

```

# Wenn nicht bereits gemacht...
library(emu)
library(lattice)
# Um Trackdateien aus einer Webseite einzulesen
library(RCurl)
source(file.path(pfadu, "readtrackfromurl.R"))
# Für Ellipsen
library(latticeExtra)
source(file.path(pfadu, "ellipsefun.R"))
# Für 3-point central differencing
source(file.path(pfadu, "cendiff.R"))

# 1. Die Diskrete-Cosinus-Transformation: Ein Überblick
#####
mal.s = emu.query("malerin", "*", "Word!=x")
# oder
mal.s = read.emusegs(file.path(paste(pfadu, "malerin", sep="/"), "mal.s.txt"))

mal.f0 = emu.track(mal.s, "F0")
# oder
mal.f0 = readtrackfromurl("mal.f0.txt",file.path(paste(pfadu, "malerin", sep="/")) ,
pfad)

# 2. Ist die Krümmung größer in gespanntem /tut/ vs. ungespanntem /tʊt/?
#####
# Wenn nicht bereits gemacht:
# Segment-list
v = read.emusegs(file.path(paste(pfadu, "dfg", sep="/"), "v.txt"))
# Label-vector: consonantal context
c.l = scan(file.path(paste(pfadu, "dfg", sep="/"), "c.l.txt"), what="")
# Label-vector: ±tense vowel
t.l = scan(file.path(paste(pfadu, "dfg", sep="/"), "t.l.txt"), what="")
# Formant-trackdata for the segment list v
# i.e. derived from emu.track(v, "fm")
v.fm = readtrackfromurl("v.fm.txt",file.path(paste(pfadu, "dfg", sep="/")) , pfad)
# get the vowel label
v.l = label(v)
# get the speaker labels from the utterance names
s.l = substring(utt(v), 1, 2)
# get the rate label (a=slow, b = fast from the utterance name)
r.l = substring(utt(v), 3, 3)

# 3. Inwiefern unterscheiden sich /ai, au, oY/ in F1- und/oder F2-Transitionen?
#####

# 4. Inwiefern ist die F2-Transition für den /ɛ, ɘ/ Unterschied im Polnischen
relevant?
#####
#####

```

```
#####  
#####  
##### Segmentliste der polnischen Frikative (unkorrigiert)  
# si = /ɕ/, sz = /ʑ/  
# Entweder:  
pfric.s = emu.query("pol13", "*", "[#phon = s | si | sz & Start ( word,phon ) = 1 ->  
phon = e | o | a ]")  
# Oder:  
pfric.s = read.emusegs(file.path(paste(pfadu, "pol", sep="/"), "pfric.s.txt"))  
  
##### Segmentliste der danach kommenden Vokale  
# Entweder:  
pvow.s = emu.query("pol13", "*", "[phon = s | si | sz & Start ( word,phon ) = 1 -> #  
phon = e | o | a ]")  
# Oder:  
pvow.s = read.emusegs(file.path(paste(pfadu, "pol", sep="/"), "pvow.s.txt"))  
  
##### Label-Vektor der Akzent-Etikettierungen  
#  
# Entweder  
pacc.l = emu.requery(pfric.s, "phon", "acc", j=T)  
# Oder  
pacc.l = scan(file.path(paste(pfadu, "pol", sep="/"), "pacc.l.txt"), what="")  
  
##### Trackdatei von F2  
# Entweder  
pvow.fm = emu.track(pvow.s, "fm")[,1:2]  
# Oder  
pvow.fm = readtrackfromurl("pvow.fm.txt",file.path(paste(pfadu, "pol", sep="/")) ,  
pfad)  
  
#####  
#####  
#####  
#####  
  
##### Segmentliste der polnischen Frikative (Formanten teilweise  
korrigiert)  
# si = /ɕ/, sz = /ʑ/  
# Entweder:  
pfric.s = emu.query("pol13b", "*", "[#phon = s | si | sz & Start ( word,phon ) = 1 ->  
phon = e | o | a ]")  
# Oder:  
pfric.s = read.emusegs(file.path(paste(pfadu, "polb", sep="/"), "pfric.s.txt"))  
  
##### Segmentliste der danach kommenden Vokale  
# Entweder:  
pvow.s = emu.query("pol13b", "*", "[phon = s | si | sz & Start ( word,phon ) = 1 -> #  
phon = e | o | a ]")  
# Oder:  
pvow.s = read.emusegs(file.path(paste(pfadu, "polb", sep="/"), "pvow.s.txt"))  
  
##### Label-Vektor der Akzent-Etikettierungen
```

```
#
# Entweder
pacc.l = emu.requery(pfri.s, "phon", "acc", j=T)
# Oder
pacc.l = scan(file.path(paste(pfad, "polb", sep="/"), "pacc.l.txt"), what="")

##### Trackdatei von F2
# Entweder
pvow.fm = emu.track(pvow.s, "fm")[,1:2]
# Oder
pvow.fm = readtrackfromurl("pvow.fm.txt",file.path(paste(pfad, "polb", sep="/")) ,
pfad)

#####
#####
#####
#####
```