The Grammar of Graphics

Pedro Alcocer

January 10, 2010

ggplot2 is a plotting system for R, based on the grammar of graphics, which tries to take the good parts of the standard graphics utilities and none of the bad parts. It takes care of many of the fiddly details that make plotting a hassle (like drawing legends) as well as providing a powerful model of graphics that makes it easy to produce complex multi-layered graphics.

This module is supposed to be a superficial introduction to ggplot2.

1 Preliminaries

Make certain you have installed the latest version of the packages that this document depends on with:

```
> install.packages("languageR")
> install.packages("ggplot2")
```

We will be using the english dataset from the languageR package. This data set gives mean visual lexical decision latencies and word naming latencies to 2284 monomorphemic English nouns and verbs, averaged for old and young subjects, with various predictor variables.

Learn more about the english dataset with:

```
> ?english
```

Actually inspect the dataset with:

```
> head(english)
     RTlexdec RTnaming Familiarity   Word AgeSubject WordCategory WrittenFrequency
1 6.543754 6.145044        2.37    doe      young            N         3.912023
2 6.397596 6.246882        4.43  whore      young            N         4.521789
3 6.304942 6.143756        5.60 stress      young            N         6.505784
4 6.424221 6.131878        3.87   pork      young            N         5.017280
5 6.450597 6.198479        3.93   plug      young            N         4.890349
6 6.531970 6.167726        3.27   prop      young            N         4.770685

> tail(english)
     RTlexdec RTnaming Familiarity  Word AgeSubject WordCategory WrittenFrequency
4563 6.608770 6.503839        3.70   spy        old            V         5.023881
4564 6.753998 6.446513        2.40   jag        old            V         2.079442
4565 6.711022 6.506979        3.17  hash        old            V         3.663562
4566 6.592332 6.386879        3.87  dash        old            V         5.043425
4567 6.565561 6.519884        4.97 flirt        old            V         3.135494
4568 6.667300 6.496624        3.03  hawk        old            V         4.276666
```

2 Building plots

We are interested in discovering how the lexical decision reaction times (RTlexdec) are distributed. A histogram would be very appropriate here. Think about the plot. On the x-axis should contain the reaction times and y-axis should contain the counts.

We begin a plot by describing what relationships we want to plot. This alone doesn't plot anything because we haven't specified how to plot. The following command describes the relationship we're interested in (i.e., just the behavior of RTlexdec) and stores this description in the variable p. Calling p results in an empty plot window.

```
> p <- ggplot(english, aes(x = RTlexdec))
> p
```

ggplot2 thinks about plots as data relationships and ways to display those relationships. Typically, the data relationships are described with the ggplot() function. This function takes two arguments: (1) the data frame which contains your data, in this case english and (2) a description of the variables which you wish to plot within an aes() function call. In this case, we are only interested in plotting something along the x axis, so the call is aes(x = RTlexdec).

How relationships are actually plotted is handled by the geom and stat family of commands. There are many geoms available to you.[1] For instance, to plot a histogram, we would add the geom geom_histogram() to the p object we created.

```
> p + geom_histogram()
```

To plot a smooth density estimate, we use the geom_density() geom, instead.

```
> p + geom_density()
```

Now we know how to plot histograms and smoothed density estimates. So far, we haven't done anything that the base R graphics system can't do. Let's move beyond base.

[1] For a complete list of geom and stat commands see the online ggplot2 reference manual at http://had.co.nz/ggplot2/

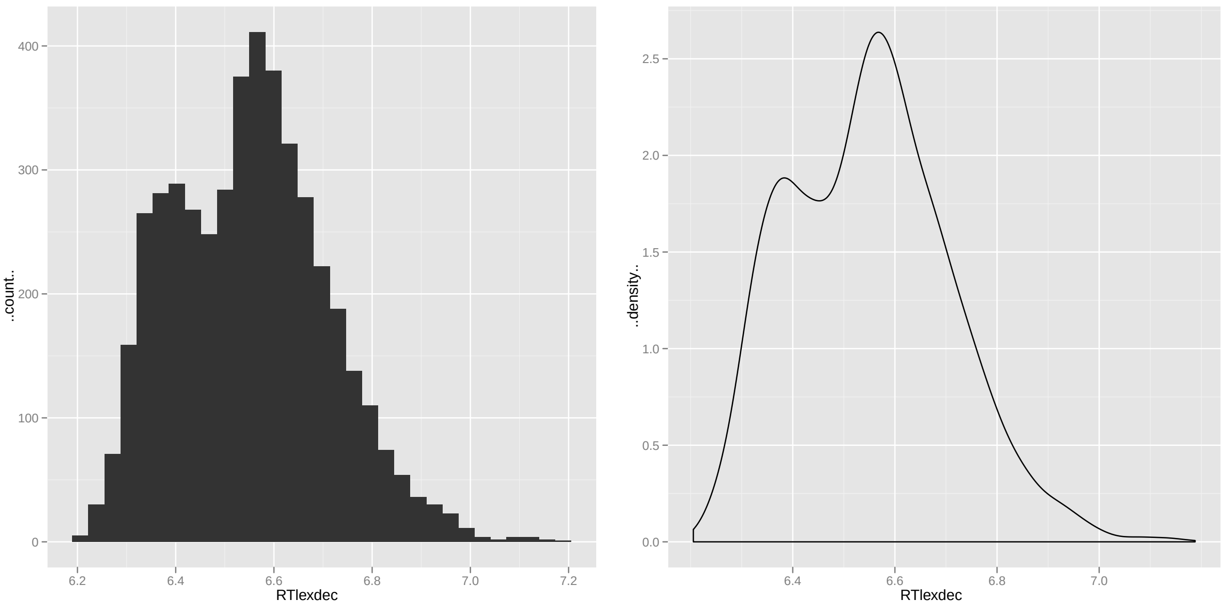


Figure 1: (Left) A histogram of the distribution of the `RTlexdec` response variable made with the `geom_histogram` geom. (Right) A smooth density estimate of the same, made with the `geom_density` geom.

3 Beyond base

Notice that the histogram and the density estimate seem to have two peaks. This may indicate that we are looking at two overlapping distributions. `ggplot2` makes it very easy to divide data by another dimension and display it in several ways.

3.1 Splitting by color

First, let's try to find what predictor might be causing the two distributions. We'll consider three predictors: `WordCategory`, the category of the word that is being presented, N or V; `CV`, whether the word in question begins with a consonant or a vowel; and `AgeSubject`, whether the subject falls into the "young" age group or the "old" age group. We'll plot a smoothed density estimate and separate the two groups based on color.

```
> p + geom_density(aes(color = WordCategory))
> p + geom_density(aes(color = CV))
> p + geom_density(aes(color = AgeSubject))
```

Note that the predictor you are splitting by must be a factor in R. You will get an error if you try to split by a numerical predictor. You can, however, convert integer predictors into factors with `factor()`.

```
> p + geom_density(aes(color = LengthInLetters)) # Doesn't work.
> p + geom_density(aes(color = factor(LengthInLetters))) # Does what you expect.
```

3.2 Faceting

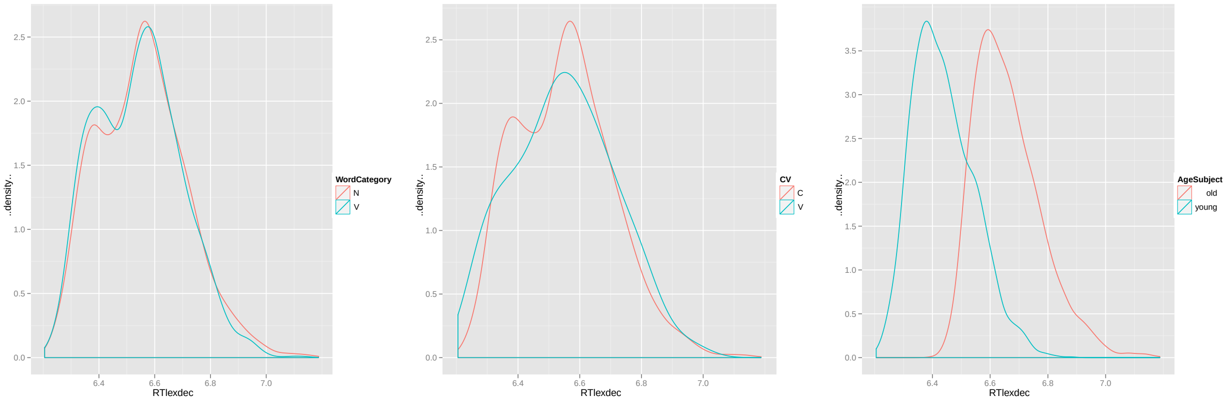
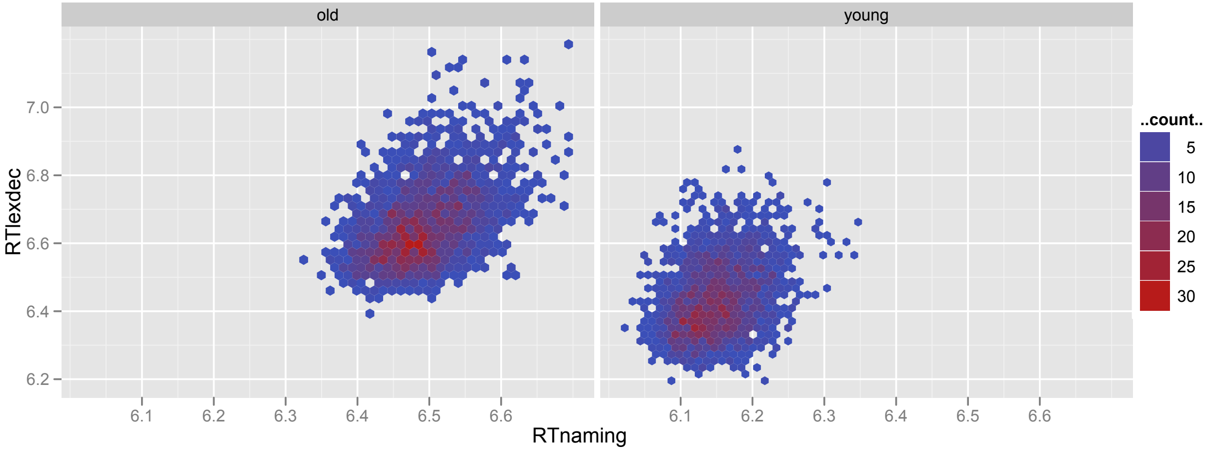


Figure 2: (Left) `WordCategory` (Center) `CV` (Right) `AgeSubject`


old
young
RTlexdec
RTnaming
..count..