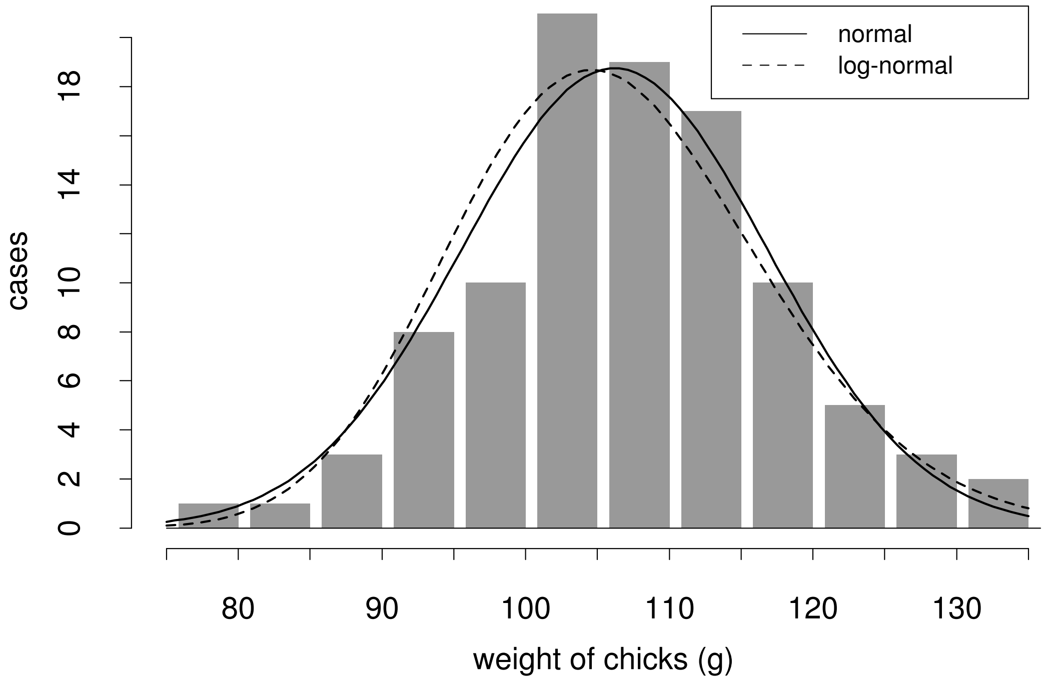
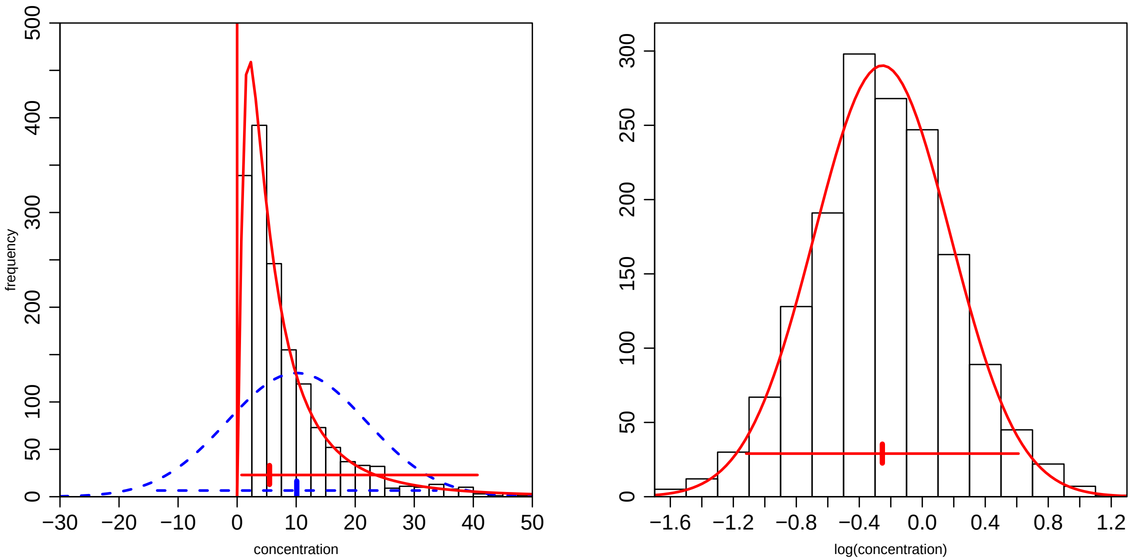
4. Verteilungen, Simulation

a Verteilung = “idealisiertes Histogramm”



b Werbespot für die Lognormal-Verteilung



c Verteilung

charakterisiert durch Dichte, kumulative Verteilung, Quantilfunktion.

Funktionen:

- Dichte-Funktion **dnorm(0.5, mean=0, sd=1)**

  ```
  [1] 0.35207
  ```

- Kumulative Verteilungsfunktion: **pnorm(1.96, mean=0, sd=1)**

  ```
  [1] 0.975
  ```

- Quantil-Funktion: **qnorm(c(0.25,0.975), mean=100, sd=10)**

  ```
  [1] 93.26 119.60
  ```

d Ebenso für andere Verteilungen.
Diskrete Verteilung $\longrightarrow$ Wahrsch. $P(X = k)$ aus **d...**

```
> dbinom(0:5, size=5, p=0.3)
[1] 0.168 0.360 0.308 0.132 0.028 0.0024
```

e Simulation

- Frage: Median ist Charakteristik einer Stichprobe.

  Daten zufällig, Annahme normalverteilt.

  $\longrightarrow$ Median ist Zufallsvariable. Verteilung des Medians?

- ... hängt vom Stichprobenumfang n ab. – $n = 7$.

  Modell: $X_1, X_2, ..., X_7 \sim \mathcal{N}(\mu, \sigma^2)$, unabhängig.

  Was heisst das?

- Zufallszahlen, normalverteilt, $\mu = 3$, $\sigma = 4$:

```
> ( ds1 <- rnorm(7,3,4) )
[1]  0.494  3.735 -0.343  9.381  4.318 -0.282  4.950
> median(ds1)
[1] 3.73
```

- Simulation: Diesen Vorgang 1000 Mal wiederholen
  und Resultat speichern.
  Vermeiden wir **for** -Schleife!

```
> ds <- matrix(rnorm(7*1000,3,4), nrow=1000, ncol=
> rs.median <- apply(ds, 1, median)
> hist(rs.median)
```

So kann man die Wahrscheinlichkeits-Rechnung vermeiden!

Empirischer Zugang zur Theorie!

$\longrightarrow$ Für beliebig komplizierte Funktionen (Schätzungen, ...)

kann man so die Verteilung erhalten.

f Bootstrap

Nachteil der gerade benützten Simulation:

Man muss das Modell (Verteilung) festlegen.

Bootstrap (“nicht-parametrischer” b.):

Benütze die Stichprobe = “empirische Verteilung” als Näherung für die wahre Verteilung der Daten.

“Empirische Verteilung”: Möglich sind die Werte der Beobachtungen,

Jede Beob hat die Wahrscheinlichkeit $1/n$ in jedem "Zug".

Zufall? $\longrightarrow$ Zufällige Auswahl aus den Daten, mit Zurücklegen,

Gleiche Anzahl wie Ausgangs-Stichprobe.

```
> dx <- dd[dd[,"stelle"]==1,"ersch"]
> showd(dx)
      r.1  r.2  r.3  ...  r.6  r.8 r.11 r.14
data 0.32 0.53 0.50      1.01 2.45 1.49 4.31
> mean(dx, trim=0.2)
[1] 1.77

> ( ts <- sample(14, replace=T) )
 [1]  4  1  9 12  4  2  6  1  6  6 13  1 11  7
> mean(dx[ts], trim=0.2)
[1] 1.29
```

Package 'boot'

```
> require(boot)
Loading required package: boot
> ?boot
f.mean <- function(x,i) { mean(x[i], trim=0.2) }
rb <- boot(dx, f.mean, R=1000)
plot(rb)
```