Aufgabe zu Permutationen

- Implementieren Sie einen approximativen Permutationstest für die Stichprobenkonstellation $X \sim N(1.4, 2)$ vom Umfang 50, $Y \sim N(1, 2)$ vom Umfang 75 und die Hypothese, dass $\mu_X = \mu_Y$ über die Teststatistik $\bar{X} - \bar{Y}$. (Dies ist äquivalent zur Bestimmung des p_{approx}.)

- Überlegen Sie hierzu, welche Parameter festgelegt und aus welchen Grundgesamtheiten welche Stichproben gezogen werden müssen.

- Inwiefern ist Fischer's exakter Test ein Permutationstest?

Lösung:

```
x<-rnorm(50, mean=1, sd=sqrt(2))
y<-rnorm(75, mean=1, sd=sqrt(2))
lx <- 50 ; ly <- 75
thetadach <- mean(x)-mean(y)

papprox <- rep(NA, 1000)
pooled <- c(x, y)
approx.runs <- 10000
thetastern <- rep(NA, approx.runs)

#### beide Möglichkeiten
#### 1. Poolen und die gepoolte Stichprobe permutieren,
#### Die ersten 50 sind x, die 75 danach y

for (i in 1:approx.runs) {
  permutation <- sample(pooled, length(pooled), replace=FALSE)
  xstern <- permutation[1:lx]
```

```
  ystern <- permutation[(lx+1):(lx+ly)]
  thetastern[i] <- abs(mean(xstern) - mean(ystern))
}
cat("p-approx von ", abs(thetadach), ": ",
     length(which(thetastern > abs(thetadach)))/approx.runs , "\n")

### 2. Poolen, dann die Indices ziehen, die x bilden,
### der Rest ist y

for (i in 1:approx.runs) {
  permutationx <- sample(1:length(pooled), lx, replace=FALSE)
  xstern <- pooled[permutationx]
  ystern <- pooled[-permutationx]
  thetastern[i] <- abs(mean(xstern) - mean(ystern))
}
cat("p-approx von ", abs(thetadach), ": ",
     length(which(thetastern > abs(thetadach)))/approx.runs , "\n")
### Die p-Werte sollten sehr ähnlich sein.
```

Fazit Resampling

- Mittels Resampling können Streuungsmaße für Schätzer auch in den Fällen angegeben werden, in denen nur eine Stichprobe vorliegt.
- Dies geschieht, abgesehen vom parametrischen Jackknife, ohne Verteilungsannahme.
- Bootstrapping arbeitet mit Stichproben des vollen Umfangs mit Zurücklegen,
- der Jackknife mit Teilstichproben ohne Zurücklegen.
- Crossvalidation kann gut genutzt werden, um Modelle auf ihre Gültigkeit zu überprüfen. (Trainingsset, Testset)

- Sowohl Crossvalidation als auch Jackknife kommen oft ohne echtes Stichprobenziehen aus, da systematisch die Teilstichproben erzeugt werden können.

- Beim Testen mit Resamplingmethoden ist darauf zu achten, dass man die empirischen p-Werte tatsächlich unter der Nullhypothese erzeugt werden.

- Permutationstests nutzen im exakten Fall alle möglichen Permutationen, können aber durch eine zufällige Auswahl von Permutationen approximiert werden.

Zusatzaufgaben

- (Zusatzaufgabe) Schreiben Sie ein Programm, dass alle Permutationen eines gegebenen Vektors ausgibt.

- (Zusatzaufgabe) Schreiben Sie ein Programm, dass alle Permutationen für den obigen Zweistichprobenfall erzeugt. Gegeben seien zwei Vektoren, einmal die Messwerte und einmal die Gruppenzugehörigkeiten.

Warteschlangentheorie (und Simulation)

- Nach Lecture Notes von Sandy Rutherford, IRMACS, 2006
- Die Warteschlangentheorie nahm ihren Ausgang zu Beginn des 20. Jhdts. in Dänemark in der Telekommunikationsindustrie!
- Agner Krarup Erlang war Mathematiker, angestellt bei der Telefongesellschaft von Kopenhagen.
- Seine Aufgabe war, herauszufinden, wie viele Telefonkabel in verschiedene Dörfer Dänemarks gelegt werden müssen, damit die Wahrscheinlichkeit, dass ein potentieller Kunde ein Besetztzeichen zu hören bekommt kleiner ist als 5%.
- Titel der Arbeit in der "Nyt tidskrift for Matematik" (1909): The Theory of Probabilities and Telephone Conversations.

Anwendungsfälle der Warteschlangentheorie

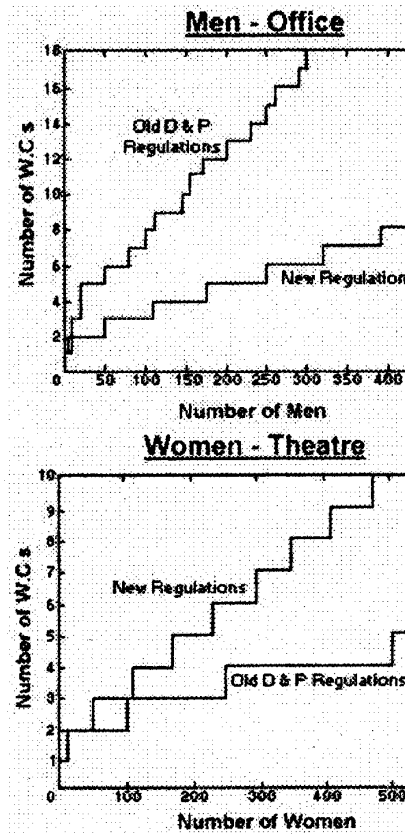
- Mautstationen auf Autobahnen
- Anflüge auf einen Flughafen
- Wartelisten bei Operationen
- Personaleinsatzplanung in Callcentern
- Wie viele Toiletten braucht ein Public Viewing?
- Wann werden wie viele Kassen im Supermarkt geöffnet?
- Jobs in einem Rechner

Typische Fragestellungen

- Wie viele Kunden befinden sich zu jedem Zeitpunkt im System?
- Wie oft kollabiert das System?
- Wie lange ist ein Kunde im Schnitt im System?

Neu Seelands Toilettenregeln

Neuseeland legt sogar die Anzahl der Toiletten über Warteschlangentheorie fest!



Formalisierung der Einzelfälle

- Aus den Anwendungsbeispielen lassen sich gemeinsame Parameter der Probleme destillieren.

  1. Die Warteschlangen entstehen aus einem Ankunftsstrom.
  2. Die "Kunden" (Autos, ...) kommen mit zufälligen (oder deterministischen) Zeitabständen an, um eine bestimmte Leistung abzurufen.
  3. Dann werden sie durch einen oder mehrere Bediener (server) nacheinander bedient. Auch diese Bedienung erzeugt einen Strom von abgearbeiteten Kunden.
  4. Die einzelnen Bedienungszeiten selbst sind wieder zufällig oder deterministisch.

- Aus den Eigenschaften des Ankunfts- und des Bearbeitungsstroms ergeben sich dann Größen für die mittlere Bearbeitungszeit, mittlere Wartezeit u.ä.

Mögliche Verfeinerungen des Modells

- Die Verteilungen der Ankunftszeiten und der Bedienzeiten können festgelegt werden.

- Es ist eine wichtige Unsymmetrie zwischen Ankünften und Bedienungen festzuhalten: Wenn keine Kunden da sind, wird niemand bedient! Diese Unsymmetrie hat gewichtige Auswirkungen auf den Ausgangsstrom!

- Die Systemkapazität kann eine Rolle spielen. (Fassungsvermögen eines Warteraums)

- Ein voller Warteraum führt zum Abweisen von Kunden oder ein fast voller Raum dazu, dass Neuankömlinge keine Lust haben sich anzustellen.

- Die Anzahl von (unabhängigen) Ankunftsströmen und von Bedienern kann variiert werden.

- Das Regime der Bedienung kann festgelegt werden. Übliche Regimes sind
  - FIFO First in first out,
  - LIFO Last in first out,
  - SIRO Service in random order,
  - sogenannte Prioritätsschemata. (Unix Jobkontrolle).

- Es kann sein, dass mehrere Stationen hintereinander durchlaufen werden müssen (Laufzettel!, auch wiederholt), dann bekommt man sogenannte Warteschlangennetze.

- Die Kunden können verschieden geduldig sein, sie können versuchen sich durch Schlangenwechsel vorzukämpfen etc.

- Nur die einfachsten Fälle sind analytisch zu lösen, im Allgemeinen nimmt man Simulationen zur Hilfe!

Deterministische Netze

- Der einfachste Fall. Kein Zufall!
- Kunden kommen mit einer konstanten Rate r (pro Zeiteinheit) an, der Abstand zwischen zwei Ankünften ist folglich $\frac{1}{r}$.
- Die Bedienung erfolgt mit einer Rate s, also ist die Bedienzeit $\frac{1}{s}$.
- Beginnend mit einem leeren System sind also zum Zeitpunkt t

$$N(t) = |\frac{t}{1/r}| - \lfloor\frac{t-1/r}{1/s}\rfloor = \lfloor rt\rfloor - |st - \frac{s}{r}|$$

  Anfragen in der Warteschlange.

- Es lassen sich direkt drei mögliche Fälle ablesen:

  1. Wenn $r > s$, dann wächst die Warteschlange mit t gegen unendlich.
  2. Wenn $r = s$, dann ist die Schlange immer genau eine Anfrage lang. Jeder wird sofort bedient.
  3. $r < s$, dann oszilliert die Länge zwischen 0 und 1.

- Achtung: Die Wartezeit geht in diesen Betrachtungen immer bis zum Ende der Bedienung!

Standardbezeichung von Warteschlangen

- Für Warteschlangen hat sich eine "Kurzbezeichnung" durchgesetzt, die die wichtigesten Fälle mit fünf Kennziffern beschreibt.

- Das Quintupel
  A (Vtg Ankünfte)/ B (Vtg Service) / X (Zahl Bedienungen / Y (Kapazität) / Z (Warteschlangenregime)
  beschreibt eine Warteschlange.
  Dabei hat sich eingebürgert,

  – dass für die Verteilungen gilt: D steht für deterministisch, M für Poissonvtg, G allgemeine Vtg; evtl. müssen noch Parameter spezifiziert werden.
  – das X, Z natürliche Zahlen oder unendlich sind: $1, 2, \ldots, \infty$,
  – dass das Warteschlangenregime W angegeben wird.

- Beispiele für Warteschlangen sind:
  - D/D/1/∞/FIFO (deterministisches System),
  - M/M/1/∞/FIFO (auch kurz M/M/1).

Little's Theorem

- Little hat eine verteilungsunabhängige Aussage über die Anzahl der Anfragen im System herleiten können.

- **Little's Theorem:** Betrachtet man eine G/G/1 Warteschlange in einem Gleichgewichtszustand (*steady state*). Bezeichne nun L die Anzahl der Anfragen im System, λ die mittlere Ankunftsrate und W die erwartete Aufenthaltszeit im System, dann gilt:

$$L = \lambda W.$$

- Beweis: Eilon 1969.

Wartezeitverteilungen

- Die Prozesse der Ankünfte und der Abgänge des Systems bilden einen Geburts- und Todesprozess. Insbesondere handelt es sich beim Ankunftsstrom oft um einen sog. Poissonprozess.

- Der Name Poissonprozess rührt daher, dass sich die Poissonverteilung unter ganz natürlichen Annahmen ergibt.

- Sei $N(t)$ die Anzahl von Ereignissen, die in einem Zeit-Intervall t eintreten, z.B ankommende Kunden.

- Sei dann $P_n(t) = Prob(N(t) = n)$ die Wahrscheinlichkeitsverteilung für $N(t)$.

- Setzt man zusätzlich voraus, dass Ereignisse unabhängig sind, und dass die Wahrscheinlichkeiten für Ereignisse proportional zur Intervalllänge sind, dann gilt:

$$P_n(t) = e^{-\lambda t}\frac{(\lambda t)^n}{n!}, \quad n = 1, 2, \ldots.$$

  λ heißt auch die Rate oder Intensität des Prozesses. $P_n(t)$ ist bekanntlich die Dichte einer Poissonverteilung.

- Bekanntermaßen gilt, dass die Zeiten zwischen zwei Ereignissen Exponentialverteilt sind mit Mittelwert $\frac{1}{\lambda}$.

- Die Verteilung der Summe von n unabhängigen ZV $\sim Exp(\lambda)$ heißt Erlang-Verteilung mit Parametern n und λ, Erl(n, λ).

- Es gilt die interessante Beziehung

$$F_{Erlang}(n+1) + F_{Poisson}(n) = 1.$$

- Zufallsvariablen aus der Erlangverteilung können aus Gleichverteilungen erzeugt werden.
  Seien $U_1, \ldots U_n$ i.i.d $U_{[0,1]}$, dann gilt

$$-\frac{1}{\lambda}\mathsf{ln}\left(\prod_1^n u_i\right) \sim \mathsf{Erl}(\lambda, n).$$

Simulation der Erlang Verteilung

- Implementieren Sie zwei Zufallsgeneratoren `erlang1()` und `erlang2()` für die Erlang-Verteilung. λ und n sowie die Zahl der zu erzeugenden Zufallszahlen sollen übergeben werden können.

- Einmal sollen die Zufallszahlen aus der Gleichverteilung, das andere Mal aus der Exponentialverteilung gebildet werden.

- Vergleichen Sie für den Stichprobenumfang 1000 die empirischen Verteilungsfunktionen für beide Verfahren in einem geeigneten Q-Q-Plot!

Simulation des Ankunftsprozesses

- Der Ankunftsprozess $N(t)$ genüge einer Poissonverteilung mit Parameter $\lambda = 2$.

- Erzeugen Sie einen Vektor, der die Ankunftszeiten der ersten 100 Kunden enthält.

- Der Bearbeitungsprozess $B(t)$ genüge einer Poissonverteilung mit Parameter $\lambda = 3$.

- Erzeugen Sie einen Vektor der ersten 100 Bearbeitungszeiten.

- Können sie daraus den die Austrittszeitpunkte der $L(t)$ für die ersten 100 Kunden ableiten? Angenommen sei, dass der Prozess zum Zeitpunkt 0 leer mit dem Warten auf den ersten Kunden startet.