# Statistics 571 — Lab 1

The following tasks are things you should try in discussion section. Your TA will demonstrate these commands and can help if you get stuck. Those of you with some previous R experience can help those without.

Also note that the typing in of R commands is aided greatly by using the arrow keys. In particular, the up and down arrow keys will recall previous commands; do this and edit rather than typing each new command from scratch.

1. Read in the cow data. Note: you need to change the working directory in R to be the one where the `cows.txt` file is located. It is recommended to make a separate directory for each project.

   ```
   > cows = read.table("cows.txt", header = T)
   > str(cows)
   ```

2. Load the `lattice` library.

   ```
   > library(lattice)
   ```

3. Use the function `densityplot()` to display the distribution of variables *dry*, *fat*, *milk*, and *protein*. Most `lattice` functions are of the form *functionName(formula,data frame)*, where *formula* is an expression involving names of variables in the data frame. For `densityplot()`, the format of the formula is the tilda character followed by the variable name.

   ```
   > densityplot(˜dry, cows)
   > densityplot(˜fat, cows)
   > densityplot(˜milk, cows)
   > densityplot(˜protein, cows)
   ```

4. Use the function `dotplot()` to plot *fat* separated by each level *treatment*.

   ```
   > dotplot(˜fat | treatment, cows)
   ```

   Note that the natural order of treatment groups by level is not displayed; rather treatment levels are alphabetical. We can correct this with the lattice command `reorder()`.

   ```
   > cows$treatment = reorder(cows$treatment, cows$level)
   > dotplot(˜fat | treatment, cows)
   ```

   Notice that it appears that the distribution of fat increases as the level of the additive increases. (We will learn to test this formally later in the semester.)

5. For each variable *dry*, *fat*, *milk*, and *protein*, compute the mean, median, and sd for all observations and separately for each treatment group. Note the use of the command `with()` which provides a means to not type the name of the data frame and the dollar sign at every instance of a variable name and is a useful alternative.

```
> mean(cows$dry)
> with(cows, mean(dry))
> sapply(split(cows$dry, cows$treatment), mean)
> with(cows, sapply(split(dry, treatment), mean))
> with(cows, sapply(split(dry, treatment), median))
> with(cows, sapply(split(dry, treatment), sd))
```

Which of these variables have an apparent trend (increasing or decreasing mean) associated with the treatment?

6. One factor used to determine the value of the milk is the total amount of protein produced. The variable *protein* measures grams of protein per 100g of milk. The variable *milk* is the average daily weight (kg) of milk produced. Create a variable for the average daily weight of protein in the milk and compare this variable by treatment group.

   ```
   > cows$total.protein = with(cows, protein * milk)
   > dotplot(total.protein ~ treatment, cows)
   > with(cows, sapply(split(total.protein, treatment), mean))
   ```

7. Repeat the previous problem to examine total fat.

   ```
   > cows$total.fat = with(cows, protein * fat)
   > dotplot(total.fat ~ treatment, cows)
   > with(cows, sapply(split(total.fat, treatment), mean))
   ```

8. Compute weight gain as the difference between final weight and initial weight. Compare weight gain by treatment group.

   ```
   > cows$weight.gain = with(cows, final.weight - initial.weight)
   > dotplot(weight.gain ~ treatment, cows)
   > with(cows, sapply(split(weight.gain, treatment), mean))
   ```

9. Use any remaining time to either explore other aspects of the cow data set or read in and examine the plantation data set.

10. You may also work through examples in the R for Statistics 571 handout.