Programmieren mit Statistischer Software **6. Block**
Grün **SS 2017**

6 Funktionen in R

Aufgabe 1:

Schreiben Sie eine Funktion in R, die das α-Quantil bestimmt. Die Funktion soll folgendermaßen definiert sein:

```
> quantile <- function(x, alpha) {}
```

- Berechnen Sie das Quantil wie dies in der LV Methodenlehre I definiert wurde (siehe C. Duller, Einführung in die Statistik mit Excel und SPSS).
- Bestimmen Sie das 0.30-Quantil der Zahlen von 1 bis 100. Vergleichen Sie das Ergebnis, mit dem der `quantile()` Funktion aus dem Paket **stats** in R.

Hinweis: Eine Funktion zum Bestimmen des Median finden Sie in Ligges (2005) auf Seite 74.

Aufgabe 2:

Für einen ausgefüllten **Lottoschein** (6 aus 45) soll für n Durchläufe die Anzahl der Richtigen simuliert werden.

Schreiben Sie eine Funktion

```
> lotto <- function(schein, n) { ... }
```

welche einen ausgefüllten Lottoschein als Vektor bestehend aus 6 Zahlen zwischen 1 und 45 und die Anzahl der Durchläufe übernimmt. Rückgabewert ist ein Vektor mit dem ersten Element die Anzahl "Keine Richtig", zweites Element die Anzahl "Eine Richtig", ..., siebtes Element die Anzahl "Sechs richtige" (Simulation ohne "Fünf Richtige mit Zusatzzahl").

Berechnen Sie die Häufigkeiten bei 1000 Durchläufen für folgenden Schein:

```
> set.seed(1234)
> schein <- c(1, 7, 18, 36, 9, 45)
> schein1000 <- lotto(schein, 1000)
```

Vergleichen Sie diese Werte mit den theoretisch erwarteten Werten.

Aufgabe 3:

Der Binomialkoeffizient von n und m ist definiert als

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}.$$

Schreiben Sie eine Funktion `binomialkoeffizient`, die folgendermaßen definiert ist:

```
> binomialkoeffizient <- function(n, m) {}
```

und mithilfe von `factorial()` den Binomialkoeffizient berechnet.

- Überprüfen Sie, ob $n \geq m > 0$ gilt und geben Sie ansonsten 0 zurück.

Berechnen Sie damit die folgenden Binomialkoeffizienten:

$$\binom{4}{2}, \binom{50}{20}, \binom{500}{200}$$

Aufgabe 4:

Verwenden Sie `log()`, `sum()` und `exp()`, um eine verbesserte Funktion zum Berechnen des Binomialkoeffizienten zu erstellen, die auch für größere Werte von n und m noch ein Ergebnis liefert. D.h.

$$\binom{n}{m} = \exp\left(\left(\sum_{i=1}^{n} \log(i)\right) - \left(\sum_{i=1}^{m} \log(i)\right) - \left(\sum_{i=1}^{n-m} \log(i)\right)\right).$$

Berechnen Sie damit ebenfalls die folgenden Binomialkoeffizienten:

$$\binom{4}{2}, \binom{50}{20}, \binom{500}{200}$$

Aufgabe 5:

Schreiben Sie eine Funktion, die den Spearmanschen Rangkorrelationskoeffizient ohne Bindungen berechnet. Dieser wird berechnet mittels

$$\rho_s = 1 - \frac{6 \sum d_i^2}{N \cdot (N^2 - 1)},$$

wobei r_i, s_i die Ränge sind und $d_i = r_i - s_i$ die Rangzahlendifferenz der i-ten Erhebungseinheit (siehe C. Duller, Einführung in die Statistik mit Excel und SPSS).

Die Funktion soll folgendermaßen definiert sein:

```
> spearman <- function(x, y, na.rm = TRUE) {}
```

- Überprüfen Sie, ob `x` und `y` gleich lang sind. Wenn nicht, sollen vom längeren Vektor nur die ersten n Beobachtunten verwendet werden, wenn n die Länge des kürzeren Vektors ist.
- Falls `na.rm` gleich `TRUE` ist, sollen alle Einträge in `x` und `y` entfernt werden, wo an der Stelle `x` oder `y` gleich `NA` sind. Falls `na.rm` gleich `FALSE` und `x` oder `y` `NA`s enthalten, dann wird `NA` zurückgegeben.
- Überprüfen Sie, ob in den Daten keine Bindungen vorhanden sind. Falls ja, geben Sie `NA` zurück.

Wenden Sie die Funktion folgendermaßen an:

```
> spearman(swiss$Fertility, swiss$Agriculture)
> x <- subset(swiss, Fertility > 70)
> spearman(x$Fertility, x$Agriculture)
> spearman(x$Fertility, x$Agriculture[1:20])
```