



Oblig 1c

Gruppe 98

av

Christopher Sanden

i

MA-223

Statistikk

Fakultet for teknologi og realfag

Universitetet i Agder

Arendal, Februar 2026

# Innhold

<b>1</b>	<b>Innledning</b>	<b>1</b>
<b>2</b>	<b>Oppgave 1: Bokoppgave 24 (Kapittel 4)</b>	<b>2</b>
2.1	(a) Ordnet, med tilbakelegging . . . . .	2
2.2	(b) Uordnet, med tilbakelegging . . . . .	2
2.3	(c) Ordnet, uten tilbakelegging . . . . .	2
2.4	(d) Uordnet, uten tilbakelegging . . . . .	2
2.5	(e) Praktiske eksempler . . . . .	2
<b>3</b>	<b>Oppgave 2: Bokoppgave 19 (Kapittel 5)</b>	<b>3</b>
3.1	(a) Ett trekk . . . . .	3
3.2	(b) Fem trekk med tilbakelegging . . . . .	3
3.3	(c) Fem trekk uten tilbakelegging . . . . .	5
<b>4</b>	<b>Oppgave 3: Mengderegning og Venn-diagram</b>	<b>6</b>
4.1	(a) Venn-diagram . . . . .	6
4.2	(b) Union og differanse . . . . .	8
4.3	(c) Komplementer . . . . .	8
4.4	(d) Betinget sannsynlighet . . . . .	8
<b>5</b>	<b>Oppgave 4: R-hjørnet</b>	<b>9</b>
5.1	(a) Egen funksjon for binomialkoeffisient . . . . .	9
5.2	(b) Birthday-problemet . . . . .	10
5.3	(c) Sample-funksjonen . . . . .	11
5.4	(d) og (e) For-løkker . . . . .	12
5.5	(f) Forskjellen på replace = TRUE og FALSE . . . . .	13
<b>Tillegg A</b>	<b>Vedlegg</b>	<b>14</b>
A.1	Oppgave1 . . . . .	14
A.2	Oppgave 2 . . . . .	15
A.3	Oppgave 3 . . . . .	17
A.4	Oppgave 4 . . . . .	19

## Figurer

1	Stolpediagram som viser sannsynligheten, ut av 1, for de forskjellige scenario-ene .	4
2	Venn diagram tilhørende oppgave 3 . . . . .	7

## List of Listings

1	Egendefinert funksjon for binominalkoeffisient . . . . .	9
2	Forklaring av Bursdagsparadoksetfunksjonen . . . . .	10
3	Forklaring av kode gitt i oppgavedokumentet . . . . .	11
4	For-løkker etterspurt av oppgavene . . . . .	12
5	Trekking med og uten tilbakelegging . . . . .	13

# 1 Innledning

Denne rapporten inneholder løsning av bokoppgave 24 (kapittel 4) og bokoppgave 19 (kapittel 5), samt tilhørende oppgaver i oblig-dokumentet. Alle beregninger er kontrollert i R, og tilhørende kode finnes i egne .R-filer for hver oppgave. Se Vedlegg(A) for komplette filer.

## 2 Oppgave 1: Bokoppgave 24 (Kapittel 4)

Vi skal bestemme hvor mange måter man kan trekke 5 elementer fra 12, under ulike betingelser om orden og tilbakelegging.

### 2.1 (a) Ordnet, med tilbakelegging

Når trekkene er ordnet og med tilbakelegging, kan hvert av de 5 valgene gjøres blant 12 elementer. Antall muligheter er derfor:

$$12^5 = 248\,832$$

### 2.2 (b) Uordnet, med tilbakelegging

Når rekkefølgen ikke spiller noen rolle og elementer kan gjentas, brukes kombinasjon med repetisjon:

$$\binom{12 + 5 - 1}{5} = \binom{16}{5} = 4368$$

### 2.3 (c) Ordnet, uten tilbakelegging

Når rekkefølgen spiller rolle og elementene ikke kan gjentas, brukes permutasjon:

$$P(12, 5) = 12 \cdot 11 \cdot 10 \cdot 9 \cdot 8 = 95\,040$$

### 2.4 (d) Uordnet, uten tilbakelegging

Når rekkefølgen ikke spiller rolle og elementene ikke kan gjentas, brukes vanlig kombinasjon:

$$\binom{12}{5} = 792$$

### 2.5 (e) Praktiske eksempler

Eksempler på de fire situasjonene:

- Ordnet med tilbakelegging: En 5-sifret kode der hvert siffer kan gjentas.
- Uordnet med tilbakelegging: Valg av 5 iskuler fra 12 smaker.
- Ordnet uten tilbakelegging: Velg 5 studenter fra en gruppe på 12 og sorter dem etter alder.
- Uordnet uten tilbakelegging: Trekk 5 kort fra en bunke på 12 uten å legge dem tilbake.

### 3 Oppgave 2: Bokoppgave 19 (Kapittel 5)

Vi har en Twist-pose med 7 kokos og 13 lakris, totalt 20 biter.

#### 3.1 (a) Ett trekk

Sannsynligheten for å trekke kokos er:

$$P(\text{kokos}) = \frac{7}{20} = 0.35$$

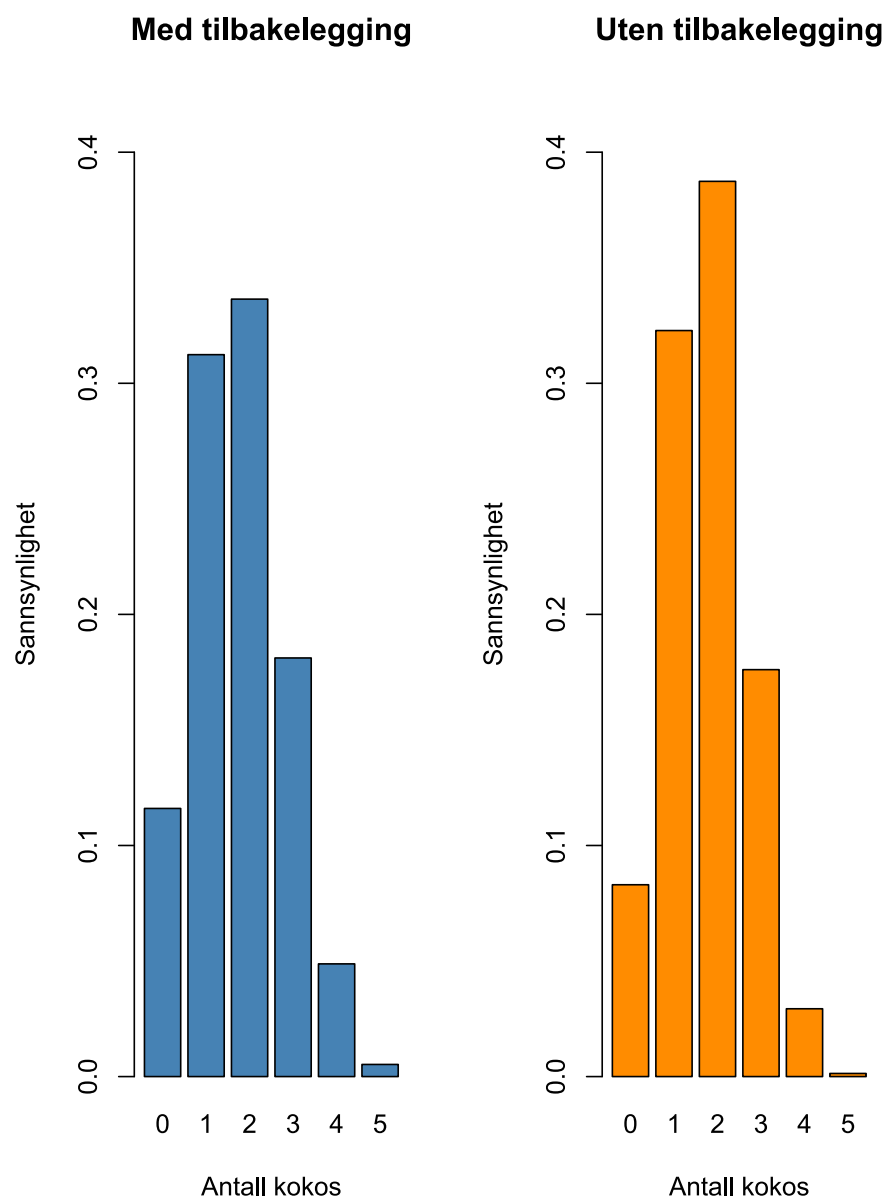
#### 3.2 (b) Fem trekk med tilbakelegging

Når vi legger tilbake etter hvert trekk, er sannsynligheten konstant i hvert forsøk. Antall kokos i 5 trekk følger derfor en binomisk fordeling:

$$X \sim \text{Bin}(n = 5, p = 0.35)$$

$$P(X = k) = \binom{5}{k} (0.35)^k (0.65)^{5-k}$$

Stolpediagrammet nedenfor viser sannsynlighetene for  $k = 0, 1, 2, 3, 4, 5$ . Se A.2 for R-kode til utregninger.



**Figur 1:** Stolpediagram som viser sannsynligheten, ut av 1, for de forskjellige scenario-ene

### 3.3 (c) Fem trekk uten tilbakelegging

Når vi ikke legger tilbake, endres sannsynligheten underveis. Antall kokos følger da en hypergeometrisk fordeling:

$$P(X = k) = \frac{\binom{7}{k} \binom{13}{5-k}}{\binom{20}{5}}$$

Stolpediagrammet nedenfor viser sannsynlighetene for  $k = 0, 1, 2, 3, 4, 5$ .

Forskjellen mellom (b) og (c) er at i (b) er trekkene uavhengige, mens i (c) påvirker hvert trekk det neste.



## 4 Oppgave 3: Mengderegning og Venn-diagram

Vi tar utgangspunkt i de gitte sannsynlighetene:

$$P(A) = 0.8, \quad P(B) = 0.2, \quad P(A \cap B) = 0.1.$$

Her betyr  $P(A)$  sannsynligheten for at hendelsen  $A$  inntreffer,  $P(B)$  sannsynligheten for at  $B$  inntreffer, og  $P(A \cap B)$  sannsynligheten for at både  $A$  og  $B$  inntreffer samtidig. Videre er  $A \cup B$  hendelsen at minst én av  $A$  eller  $B$  inntreffer, og  $A^c$  betyr at  $A$  *ikke* inntreffer (komplementet).

### 4.1 (a) Venn-diagram

For å fylle Venn-diagrammet deler vi opp i fire områder:

$$P(A \setminus B) = P(A) - P(A \cap B) = 0.8 - 0.1 = 0.7$$

$$P(B \setminus A) = P(B) - P(A \cap B) = 0.2 - 0.1 = 0.1$$

$$P(A \cap B) = 0.1$$

Området utenfor begge mengdene er komplementet til  $A \cup B$ :

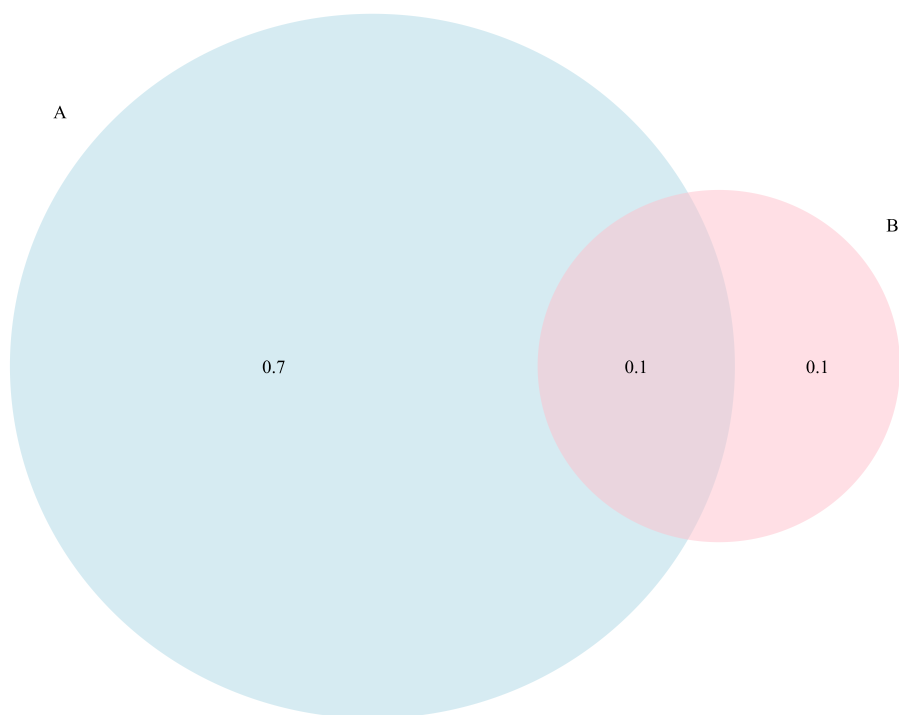
$$P((A \cup B)^c) = 1 - P(A \cup B).$$

Vi finner først  $P(A \cup B)$  (se del (b)), og får  $P(A \cup B) = 0.9$ . Dermed:

$$P((A \cup B)^c) = 1 - 0.9 = 0.1.$$

Dette tilsvarer også:

$$1 - P(A) - P(B) + P(A \cap B) = 1 - 0.8 - 0.2 + 0.1 = 0.1.$$



**Figur 2:** Venn diagram tilhørende oppgave 3

## 4.2 (b) Union og differanse

Union:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) = 0.8 + 0.2 - 0.1 = 0.9.$$

Differanse ("A men ikke B"):

$$P(A \setminus B) = P(A) - P(A \cap B) = 0.8 - 0.1 = 0.7.$$

## 4.3 (c) Komplementet

Komplementet til snittet:

$$P((A \cap B)^c) = 1 - P(A \cap B) = 1 - 0.1 = 0.9.$$

Ved De Morgans lover gjelder  $(A \cap B)^c = A^c \cup B^c$ , så:

$$P(A^c \cup B^c) = P((A \cap B)^c) = 0.9.$$

## 4.4 (d) Betinget sannsynlighet

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{0.1}{0.2} = 0.5$$

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)} = \frac{0.1}{0.8} = 0.125.$$

Forskjellen er at  $P(A \mid B)$  måler hvor stor andel av  $B$  som også ligger i  $A$ , mens  $P(B \mid A)$  måler hvor stor andel av  $A$  som også ligger i  $B$ .

## 5 Oppgave 4: R-hjørnet

### 5.1 (a) Egen funksjon for binomialkoeffisient

---

```
1      #4a
2      myChoose = function(n, k){
3        factorial(n) / (factorial(k) * factorial(n - k))
4      }
5
6      choose(4, 2)
7      myChoose(4, 2)
```

---

**Listing 1:** Egendefinert funksjon for binominalkoeffisient

Vi implementerte en egen funksjon for å beregne binomialkoeffisienten uten å bruke `choose()`.

## 5.2 (b) Birthday-problemet

---

```
1      #4b
2      #Definerer funksjonen bd med parameter n
3      bd = function(n){
4
5          #Setter antall "frie" dager
6          dager = 365 + 1 - 1:n
7
8          #Deler antall "frie" dager på antall
9          #dager i et år - gir et desimaltall som
10         #ganges med 100 for å få den eksakte prosentverdien
11         #for sannsynligheten at ingen i samme set har
12         #samme bursdag
13         sjanser = dager / 365
14
15         #Utfører utregningen og lagrer resultatet til
16         #variablen output
17         output = prod(sjanser)
18
19         #Printer resultatet
20         output
21     }
22     #Regner ut og printer sannsynligheten for at
23     #to personer i samme set har samme bursdag
24     #Vi gjør "1 - bd()" for å finne sannsynligheten
25     1 - bd(22)
26     1 - bd(23)
27     #Denne funksjonen regner sannsynligheten for at ingen
28     #personer i samme set har bursdag på samme dag
29     #Bursdagsparadokset er et berømt statistikk eksempel
```

---

**Listing 2:** Forklaring av Bursdagsparadoksetfunksjonen

Funksjonen beregner sannsynligheten for at minst to personer i en gruppe deler bursdag.

### 5.3 (c) Sample-funksjonen

---

```
1      #4c
2      #Setter startpunktet til en Random Number Generator(RNG)
3      set.seed(314)
4
5      #Lager en mengde / sekvens fra 3 til 27 (3, 4, 5, ..., 27)
6      base = 3:27
7
8      #Trekker 14 tall fra sekvensen "base" med tilbakelegging
9      a = sample(base, 14, replace=TRUE)
```

---

**Listing 3:** Forklaring av kode gitt i oppgavedokumentet

Kommandoen `sample()` trekker tilfeldige tall fra et gitt intervall.

## 5.4 (d) og (e) For-løkker

---

```
1      #4d
2      totalSum = 0
3      for(verdi in a){
4          totalSum = totalSum + verdi
5      }
6      totalSum
7      sum(a)
8
9
10     #4e
11     totalProd = 1
12     for (verdi in a) {
13         totalProd = totalProd * verdi
14     }
15     totalProd
16     prod(a)
```

---

**Listing 4:** For-løkker etterspurt av oppgavene

Vi implementerte egne for-løkker for henholdsvis summasjon og produkt.

## 5.5 (f) Forskjellen på `replace = TRUE` og `FALSE`

---

```
1      #4f
2      #Med tilbakelegging - kan få samme verdi flere gange
3      a = sample(base, 14, replace=TRUE)
4      sort(a)
5
6      #Uten tilbakelegging - kan kun få atomære verdier
7      a = sample(base, 14, replace=FALSE)
8      sort(a)
```

---

**Listing 5:** Trekking med og uten tilbakelegging

`replace = TRUE` tillater at samme element kan trekkes flere ganger, mens `replace = FALSE` gir trekking uten tilbakelegging.



## A Vedlegg

### A.1 Oppgave1

---

```
1  #Hvor mange måter kan du plukke 5 elementer fra 12
2  #når trekket er...
3
4  n <- 12
5  k <- 5
6
7  #a) ordnet, med tilbakelegging
8  n^k
9
10 #b) uordnet, med tilbakelegging
11 choose((n + k - 1), k)
12
13 #c) ordnet, uten tilbakelegging
14 factorial(n) / factorial(n - k)
15
16 #d) uordnet, uten tilbakelegging
17 choose(n, k)
18
19 #Eksempel på a) - PIN-kode med 5 siffer. Man kan gjenbruke ett siffer flere
20 #ganger
21
22 #Eksempel på b) - Velg 5 is-kuler fra et utvalg av 5 smaker. Hver kule trenger
23 #ikke være unik
24
25 #Eksempel på c) - Velg 5 studenter fra en gruppe på 12 og sorter dem etter alder
26
27 #Eksempel på d) - Trekk 5 kort fra en bunke på 12 uten å legge dem tilbake
```

---

## A.2 Oppgave 2

---

```
1  #Twist-posen
2  #a) Du skal rydde opp etter en fest, og finner en Twist-pose med kun 7 kokos
3  #og 13 lakris igjen. Du plukker en tilfeldig twist. Hva er sannsynligheten
4  #for at den blir en kokos?
5  kokos <- 7
6  lakris <- 13
7  total = kokos + lakris
8
9  P_kokos = kokos / total
10
11
12  #b) Du liker verken lakris eller kokos, så du legger den du har plukket tilbake
13  #i posen for å plukke en ny. Du prøver deg slik 5 ganger, og legger tilbake
14  #for hver gang. Lag stoplediagram over sannsynligheten for at du har plukket
15  #kokos respektivt 0, 1, 2, 3, 4 og 5 ganger
16  N <- total
17  S <- kokos
18  p <- S / N
19  n <- 5
20
21  #Formel 4.2.3 i formelheftet
22  ProbComb = function(k){
23    choose(n, k) * (p^k) * (1 - p)^(n - k)
24  }
25  ProbComb(0)
26  ProbComb(1)
27  ProbComb(2)
28  ProbComb(3)
29  ProbComb(4)
30  ProbComb(5)
31
32
33
34
35  #Litt senere på morran er du så sultern at selv kokos og lakris går an, så
36  #du plukker 5 twist og spiser dem. Lag stolpediagram over sannsynligheten for
37  #at du spiser respektive 0, 1, 2, 3, 4 og 5 kokos
38  ProbCombNoReplace = function(k){
39    (choose(S, k) * choose((N - S), n - k) / choose(N, n))
40  }
41  ProbCombNoReplace(0)
42  ProbCombNoReplace(1)
43  ProbCombNoReplace(2)
44  ProbCombNoReplace(3)
45  ProbCombNoReplace(4)
46  ProbCombNoReplace(5)
```

```

47
48
49 #Diagrammer
50 k_values <- 0:n
51 probs_binom <- sapply(k_values, ProbComb)
52 probs_hyper <- sapply(k_values, ProbCombNoReplace)
53
54 max_b <- max(probs_binom)
55 max_h <- max(probs_hyper)
56
57 max_y <- max(max_b, max_h)
58
59 ylim_val <- c(0, max_y * 1.1)
60
61
62 par(mfrow = c(1, 2))
63
64 barplot(probs_binom, names.arg = k_values, col = "steelblue",
65         ylim = ylim_val, xlab = "Antall kokos", ylab = "Sannsynlighet",
66         main = "Med tilbakelegging")
67
68 barplot(probs_hyper, names.arg = k_values, col = "darkorange",
69         ylim = ylim_val, xlab = "Antall kokos", ylab = "Sannsynlighet",
70         main = "Uten tilbakelegging")
71 par(mfrow = c(1, 1))
72

```

---

### A.3 Oppgave 3

---

```
1  install.packages("VennDiagram")
2
3  library(VennDiagram)
4
5  P_A <- 0.8
6  P_B <- 0.2
7  P_AB <- 0.1
8
9  bare_A <- P_A - P_AB
10 bare_B <- P_B - P_AB
11 utenfor <- 1 - (bare_A + bare_B + P_AB)
12 bare_A
13 bare_B
14 utenfor
15
16 #Sjekk at verdier stemmer
17 bare_A + bare_B + utenfor + P_AB
18
19
20 #3b
21 P_union <- P_A + P_B - P_AB
22 P_union
23
24 P_A_uten_B <- bare_A
25 P_A_uten_B
26
27
28 #3c
29 P_komplement_intersection <- 1 - P_AB
30 P_komplement_intersection
31 #DeMorgan's lov sier at  $P(AB)^c = P(A^c \cup B^c)$ 
32 #Derfor er sannsynligheten den samme
33
34
35 #3d
36 P_A_gitt_B <- P_AB / P_B
37 P_B_gitt_A <- P_AB / P_A
38 P_A_gitt_B
39 P_B_gitt_A
40 #Forskjellen mellom disse to er at P_A_gitt_B er
41 #sannsynligheten for at A skjer gitt at B allerede
42 #har skjedd - P_B_gitt_A er da sannsynligheten for
43 #at B skjer gitt at A allerede har skjedd
44
45
46 #Venn Diagrammet
```

```
47 draw.pairwise.venn(area1 = P_A, area2 = P_B,  
48                     cross.area = P_AB, category = c("A", "B"),  
49                     fill = c("lightblue", "pink"),  
50                     lty = "blank")
```

---

## A.4 Oppgave 4

---

```
1  #4a
2  myChoose = function(n, k){
3    factorial(n) / (factorial(k) * factorial(n - k))
4  }
5
6  choose(4, 2)
7  myChoose(4, 2)
8
9
10 #4b
11 #Definerer funksjonen bd med parameter n
12 bd = function(n){
13
14   #Setter antall "frie" dager
15   dager = 365 + 1 - 1:n
16
17   #Deler antall "frie" dager på antall
18   #dager i et år - gir et desimaltall som
19   #ganges med 100 for å få den eksakte prosentverdien
20   #for sannsynligheten at ingen i samme set har
21   #samme bursdag
22   sjanser = dager / 365
23
24   #Utfører utregningen og lagrer resultatet til
25   #variablen output
26   output = prod(sjanser)
27
28   #Printer resultatet
29   output
30 }
31 #Regner ut og printer sannsynligheten for at
32 #to personer i samme set har samme bursdag
33 #Vi gjør "1 - bd()" for å finne sannsynligheten
34 1 - bd(22)
35 1 - bd(23)
36 #Denne funksjonen regner sannsynligheten for at ingen
37 #personer i samme set har bursdag på samme dag
38 #Bursdagsparadokset er et berømt statistikk eksempel
39
40
41 #4c
42 #Setter startpunktet til en Random Number Generator(RNG)
43 set.seed(314)
44
45 #Lager en mengde / sekvens fra 3 til 27 (3, 4, 5, ..., 27)
46 base = 3:27
```

```

47
48 #Trekker 14 tall fra sekvensen "base" med tilbakelegging
49 a = sample(base, 14, replace=TRUE)
50
51
52 #4d
53 totalSum = 0
54 for(verdi in a){
55     totalSum = totalSum + verdi
56 }
57 totalSum
58 sum(a)
59
60
61 #4e
62 totalProd = 1
63 for (verdi in a) {
64     totalProd = totalProd * verdi
65 }
66 totalProd
67 prod(a)
68
69
70 #4f
71 #Med tilbakelegging - kan få samme verdi flere gange
72 a = sample(base, 14, replace=TRUE)
73 sort(a)
74
75 #Uten tilbakelegging - kan kun få atomære verdier
76 a = sample(base, 14, replace=FALSE)
77 sort(a)
78
79
80
81
82
83
84

```

---