

# Simple plots

*Mark Andrews*

*April 5, 2017*

## Introduction

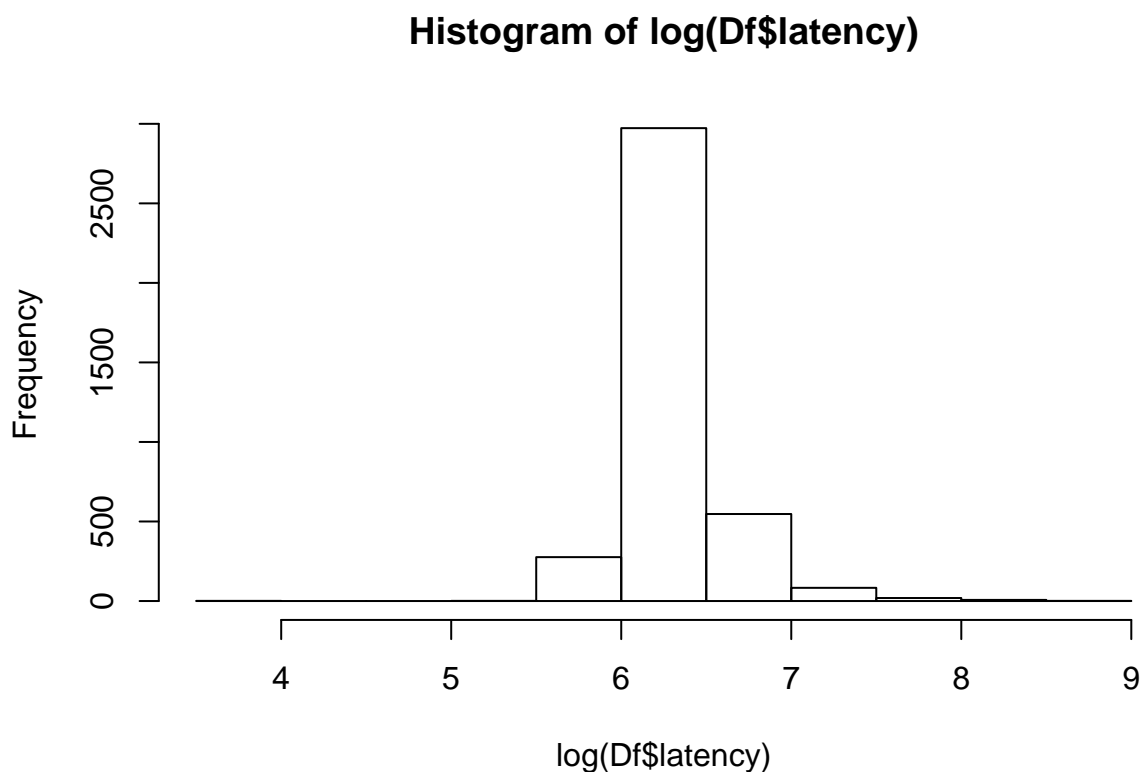
There are two main ways to do plots in R: base R plots, and ggplot plots. There are other approaches too, but these two are the most widely used. Here, we cover base R plots.

Base R plots allows you to do simple plot relatively easily, but begins to be unwieldy when you need to more complex things. On the other hand, there is a lot of fine grained control in base R plots, so with persistence, you can usually get what you need.

## Histograms

For this, we'll get a data frame, and do a histogram of the log of the latency (reaction time).

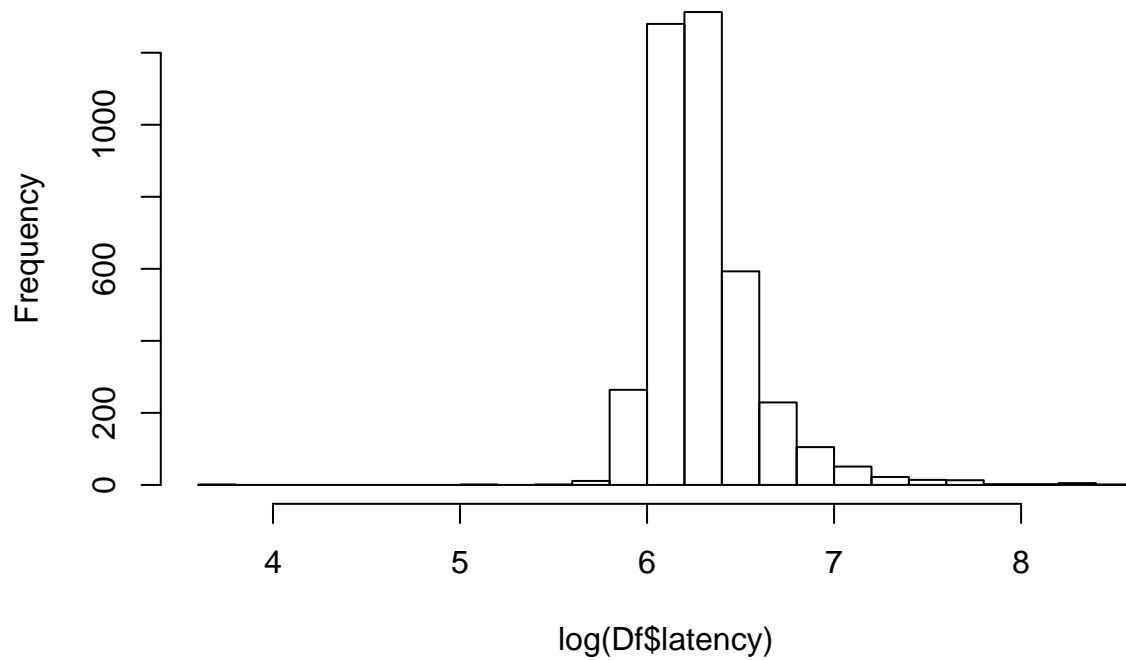
```
Df <- read.csv('../data/LexicalDecision.csv', header=T)
hist(log(Df$latency))
```



We can use more bins than the default of 10.

```
hist(log(Df$latency), 25)
```

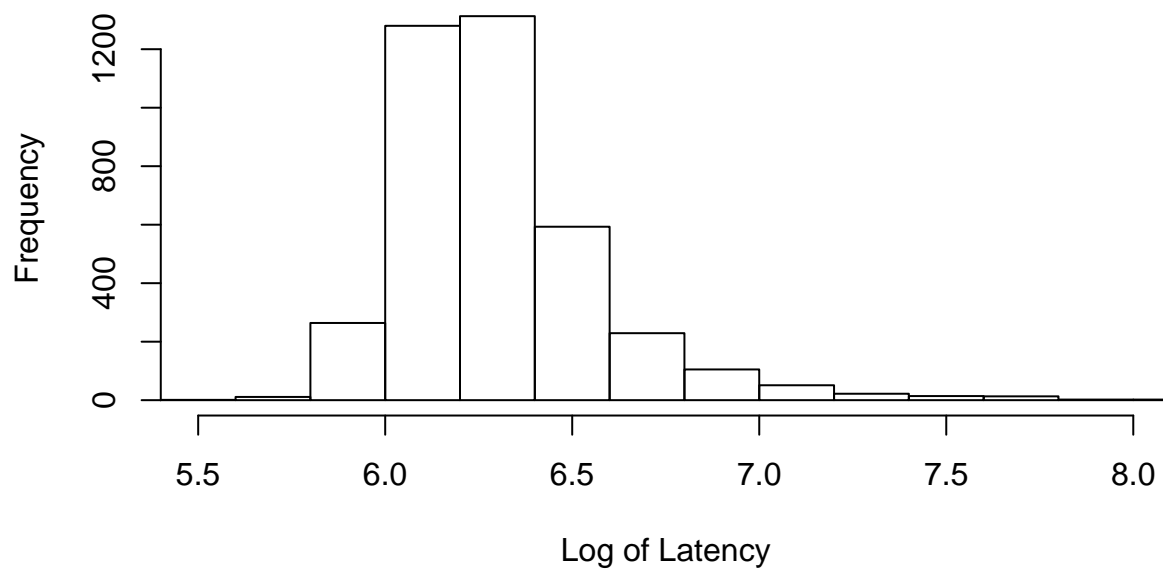
**Histogram of  $\log(\text{Df\$latency})$**



Now, we'll give it a better title, a better set of labels and fix the axes:

```
hist(log(Df$latency),  
     xlab='Log of Latency',  
     ylab='Frequency',  
     xlim=c(5.5, 8),  
     main='My prettier histogram',  
     breaks=25)
```

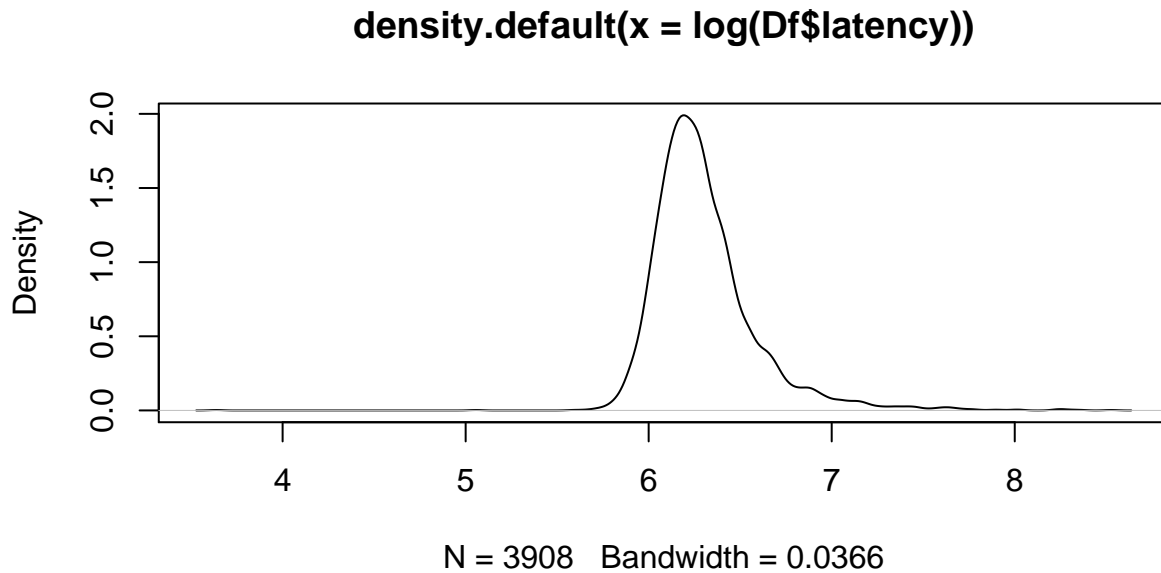
**My prettier histogram**



## Density plot

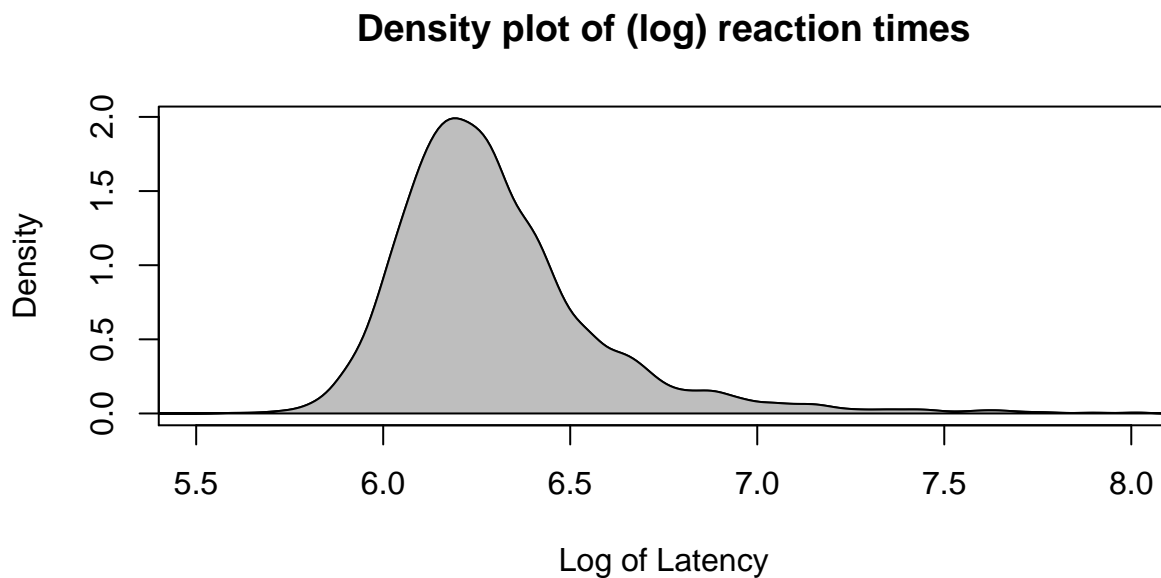
We could also do a density plot:

```
d <- density(log(Df$latency)) # get the density data
plot(d) # plots the results
```



And we'll fix that up too

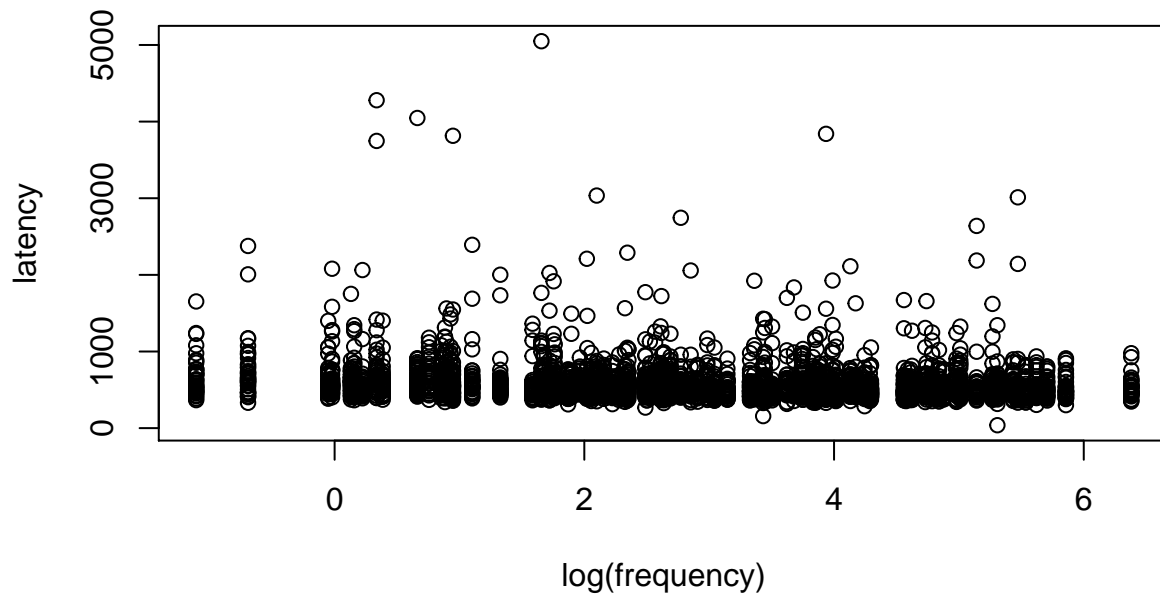
```
plot(d,
      xlab='Log of Latency',
      ylab='Density',
      xlim=c(5.5, 8),
      main='Density plot of (log) reaction times')
polygon(d, col="gray", border="black")
```



## Scatterplots

We can make scatterplots with the `plot` function and the formula  $y \sim x$ , where  $y$  is the outcome variable (y axis variable) and  $x$  is the predictor variable (x axis variable).

```
# Plot latency as a function of log frequency
plot(latency ~ log(frequency), data=Df)
```



That's a big ugly, so we'll fix it up:

```
plot(latency ~ log(frequency),
     xlim=c(-1, 6),
     ylim=c(250, 1500),
     xlab='Log of word frequency',
     ylab="Reaction time (seconds)",
     main='Predicting reaction times',
     pch=16, # point character
     cex=0.5,
     col='blue',
     data=Df)
abline(lm(latency ~ log(frequency), data=Df), col='red', lwd=3)
```

## Predicting reaction times

