LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Statistische Software (R-Vertiefung)

Paul Fink, M.Sc.

Institut für Statistik
Ludwig-Maximilians-Universität München

Operationen mit Zeichenketten

Formatierung und Ausgabe

Wir beginnen mit der Formatierung und Ausgabe von Zeichenketten, da wir manche dieser Befehle benötigen, um uns das Ergebnis von bestimmten Operationen auf Zeichenketten anzusehen.

Die wichtigsten Befehle zur Ausgabe und Formatierung sind `print`, `cat` und `format`. `print()` ist dabei ein sog. generischer Befehl, der für jede Klasse (zur Erinnerung: R ist objektorientiert) zur Verfügung steht.

Beispiele

```
> X <- matrix(data = 4:9, nrow = 3, ncol = 2, byrow = TRUE)
> print(X)
     [,1] [,2]
[1,]    4    5
[2,]    6    7
[3,]    8    9
```


Formatierung und Ausgabe

Damit wird die Matrix in die Konsole oder Ausgabedatei (Batch-Modus) geschrieben. Die Option `digits` erlaubt beispielsweise die Angabe der auszugebenden Stellen.

```
> print( sqrt(2) )
[1] 1.414214
> print( sqrt(2), digits=5 )
[1] 1.4142
> print( sqrt(2), digits=16 )
[1] 1.414213562373095
```

Die `format` erlaubt eine umfangreichere Formatierung.

```
> print( format( sqrt(2), digits=10, nsmall=15 ) )
[1] "1.414213562373095"
> print( format( 0.5, digits=10, nsmall=15 ) )
[1] "0.500000000000000"
> print( format( 0.5, digits=10, nsmall=15 ), quote=FALSE )
[1] 0.500000000000000
```


Formatierung und Ausgabe

Die Funktion `cat()` wandelt alle übergebenen Argumente in Zeichenketten um, konkateniert diese und gibt die gesamte Zeichenkette auf der Konsole aus (die Funktion `paste()` fügt ebenfalls Zeichenketten zusammen, das Ergebnis kann jedoch im Gegensatz zu `cat` einer Variablen zugewiesen werden). Der Parameter `sep` ist die Zeichenkette, die als Trennung zwischen den Eingabe-Zeichenketten dient. Ein eventuell gewünschter Zeilenumbruch muss durch ein `"\n"` (newline) herbeigeführt werden.

```
> d <- date()
> cat("Heute ist:", d, "\n" )
Heute ist: Mon Jun  3 17:18:47 2013
> x <- 7
> cat("Das Quadrat von", x, "ist", x^2, "!\n")
Das Quadrat von 7 ist 49 !
> cat("Die Wurzel von", x, "ist ungefaehr", format(sqrt(x), digits=3), "\n" )
Die Wurzel von 7 ist ungefaehr 2.65
```

Operationen mit Zeichenketten

R bietet eine Reihe von Funktionen zur Manipulation von Zeichenketten zur Verfügung.

- Zunächst erlaubt der `paste` Befehl das Zusammenfügen von Zeichenketten und Zahlen. Beispiel:

```
> paste("Aufgabe", 1, "a)", sep="_")
[1] "Aufgabe_1_a)"
> x <- paste("Aufgabe", 1:5, sep="_")
> x
[1] "Aufgabe_1" "Aufgabe_2" "Aufgabe_3" "Aufgabe_4" "Aufgabe_5"
> x[1]
[1] "Aufgabe_1"
> x[2]
[1] "Aufgabe_2"
> x[5]
[1] "Aufgabe_5"
```

  D.h., `x` ist ein Array von Zeichenketten. Dies erlaubt beispielsweise das Generieren von Dateinamen in Programmen,

Operationen mit Zeichenketten

wenn mehrere Ausgabedateien gewünscht werden. Verwendet man den Parameter `collapse`, so ist das Ergebnis kein Array von Zeichenketten mehr, sondern nur noch eine einzelne Zeichenkette:

```
> x <- paste("Aufgabe", 1:5, sep = "_", collapse = "*")
> x[1]
[1] "Aufgabe_1*Aufgabe_2*Aufgabe_3*Aufgabe_4*Aufgabe_5"
```

`paste0` ist eine Abkürzung für `paste( , sep="")`

Operationen mit Zeichenketten

- Mit dem Befehl `strsplit` kann man eine Zeichenkette in Teile zerlegen. Als "Split" kann sowohl ein einzelnes Zeichen, als auch eine Zeichenkette definiert werde:

```
> x <- "Die&!Syntax&!von&!paste&!findet!&man!&wie!&immer!&in der Online-Hilfe"
> strsplit(x,"!")
[[1]]
[1] "Die&"                "Syntax&"             "von&"
[4] "paste&"              "findet"              "&man"
[7] "&wie"                "&immer"              "&in der Online-Hilfe"
> strsplit(x,"&!")
[[1]]
[1] "Die"
[2] "Syntax"
[3] "von"
[4] "paste"
[5] "findet!&man!&wie!&immer!&in der Online-Hilfe"
> l1 <- strsplit(x,"!&")
> l1
[[1]]
[1] "Die&!Syntax&!von&!paste&!findet" "man"
[3] "wie"                             "immer"
[5] "in der Online-Hilfe"
```


Operationen mit Zeichenketten

Man beachte den Zugriff auf die einzelnen Teile:

```
> l1[[1]][1]
[1] "Die&!Syntax&!von&!paste&!findet"
> l1[[1]][2]
[1] "man"
> l1[[1]][5]
[1] "in der Online-Hilfe"
```

- Neben `strsplit` existieren noch eine Reihe von Befehlen, die zum Suchen und Ersetzen in Zeichenketten verwendet werden können. Beipiele:

Operationen mit Zeichenketten

```
> x <- "R Kurs Sommersemester 2012"
> y <- "Anzahl der Kursteilnehmer: 90"
> nchar(x)
[1] 26
> sub("90", "55", y)
[1] "Anzahl der Kursteilnehmer: 55"
> toupper(x)
[1] "R KURS SOMMERSEMESTER 2012"
> grep("em", c(x,y) )
[1] 1
> # "em" ist im 1-ten Element, also in x
> ep <- parse(
+   text = "X <- matrix(data = runif(n = 4), nrow = 2, ncol = 2, byrow = TRUE)")
> eval(ep)
> X
          [,1]      [,2]
[1,] 0.6766134 0.5784806
[2,] 0.5319551 0.1599175
```

Die letzten 3 Befehle zeigen, wie eine Zeichenkette, welche einen Ausdruck in R enthält, zur Ausführung gebracht werden kann.

Operationen mit Zeichenketten

Die folgende Tabelle gibt eine Übersicht

`cat()`	nur für Ausgabe in Konsole und Dateien
`deparse()`	Ausdruck in Zeichenfolge konvertieren
`formatC()`	Formatierung im C Stil
`grep()`	Suchen von Zeichenfolgen
`match()` , `pmatch()` `charmatch()`	Suchen von (Teil-)Zeichenketten in Zeichenketten
`nchar()`	Anzahl Zeichen in einer Zeichenkette
`parse()`	Konvertierung in einen Ausdruck
`paste()`, `paste0()`	Zusammensetzen von Zeichenketten
`strsplit()`	Zerlegen von Zeichenketten
`sub()` , `gsub()`	Ersetzen von (Teil-)Zeichenfolgen
`substring()`	Ausgabe und Ersetzung von Teil-Zeichenfolgen
`toupper()` , `tolower()`	Umwandlung in Groß- bzw. Kleinbuchstaben
`\t`, `\n`	Tabstopp-Einrückung, Zeilenumbruch

Operationen mit Zeichenketten

- Das Suchen und Ersetzen von Zeichenketten kann direkt oder über sogenannte reguläre Ausdrücke erfolgen.

```
> x <- "Heute leben wir\t im Jahr\t 2031\n"
> cat(x)
Heute leben wir         im Jahr         2031
> x
[1] "Heute leben wir\t im Jahr\t 2031\n"
```

Unterschied zwischen `sub` und `gsub`

```
> sub("\t", "*", x)
[1] "Heute leben wir* im Jahr\t 2031\n"
> gsub("\t", "*", x)
[1] "Heute leben wir* im Jahr* 2031\n"
```

Fester vs. regulärer Ausdruck

```
> gsub("2031", "2013", x)
[1] "Heute leben wir\t im Jahr\t 2013\n"
> gsub("\\d{4}", "2013", x)
[1] "Heute leben wir\t im Jahr\t 2013\n"
```