
 footnote("The data was taken from base R's mtcars data set.")
```



# Example: Customized Summary Statistics Table

| Task | Code | Output |
|------|------|--------|
|------|------|--------|

## Summary Statistics

|                  | 4 Cylinders |           | 6 Cylinders |           | 8 Cylinders |           |
|------------------|-------------|-----------|-------------|-----------|-------------|-----------|
|                  | Mean        | Std. Dev. | Mean        | Std. Dev. | Mean        | Std. Dev. |
| Horse Power      | 82.64       | 20.93     | 122.29      | 24.26     | 209.21      | 50.98     |
| Miles per Gallon | 26.66       | 4.51      | 19.74       | 1.45      | 15.10       | 2.56      |
| 1/4 Mile in sec. | 19.14       | 1.68      | 17.98       | 1.71      | 16.77       | 1.20      |

*Note:*

The data was taken from base R's mtcars data set.

# Tables in Quarto

- Quarto allows for convenient compilation of tables, enabling the author to
  - Specify column alignment and widths
  - Provide captions, subcaptions, and cross-references
  - Generate tables dynamically from code chunks
- Options can be added in the respective code chunk
  - `label` for cross-referencing
  - `tbl-cap` for table captions
  - ...
- For cross-references, add the `tbl-` prefix to your reference, e. g. `@tbl-sum_stats`
- See the [documentation](#) for more information

# Bibliography

Quarto uses Pandoc to automatically generate citations and a bibliography from citations provided in the document and a bibliographic data source (such as BibLaTeX files) referenced in the YAML header:

---

|      |          |
|------|----------|
| YAML | BibLaTeX |
|------|----------|

---

- The bibliographic data source is defined in the bibliography key
- If the bibliography does not reside in the same directory as the Quarto document, the path has to be adjusted accordingly
- You can set the citation method and bibliography style manually if necessary. For further information see [here](#)

```

title: "Border Fences and Migration"
subtitle: "Replication of Feigenberg (2020)"
author: "Sven Werenbeck-Ueding"
date: "2023-01-02"
aspectratio: 169
format: beamer
bibliography: references.bib
editor: source

```

# Bibliography

Quarto uses Pandoc to automatically generate citations and a bibliography from citations provided in the document and a bibliographic data source (such as BibLaTeX files) referenced in the YAML header:

| YAML | BibLaTeX |
|------|----------|
|------|----------|

```
@article{f20,
 Author = {Feigenberg, Benjamin},
 Title = {Fenced Out: The Impact of Border Construction on US-Mexico Migration},
 Journal = {American Economic Journal: Applied Economics},
 Volume = {12},
 Number = {3},
 Year = {2020},
 Month = {July},
 Pages = {106-39},
 DOI = {10.1257/app.20170231}
}
```

# Citations

- Citations must have a key consisting of @ and the citation identifier in the bibliography file
- Multiple citations in parantheses must be separated by semicolons
- You can also write inside squared parantheses if, for example, you want to add "see" before the citation
- For a consistent naming of citation identifiers, I recommend to use the first letter of the authors last name and the year of publication, e. g. Feigenberg (2020) → f20

| Markdown          | Output                                                    |
|-------------------|-----------------------------------------------------------|
| @f20              | Feigenberg (2020)                                         |
| [@f20]            | (Feigenberg, 2020)                                        |
| [@atw19;<br>@f20] | (Athey, Tibshirani, and Wager, 2019;<br>Feigenberg, 2020) |

# References Slide

- Pandoc automatically generates a references slide at the end of the document
- If the user provides a `div` with the reference `{#refs}`, the bibliography will be placed there instead
- To keep more control over the bibliography slide(s), add the markdown text on the right to the end of your document
- To allow automatic slide breaks in the references (if you have many citations in the bibliography spanning multiple slides), simply add `{.allowframebreaks}` to the slide header

```
References {.allowframebreaks}
```

```
::: {#refs}
```

```
:::
```

# Multi-Column Layout

| Code | Output |
|------|--------|
|------|--------|

```
:::: {.columns}
::: {.column width="50%"}
- Data was taken from the mtcars data set that comes with R

- Gas consumption of cars decreases with horsepower

- Some further insights from the figure
:::

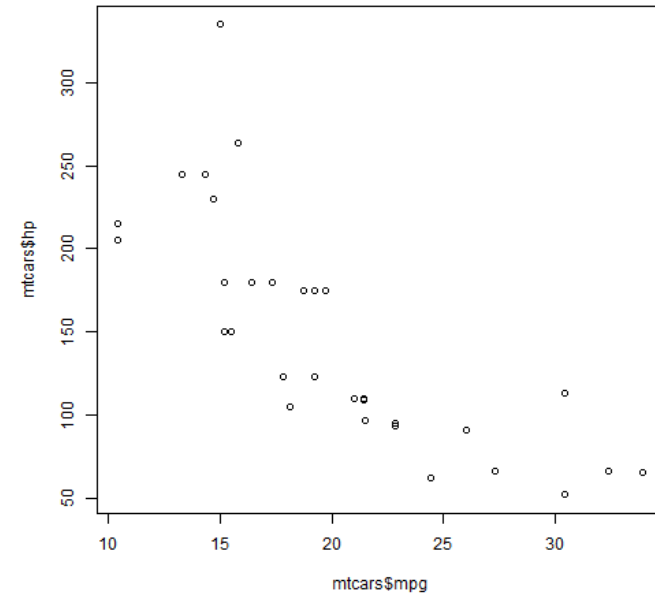
:::: {.column width="50%"}
```{r}
#| echo: false
#| out.width: "66%"
#| fig.align: "center"

plot(mtcars$hp ~ mtcars$mpg)
```
:::
::::
```

# Multi-Column Layout

| Code | Output |
|------|--------|
|------|--------|

- Data was taken from the mtcars data set that comes with R
- Gas consumption of cars decreases with horsepower
- Some further insights from the figure





# Themes

- Themes can be changed by defining the theme in the YAML header
- Same applies to color themes and the `colortheme` key
- A full list of beamer themes can be found [here](#)
  - Rows correspond to themes
  - Columns correspond to color themes

```

title: "Border Fences and Migration"
subtitle: "Replication of Feigenberg (2020)"
author: "Sven Werenbeck-Ueding"
date: "2023-01-02"
aspectratio: 169
format:
 beamer:
 theme: Frankfurt
 colortheme: seahorse
bibliography: references.bib
editor: source

```

# Adjusting Themes

- Beamer themes are generally not that aesthetically pleasant
- You may want to tweak the colors of your chosen theme
- This can be done by referencing a LaTeX file (.tex) in the YAML header key `include-in-header` of the beamer format
- Quarto puts the content of that file in the header, i. e. before the document body begins, when rendering the LaTeX output
- You can find an example for a custom theme in the repository of this course

```

title: "Border Fences and Migration"
subtitle: "Replication of Feigenberg (2020)"
author: "Sven Werenbeck-Ueding"
date: "2023-01-02"
aspectratio: 169
format:
 beamer:
 theme: Frankfurt
 include-in-header: custom_theme.tex
bibliography: references.bib
editor: source

```

# References

Athey, S., J. Tibshirani, and S. Wager (2019). "Generalized random forests". In: *The Annals of Statistics* 47.2, pp. 1148 - 1178. DOI: [10.1214/18-AOS1709](https://doi.org/10.1214/18-AOS1709). URL: <https://doi.org/10.1214/18-AOS1709>.

Feigenberg, B. (2020). "Fenced Out: The Impact of Border Construction on US-Mexico Migration". In: *American Economic Journal: Applied Economics* 12.3, pp. 106-39. DOI: [10.1257/app.20170231](https://doi.org/10.1257/app.20170231). URL: <https://www.aeaweb.org/articles?id=10.1257/app.20170231>.

Quarto (2023). *Tutorial: Hello, Quarto*. URL: <https://quarto.org/docs/get-started/hello/rstudio.html#overview> (visited on Jan. 02, 2023).

Wickham, H. and G. Grolemund (2016). *R for data science. import, tidy, transform, visualize, and model data*. O'Reilly. URL: <https://r4ds.had.co.nz/>.

Xie, Y., C. Dervieux, and E. Riederer (2023). *R Markdown Cookbook*. Chapman & Hall/CRC. URL: <https://bookdown.org/yihui/rmarkdown-cookbook/>.

Zhu, H. (2023). *kableExtra*. URL: <https://github.com/haozhu233/kableExtra/> (visited on Sep. 17, 2023).