

Introduction to \mathcal{R}

Session 6. (G)LMs

Dag Tanneberg¹

Potsdam Center for Quantitative Research
University of Potsdam, Germany
November 8/9, 2018

¹Quantitative Social Scientist, www.dtanneberg.de

Introduction

Before we start...

- Quit & reopen RStudio.
- Open the notebook “notebook_06.Rmd” in RStudio
- Load “./06/dta/asoiaf.csv” from the course material.
- Execute the following code below. What does it do?

```
asoiaf[, "is_dead"] <- !is.na(asoiaf[, "book_of_death"])
```

What do we intent to do?

- **Question:** What's the chance that Jon Snow is going to die?
- **Means:** Regression on a linear combination of predictors

$$p(Death = 1|x, \beta) = \sum_K \beta_k x_k$$

- **Caveat:** Chance of death is not a well-behaved response.

Outline

- 1 Introduction
- 2 Linear Models in \mathcal{R}
- 3 Working With Results
- 4 Generalized Linear Models
- 5 Checking Assumptions
- 6 Summary

Linear Models in \mathcal{R}

Generic Format of Fitting GLMs

```
fit <- lm(  
  formula = <formula>,  
  family = <family>(link = "<link>"),  
  data = <data>,  
  subset = <subset>,  
  weights = <weights>, # Be careful! Meaning changes  
                        # depending on <family>.  
  na.action = na.omit, # Retains only complete cases.  
  <...> # Further options.  
)
```

\mathcal{R} 's Formula Interface²

Generic Example

$$y \sim x_1 + x_2 + \cdots + x_k$$

Formula Creation

Symbol	Meaning	Example
:	Specify an interaction	$y \sim x : z \Rightarrow y = xz$
*	Specify all possible interactions	$y \sim x * z \Rightarrow y = x + z + xz$
^	Specify interactions up to some degree	$y \sim (x + z)^2 \Rightarrow y = x + z + xz$
.	Wildcard for all other variables	$y \sim . \Rightarrow y = x + z + w + \dots$
-	Remove variable(s)	$y \sim (x + z)^2 \setminus x : z \Rightarrow y = x + z$
-1 OR 0+	Remove the intercept	$y \sim x - 1$ OR $y \sim 0 + x$
$I()$	Arithmetical transformation	$y \sim I(x^2) \Rightarrow y = x^2$
<i>function</i>	Other mathematical transformations	$\log_{10}(y) \sim x \Rightarrow \log_{10}(y) = x$

²Adapted from Kabacoff, R. 2011. *R in Action*. Shelter Island: Manning Publications, p. 178.

\mathcal{R} 's Formula Interface, contd.

Exercise How would you write the following formulas?³

1 $y = a + x + z + xz$

2 $y = a + x + x^2 + x^3$

3 $\log_{10}(y) = x + z + w + xz + xw + wz$

4 y as a function of variables in the data but k

³Assume a is the constant.

Get Your Hands Dirty

Now it's your turn. Use the **asoiaf** data to

- regress **is_dead** on
- **allegiances**,
- the full interaction of **gender** and **nobility**, and
- a cubic polynomial on **age_in_chapters**.
- Save the results to an object called **myfit**.

One Solution to the Exercise

```
myfit <- glm(  
  formula = is_dead ~ 0 + allegiances +  
    gender * nobility +  
    age_in_chapters + I(age_in_chapters^2) +  
    I(age_in_chapters^3),  
  data = asoiaf  
)
```

Working With Results

A Menu of Typical Options⁴

Function	Output
summary()	Display detailed model results
coef()	Display fitted model parameters
confint()	Provide confidence intervals
fitted()	Return fitted values
residuals()	Return residual values
anova()	Return an ANOVA table for a fitted model or compare fitted models
vcov()	Return the variance-covariance matrix
AIC()	Return Akaike's Information Criterion
plot()	Display diagnostics plots
predict()	Predict response values for new data

⁴Adapted from Kabacoff, R. 2011. R in Action. Shelter Island: Manning Publications, p. 179.

How to Predict New Data

Generic Sequence

- 1 Define scenarios to predict
- 2 Create a date frame which contains those scenarios
- 3 Use `predict()` to return quantities of interest
- 4 Summarize the results

Example: How does the chance to die change over time?

```
# Steps 1 & 2 Prediction scenarios
pred_dta <- data.frame(
  allegiances = "Baratheon",
  gender = mean(asoiaf$gender),
  nobility = mean(asoiaf$nobility),
  age_in_chapters = 1:343, stringsAsFactors = FALSE
)
# Step 3 Predict
pred_dta[, "fitted"] <- predict(myfit,
  newdata = pred_dta)
# Step 4 Summarize
ggplot(data = pred_dta,
  aes(x = age_in_chapters, y = fitted)) + geom_line()
```

Generalized Linear Models

Intuition

■ Problem

- a. We don't observe probabilities but discrete events.
- b. Probabilities are restricted to $[0, 1]$, but $\mathbf{X}\beta$ can take any value.

■ Challenge: Map $\mathbf{X}\beta$ to the domain of our response

■ Applies to many quantities of interest, e.g.,

- Household income
- Satisfaction with democracy
- Number of bills per session of parliament
- ...

■ GLMs: link function $g(\cdot)$ relates response to $\mathbf{X}\beta$

- logit transformation $[\ln(\frac{p}{1-p})]$ for binary DVs
- natural log ($\ln(\mu)$) for count data

Generic Format of Fitting GLMs

```
fit <- glm(  
  formula = <formula>,  
  family = <family>(link = "<link>"),  
  data = <data>,  
  weights = <weights>, # Be careful! Meaning changes  
                    # depending on <family>.  
  subset = <subset>,  
  na.action = na.omit, # Retains only complete cases.  
  <...> # Options to tweak the optimizer.  
)
```

Family Generators and Link Functions in $\text{glm}()$ ⁵

A Practical Example

```
glm(<...>, family = binomial(link = "logit"), <...>)
```

family	link = "<arg>"							
	μ identity	μ^{-1} inverse	$\ln(\mu)$ log	$\ln(\frac{\mu}{1-\mu})$ logit	$\Phi(\mu)$ probit	$\ln[-\ln(1-\mu)]$ cloglog	$\sqrt{\mu}$ sqrt	$\frac{1}{\mu^2}$ 1/mu^2
gaussian()	●	○	○					
binomial()			○	●	○	○		
poisson()	○		●				○	
Gamma()	○	●	○					
inverse.gaussian()	○	○	○					●
quasi()	●	○	○	○	○	○	○	○
quasibinomial()				●	○	○		
quasi()	○		●				○	

Legend: ● default, ○ possible

⁵Adapted from Fox, J. and S. Weisberg. 2011. An R Companion to Applied Regression. 2nd ed. London: SAGE, pp. 231, 233.

Get Your Hands Dirty

Now it's your turn. Is John Snow going to die? Setup possible scenarios and evaluate the results.

One Possible Solution

```
jon_snow <- which(asoiaf$name == "Jon Snow")
pred_dta <- asoiaf[rep(jon_snow, 3), ]; rm(jon_snow)
pred_dta[2, "allegiances"] <- "Stark"
pred_dta[3, "allegiances"] <- "Targaryen"
pred_dta[, "fitted"] <- predict(
  myfit, newdata = pred_dta, type = "response"
)
pred_dta[, "fitted"]
```

Checking Assumptions

Checking Assumptions

- Always check your diagnostic plots

```
plot(myfit)
```

- For detailed instructions see: Fox, J. and S. Weisberg. 2011. An R Companion to Applied Regression. 2nd ed. London: SAGE.

Summary

Summary

- base \mathcal{R} offers many probability models
- Numerous extensions are available (see the **CRAN Taskviews**)
- Discussion of marginal effects requires some acrobatics
 - Familiarize yourselves with Fox's and Weisberg's **car** package
 - Check out Daniel Lüdtke's **ggeffects** package