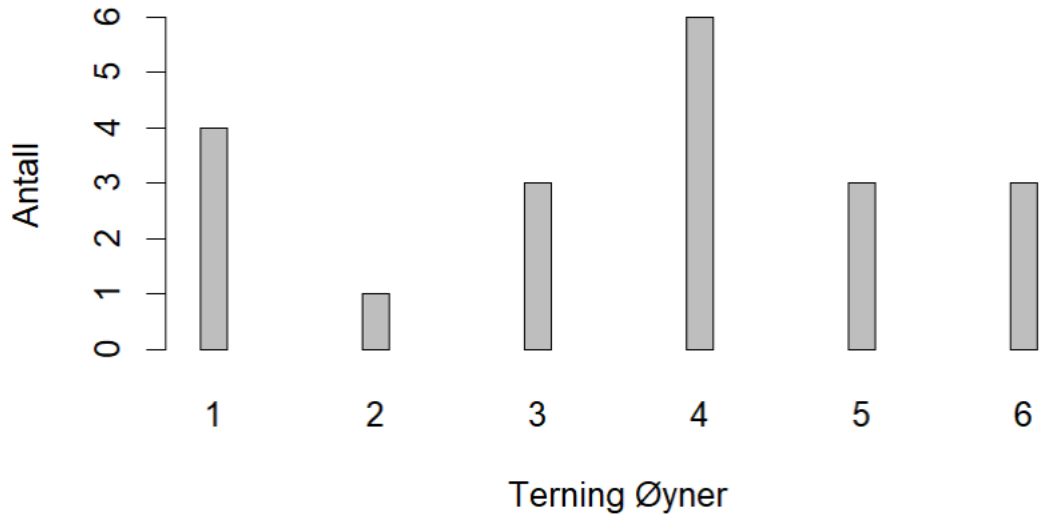
Oppgave 1

R-Kode

```r
n=20
terning=sample(seq(1:6),n,replace=T)
frekvenser=hist(terning,breaks=c(0,1,2,3,4,5,6)+.5,plot=F)
frekvenser$counts #dette er resultat av hist()
barplot(frekvenser$counts, names.arg=c(1,2,3,4,5,6),
        xlab = "Terning Øyner", ylab = "Antall",
        width=.1,space=c(5)) #strekdiagram
summary(terning)
mean(terning) #gjennomsnitt
median(terning) #median
var(terning) #varians
sqrt(var(terning))#standardavvik
sd(terning) #standardavvik
```

- **Strekdiagram**



- **Gjennomsnitt, median, varians og standardavvik**

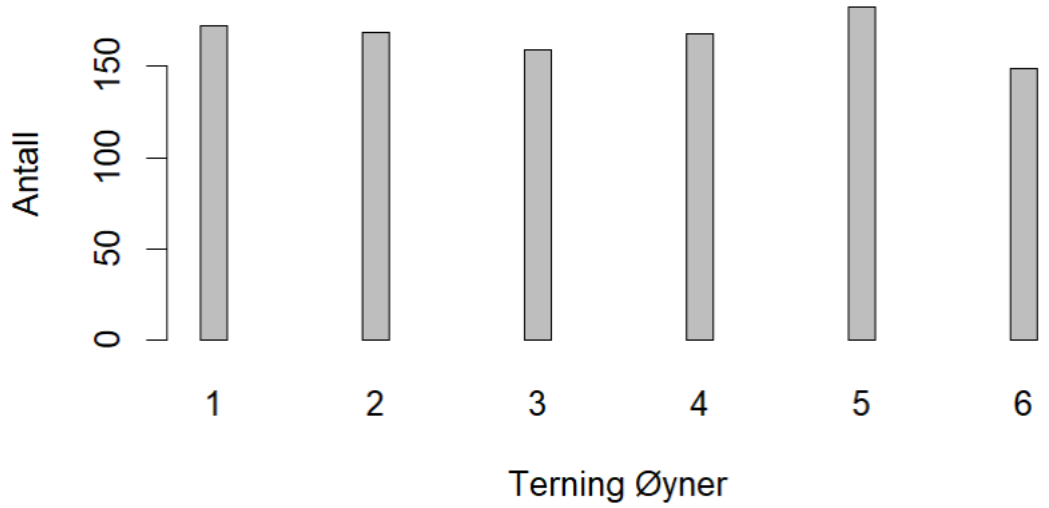
```
> mean(terning) #gjennomsnitt
[1] 3.6
> median(terning) #median
[1] 4
> var(terning) #varians
[1] 2.884211
> sqrt(var(terning))#standardavvik
[1] 1.698296
> sd(terning) #standardavvik
[1] 1.698296
```

- **Sammenligning av median og gjennomsnitt**

  I dette tilfelle er gjennomsnittet 3.6 og medianen er 4, det er en forskjell på 0.4

```
> mean(terning) #gjennomsnitt
[1] 3.6
> median(terning) #median
[1] 4
```

- **1000 simulerte terningkast**




```
> mean(terning) #gjennomsnitt
[1] 3.468
> median(terning) #median
[1] 3.5
> var(terning) #varians
[1] 2.881858
> sqrt(var(terning))#standardavvik
[1] 1.697604
> sd(terning) #standardavvik
[1] 1.697604
```

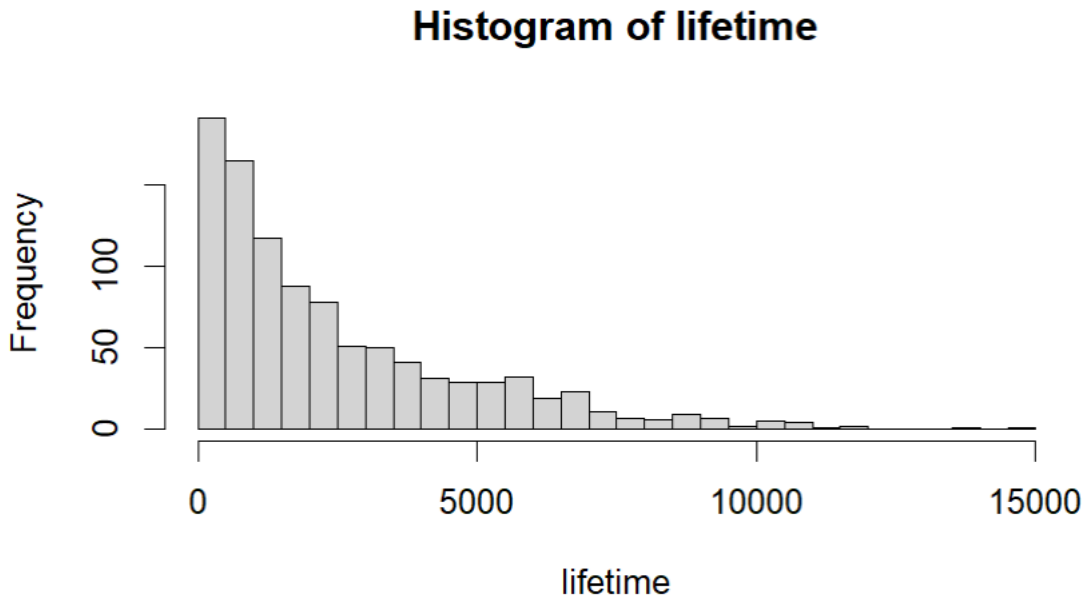
  - Hvis man sammenligner disse resultatene med 20 simulerte terningkast, ser man at resultatene blir mye mer presise. Utregningene har flere desimaler og strekdiagremmet er jevnere fordelt.

Oppgave 2

R-kode

```
1  lifetime=rexp(1000,rate=1/2500)
2  lifetime=round(lifetime,digits=0)
3  hist(lifetime,nclass=25)
4  sort(lifetime) #Sorterer fra minste til største verdi
5  mean(lifetime) #Gjennomsnittet
6  median(lifetime) #Medianen
```

- **Histogram**



- **Gjennomsnitt og median**

```
> mean(lifetime) #Gjennomsnittet
[1] 2487.49
> median(lifetime) #Medianen
[1] 1631
```

Det er en ganske stor forskjell på gjennomsnittet og medianen, dette er fordi det er noen få veldig store verdier, som derfor vil øke gjennomsnittet betydelig.

- **Sortert liste av datamaterialet**

```
[452]  1426  1439  1441  1446  1447  1449  1450  1451  1454  1456  1464
[463]  1466  1468  1469  1471  1474  1478  1479  1480  1489  1493  1498
[474]  1501  1501  1509  1514  1520  1534  1538  1548  1555  1557  1558
[485]  1563  1564  1573  1579  1583  1584  1589  1596  1598  1601  1605
[496]  1613  1615  1616  1617  1627  1635  1639  1645  1652  1653  1670
[507]  1675  1679  1684  1686  1690  1691  1704  1725  1745  1746  1751
[518]  1771  1771  1780  1781  1785  1795  1796  1800  1807  1810  1815
[529]  1819  1824  1829  1830  1834  1840  1847  1849  1853  1857  1860
[540]  1867  1870  1870  1875  1892  1892  1899  1903  1914  1920  1926
[551]  1927  1931  1934  1941  1944  1947  1969  1980  1988  1993  1994
```

Her er en liten del av det sorterte datamaterialet som består av 1000 verdier. Det markerte området på listen er verdiene 500 og 501, som er verdiene i midten av listen. Får å finne medianen av denne sorterte listen må man ta gjennomsnittet av disse verdiene: $1627 + 1635 = 3262$, og $3262 \div 2 = 1631$, som er den samme verdien «R» regnet ut for oss tidligere.

Oppgave 3

R-kode

```
A=mean(lifetime)-2*sd(lifetime)#intervallets nedre grense
B=mean(lifetime)+2*sd(lifetime) #intervallets øvre grense
A #nedre grense
B #øvre grense
x=hist(lifetime,breaks=c(A,B,max(lifetime),include.lowest=F),plot=F)
attributes(x)
x$counts
```

- **Regn ut intervallgrensene i $(\bar{x} - 2 \times s, \bar{x} + 2 \times s)$.**

```
> A #nedre grense
[1] -2347.923
> B #øvre grense
[1] 7322.903
```

Nedre grense er $(\bar{x} - 2 \times s)$

Øver grense er $(\bar{x} + 2 \times s)$

- **Hvor mange levetider i datamaterialet er innenfor dette intervallet**

```
> x$counts
[1]   0 948  52
```

  - Det er 0 tilfeller som er lavere enn nedre grense
  - Det er 52 tilfeller som er større enn øvre grense
  - Det er 948 tilfeller som er mellom grensene
  - Det vil si at 94.8% er innenfor intervallene

- **Hvordan stemmer dette med Tsjebytsjevs regelen**

  Når vi regner med 2 i intervallgrensene følger det at minst 75% av verdiene skal være innenfor grensene. Dette stemmer i vårt tilfelle, siden det er 94,8% som er innenfor grensene.

  Hvis vi kjører samme «R-kode» flere ganger ser vi at den alltid vil være innenfor:

```
> x$counts
[1]   0 951  49
> x$counts
[1]   0 947  53
> x$counts
[1]   0 950  50
> x$counts
[1]   0 953  47
> x$counts
[1]   0 948  52
> x$counts
[1]   0 949  51
```