

## 8 Weitere Details der Sprache, Vergleich von Objekten

### Aufgabe 1:

Simulieren Sie 10000 normalverteilte Zufallsvariablen  $X$  mit Erwartungswert 1 und Varianz 2. Gegeben  $X$  simulieren Sie weitere 10000 auf  $[-1 + X, 1 + X]$  gleichverteilte Zufallsvariablen,  $U$ . Nehmen Sie weiters an, dass sich aus der Summe der Zufallsvariablen eine weitere Zufallsvariable  $Y$  ergibt, d.h.

$$Y = X + U.$$

- Schätzen Sie den Erwartungswert der Zufallsvariable  $Y$  mithilfe des empirischen Mittels und vergleichen Sie ihn mit dem Wert, wo sie die geschätzten Erwartungswerte für  $X$  und  $U$  addieren.
- Schätzen Sie die Varianz von  $Y$  durch die empirische Varianz und vergleichen Sie sie mit dem Wert, wo sie die geschätzten Varianzen für  $X$  und  $U$  addieren.

### Aufgabe 2:

Schreiben Sie eine Funktion

```
> esample <- function(value, prob, g = function(x) x) {}
```

die bzgl. einer diskreten Verteilung auf den Werten `value` mit den Wahrscheinlichkeiten `prob` den Erwartungswert der Funktion `g` bestimmt.

Überprüfen Sie die Argumente innerhalb der Funktion auf ihre Zulässigkeit.

- Falls `prob` fehlt, sollen alle Werte gleich wahrscheinlich sein.
- Falls `prob` nichtnegative Werte enthält, stellen Sie sicher, dass die Summe 1 ist, werfen Sie aber keine Warnung oder Fehler dazu aus.

Berechnen Sie damit

```
> n <- 10  
> esample(1:n)  
> esample(0:n, prob = dpois(0:n, 1))  
> esample(1:n, g = function(x) x^2)
```

### Aufgabe 3:

Schreiben Sie eine Funktion `rtnorm()`, die Beobachtungen aus einer gestutzten (trunkierten) Normalverteilung zieht. Die Argumente sind dabei `n`, `mean` und `sd` wie bei `rnorm()`. Zusätzlich gibt es die Argumente `lower` und `upper`, die die oberen und unteren Schranken des Wertebereichs der Verteilung angeben.

Die Dichte der gestutzten Normalverteilung mit Parameter  $\mu$ ,  $\sigma$ ,  $L$  und  $U$  ist gegeben durch

$$f(y|\mu, \sigma, L, U) = \frac{1}{\Phi(U|\mu, \sigma) - \Phi(L|\mu, \sigma)} \phi(y|\mu, \sigma) \mathbb{1}_{[L, U]}(y),$$

wobei  $\phi(\cdot|\mu, \sigma)$  die Dichte der Normalverteilung mit Erwartungswert  $\mu$  und Standardabweichung  $\sigma$  und  $\Phi(\cdot|\mu, \sigma)$  die entsprechende Verteilungsfunktion sind.

Die Funktion soll  $2n$  Beobachtungen aus der Normalverteilung ziehen und dann die ersten  $n$  zurückgeben, die in den Wertebereich  $[L, U]$  fallen. Falls dies auf weniger als  $n$  Beobachtungen zutrifft, wird ein Fehler ausgegeben.

- Verwenden Sie die Funktion, um mithilfe von Simulation den Erwartungswert der Verteilung zu bestimmen, die sich ergibt, wenn man eine standardnormalverteilte Zufallsvariable auf positive Werte stutzt. Setzen Sie vorab einen Seed, sodass der Aufruf erfolgreich ist.

#### Aufgabe 4:

Bestimmen Sie, wie häufig die Strategie erfolgreich ist,  $2n$  Beobachtungen aus einer Standardnormalverteilung zu ziehen, wenn man  $n$  haben möchte, die aus einer auf die positiven Werte gestutzten Standardnormalverteilung generiert wurden.

- Schreiben Sie eine Funktion, die die Funktion `rtnorm()` wiederholt aufruft. Verwenden Sie `try()`, um den Erfolg zu messen.
- Die Funktion soll den Anteil der erfolgreichen Aufrufe zurückgeben.
- Verwenden Sie die Funktion, um für  $m$  Versuche den Anteil der erfolgreichen Aufrufe für eine auf positive Werte gestutzte Standardnormalverteilung mit  $n = 10$  und  $n = 100$  zu schätzen.
- Wie hoch ist der Anteil, wenn Sie ihn theoretisch bestimmen?

#### Hinweis:

Wenn man bei Simulationen dieselben Ausdrücke mit unterschiedlichen Zufallszahlen durchführen möchte, kann man dafür auch `replicate()` verwenden. Z.B.

```
> y <- replicate(1000, mean(rnorm(100)))
> mean(y); sd(y)
```

```
[1] -0.003119782
```

```
[1] 0.1032096
```

#### Aufgabe 5:

Schreiben Sie eine Funktion

```
> mysummary <- function(x, g) {}
```

die einen Datensatz  $x$  bzgl. der kategoriellen Variable  $g$  aufspaltet und für jeden Teildatensatz den Mittelwert und die Standardabweichung jeder Variable ausrechnet und dies in einer benannten Liste abspeichert und als Objekt der Klasse "mysummary" zurückgibt, das auch die Information enthält, wieviele Beobachtungen im Datensatz enthalten sind.

Schreiben Sie dann weiters eine `print()` Methode für diese Klasse, die folgenden Output erzeugt.

```
> ms <- mysummary(iris[, 1:4], iris$Species)
> class(ms)
```

```
[1] "mysummary"
```

```
> print(ms, digits = 2)
```

Der Datensatz enthält 150 Beobachtungen.

```
$setosa
      Sepal.Length Sepal.Width Petal.Length Petal.Width
Mean           5.01         3.43         1.46         0.25
SD             0.35         0.38         0.17         0.11
```

```
$versicolor
      Sepal.Length Sepal.Width Petal.Length Petal.Width
Mean           5.94         2.77         4.26         1.3
SD             0.52         0.31         0.47         0.2
```

```
$virginica
      Sepal.Length Sepal.Width Petal.Length Petal.Width
Mean           6.59         2.97         5.55         2.03
SD             0.64         0.32         0.55         0.27
```

Eine `print()` Methode wird implementiert, indem man folgende Funktion definiert

```
> print.mysummary <- function(x, ...) {}
```