

# Erstellung von Berichten und Folien mit Quarto

rstatsZH - Data Science mit R

Lars Schöbitz

Nov 19, 2024

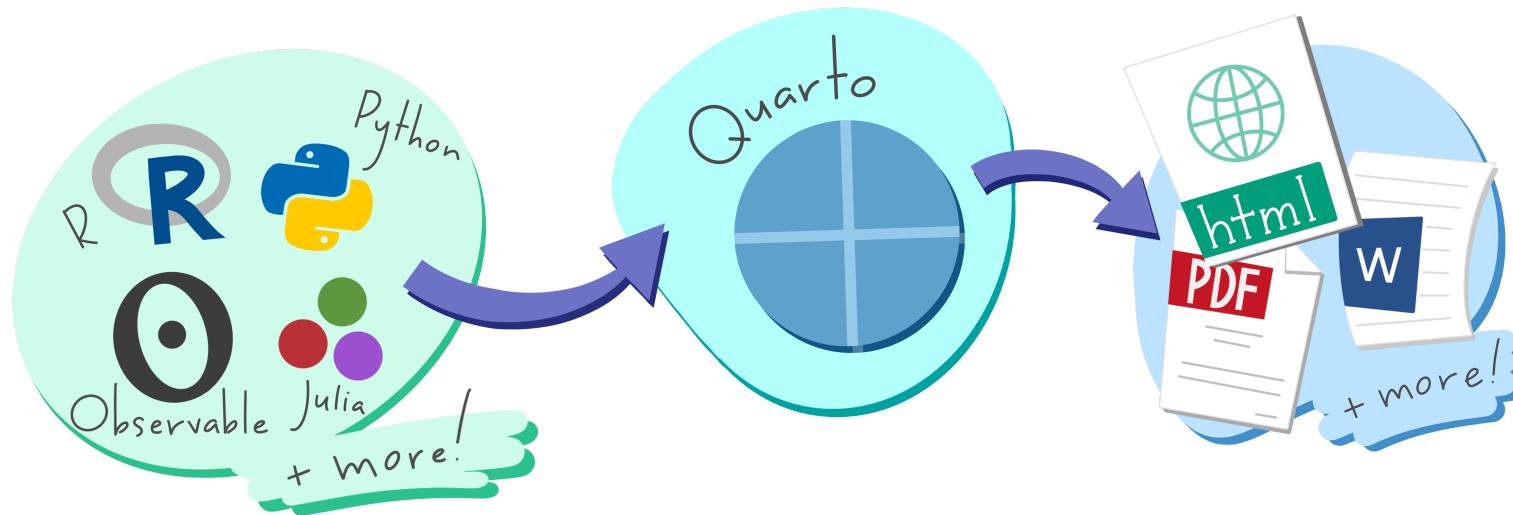
# Lernziele (für diese Woche)

1. Die Lernenden können Querverweise zu Diagrammen und Tabellen in Quarto Dokumenten erstellen.
2. Die Lernenden können Folien im Format revealjs HTML erstellen.
3. Die Lernenden können Metadaten eines Quarto Dokuments anpassen.

# Quarto

# Quarto ...

- ist ein neues, open-source, wissenschaftliches und technisches Publikationssystem
- zielt darauf ab, den Prozess der Erstellung von Berichten und der Zusammenarbeit drastisch zu verbessern



Artwork from “Hello, Quarto” keynote by Julia Lowndes and Mine Çetinkaya-Rundel, presented at RStudio Conference 2022. Illustrated by Allison Horst.

# Artikel & Berichte (HTML)

[quarto-dev.github.io/quarto-gallery/articles/html/html.html](https://quarto-dev.github.io/quarto-gallery/articles/html/html.html)

# Artikel & Berichte (HTML)

Besuche [quarto.org!](https://quarto.org)

[quarto.org/docs/output-formats/html-basics.html](https://quarto.org/docs/output-formats/html-basics.html)

# Folien (HTML revealjs)

[apreshill.github.io/palmerpenguins-useR-2022/#/title-slide](https://apreshill.github.io/palmerpenguins-useR-2022/#/title-slide)

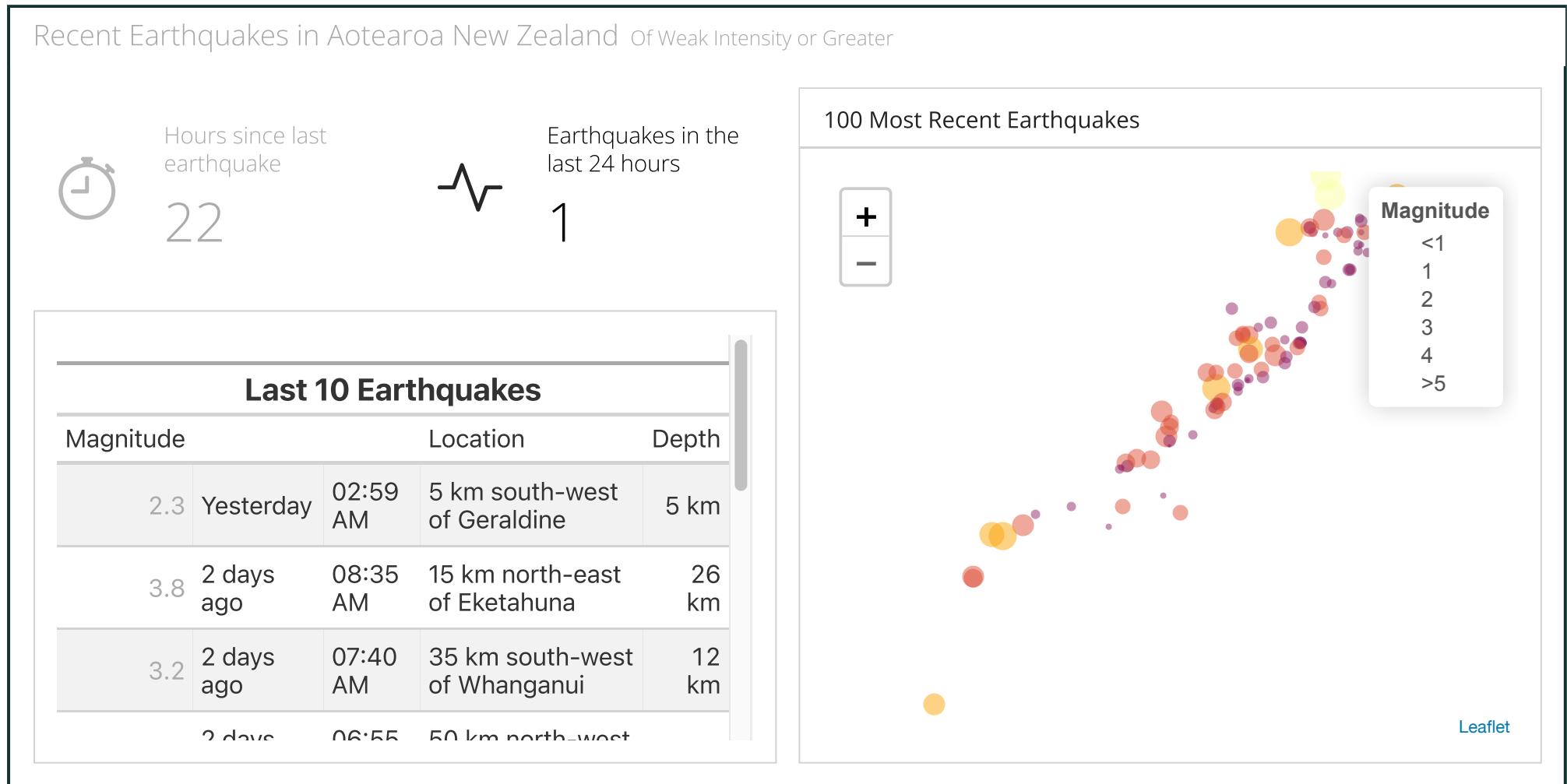
# Folien (HTML revealjs)

Besuche [quarto.org!](https://quarto.org)

[quarto.org/docs/presentations/revealjs/](https://quarto.org/docs/presentations/revealjs/)

# Dashboards

[pub.demo.posit.team/public/nz-quakes/quakes.html](http://pub.demo.posit.team/public/nz-quakes/quakes.html)



# Dashboards

Besuche [quarto.org!](https://quarto.org)

[quarto.org/docs/dashboards/](https://quarto.org/docs/dashboards/)

# Gallery auf quarto.org

[quarto.org/docs/gallery/](https://quarto.org/docs/gallery/)

The screenshot shows the Quarto Gallery website. At the top, there is a navigation bar with links to Overview, Get Started, Guide, Extensions, Reference, **Gallery**, Blog, Help, and social media icons for Twitter, GitHub, RSS, and a search icon.

## Gallery

Quarto can produce a wide variety of output formats. Here are some examples:

- Articles & Reports
- Presentations
- Dashboards
- Websites
- Books
- Interactive Docs

### 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`.

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

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

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

Figure 2: A full width figure.

Articles and reports with Python and R

# Anatomie eines Quarto-Dokuments

# Komponenten

1. Metadaten: YAML
2. Text: Markdown
3. Code: Ausgeführt via `knitr` (R) oder `jupyter` (python)

**Füge alles zusammen** und du erhältst anschauliche, schöne und nützliche Ergebnisse!

# Literate programming

DE: wortgetreue Programmierung

Bei der wortgetreuen Programmierung wird die Programmlogik in einer menschlichen Sprache mit eingeschlossenen Codeschnipseln geschrieben.

```
1 ---  
2 title: "ggplot2 demo"  
3 date: "19/11/2024"  
4 format: html  
5 ---  
6  
7 ## Pinguine  
8  
9 Es gibt einen linearen Zusammenhang zwischen Länge und Tiefe des Schnabels, abhängig von der A:  
10  
11 ````{r}  
12 library(ggplot2)  
13 library(palmerpenguins)  
14  
15 ggplot(penguins, aes(x = bill_length_mm,  
16                      y = bill_depth_mm,  
17                      color = species)) +  
18   geom_point()  
19 ````
```

# Metadaten

# YAML

“Yet Another Markup Language” oder “YAML Ain’t Markup Language” wird verwendet, um Metadaten auf Dokumentebene bereitzustellen.

```
1  ---
2  key: value
3  ---
```

# Output (Ausgabe) Optionen

```
1 ---  
2 format: etwas  
3 ---
```

```
1 ---  
2 format: html  
3 ---
```

```
1 ---  
2 format: pdf  
3 ---
```

```
1 ---  
2 format: revealjs  
3 ---
```

# Output (Ausgabe) Optionen Argumente

Einrückung ist wichtig!

```
1  ---
2  format:
3    html:
4      toc: true
5      code-fold: true
6  ---
```

# YAML Validierung

- Ungültig: Kein Leerzeichen nach :

```
1 ---  
2 format:html  
3 ---
```

- Ungültig: Als fehlend gelesen

```
1 ---  
2 format:  
3 html  
4 ---
```

# YAML Validierung

Es gibt mehrere Möglichkeiten, gültiges YAML zu formatieren:

- Gültig: Es gibt ein Leerzeichen nach :

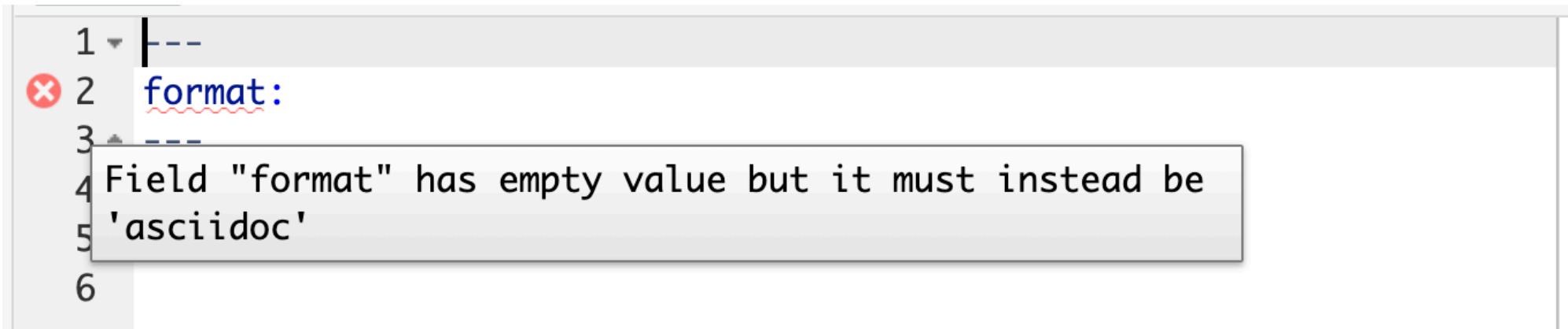
```
1 format: html
```

- Gültig: `format: html` mit Auswahlen, die mit korrekter Einrückung erfolgen

```
1 format:  
2   html:  
3     toc: true
```

# Quarto linter

Lint oder ein Linter ist ein statisches Code-Analyse-Tool, das Programmierfehler, Bugs, stilistische Fehler und verdächtige Konstrukte aufzeigt.



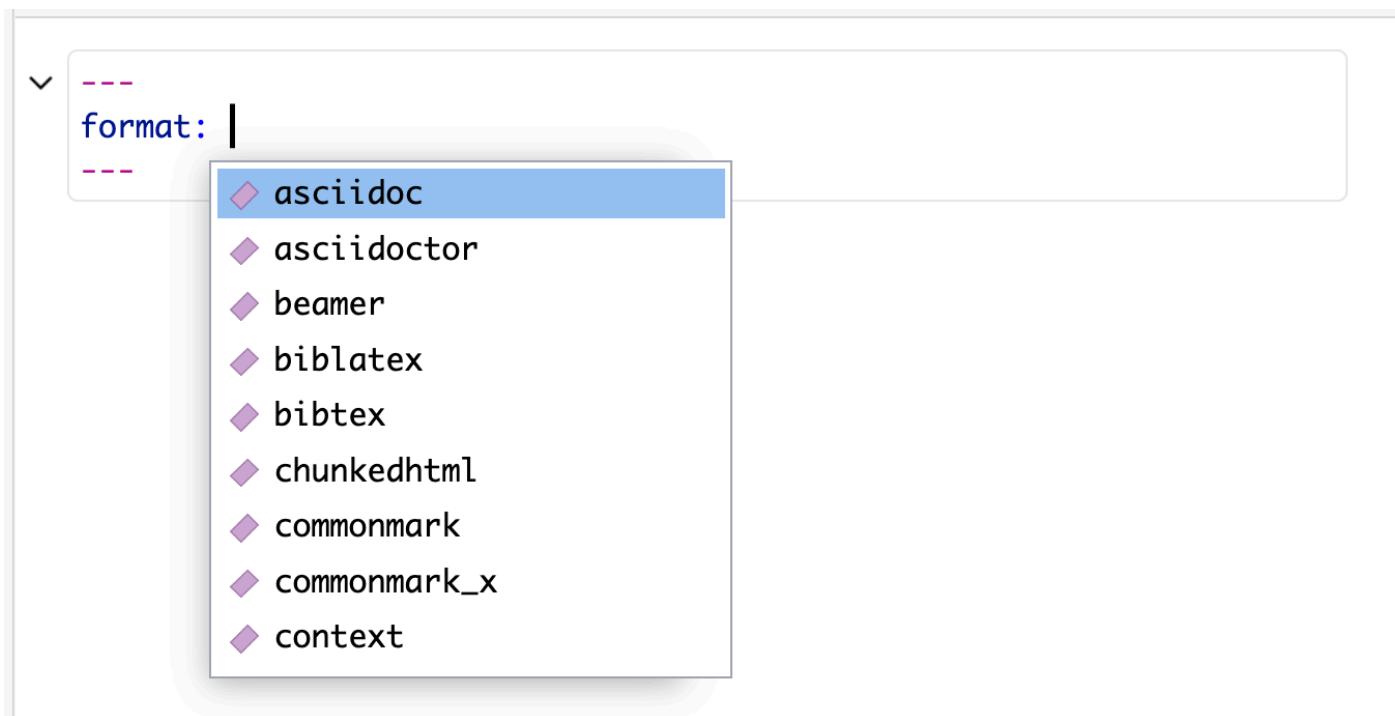
The screenshot shows a code editor window with the following content:

```
1 ---  
2 format:  
3 ---  
4 Field "format" has empty value but it must instead be  
5 'asciidoc'  
6
```

A red circle with a white 'X' is positioned next to the line number 2, indicating an error. A tooltip or callout box is overlaid on the code, highlighting the word 'format:' and providing the error message: 'Field "format" has empty value but it must instead be 'asciidoc''. The rest of the code and line numbers 1, 3, 4, 5, and 6 are visible below the error message.

# Quarto YAML Intelligenz

RStudio + VSCode bieten eine umfangreiche Tab-Vervollständigung – fange mit einem Wort an und tippe auf die Tab-Taste, um es zu vervollständigen, oder Strg + Leertaste um alle möglichen Optionen zu sehen.



# Liste der gültigen YAML-Felder

- Viele YAML-Felder sind bei verschiedenen Ausgaben gleich
- Aber auch jeder Ausgabetyp hat einen eigenen Satz an gültigen YAML-Feldern und Optionen
- Komplette Liste: [quarto.org/docs/reference/formats/html](https://quarto.org/docs/reference/formats/html)

# Text

# Text Formatierung

Markdown Syntax	Ausgabe
*italics* and **bold**	<i>italics</i> and <b>bold</b>
superscript^2^ / subscript~2~	superscript <sup>2</sup> / subscript <sub>2</sub>
~~strikethrough~~	<del>strikethrough</del>
`verbatim code`	verbatim code

# Headings (Überschriften)

## Markdown Syntax

```
# Header 1
```

## Ausgabe

**Header  
1**

---

```
## Header 2
```

**Header 2**

---

```
### Header 3
```

**Header 3**

---

```
#### Header 4
```

**Header 4**

---

```
##### Header 5
```

**Header 5**

---

```
##### Header 6
```

**Header 6**

# Links

Es gibt verschiedene Arten von “Links” oder Hyperlinks.

## Markdown

```
1 Du kannst [benannte Hyperlinks] (https://quarto.org/),  
2 direkte URLs wie <https://quarto.org/> und Links zu  
3 [anderen Orten] (#quarto-anatomy) in  
4 das Dokument einbetten. Die Syntax für das Einbetten eines  
5 Inline-Bildes ist ähnlich: ![Pinguine spielen mit Ball](img/md-0
```

## Ausgabe

Du kannst [benannte Hyperlinks] (<https://quarto.org/>), direkte URLs wie <https://quarto.org/> und Links zu [anderen Orten](#) in das Dokument einbetten. Die Syntax für das Einbetten eines Inline-Bildes ist ähnlich:



# Listen

Ungeordnete Liste:

Markdown:

```
1 - unordered list
2   - sub-item 1
3   - sub-item 1
4     - sub-sub-item 1
```

Ausgabe

- unordered list
  - sub-item 1
  - sub-item 1
    - sub-sub-item 1

Geordnete Liste:

Markdown:

```
1 1. ordered list
2 2. item 2
3   i. sub-item 1
4     A. sub-sub-item 1
```

Ausgabe

1. ordered list
2. item 2
  - i. sub-item 1
    - a. sub-sub-item 1

# Zitate

## Markdown:

```
1 > Let us change our traditional attitude to the construction of p  
2 > - Donald Knuth, Literate Programming
```

## Output:

Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do. - Donald Knuth, Literate Programming

# Fussnoten

Die Nummerierung und Formatierung von Fussnoten wird unterstützt.

# Inline-Fussnoten

Hier ist eine Inline-Notiz.<sup>1</sup> [Inline-Notizen sind einfacher zu schreiben, da man nicht erst einen Bezeichner auswählen und nach unten gehen muss, um um die Notiz zu schreiben.]

Hier ist eine Inline-Notiz.<sup>1</sup>



Verwende den visuellen Editor-Modus (Visual editor)

Für das Hinzufügen von Fussnoten wird der visuelle Editor-Modus empfohlen.

# Ihr seid dran: 01-markdown-syntax.qmd

1. Öffne [posit.cloud](#) in deinem Browser (verwende dein Lesezeichen).
2. Öffne den rstatszh-k009 Arbeitsbereich (Workspace) für den Kurs.
3. Klicke auf Start neben md-09-uebungen.
4. Suche im Dateimanager im Fenster unten rechts die Datei 01-markdown-syntax.qmd und klicke darauf, um sie im Fenster oben links zu öffnen.
5. Verwende den Quelltexteditor (Source editor) Modus
6. Folge den Anweisungen in der Datei.

# Pause machen

Bitte steh auf und beweg dich. Lasst eure E-Mails in Frieden ruhen.

# Code

# Anatomie eines Code-Block

```
1 ````{r}
2 #| label: penguins
3 #| message: false
4
5 library(tidyverse)
6 library(palmerpenguins)
7 library(knitr)
8
9 penguins |>
10   count(species) |>
11   kable()
12 ````
```

- Hat 3x Backticks an jedem Ende
- Engine (`r`) wird zwischen geschweiften Klammern `{r}` angegeben
- Optionen, die mit dem Zeichen `# |` angegeben werden (Hashpipe)

species	n
Adelie	152
Chinstrap	68
Gentoo	124

# Code, für wen ist es?

- Die Art und Weise, wie du Code anzeigenst, ist in verschiedenen Kontexten sehr unterschiedlich.
- In einem Lehrszenario wie heute möchte ich *wirklich* Code anzeigen.
- In einem Unternehmen möchtest du vielleicht eine datenwissenschaftliche Ausgabe haben, die den Quellcode anzeigt, UND eine Ausgabe, die sich an die Nutzer:innen richtet und den Code verbirgt.

# Ein- und Ausblenden von Code mit echo

- Die Option `echo` zeigt den Code an, wenn sie auf `true` gesetzt ist, und verbirgt ihn, wenn sie auf `false` gesetzt ist.

# Tabellen und Diagramme

- In reproduzierbaren Berichten und Manuskripten sind **Tabellen** und **Figuren** die am häufigsten enthaltenen Codeausgaben.

# Tabellen

# Markdown Tabellen

## Markdown:

1	Right	Left	Default	Center
2	-----:	:-----	-----:	-----:
3	12	12	12	12
4	123	123	123	123
5	1	1	1	1



Verwende den visuellen Editor-Modus (Visual editor)

Für das Hinzufügen von Fussnoten wird der visuelle Editor-Modus empfohlen.

## Ausgabe:

Right	Left	Default	Center
12	12	12	12
123	123	123	123
1	1	1	1

# Tabellen mit Code

Das Paket knitr kann Datenrahmen in Tabellen umwandeln mit `knitr::kable()`:

species	island	n
Adelie	Biscoe	44
Adelie	Dream	56
Adelie	Torgersen	52
Chinstrap	Dream	68
Gentoo	Biscoe	124

# Tabellen mit Code

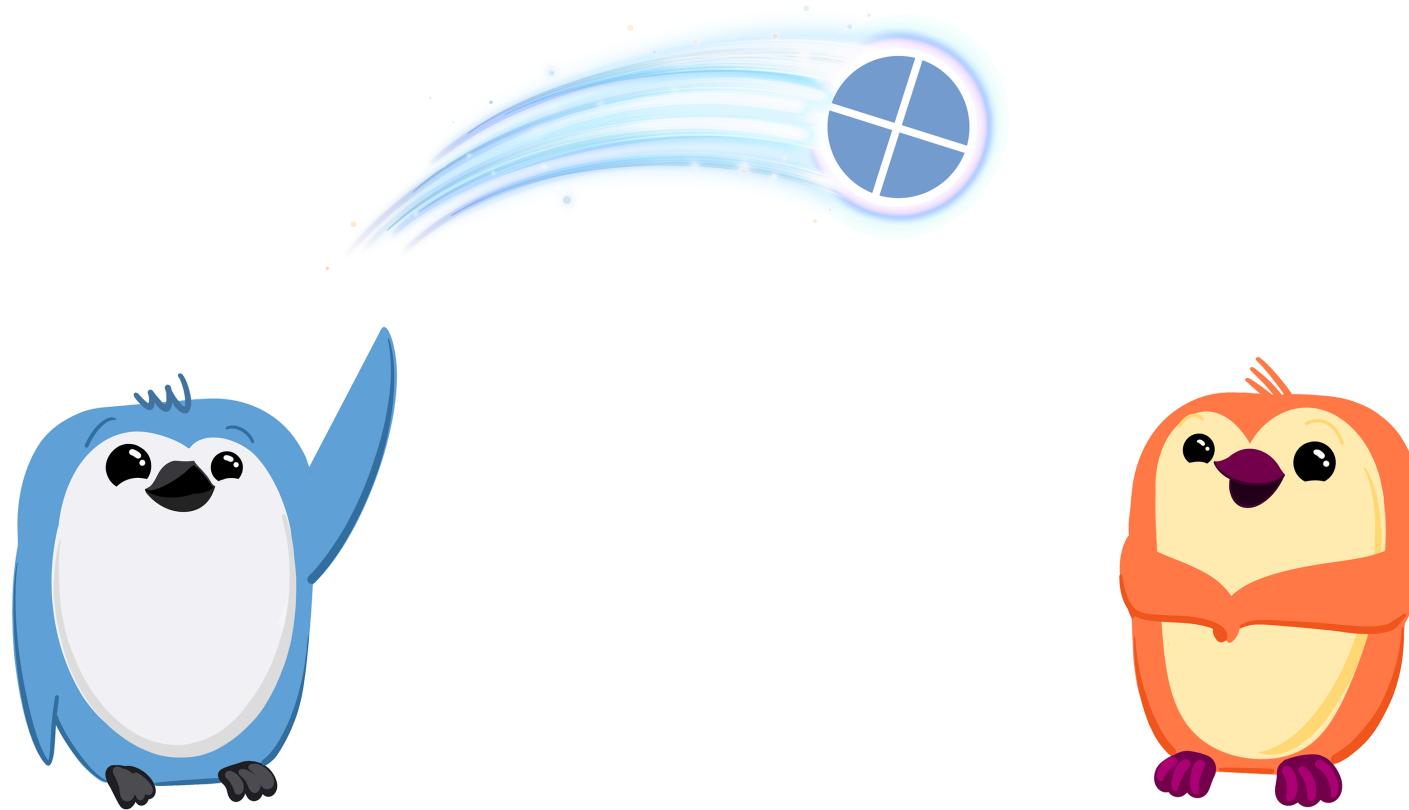
Wenn du umfangreichere Tabellen möchtest, probiere das **gt-Paket** und alles, was es bietet!

species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
Adelie	Torgersen	39.1	18.7	181	3750	male	2007
Adelie	Torgersen	39.5	17.4	186	3800	female	2007
Adelie	Torgersen	40.3	18.0	195	3250	female	2007
Adelie	Torgersen	NA	NA	NA	NA	NA	2007
Adelie	Torgersen	36.7	19.3	193	3450	female	2007
Adelie	Torgersen	39.3	20.6	190	3650	male	2007

# Abbildungen & Diagramme

# Markdown Abbildungen

![Pinguine spielen mit einem Quarto-Ball](img/md-09/penguins-quarto-ball.png)



Pinguine spielen mit einem Quarto-Ball

# Unterabbildungen

## Markdown:

```
:::: {#fig-penguins layout-ncol=2}
```

```
![Blauer Pinguin](img/md-09/blue-penguin.png){#fig-blue  
width="250px"}
```

```
![Orangefarbener Pinguin](img/md-09/orange-penguin.png){#fig-orange  
width="250px"}
```

Zwei Pinguine

```
:::
```

# Unterabbildungen

Ausgabe:



(a) Blauer Pinguin



(b) Orangefarbener Pinguin

Figure 1: Zwei Pinguine

# Suche nach den Abbildungen

An Orten wie Markdown, YAML oder beim Einlesen von Dateien musst du **absolute** oder **relative** Dateipfade verwenden:

- Absolut = Schlecht: "/Users/lars/uebungen" - Auf wessen Computer wird das funktionieren?
- Relativ = Besser:
  - ".../" = ein Verzeichnis nach oben, ".../.../" = zwei Verzeichnisse nach oben, etc.
  - "/.." oder "/" = beginnen im "root"-Verzeichnis deines aktuellen Computers

# Abbildungen mit Code (Diagramme)

```
1  ````{r}
2  #| fig-width: 4
3  #| fig-align: right
4
5  knitr:::include_graphics("img/md-09/penguins-quarto-ball.png")
6  ````
```



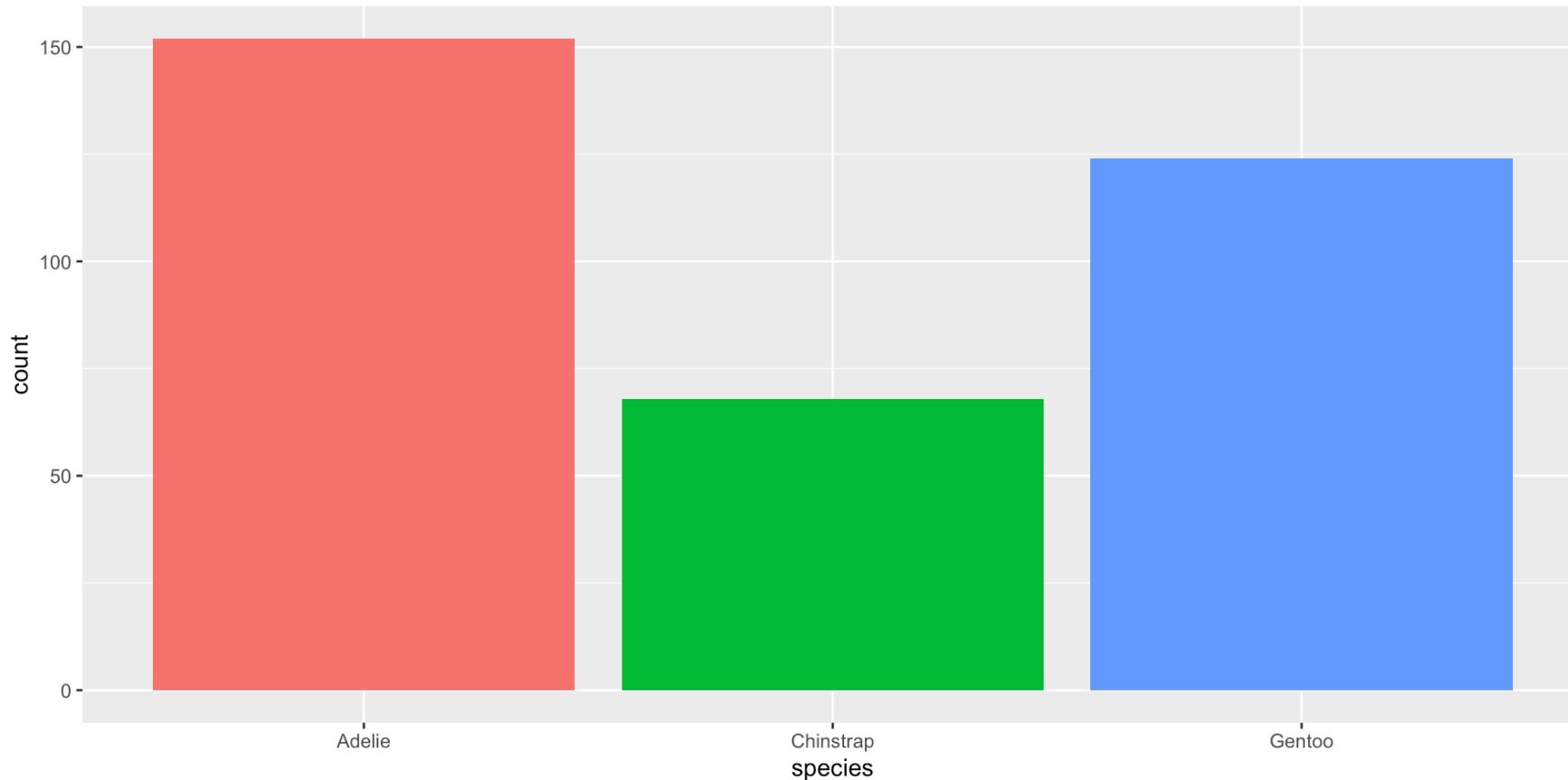
# Pfade in R-Code referenzieren

Verwende das `here`-Paket, um auf das Projektverzeichnis zu verweisen, da `here::here()` immer im obersten Verzeichnis einer `.Rproj` beginnt:

```
[1] "/Users/lschoebitz/Documents/gitrepos/gh-org-rstatszh-k009/website"
```

# Abbildungen mit Code (Diagramme)

```
1  ```{r}
2  ggplot(penguins, aes(x = species, fill = species)) +
3    geom_bar(show.legend = FALSE)
4  ```
```



# Querverweise (cross references)

# Querverweise (cross references)

- Unterstützen Leser:innen sich in deinem Dokument zurechtzufinden, indem du nummerierte Verweise und Hyperlinks zu Elementen wie Abbildungen und Tabellen einfügst.
- Querverweise Schritte:
  - Füge der Abbildung oder Tabelle eine Beschriftung hinzu: # | **tbl-cap:** oder # | **fig-cap:**.
  - Füge der Abbildung oder Tabelle eine Bezeichungen hinzu: # | **label:** beginnend mit **fig-** oder **tbl-**.
  - Verweise im Text auf die Abbildung oder Tabelle mit **@fig-...** oder **@tbl-....**

# Abbildung Querverweise

Das Vorhandensein der Beschriftung ([Blauer Pinguin](#)) und der Bezeichnung (#fig-blue-penguin) machen diese Abbildung referenzierbar:

## Markdown:

```
1 Siehe @fig-blue-penguin für einen süßen blauen Pinguin.  
2 ![Blauer Pinguin](img/md-09/blue-penguin.png){#fig-blue-penguin}
```

## Ausgabe:

Siehe [Figure 2](#) für einen süßen blauen Pinguin.



Figure 2: Blauer Pinguin

# Tabelle Querverweise

Das Vorhandensein der Überschrift ([Ein paar Pinguine](#)) und der Bezeichnung (#tbl-pinguine) machen diese Tabelle referenzierbar:

## Markdown:

```
1 Siehe @tbl-penguins für Daten über einige Pinguine
2
3 ````{r}
4 #| label: tbl-pinguine
5 #| tbl-cap: Ein paar Pinguine
6
7 head(penguins) |>
8   gt()
9 ````
```

## Ausgabe:

Siehe [Table 1](#) für Daten über einige Pinguine.

Table 1: Ein paar Pinguine

species	island	bill_length_mm	bill_depth_mm	flipper_length_mm
Adelie	Torgersen	39.1	18.7	18.0
Adelie	Torgersen	39.5	17.4	18.0
Adelie	Torgersen	40.3	18.0	18.0
Adelie	Torgersen	NA	NA	NA
Adelie	Torgersen	36.7	19.3	19.0
Adelie	Torgersen	39.3	20.6	19.0

# Querverweise Schritte:

1. Füge der Abbildung oder Tabelle eine Beschriftung hinzu:

- `# | tbl-cap:` oder `# | fig-cap::`

2. Füge der Abbildung oder Tabelle eine Bezeichungen hinzu:

- `# | label:` beginnend mit `fig-` oder `tbl-`.

3. Verweise im Text auf die Abbildung oder Tabelle mit:

- `@fig-...` oder `@tbl-....`

# Ihr seid dran: 02-querverweise.qmd

1. Öffne [posit.cloud](#) in deinem Browser (verwende dein Lesezeichen).
2. Öffne den rstatszh-k009 Arbeitsbereich (Workspace) für den Kurs.
3. Klicke auf **Continue** neben **md-09-uebungen**.
4. Suche im Dateimanager im Fenster unten rechts die Datei **02-querverweise.qmd** und klicke darauf, um sie im Fenster oben links zu öffnen.
5. Verwende den **Quelltexteditor (Source editor)** Modus
6. Folge den Anweisungen in der Datei.

# Geht's auch auf Deutsch?

Ja!

```
1  ---
2  lang: de
3  ---
```

# Pause machen

Bitte steh auf und beweg dich. Lasst eure E-Mails in Frieden ruhen.

# Anatomie einer Quarto-Präsentation (Folien)

# Komponenten

1. Metadaten: YAML
2. Text: Markdown
3. Code: Ausgeführt via `knitr` (R) oder `jupyter` (python)

**Füge alles zusammen und du erhältst anschauliche, schöne und nützliche Folien!**

# Wir sind dran: 03-folien.qmd

1. Öffne [posit.cloud](#) in deinem Browser (verwende dein Lesezeichen).
2. Öffne den rstatszh-k009 Arbeitsbereich (Workspace) für den Kurs.
3. Klicke auf **Continue** neben **md-09-uebungen**.
4. Suche im Dateimanager im Fenster unten rechts die Datei **03-folien.qmd** und klicke darauf, um sie im Fenster oben links zu öffnen.

# Ihr seid dran: 03-folien.qmd

Mach da weiter, wo wir aufgehört haben, und:

- Ändere den Folien Übergang ([transition style](#)) zwischen den Folien (auf quarto.org: Suche nach “slide transitions” im  Suchfeld oben rechts)
- Füge Folien Nummern ([slide numbers](#)) zu deinen Folien (auf quarto.org: Suche nach “slide numbers” im  Suchfeld oben rechts)

# Quarto Präsentation Formate

# Quarto Präsentation Formate

- revealjs für HTML-Folien
  - Im Wesentlichen der Ersatz für xaringan, aber mit Pandoc-nativer Syntax
- beamer für LaTeX-Folien
- PowerPoint für den Fall, dass du über MS Office zusammenarbeiten *musst*

# Besuche quarto.org

[quarto.org/docs/presentations](https://quarto.org/docs/presentations)

The screenshot shows the Quarto website's navigation bar at the top, featuring the Quarto logo, a search icon, and social media links. Below the navigation, a breadcrumb trail indicates the current page is 'Guide > Presentations > Overview'. The main content area has a large title 'Presentations' and a sub-section 'Overview'. A paragraph explains that Quarto supports various presentation formats like revealjs, pptx, and beamer. A bulleted list details these formats: 'revealjs — reveal.js (HTML)', 'pptx — PowerPoint (MS Office)', and 'beamer — Beamer (LaTeX/PDF)'. A note below states that while there are pros and cons to each, revealjs is the most capable and recommended unless specific requirements exist.

Guide > Presentations > Overview

# Presentations

## Overview

Quarto supports a variety of formats for creating presentations, including:

- `revealjs` — reveal.js (HTML)
- `pptx` — PowerPoint (MS Office)
- `beamer` — Beamer (LaTeX/PDF)

There are pros and cons to each of these formats. The most capable format by far is `revealjs`, so it is highly recommended unless you have specific Office or LaTeX output requirements. Note that `revealjs` presentations can be presented as HTML

# Zeitpuffer: Modul 9

Kann ich noch etwas zum  
heutigen Modul erklären?

# Zusatzaufgaben Modul 9

# Modul 9 Dokumentation

[rstatszh-k009.github.io/website/module/md-09.html](https://rstatszh-k009.github.io/website/module/md-09.html)

# Zusatzaufgaben Abgabedatum

- Abgabedatum: Montag, 25. November
- Korrektur- und Feedbackphase bis zu: Donnerstag, 28. November

# Danke

Danke! 🌻

Folien erstellt mit revealjs und Quarto:

<https://quarto.org/docs/presentations/revealjs/>

Zugang zu Folien als [PDF auf GitHub](#)

Alle Materialien sind lizenziert unter [Creative Commons Attribution Share Alike 4.0 International.](#)