

Einführung in die Datenanalyse mit \mathcal{R}

Weiterführende Hinweise

Dag Tanneberg*

17. Oktober 2018

1 Auswertung multiplikativer Interaktionsterme

Die letzte Sitzung der Veranstaltung riss die Schätzung und Auswertung generalisierter linearer Modelle (GLM) an. Die Illustration griff auf einen Datensatz zurück, der Informationen über alle namentlich erwähnten Charaktere der Fantasyreihe “A Song of Ice and Fire” bereitstellte. Die Fragestellung lautete, mit welcher Wahrscheinlichkeit der Hauptcharakter Jon Snow sterben würde. Zu ihrer Beantwortung griff die Veranstaltung auf Angaben über die Zugehörigkeit zu einem Adelshaus, das Alter¹, das Geschlecht sowie den Adelsstatus zurück. Das vorgeschlagene logistische Regressionsmodell beinhaltete außerdem eine Interaktion von Geschlecht und Adelsstatus sowie ein kubisches Polynom des Alters. Offen blieb zum Schluss die Auswertung multiplikativer Interaktionsterme. Dieser Abschnitt demonstriert eine Strategie zur graphischen Auswertung entsprechender Terme.

Zunächst sei an das Regressionsmodell erinnert, mit dessen Hilfe

```
fit <- glm( # generate model
  died ~ 0 + allegiances +
    gender*nobility +
    age_in_chapters + I(age_in_chapters^2) + I(age_in_chapters^3),
  family = binomial(link = "logit"),
  data = asoiaf
)
```

Multiplikative Interaktionsterme können in verschiedenen Formen ausgewertet werden. In den Sozialwissenschaften erfreuen sich sogenannte Marginal Effect Plots besonderer Beliebtheit und sie treten in verschiedenen Varianten auf.

*dag.tanneberg@uni-potsdam.de

¹ In Ermangelung besserer Informationen misst die entsprechende Variable nicht das natürliche Alter, sondern zählt die Anzahl der seit Einführung des Character verstrichenen Kapitel. Mit etwas gutem Willen handelt es sich um eine Messung der Prozesszeit.

1. Darstellung der partiellen Ableitung

2. Darstellung der

```
pred_data <- expand.grid(  
  allegiances = c("Baratheon", "Lannister", "Stark", "Targaryen"),  
  nobility = 0:1, gender = 0:1,  
  age_in_chapters = median(asoiaf[, "age_in_chapters"], na.rm = TRUE),  
  stringsAsFactors = FALSE  
)
```

Der resultierende Datensatz enthält eine Zeile für jede der vier möglichen Kombinationen von **gender** und **nobility**. Die werte der Variablen **allegiances** und **age_in_chapters** bleiben über alle Reihen hinweg konstant.

##	allegiances	nobility	gender	age_in_chapters
## 1	Baratheon	0	0	126
## 2	Lannister	0	0	126
## 3	Stark	0	0	126
## 4	Targaryen	0	0	126
## 5	Baratheon	1	0	126
## 6	Lannister	1	0	126
## 7	Stark	1	0	126
## 8	Targaryen	1	0	126
## 9	Baratheon	0	1	126
## 10	Lannister	0	1	126
## 11	Stark	0	1	126
## 12	Targaryen	0	1	126
## 13	Baratheon	1	1	126
## 14	Lannister	1	1	126
## 15	Stark	1	1	126
## 16	Targaryen	1	1	126

Mit Hilfe der Funktion **predict()**

```
pred_data[, "fitted"] <- predict(fit, newdata = pred_data, type = "response")  
pred_data
```

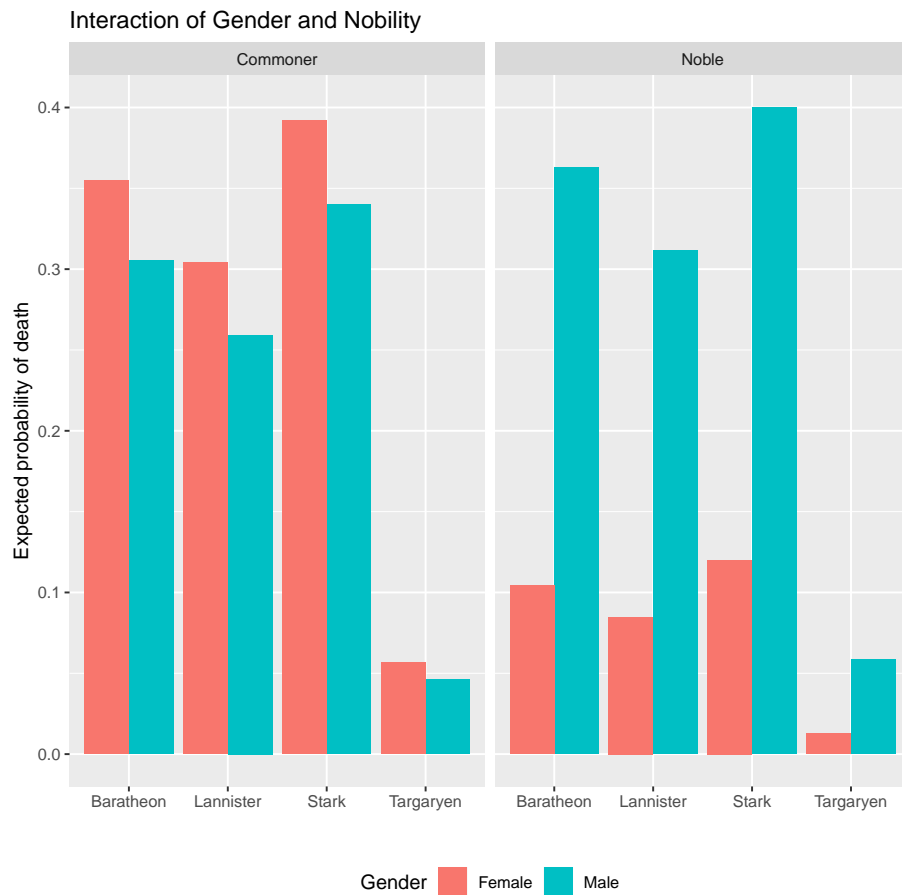
##	allegiances	nobility	gender	age_in_chapters	fitted
## 1	Baratheon	0	0	126	0.35480247
## 2	Lannister	0	0	126	0.30433574
## 3	Stark	0	0	126	0.39193082
## 4	Targaryen	0	0	126	0.05682782
## 5	Baratheon	1	0	126	0.10427826
## 6	Lannister	1	0	126	0.08476429

## 7	Stark	1	0	126	0.12006921
## 8	Targaryen	1	0	126	0.01259482
## 9	Baratheon	0	1	126	0.30551914
## 10	Lannister	0	1	126	0.25924574
## 11	Stark	0	1	126	0.34020963
## 12	Targaryen	0	1	126	0.04598431
## 13	Baratheon	1	1	126	0.36266208
## 14	Lannister	1	1	126	0.31161734
## 15	Stark	1	1	126	0.40010286
## 16	Targaryen	1	1	126	0.05868707

```

pred_data <- within(pred_data, {
  # Note: within() requires curly brackets whenever you want to
  # evaluate multiple statements.
  gender_label <- factor(gender, 0:1, c("Female", "Male"))
  nobility_label <- factor(nobility, 0:1, c("Commoner", "Noble"))
})
ggplot(
  data = pred_data,
  mapping = aes(x = allegiances, y = fitted, fill = gender_label)
) +
  geom_bar(stat = 'identity', position = "dodge") +
  facet_grid(~ nobility_label) +
  labs(
    title = "Interaction of Gender and Nobility",
    x = "",
    y = "Expected probability of death",
    fill = "Gender"
  ) +
  theme(legend.position = "bottom")

```



Häufig verwenden die Sozialwissenschaften Marginal Effect Plots, d. h. grafische Darstellungen der slope.

```
fit2 <- glm( # generate model
  died ~ 0 + allegiances +
    gender + nobility * age_in_chapters,
  family = binomial(link = "logit"),
  data = asoiaf
)
summary(fit2)

##
## Call:
## glm(formula = died ~ 0 + allegiances + gender + nobility * age_in_chapters,
##      family = binomial(link = "logit"), data = asoiaf)
##
## Deviance Residuals:
```

```

##      Min      1Q   Median      3Q      Max
## -2.7193 -0.5640 -0.2196   0.5709   3.0360
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## allegiancesArryn      0.681880   0.732736   0.931  0.35206
## allegiancesBaratheon  2.421455   0.531316   4.557 5.18e-06 ***
## allegiancesGreyjoy    0.677669   0.452667   1.497  0.13438
## allegiancesLannister  2.060059   0.466747   4.414 1.02e-05 ***
## allegiancesMartell   -0.121929   0.713560  -0.171  0.86432
## allegiancesNight's Watch 3.067868   0.506599   6.056 1.40e-09 ***
## allegiancesNone       1.127318   0.350692   3.215  0.00131 **
## allegiancesStark      2.378484   0.430761   5.522 3.36e-08 ***
## allegiancesTargaryen  0.151837   0.575078   0.264  0.79176
## allegiancesTully      1.914210   0.660467   2.898  0.00375 **
## allegiancesTyrell     0.713196   0.878618   0.812  0.41695
## allegiancesWildling   2.155148   0.487756   4.418 9.94e-06 ***
## gender                0.604277   0.293921   2.056  0.03979 *
## nobility              -1.395722   0.331915  -4.205 2.61e-05 ***
## age_in_chapters       -0.026748   0.002422 -11.046 < 2e-16 ***
## nobility:age_in_chapters 0.012813   0.002899   4.421 9.84e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1243.51  on 897  degrees of freedom
## Residual deviance:  666.25  on 881  degrees of freedom
##      (20 observations deleted due to missingness)
## AIC: 698.25
##
## Number of Fisher Scoring iterations: 6

```