

Datenmanagement

Dateipfade

1. Verwenden Sie den Befehl `sys.frame(1)$ofile`, um den absoluten Pfad des aktuellen Skripts zu ermitteln, und speichern Sie diesen in einer Variablen `skriptpfad`. Definieren Sie mithilfe dieser Variablen eine neue Variable `skriptordner`, die nur den Pfad zum Verzeichnis des Skripts (ohne Dateinamen) enthält. Dieser Ordner wird im Folgenden als Skriptordner bezeichnet. Geben Sie den absoluten Pfad des Skriptordners aus.
2. Ermitteln Sie das aktuelle Arbeitsverzeichnis mit `getwd()`, speichern Sie den Pfad in einer Variablen `current_wd` und geben ihn aus. Ändern Sie anschließend das Arbeitsverzeichnis mit `setwd()` auf den Skriptordner (verwenden Sie dazu die Variable `skriptordner`). Geben Sie das neue Arbeitsverzeichnis zur Überprüfung aus.
3. Ändern Sie Ihr Arbeitsverzeichnis zu einem anderen Ordner Ihrer Wahl (z.B. zum übergeordneten Verzeichnis mit `setwd("../")`). Überprüfen Sie mit `getwd()`, ob die Änderung erfolgreich war. Setzen Sie das Arbeitsverzeichnis anschließend wieder auf den ursprünglichen Skriptordner zurück und überprüfen Sie erneut, ob die Rücksetzung funktioniert hat.

Daten einlesen und speichern

4. Stellen Sie sicher, dass Ihr Arbeitsverzeichnis auf den Skriptordner gesetzt ist. Lesen Sie die Datei *Beispieldatensatz.csv* mit `read.csv()` in ein Dataframe ein. Speichern Sie dafür zunächst den Dateipfad in einer Pfadvariablen (z.B. `dateipfad`) und verwenden Sie diese Variable beim Einlesen. Geben Sie die ersten Zeilen des Dataframes mit `head()` aus.
5. Fügen Sie dem eingelesenen Dataframe eine neue Spalte oder Zeile mit selbst gewählten Werten hinzu. Speichern Sie das erweiterte Dataframe mit `write.table()` als neue CSV-Datei *Beispieldatensatz_erweitert.csv* im Skriptordner. Verwenden Sie auch hier eine Pfadvariable (z.B. `ausgabepfad`). Überprüfen Sie, ob die Datei erfolgreich gespeichert wurde, indem Sie sie erneut einlesen und ausgeben.
6. Ändern Sie Ihr Arbeitsverzeichnis zum übergeordneten Verzeichnis mit `setwd("../")`. Passen Sie die Pfadvariablen aus Aufgabe 4 und 5 so an, dass sie relative Pfade verwenden, die von Ihrem neuen aktuellen Arbeitsverzeichnis aus auf die Dateien im Skriptordner verweisen. Führen Sie das Einlesen und Speichern erneut aus, um zu überprüfen, ob die angepassten Pfade korrekt funktionieren.

Hinweise:

- Es ändert sich nur das Arbeitsverzeichnis und die Pfaddefinition, nicht die tatsächliche Ordnerstruktur.
- Wenn Ihr Skriptordner z.B. *Uebungen* heißt und Sie nun im übergeordneten Ordner *Programmierkurs* sind, könnte der absolute Pfad zur CSV-Datei z.B. `/home/user/Uni/Programmierkurs/Uebungen/Beispieldatensatz.csv` lauten.

Datensimulation

7. Der folgende Code generiert zwei Vektoren, die zwei Variablen eines Datensatzes mit Stichprobengröße $n = 30$ repräsentieren. Die erste Variable `var_1` enthält Gruppenzugehörigkeiten („Winter“ und „Sommer“) mit den gruppenspezifischen Stichprobengrößen `n_1` bzw. `n_2`. Die zweite Variable `var_2` enthält Zahlenwerte.

```
n <- 30
n_1 <- (n / 3) * 2
n_2 <- (n / 3) * 1

var_1 <- c(
  rep("Winter", n_1),
  rep("Sommer", n_2)
)

set.seed(0)
var_2 <- c(
  runif(n_1, min = 0, max = 5),
  runif(n_2, min = 5, max = 9)
)
```

Passen Sie den Code so an, dass die Stichprobengröße $n = 150$ beträgt und beide Gruppen gleich groß sind. Die erste Hälfte der Beobachtungen repräsentiert die Gruppe „Winter“, die zweite die Gruppe „Sommer“.

- Das Setzen des Zufallsgenerator-Seeds mit `set.seed(0)` ermöglicht es, reproduzierbare Ergebnisse zu erhalten.
- Die Funktion `runif()` wird zur Generierung von Zufallszahlen für `var_2` verwendet (?`runif` für Details).
- Passen Sie die Argumente so an, dass für „Winter“ Werte zwischen 0 und 10 und für „Sommer“ Werte zwischen 0 und 40 generiert werden.
- Runden Sie die Werte auf eine Nachkommastelle mit `round()`.
- Geben Sie beide Vektoren mit `print()` aus.

Optional: Überlegen Sie sich ein Beispiel, was die Variable `var_2` darstellen könnte.

8. Erstellen Sie ein Dataframe aus den Vektoren `var_1` und `var_2`. Vergeben Sie sinnvolle Spaltennamen (z.B. `Jahreszeit` und `Messwert`). Geben Sie die ersten Zeilen des Dataframes mit `head()` aus und speichern Sie es als CSV-Datei im Skriptordner unter dem Namen `simulierte_saisondaten.csv`.