• **Q1 - What comment can you make about the log transformation of the data?**
   Without the transformation only a few high intensity dots are visible. After transformation even relative low intensity spots become visible.

• **Q2 - Do you think the picture is more or less detailed after the log transformation?**
   Well, more! Importantly, defects or abnormalities in the array are often visible only when the data has been log-transformed.

• **Q3 - Comments about the transformation on the data?**
   Without log transformation, most of the data points are located in the lower-left corner of the plot. Log transformation is often used when visualizing microarray data, because it brings out the structure of the data. Traditionally, we use logarithm base 2 ($\log_2$) with microarray data.

• **Q4 - Plot the intensities of different CEL files against each others.**
   Try to play around with the different data columns, e.g. try:
   ```r
   h3 <- hexbin(log2(intensity(mixture.batch)[, 1]),
                 log2(intensity(mixture.batch)[, 9]))
   plot(h3, main = "CNS vs. liver")
   ```

• **Q5 - What observation can you make?**
   These plots are dominated by the dark gray diagonal line, indicating that the majority of probe intensities are relatively unchanged between the two samples. The lighter gray hexagons off the main diagonal represent a smaller number of probes with a noticeable difference. If you look at two samples from different tissues (e.g. 1 and 9) you might notice an increased number of changed probes relative to comparisons between two samples from the same tissue (e.g. 1 and 2).

• **Q6 - Comments?**
   Without log transformation, most of the data points lie in the far left, so most of the plotting region is used to represent a small proportion of the data. As before, we use log transformation so we can better visualize the full range of data.

• **Q7 - What observation can you make?**
   The distribution of non-normalized data shows systematic differences between the samples. Normalization removes these systematic differences, giving each sample the (approximately) same distribution of values.
Q8 - How would you perform the normalization method called 'constant'?
This is actually an exercise in using R’s help system. You might first look up the normalize function:
```r
?normalize
```
The help on this function suggests that you look up:
```r
?normalize.AffyBatch
```
... which in turn tells you that this might work:
```r
mixture.batch.const <- normalize(mixture.batch, method="constant")
```
Alternatively, you might have just guessed the right answer without using the R help system at all...

Q9 - There are typically around 11–20 unique probes matching each target (i.e. each gene), so how come the number of data points has decreased with more than a factor of 20?
On all but the newest Affymetrix GeneChips, each ‘unique probe’ actually consists of a pair of probes, one Perfect Match (PM), which is the actual probe, and one MisMatch (MM), which is the same probe, but for one base (position 13) which has been mutated. Hence, each ‘unique probe’ actually returns two intensities, one for th PM and one for the MM probes. Thus the factor above should not be between 11-20, but between 22-40.

Q10 - What do the plots tell you?
The un-normalized data show a fairly clean bell curve as would be expected for a more or less random correlation with the dilution vector. The normalized data show a mixture of two bell curves each clearly centered away from zero. These two curves represent correlation and anti-correlation with the dilution vector, which is to be expected with the experimental setup used. The measured mRNA concentrations show greater correlation to the degree to which we have diluted them with normalization than without: Therefore normalization does not destroy biological information, but rather eliminates noise, making the biological information more apparent.