Frank Bengtson **2013**
frank.bengtson@gmail.com

Nogle anvendelser af programmel R, bl.a. til hypotesetest

R er specielt egnet til statistik og simulering og kan frit installeres på egen pc. R udfører en programlinje ad gangen, viser resultatet straks og er derfor let at eksperimentere med. Man kan tilpasse og indsætte egne data i R programmer, de virker så som alle andre værktøjer. R leverer betydeligt pænere grafik end mange kommercielle produkter, et plot kan gemmes i flere formater f.eks. ***.eps og ***.ps, hvorfra der kan konverteres til andre formater. Simulering giver en rimelig begrundelse for χ^2-fordelingers optræden i χ^2-test. R burde derfor også være egnet til ungdomsuddannelserne. Der findes mange tilgængelige vejledninger på www, f.eks. MORTEN LARSEN og PETER SESTOFT *Noter om R*, men det er ganske få ting der skal til.

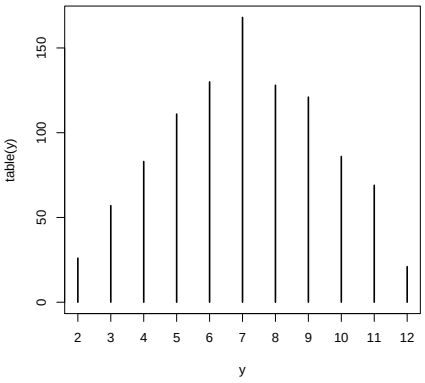
Eks. 1. Sum af to terninger simulering

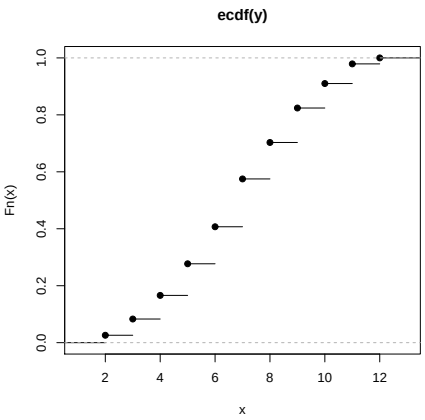
```
script vindue:
#
#minimaludgave - sum af to terninger - simulering
#
x1<-sample.int(6,size=1000,replace=TRUE) # ligefordeling når ikke andet siges
x1
summary(x1)
x2<-sample.int(6,size=1000,replace=TRUE)
y<-x1+x2
y
summary(y)
table(y)
plot(table(y))
plot(ecdf(y))


R Console vindue:

> summary(x1)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   2.000   4.000   3.468   5.000   6.000

> summary(y)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   5.000   7.000   7.026   9.000  12.000
> table(y)
y
  2   3   4   5   6   7   8   9  10  11  12
 28  53  79 120 128 168 146 103  94  56  25
```





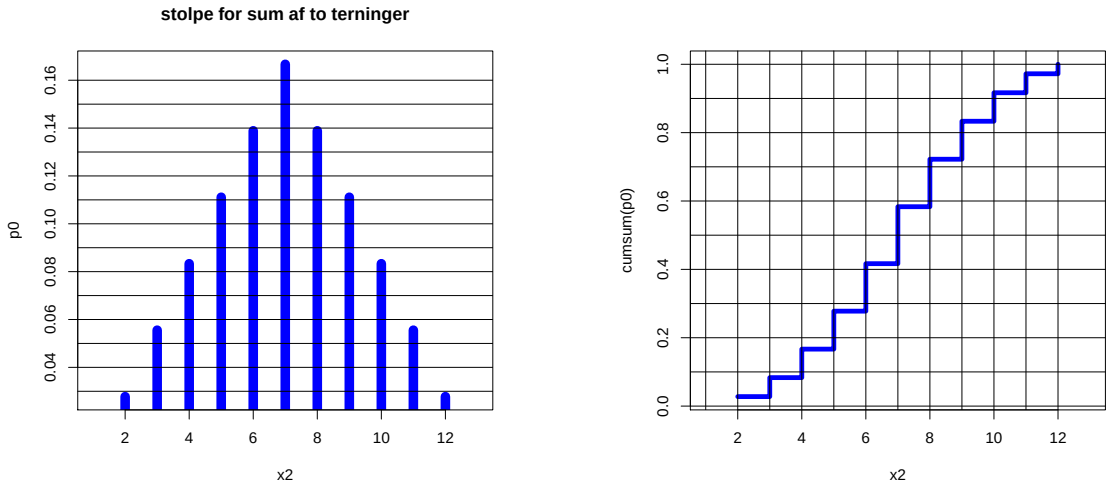
Programlinjerne indskrives i et *script* og gemmes i en fil ***.*R* i en mappe f.eks. *R1*. Programmet fortolker en linje ad gangen. Curseren anbringes på første linje og kommandoen udføres ved at taste *ctrl* og *R*. Resultatet vises i *R Console*, hvilket gør det let at kontrollere om det var det ønskede.

Der kan eksperimenteres med f.eks. $X \cdot X, X_1 \cdot X_2, X_1 + X_2 + X_3 + X_4$. □

Eks. 2. Sum af to terninger sandsynlighedsmodel

```
#
```

```
#sandsynlighedsmodel
#
x2<-2:12
x2
p0<-1:11
p0[7:11]<-5:1
p0<-p0/36
p0
plot(x2,p0,lwd=10,xlim=range(1,13),type="h",col="blue",main="stolpe for sum af to terninger")
for (k in 0:16){abline(h=k*0.01)}
plot(x2,cumsum(p0),type="s",lwd=5,col="blue",xlim=range(1,13))
for (k in 0:10){abline(h=k*0.1)}
for (k in 1:12){abline(v=k)}
```



□

Teststørrelser og simulering

Ved simulering antages hypotesen opfyldt og de indbyggede random funktioner giver passende variation i observationerne i overensstemmelse med forudsætningerne. F.eks. kan 600 terningkast simuleres med phyp $= (p_1, \ldots, p_6) = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$

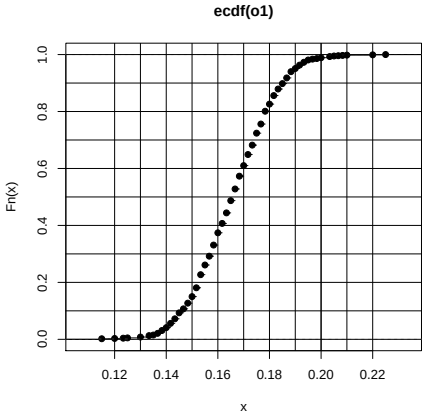
- sample(6,600,replace=TRUE,phyp)
- rmultinom(1,600,phyp)[1:6], rmultinom(n, size, phyp)[1]
- round(runif(600,0,6)+0.5)), ligefordeling, rykker lidt til højre

, hvor obs. er normalfordelte om $(1, 2, 3, 4, 5, 6)$. Med eks. nedenfor er lavet statisik på sand. for obs. 1. Ved at variere antal kast ses hvordan spredningen varierer.

[1] *R: A Language and Environment for Statistical Computing* The rmultinom() algorithm draws binomials X_j from $\mathrm{Bin}(n_j; P_j)$ sequentially, where $n_1 = N$ ($N :=$ size), $P_1 = \pi_1$ (π is prob scaled to sum 1), and for $j \geq 2$, recursively, $n_j = N - \sum_{k=1}^{j-1} X_k$ and $P_j = \pi_j / \left(1 - \sum_{k=1}^{j-1} \pi_k\right)$.

```
# statistik på udfald
o1<-rep(0,1000)
phyp<-rep(1/6,6)
phyp
for (m in 1:1000){
         x<-table(sample(6,600,replace=TRUE,phyp))
         o1[m]<-x[1]/600}
plot(ecdf(o1))
for (k in 0:10){abline(h=k*0.1)}
for (k in 1:10){abline(v=0.01*k+0.12)}
summary(o1)
var(o1)
sd(o1)

> summary(o1)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.1150  0.1550  0.1667  0.1663  0.1767  0.2250
> var(o1)
[1] 0.0002312518
> sd(o1)
[1] 0.01520697
```



Ved simulationen bør middelværdien for p_1 være $1/6$ og eks. giver mean=0.1663, og ved $n = 600$ er den forventede varians på hypotesens p_1

$$p_1(1-p_1)/n = \tfrac{1}{6}(1-\tfrac{1}{6})/600 = 5/36/600 = 0.00023$$

og spredningen $\sigma = \sqrt{0.00023} = 0.0152$ svarer til observeret $\mathrm{sd}(o1) = 0.0152$.

χ^2- teststørrelsen eller Pearsonteststørrelsen beregnes som

$$K(x) = \sum \frac{(\text{obs. antal} - \text{forv. antal})^2}{\text{forv. antal}} = \frac{(x_1 - np_1)^2}{np_1} + \cdots + \frac{(x_m - np_m)^2}{np_m},$$

og testsandsynligheden

$$\epsilon(x) \approx 1 - F_{\chi^2_{df}}(K(x)).$$

Små værdier af $K(x)$ svarer til x nær den forventede. Simulationen giver indtryk af tæthed og fordeling af K(x) under hypotesen.

Eks. 3. En terning, tæthed og fordeling af K(x)

Kastes 60 gange med en alm. terning er forventet $np = (10, 10, 10, 10, 10, 10)$. Ved 60 kast observeres f.eks. $x = (x_1, \ldots, x_6) = (16, 12, 9, 9, 7, 7)$. I eks. her er

$$K(x) = \frac{(16-10)^2}{10} + \frac{(12-10)^2}{10} + \frac{(9-10)^2}{10} + \frac{(9-10)^2}{10} + \frac{(7-10)^2}{10} + \frac{(7-10)^2}{10} = 6.$$

De grønne/kraftige grafer er tæthed fra histogram og ecdf (empirisk cumuleret tæthed) fra observationerne og de blå/tynde er tæthed og fordeling for χ^2-fordelingen med 5 frihedsgrader. Af nederste fig. aflæses, at de 5% der ligger længst fra den forventede har $K(x)$ større end 11, og de 2.5% der ligger længst fra har $K(x)$ større end 13. som i eks. her er - kan aflæses af nederste graf

$$\epsilon((16,12,9,9,7,7)) \approx 1 - F_{\chi^2_5}(K((16,12,9,9,7,7))) = 1 - F_{\chi^2_5}(6) = 0.31,$$

Med $x = (6, 16, 7, 16, 6, 9)$ er

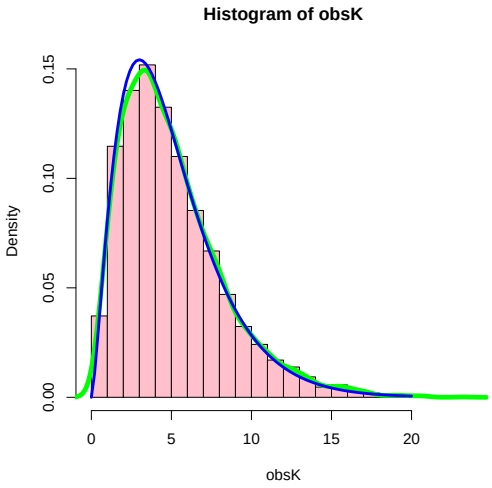
$$\epsilon(x) = \epsilon((6,16,7,16,6,9)) \approx 1 - F_{\chi^2_5}(K((6,16,7,16,6,9))) = 1 - F_{\chi^2_5}(11.4) = 0.044.$$

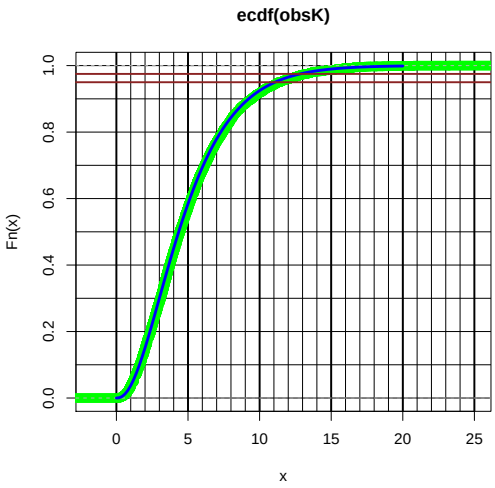
Man kan lave vilkårlige sandsynlighedsvektorer, f.eks. $p = (1,1,1,1,1,2)$ efterfulgt af normering $p = (1,1,1,1,1,2)/\text{sum}(p)$ og en mere skæv med „p<-rep(1:6)“, $p = (1,2,3,4,5,6)$ efterfulgt af „p<-p/sum(p)“. Det observeres at tæthed og fordeling for K(x) ikke ændres.

```
phyp<-rep(1/6,6)
phyp
antalkast<-60
forv<-antalkast*phyp
forv
sum(forv)
K<-function(x){sum((x-forv)^2/forv)}
eps5<-function(x){1-pchisq(K(x),5)}
x[1]<-16;x[2]<-12;x[3]<-9;x[4]<-9;x[5]<-7;x[6]<-7;
#x<-table(sample(6,antalkast,replace=TRUE,phyp))[]  # giver hyppighed
# se de forskellige
for (m in 1:20){
   x<-table(sample(6,antalkast,replace=TRUE,phyp))[]#hyppighed
   #x<-runif(6,6,antalkast)
   #x<-round(antalkast*x/sum(x))
   #x<-table((round(runif(antalkast,0,6)+0.5)))
   #x<-rmultinom(1,antalkast,p1)[1:6]
   print(x)
   print(sum(x))
   print(K(x))}
antaleks<-600
obsK<-rep(0,antaleks)
for (m in 1:antaleks)
    { x<-table(sample(6,antalkast,replace=TRUE,phyp))[1:6]
      #x<-runif(6,6,antalkast)
      #x<-table(round(runif(antalkast,0,6)+0.5))[]
      obsK[m]<-K(x) }
summary(obsK)
#obsK

hist(obsK,freq=FALSE,breaks=20,col="pink")
lines(density(obsK),lwd=5,col="green")
x1<-seq(0,30,by=.1)
y1<-dchisq(x1,df=5) # tæthed
lines(x1,y1,lwd=3,col="blue")

plot(ecdf(obsK),col="green",lwd=10)
for (k in 0:25){abline(v=k)}
for (k in 0:5){abline(v=k*5,lwd=2)}
for (k in 0:10){abline(h=k*0.1)}
y1<-pchisq(x1,5) # fordeling
#y1
lines(x1,y1,lwd=3,col="blue")
abline(h=0.975,col="brown4",lwd=2)
abline(h=0.95,col="brown4",lwd=2)
abline(v=5,col="brown4",lwd=2)
```





□

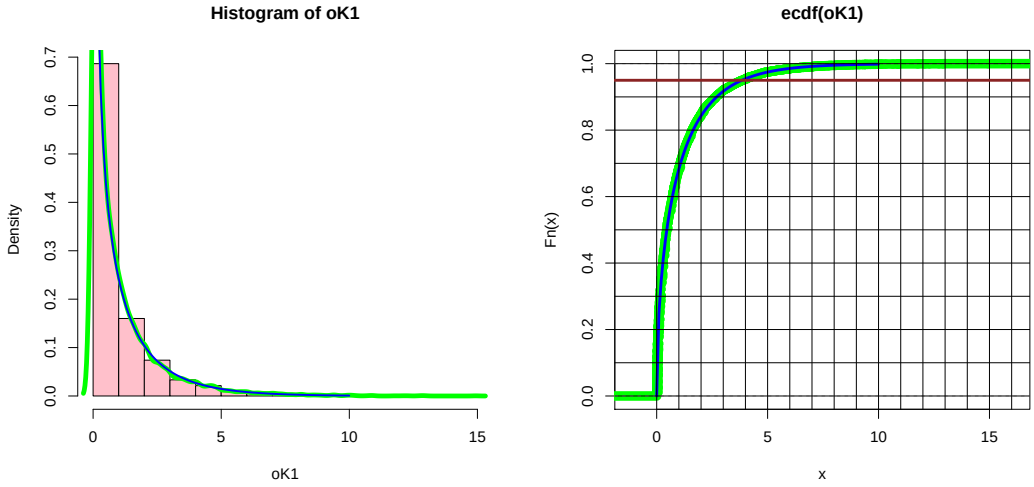
Eks. 4. Antalstabel

Et kendt eks. på en 2x2 tabel. *Er der uafhængighed mellem tøjkøb pr. md. og køn?* Hvis det antages, at forbruget er uafhængigt af køn, må sandsynligheden for søjlerne være $p = (p_1, p_2) = (\frac{158}{360}, \frac{202}{360}) = (0.44, 0.56)$, og de forventede antal kan beregnes og sammenlignes med de observerede antal. Indtastning af data og beregning af teststørrelse med de øverste fem linjer i R prog. nedenfor.

Også her stemmer simulationer pænt med χ^2-fordelingen med 1 frihedsgrad. De grønne/kraftige grafer er tæthed fra histogram og ecdf fra observationerne og de blå/tynde er tæthed og fordeling for χ^2-fordelingen med 1 frihedsgrad. De 5% obs. med største K(x) værdier er større end 4. De opfylder også hypotesen, men anses for at være for ekstreme. Her beregnes K værdien med den mere overskuelige K1(data.table,forv.table)

	tøjkøb kr. pr. md.	mindre end 1500	mere end 1500	
observerede	kvinder	98	102	200
	mænd	60	100	160
		158	202	360
forventede	kvinder	$\frac{158}{360} \cdot 200 = 87.78$	$\frac{202}{360} \cdot 200 = 112.22$	200
	mænd	$\frac{158}{360} \cdot 160 = 70.22$	$\frac{202}{360} \cdot 160 = 89.78$	160

def. nedenfor. K kan beregnes som K(data.table) som det gøres i næste eks.

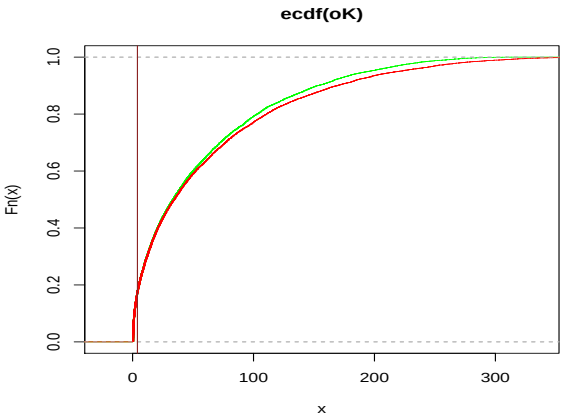


$$
\begin{aligned}
x = \text{data.table} &= \begin{bmatrix} 98 & 102 \\ 60 & 100 \end{bmatrix} \\
K(x) = K\left(\begin{bmatrix} 98 & 102 \\ 60 & 100 \end{bmatrix}\right) & \\
&= K1(\text{data.table}, \text{forv.table}) \\
&= K1\left(\begin{bmatrix} 98 & 102 \\ 60 & 100 \end{bmatrix}, \begin{bmatrix} 87.78 & 112.22 \\ 70.22 & 89.78 \end{bmatrix}\right) = 4.8, \\
\epsilon(x) = \epsilon\left(\begin{bmatrix} 98 & 102 \\ 60 & 100 \end{bmatrix}\right) &\approx 1 - F_{\chi^2_1}(4.8) = 0.029.
\end{aligned}
$$

$F_{\chi^2_1}(4.8)$ kan aflæses af blå/tynde graf.

$$
\begin{aligned}
x = \text{data.table} &= \begin{bmatrix} 83 & 117 \\ 82 & 78 \end{bmatrix} \\
K(x) = K\left(\begin{bmatrix} 83 & 117 \\ 82 & 78 \end{bmatrix}\right) &= 3.4, \\
\epsilon(x) = \epsilon\left(\begin{bmatrix} 83 & 117 \\ 82 & 78 \end{bmatrix}\right) &\approx 1 - F_{\chi^2_1}(3.4) = 0.065.
\end{aligned}
$$

Man kan undersøge svar der er næsten vilkårlige, her vælges kun obs. større end seks. I fig. er den grønne graf K(x) og den røde kvotienttesteststørrelse $-2\log(Q(x))$. K(x) værdierne er ret store, men ca. 18% af de næsten tilfældige svar har K(x)<4.



R Script

```
row1<-rep(0,2)
#row1<-c(98,102)[1:2]    # plejer at virke, giver error på nogle pc
row2<-rep(0,2)
row1[1]<-98; row1[2]<-102
row1                                    # vis første række
row2[1]<-60; row2[2]<-100
row2                                    # vis anden række
rbind(row1,row2)->data.table            # tabel, skema eller matrix
data.table                              # vis det indtastede
chisq.test(data.table)
# chi-i-anden teststørr.
# denne lille programstump leverer resultatet,
# men det viser måske ikke særlig meget
# egen beregning af chisq.test med forventet
ph<-(row1+row2)/sum(data.table)  # under hypotese
ph
forv1<-(row1+row2)*sum(row1)/sum(data.table)   # forventede
forv2<-(row1+row2)*sum(row2)/sum(data.table)
rbind(forv1,forv2) -> forv.table
forv.table              #
K1<-function(data.table,forv.table){sum((data.table-forv.table)^2/forv.table)}
#K1<-function(x,f){sum((x-f)^2/f)}
eps1<-function(x){1-pchisq(x,df=1)}
K1(data.table,forv.table)
eps1(K1(data.table,forv.table))
#
###############################################
# simulering under hypotesen ph
#
antal=10000
oK1<-rep(0,antal)
for (m in 1:antal)
   {  row1<-rmultinom(1,200,prob=ph)[1:2]
      row2<-rmultinom(1,160,prob=ph)[1:2]
      rbind(row1,row2) -> data.table
      forv1<-(row1+row2)*sum(row1)/sum(data.table)   # forventede
      forv2<-(row1+row2)*sum(row2)/sum(data.table)
      rbind(forv1,forv2) -> forv.table
      oK1[m]<-K1(data.table,forv.table)}
#oK1
hist(oK1,freq=FALSE,breaks=20,col="pink")
lines(density(oK1),lwd=2,col="green")
x1<-seq(0,10,by=.1)
y1<-dchisq(x1,df=1) # tæthed
lines(x1,y1,lwd=2,col="blue")

plot(ecdf(oK1),col="green",lwd=10)
y1<-pchisq(x1,df=1) # fordeling
lines(x1,y1,lwd=3,col="blue")
for (k in 0:10){abline(h=k*0.1)}
for (k in 0:30){abline(v=k)}
abline(h=0.95,col="brown4",lwd=3)
```

```
# model med næsten tilfældige svar
# row1 mindst 6 sum 200
row1[1]<-sample(6:194,1) # jævnt fordelt
row1[2]<-200-row1[1]
oR1[m]<-row1[1] # til stat
oR2[m]<-row1[2] # til stat
# row2 mindst 6 sum 160
row2[1]<-sample(6:154,1)
row2[2]<-160-row2[1]
rbind(row1,row2)->data.table

ncol(data.table)
nrow(data.table)
# beregning af K(data.table)
K<-function(data.table){
           forv.table<-data.table
           phyp<-colSums(data.table)/sum(data.table)
           for (m in 1:nrow(data.table)){
           forv.table[m, ]<-rowSums(data.table)[m]*phyp} # forventede row
           K<-sum((data.table-forv.table)^2/forv.table)
           return(K)}
K(data.table)
min(forv.table) # mindste værdi af forventede, skal være større end 5

# kvotienttesteststørrelse
Q<-function(data.table){
Q<-1
for (m in 1:nrow(data.table)) #rækker
{for (n in 1:ncol(data.table))  #søjler
{Q<-Q*(rowSums(data.table)[m]*colSums(data.table)[n]/
 (sum(data.table)*data.table[m,n]))^(data.table[m,n])}}
return(Q)}
```