Eine Einführung in R

Silke Rolles

July 13, 2011

Allgemeine Informationen zu R

- R erhalten Sie kostenlos auf der Webseite des R Project.
  http://www.r-project.org/

Allgemeine Informationen zu R

- R erhalten Sie kostenlos auf der Webseite des R Project. http://www.r-project.org/
- Das Skript der "Einführung in R" vom SS 2009 sowie die R-Beispiele der "Einführung in die Wahrscheinlichkeitstheorie" vom WS 2009/10 finden Sie auf der Vorlesungsseite.

Allgemeine Informationen zu R

- R erhalten Sie kostenlos auf der Webseite des R Project.
  http://www.r-project.org/
- Das Skript der "Einführung in R" vom SS 2009 sowie die R-Beispiele der "Einführung in die Wahrscheinlichkeitstheorie" vom WS 2009/10 finden Sie auf der Vorlesungsseite.
- Einen Überblick der wichtigsten Befehle gibt die R Reference Card
  http://cran.r-project.org/doc/contrib/Short-refcard.pdf

Allgemeine Informationen zu R

- Programmstart: `R`

Allgemeine Informationen zu R

- Programmstart: `R`
- Programmende: `q()`

Allgemeine Informationen zu R

- Programmstart: `R`
- Programmende: `q()`
- Hilfefunktion: `help()`
  Hilfe zur Funktion rnorm: `help(rnorm)`

Verteilungen

Funktion	**Verteilung**
`beta()`	Beta-Verteilung
`binom()`	Binomial-Verteilung
`chisq()`	χ^2-Verteilung
`exp()`	Exponential-Verteilung
`f()`	F-Verteilung
`gamma()`	Gamma-Verteilung
`geom()`	Geometrische Verteilung
`hyper()`	Hypergeometrische Verteilung
`norm()`	Normalverteilung
`pois()`	Poissonverteilung
`t()`	t-Verteilung
`unif()`	Gleichverteilung

Verteilungen

Dem Namen der Funktion wird ein Buchstabe vorangestellt:

Verteilungen

Dem Namen der Funktion wird ein Buchstabe vorangestellt:

- `d` (density) Dichtefunktion

Verteilungen

Dem Namen der Funktion wird ein Buchstabe vorangestellt:

- `d` (density) Dichtefunktion
- `p` (probability) Verteilungsfunktion

Verteilungen

Dem Namen der Funktion wird ein Buchstabe vorangestellt:

- `d` (density) Dichtefunktion
- `p` (probability) Verteilungsfunktion
- `q` (quantiles) Quantile

Verteilungen

Dem Namen der Funktion wird ein Buchstabe vorangestellt:

- `d` (density) Dichtefunktion
- `p` (probability) Verteilungsfunktion
- `q` (quantiles) Quantile
- `r` (random) Pseudo-Zufallszahlen

Eine Dichtefunktion plotten

Wir plotten die Dichte der $N(0,1)$-Verteilung:

```
x <- seq(-5,5, by=0.005)
plot(x,dnorm(x), type="l", xlab="x", ylab="f(x)",
main="Dichte f der Standardnormalverteilung")
```

Eine Dichtefunktion plotten

Wir plotten die Dichte der $N(0,1)$-Verteilung:

```
x <- seq(-5,5, by=0.005)
plot(x,dnorm(x), type="l", xlab="x", ylab="f(x)",
main="Dichte f der Standardnormalverteilung")
```

Dabei bedeutet:

- `dnorm(x)`: Dichte der N(0,1)-Verteilung an der Stelle x
  Dargestellt wird $x \mapsto \mathit{dnorm}(x)$ für $x \in [-5,5] \cap 0.005\mathbb{Z}$.
- `type=l`: Linien plotten
- `xlab`: Titel der x-Achse
- `ylab`: Titel der y-Achse
- `main`: Titel der Grafik

Die Verteilungsfunktion

Wir plotten die Verteilungsfunktion Φ der $N(0,1)$-Verteilung:

```
x <- seq(-5,5, by=0.01)
plot(x,pnorm(x), type="l", xlab="x", ylab="Phi(x)",
main="Verteilungsfunktion Phi der
Standardnormalverteilung")
```

Dabei ist `pnorm(x)`$=\Phi(x)$.
Dargestellt wird $x \mapsto \Phi(x)$ für $x \in [-5,5] \cap 0.01\mathbb{Z}$.

Quantile

- Das α-Quantil der $N(0,1)$-Verteilung bestimmt man mit `qnorm(`α`)`

Quantile

- Das α-Quantil der $N(0,1)$-Verteilung bestimmt man mit `qnorm(`α`)`
- Zum Beispiel berechnet man das 95%-Quantil mit dem Kommando `qnorm(0.95)`

Die Quantilsfunktion

Wir plotten die Quantilsfunktion z der $N(0,1)$-Verteilung:

```
x< -seq(0,1, by=0.005)
plot(x,qnorm(x), type="l", xlab="x", ylab="z(x)",
main="Quantilsfunktion z der
Standardnormalverteilung")
```

Dabei ist `qnorm(x)`$=z(x)$ die Quantilsfunktion an der Stelle x.
Dargestellt wird $x \mapsto z(x)$ für $x \in (0,1) \cap 0.005\mathbb{Z}$.

Empirisches Mittel und empirische Varianz

- Wir erzeugen einen Vektor mit den Komponenten 3,7,5:
  `x<- c(3,7,5)`

Empirisches Mittel und empirische Varianz

- Wir erzeugen einen Vektor mit den Komponenten 3,7,5:
  `x< -c(3,7,5)`
- Empirisches Mittel von x:

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

```
mean(x)
```

Empirisches Mittel und empirische Varianz

- Wir erzeugen einen Vektor mit den Komponenten 3,7,5:
  `x< −c(3,7,5)`
- Empirisches Mittel von x:

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

`mean(x)`

- Empirische Varianz von x:

$$s_x^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})^2$$

`var(x)`

Empirische Kovarianz

Empirische Kovarianz von x und y:

$$s_{x,y} = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})$$

```
cov(x,y)
```

Erzeugung von Zufallszahlen

- Man kann n $N(\mu, \sigma^2)$-verteilte Zufallsvariablen mit dem Kommando

  `rnorm(n,mean=`μ`,sd=`σ`)`

  erzeugen.

Erzeugung von Zufallszahlen

- Man kann n $N(\mu, \sigma^2)$-verteilte Zufallsvariablen mit dem Kommando

  `rnorm(n,mean=`μ`,sd=`σ`)`

  erzeugen.

- Beispiel: Erzeugung von 100 $N(0, 1)$-verteilten Zufallsvariablen

```
x< -rnorm(100,mean=0,sd=1)
mean(x)
var(x)
```

Lineare Regression

Modell:

$$Y_i = \gamma_0 + \gamma_1 x_i + \sigma \xi_i,\ 1 \leq i \leq n.$$

Lineare Regression

Modell:

$$Y_i = \gamma_0 + \gamma_1 x_i + \sigma \xi_i,\ 1 \leq i \leq n.$$

Dabei sind

- $x_i \in \mathbb{R}$ bekannt $(1 \leq i \leq n)$

Lineare Regression

Modell:

$$Y_i = \gamma_0 + \gamma_1 x_i + \sigma \xi_i,\ 1 \leq i \leq n.$$

Dabei sind

- $x_i \in \mathbb{R}$ bekannt $(1 \leq i \leq n)$
- $\gamma_0, \gamma_1 \in \mathbb{R}$, $\sigma > 0$ unbekannt

Lineare Regression

Modell:

$$Y_i = \gamma_0 + \gamma_1 x_i + \sigma \xi_i,\ 1 \leq i \leq n.$$

Dabei sind

- $x_i \in \mathbb{R}$ bekannt $(1 \leq i \leq n)$
- $\gamma_0, \gamma_1 \in \mathbb{R}$, $\sigma > 0$ unbekannt
- ξ_i, $1 \leq i \leq n$, unabhängig mit $E[\xi_i] = 0$ und $\mathrm{Var}(\xi_i) = 1$.

Lineare Regression

Modell: $Y_i = \gamma_0 + \gamma_1 x_i + \sigma\xi_i$, $1 \leq i \leq n$.

Erwartungstreue Schätzer für die unbekannten Koeffizienten γ_0 und γ_1 sind:

$$\hat{\gamma}_1 = \frac{S_{Y,x}}{s_x^2}$$

$$\hat{\gamma}_0 = \overline{Y} - \hat{\gamma}_1 \overline{x} = \overline{Y} - \frac{S_{Y,x}}{s_x^2} \cdot \overline{x}$$

mit

$$\overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i, \quad \overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{empirische Mittel}$$

$$S_{Y,x} = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \overline{Y})(x_i - \overline{x}) \quad \text{empirische Kovarianz}$$

$$s_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2 \quad \text{empirische Varianz}$$

Lineare Regression

Ein einfaches Beispiel:

```
x <- c(1:10)
xi <- rnorm(10)
y <- 1+2*x+xi
linearesmodell<-lm(y~x)
coef(linearesmodell)
```

Lineare Regression

Ein einfaches Beispiel:

```
x <- c(1:10)
xi <- rnorm(10)
y <- 1+2*x+xi
linearesmodell<-lm(y~x)
coef(linearesmodell)
```

Das bedeutet:

- $x = (1, 2, 3, \ldots, 10)$

Lineare Regression

Ein einfaches Beispiel:

```
x <- c(1:10)
xi <- rnorm(10)
y <- 1+2*x+xi
linearesmodell<-lm(y~x)
coef(linearesmodell)
```

Das bedeutet:

- $x = (1, 2, 3, \ldots, 10)$
- $y_i = 1 + 2 \cdot x_i + \xi_i$ mit unabhängigen $\xi_i \sim N(0, 1)$.

Lineare Regression

Ein einfaches Beispiel:

```
x <- c(1:10)
xi <- rnorm(10)
y <- 1+2*x+xi
linearesmodell<-lm(y~x)
coef(linearesmodell)
```

Das bedeutet:

- $x = (1, 2, 3, \ldots, 10)$
- $y_i = 1 + 2 \cdot x_i + \xi_i$ mit unabhängigen $\xi_i \sim N(0, 1)$.
- `linearesmodell<-lm(y~ x)` besagt, dass zwischen y und x ein linearer Zusammenhang wie oben beschrieben besteht.

Lineare Regression

Ein einfaches Beispiel:

```
x <- c(1:10)
xi <- rnorm(10)
y <- 1+2*x+xi
linearesmodell<-lm(y~x)
coef(linearesmodell)
```

Das bedeutet:

- $x = (1, 2, 3, \ldots, 10)$
- $y_i = 1 + 2 \cdot x_i + \xi_i$ mit unabhängigen $\xi_i \sim N(0, 1)$.
- `linearesmodell<-lm(y~ x)` besagt, dass zwischen y und x ein linearer Zusammenhang wie oben beschrieben besteht.
- `coef(linearesmodell)` liefert $\hat{\gamma}_0$ und $\hat{\gamma}_1$.

Lineare Regression

- `fitted(linearesmodell)` liefert die Punkte $(y_i, \hat{\gamma}_0 + \hat{\gamma}_1 x_i)$ entlang der Regressionsgeraden.

Lineare Regression

- `fitted(linearesmodell)` liefert die Punkte $(y_i, \hat{\gamma}_0 + \hat{\gamma}_1 x_i)$ entlang der Regressionsgeraden.
- Man bezeichnet $y_i - (\hat{\gamma}_0 + \hat{\gamma}_1 x_i)$, $1 \leq i \leq n$, also die Messwerte minus die gefitteten Werte, als **Residuen**.

Lineare Regression

- `fitted(linearesmodell)` liefert die Punkte $(y_i, \hat{\gamma}_0 + \hat{\gamma}_1 x_i)$ entlang der Regressionsgeraden.
- Man bezeichnet $y_i - (\hat{\gamma}_0 + \hat{\gamma}_1 x_i)$, $1 \leq i \leq n$, also die Messwerte minus die gefitteten Werte, als **Residuen**.
- `residuals(linearesmodell)` liefert also dasselbe wie

```
f <- fitted(linearesmodell)
y-f
```

Lineare Regression

- `fitted(linearesmodell)` liefert die Punkte $(y_i, \hat{\gamma}_0 + \hat{\gamma}_1 x_i)$ entlang der Regressionsgeraden.
- Man bezeichnet $y_i - (\hat{\gamma}_0 + \hat{\gamma}_1 x_i)$, $1 \leq i \leq n$, also die Messwerte minus die gefitteten Werte, als **Residuen**.
- `residuals(linearesmodell)` liefert also dasselbe wie

```
f <- fitted(linearesmodell)
y-f
```

- `plot(x,y)` plottet die Daten

Lineare Regression

- `fitted(linearesmodell)` liefert die Punkte $(y_i, \hat{\gamma}_0 + \hat{\gamma}_1 x_i)$ entlang der Regressionsgeraden.
- Man bezeichnet $y_i - (\hat{\gamma}_0 + \hat{\gamma}_1 x_i)$, $1 \leq i \leq n$, also die Messwerte minus die gefitteten Werte, als **Residuen**.
- `residuals(linearesmodell)` liefert also dasselbe wie

```
f <- fitted(linearesmodell)
y-f
```

- `plot(x,y)` plottet die Daten
- `abline(linearesmodell)` fügt die Regressionsgerade hinzu

Hypothesentests und Konfidenzintervalle

```
x <- rnorm(20,mean=0,sd=5)
t <- t.test(x)
```

führt einen zweiseitigen t-Test $H_0 : \mu = 0$ gegen $\mu \neq 0$ aus.

Hypothesentests und Konfidenzintervalle

```
x <- rnorm(20,mean=0,sd=5)
t <- t.test(x)
```

führt einen zweiseitigen t-Test $H_0 : \mu = 0$ gegen $\mu \neq 0$ aus.
Beim Output ist

- `t`: Teststatistik
- `df`: degrees of freedom, also der Parameter n der t_n-Verteilung
- `p.value`: p-Wert
- `conf.level=0.95` bedeutet, dass ein 95%-Konfidenzintervall bestimmt wird.

Hypothesentests und Konfidenzintervalle

```
x <- rnorm(20,mean=0,sd=5)
t <- t.test(x,alternative="less",conf.level=0.99)
```

führt einen einseitigen t-Test $H_0 : \mu = 0$ gegen $\mu < 0$ aus.
Es wird ein 99%-Konfidenzintervall bestimmt.