

Note om Monte Carlo metoden

Kasper K. Berthelsen

Version 1.2 — 25. marts 2014

1 Introduktion

Betegnelsen Monte Carlo dækker over en lang række metoder. Fælles for disse metoder er, at de anvendes til at opnå tilnærmede løsninger vha. (computer) simulation af en eller flere stokastiske variable. Denne note omhandler såkaldt simpel Monte Carlo.

Eksempel: Terningkast

Hvad er det forventede antal øjne ved et kast med en fair terning? Vi starter med en matematisk beskrivelse af en terning. Antag X er en stokastisk variabel, der angiver antallet af øjne ved et kast med en fair terning. Dvs. X kan tage værdierne 1 til 6 med lige stor sandsynlighed. Dvs. $P(X = 1) = P(X = 2) = \dots = P(X = 6) = \frac{1}{6}$. Lad $p(x)$ være den tilsvarende sandsynlighedsfunktion, dvs. $p(x) = P(X = x)$. Den forventede værdi af X er da $\mathbb{E}[X] = \sum_{x=1}^6 xp(x) = \sum_{x=1}^6 x \frac{1}{6} = \frac{1}{6}(1 + 2 + \dots + 6) = 3.5$. Det forventede antal øjne ved et kast med en fair terning er altså 3.5. At det forventede antal øjne ved et kast med en fair terning er 3.5 kan fortolkes som det gennemsnitlige antal øjne ved gentagne kast i det lange løb.

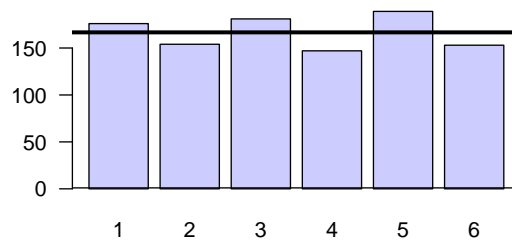
En alternativ og tilnærmet metode er at kaste med en terning et stort antal gange og finde det gennemsnitlige antal øjne i de mange kast. Vi anvender en computer til at udføre de mange terningkast. Man kan nemt få R til at kaste en fair terning 1000 gange:

```
x = sample(size = 1000, x = 1:6,
           prob = c(1, 1, 1, 1, 1, 1)/6, replace = TRUE)
```

Figur 1 viser fordelingen af antal øjne i de 1000 kast. Vi kan udregne gennemsnittet af de 1000 kast:

```
mean(x)

## [1] 3.478
```



Figur 1: Fordeling af antal øjne i 1000 kast med en terning. Den vandrette linje svarer til det forventede antal kast med hhv. 1,2,...,6 øjne i de 1000 kast, Konkret er det forventede antal $\frac{1}{6}1000 \approx 166.66$.

Vi ser at det gennemsnitlige antal øjne i de 1000 kast er 3.478, hvilket hermed er vores estimat af det forventede antal øjne. Estimatet er tæt på det korrekte svar 3.5. Vi vil senere se på, hvordan vi sikre os, at estimatet er “tilstrækkeligt” præcist.

Gentager vi eksperimentet får vi et andet resultat:

```
x = sample(size = 1000, x = 1:6,
            prob = c(1, 1, 1, 1, 1, 1)/6, replace = TRUE)
mean(x)
## [1] 3.552
```

Det gennemsnitlige antal øjne i dette eksperiment er 3.552, hvilket ikke er det samme som i første eksperiment. Dette skyldes at der er anvendt 1000 nye (og uafhængige) terningkast. Pga. det stokastiske (dvs. tilfældige) element i metoden (de mange tilfældige terningkast) får man i praksis aldrig samme svar.

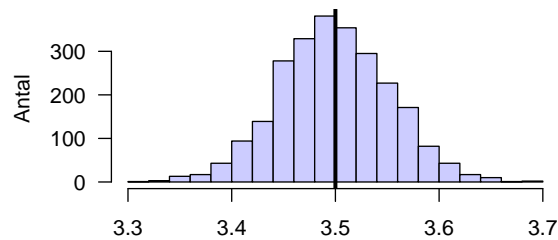
Lad os et øjeblik undersøge variation i svarene ovenfor nærmere. Vi bemærker først at ovenstående R kode kan skrives mere kompakt, så eksperimentet kan opnås i “et hug”:

```
mean(sample(size = 1000, x = 1:6, replace = TRUE))
## [1] 3.571
```

Bemærk endnu engang af vi får en ny middelværdi. Vha. kommandoen `replicate` kan vi gentage eksperimentet mange gange, fx. 2500 gange:

```
middel = replicate(2500, mean(sample(size = 1000,
                                     x = 1:6, replace = TRUE)))
```

Vektoren `middel` har længde 2500 og indeholder gennemsnittene fra de 2500 eksperimenter, hvor hver gennemsnit er baseret på 1000 terningkast. Hvert gennemsnit er et estimat af $E[X]$. Figur 2 viser et histogram over de 2500 estimater.



Figur 2: Histogram over det gennemsnitlige antal øjne i 1000 kast i de 2500 eksperimenter.

Det ses at de fleste estimerer ligger mellem 3.35 og 3.65 med den sande værdi 3.5 lige i midten. Med andre ord, så ligger langt de fleste estimerer ikke længere end 0.15 fra den sande forventede værdi.

Ovenstående er et simpelt eksempel på anvendelse af Monte Carlo metoden, og derfor ofte refereret til som *simpel Monte Carlo*. I det følgende går vi mere i detaljer med simpel Monte Carlo som en teknik til at estimere forventede værdier. Ovenstående eksempel er mest af akademisk interesse, da det er muligt at udregne den sande forventede værdi. I de følgende afsnit vil vi se på eksempler, hvor der er mere vanskeligt — eller helt umuligt at finde en simpel formel til udregning af den forventede værdi. Vi vil også se, at Monte Carlo metoden, måske lidt overraskende, også kan anvendes til at estimere sandsynligheder.

2 Monte Carlo — estimation af middelværdi

Motivationen for Monte Carlo metoden er ganske simpel: Vi har en stokastisk variabel X og ønsker at bestemme den forventede værdi $\mathbb{E}[X]$ af X . I andre tilfælde er vi interesseret i at finde den forventede værdi af en funktion af X , fx. hvad er den forventede værdi af X^2 ? I nogle tilfælde er det muligt at udregne den forventede værdi af interesse. I afsnit 1 ønskede vi at bestemme det forventede antal øjne i et kast med en fair terning. I tilfældet med antal øjne i et kast med en fair terning så vi, at det er (simpelt) at udregne den forventede værdi og det er dermed ikke nødvendigt at anvende Monte Carlo metoden.

I andre tilfælde er det mere kompliceret eller helt umuligt at udregne den forventede værdi. I disse tilfælde kan Monte Carlo metoden anvendes til at *estimere* den forventede værdi. Som vi så i afsnit 1 er ideen bag Monte Carlo metoden ganske simpel: Vi simulerer et antal realisationer x_1, \dots, x_n af den stokastiske variabel X og udregner gennemsnittet af disse. Resultatet er et estimat af den forventede værdi $\mathbb{E}[X]$. I afsnit 1 svarer en simulation af en realisation af X til et enkelt terningkast.

I det følgende antager vi, at X er en kontinuert stokastisk variabel, men alle resultater gælder også for diskrete stokastiske variable. Mere specifikt, antag at X er en kontinuert stokastisk variabel med tæthedsfunktion $f(x)$. Dvs. at

sandsynligheden for at X falder mellem a og b ($a \leq b$) er

$$P(a \leq X \leq b) = \int_a^b f(x) dx. \quad (1)$$

I dette tilfælde udregnes den forventede værdi af $h(X)$ som

$$\mathbb{E}[h(X)] = \int_{-\infty}^{\infty} h(x) f(x) dx. \quad (2)$$

Vi er interesserede i eksempler, hvor det er umuligt eller bare meget svært at udregne integralet i (2). For senere at kunne anvende Monte Carlo metoden er det også nødvendigt at antage, at det er muligt at simulere X .

Eksempel: log-normalfordelt stokastisk variabel

Antag X er standard normalfordelt, dvs. $X \sim \mathcal{N}(0, 1)$. Vi er interesserede i at finde middelværdien af $\exp(X)$, hvor \exp betegner eksponentialfunktionen, som også skrives e^x . I dette tilfælde er middelværdien $\mathbb{E}[\exp(X)]$ givet ved

$$\mathbb{E}[\exp(X)] = \int_{-\infty}^{\infty} \exp(x) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right) dx.$$

Dette integrale kan faktisk udregnes og det er dermed muligt at bestemme den (sande) forventede værdi. Vi vælger dog en Monte Carlo tilgang.

Egenskaber ved simpel Monte Carlo

Lad μ betegne den ukendte middelværdi $\mathbb{E}[h(X)]$, og lad σ^2 betegne variansen $\text{Var}[h(X)]$, hvor vi antager at $\sigma^2 < \infty$, dvs. at variansen af $h(X)$ er endelig.

Antag at X_1, \dots, X_n er uafhængige stokastiske variable med samme fordeling som X , dvs. $\mathbb{E}[h(X_i)] = \mu$ og $\text{Var}[h(X)] = \sigma^2$ for $i = 1, \dots, n$. Definer gennemsnittet

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n h(X_i) = \frac{1}{n} (h(X_1) + h(X_2) + \dots + h(X_n)).$$

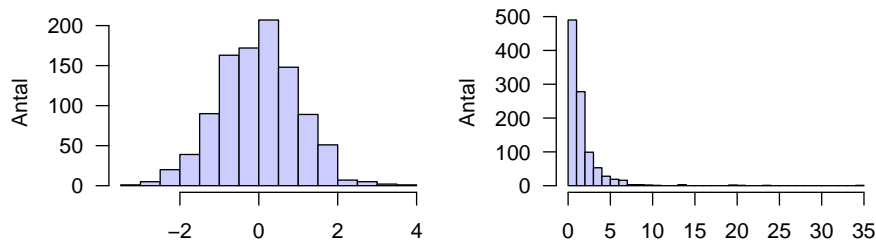
Af regneregler for linearkombinationer af stokastiske variable følger, at

$$\mathbb{E}[\hat{\mu}] = \mu \quad \text{og} \quad \text{Var}[\hat{\mu}] = \frac{\sigma^2}{n}.$$

Med andre ord er $\hat{\mu}$ en unbiased *estimator* for μ . Hvis hvert x_i er en konkret simulation af X_i , så er

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n h(x_i)$$

et unbiased Monte Carlo *estimat* af μ .



Figur 3: Histogram for hhv. x (til venstre) og $\exp(x)$ (til højre).

Eksempel: log-normal — *fortsat*

Vha. R er det enkelt at simulere fx. 1000 værdier x_1, \dots, x_{1000} , fra en standard normalfordeling:

```
x = rnorm(1000)
```

Figur 3 viser et histogram for hhv. x og $\exp(x)$. Herefter er et Monte Carlo estimat af $\mathbb{E}[\exp(X)]$ givet ved gennemsnittet

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n \exp(x_i).$$

I R giver en udregning af gennemsnittet:

```
mean(exp(x))
## [1] 1.638
```

Dvs. $\hat{\mu} = 1.6379$ er et estimat af $\mathbb{E}[\exp(X)]$. Spørgsmålet er hvor præcist et estimat $\hat{\mu}$ er.

Konfidensinterval for Monte Carlo estimat

For et tilstrækkelig stort n gælder der ifølge central grænseværdi sætning tilnærmelsesvist at

$$\frac{\hat{\mu} - \mu}{\sigma/\sqrt{n}} \sim \mathcal{N}(0, 1).$$

Heraf følger, at et *tilnærmet* $(1 - \alpha)100\%$ konfidensinterval for μ er givet ved

$$\hat{\mu} \pm z_{\alpha/2} \frac{s}{\sqrt{n}}, \quad (3)$$

hvor stikprøvestandardafvigelsen s er givet ved den sædvanlige formel:

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (h(x_i) - \hat{\mu})^2}.$$

Eksempel: log-normal — *fortsat*

Vi starter med at finde standardafvigelsen s :

```
sd(exp(x))  
## [1] 2.275
```

Vi kan nu udregne et 95% ved at indsætte i formel (3):

```
mean(exp(x)) + c(-1, 1) * qnorm(0.975) * sd(exp(x))/sqrt(1000)  
## [1] 1.497 1.779
```

Vi er hermed 95% sikre på at den forventede værdi af $\exp(X)$ ligger mellem 1.4969 og 1.7789. Dette er umiddelbart dette et relativt bredt konfidensinterval.

Bestemmelse af stikprøvestørrelse

Når man anvender Monte Carlo metoden til at estimere den forventede værdi, så er det afgørende, at det tilhørende konfidensinterval ikke er for langt. Er konfidensintervallet for langt, indikere det, at den valgte stikprøvestørrelse, n , ikke er stor nok. For at afgøre, hvordan vi opnår et konfidensinterval af en passende længde bemærker vi først, at formelen for konfidensinterval (3) kan skrives som

$$\hat{\mu} \pm ME,$$

hvor $ME = z_{\alpha/2} \frac{s}{\sqrt{n}}$ er fejlmarginen (*Margin of Error*). Hvis vi isolerer n i udtrykket for fejlmarginen får vi

$$n = \frac{(z_{\alpha/2})^2 s^2}{(ME)^2}. \quad (4)$$

Formel (4) giver en anslået værdi for n , der vil sikre en fejlmargin af størrelse ME . Grunden til at det kun er anslået er, at vi anvender den estimerede standardafvigelse s og ikke σ . Bemærk, at estimatet s er fundet på baggrund af et eksisterende simulationseksperiment.

Eksempel: log-normal — *fortsat*

Antag nu at vi ønsker en fejlmargin på $ME = 0,01$. Vi finder n i R:

```
n = (qnorm(0.975) * sd(exp(x))/0.01)^2  
n  
## [1] 198790
```

Vi gentager ovenstående simulationer og udregner, men nu med $n = 1.9879 \times 10^5$:

```
x = rnorm(n)
mean(exp(x))
## [1] 1.646
sd(exp(x))
## [1] 2.176
mean(exp(x)) + c(-1, 1) * qnorm(0.975) * sd(exp(x))/sqrt(n)
## [1] 1.636 1.655
```

Vi er hermed 95% sikre på, at den sande middelværdi μ ligger i intervallet fra 1.6363 til 1.6555. Vi har konkret opnået at fejlmarginen er 0.0096, hvilket skal sammenlignes med et ønske om en fejlmargin på 0.01.

Flere eksempler på anvendelse af Monte Carlo

Eksempel: Kontrol af udregning

Antag $X \sim \mathcal{N}(\mu, \sigma^2)$, dvs. X er en normalfordelt stokastisk variabel med middelværdi μ og varians σ^2 . For stokastiske variable gælder generelt $\text{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$. Isoleres $\mathbb{E}[X^2]$ opnås $\mathbb{E}[X^2] = \sigma^2 + \mu^2$. Antag $\mu = 3$ og $\sigma^2 = 14$. Der gælder derfor $\mathbb{E}[X^2] = 14 + 3^2 = 23$. Et Monte Carlo eksperiment giver en indikation på om udregningen er korrekt:

```
x = rnorm(10000, 3, sqrt(14))
mean(x^2)
## [1] 22.92
```

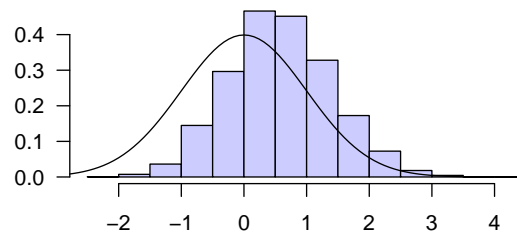
Dvs. et Monte Carlo estimat af $\mathbb{E}[X]$ er 22.9249, hvilket indikere at den uregnede værdi 23 ikke kan være helt forkert (der *er* korrekt). Vi kan supplere med et tilnærmet 95% konfidensinterval:

```
mean(x^2) + c(-1, 1) * qnorm(0.975) * sd(x^2)/sqrt(n)
## [1] 22.79 23.06
```

Eksempel: Maksimum af to stokastiske variable

Antag X_1 og X_2 er to uafhængige og standard normalfordelte stokastiske variable. Vi er interesseret i stokastisk variabel X der til enhver tid er maksimum af X_1 og X_2 . Dette kan formelt skrives som $X = \max(X_1, X_2)$. Det er ikke enkelt at udregne den forventede værdi af X eller andre af dens egenskaber. For at finde et estimat af $\mathbb{E}[X]$ anvender vi en Monte Carlo tilgang.

Først simulerer vi n realisationer af X_1 som vi betegner $x_{11}, x_{12}, \dots, x_{1n}$. Tilsvarende simulerer vi n realisationer af X_2 som vi betegner $x_{21}, x_{22}, \dots, x_{2n}$. For hvert par x_{1i} og x_{2i} beregner vi maksimum $x_i = \max(x_{1i}, x_{2i})$. Estimatet af $\mathbb{E}[X]$ er da gennemsnittet af x_1, x_2, \dots, x_n . I tilfældet med $n = 10000$ udføres dette Monte Carlo i R som følger:



Figur 4: Histogram for x samt tæthedsfunktionen for en standard normalfordelt stokastisk variabel.

```
x1 = rnorm(10000)
x2 = rnorm(10000)
x = pmax(x1, x2) ## pmax(...) tager parvise maksima
mean(x)
## [1] 0.5775
```

Monte Carlo estimatet af X er derfor 0.5775, hvilket som forventet er højere end middelværdien for X_1 og X_2 , der for begges tilfælde er nul. Figur 4 viser et histogram for x samt tæthedsfunktionen for en standard normalfordelt stokastisk variabel. Som forventet er histogrammet forskudt til højre i forhold til tæthedsfunktionen.

3 Monte Carlo — estimation af sandsynlighed

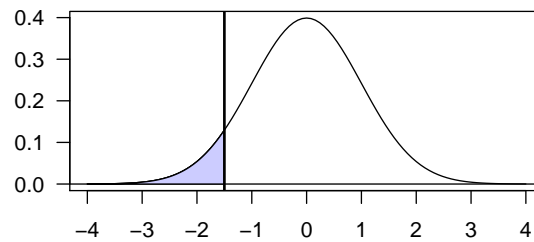
I dette afsnit skal vi se, hvordan man kan bruge Monte Carlo metoden i afsnit 2 til at estimere sandsynligheder.

Eksempel: Halesandsynlighed i en standard normalfordeling

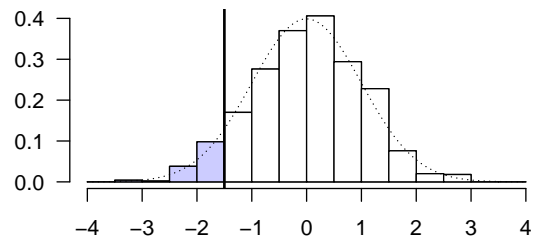
Antag at X er standard normalfordelt. Vi ønsker at bestemme sandsynligheden for, at X er mindre end eller lig -1.5. I det følgende betegner vi denne sandsynlighed π . Sandsynligheden π er illustreret på Figur 5. I praksis kan vi nemt finde π vha. R:

```
pnorm(-1.5)
## [1] 0.06681
```

Antag nu, at vi ikke har adgang til en kommando som `pnorm`, men at vi vha. en computer kan simulere værdier fra en normalfordeling. Vha. R kan vi nemt simulere 1000 værdier fra en standard normalfordeling:



Figur 5: Tæthedsfunktionen for en standardnormalfordelt stokastisk variabel. Det farvede område svarer sandsynligheden for at X er mindre end eller lig -1.5.



Figur 6: Histogram for de 1000 simulerede værdier. Det farvede område svarer til værdier mindre end eller lig -1.5. Den prikkede linje er tæthedsfunktionen for en standardnormalfordeling.

```
x = rnorm(1000)
```

Figur 6 viser et histogram over de 1000 simulerede værdier. I histogrammet er andelen af x_i 'er der er mindre end eller lig -1.5 på 7.1%. Dette er et estimat af π . Vi kan tænke på π som sandsynligheden for succes, hvor en succes er at X er mindre end eller lig -1.5. Fortolkning af en sandsynlighed π er *andelen* af succeser i det lange løb. Dvs. estimation af en sandsynlighed er det samme som at estimere en andel — og vi kan dermed genbruge den sædvanlige teori for estimation af andele.

Antag vi har n uafhængige stokastiske variable X_1, \dots, X_n , hvor $X_i \sim \mathcal{N}(0, 1)$. For hver X_i indfør en ny stokastisk variabel Y_i der kan tage værdierne 0 og 1. Hver Y_i er en funktion af X_i på følgende måde:

$$Y_i = \begin{cases} 1 & \text{hvis } X_i \leq -1.5 \\ 0 & \text{hvis } X_i > -1.5 \end{cases}$$

Da $P(X \leq -1.5) = \pi$ gælder der at $P(Y_i = 1) = \pi$, dvs. Y_1, \dots, Y_n er uafhængige bernoulli variable med sandsynlighedsparameter π . Heraf følger det, at

$$\mathbb{E}[Y_i] = \pi \quad \text{og} \quad \text{Var}[Y_i] = \pi(1 - \pi).$$

Lad $\hat{\pi}$ betegne gennemsnittet af Y_1, \dots, Y_n , da gælder

$$\mathbb{E}[\hat{\pi}] = \pi \quad \text{og} \quad \text{Var}[\hat{\pi}] = \frac{\pi(1-\pi)}{n}.$$

Vi har således at $\hat{\pi}$ er en unbiased estimator for sandsynligheden π .

I R udregnes y_i 'erne nemt:

```
y = 1 * (x <= -1.5)
```

Som eksempel oplister vi de først fem x_i 'er:

```
x[1:5]
## [1]  1.5951 -2.0326  0.1982  1.9852 -1.4935
```

Bemærk at x_1 er større end -1.5 og x_2 er mindre end -1.5. En oplistning af det først fem y_i 'er viser, hvordan x_i 'erne er blevet konverteret til 0'er og 1'er. Bemærk at $y_1 = 0$ og $y_2 = 1$.

```
y[1:5]
## [1] 0 1 0 0 0
```

Herefter er det enkelt at finde gennemsnittet af y_i 'erne og Monte Carlo estimatet $\hat{\pi}$:

```
mean(y)
## [1] 0.071
```

Vi har hermed at $\hat{\pi} = 0.071$ er et unbiased Monte Carlo estimat af sandsynligheden π . I praksis er det ikke nødvendigt at introducere y . Udregningerne kan udføres direkte på x :

```
mean(x <= -1.5)
## [1] 0.071
```

Bemærk at spørgsmålet om at estimere en sandsynlighed blev konverteret til et spørgsmål om at estimere en middelværdi. Hermed er estimation af en sandsynlighed principielt ikke forskellig fra at estimere en middelværdi.

Det følger af central grænseværdi sætning, at for tilstrækkelige store værdier af n gælder der

$$\frac{\hat{\pi} - \pi}{\sqrt{\pi(1-\pi)/n}} \stackrel{\text{ca.}}{\sim} \mathcal{N}(0, 1).$$

Hvis vi erstatter variansen $\pi(1-\pi)/n$ under kvadratroden med estimatet $\hat{\pi}(1-\hat{\pi})/n$, så giver en omskrivning, at et tilnærmet $(1-\alpha)100\%$ konfidensinterval for π er givet ved

$$\hat{\pi} \pm z_{\alpha/2} \sqrt{\frac{\hat{\pi}(1-\hat{\pi})}{n}}. \quad (5)$$

Et 95% konfidensintervallet for π udregnes vha. formel (5) i R som:

```
mean(y) + c(-1, 1) * qnorm(0.975) *
               sqrt(mean(y) * (1 - mean(y))/1000)

## [1] 0.05508 0.08692
```

Dvs. vi er 95% sikre på, at den sande sandsynlighed π ligger mellem 0.0551 og 0.0869.

Hvis konfidensintervallet er for bredt, kan vi anslå et passende n ved udregninger som i forrige afsnit. Hvis vi ønsker en fejlmargen ME , så er en anslået værdi for n givet ved

$$n = \hat{\pi}(1-\hat{\pi})(z_{\alpha/2}/ME)^2.$$

4 Monte Carlo p-værdi

Vi har ovenfor set, hvordan Monte Carlo metoden kan anvendes til at estimere en sandsynlighed. I forbindelse med hypotesetest er p-værdien sandsynligheden for, under H_0 , at observere en mere kritisk teststørrelse end den der er observeret for data. Lad data være repræsenteret ved en vektor bestående af n observationer: $\mathbf{x} = (x_1, x_2, \dots, x_n)$. Antag desuden, at vi har en teststørrelse $H(\mathbf{x})$.

Eksempel: Kast med en fair mønt

Vi har kastet med en mønt 100 gange og observeret krone 39 gange. I dette tilfælde er $\mathbf{x} = 39$. Vi ønsker at undersøge om mønten er fair, dvs. om det er lige sandsynligt at få plat og krone:

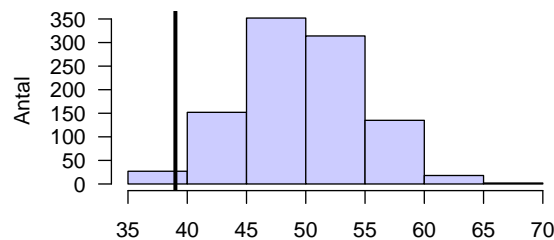
$$H_0 : \text{Mønten er fair} \quad \text{vs} \quad H_1 : \text{Mønten er ikke fair}$$

Under H_0 forventer vi, at observere 50 krone. Som teststørrelse vælger vi forskellen mellem det observerede antal og 50. Med andre ord

$$H(\mathbf{x}) = |\mathbf{x} - 50|.$$

I det konkrete tilfælde har vi $H(39) = |39 - 50| = |-11| = 11$. I R anvendes funktionen `abs` til at finde den absolutte værdi:

```
abs(39 - 50)
## [1] 11
```



Figur 7: Histogram over antal krone i 100 kast. Den lodrette linje angiver det observerede antal krone, $x = 39$.

Spørgsmålet er hvor kritisk 39 krone i 100 kast er for hypotesen om en fair mønt. p-værdien er sandsynligheden for at observere en mindst lige så kritisk test størrelse “næste gang” under antagelse af, at H_0 er sand. Vi kan tænke på processen at kaste en *fair* mønt 100 gange og tælle antal krone som et (H_0) eksperiment. I dette tilfælde er p-værdien sandsynligheden for at et H_0 eksperiment fører til et antal krone der afviger fra 50 med mindst 11 kast. I praksis er denne sandsynlighed er ikke svær at udregne, men vi forsøger os alligevel med Monte Carlo metoden. Ideen er at udføre H_0 eksperimentet mange gange og derefter bestemme andelen af H_0 eksperimenter der førte til en teststørrelse der var mindst lige så kritisk som $H(\mathbf{x})$. Denne andel er et estimat af p-værdien.

I det følgende betegner vi resultatet af et H_0 eksperiment som \mathbf{x}^* . p-værdien er med andre ord sandsynligheden for at et H_0 eksperiment fører til en teststørrelse $H(\mathbf{x}^*)$ der er mindst lige så kritisk som $H(\mathbf{x})$.

Ovenstående ideer kan opsummeres i følgende algoritme til estimation af p-værdi

- Udregn $H(\mathbf{x})$
- For $i = 1, \dots, n$ udfør trin 1 til 3
 1. Find \mathbf{x}^* ved at udføre H_0 eksperiment.
 2. Udregn $H(\mathbf{x}^*)$.
 3. Hvis $H(\mathbf{x}^*)$ er *mindst* lige så kritisk for H_0 som $H(\mathbf{x})$, så sæt $y_i = 1$ ellers sæt $y_i = 0$.
- Et estimat af p-værdien er $\bar{y} = n^{-1} \sum_{i=1}^n y_i$.

Bemærk, at selvom \bar{y} er et gennemsnit, så er det præcis det samme som andelen af de n H_0 eksperimenter der fører til en teststørrelse $H(\mathbf{x}^*)$ der er mindst lige så kritisk som den observerede teststørrelse $H(\mathbf{x})$.

Eksempel: Kast med en fair mønt — *fortsat*

Under H_0 er antallet af krone i 100 kast med en fair mønt binomial fordelt med sandsynlighedsparameter 0,5 og antalsparameter 100. Med R kan man derfor udføre et H_0 eksperiment som

```
rbinom(1, size = 100, prob = 0.5)
## [1] 48
```

Vi kan udføre 10.000 eksperimenter ved

```
x.star = rbinom(10000, size = 100, prob = 0.5)
```

Figur 7 viser fordelingen i de 10000 x_i 'er. Allerede ved at kigge på histogrammet i Figur 7 kan vi se, at det er sjældent man opnår 39 eller krone i 100 kast med en fair mønt.

Vi kan nu udregne teststørrelsen for hvert x_i^* og de tilhørende y_i værdier:

```
H = abs(x.star - 50) ## Udregn 10.000 teststørrelser
y = H >= 11 ## Udregn y
mean(y)
## [1] 0.035
```

Dvs. den estimerede p-værdi er i dette tilfælde 0.035. Et tilnærmet 95% konfidensinterval for p-værdien kan udregnes vha. formel (5):

```
mean(y) + c(-1, 1) * qnorm(0.975) * sqrt(mean(y) * (1 - mean(y))/10000)
## [1] 0.0314 0.0386
```

Som nævnt er det ikke svært at udregne den korrekte p-værdi i dette tilfælde. Da x^* er binomialfordelt kan p-værdien findes som

```
2 * pbinom(39, size = 100, prob = 0.5)
## [1] 0.0352
```

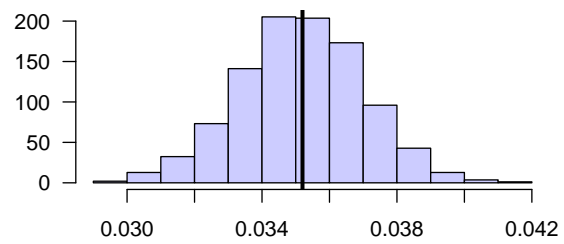
Som sagt er Monte Carlo p-værdien kun et estimat, og hver gang R-koden anvendes opnås et nyt resultat fordi nye H_0 eksperimenter udføres. For at illustrere denne tilfældige variation i estimatet kan vi udføre Monte Carlo estimationen mange gange. Først bemærker vi, at ovenstående procedure kan komprimeres til en linje:

```
mean(abs(rbinom(10000, size = 100, prob = 0.5) - 50) >= 11)
## [1] 0.036
```

Denne udregning kan nemt gentages, fx. 1000 gange, vha. kommandoen `replicate`:

```
p = replicate(2500, mean(abs(rbinom(10000, size = 100, prob = 0.5) - 50) >=
11))
```

Figur 8 viser et histogram over de 2500 Monte Carlo p-værdier. Som det ses er der en vis variation i Monte Carlo p-værdierne. Denne variation kan mindskes ved at øge antallet af H_0 eksperimenter fra 10.000 til fx. 1.000.000.



Figur 8: Histogram over Monte Carlo p -værdier. Den lodrette streg angiver den sande p -værdi.