Einführung in R

Dr. Mike Kühne

Technische Universität Dresden
Institut für Soziologie

Übersicht

Übersicht

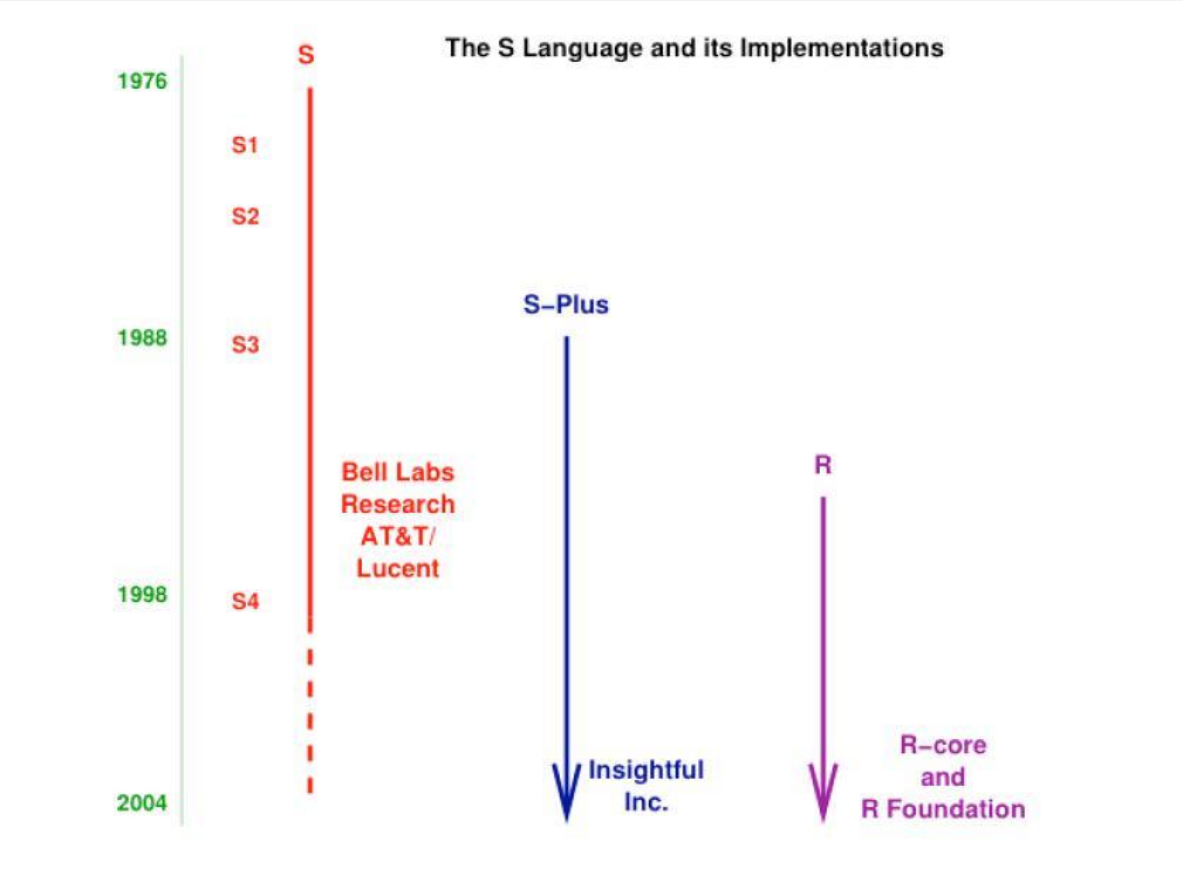
Was ist R?

- Flexible Programmiersprache
- open source software
- Software-System, was auf der Sprache R aufbaut

Geschichte

- R ist eine weitere Entwicklung, die auf der Sprache S aufbaut.
- S ist ein Sprachstandard für das Arbeiten mit Daten.
- S wurde seit den 1970ern in den Bell Laboratories von Rick Becker, John Chambers und Alan Wilks entwickelt.
- Seit ca. 1988 gibt es eine kommerzielle Variante Splus, welche ebenfalls den Sprachstandard S implementiert.
- Ross Ihaka und Robert Gentleman entwickeln für die Lehre Anfang der 1990er die Sprache R .
- Seit 1995 steht R unter der GPL (GNU General Public License).

Geschichte



Quelle: Vortrag von John M. Chambers 2006

Übersicht

Vorteile

Vorteile

- Open Source Software (Code und Software sind frei zugänglich)
- Professioneller Support u.a. über die Mailing-Liste
- Professionelle Dokumentation in den Manuals
- Professionelle Grafikerstellung
- Integration in LaTeX

Nachteile

Nachteile

- Die Standardinstallation besitzt nur eine eingeschränkte Benutzeroberfläche (GUI)
- R ist keine Interpretersprache. Der Programmiercode wird nicht kompiliert, sondern zur Laufzeit interpretiert. Bei sehr umfangreichen Rechenoperationen und sehr großen Datensätzen, kann es unter Windows zu zeitlichen Verzögerungen kommen.

Übersicht

Manuals

Bei einer Standardinstallation im Ordner: $C:\backslash Programme\backslash r\backslash R-2.7.2\backslash doc\backslash manual$

- R-FAQ.html
- R-intro.pdf (englischsprachige Einführung)
- R-data.pdf (Einlesen und Ausgeben anderer Datenformate)

Im Internet: http://cran.r-project.org/other-docs.html

- Dokumente unter 100 Seiten
- Dokumente über 100 Seiten
- nicht englischsprachige Dokumente

Internet

Internetsuche

- Das Archiv der Mailingliste: http://tolstoy.newcastle.edu.au/R/
- Kombinationssuche verschiedener Quellen: http://www.rseek.org
- Ein Wiki unter: http://wiki.r-project.org

Einführungen

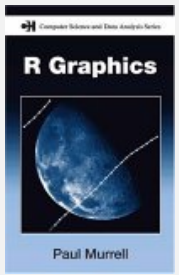
Peter Dalgaard (2008): Introductory Statistics with R (Statistics and Computing). Springer

Dolic, Dubravko (2003): Statistik mit R: Einführung für Wirtschafts- und Sozialwissenschaftler. Oldenbourg

Ligges, Uwe (2008): Programmieren mit R. Springer

Grafiken

Murrell, Paul (2005): R Graphics. Chapman & Hall



Webseite zum Buch:
http://www.stat.auckland.ac.nz/ paul/RGraphics/rgraphics.html

Statistik

Venables, William and Ripley, Brian (2003): Modern Applied Statistics with S. Springer

Sachs, Lothar und Hedderich, Jürgen (2006): Angewandte Statistik–Methodensammlung mit R. Springer

Übersicht

Konventionen

- input ">"
- output "[1]"
- unvollständige inputs Sollte ein Ausdruck am Ende einer Zeile syntaktisch nicht vollständig sein, erscheint ein "+"
- Kommentare (#).

```
> 1 + 1
[1] 2
> 1 +
+
> 100
[1] 101
> # Das ist ein Kommentar
```

Operatoren der Grundrechenoperationen

Operatoren: `+ - * / ^`

```
> 12+13
[1] 25
> 121-11
[1] 110
> 4*5
[1] 20
> 5/35
[1] 0.1428571
> 2^3
[1] 8
```

Weitere Regeln

```
> 10*3+1 # Punkt- vor Strichrechnung
[1] 21
> sqrt(16) # einfache Quadratwurzel
[1] 4
> exp(2) # Exponentialfunktion
[1] 7.389056
> sin(15) # Trigonometrische Funktionen
[1] 0.6502878
```

Dateneingabe

Es existieren zahlreiche Möglichkeiten der Dateneingabe. Beispielsweise im Erzeugen eines **Vektors** (als endliche Folge von einzelnen **Elementen**) mit der Funktion `c()`.

```
> vector.1 <- c(1,2,3)
```

Auf alle Objekte in R kann anhand ihrer Namen zugegriffen werden.

```
> vector.1
[1] 1 2 3
```

Berechnung BMI

Zuerst werden 2 Vector erstellt[1]:

```
> Gewicht <- c(60, 72, 57, 90, 95, 72)
> Groesse <- c(1.75, 1.80, 1.65, 1.90, 1.74, 1.91)
```

Anschließend Berechnung des Body Mass Index (BMI):

```
> BMI <- Gewicht/ Groesse^2
> BMI
[1] 19.59184 22.22222 20.93664 24.93075 31.37799 19.73630
```

[1] Beispiel aus Dalgaard (2008)

Berechnung eines Mittelwerts

Der arithmetische Mittelwert $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i = \frac{x_1 + x_2 + \ldots + x_n}{n}$ einer Variable lässt sich über Umwege unter Rückgriff auf die Funktionen `sum()` und `length()` berechnen.

```
> sum(Gewicht)
[1] 446
> length(Gewicht)
[1] 6
> sum(Gewicht)/length(Gewicht)
[1] 74.33333
```

Empirische Standardabweichung

Die Standardabweichung von Stichprobendaten

$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})^2}$ ergibt sich aus:

```
> mittelwert <- sum(Gewicht)/length(Gewicht)
> Gewicht-mittelwert
[1] -14.333333  -2.333333 -17.333333  15.666667  20.666667
[6] -2.333333
> sqrt(sum((Gewicht - mittelwert)^2)/ (length(Gewicht)-1))
[1] 15.42293
```

Als Statistikprogramm verfügt R über feste Funktionen für Mittelwertberechnung und Ermittlung der Standardabweichung.

```
> mean(Gewicht)
[1] 74.33333
> sd(Gewicht)
[1] 15.42293
```

Histogramm

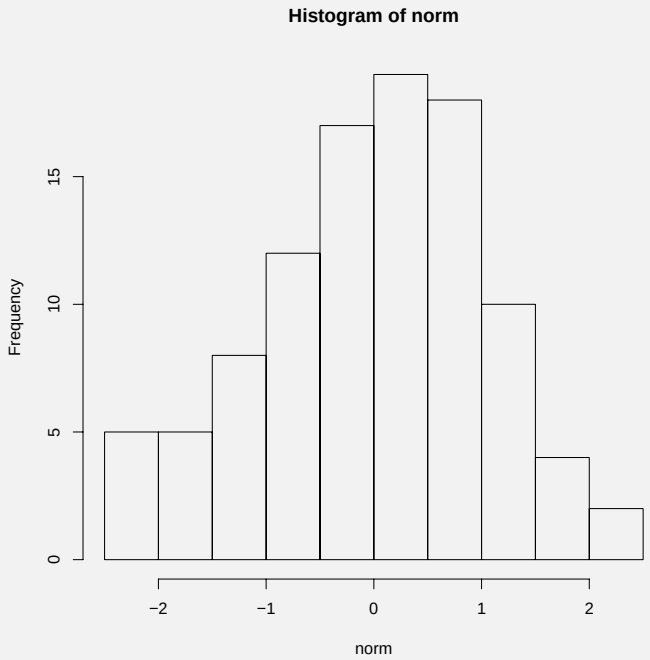
Mit der Funktion `hist()` kann man ein Histogramm anfordern.

```
> hist(Gewicht)
```

Funktionen

Wir können mit der Funktion `rnorm()` eine normalverteilte Zufallsvariable mit 100 Fällen erzeugen und wiederum ein Histogramm anfordern.

```
> norm <- rnorm(100)
> hist(norm)
```



Funktionen

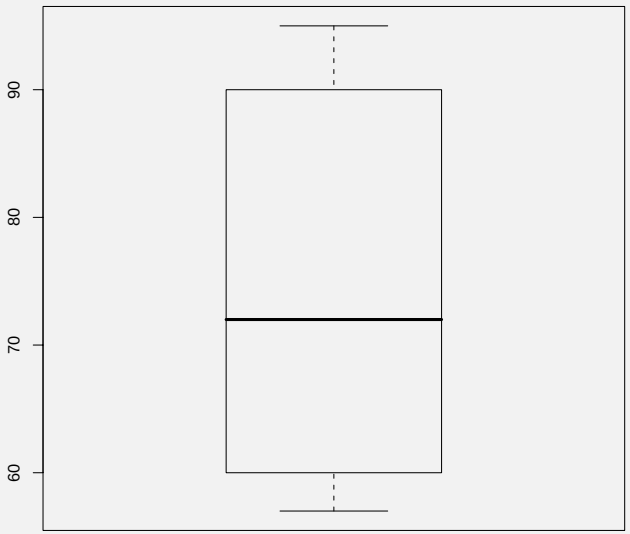
Es stehen zahlreiche andere Funktionen zur Verfügung. Unter anderem zur Erzeugung von einfachen Tabellen...

```
> table(Gewicht)
57 60 72 90 95
 1  1  2  1  1
```

Boxplot

...und Boxplots:

```
> boxplot(Gewicht)
```



Multivariate Statistik

Annahme: Das Gewicht (AV) einer Person kann auf die Größe (UV) zurückgeführt werden (Regression).

```
lm(Gewicht~Groesse)

Call:
lm(formula = Gewicht ~ Groesse)

Coefficients:
(Intercept)      Groesse
     -46.34        67.35
```

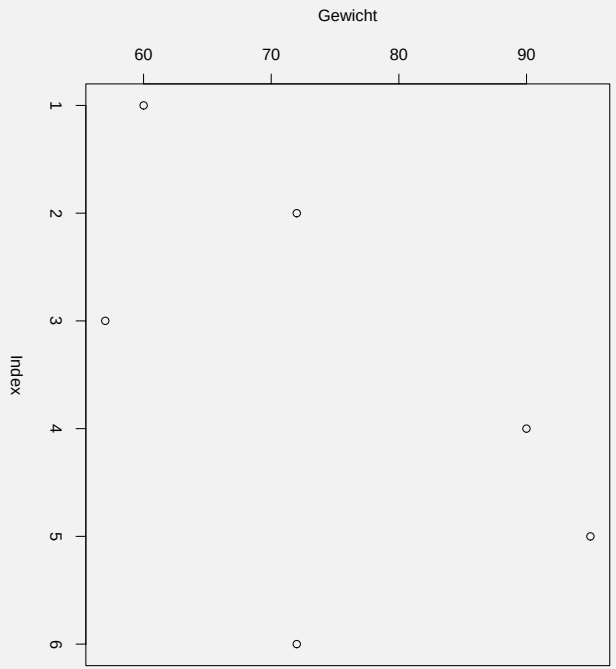
Weitere Informationen kann man mit der Funktion `summary()` anfordern:

```
summary(lm(Gewicht~Groesse))

Call:
lm(formula = Gewicht ~ Groesse)

Residuals:
      1       2       3       4       5       6
-11.527  -2.895  -7.792   8.370  24.147 -10.303

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -46.34     124.02  -0.374    0.728
Groesse        67.35      69.13   0.974    0.385


Residual standard error: 15.5 on 4 degrees of freedom Multiple
R-Squared: 0.1918,     Adjusted R-squared: -0.01027 F-statistic:
0.9492 on 1 and 4 DF,  p-value: 0.3851
```

Einfache Grafiken

Simple Grafiken werden mit der Funktion `plot()` erstellt:

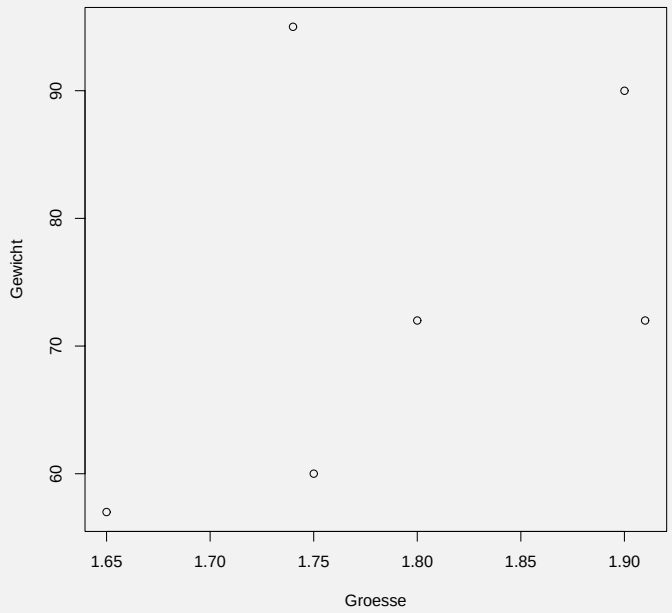
```
> plot(Gewicht)
```



Streudiagramme

Streudiagramme können mit `plot(x, y)` erzeugt werden:

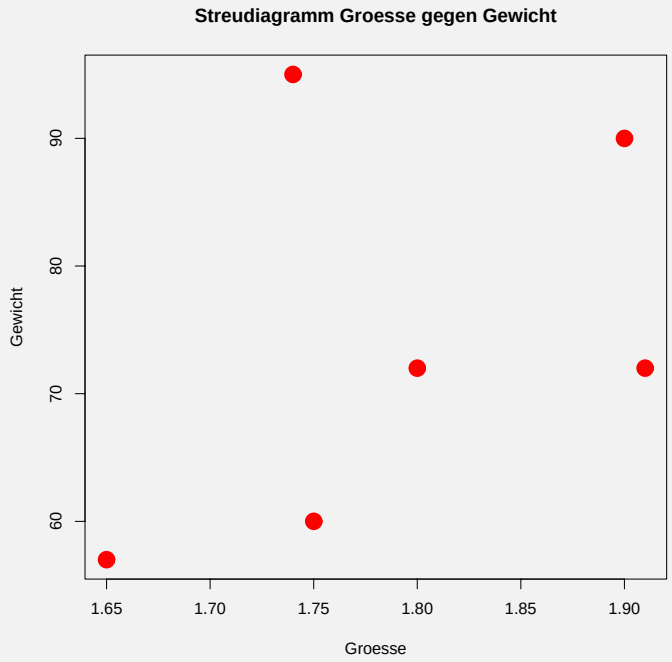
```
> plot(Groesse, Gewicht)
```



Spezifikationen der Funktionen

Mit einfachen Spezifikationen kann jede Grafik angepasst werden:

```
> plot(Groesse, Gewicht, type = "p", col = "red", lwd=10,
+ main = "Streudiagramm Groesse gegen Gewicht", xlab="Groesse",
+ ylab="Gewicht")
```



Komplexe Grafiken

R eignet sich hervorragend zur Erstellung von Grafiken. Zahlreiche Beispiele findet man im Internet:

- Die Grafik auf der Startseite des R Projektes: http://www.r-project.org/
- Die Beispiel aus Murrell (2005): http://www.stat.auckland.ac.nz/ paul/RGraphics/rgraphics.html
- Die R Graph Gallery: http://addictedtor.free.fr/graphiques/RGraphGallery.php?graph=30

Programmiersprache

Grundsätzliche Aufbau einer Funktion

```
fname<-function(Argumente)
{
Koerper der Funktion
return(Ergebnis)
}
```

Beispiel: Berechnung des Mittelwerts

```
> func.mw<-function(x){
+   mw<-sum(x)/length(x)
+   return(mw)
+   }
```

Anwendung 1: Berechnung des Mittelwerts

```
func.mw(Gewicht)
74.33333
```

Vorteil von Funktionen: Konservierbarkeit von Algorithmen

Anwendung 2: Wie oft fällt die Zahl 6, wenn man den Würfel 1000 mal wirft?

```
>  wuerfel <- function( N, Augenzahl)
+ {
+       # Generieren von N Würfen eines Würfels mit 6 Seiten:
+   x <-sample(1:6, N, replace = TRUE)
+       # Zählen, wie oft die ugenzahl "Augenzahl" vorkommt:
+       sum(x == Augenzahl)
+ }
```

```
> wuerfel(1000,6)
[1] 179 # ist abhängig von den Zufallszahlen
```

Angepasstes Beispiel aus Ligges (2005, S. 100)