

# Übungseinheit 2 – Daten visualisieren in R

## Erste Schritte mit ggplot

### Übung 1: Vorbereitendes

- Erstellen Sie ein neues Verzeichnis mit dem Namen *GDSR\_2* und definieren Sie dieses Verzeichnis als Ihr Arbeitsverzeichnis für heute.
- Erstellen Sie ein neues R-Skript mit dem Namen *Aufgabenblatt\_2.R* und speichern Sie dieses ebenfalls in Ihrem Arbeitsverzeichnis ab.
- Ergänzen Sie in Ihrem R-Skript folgende erste Zeile (beachten Sie, dass Sie Kommentare mit # zu Ihrem Skript hinzufügen können):

```
rm(list=ls()) # Entfernt alle Objekte (Datensätze, Variablen usw.) aus dem Speicher
```

- Installieren Sie die Pakete *tidyverse* und *palmerpenguins* (falls möglich und noch nicht noch nicht vorhanden) via des Befehls `install.packages(Paketname)`.
- Laden Sie die beiden Pakete via des Befehls `library(Paketname)`.
- Betrachten Sie den nun über das Paket *palmerpenguins* geladenen Datensatz *penguins* mit dem Befehl `View(...)`.
- Informieren Sie sich über den Datensatz mit dem Befehl `?penguins`.
- Der Datensatz *penguins* enthält 11 Beobachtungen, in denen eine oder mehrere Informationen fehlen. Erzeugen Sie ein neues R-Objekt mit dem Namen *Pinguine*, in dem Sie alle vollständigen Beobachtungen des Datensatzes *penguins* ablegen. Nutzen Sie dazu den Befehl `na.omit(Datensatzname)`. Überlegen Sie, warum Ihnen die Daten erst jetzt in der Environment angezeigt werden?
- Nutzen Sie die Funktion `dim( )`, um die Anzahl an Beobachtungen und Variablen des Datensatzes *Pinguine* zu bestimmen.
- Speichern Sie Ihr R-Skript und führen Sie Ihr gesamtes Skript noch einmal aus und prüfen Sie, ob Ihr Vorgehen replizierbar ist.

### Übung 2: Analyse

- Sie interessieren sich für den Zusammenhang zwischen dem Gewicht eines Pinguins (*body\_mass\_g*) und der Länge des Schnabels (*bill\_length\_mm*). Evtl. können ja Pinguine mit längeren Schnäbeln besser Nahrung sammeln und sind daher schwerer.

- ☐ Berechnen Sie den Mittelwert von *body\_mass\_g* und *bill\_length\_mm* und speichern Sie die Ergebnisse in zwei neuen R-Elementen mit den Namen *body\_mass\_mean* und *bill\_length\_mean* ab.
- ☐ Nutzen Sie die Funktion `ggplot()` um ein Plot Objekt zu definieren, das auf dem Datensatz *Penguins* basiert und auf der X-Achse *bill\_length\_mm* darstellt und *body\_mass\_g* auf der Y-Achse.
- ☐ Ergänzen Sie nun eine Schicht (`geom`), die ein Streudiagramm der beiden Variablen darstellt.
- ☐ Färben Sie nun die Datenpunkte so ein, dass die drei Arten von Pinguinen unterscheidbar sind.
- ☐ Nutzen Sie `geom_smooth()` um eine Trendlinie hinzuzufügen. Sie wollen eine lineare Regressionsfunktion haben (nicht den Standard). Nutzen Sie die Hilfefunktion `?geom_smooth()` um herauszufinden, wie Sie diese erhalten können.
- ☐ Warum werden Ihnen nun drei Trendlinien angezeigt? Ändern Sie Ihre Abbildung so, dass nur eine Trendlinie (basierend auf allen Datenpunkten) berechnet wird.
- ☐ Nutzen Sie nun das *shape*-Attribut, um die Pinguinarten auch in der Form unterscheidbar zu machen.
- ☐ Nutzen Sie `labs()`, um Ihrer Abbildung einen Titel, einen Untertitel sowie sinnvolle Achsenbeschriftungen zu geben.
- ☐ Nutzen Sie die geoms: `geom_hline()` und `geom_vline()` um eine horizontale und eine vertikale Linie bei den oben berechneten Mittelwerten einzuzichnen. Färben Sie die Linien blau ein.
- ☐ Definieren Sie sich eine neue Farbpalette (einfach eine Liste anlegen mit `c()`) mit den Farbcodes aus dem Corporate Design der Stadt Würzburg. Wir nutzen diese drei `"#b30040"`, `"#86b3b3"`, `"#9ea49e"`. Geben Sie Ihrer Palette den Namen *WB\_cols*.
- ☐ Ändern Sie die Farbpalette in der Abbildung auf `scale_colour_manual(values = WB_cols)` und formatieren Sie Ihre Abbildung mit dem Theme: `theme_bw()`.
- ☐ Nutzen Sie die Exportfunktion von RStudio und speichern Sie die Abbildung als *Abbildung\_1.pdf* in Ihr Arbeitsverzeichnis ab.
- ☐ Erzeugen Sie eine zweite Abbildung, in der der Zusammenhang zwischen *body\_mass\_g* und *bill\_length\_mm* nach Pinguinarten getrennt angezeigt wird. Speichern Sie diese Abbildung als *Abbildung\_2.pdf* ab.
- ☐ Speichern Sie das R-Skript ab und prüfen Sie, ob Ihre Ergebnisse replizierbar sind.