

```
library(lme4)
library(lattice)
source(file.path(pfadu, "sigmoid.txt"))

ovokal = read.table(file.path(pfadu, "ovokal.txt"))
tab = with(ovokal, table(Jahr, Vokal))
prop = prop.table(tab, 1)
barchart(prop, auto.key=T, horizontal=F)
o = glm(Vokal ~ Jahr, family=binomial, data = ovokal)
k = coef(o)[1]
m = coef(o)[2]
sigmoid(prop, k, m, xlab = "Jahr", ylab = "Proportion", ylim=c(0,1))

labr = read.table(file.path(pfadu, "labr.txt"))
tab.lab = with(labr, table(Stimulus, Urteil))
prop.lab = prop.table(tab.lab, 1)
barchart(prop.lab, auto.key=T, horizontal=F)

# Sigmoid anpassen
# Modell
o.lab = lmer(Urteil ~ Stimulus + (1+Stimulus|Vpn), family=binomial, data = labr)
# Bevölkerung k und m
k.lab = fixef(o.lab)[1]
m.lab = fixef(o.lab)[2]
sigmoid(prop.lab, k.lab, m.lab, xlab = "F2 (Hz)", ylab="Proportion /Y/")

# Umkipppunkt der Bevölkerung überlagern
u.lab = -k.lab/m.lab
abline(v=u.lab, lty=2)
abline(h=.5, lty=2)

# k und m Werte pro Vpn sind hier:
coef.lab = coef(o.lab)[[1]]
# die Umkipppunkte pro Vpn
um.lab = -coef.lab[,1]/coef.lab[,2]
boxplot(um.lab)

alvr = read.table(file.path(pfadu, "alvr.txt"))
# Proportionen berechnen
tab.alv =
prop.alv =

# Sigmoid anpassen
o.alv =
# m und k der Bevölkerung
k.alv =
m.alv =
sigmoid(prop.alv, k.alv, m.alv, xlab = "F2 (Hz)", ylab="Proportion /Y/")

# Umkipppunkt der Bevölkerung überlagern
u.alv =

# k und m Werte pro Vpn
```

```
coef.alv =  
# die Umkipppunkte pro Vpn  
um.alv  
boxplot(um.alv)
```