

Navigate through R

Vorbereitung

- Verwenden Sie wieder die Ordnerstruktur von letzter Woche:
Hauptordner ProgStat mit Unterordnern Daten, Programme, Ergebnisse
- Öffnen Sie ein neues Syntax-Fenster.
- Schreiben Sie als Kommentar in die erste Zeile, was in dieser Syntax gemacht wird.
- Speichern Sie die Syntax unter einem sinnvollen Namen in Ihrem Programme-Ordner. (Denken Sie daran, die Syntax beim Arbeiten regelmäßig wieder zu speichern.)
- Löschen Sie alle Elemente aus dem Workspace.
- Setzen Sie das Working Directory (Arbeitsverzeichnis) auf den Hauptordner (ProgStat).

Global environment und Search Path

- Berechnen Sie den Mittelwert der Zahlen 1 bis 5.
- Die folgende Syntax definiert eine Funktion namens 'mean', die als Output immer 'Hallo' zurückgibt. Kopieren Sie diese in Ihre Syntax und führen Sie diese aus.

```
mean <- function(...){  
  return("Hallo")  
}
```

- Versuchen Sie, mit der Syntax von vorher den Mittelwert der Zahlen 1 bis 5 zu berechnen. Warum funktioniert das nicht mehr?
- Schauen Sie sich an, in welchen 'search paths' die Funktion 'mean' vorhanden ist?
- Ändern Sie den Funktionsaufruf so, dass wieder der Mittelwert berechnet wird.
- Was können Sie alternativ tun, damit der ursprüngliche Funktionsaufruf wieder funktioniert? Probieren Sie es aus.
- Laden Sie das Paket 'HSAUR2'. Wo wird es in den Search Path eingefügt, oben oder unten?
- Was bedeutet das? Angenommen, zwei Funktionen (mit demselben Namen) sind in zwei verschiedenen Paketen enthalten. Welche wird dann verwendet?

Packages und Namespaces

- In welcher Umgebung ('environment') ist die Funktion 'mean' definiert?
- In welcher Umgebung ('environment') ist die Funktion 'cor' definiert?
- Laden Sie das R-Paket 'samplingbook'.
- Speichern Sie den zugehörigen Namespace in einer Variablen.
- Geben Sie sich eine Liste mit allen (darin) definierten Objekten aus.

- Welche Objekte davon werden exportiert?
- Schauen Sie sich die Syntax von 'Smean' an.
- Schauen Sie sich die Syntax von 'print.Smean' an.

Generic functions

- Ist die Funktion 'mean' eine S3 oder S4-Funktion?
- Schauen Sie sich alle vorhandenen Funktionen an. Welche davon sind sichtbar und welche unsichtbar?
- Öffnen Sie die Hilfe zu 'mean.Date'.
- Schauen Sie die Syntax zu mean.Date an.
- Schauen Sie die Syntax zu mean.default an.

- Ist die Funktion 'summary' eine S3 oder S4-Funktion?
- Schauen Sie sich alle vorhandenen Funktionen an. Welche davon sind sichtbar und welche unsichtbar?
- Laden Sie das Paket 'stats4'.
- Ist die Funktion 'summary' jetzt eine S3 oder S4-Funktion? Begründen Sie das mit einem sinnvollen Aufruf.

Function Browsing und Debugging

Gegeben ist die folgende Funktion, die für einen Vektor `x` jeweils die Summe der benachbarten Werte berechnet. Der erste Eintrag im Ergebnisvektor ist gleich dem ersten Eintrag von `x`. Alle weiteren Einträge sind die Summe aus dem jeweiligen Eintrag und dem davor.

```
sum_benachbart <- function(x){
  # Ergebnisvektor initialisieren
  y <- c()
  # Das erste Element speichern
  y[1] <- x[1]

  # In einer Schleife alle weiteren Elemente berechnen
  for(i in 2:length(x)){
    y[i] <- x[i] + x[i-1]
  }
  # Ergebnis zurückgeben
  return(y)
}
```

- Kopieren Sie die Funktion in Ihre Syntax.
- Rufen Sie die Funktion nacheinander mit den folgenden drei Vektoren auf
 - `c(1, 2, 4, 8, 16, 32)`
 - `c("a", "b", "c", "d")`
 - `c(1, 2, 4, "text", 16, 32)`
- Wenden Sie 'debug()', 'options(error=recover)' und 'traceback()' auf dieses Beispiel an.