

# Lineare Regression mit Logratio in R

Benjamin Schlegel

30. Mai 2016

Bei [zusammengesetzten Daten](#) (compositional data) sind die Werte einer Variable abhängig. Um dieses Problem zu lösen, kann eine lineare Regression mit logratio berechnet werden.

Mit folgender Funktion kann das lineare Modell geschätzt werden. Die Funktion rechnet die logratio Skala automatisch in die normale Skala zurück und gibt die korrekten p-Werte wieder.

```
lm.logratio = function(model, data){
  model = lm(model,data = data)
  beta = exp(coefficients(model))
  vcov = beta^2*vcov(model)
  sd = t = p = c()
  result = matrix(ncol=4,nrow=length(beta))
  for(i in 1:length(vcov[,1])){
    sd[i] = sqrt(vcov[i,i])
    t[i] = beta[i] / sd[i]
    p[i] = 2 * pt(t[i], length(data[,1])-1, lower=FALSE)
    result[i,1] = beta[i]
    result[i,2] = sd[i]
    result[i,3] = t[i]
    result[i,4] = p[i]
  }
  colnames(result) = c("Estimate","Std. Error","t value","Pr(>|t|)")
  rownames(result) = names(beta)
  return(result)
}
```

Ein Beispiel soll illustrieren, wie die Funktion verwendet werden kann. Dafür müssen zuerst die Ständeratswahlen 2015 Daten von [Data](#) heruntergeladen werden und anschliessend in R eingelesen werden.

```
data_sr = read.csv("sr_wahlen15.csv")
```

Anschliessend kann mit der Funktion ein Modell geschätzt werden. Wir wollen testen, ob Bisherige eher gewählt werden als Neue. Als Kontrollvariablen verwenden wir die Kontextvariablen Halbkanton und ob es einen neuen Sitz zu holen gibt. Die abhängige Variable ist bereits auf der logratio Skala.

```
lm.logratio(stimmanteilLogRatio ~ bisher + halbkanton + freier_sitz, data = data_sr)
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.4413509	0.05912404	7.464830	6.023097e-12
bisher	4.9064048	1.22138090	4.017096	9.236441e-05
halbkanton	7.0252652	2.51847966	2.789487	5.955247e-03

freier\_sitz 0.8618307 0.17966181 4.796961 3.811271e-06

Im Output ist ersichtlich, dass alle Variablen signifikant sind (p