

Reproduzierbare Wissenschaft mit Hilfe von Quarto

Johann Popp

2023-09-27

Im Sinne reproduzierbarer Wissenschaft ist es wünschenswert, die gesamte Datenanalyse vom Laden der Daten über die Datenaufbereitung, deskriptive Analyse, Modellbildung bis hin zur Beschreibung und Interpretation in einem Quelldokument zusammenzufassen, aus dem dann Berichte und Präsentationen erzeugt werden können. Quarto [1, <https://quarto.org>] hilft Ihnen dabei, indem es Freitext mit Programmcode aus R, Python, Julia und anderen Programmiersprachen zu einem Quelldokument verknüpft. Aus diesem Dokument werden dann Berichte und Präsentationen in verschiedenen Formaten (z.B. HTML, PDF, MS-Word) erzeugt.

Quarto ist das Nachfolgeprogramm von R Markdown [2]. Ich demonstriere es hier in Verknüpfung mit R [3] und RStudio [4]. Es funktioniert aber auch unabhängig davon z.B. mit Jupyter Notebooks [5].

1 Beispiel

Sehen Sie sich zum Beispiel einmal das Quelldokument aus Abbildung 1 an und das HTML-Dokument (Abbildung 2), das daraus generiert wurde. Dieses Handout wird Sie Schritt für Schritt durch die Erstellung dieses Beispieldokuments leiten und dabei wesentliche Quarto-Methoden erklären.

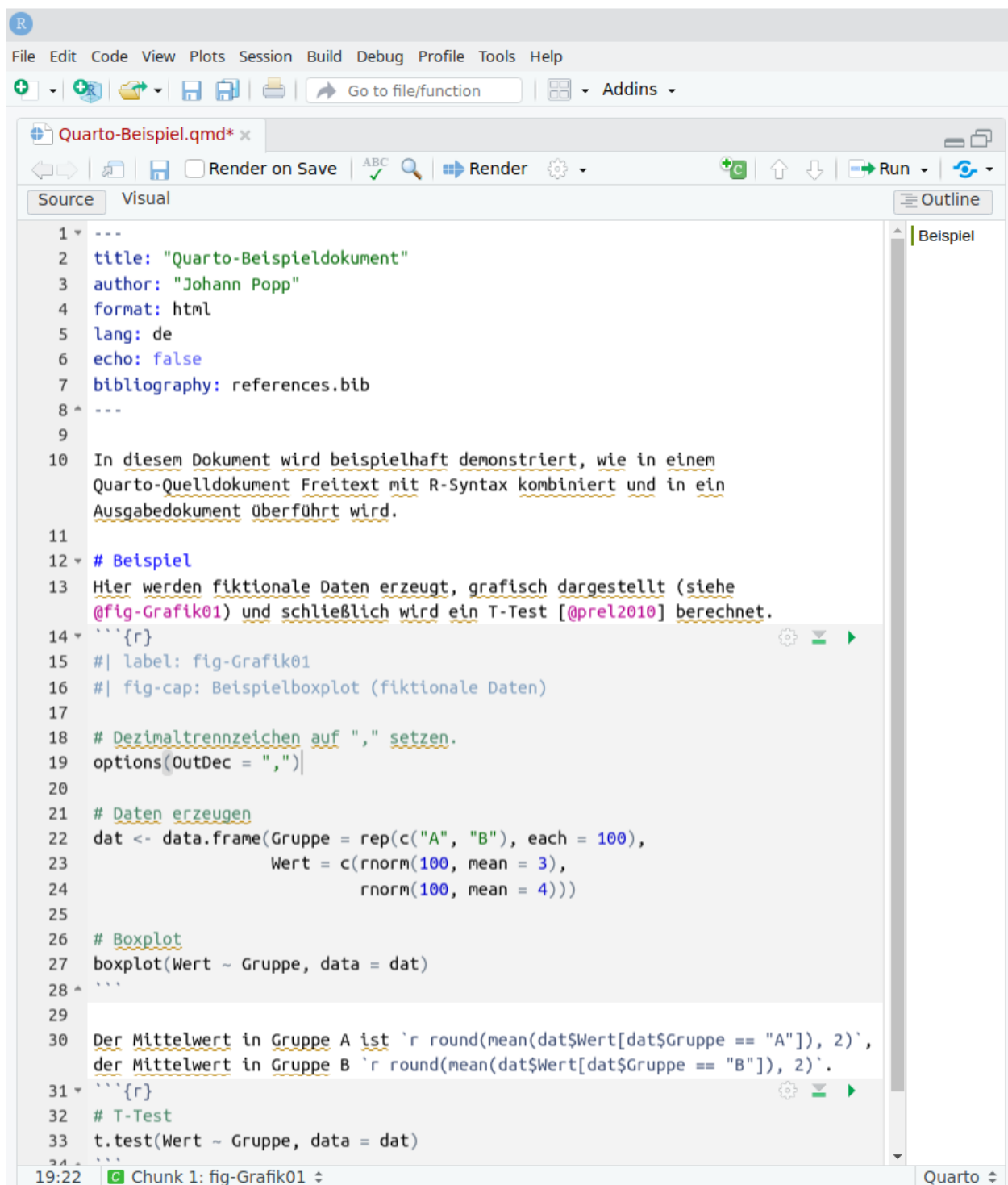


Abbildung 1: Beispiel für ein Quarto-Quelldokument

Quarto-Beispieldokument

AUTOR:IN
Johann Popp

In diesem Dokument wird beispielhaft demonstriert, wie in einem Quarto-Quelldokument Freitext mit R-Syntax kombiniert und in ein Ausgabedokument überführt wird.

Beispiel

Hier werden fiktionale Daten erzeugt, grafisch dargestellt (siehe [Abbildung 1](#)) und schließlich wird ein T-Test ([Prel u. a. 2010](#)) berechnet.

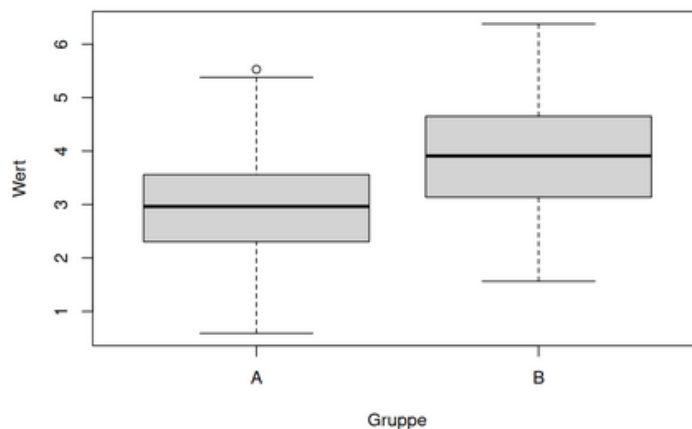


Abbildung 1: Beispielboxplot (fiktionale Daten)

Der Mittelwert in Gruppe A ist 2,97, der Mittelwert in Gruppe B 3,95.

```
Welch Two Sample t-test

data: Wert by Gruppe
t = -6,9999, df = 196,48, p-value = 3,954e-11
alternative hypothesis: true difference in means between group A and group B is not equal
95 percent confidence interval:
 -1,2605332 -0,7063826
sample estimates:
mean in group A mean in group B
  2,970445      3,953903
```

Literatur

Prel, Jean-Baptist du, Bernd Röhrig, Gerhard Hommel, und Maria Blettner. 2010. „Auswahl statistischer Testverfahren: Teil 12 der Serie zur Bewertung wissenschaftlicher Publikationen“. *Deutsches Ärzteblatt International* 107 (19): 343–48. <https://doi.org/10.3238/arztebl.2010.0343>.

Abbildung 2: Beispiel für eine HTML-Ausgabe eines Quarto-Dokuments

1.1 Erstellen eines Quarto-Quelltext-Dokumentes

Ein Quarto-Quelltext-Dokument ist erst einmal nichts anderes als eine Textdatei mit der Dateinamensendung `.qmd`. Wenn Sie dieses Dokument aus RStudio heraus erstellen, können Sie auf eine Reihe von Arbeitshilfen zurückgreifen, nicht zuletzt wurde Quarto auch automatisch zusammen mit RStudio installiert.

Um jetzt also ein Quarto-Dokument zu erstellen, klicken Sie links oben  und dann  Quarto Document..., um zu folgendem Dialogfeld zu gelangen:

New - New File - Quarto Document...

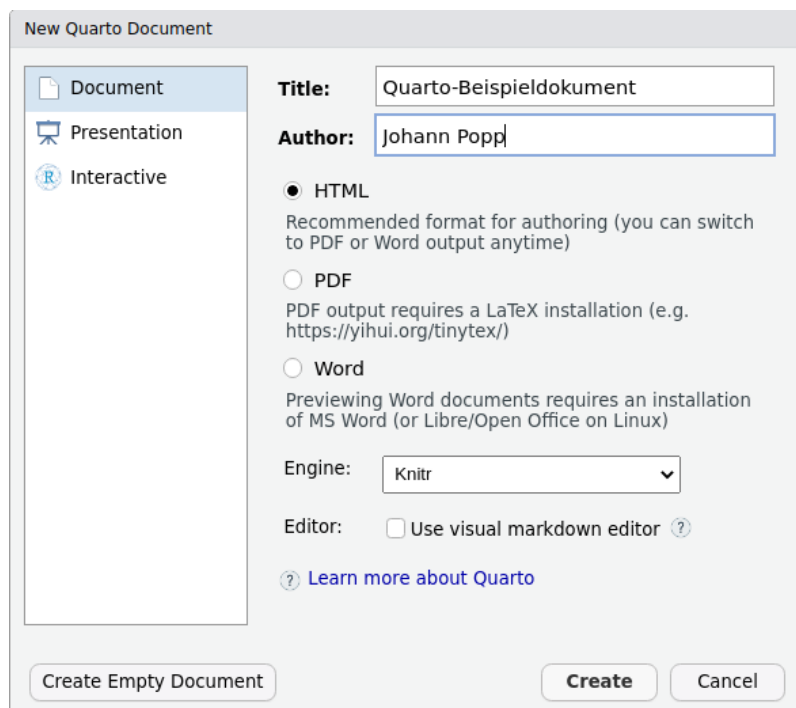


Abbildung 3: Dialogfeld zur Erstellung eines Quarto-Dokuments.

In der linken Spalte dieses Dialogfeldes können Sie zwischen Vorlagen für Dokumente, Präsentationen oder interaktive Dokumente wählen. Bleiben Sie hier erst einmal bei *Document*.

Rechts können Sie Titel und Autor/in des Dokuments angeben. Sie können außerdem zwischen den Ausgabeformaten *HTML*, *PDF* oder *Word* auswählen. Bleiben Sie bei der Vorgabe *HTML*. Die “Engine” können Sie gut bei *Knitr* belassen. Bei

“Editor” wählen Sie aber bitte fürs erste “Use visual markdown editor” ab.

Klicken Sie jetzt *Create Empty Document*.

Jetzt wird ein Quarto-Quelltextdokument geöffnet, das so wie in Abbildung 4 aussehen sollte.

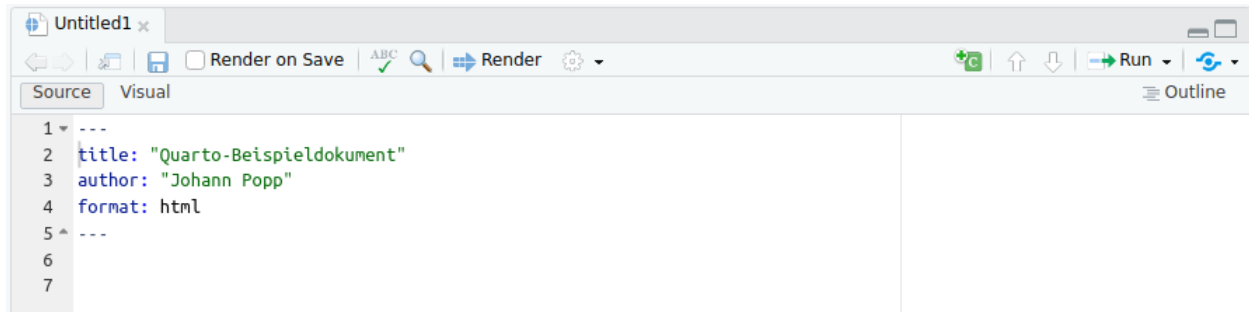


Abbildung 4: Ein leeres Quarto-Quelldokument

Sieht es bei Ihnen anders aus? Vielleicht haben Sie mehr als fünf Zeilen Quelltext? Dann haben Sie aus Versehen *Create* statt *Create Empty Document* geklickt. Das ist nicht schlimm. Löschen Sie dann einfach alles, ab Zeile 6.

Was Sie hier in den ersten fünf Zeilen sehen ist der sogenannte YAML-Header. Hier werden grundsätzliche Informationen angegeben, die das gesamte Dokument betreffen. Im Moment haben Sie den Titel des Dokumentes, den Autor oder die Autorin und das Ausgabeformat (html) angegeben. Wir werden später noch einmal auf den YAML-header zurück kommen.

Geben Sie jetzt die ersten Absätze Freitext ein, wie in Abbildung 5 .

Freitext eingeben

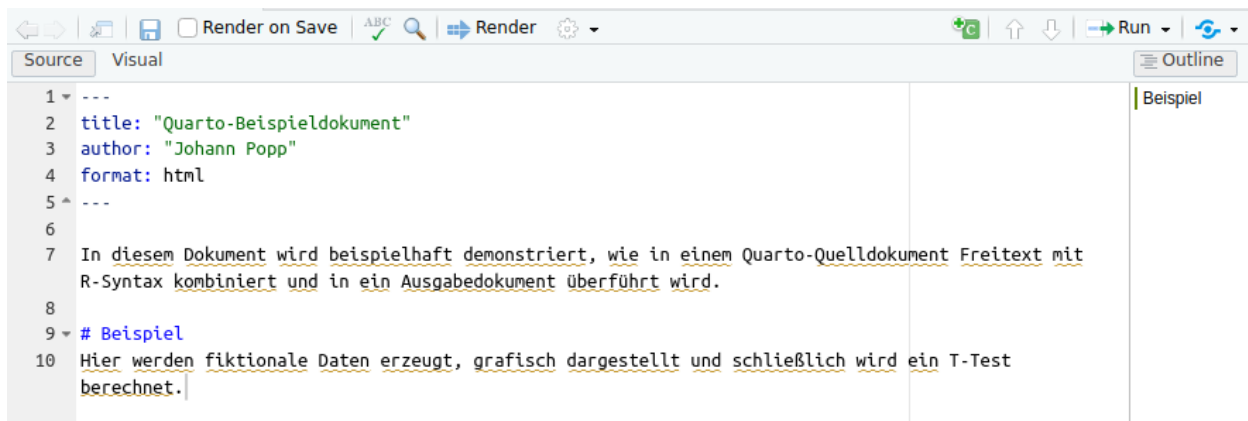



Abbildung 5: Quarto-Quelltext mit Freitext

Klicken Sie Jetzt auf  **Render**, um aus dem Quelldokument ein HTML-Ausgabedokument zu generieren. Das Ergebniss sollte etwa so aussehen wie in Abbildung 6.

Ausgabedokument mit *Render* erzeugen

Quarto-Beispieldokument


AUTHOR
Johann Popp

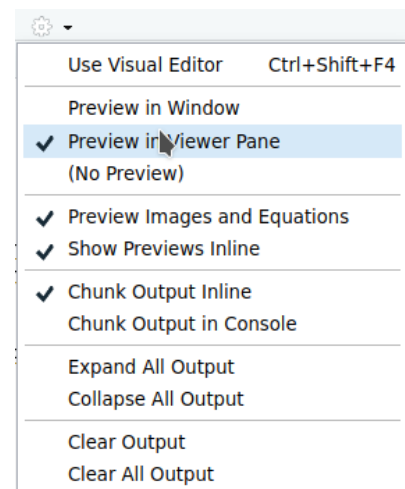
In diesem Dokument wird beispielhaft demonstriert, wie in einem Quarto-Quelldokument Freitext mit R-Syntax kombiniert und in ein Ausgabedokument überführt wird.

Beispiel

Hier werden fiktionale Daten erzeugt, grafisch dargestellt und schließlich wird ein T-Test berechnet.

Abbildung 6: Erste HTML-Ausgabe eines Quarto-Dokuments.

Wahrscheinlich wurde das Ausgabedokument bei Ihnen nicht innerhalb von RStudio angezeigt, sondern in Ihrem Internet-Browser. Für das Erstellen des Dokuments ist es praktischer, wenn Sie die Ausgabe direkt neben dem Quelldokument in RStudio sehen können. Wählen Sie dafür in den Einstellungen *Preview in Viewer Pane* aus (und klicken Sie noch einmal  **Render**).



Das Ausgabedokument zeigt zuerst Titel und Autor aus dem YAML-header. Dann folgt der Text, den Sie eingegeben haben. Auffällig ist, dass die Überschrift “Beispiel” im Quelldokument `# Beispiel` geschrieben wurde. Formatierungen werden nämlich im Quelldokument durch bestimmte Zeichen gekennzeichnet und nicht unmittelbar im Ergebnis sichtbar. (Überschriften der zweiten Gliederungsebene werden z.B. durch zwei Rauten gekennzeichnet, *kursive Schrift* durch ein Sternchen zu Beginn und zum Ende). Dieses Kodierungsschema nennt sich *Markdown* [6].

Markdown-Format

Sie brauchen sich nicht alle Markdown-Kodierungen merken, denn in Quarto können die Quelltexte auch im *Visual*-Modus ansehen. Klicken Sie links oben doch einmal auf **Visual**.

Visual-Modus

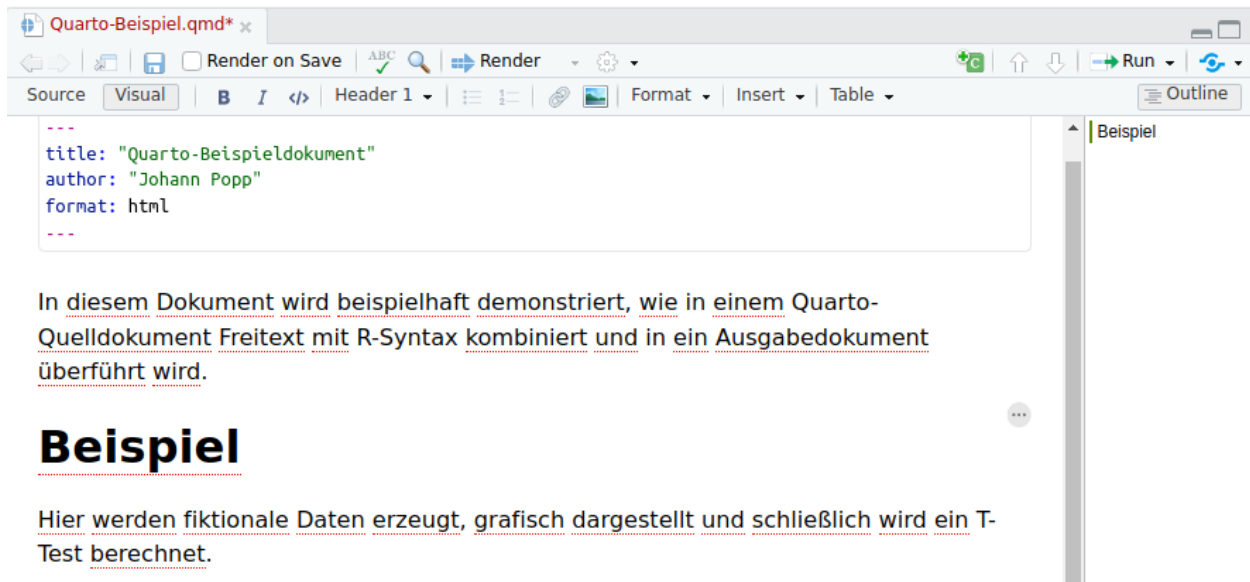



Abbildung 7: Quarto-Quelldokument im *Visual*-Modus

In diesem Modus sehen die Textformatierungen so aus, wie im Ausgabedokument. Außerdem erscheint eine Symbolleiste, in der Sie per Mausklick Textformatierungen einstellen, Aufzählungen und Nummerierungen generieren und weitere Elemente einfügen können.



Ich empfehle Ihnen, immer wieder zwischen dem *Source*-Modus und dem *Visual*-Modus hin und her zu schalten. Manche Dinge funktionieren hier einfacher, andere dort.

1.2 R-Syntax einfügen


Fügen Sie jetzt R-Syntax hinzu, indem Sie rechts oben  klicken. Es erscheint ein graues Feld, beginnend mit `{r}`, in das Sie R-Funktionen wie in einem gewöhnlichen R-Skript eingeben können. Das nennt sich *R-Chunk*. Erstellen Sie doch gleich mal die fiktionalen Daten, mit denen Sie rechnen wollen.

Insert - Executable Cell - R

```
```${r}

Daten erzeugen
data.frame(Gruppe = rep(c("A", "B"), each = 100),
 Wert = c(rnorm(100, mean = 3), rnorm(100, mean = 4)))
```
```

Abbildung 8: Erster R-Chunk innerhalb eines Quarto-Dokuments

Es wird eine Datenmatrix erstellt, mit einer Variable “Gruppe”, die 100 mal “A” enthält und 100 mal “B”, sowie einer Variable “Wert”, die 100 normalverteilte Zufallszahlen um den Mittelwert 3 enthält und 100 um den Mittelwert 4. Um die Funktion auszuführen, klicken Sie wie in normalen R-Skripten rechts oben auf  oder auf der Tastatur *Strg* - *Enter*.

Ordnen Sie die Daten einem Objekt zu, da Sie sie nicht einfach nur einmal ausgeben lassen, sondern mit ihnen weiter arbeiten wollen.

```
```${r}

Daten erzeugen
dat <- data.frame(Gruppe = rep(c("A", "B"), each = 100),
 Wert = c(rnorm(100, mean = 3),
 rnorm(100, mean = 4)))
```
```


Daten erzeugen und dem Objekt `dat` zuweisen

Mit diesen Daten können Sie nun einen Boxplot erzeugen. (Er sieht vielleicht ein wenig anders aus als in diesem Handout, weil er auf Zufallszahlen basiert.)

Einen Boxplot hinzufügen

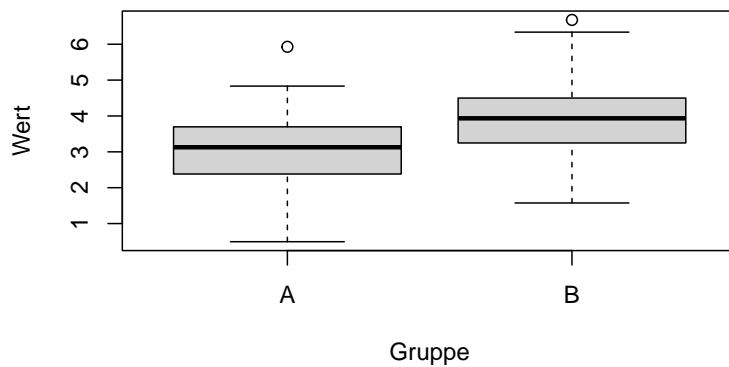
```
# Daten erzeugen
dat <- data.frame(Gruppe = rep(c("A", "B"), each = 100),
                  Wert = c(rnorm(100, mean = 3),
                           rnorm(100, mean = 4)))

# Boxplot
boxplot(Wert ~ Gruppe, data = dat)
```

Schauen Sie sich jetzt mit  **Render** an, wie das ganze im Ausgabedokument aussieht:

```
# Daten erzeugen
dat <- data.frame(Gruppe = rep(c("A", "B"), each = 100),
                  Wert = c(rnorm(100, mean = 3),
                           rnorm(100, mean = 4)))

# Boxplot
boxplot(Wert ~ Gruppe, data = dat)
```



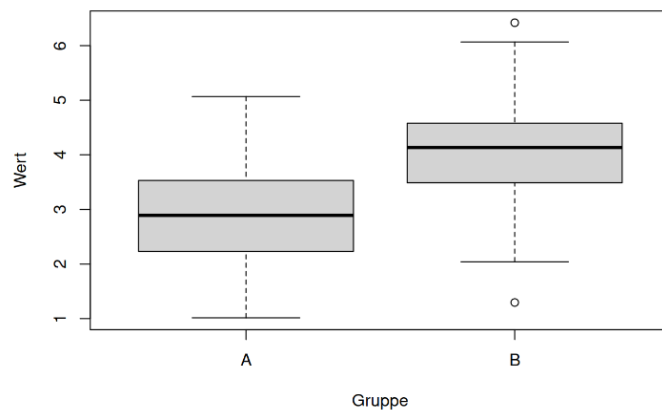
1.3 Beschriftungen

Natürlich soll Ihre Abbildung auch eine Beschriftung erhalten. Sie wird der Option `#| fig-cap:` eingefügt. Solche Optionen - die grundsätzlich mit `#|` beginnen - werden immer in die ersten Zeilen des R-Chunks eingegeben

```
```{r}
#| fig-cap: Beispielboxplot (fiktive Daten)|

Boxplot
boxplot(Wert ~ Gruppe, data = dat)
```
```

`#| fig-cap:`



Beispielboxplot (fiktionale Daten)

Die Beschriftung wurde schon einmal erzeugt. Um sie auch noch zu nummerieren, muss den R-Chunkt auch noch ein `#| label:` hinzugefügt werden, das mit `fig-` beginnt:

```
```{r}
#| label: fig-Grafik01|
#| fig-cap: Beispielboxplot (fiktive Daten)

Boxplot
boxplot(Wert ~ Gruppe, data = dat)
```
```

Automatische Nummerierung mit `#| label: fig-...`

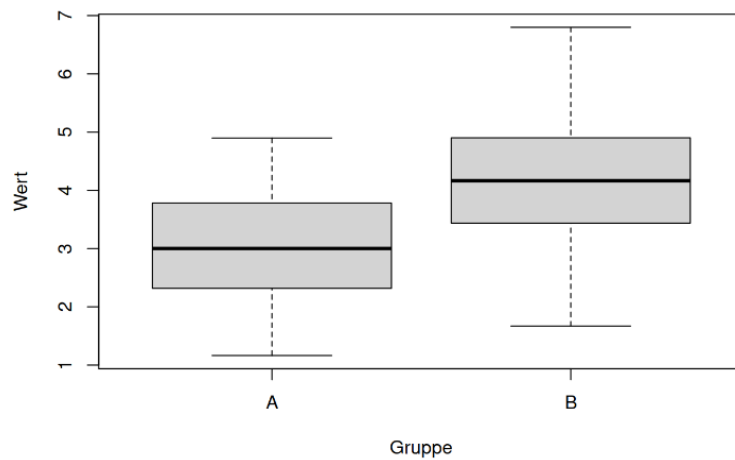


Figure 1: Beispielboxplot (fiktionale Daten)

Nachdem die automatische Nummerierung funktioniert, sollte die Bezeichnung nun aber auch in deutscher Sprache sein, also “Abbildung” statt “Figure”. Das können Sie erreichen, indem Sie die Spracheinstellungen des gesamten Dokuments auf Deutsch setzten. Gehen Sie zurück zum YAML-header und ergänzen Sie `lang: de`.

```

1 ---
2 title: "Quarto-Beispieldokument"
3 author: "Johann Popp"
4 format: html
5 lang: de
6 ---

```

Mit `lang: de` deutsche
Spracheinstellung im YAML-header
festlegen

Jetzt sollte der Boxplot so aussehen:

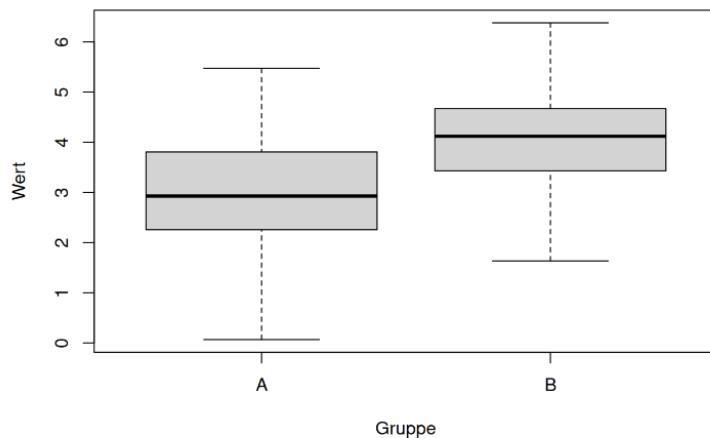


Abbildung 1: Beispielboxplot (fiktionale Daten)

Wenn Sie in Ihrem Ausgabedokument die R-Syntax ausblenden wollen, können Sie im YAML-header die Option `echo: false` ergänzen, um sie für das gesamte Dokument auszublenden. Sie können das aber auch nur für einzelne R-Chunks festlegen, wenn sie dort `#| echo: false` einfügen.

`echo: false` blendet die R-Syntax aus.

1.4 Querverweise

Um in Ihren Text Querverweise auf eine Grafik einzufügen, tippen Sie einfach `@` in Verbindung mit dem Label, dass Sie der Abbildung zugewiesen haben.

Hier werden fiktionale Daten erzeugt, grafisch dargestellt (siehe @fig-Grafik01) und schließlich wird ein T-Test berechnet.

`@fig-...`

```
```{r}
#| label: fig-Grafik01
#| fig-cap: Beispielboxplot (fiktionale Daten)
```

Hier werden fiktionale Daten erzeugt, grafisch dargestellt (siehe [Abbildung 1](#)) und schließlich wird ein T-Test berechnet.

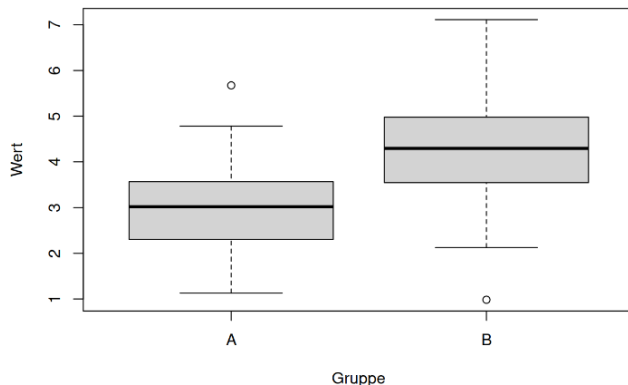


Abbildung 1: Beispielboxplot (fiktionale Daten)

## 1.5 Zitationen

Um Quellenangaben zu machen, wechseln Sie in den *Visual-*Modus und klicken Sie *Insert - @ Citation....*

*Insert - @ Citation...*

In dem jetzt erscheinenden Dialogfeld können Sie Zitationsangaben aus verschiedenen Datenbanken abrufen. Wenn Sie es installiert haben, greift Quarto direkt auf Ihre Zotero-Datenbank [7] zurück. Sie können aber z.B. auch direkt in PubMed nach Zitationen suchen.

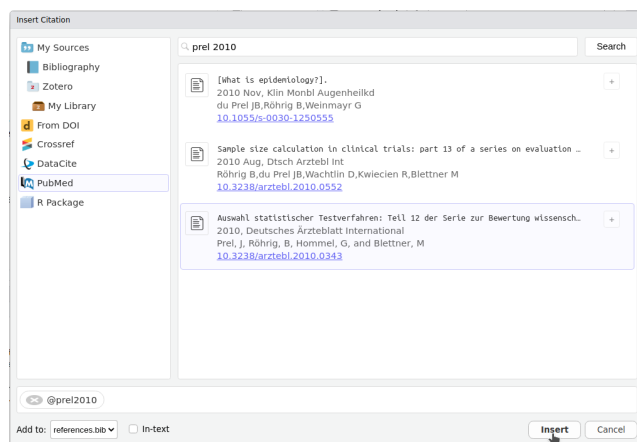



Abbildung 9: Dialogfeld zum Einfügen von Zitationen.

Um das Beispiel nachzuvollziehen, wählen Sie PubMed, suchen Sie nach “prel 2010” und wählen Sie den Artikel zur Auswahl statistischer Tests aus<sup>1</sup>. Klicken Sie dann auf *Insert*.

<sup>1</sup> So praktisch diese schnelle Online-Suche ist. Sie ersetzt keine gründliche Literaturrecherche!

Quarto schreibt jetzt einen Querverweis (hier `[ \@prel2010]`) ins Quelldokument. Gleichzeitig wird eine Textdatei “references.bib” in Ihrem Projektordner erzeugt, die die Bibliografischen Angaben enthält und im YAML-header wird die Zeile `bibliography: references.bib` ergänzt, um Quarto beim Erstellen der Ausgabedatei mitzuteilen, dass die bibliografischen Informationen dort zu finden sind.

Probieren Sie jetzt noch einmal  und schauen sich das Ergebnis an. Die Zitation sollte an der entsprechenden Stelle im Text eingefügt sein und am Ende des Dokuments befindet sich jetzt ein Literaturverzeichnis.

Wenn Sie eine Seitenangabe ergänzen möchten, schreiben Sie sie einfach innerhalb der eckigen Klammern hinter den Querverweis: `[@prel2010, S. 346]`. Wenn Sie die Auto/innen im Kurzverweis weg lassen wollen, entfernen Sie die eckigen Klammern: `@prel2010`.

Zusatz: `[@label, Zusatz]`

Nur Jahr: `@label`

Unter <https://www.zotero.org/styles> können Sie auch andere Zitationsstile im csl-Format herunterladen. Speichern Sie die entsprechende csl-Datei in Ihrem Projektordner und ergänzen Sie den YAML-header um `csl: Name_der_Datei.csl`.

Andere Zitationsstile verwenden

## 1.6 T-Test

Um endlich den T-Test zu berechnen, fügen Sie einen weiteren R-Chunk ein:

```
```{r}
# T-Test
t.test(Wert ~ Gruppe, data = dat)
```
```

Ausgabe des T-Tests

Welch Two Sample t-test

```
data: Wert by Gruppe
t = -6,4424, df = 196,39, p-value = 8,87e-10
alternative hypothesis: true difference in means between group A and group B is not equal to 0
95 percent confidence interval:
 -1,1728150 -0,6230693
sample estimates:
mean in group A mean in group B
 3,014664 3,912606
```

## 1.7 R-Ausgaben im Fließtext

Die Ausgabe des T-Tests wurde jetzt genauso dargestellt, wie sie auch in R ausgegeben wird. Es ist aber auch möglich, R-Ausgaben direkt in den Freitext zu integrieren. Rahmen Sie die R-Funktion dafür nach dem Muster ``r funktion()'`<sup>2</sup> ein, z.B. die Mittelwerte der beiden Gruppen:

```
Der Mittelwert in Gruppe A ist `r mean(dat$Wert[dat$Gruppe == "A"])`,
der Mittelwert in Gruppe B `r mean(dat$Wert[dat$Gruppe == "B"])`.
```

In der Ausgabe sieht das dann so aus:

Abbildung 1.1: Beispieloutput (inklusive Daten)

```
Der Mittelwert in Gruppe A ist 3.0348677, der Mittelwert in Gruppe B
4.1875778.
```

Zwei Probleme gibt es bei dieser Ausgabe noch. Erstens werden viel zu viele Nachkommastellen angegeben. Die lassen sich leicht mit der R-Funktion `round()` reduzieren.

```
Der Mittelwert in Gruppe A ist `r round(mean(dat$Wert[dat$Gruppe == "A"]),
2)`, der Mittelwert in Gruppe B `r round(mean(dat$Wert[dat$Gruppe ==
"B"]), 2)`.
```

Zweitens wird der Dezimaltrenner als Punkt angezeigt. Das ist die Standardeinstellung von R, aber Sie können es ändern. Fügen Sie in den ersten R-Chunk `options(OutDec = ",")` ein.

```
Dezimaltrennzeichen auf "," setzen.
options(OutDec = ",")
```

``r funktion()'`

<sup>2</sup> Das Zeichen zum Einfügen von R-Ausgaben in den Fließtext ist kein einfaches Anführungszeichen (Hochstell - Raute) sondern ein Akzent (Taste rechts neben dem ß, hochgestellt).

Jetzt haben Sie alle Elemente beisammen, um das Beispieldokument zu reproduzieren und damit auch fast alle Aspekte, die für die Erstellung wissenschaftlicher Dokumente wichtig sind. Vergleichen Sie bitte Ihr Quelldokument und Ihr Ausgabedokument noch einmal mit [Abbildung 1](#) und [Abbildung 2](#).

## 2 Tabellen

Ein weiterer wichtiger Aspekt bei der Erstellung wissenschaftlicher Dokumente sind Tabellen. Es gibt in Quarto verschiedene Möglichkeiten dafür. Bei einfachen Tabellen eignet sich die Funktion `kable()` aus dem vorinstallierten Paket *knitr* [8] sehr gut.

```
```{r}
#| label: tbl-Deskriptiv
#| tbl-cap: Deskriptive Statistiken der fiktiven Daten.

# Statistiken in einem data.frame bündeln.
desk <- data.frame(Gesamt = unclass(summary(dat$Wert)),
                   Gruppe.A = unclass(summary(dat$Wert[dat$Gruppe == "A"])),
                   Gruppe.B = unclass(summary(dat$Wert[dat$Gruppe == "B"])))

# Tabelle mit kable() erstellen.
knitr::kable(round(desk, 2))
```
```

Tabelle 1: Deskriptive Statistiken der fiktiven Daten.

|         | Gesamt | Gruppe.A | Gruppe.B |
|---------|--------|----------|----------|
| Min.    | 0,49   | 0,49     | 1,57     |
| 1st Qu. | 2,78   | 2,38     | 3,27     |
| Median  | 3,48   | 3,13     | 3,93     |
| Mean    | 3,46   | 3,01     | 3,91     |
| 3rd Qu. | 4,10   | 3,69     | 4,49     |
| Max.    | 6,68   | 5,93     | 6,68     |



Label und Beschriftung werden hier analog zu den Abbildungen mit dem Präfix `tbl-` angegeben, um festzulegen, dass es sich um Tabellen handelt.

Für kompliziertere Tabellen empfehle ich das Paket *flextable* [9], das Sie dann allerdings zusätzlich installieren müssen.

`flextable`

### 3 Noch ein paar YAML-Einstellungen

Wenn Sie nummerierte Überschriften haben möchten, ergänzen Sie im YAML-header `number-sections: true`.

`number-sections:`

Für ein Inhaltsverzeichnis ergänzen Sie `toc: true`.

`toc:`

Ein Abbildungsverzeichnis erhalten Sie mit `lof: true` und ein Verzeichnis der Tabellen mit `lot: true`.

`lof:` und `lot:`

Probieren Sie einmal `format: docx` oder `format: odt` aus, um ein MS-Word- oder LibreOffice-Dokument zu erstellen.

`format: docx` und `format: odt`

Wenn Sie mit `format: pdf` ein pdf-Dokument erstellen wollen, muss zusätzlich ein Programm zur Erstellung von LaTeX-Dokumenten installiert sein (siehe Kapitel 4).

`format: pdf`

Probieren Sie einmal ein Präsentationsformat aus mit `format: pptx`. `format: beamer` erstellt eine pdf-Präsentation und `format: revealjs` eine sehr flexible HTML-Präsentation.

Präsentationsformate `pptx`, `beamer` und `revealjs`

### 4 Ein kleiner Blick unter die Motorhaube

Wenn Sie ein Quarto-Quelldokument in ein Ausgabedokument umwandeln, werden mehrere Schritte durchlaufen:

- Das qmd-Dokument wird vom R-Paket *knitr* [8] in ein *Markdown*-Dokument [6] umgewandelt. Insbesondere ist das für die R-Chunks notwendig. Der Fließtext folgt bereits dem *Markdown*-Format.
- Das *Markdown*-Dokument (mit der Endung `.md`) wird von dem Programm *Pandoc* [10] weiter in ganz verschiedene Dokumente (z.B. im HTML-, docx-, oder pdf-Format) umgewandelt.

*knitr*, *Markdown* und *pandoc* werden mit *RStudio* automatisch mit installiert. Für die Umwandlung in ein pdf-Dokument muss zusätzlich eine LaTeX-Distribution installiert sein.

Falls Sie noch kein LaTeX-Programm auf Ihrem Computer haben, empfehle ich eine Installation über das R-Paket *tinytex* [11]. Dabei wird eine minimale Programminstallation vorgenommen und nur die wirklich notwendigen Zusatzpakete werden beim Erstellen der Ausgabedokumente automatisch nachinstalliert.

*tinytex*

```
Das R-Paket Tinytex installieren
install.packages("tinytex")

Das Programm Tex installieren
tinytex::install_tinytex()
```

Vom Prinzip her sollte sich das gleiche Quarto-Quelldokument problemlos in alle möglichen Ausgabeformate übersetzen lassen. Je ausgefeilter und detaillierter Sie allerdings das Layout beeinflussen, umso größer werden die Unterschiede in den Ausgaben und Sie müssen sich dann doch eher auf ein Format konzentrieren.

Viele Ausgabeformate aus dem gleichen Quelldokument?

Unter <https://quarto.org/> finden Sie eine umfangreiche Dokumentation von Quarto.

Dokumentation: <https://quarto.org/>

## Literatur

- [1] Allaire JJ, Teague C, Scheidegger C, u. a. *Quarto*. Epub ahead of print Januar 2022. DOI: [10.5281/zenodo.5960048](https://doi.org/10.5281/zenodo.5960048).
- [2] Xie Y, Allaire JJ, Golemund G. *R Markdown: the definitive guide*. Boca Raton: Taylor & Francis, CRC Press, <https://bookdown.org/yihui/rmarkdown> (2018).
- [3] R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, <https://www.R-project.org/> (2018).

- [4] RStudio Team. *RStudio: Integrated development environment for r*. Boston, MA, <http://www.rstudio.com/> (2021).
- [5] Kluyver T, Ragan-Kelley B, Pérez F, u. a. Jupyter Notebooks – a publishing format for reproducible computational workflows. In: Loizides F, Schmidt B (Hrsg). IOS Press, 2016, S. 87–90.
- [6] Gruber J. *Markdown*. The Daring Fireball Company LLC, <https://daringfireball.net/projects/markdown/> (2004).
- [7] Roy Rosenzweig Center for History and New Media. *Zotero*, [www.zotero.org/download](http://www.zotero.org/download) (2023).
- [8] Xie Y. *knitr: A General-Purpose Package for Dynamic Report Generation in R*, <https://yihui.name/knitr/> (2023).
- [9] Gohel D. *flextable: Functions for tabular reporting*, <https://CRAN.R-project.org/package=flextable> (2020).
- [10] Mac Farlane J. *Pandoc*, <https://pandoc.org/index.html> (2023).
- [11] Xie Y. *tinytex: Helper functions to install and maintain TeX live, and compile LaTeX documents*, <https://github.com/yihui/tinytex> (2021).