

```
# Eine Einführung in R

# '0 Allgemein'
# '0.1 R starten, R beenden'
# Wenn man R startet, beginnt man einen Session. Man beendet einen Session mit
q()

# (Hier die Frage 'Save workspace image' mit 'cancel' beantworten).

# 0.2 'Verzeichnis erzeugen für das Import/Export Daten'
# Wir werden im Laufe des Semesters öfters Daten in R importieren
# und von R exportieren müssen. Erzeugen Sie dafür ein Verzeichnis
# auf Ihrer Festplatte und geben Sie den Pfad von diesem Verzeichnis in " " ein in R
.

# z.B.
pfad = "/Users/jmh/d/R"

# Sichern Sie, dass dieses Objekt mit dem erwünschten Pfad existiert:

ls()
# [1] "pfad"

pfad
# [1] "/Users/jmh/d/R"

# Wir werden öfters sogenannte 'Data-Frames' aus einer Webseite
# (Tabellen als Textdateien) laden. Hier wird
# der Pfad dieser Webseite festgelegt:
pfadu = "http://www.phonetik.uni-muenchen.de/~jmh/Lehre/Rdf"

# Jetzt R beenden:
q()

# und antworten Sie nur dieses eine Mal
# y(es) auf die Frage 'Save workspace image?'.
# Dies soll das einzige Mal sein, in dem diese Frage mit 'yes' beantwortet
# wird: sonst in Zukunft immer n(o).

# R wieder starten, und verifizieren, dass "pfad" und "pfadu" existieren:

ls()
# "pfad" "pfadu"

# '0.3 Objekte in R erzeugen und listen'

# Ein neues Objekt wird mit '=' (assign) oder '<-' erzeugt
# Es gibt numerische Objekte
x = 3
y <- 4
ls()
```

```
# Und auch Schriftzeichen-Objekte in ""
z = "etwas"

# Wenn Sie den Inhalt des Objektes sehen wollen, einfach
# den Objekt-Namen eingeben
x
# 3

# Neue Objekte erzeugen mit dem selben Inhalt
x = y = z = 4

# Objekte werden in R gleich überschrieben ohne Warnmeldung
y = 4
y
# 4
y ="Phonetik"
# y
# "Phonetik"

# '0.4 Objekte in R permanent sichern'
# Am besten mit 'save()', wie folgt. Hier ist "etwas" irgendein beliebiger Name
save(list = ls(), file = file.path(pfad, "etwas"))

# Dann 'q()' und (immer) mit 'n' beantworten - siehe 0.2

# '0.5 Objekte wieder einlesen mit attach()'

# R starten und dann:
attach(file.path(pfad, "etwas"))

# Hier ist "etwas" der Name in '0.4' oben. Dann sind die Objekte wieder da
y
# "Phonetik"
x
# 4

# Diese Objekte werden in R in einem library zusammengepackt; alle
# im Session vorhandenen Libraries sind durch 'search()' vorhanden:
search()
# [1] ".GlobalEnv"          "file:/Users/jmh/d/R/etwas"
# [3] "tools:RGUI"          "package:stats"          # usw.
# Hier sehen wir, dass unser library an zweiter Stelle ist.
# Die darin gespeicherten Objekte sind also auf diese Weise sichtbar:

ls(pos=2)
# [1] "pfad" "x"    "y"    "z"

# Jeder Library kann mit 'detach()' wieder entfernt werden.
detach(2)

search()
# [1] ".GlobalEnv"          "tools:RGUI"            "package:stats"
```

```

# und die Objekte sind daher nicht mehr vorhanden:
x
# Error: object 'x' not found

# '0.6 Hilfeseiten'
# Für eine Einführung in R: 'An Introduction to R' in
help.start()

# Wenn Sie mehr über eine Funktion wissen wollen:
help(pnorm)
# oder
?pnorm

example(density)

apropos("spline")

help.search("norm")

# Überblick der vorhandenen R-Funktionen:
# http://cran.r-project.org/doc/contrib/Short-refcard.pdf

# '0.7 Data-Frames einlesen'
# Ein Data-Frame ist eine Matrix mit verschiedenen Reihen und
# Spalten, die wir für die statistische Analyse sehr oft
# verwenden werden. Die Data-Frames können so eingelesen
# werden (angenommen '0.2' durchgeführt wurde):

read.table(file.path(pfadu, "ai.txt"))

# das gleiche und als Objekt sichern:
ai.df = read.table(file.path(pfadu, "ai.txt"))
ai.df
#      F1      Kiefer      Lippe
# 1    773 -25.47651 -24.59616
# 2    287 -27.02695 -26.44491
# 3   1006 -27.24509 -27.59161
# 4    814 -26.05803 -27.17365
# usw.
vlax.df = read.table(file.path(pfadu, "vlax.txt"))
# Data-Frame ablegen
write.table(vlax.df, file.path(pfad, "v.txt"))
# besser: damit keine "" erscheinen
write.table(vlax.df, file.path(pfad, "v.txt"), quote=F)

# '0.8 Data-Frames selber aufbauen und erzeugen'
# 10 Etikettierungen, 5 /i/, 5 /o/
v.l = c("i", "i", "i", "i", "i", "o", "o", "o", "o", "o")
# oder
v.l = c(rep("i", 5), rep("o", 5))
f1 = c(240, 220, 250, 210, 280, 520, 510, 605, 670, 613)
vok.df = data.frame(f1, factor(v.l))

```

```
# mit anderen Namen
vok.df = data.frame(F1 = f1, V = factor(v.l))
class(vok.df)
summary(vok.df)

# Data-Frame ablegen und wieder einlesen
write.table(vok.df, file.path(pfad, "vok.txt"), quote=F)
read.table(file.path(pfad, "vok.txt"))
summary(vok)
class(vok.df$V)
```