

2. Dein erstes Quarto-Dokument

Von der leeren Datei zum gerenderten Word-Dokument

Dr. Paul Schmidt

In diesem Kapitel erstellen wir ein einfaches Quarto-Dokument von Grund auf und rendern es zu Microsoft Word. Dabei lernt man die grundlegenden Bausteine kennen: den YAML-Header, Markdown-Textformatierung, R-Code-Chunks und Inline-Code.

Ein neues Quarto-Dokument erstellen

In RStudio (oder Positron) geht man zu **File** → **New File** → **Quarto Document**. Es erscheint ein Dialog wie dieser:

Wähle vorerst **Word** als Ausgabeformat und klicke auf **Create**. RStudio generiert eine Vorlagendatei mit einem Beispielinhalt.

💡 Tipp

Man kann eine neue `.qmd`-Datei auch manuell erstellen — es ist einfach eine Textdatei. Der Dialog bietet lediglich einen praktischen Startpunkt.

Der YAML-Header

Jedes Quarto-Dokument beginnt mit einem YAML-Header, der von drei Bindestrichen (`---`) umschlossen ist. Dieser Header enthält Metadaten und Einstellungen für das Dokument. Lösche den Vorlageninhalt und beginne mit diesem minimalen Header:

```
---
```

```
title: "Pinguin-Report"
author: "Dein Name"
date: today
format: docx
---
```

Das bedeutet im Einzelnen:

- `title` , `author`: Erscheinen oben im gerenderten Dokument
- `date: today`: Fügt automatisch das aktuelle Datum beim Rendern ein
- `format: docx`: Weist Quarto an, ein Word-Dokument zu erzeugen

ℹ Hinweis

YAML ist empfindlich gegenüber Einrückung und Abständen. Es muss ein Leerzeichen nach jedem Doppelpunkt stehen, und Tabs und Leerzeichen dürfen nicht gemischt werden.

Markdown-Grundlagen

Unterhalb des YAML-Headers schreibt man den Inhalt mit Markdown — einer einfachen Formatierungssyntax. Hier sind die Grundlagen:

Überschriften

```
# Überschrift erster Ebene
## Überschrift zweiter Ebene
### Überschrift dritter Ebene
```

In Word werden diese als Überschrift 1, Überschrift 2 und Überschrift 3 formatiert.

Textformatierung

Das ist ****fetter**** Text und das ist *kursiver* Text.

Man kann auch Unterstriche für *Hervorhebung* verwenden.

Listen

- Erster Punkt
 - Zweiter Punkt
 - Dritter Punkt
1. Nummerierter Punkt
 2. Ein weiterer nummerierter Punkt

! Wichtig

In Quarto (und Markdown generell) braucht man eine **Leerzeile vor einer Liste**, damit sie korrekt gerendert wird. Das ist eine häufige Fehlerquelle für Anfänger.

Links

Besuche die [Quarto-Website] (<https://quarto.org>) für weitere Informationen.

Dein erster R-Code-Chunk

Jetzt kommt der spannende Teil — R-Code hinzufügen, der beim Rendern des Dokuments ausgeführt wird. Code-Chunks werden von dreifachen Backticks mit `{r}` umschlossen:

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

Beim Rendern werden sowohl der Code als auch seine Ausgabe im Word-Dokument angezeigt.

Schreiben wir etwas Sinnvollereres. Wir verwenden den Palmer Penguins Datensatz, der Messungen von Pinguinen aus drei Arten enthält. Zuerst laden wir die notwendigen Pakete und filtern die Daten, um nur Adelie-Pinguine einzuschließen:

```
library(tidyverse)
library(palmerpenguins)

adelie <- penguins %>%
```

```
filter(species == "Adelie") %>%
drop_na()
```

Nun können wir unsere Daten erkunden:

```
glimpse(adelie)
```

```
Rows: 146
Columns: 8
$ species      <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adel...
$ island        <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgers...
$ bill_length_mm <dbl> 39.1, 39.5, 40.3, 36.7, 39.3, 38.9, 39.2, 41.1, 38.6...
$ bill_depth_mm  <dbl> 18.7, 17.4, 18.0, 19.3, 20.6, 17.8, 19.6, 17.6, 21.2...
$ flipper_length_mm <int> 181, 186, 195, 193, 190, 181, 195, 182, 191, 198, 18...
$ body_mass_g    <int> 3750, 3800, 3250, 3450, 3650, 3625, 4675, 3200, 3800...
$ sex            <fct> male, female, female, female, male, female, male, fe...
$ year           <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007...
```

Wir haben 146 Adelie-Pinguine in unserem Datensatz mit Messungen von Schnabellänge, Schnabeltiefe, Flossenlänge und Körpermasse.

Inline-Code

Ist der letzte Satz aufgefallen? Er enthält die genaue Anzahl der Pinguine, aber diese Zahl wurde nicht manuell eingetippt. Stattdessen wurde **Inline-Code** verwendet — ein Backtick, gefolgt von `{r}`, einem Leerzeichen und dann einem R-Ausdruck:

146

Das `{r}` weist Quarto an, den R-Ausdruck auszuführen und das Ergebnis direkt in den Text einzufügen. Das ist unglaublich mächtig — wenn sich die Daten ändern, aktualisieren sich die Zahlen im Text automatisch.

Hier sind weitere Beispiele:

Man schreibt Man erhält

38.8 mm 38.8 mm

3706 g 3706 g

Das Dokument rendern

Um das Dokument zu Word zu rendern, gibt es mehrere Optionen:

1. **Auf den Render-Button klicken** in RStudio (der blaue Pfeil oben im Editor)
2. **Tastenkürzel:** `Ctrl+Shift+K` (Windows/Linux) oder `Cmd+Shift+K` (Mac)
3. **Kommandozeile:** `quarto render dein_dokument.qmd`

Quarto führt allen R-Code aus, kombiniert ihn mit dem Text und erzeugt eine `.docx`-Datei im selben Verzeichnis wie die `.qmd`-Datei.

💡 Tipp

Wenn man “Render on Save” in RStudio aktiviert (Checkbox neben dem Render-Button), wird das Dokument automatisch bei jedem Speichern neu gerendert. Das ist nützlich, um Änderungen während der Entwicklung schnell zu sehen.

Ein vollständiges Beispiel

Hier ist ein vollständiges minimales Dokument, das man kopieren und als Startpunkt verwenden kann:

```
---
title: "Pinguin-Report"
author: "Dein Name"
date: today
format: docx
---

# Einleitung

Dieser Report analysiert Daten aus dem Palmer Penguins Datensatz mit Fokus auf Adelie-Pinguine.

```{r}
#| label: setup
#| message: false
library(tidyverse)
library(palmerpenguins)

adelie <- penguins %>%
 filter(species == "Adelie") %>%
 drop_na()
```

# Datenübersicht

Unser Datensatz enthält 146 Adelie-Pinguine. Die mittlere Schnabellänge beträgt 38.8 mm.

```{r}
#| label: summary
summary(adelie$bill_length_mm)
```

# Fazit

Wir haben erfolgreich unser erstes Quarto-Dokument mit eingebettetem R-Code erstellt.
```

💡 Übung: Erstelle dein erstes Dokument

1. Erstelle ein neues Quarto-Dokument in RStudio
2. Ersetze den Inhalt durch das obige Beispiel
3. Rendere es zu Word
4. Öffne die Word-Datei und betrachte das Ergebnis
5. Versuche, die Art von “Adelie” zu “Gentoo” zu ändern und erneut zu rendern — beachte, wie sich alle Zahlen automatisch aktualisieren

Was kommt als Nächstes

Das gerenderte Word-Dokument funktioniert, sieht aber wahrscheinlich noch nicht sehr professionell aus. Die R-Ausgabe erscheint als Klartext, und die Formatierung ist generisch. In den folgenden Kapiteln lernen wir:

- Zu kontrollieren, welcher Code und welche Ausgabe im Dokument erscheint (Kapitel 3)
- Word-Vorlagen für professionelle Formatierung zu verwenden (Kapitel 4)
- Schöne Tabellen mit `flextable` zu erstellen (Kapitel 5)
- Gut formatierte Plots hinzuzufügen (Kapitel 6)

Bibliography
