
Statistische Methoden der Datenanalyse

Markus Schumacher

Übung II

Markus Warsinsky

7.11.2011

Anwesenheitsaufgaben

Aufgabe 10 *Wechseln oder nicht — mal simuliert!*

In einer Spielshow gibt es drei Türen. Hinter einer Tür wartet ein Gewinn, hinter den beiden anderen Türen befinden sich Nieten. Nachdem der Kandidat eine Tür ausgewählt hat, öffnet der Moderator eine Tür mit einer Niete (aber nicht die Tür, die der Kandidat gewählt hat). Der Kandidat darf erneut eine der beiden übrigen Türen wählen.

In dieser Programmierübung wollen wir diese Ihnen bereits bekannte Aufgabenstellung mithilfe einer Simulation bearbeiten. Dazu wollen wir die Spielshowsituation N mal nacheinander zufällig nachstellen. Dazu erst folgende Grundbetrachtungen:

- Wir nehmen an, dass die drei Türen als Tür 0, Tür 1 und Tür 2 bezeichnet seien, so dass wir eine Beziehung zwischen den ganzen Zahlen und den Türen erhalten.
- Ohne Beschränkung der Allgemeinheit wähle der Kandidat jedesmal Tür 0.
- Die wahre Platzierung des Preises wird in jedem Zufallsexperiment zufällig ermittelt.
- Sollte der Kandidat bereits richtig liegen, öffnet der Showmaster zufällig Tür 1 oder Tür 2. Sollte der Preis nicht hinter Tür 0 liegen, so bleibt dem Showmaster selbstverständlich nur eine Alternative.

Gehen Sie nun wie folgt vor:

- (i) Zunächst wollen wir die Spielshow nur einmal nachstellen. Programmieren Sie ein ROOT-Skript mit den folgenden Komponenten:
 - a) Einen Zufallszahlengenerator vom Typ `TRandom3`.
 - b) Ermitteln Sie mittels dieses Zufallszahlengenerators die wahre Platzierung des Preises als Zufallszahl zwischen 0 und 2 (`Integer(3)`). Merken Sie sich diese Zahl in einer Variablen `int ngewinn`.
 - c) Erstellen Sie eine weitere ganze Zahl, der Sie zunächst den Wert 0 zuweisen (`int nnotshow=0`). Sollte der Kandidat bereits richtig liegen, ermitteln Sie eine weitere Zufallszahl zwischen 1 und 2 (`Integer(2)+1`), die angibt, welches Tor nicht geöffnet wird und merken Sie sich diese Zahl mittels `nnotshow`. Ob der Kandidat bereits richtig liegt, oder nicht, ermitteln Sie mittels einer `if`-Abfrage. Sollte der Kandidat falsch liegen, so merken Sie sich die Tür, die automatisch geöffnet wird in der Variablen `nnotshow`.
 - d) Gehen Sie nun davon aus, dass der Kandidat die Tür wechselt. Ermitteln Sie, ob der Kandidat gewonnen hat, oder nicht und geben Sie sich das Resultat auf dem Bildschirm aus. (Tipp: Vergleichen Sie `nnotshow` mit `ngewinn`.)
- (ii) Nun möchten wir dieses Experiment N ausführen, für den Anfang z.B. $N = 10$. Gehen Sie dazu wie folgt vor:
 - a) Erstellen Sie sich am Anfang des Skriptes eine Variable `int npoints=10`;

- b) Erstellen Sie eine `for`-Schleife, die `npoints` mal das Experiment durchführt. Stellen Sie sicher, dass der Zufallszahlengenerator ausserhalb der `for`-Schleife erstellt wird, da sonst bei jedem Zufallsexperiment identische Zufallszahlen erstellt werden. Lassen Sie wie bereits zuvor für jedes Zufallsexperiment am Bildschirm ausgeben, ob der Kandidat gewonnen hat, oder nicht.
- (iii) Ändern Sie nun den Startwert für den Zufallszahlengenerator mittels `SetSeed`. Beachten Sie, dass sich die Abfolge der Ergebnisse ändert.
- (iv) Nun wollen wir ermitteln, wie oft der Kandidat bei dieser Strategie (Wechseln) im Mittel gewinnen würde. Dazu wollen wir zunächst nur eine Zahl ermitteln, den Mittelwert der Stichprobe x_1, \dots, x_N , wobei die x_i entweder 0 (verloren) oder 1 (gewonnen) sein können. Erstellen Sie sich dazu ausserhalb der `for`-Schleife eine Fließkommazahl `float mittelwert=0;`. Zählen Sie für jedes Zufallsexperiment zu dieser Zahl 1 hinzu, wenn der Kandidat gewinnt und 0, falls nicht. Dividieren Sie am Ende durch die Anzahl der Zufallsexperimente und lassen Sie sich den erhaltenen Mittelwert am Bildschirm ausgeben. Erhöhen Sie `npoints`, was beobachten Sie?
- (v) Letztendlich wollen wir noch den Mittelwert in Abhängigkeit von der Stichprobengröße graphisch darstellen. Dazu machen wir uns zunutze, dass man den Mittelwert bei Vergrößerung einer Stichprobe aus dem alten Mittelwert und der Größe der Stichprobe ermitteln kann:

$$\bar{x}_N = \bar{x}_{N-1} + \frac{x_n - \bar{x}_{N-1}}{N}$$

Gehen Sie nun wie folgt vor:

- a) Benutzen Sie diese Rekursionsformel um an jedem Punkt der `for`-Schleife den aktuellen Mittelwert zu berechnen.
- b) Erstellen Sie ausserhalb der `for`-Schleife ein Histogramm mit `npoints` bins zwischen 0 und `npoints`.
- c) Benutzen Sie innerhalb der `for`-Schleife die Methode `SetBinContent(i,mittelwert)`, um das `i`-te Bin auf den jeweils aktuellen Mittelwert zu setzen.
- d) Zeichnen Sie schliesslich das Histogramm am Ende des Skriptes. Probieren Sie verschiedene Werte für `npoints`. Ändern Sie den Startwert des Zufallszahlengenerators. Was beobachten Sie?

Aufgabe 11 *Binomialverteilung*

In dieser Programmierübung soll die sogenannte Binomialverteilung zufällig erzeugt werden, indem flach verteilte Zufallszahlen zwischen 0 und 1 erzeugt werden. Die Binomialverteilung beschreibt die Wahrscheinlichkeit, bei N Versuchen, wobei bei jedem Versuch ein bestimmtes Ereignis mit einer Wahrscheinlichkeit μ auftritt genau n mal einen Erfolg zu haben. Als Beispiel kann man sich die Wahrscheinlichkeit vorstellen, bei $N = 20$ Würfelwürfen genau $n = 5$ mal eine sechs zu würfeln ($\mu = 1/6$ bei einem normalen Würfel).

Um die Binomialverteilung zu erzeugen gehen Sie nun wie folgt vor:

- (i) Zunächst wollen wir das Zufallsexperiment nur einmal durchführen. Erstellen Sie ein ROOT-Skript, das die folgenden Aufgaben durchführt:
 - a) Erstellen eines Zufallszahlengenerators vom Typ `TRandom3`.
 - b) Definition einer ganzen Zahl `int nwuerfe=20;`
 - c) Definition einer Fließkommazahl `float mu=1./6.;`
 - d) Definition einer ganzen Zahl `int nsechs=0;`, die benutzt werden soll, um sich die Anzahl der Erfolge zu merken.
 - e) Erstellen Sie dann eine `for`-Schleife, die genau `nwuerfe` mal mittels der `Uniform()`-Funktion des Zufallszahlengenerators eine Zufallszahl zwischen 0 und 1 erstellt. Sollte diese Zufallszahl kleiner sein als `mu`, so wird `nsechs` um eins erhöht. Geben Sie nach der `for`-Schleife das Resultat für `nsechs` am Bildschirm aus.
- (ii) Nun wollen wir dieses Zufallsexperiment sehr oft durchführen. Gehen Sie dazu wie folgt vor:
 - a) Erstellen Sie eine weitere `for`-Schleife, die alle anderen Schritte, die zuvor ausgeführt wurden beispielsweise 100 mal ausführt. Beachten Sie allerdings, dass Sie weiterhin den Zufallszahlengenerator vor dieser `for`-Schleife erstellen, da sich sonst die Sequenz der Zufallszahlen

in jedem Experiment wiederholen würde. Geben Sie weiterhin für jedes Zufallsexperiment den Wert von `nsechs` am Bildschirm aus, um zu testen, dass sich dieser ändert.

- b) Erstellen Sie nun ausserhalb beider `for`-Schleifen ein Histogramm mit `nwerfe+1` Bins von `-0.5` bis `nwerfe+0.5`. Füllen Sie für jedes Zufallsexperiment den Wert von `nsechs` in das Histogramm. Normieren Sie das Histogramm auf eins und stellen Sie es am Bildschirm dar.
- c) Verändern Sie nun einige der Parameter. Was passiert, wenn man `nwerfe` erhöht?

Die so erstellte Simulation der Binomialverteilung wird in der morgigen Vorlesung besprochen!

Hausaufgaben

Aufgabe 12 *Rekursive Mittelwert- und Varianzberechnung*

4 Punkte

Gegeben sei eine Stichprobe (x_1, \dots, x_n) . Zeigen Sie:

- (i) Wenn \bar{x}_n der Mittelwert der ersten n Elemente ist, dann gilt:

$$\bar{x}_n = \bar{x}_{n-1} + \frac{x_n - \bar{x}_{n-1}}{n}$$

- (ii) Wenn σ_n^2 die Varianz der ersten n Elemente ist, dann gilt:

$$\sigma_n^2 = \frac{(n-1)\sigma_{n-1}^2 + (x_n - \bar{x}_n)(x_n - \bar{x}_{n-1})}{n}$$

Aufgabe 13 *Binomialverteilung in der Praxis*

3 Punkte

Nehmen Sie an, ein Programmierer macht in 99,5% seiner erstellten Programmzeilen keinen Fehler.

- (i) Wie viele Programmzeilen muss er schreiben, um mit einer Wahrscheinlichkeit von mindestens 50% mindestens einen Fehler zu machen?
- (ii) Wie viele Programmzeilen muss er schreiben, um mit einer Wahrscheinlichkeit von mindestens 50% mindesten zwei Fehler zu machen?

Aufgabe 14 *Variablentransformation*

3 Punkte

Betrachtet seien zwei Zufallsvariablen x_1 und x_2 mit zugehörigen Wahrscheinlichkeitsdichtefunktionen $f_1(x_1)$ und $f_2(x_2)$. Die beiden Zufallsvariablen seien unabhängig, das heisst für die kombinierte Wahrscheinlichkeitsdichteverteilung gilt:

$$f(x_1, x_2) = f_1(x_1) \cdot f_2(x_2)$$

Zeigen Sie: Falls x_1 und x_2 in zwei neue Zufallsvariablen y_1 und y_2 transformiert werden, so dass $y_1 = y_1(x_1)$ und $y_2 = y_2(x_2)$ gilt, so sind auch die beiden neuen Zufallsvariablen unabhängig voneinander.

Aufgabe 15 *Mellinfaltung*

4 Punkte

Gegeben seien zwei unabhängige Zufallsvariablen x und y , die beide gleichverteilt sind zwischen 0 und 1, d.h. ihre Wahrscheinlichkeitsdichtefunktion ist gegeben durch

$$g(x) = \begin{cases} 1 & 0 < x < 1 \\ 0 & \text{sonst} \end{cases} \quad (1)$$

und analog durch $h(y)$. Benutzen Sie die Mellin-Faltung, um zu zeigen, dass die Wahrscheinlichkeitsdichtefunktion $f(z)$ von $z = xy$ gegeben ist durch:

$$f(z) = \begin{cases} -\ln z & 0 < z < 1 \\ 0 & \text{sonst.} \end{cases} \quad (2)$$

Aufgabe 16 *Fehlerfortpflanzung I*

2 Punkte

Nehmen Sie an, sie möchten die Länge eines Werkstücks möglichst exakt wissen. Was ist zu bevorzugen, um eine möglichst genaue Messung zu erhalten:

- Eine Einzelmessung mit einer Genauigkeit von 0,2 mm, oder
- eine 10-fache unabhängige Messung mit einer Genauigkeit von 1 mm?

Aufgabe 17 *Fehlerfortpflanzung II*

2 Punkte

Zwischen der an einem Leiter angelegten Spannung U , dem Widerstand R und dem fließenden Strom I besteht der Zusammenhang (Ohmsches Gesetz)

$$U = R \cdot I.$$

- (i) Nehmen Sie an, der Strom wurde gemessen zu $I = 1120 \pm 10$ mA und der Widerstand zu $1400 \pm 30 \Omega$. Berechnen Sie den Wert der Spannung und ihren Fehler.
- (ii) Nehmen Sie an, die Spannung wurde gemessen zu $U = 45 \pm 1$ V und der Widerstand zu $900 \pm 10 \Omega$. Berechnen Sie den Wert des Stromes und seinen Fehler.

Aufgabe 18 *Fehlerfortpflanzung III*

2 Punkte

Gegeben sei ein Winkel $\theta = 0,56 \pm 0,01$. Wie groß sind die Fehler auf $\sin \theta$, $\cos \theta$ und $\tan \theta$? Wie groß sind sie für den Fall $\theta = 1,56 \pm 0,01$?