LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

# Statistische Software (R)

**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 = 2)
[1] 1.4
> print(sqrt(2), digits = 4)
[1] 1.414
> print(sqrt(2) + 100, digits = 2)
[1] 101
> print(sqrt(2) + 100, digits = 4)
[1] 101.4
```

Die Funktion `format` erlaubt eine umfangreichere Formatierung.

```
> print(format(sqrt(2), digits = 3, nsmall = 5))
[1] "1.41421"
> print(format(0.5, digits = 1, nsmall = 5))
[1] "0.50000"
> print(format(0.5, digits = 1, nsmall = 5), quote = FALSE)
[1] 0.50000
```



## Funktion cat

Die Funktion `cat()` wandelt alle übergebenen Argumente in Zeichenketten um, konkateniert diese und gibt die gesamte Zeichenkette auf der Konsole aus. 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 (automatisch in RStudio).

```
> 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
```

**Achtung**: `cat` hat als Rückgabewert immer `NULL`!

```
> res <- cat("Das Quadrat von", x, "ist", x^2, "!\n")
Das Quadrat von 7 ist 49 !
> res
NULL
```

# Operationen mit Zeichenketten

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

- Funktion `paste` zum Zusammenfügen von Zeichenketten und Zahlen.

  Beispiel (arbeitet Vektorwertig!):

```
> 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"
```

  `sep` gibt an, was zwischen die einzelnen Objekte gehängt werden soll.

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



# Operationen mit Zeichenketten

Über das Argument `collapse`, kann man einen Vektor von Zeichenketten in eine einzige zusammenführen.

```
> x
[1] "Aufgabe_1" "Aufgabe_2" "Aufgabe_3" "Aufgabe_4" "Aufgabe_5"
> xc <- paste(x, collapse = "*")
> xc
[1] "Aufgabe_1*Aufgabe_2*Aufgabe_3*Aufgabe_4*Aufgabe_5"
```



# Operationen mit Zeichenketten

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

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



# Operationen mit Zeichenketten

**Achtung**: Es liefert eine Liste zurück!

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

**Achtung:** Die "Split"-Zeichenkette wird standardmäßig als sog. *regulärer Ausdruck* interpretiert!

Mit `fixed = TRUE` wird das verhindert.

```
> y <- "Dieser?Ausdruck?geht?schief"
> strsplit(y, "?")
[[1]]
 [1] "D" "i" "e" "s" "e" "r" "?" "A" "u" "s" "d" "r" "u" "c" "k" "?" "g" "e" "h"
[20] "t" "?" "s" "c" "h" "i" "e" "f"
> strsplit(y, "?", fixed = TRUE)
[[1]]
[1] "Dieser"   "Ausdruck" "geht"     "schief"
```

# Operationen mit Zeichenketten

- Weitere nützliche Funktionen

```
> x <- "R Kurs Sommersemester 2014"
> nchar(x) # Anzahl Zeichen in der Zeichenkette
[1] 26
> toupper(x) # Alles in Grossbuchstaben
[1] "R KURS SOMMERSEMESTER 2014"
> ep <- parse(
+ text = "mat <- matrix(data = runif(n = 4), nrow = 2, ncol = 2, byrow = TRUE)")
> eval(ep)
> mat
          [,1]      [,2]
[1,] 0.5497525 0.8327762
[2,] 0.4113681 0.2635774
```

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



# Operationen mit Zeichenketten

- Suchen und Ersetzen in Zeichenketten

```
> x <- "R Kurs Sommersemester 2014"
> y <- "Anzahl der Kursteilnehmer: 900"
> sub("900", "90", y)
[1] "Anzahl der Kursteilnehmer: 90"
> grep("em", c(x,y) )
[1] 1
> # "em" ist im 1-ten Element, also in x

> x <- "Heute leben wir\t im Jahr\t 2041\n"
> cat(x)
Heute leben wir         im Jahr         2041
> x
[1] "Heute leben wir\t im Jahr\t 2041\n"
```

Unterschied zwischen `sub` und `gsub`

```
> sub("\t", "*", x)
[1] "Heute leben wir* im Jahr\t 2041\n"
> gsub("\t", "*", x)
[1] "Heute leben wir* im Jahr* 2041\n"
```



# 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 |
| `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.

Fester vs. regulärer Ausdruck

```
> gsub("2041", "2014", x, fixed = TRUE)
[1] "Heute leben wir\t im Jahr\t 2014\n"
> gsub("\\d{4}", "2014", x)
[1] "Heute leben wir\t im Jahr\t 2014\n"
```

Innerhalb von regulären Ausdrücken haben u.a die folgenden Zeichen eine spezielle Bedeutung:

".", "?", "^", "$", "*", "+", Klammern aller Art