Statistical Documentation with R Markdown

Katherine Simeon

May 29, 2018

Why is statistical documentation important?

- Reproducibility
- Sanity of future self
- Good communication practice

Challenges when documenting your work

- Missing information (assumes you'll remember something but you actually won't)
- Many moving parts
- Time consuming (to create and maintain)

What makes good documentation?

Varies/depends!

- Who's the intended **audience**? (future you vs. fellow labmate vs. external person)
- What's the intended **use**? (how specific or generalizable should it be?)
- Statistical Software (in this example, I'll be using R)

Regardless, there should still be a generally standardized format:

- Version control / revision history
- Software dependencies and what files are needed (and what format they should be in)

Documentation components

- Data & the process for cleaning it
- Description of IVs & DVs (how they're quantified, levels/conditions for each variable, etc.)
- Syntax for statistical analyses
- Syntax for figures (if applicable)
- Thorough commenting

The lowdown on R Markdown

- Creates reproducible, dynamic documents with R code embedded
- Technically markdown, but specific to **R**

Getting started with R Markdown

```
install.packages('rmarkdown')
```

1. Installing package should automatically install dependencies
2. In drop-down menu, create new .Rmd file
3. Choose output format (this can also be modified later in the header)
4. Create your document (this cheatsheet is a great place to start)
5. "Knit" it into desired output format!

R Markdown Basics

R code is preceded by: `{r, *a bunch of arguments*}`

The important arguments are:

`echo=TRUE or FALSE`
Should the code be *included* in the final document?

`eval=TRUE or FALSE`
Should the code be *evaluated* in the final document?

Setting up your document

Every new file has a `{r setup}`, where you should:

- Set up your working directory
- Load necessary packages

My set up for this presentation looks like this:

```
# Make sure packages are installed
library(languageR)
library(lme4)
library(ggplot2)
library(car)
data("lexdec") # This is the data we are using today!
```

Our data

lexdec - Lexical decision latencies elicited from 21 subjects for 79 English concrete nouns.

- *Native Language* is classified for each subject as **English** or **Other**.
- Each word is sorted into two levels of *semantic class*: **animal** or **plant**.

From languageR package

lexdec

```
# This dataset has a lot of columns
dim(lexdec)
```

```
## [1] 1659   28
```

```
# Only printing the first 6 rows of the first 8 columns
head(lexdec[,1:8])
```

```
##   Subject       RT Trial Sex NativeLanguage Correct PrevType PrevCorrect
## 1      A1 6.340359    23   F        English correct     word     correct
## 2      A1 6.308098    27   F        English correct  nonword     correct
## 3      A1 6.349139    29   F        English correct  nonword     correct
## 4      A1 6.186209    30   F        English correct     word     correct
## 5      A1 6.025866    32   F        English correct  nonword     correct
## 6      A1 6.180017    33   F        English correct     word     correct
```

Example: A mixed effects model

In this model, we want to predict reaction time (RT) from the fixed effects of native language, and semantic class.

We want to account for random effects of subjects and items (words).

```
model <- lmer(RT ~ NativeLanguageN*ClassN # fixed effects on DV
              + (1+ClassN|Subject) # Random effects of subject
              + (1+NativeLanguageN|Word), # Random effects of item
              data=lexdec,REML=F)
```

Thank you to Laurel Brehm for teaching us this 2 years ago!
See her tutorial here

Sorry output does not fit on the slide

(And I was a too lazy to adjust it. . .)

```
summary(model) # Let's see the model
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: RT ~ NativeLanguageN * ClassN + (1 + ClassN | Subject) + (1 +
##     NativeLanguageN | Word)
##    Data: lexdec
##
##      AIC      BIC   logLik deviance df.resid
##   -919.6   -860.1    470.8   -941.6     1648
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
```

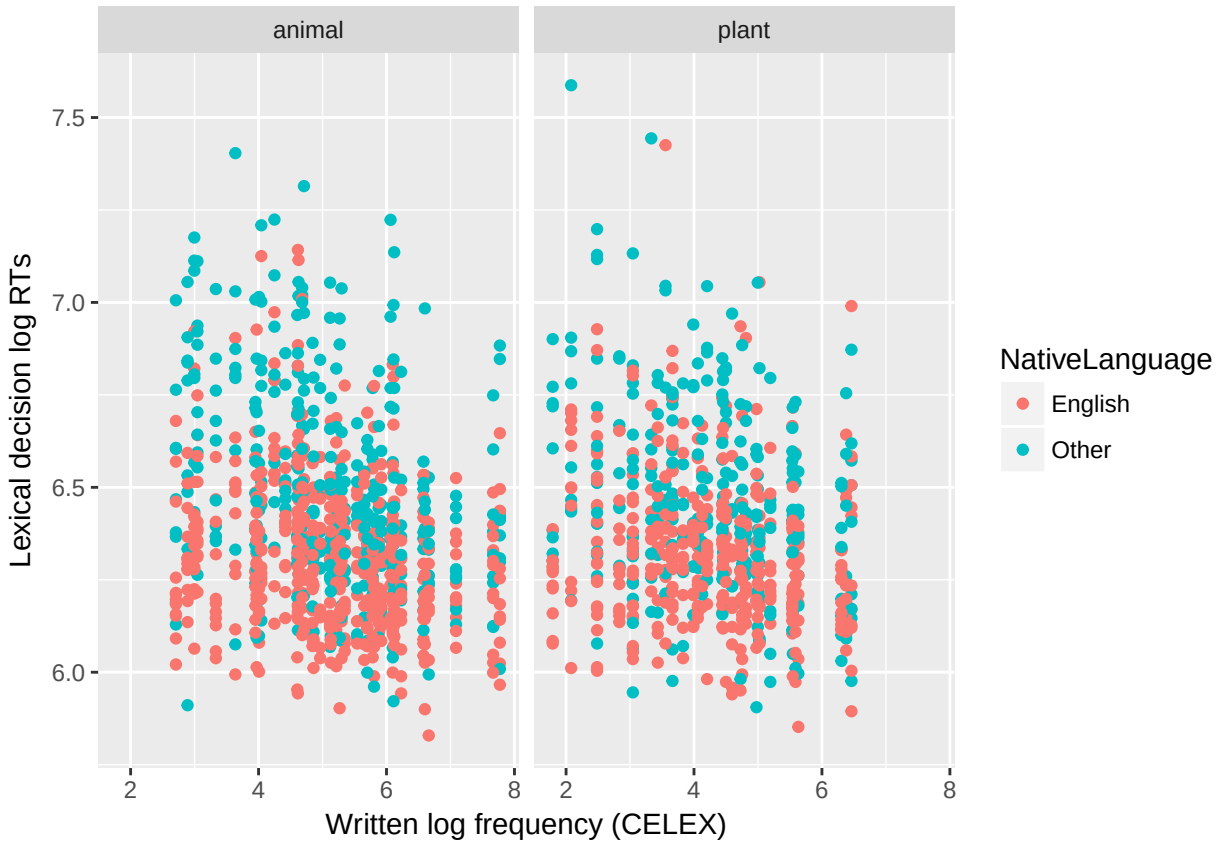
```
## -2.6972 -0.6190 -0.1215  0.4646  6.1393
##
## Random effects:
##  Groups   Name            Variance  Std.Dev. Corr
##  Word     (Intercept)     0.0064465 0.08029
##           NativeLanguageN 0.0021609 0.04649  1.00
##  Subject  (Intercept)     0.0169090 0.13003
##           ClassN          0.0001063 0.01031  1.00
##  Residual                 0.0291430 0.17071
## Number of obs: 1659, groups:  Word, 79; Subject, 21
##
## Fixed effects:
##                          Estimate Std. Error t value
## (Intercept)              6.395711   0.030378  210.54
## NativeLanguageN          0.153476   0.058209    2.64
## ClassN                  -0.008935   0.020212   -0.44
## NativeLanguageN:ClassN -0.041166   0.020548   -2.00
##
## Correlation of Fixed Effects:
##             (Intr) NtvLnN ClassN
## NativeLnggN 0.163
## ClassN      0.144  0.026
## NtvLnggN:CN 0.049  0.237  0.515
```

```
Anova(model) # Let's see p-values
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: RT
##                           Chisq Df Pr(>Chisq)
## NativeLanguageN        10.2591  1    0.00136 **
## ClassN                  0.4715  1    0.49229
## NativeLanguageN:ClassN  4.0136  1    0.04513 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Let's make a figure!

Let's plot RT based on word frequency, sorted by native language and semantic class.



Advantages of documenting in R Markdown

- Accomodates large chunks of text (no rows/paragraphs of #)
- Can organize your code into runnable chunks
- Can be "knit" into multiple formats (pdf, MS Word, HTML)

Other fun R things (that are helpful for documentation)

R Projects

- Each project is in its own directory & workspace
- Easy to share multiple files with collaborators

roxygen skeletons

- Provides the template that is used to document functions
- *Code > Insert Roxygen Skeleton*
- Package: roxygen2