



Softwareunterstützung und R-Skripte

für ein Monitoring



J. Kieschke, V. Firus

Registerstelle des EKN, Oldenburg

Erarbeitete prototypische R-Skripte

Fallzahlsimulation

Ziel: die beobachtete Fallzahl spezifischer Diagnosen und Zeiträume auf Landesebene gewichtet nach Geschlecht und Alter zufällig auf die regionalen Beobachtungseinheiten zu verteilen

Benötigt (für m+w):

- 1) csv-Tabelle mit Besetzung der verschiedenen Altersklassen für die einzelnen Regionen („pop“)
- 2) csv-Tabelle mit den Fallzahlen je Altersgruppe in Summe für alle Regionen („cases“)

```

Unbenannt - Editor
Datei Bearbeiten Format Ansicht ?

# Szenario:
# Gegeben Bevölkerung einzelner Regionen je Altersgruppe in einem Jahr
# und Fallzahlen je Altersgruppe in Summe für alle Regionen (für ein best. Geschlecht)

# simuliert wird die Fallzahlverteilung auf einzelne Regionen in einem Jahr (für ein best. Geschlecht)

# Daten:
# pop -Tabelle: 1.Spalte - Bezeichnung der Regionen, 2.Spalte - Bevölkerung der 1.Altersstufe, 3.Spalte - Bev. der 2.Alters
# cases - Tabell: 1. Spalte - einzelne Altersstufen, weitere Spalten - Fallzahlen für best. Jahre, Spaltenbezeichnung = Jahr

# Parameter für die Simulation:
# Altersgruppen:
# 1 für "0-4", 2: "5-9", 3: "10-14", 4:"15-19", 5:"20-24", 6:"25-29", 7:"30-34", 8:"35-39", 9:"40-44"
# 10:"45-49", 11:"50-54", 12:"55-59", 13:"60-64", 14:"65-69", 15:"70-74", 16:"75-79", 17:"80-84", 18:"85+"
# Jahr als string, z.B. "2005"

##### diesen Bereich vor der Ausführung bearbeiten #####

# Daten einlesen, Pfad anpassen
pop <-read.csv("C:\\Users\\...\\Bevölkerung_2006.csv",header=TRUE,sep=";",fill=TRUE,check.names=FALSE,dec=".")
cases <- read.table("C:\\...\\Fallzahlen_M.csv",header=TRUE,sep=";",fill=TRUE,check.names=FALSE,row.names="Altersgruppe")

# Parameter setzen
jahr <- "2006"
anzahlsimulationen <- 10

#####

# Funktionen #
relativ_population <- function(population)
{
  relPopulation <- population$region
  for (agegroup in (1:18) )
  {
    sumAgeGroup <- sum(population[,agegroup+1])
    relAgeGroup <- population[,agegroup+1]/sumAgeGroup
    relPopulation <- cbind(relPopulation,relAgeGroup)
  }
  relPopulation
}

simulation_fallzahl <- function(population,fallzahlen,year,numberofsimulations) # "year" als string: "2005"
{
  relPopulation <- relativ_population(population)

  region <- sort(population$region)
  caseDistribution <- region
  L <- length(region)

  for(i in (1:numberofsimulations) )
  {
    matrixAgeGroups <- region
    for (agegroup in (1:18) )
  }
}

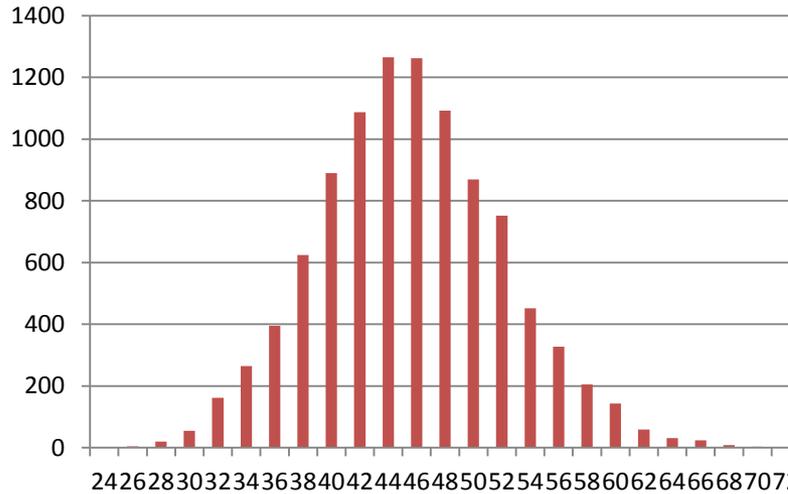
```

Fallzahlsimulation

Fallzahl	erwartet						10.000 Simulationen				
M2008	2008	Sim1	Sim2	Sim3	Sim1000 0	Mittelwert	Min	Max	max SIR	
GKZ1	22,59	26	18	17	22	22,60	8	44	1,95	
GKZ2	10,78	9	8	11	6	10,72	0	27	2,50	
GKZ3	12,52	5	12	14	9	12,48	1	28	2,24	
GKZ4	4,16	1	4	6	5	4,16	0	14	3,37	
GKZ5	0,89	2	0	0	1	0,88	0	6	6,74	
GKZ6	1,21	1	2	1	0	1,20	0	7	5,79	
GKZ7	0,93	0	3	0	0	0,92	0	6	6,45	
GKZ8	1,36	1	0	2	2	1,36	0	8	5,88	
GKZ9	0,99	4	1	0	1	0,98	0	7	7,07	
GKZ10	1,24	0	0	2	3	1,23	0	8	6,45	
GKZ11	1,74	0	1	1	3	1,73	0	8	4,60	
GKZ12	2,1	4	3	1	1	2,11	0	11	5,24	
GKZ13	1,18	1	0	2	1	1,18	0	8	6,78	
GKZ14	0,7	0	1	0	2	0,69	0	6	8,57	
GKZ15	1,42	3	2	1	1	1,43	0	8	5,63	
.....											
GKZ48	45,02	65	41	46	42	45,09	24	71	1,58	
GKZ49	3,27	0	6	2	4	3,30	0	13	3,98	
.....											

Fallzahlsimulation

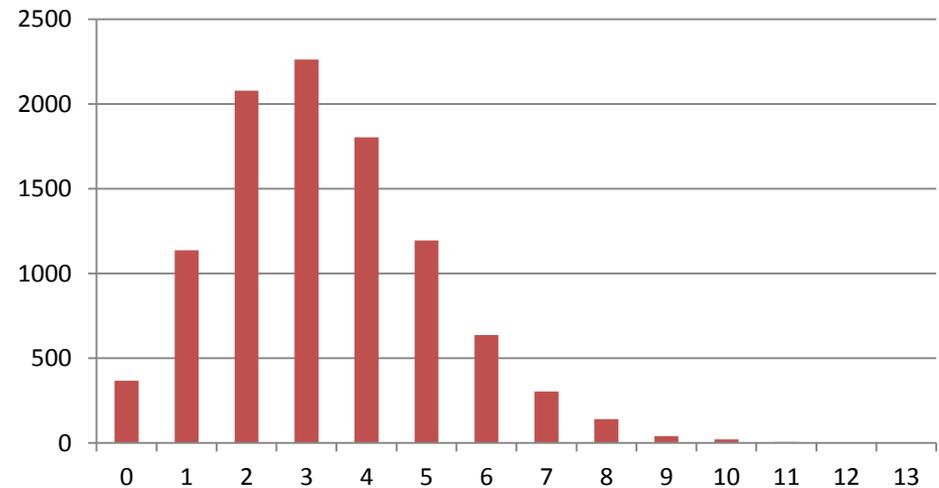
Anzahl Sim - erwartet 45 Fälle



Fallzahl	Anzahl Sim
0	367
1	1137
2	2080
3	2262
4	1803
5	1196
6	638
7	305
8	142
9	40
10	21
11	7
12	1
13	1

Fallzahl	Anzahl Sim
24	1
26	5
28	20
30	55
32	162
34	265
36	395
38	625
40	890
42	1087
44	1265
46	1262
48	1093
50	869
52	752
54	452
56	327
58	205
60	143
62	59
64	31
66	24
68	9
70	3
72	1

Anzahl Sim - erwartet 4 Fälle



Fallzahlsimulation bei diskreter Verteilung

Unterschiedliche Wahrscheinlichkeiten der Gemeinden, für signifikant erhöhte Werte

Simulation: Es wurden jeweils für die 389 Regionalen Beobachtungseinheiten in Niedersachsen für die Jahre 2003-2007 Simulationen für die Verteilung von Harnblasenkrebs bei Männern durchgeführt.

Für das Verhältnis der beobachteten (=simulierten) zur erwarteten Fallzahl wurde der p-Wert berechnet.

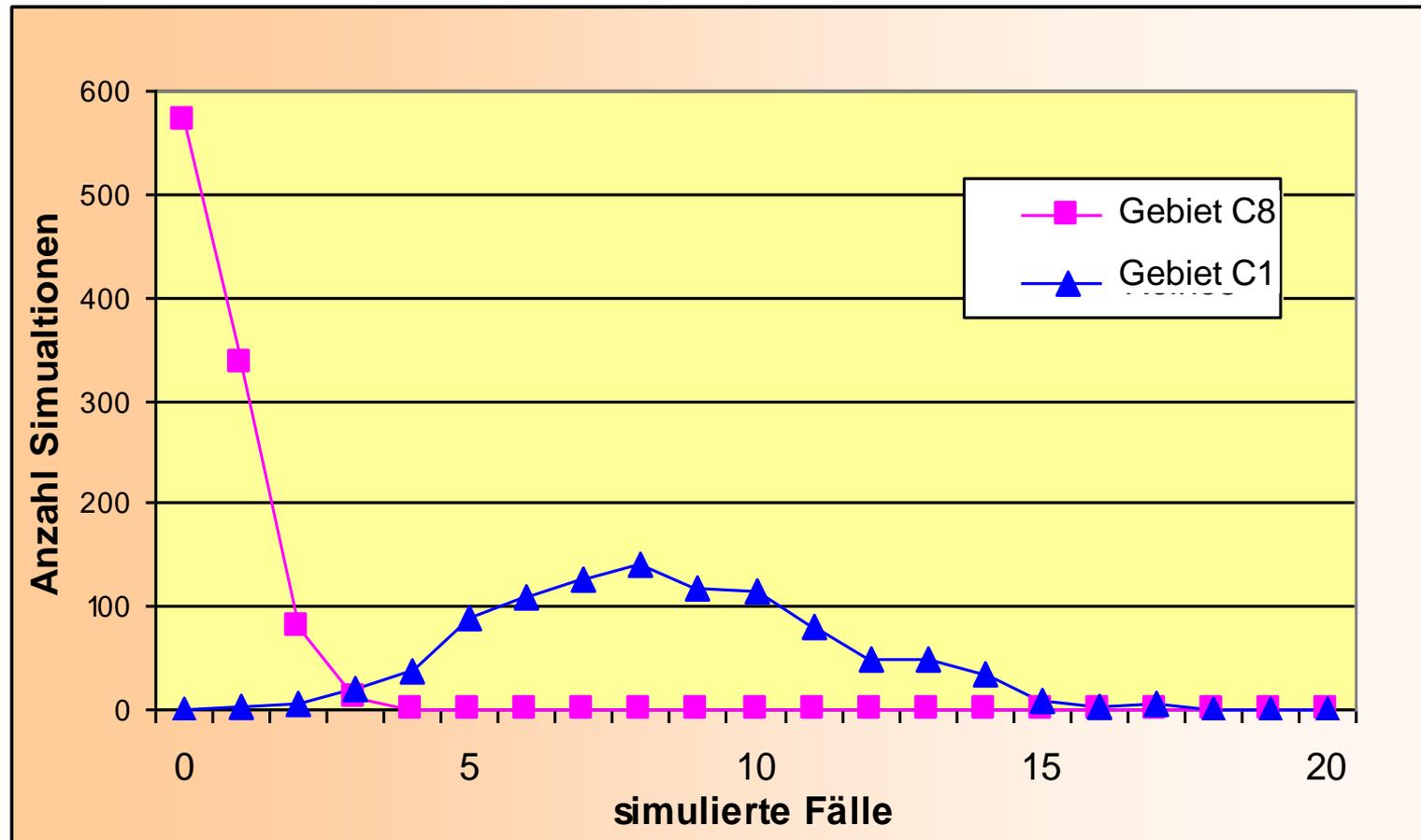
Ergebnis: bei der Poisson-Verteilung als diskreter Verteilung haben bei Anwendung starrer Signifikanzniveaus Gemeinden in Abhängigkeit ihrer erwarteten Fallzahl eine unterschiedliche Wahrscheinlichkeit, zufällig p-Werte $< 0,05$ zu erreichen.

Fallzahlsimulation bei diskreter Verteilung

für $p < 0,05$ mindestens erforderliche Fallzahl	Regionale Beobachtungseinheiten mit relativ niedrigerer erwarteter bei gleich hoher erforderlicher Fallzahl			Regionale Beobachtungseinheiten mit relativ höherer erwarteter bei gleich hoher erforderlicher Fallzahl		
	erwartete Fälle	Anzahl ReBe	Mittelwert Häufigkeit $p < 0,5$ bei 100 Simulationen	erwartete Fälle*	Anzahl ReBe	Mittelwert Häufigkeit $p < 0,5$ bei 100 Simulationen
4 Fälle	0,8-1,1	3	1,7	1,1-1,4	5	3,0
5 Fälle	1,4-1,7	28	2,0	1,7-2,0	119	4,2
6 Fälle	2,0-2,3	209	2,3	2,3-2,6	186	3,8
7 Fälle	2,6-3,0	217	2,2	3,0-3,3	190	4,0
8 Fälle	3,3-3,6	157	2,4	3,6-4,0	107	3,9
9 Fälle	4,0-4,3	95	2,4	4,3-4,7	59	4,1
10 Fälle	4,7-5,1	44	2,6	5,1-5,4	59	4,4
11 Fälle	5,4-5,8	67	2,4	5,8-6,2	36	3,5
12 Fälle	6,2-6,5	22	2,6	6,5-6,9	25	3,0
13 Fälle	6,9-7,3	36	3,0	7,3-7,7	25	4,0
14 Fälle	7,7-8,1	27	3,5	8,1-8,5	27	4,0
15 Fälle	8,5-8,9	24	3,1	8,9-9,2	21	4,8
16 Fälle	9,2-9,6	15	3,1	9,6-10,0	18	4,7
17 Fälle	10,0-10,4	23	3,1	10,4-10,8	16	5,0
18 Fälle	10,8-11,2	30	3,4	11,2-11,6	16	4,9
19 Fälle	11,6-12,0	13	3,5	12,0-12,4	11	4,7
20 Fälle	12,4-12,9	11	3,1	12,9-13,3	18	4,4
21 Fälle	13,3-13,7	15	3,5	13,7-14,1	18	3,5
22 Fälle	14,1-14,5	17	3,6	14,5-14,9	14	4,9
23 Fälle	14,9-15,3	10	3,4	15,3-15,7	14	4,9
24 Fälle	15,7-16,1	13	4,2	16,1-16,5	14	4,4
25 Fälle	16,5-17,0	11	3,5	17,0-17,4	15	4,3
26 Fälle	17,4-17,8	8	5,5	17,8-18,2	3	3,3
27 Fälle	18,2-18,6	4	3,3	18,6-19,1	11	4,9
28 Fälle	19,1-19,5	5	4,2	19,5-19,9	9	6,2
29 Fälle	19,9-20,3	7	2,3	20,3-20,7	7	3,6
Gesamt		1111	3,5		1043	4,6

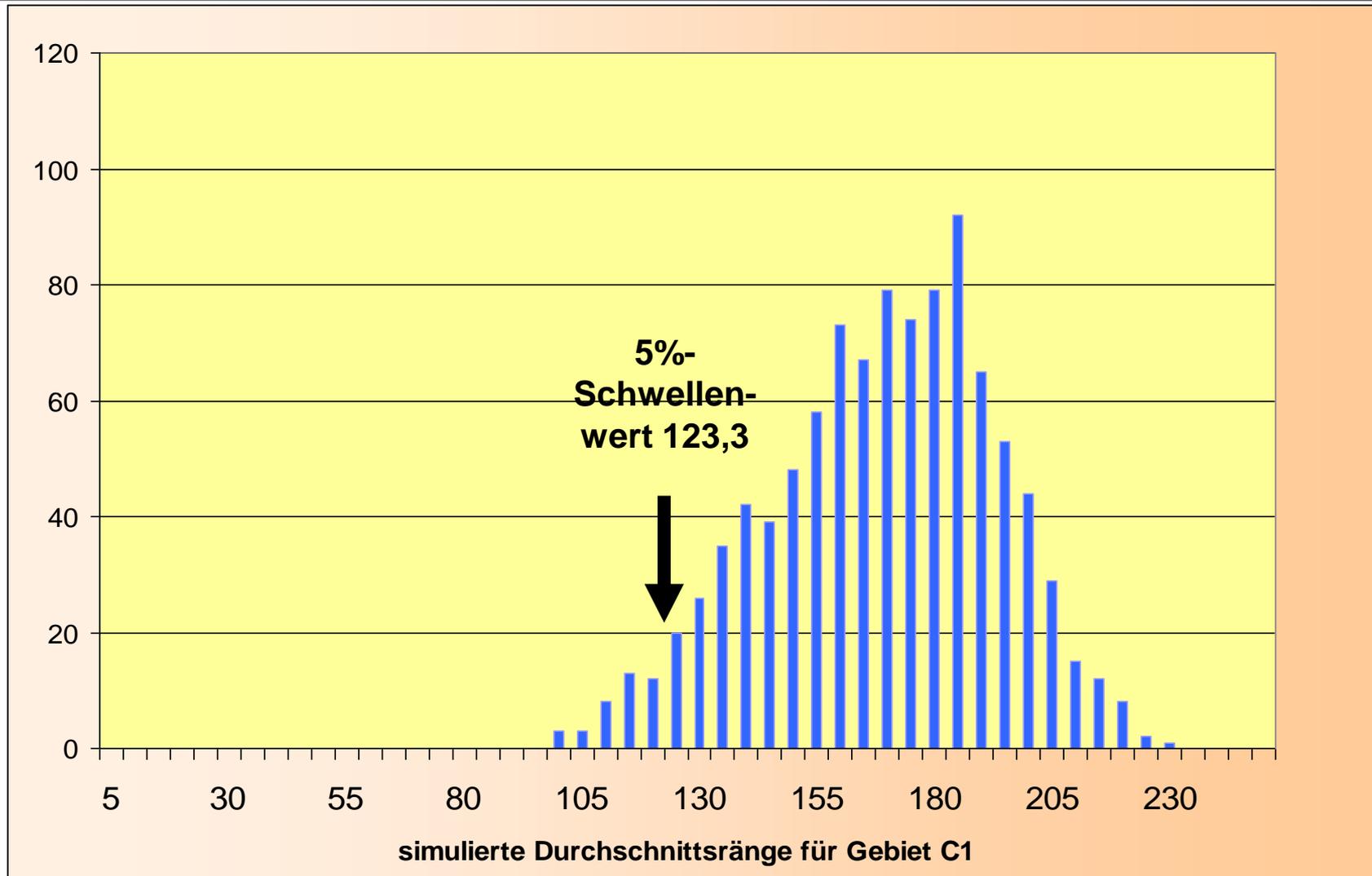
Bei 100 Sim
durchschnittlich
beobachtete Anzahl
an p-Werten $< 0,05$
in Abhängigkeit von
erforderlicher und
erwarteter Fallzahl

Fallzahlsimulation - Konstanzmethode



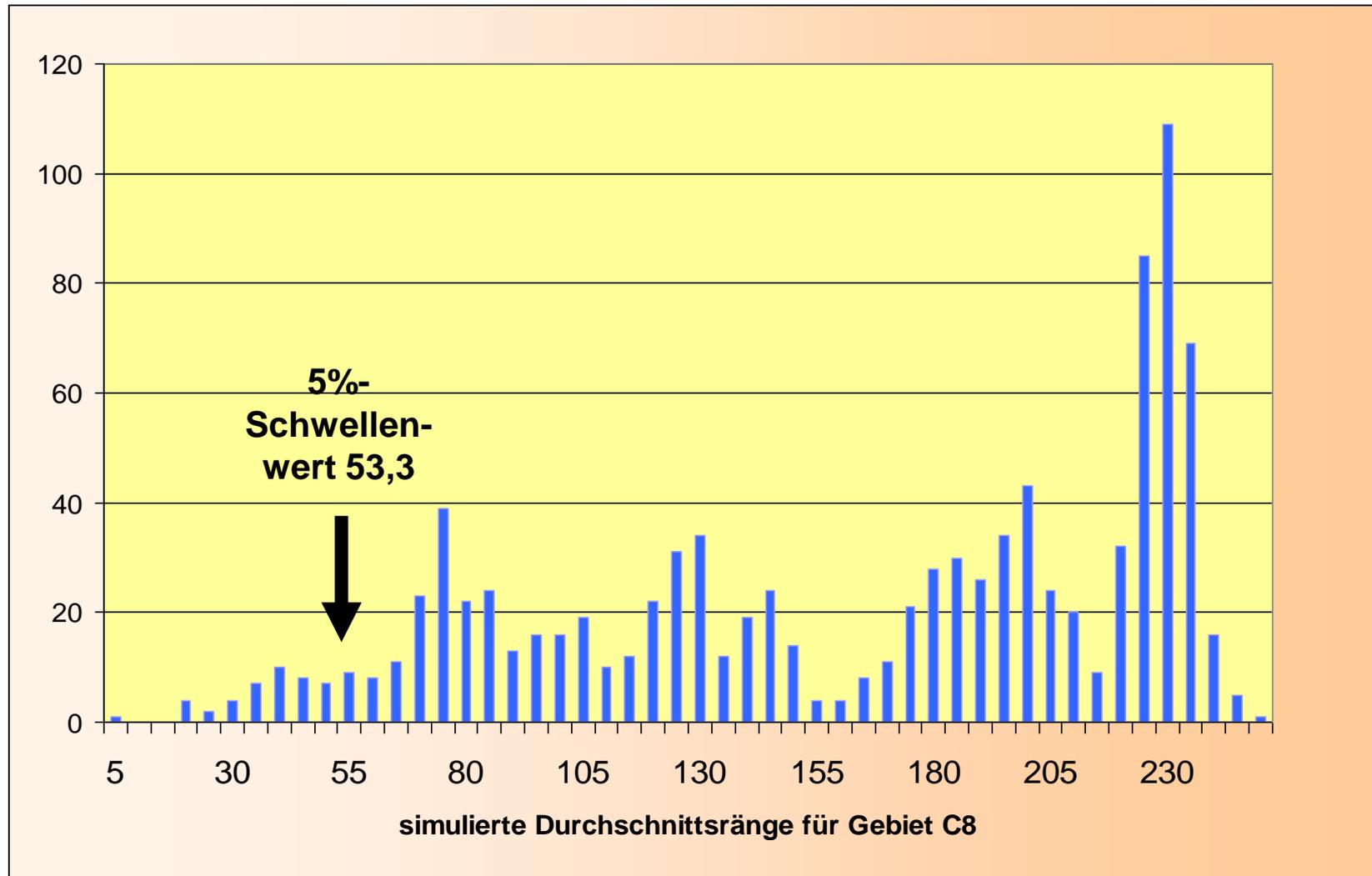
Verteilung der simulierten Fälle 2003 für die Gebiete C1 (Bezugsbev. ca. 75.000) und C8 (Bezugsbev. ca. 5.000)

Fallzahlsimulation - Konstanzmethode



Häufigkeitsverteilung der simulierten Durchschnittsränge für Gebiet C1 (Bezugsbevölkerung ca. 75.000)

Fallzahlsimulation - Konstanzmethode



Häufigkeitsverteilung der simulierten Durchschnittsränge für Gebiet C8 (Bezugsbevölkerung ca. 5.000)

Post-Alarm-Analyse nach Chen

Chen R, Connelly RR, Mantel N: Analysing Post-Alarm Data in a Monitoring-System in Order to Accept or Reject the Alarm, Statistics in Medicine, Vol. 12, 1807-1812 (1993)

Basiert auf dem Zeitintervall der folgenden 5 post-Alarmfällen. Die beobachteten Zeitintervalle dividiert durch ihre erwarteten Werte werden als "relative Zeitintervalle" bezeichnet. Die Methode basiert auf dem Median und Mittelwert der fünf relativen Intervalle.

In beiden Techniken vergleicht der statistische Test die beiden Referenzwerte t_1 und t_2 . Der Alarm ist bestätigt, wenn der Wert kleiner als t_1 ist und wird abgelehnt, wenn er oberhalb von t_2 liegt und eine Beurteilung wird vorbehalten, wenn der Wert dazwischen liegt.

Post-Alarm-Analyse nach Chen

Die Bestimmung der Referenzwerte t_1 und t_2 beruht auf der Annahme, dass die Zeitintervalle eine exponentielle Verteilung haben.

Referenzwert t_1 wird kalkuliert unter Berücksichtigung des Anteils der Falschalarme, t_2 wird unter Beachtung der Power des Tests bestimmt.

Post-Alarm-Analyse nach Chen

Beispiel Mittelwert-basierte Methode:

Die Bestimmung von t_1 und t_2 bei der Mittelwertmethode beruht auf der Annahme, dass 10 Zeitangaben der relativen Intervalle eine Chi-Quadrat-Verteilung mit 10 Freiheiten haben. Dann ist die 0,25 Wahrscheinlichkeit eines falsch positiven Alarms

$$t_1 = \chi^2_{10; 0,75} / 10 = 0,6737,$$

($\chi^2_{10; 0,75}$ ist der kritische Chi-Quadrat-Wert für 10 Freiheiten und einer Wahrscheinlichkeit von 75 % einen falschen Alarm nicht zu bestätigen.)

Für die 0,025 Wahrscheinlichkeit einen echten Alarm bei einer zweifach erhöhten Rate abzulehnen ist

$$t_2 = \chi^2_{10; 0,025} / (10 \times 2) = 1,0242.$$

Post-Alarm-Analyse nach Chen

Linn County, Iowa, 7916 Diagnosen pro Jahr erwartet,
erwartetes Intervall: 1,516 Monate.

Der Alarm ist bestätigt, wenn der Mittelwert der 5 Fälle unterhalb von $0,6737 \times 1,516 = 1,02$ Monate liegt und wird abgelehnt, wenn er größer als $1,0242 \times 1,516 = 1,55$ Monate liegt.

Ein Alarm wurde im Februar 1986 ausgelöst, die 5. Diagnose nach dem Alarm war im Mai 1986, also innerhalb von 4 Monaten, der Mittelwert der Intervalle lag bei 0,8 Monaten und der Alarm wurde bestätigt.

Sequential Probability Ratio Test

```
SPRT_Funktionen - Editor
Datei Bearbeiten Format Ansicht ?
#####
### Funktionen zur Durchführung
### der Sequential Probability Ratio Test Simulationen
#####
# vor dem Aufruf der Simulationen in R dieses Skript einmal ausführen

##### für Simulation eines Testdurchlaufs

# Hilfsfunktion
schritt <- function(xx, periode, logr, rfaktor, lambda0, mu1, tt0, tt1)
{
  teststatistik <- logr*sum(xx) - periode*(rfaktor-1)*lambda0
  if(teststatistik < tt0)
  {
    message("H0 bestätigen")
    return(list(Fallzahlen=xx, SummeFallzahlen=sum(xx), Anzahlbeobachtungsperioden=periode, untergrenze=tt0, oberegrenze=tt1))
  }
  if(teststatistik > tt1)
  {
    message("H0 ablehnen")
    return(list(Fallzahlen=xx, SummeFallzahlen=sum(xx), Anzahlbeobachtungsperioden=periode, untergrenze=tt0, oberegrenze=tt1))
  }
  else
  {
    periode <- periode + 1
    x <- c(xx, rpois(1, mu1))
    schritt(x, periode, logr, rfaktor, lambda0, mu1, tt0, tt1)
  }
}

# Hauptfunktion
SPRTsim <- function(lam0, r, mue, f1, f2)
{
  lnr <- log(r)
  t0 <- log(f2/(1-f1))
  t1 <- log((1-f2)/f1) - (1/3)*lnr

  phase <- 1
  x <- rpois(1, mue)
  schritt(x, phase, lnr, r, lam0, mue, t0, t1)
}

##### Mehrere Testsimulationen durchführen und erwartete Anzahl der Beobachtungsperioden
##### und erwartete Fallzahlen bis zur Entscheidung schätzen

# Hilfsfunktion
simSchritt <- function(xx, periode, logr, rfaktor, lambda0, mu1, tt0, tt1, confCount)
{
  teststatistik <- logr*sum(xx) - periode*(rfaktor-1)*lambda0
  if(teststatistik < tt0)
  {
    confCount <- confCount + 1
    return(list(periode, confCount, sum(xx)))
  }
  if(teststatistik > tt1)
  {
    return(list(periode, confCount, sum(xx)))
  }
  else
  {
    periode <- periode + 1
  }
}
```

... dazu im folgenden Vortrag mehr ...

Vielen Dank für Ihre
Aufmerksamkeit