

6. Plots in Quarto

Grafiken optimal einbinden und formatieren

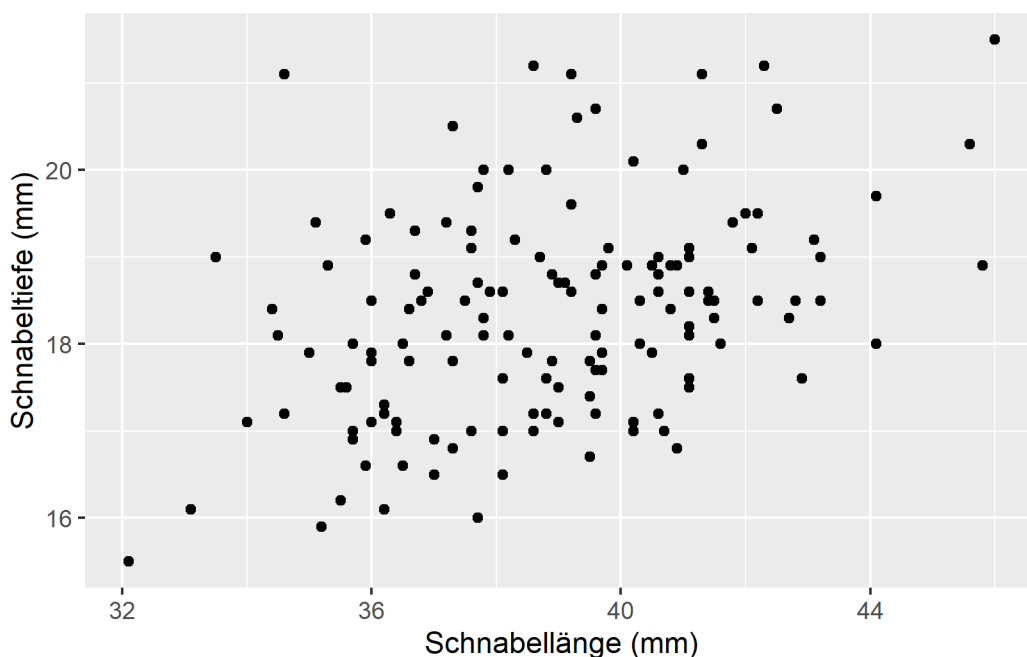
Dr. Paul Schmidt

ggplot2-Grafiken werden von Quarto automatisch in Word-Dokumente eingebunden. Aber ohne weitere Einstellungen sind sie oft zu klein, haben die falsche Auflösung oder unlesbare Beschriftungen. In diesem Kapitel lernt man, wie man Plots optimal für Word-Dokumente konfiguriert.

Ein einfacher Plot

Beginnen wir mit einem Scatterplot der Schnabelmaße:

```
ggplot(adelie, aes(x = bill_length_mm, y = bill_depth_mm)) +
  geom_point() +
  labs(
    x = "Schnabellänge (mm)",
    y = "Schnabeltiefe (mm)"
  )
```



Der Plot erscheint, aber die Größe und Proportionen sind möglicherweise nicht ideal für das Dokument.

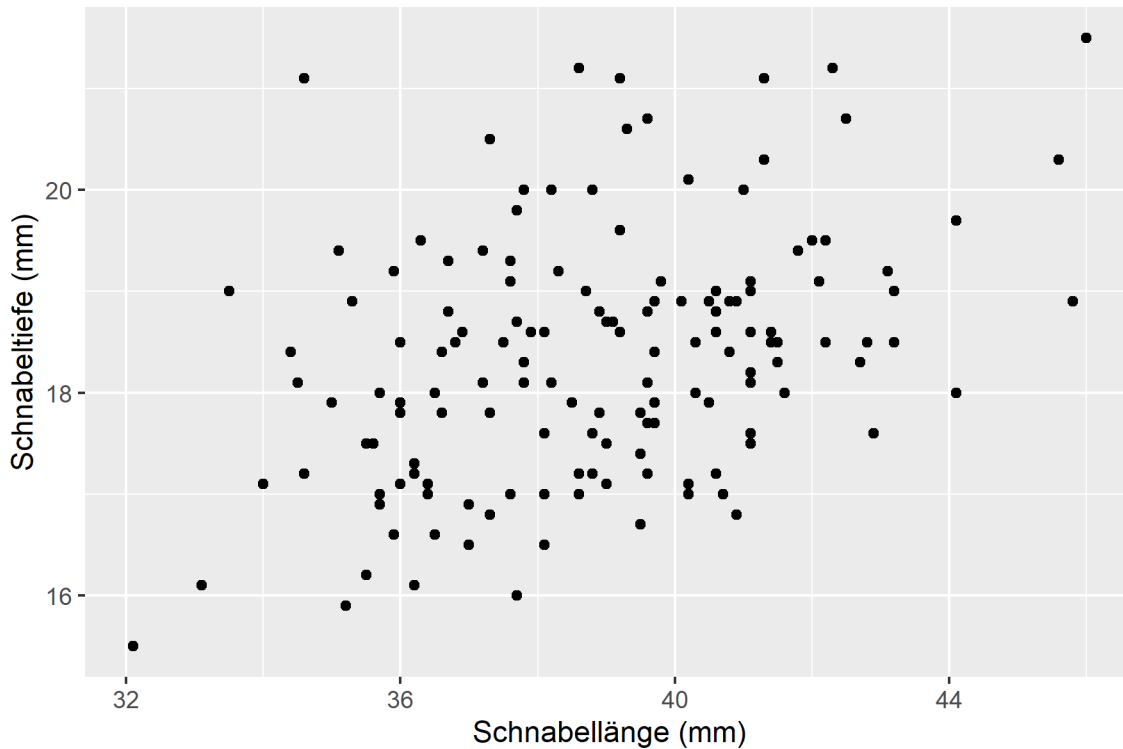
Größe kontrollieren

fig-width und fig-height

Die wichtigsten Chunk-Optionen für Plot-Größen sind `fig-width` und `fig-height` (in Zoll):

```
ggplot(adelie, aes(x = bill_length_mm, y = bill_depth_mm)) +
  geom_point() +
  labs(
```

```
x = "Schnabellänge (mm)",
y = "Schnabeltiefe (mm)"
)
```



💡 Tipp

Typische Größen für Word-Dokumente:

- Vollbreite: `fig-width: 6.5` (entspricht der Textbreite bei Standardrändern)
- Halbbreite: `fig-width: 3.25`
- Quadratisch: `fig-width: 4, fig-height: 4`

Seitenverhältnis

Für bestimmte Plot-Typen sind bestimmte Seitenverhältnisse besser geeignet:

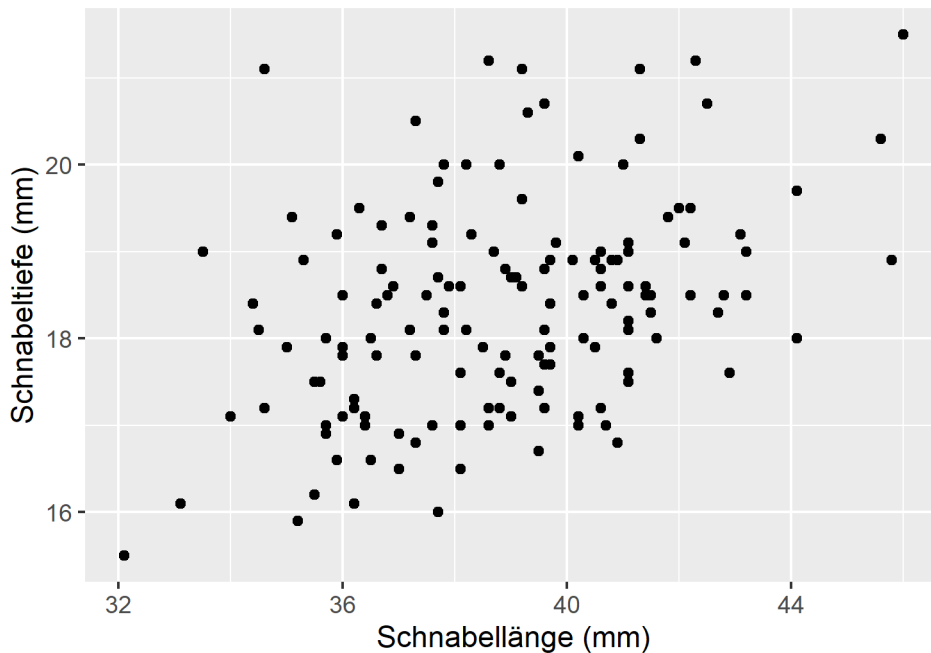
Plot-Typ	Empfohlenes Verhältnis
Scatterplot	4:3 oder 16:9
Balkendiagramm (horizontal)	breiter als hoch
Balkendiagramm (vertikal)	höher als breit
Zeitreihe	16:9 oder 2:1

Auflösung

fig-dpi

Die Auflösung wird mit `fig-dpi` gesteuert (dots per inch):

```
ggplot(adelie, aes(x = bill_length_mm, y = bill_depth_mm)) +
  geom_point() +
  labs(
    x = "Schnabellänge (mm)",
    y = "Schnabeltiefe (mm)"
  )
```



Empfehlungen:

- **96 dpi:** Bildschirm/Web (schnelles Rendering)
- **150 dpi:** Entwürfe
- **300 dpi:** Druckqualität (Standard für Publikationen)
- **600 dpi:** Hochwertige Drucke

i Hinweis

Höhere DPI bedeutet größere Dateien und längere Renderzeiten. Für Entwürfe kann man mit niedrigerer Auflösung arbeiten und erst für die finale Version auf 300 dpi erhöhen.

Bildunterschriften

fig-cap

Eine Bildunterschrift wird mit `fig-cap` hinzugefügt:

```
ggplot(adelie, aes(x = bill_length_mm, y = bill_depth_mm)) +
  geom_point(alpha = 0.6) +
  labs(
    x = "Schnabellänge (mm)",
    y = "Schnabeltiefe (mm)"
  ) +
  theme_minimal()
```

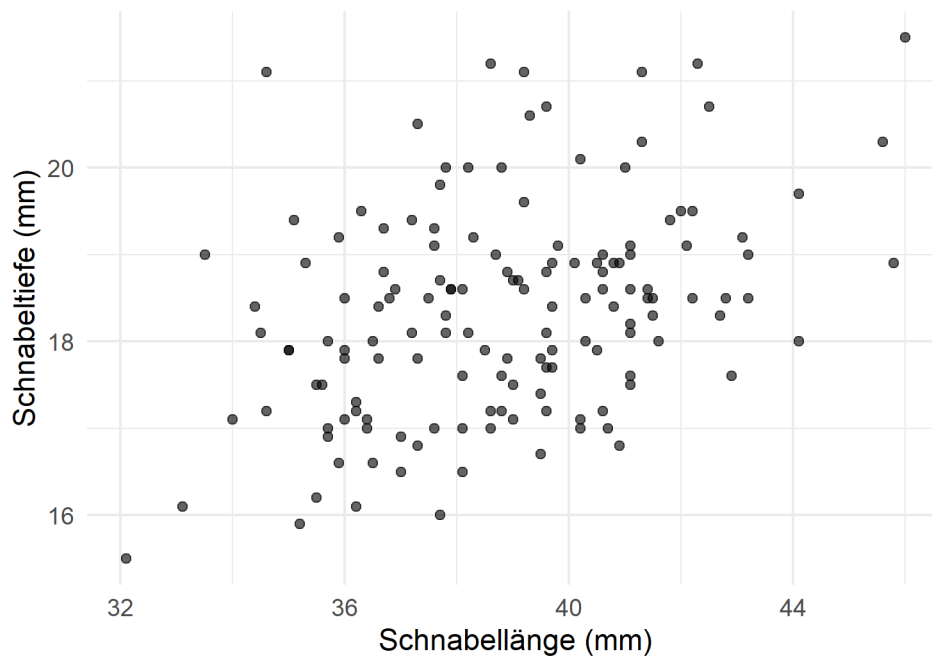


Abbildung 1: Zusammenhang zwischen Schnabellänge und Schnabeltiefe bei Adelie-Pinguinen.

! Wichtig

Für Cross-Referenzen (Kapitel 7) muss das Label mit `fig-` beginnen!

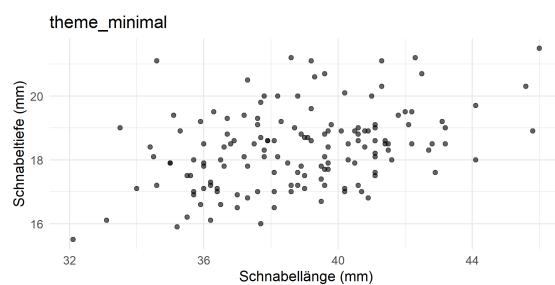
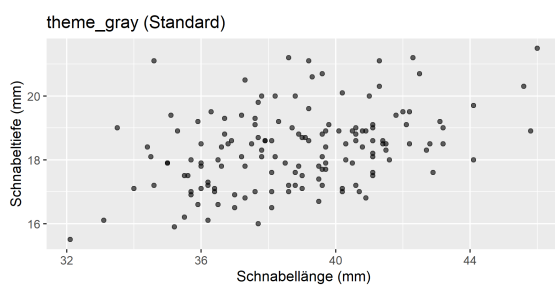
Themes für publikationsreife Plots

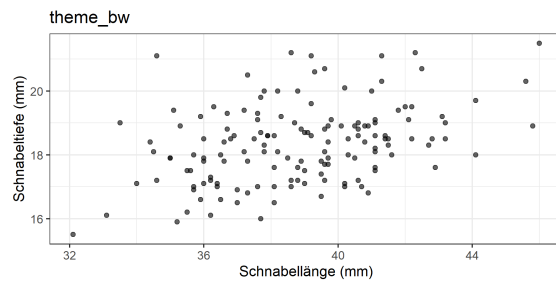
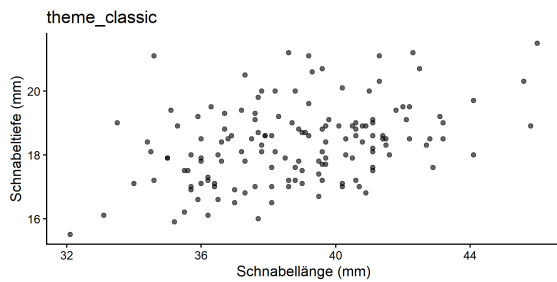
Standard-Themes

ggplot2 bietet mehrere eingebaute Themes:

```
p <- ggplot(adelie, aes(x = bill_length_mm, y = bill_depth_mm)) +
  geom_point(alpha = 0.6) +
  labs(x = "Schnabellänge (mm)", y = "Schnabeltiefe (mm)")

p + theme_gray() + ggtitle("theme_gray (Standard)")
p + theme_minimal() + ggtitle("theme_minimal")
p + theme_classic() + ggtitle("theme_classic")
p + theme_bw() + ggtitle("theme_bw")
```





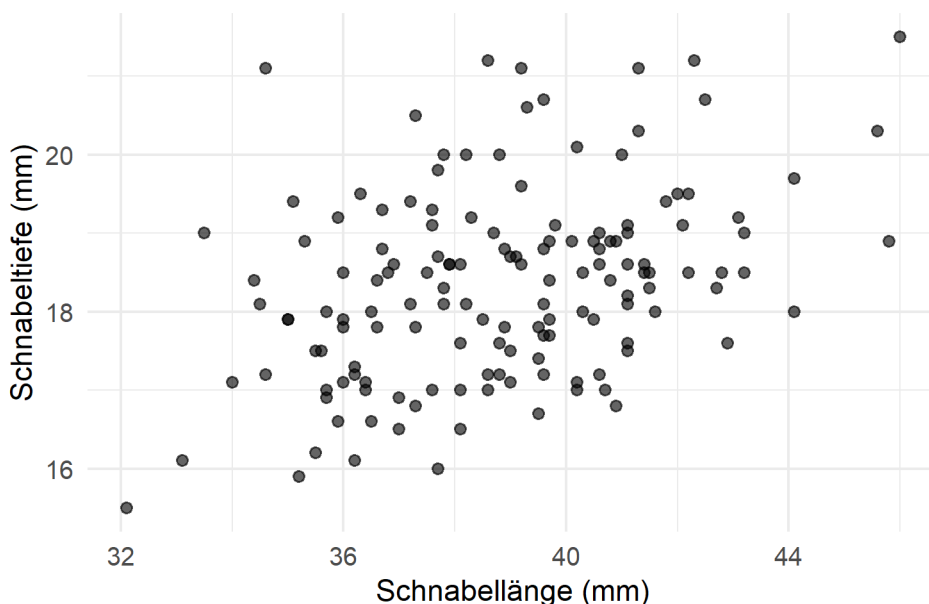
Für wissenschaftliche Publikationen sind `theme_minimal()`, `theme_classic()` oder `theme_bw()` am gebräuchlichsten.

Schriftgrößen anpassen

Ein häufiges Problem: Die Beschriftungen sind im gerenderten Dokument zu klein. Das lässt sich mit `theme()` beheben:

```
ggplot(adelie, aes(x = bill_length_mm, y = bill_depth_mm)) +
  geom_point(alpha = 0.6) +
  labs(
    x = "Schnabellänge (mm)",
    y = "Schnabeltiefe (mm)",
    title = "Schnabelmaße der Adelie-Pinguine"
  ) +
  theme_minimal(base_size = 12) +
  theme(
    axis.title = element_text(size = 11),
    plot.title = element_text(size = 13, face = "bold")
  )
```

Schnabelmaße der Adelie-Pinguine



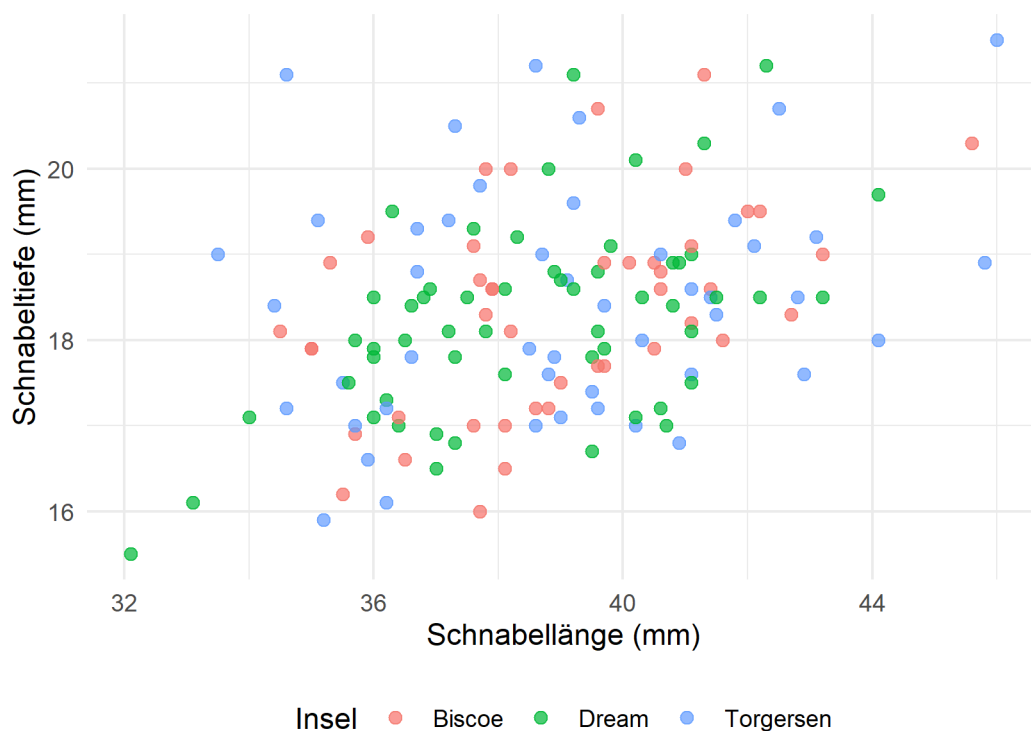
💡 Tipp

`base_size` in `theme_minimal(base_size = 12)` skaliert alle Textelemente proportional. Das ist oft einfacher als jedes Element einzeln anzupassen.

Farben

Nach Gruppen färben

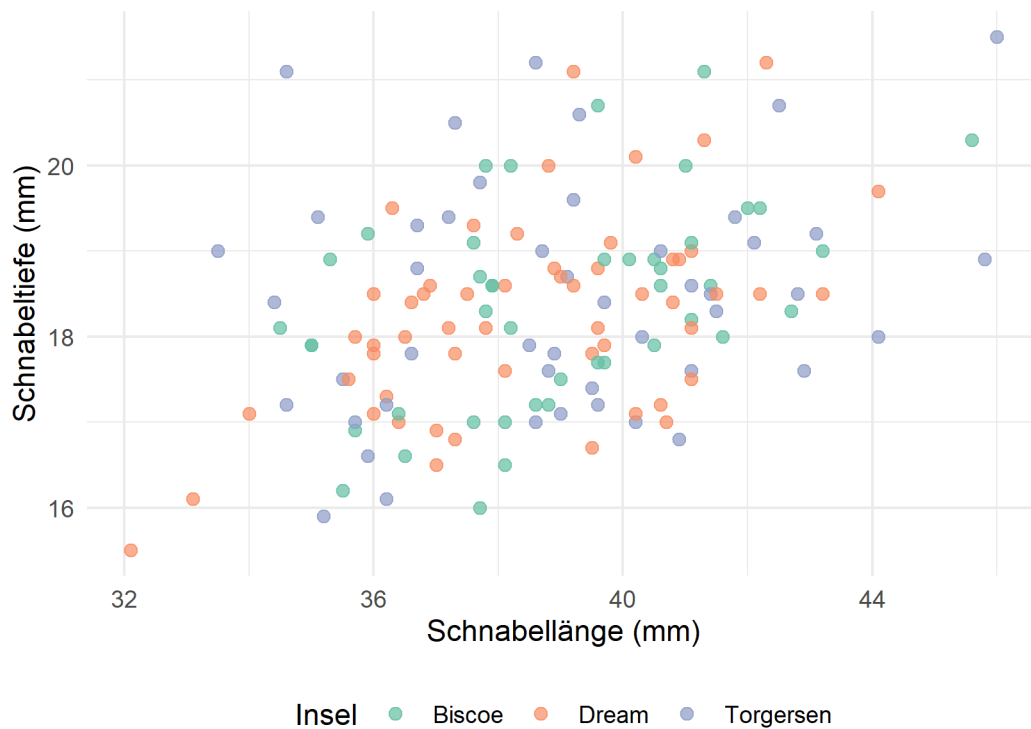
```
ggplot(adelie, aes(x = bill_length_mm, y = bill_depth_mm, color = island)) +
  geom_point(alpha = 0.7, size = 2) +
  labs(
    x = "Schnabellänge (mm)",
    y = "Schnabeltiefe (mm)",
    color = "Insel"
  ) +
  theme_minimal(base_size = 11) +
  theme(legend.position = "bottom")
```



Farbpaletten

Für wissenschaftliche Publikationen empfehle ich farbenblindfreundliche Paletten:

```
ggplot(adelie, aes(x = bill_length_mm, y = bill_depth_mm, color = island)) +
  geom_point(alpha = 0.7, size = 2) +
  scale_color_brewer(palette = "Set2") +
  labs(
    x = "Schnabellänge (mm)",
    y = "Schnabeltiefe (mm)",
    color = "Insel"
  ) +
  theme_minimal(base_size = 11) +
  theme(legend.position = "bottom")
```



Der Plot für unseren Report

Hier ist ein vollständiger, publikationsreifer Plot für unseren Pinguin-Report:

```
ggplot(adelie, aes(x = bill_length_mm, y = bill_depth_mm, color = island)) +
  geom_point(alpha = 0.7, size = 2.5) +
  geom_smooth(method = "lm", se = FALSE, linewidth = 0.8) +
  scale_color_brewer(palette = "Set2") +
  labs(
    x = "Schnabellänge (mm)",
    y = "Schnabeltiefe (mm)",
    color = "Insel"
  ) +
  theme_minimal(base_size = 11) +
  theme(
    legend.position = "bottom",
    panel.grid.minor = element_blank(),
    axis.title = element_text(size = 11),
    legend.title = element_text(size = 10),
    legend.text = element_text(size = 9)
  )
```

```
`geom_smooth()` using formula = 'y ~ x'
```

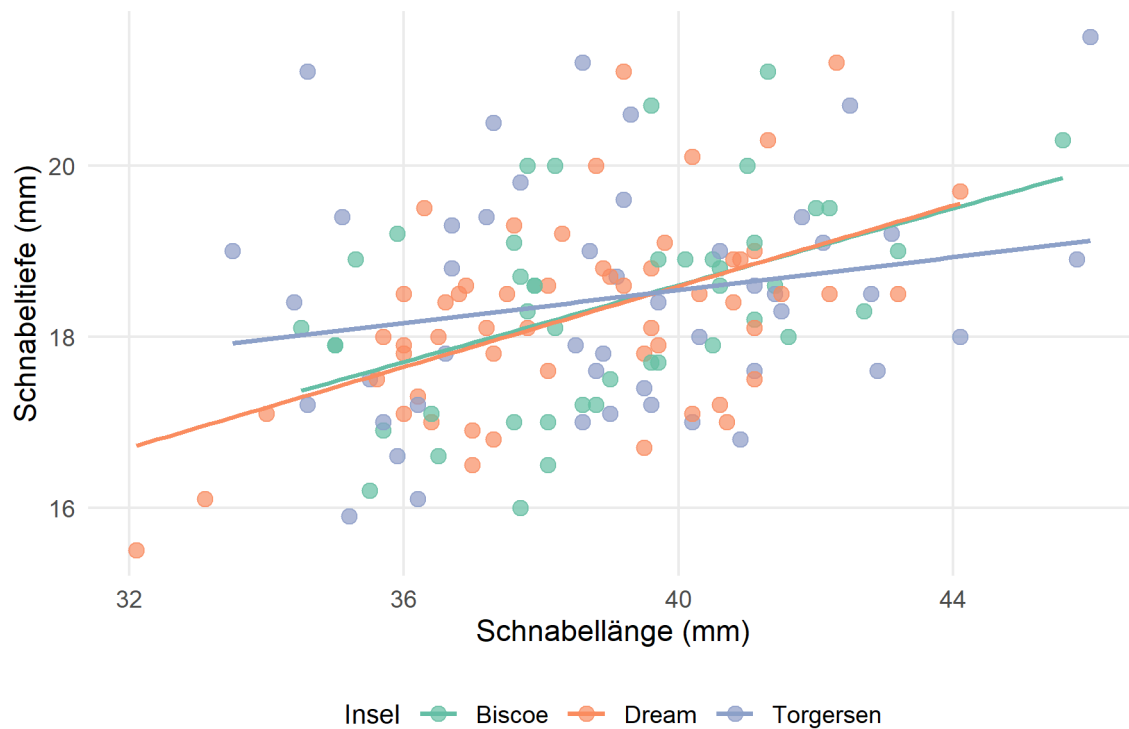


Abbildung 2: Zusammenhang zwischen Schnabellänge und Schnabeltiefe bei Adelie-Pinguinen, gruppiert nach Insel. Jeder Punkt repräsentiert ein Individuum.

Globale Plot-Einstellungen

Um nicht in jedem Chunk dieselben Optionen zu wiederholen, kann man globale Einstellungen im YAML-Header setzen:

```
---
title: "Mein Report"
format: docx
knitr:
  opts_chunk:
    fig-width: 6
    fig-height: 4
    fig-dpi: 300
---
```

Oder mit `execute:` für Quarto-spezifische Optionen:

```
execute:
  fig-width: 6
  fig-height: 4
```

Mehrere Plots

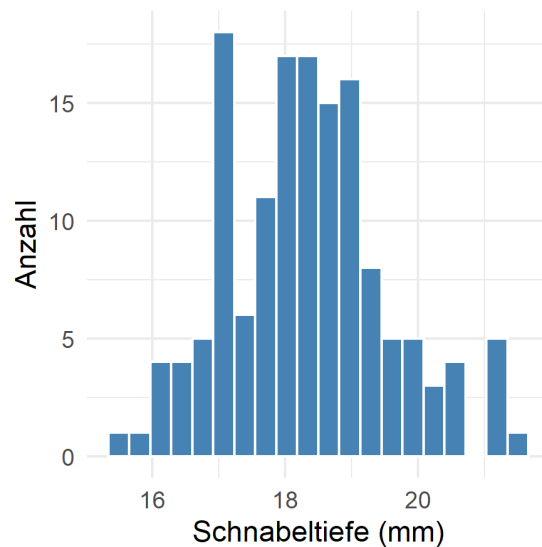
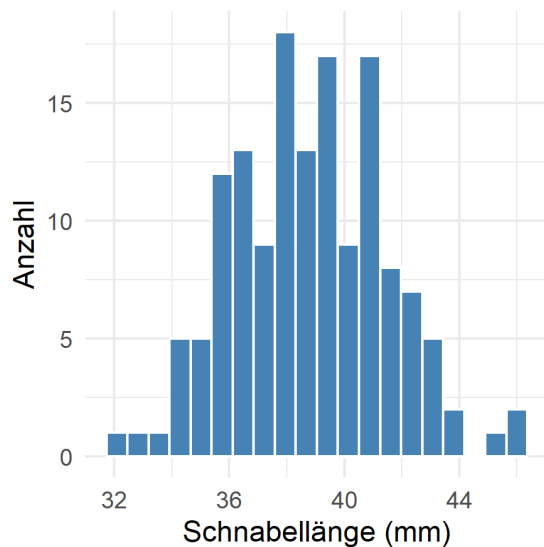
Layout in Quarto

Mit `layout-ncol` kann man mehrere Plots nebeneinander anordnen:

```
ggplot(adelie, aes(x = bill_length_mm)) +
  geom_histogram(bins = 20, fill = "steelblue", color = "white") +
  labs(x = "Schnabellänge (mm)", y = "Anzahl") +
  theme_minimal()
ggplot(adelie, aes(x = bill_depth_mm)) +
```



```
geom_histogram(bins = 20, fill = "steelblue", color = "white") +
labs(x = "Schnabeltiefe (mm)", y = "Anzahl") +
theme_minimal()
```



💡 Übung: Erstelle einen publikationsreifen Plot

1. Erstelle einen Boxplot der Körpermasse (`body_mass_g`) nach Insel
2. Setze passende Achsenbeschriftungen
3. Verwende `theme_minimal()` mit angepasster `base_size`
4. Füge eine Bildunterschrift mit `fig-cap` hinzu
5. Experimentiere mit verschiedenen `fig-width` und `fig-height` Werten

Weiterführende Ressourcen

- Quarto Figures — Offizielle Dokumentation
- ggplot2 book — Umfassende ggplot2-Referenz
- R Graphics Cookbook — Praktische Rezepte

Was kommt als Nächstes

Wir können jetzt Tabellen und Plots erstellen. In Kapitel 7 lernen wir, wie man mit Cross-Referenzen auf diese Elemente verweist — “wie in Abbildung 1 gezeigt” oder “siehe Tabelle 2”.

Bibliography