

```

library(emu)
library(lattice)
# to install some trackdata objects
library(RCurl)
source(file.path(pfadu, "readtrackfromurl.R"))
# for ellipses
library(latticeExtra)
# three-point central differencing
source(file.path(pfadu, "cendiff.R"))

##### Daten aus Woche 1
##### Either:
son.lab = emu.requery(k.s, "Segment", "Segment", seq=2, j=T)
word.lab = emu.requery(k.s, "Segment", "Word", j=T)
tip.tt = emu.track(tip.s, "tt_posz")
body.tb = emu.track(body.s, "tb_posz")

##### Or:
son.lab = scan(file.path(paste(pfadu, "physch5", sep="/"), "son.lab.txt"), what="")
word.lab = scan(file.path(paste(pfadu, "physch5", sep="/"), "word.lab.txt"), what="")
tip.tt = readtrackfromurl("tip.tt.txt", file.path(paste(pfadu, "physch5", sep="/")),
pfad)
body.tb = readtrackfromurl("body.tb.txt", file.path(paste(pfadu, "physch5", sep="/")),
, pfad)

##### Part 1, trackapply():
# If you need to apply a function to a trackdata object
# and the function is not listed in
# Arith, Compare, Math, Math2 Funktionen of help(Ops)

# writing your function

##### Part 2, scatter plots and
ellipses
dim(vowlax.fdat)

# See below for how to modify axis labels

# Ellipse superimposition
# the default (without level = .95) is for 1.5 standard-deviation ellipses
# they enclose 68% of the data points. Why?
# pchisq(1.5^2, 2)
# the above encloses 95% of data points
# remove center.pch=NULL to plot the ellipse-centroid

efun = function(x, y, ...) {
  panel.xyplot(x, y, ...)
  panel.ellipse(x, y, level = .95, center.pch=NULL, ...)
}

##### Part 3, Euclidean distances

```

```

# 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

# get the speaker labels from the utterance names

# get the rate label (a=slow, b = fast from the utterance name)

# plot the spaces for overlaid vowels, separately for ± tense,
# data at the vowel temporal midpoint

# calculate the distance to the centre of the vowel space
# separately for tense and lax vowels

##### Modifying axis labels in xyplot

axfun =
function(side, ...) {

if(side == "bottom")
{
w = seq(-2500, -1000, by = 500)
wlab = as.character(-w)
panel.axis(side=side, at = w, labels = wlab, outside=T, rot=0)
}

if(panel.number()==1)
{
if(side == "left")
{
w = seq(-1000, -200, by = 200)
wlab = as.character(-w)
panel.axis(side=side, at = w, labels = wlab, outside=T, rot=0)
}
}
# else
# axis.default(side = side, ...)
}

xyplot(-m[,1] ~ -m[,2] | vowelax.spkr, groups=vowelax.l, auto.key=T, panel=efun, axis=
axfun)

```

