

## Übungen zur Vorlesung

### Einführung in die angewandte Bioinformatik

19.06.2008

Sommersemester 2008

Blatt 8

Dieses Übungsblatt beschäftigt sich hauptsächlich mit R. Starten Sie R von der Konsole, indem Sie einfach den Befehl `R` eingeben. Im Folgenden sind die Befehle, die Sie an der R-Kommandozeile eingeben sollen, durch ein vorangestelltes Größer-Zeichen ">" gekennzeichnet. Kommentare sind durch ein vorangestelltes "#"-Zeichen gekennzeichnet. Dieser Teil der Zeile ist nur zur Erläuterung gedacht und muss nicht abgetippt werden. Probieren Sie es gleich mal aus:

```
> help.start() # Hilfe starten
> ?rnorm      # Hilfe zur rnorm-Funktion anzeigen
```

#### Aufgabe 8.1 – R als Taschenrechner

Im einfachsten Fall können Sie R als Taschenrechner benutzen.

```
> 17+4
> 17 + 2 * 2 # Gilt Punkt vor Strich? _____
> 9^2       # Was macht das "^" ? _____
> log(10)   # natürlicher oder dekadischer Logarithmus? _____
> log(exp(2))
```

Wie wird die letzte Zeile in normaler mathematischer Notation aufgeschrieben (schauen Sie sich evtl. die Hilfe zu `exp` an)? \_\_\_\_\_

Sie können eine neue Variable definieren, indem Sie ihr mit dem Zuweisungsoperator `=` einen Wert zuweisen.

```
> x = exp(2) # exp(2) berechnen und Ergebnis in Variable x speichern
> x         # Inhalt von Variable x anzeigen
> log(x)
```

#### Aufgabe 8.2 – Vektoren

Um mehrere Zahlen zu speichern, benutzt man in R *Vektoren*, die mit der in R eingebauten Funktion `c` erzeugt werden. Auch Vektoren können in Variablen gespeichert werden (hier heißt die Variable `v`).

```
> v = c(22, 10.4, 65, 9.8)
> v
```

Es gibt weitere eingebaute Funktionen. Was berechnen die folgenden Zeilen? (Schauen Sie zunächst nicht in die Hilfe.)

```
> length(v)
> sum(v)
> mean(v)
> sum(v)/length(v)
```

---

Berechnen Sie nun die Standardabweichung mit der Funktion `sd`, die Varianz mit `var` und den Median mit `median`.

Einige weitere Übungen zu Vektoren:

```
> v * 2          # alle Elemente des Vektors mit 2 multiplizieren
> seq(20, 30, 2.5) # Funktion, die Vektor mit Zahlenfolge erzeugt
> runif(5)       # Was macht dieser Funktionsaufruf?
# _____
```

### Aufgabe 8.3 – Analyse der Klausurergebnisse

Laden Sie sich von der Übungswebseite die Datei `klausur.dat` herunter und speichern Sie sie ab. Sehen Sie sich die Datei an: Sie enthält die Gesamtpunktzahlen, die bei der Klausurübung erzielt wurden. Laden Sie diesen Datensatz in eine Variable namens `klausur`.

```
> klausur = read.delim("klausur.dat") # evtl. Pfad angeben, also
# z.B. "Desktop/klausur.dat"!
> dim(klausur)          # Dimension rausfinden
> colnames(klausur)    # Überschriften (column names) anschauen
> klausur              # alle Daten ansehen
> klausur[1:10,]       # die ersten zehn Einträge anschauen
> klausur$Punkte       # nur die Werte der "Punkte"-Spalte anschauen
> klausur$Geschlecht  # nur die "Geschlecht"-Spalte anschauen
```

So zeigen Sie nur die Punkte an, die die männlichen Teilnehmer erreicht haben:

```
> klausur$Punkte[klausur$Geschlecht == "m"]
```

"==" überprüft auf Gleichheit ("=" ist für Zuweisungen!).

Wie können Sie die durchschnittlich erzielten Punkte aller Teilnehmer berechnen?

Wie können Sie die durchschnittlich erzielten Punkte der weiblichen Teilnehmer berechnen?

Bevor Sie gleich den ersten Plot zeichnen, möchten Sie sich noch etwas Tipparbeit sparen.

```
> attach(klausur) # ab jetzt kann das "klausur$" weggelassen werden
> Punkte          # statt klausur$Punkte
```

Nun wird es grafisch: Lassen Sie sich ein Histogramm und einen Boxplot über alle Punktzahlen anzeigen (probieren Sie auch, jeweils den Parameter `col="green"` hinzuzufügen).

```
> hist(Punkte, seq(5, 30, 5))
> boxplot(Punkte)
```

### Aufgabe 8.4 – Vergleich Männern und Frauen

Sie möchten jetzt wissen, ob die Punktzahlen normalverteilt sind. Zur Auffrischung schauen Sie sich an, wie die Dichtefunktion der Normalverteilung aussieht (dnorm: **d**ensity of **n**ormal distribution).

```
> curve(dnorm(x), from=-4, to=4)
```

Erzeugen sie jetzt einen Q-Q-Plot für die Punktzahlen der Frauen und einen für die Punktzahlen der Männer.

```
> qqnorm(Punkte[Geschlecht == "w"])
> qqline(Punkte[Geschlecht == "w"]) # Linie dazu
> (hier das gleiche für die Männer)
```

Wessen Punktezahlen sind einer Normalverteilung am ähnlichsten?

---

Beantworten Sie nun, ob es einen signifikanten Unterschied der Punktzahlen von Männern und Frauen gibt, indem Sie einen t-Test benutzen.

```
> t.test(Punkte[Geschlecht=="m"], Punkte[Geschlecht=="w"])
```

Ist der Unterschied signifikant? \_\_\_\_\_

### Aufgabe 8.5 – Der “Iris”-Datensatz

R enthält standardmäßig den Datensatz “Iris”. Lassen Sie sich den Datensatz ausgeben und schauen Sie sich die Hilfe an.

```
> iris
> ?iris
> attach(iris) # wieder Tipparbeit sparen
```

Visualisieren Sie eines der Attribute mit einem Histogramm, z. B. `Petal.Length`. Wie lautet der Befehl? \_\_\_\_\_

Lassen Sie sich jetzt einen Scatterplot anzeigen, der ein Paar von Attributen gegeneinander abbildet, z. B. `Petal.Length` gegen `Petal.Width`.

```
> plot(Petal.Length, Sepal.Length)
```

Mit

```
> plot(iris)
```

können Sie sich Scatterplots von allen Paaren anzeigen lassen.

Zwischen welchen der Attributen herrscht der stärkste Zusammenhang?

---

Prüfen Sie dies zusätzlich, indem Sie die Korrelation zwischen **diesen beiden** Attributen berechnen. Beispielsweise:

```
> cor(Petal.Length, Sepal.Width)
```

Wie hoch ist die Korrelation? \_\_\_\_\_

In einem nächsten Schritt geht es darum, den Zusammenhang der Attribute `Petal.Length`, `Petal.Width`, `Sepal.Width` und `Sepal.Length` zur Spezies herauszufinden.

Mit dem Befehl

```
> plot(iris[,1:4], col = as.numeric(Species))
```

können Sie sich einen Scatterplot zwischen diesen Attributen anzeigen lassen, wobei die Spezies eingefärbt ist. Welche Attributpaare eignen sich gut, um die Spezies voneinander zu trennen, welche eignen sich nicht so gut?

---

---