Practical 2

Jumping Rivers

In this question, we are going to use a `for` statement to loop} over a large data set and construct some scatter plots. To generate the data, run the following piece of R code

```
data(experiment, package = "jrProgramming")
head(exper)
```

The data frame `exper` represents an experiment, where we have ten treatments: $A, B, \dots, J$ and measurements at some time points. We want to create a scatter plot of measurement against time, for each treatment type.

1. First we create a scatter plot of one treatment:

   ```
   library(dplyr)
   treat_a = filter(exper, treat == "A")
   plot(treat_a$time, treat_a$values)
   ```

2. To generate a scatter-plot for each treatment, we need to iterate over the different treatment types:

   ```
   for(treatment in unique(exper$treat)) {
     group = filter(exper, treat == treatment)
     plot(group$time, group$values)
     readline("Hit return for next plot")
   }
   ```

   - What does `unique(exper$treat)` give?
   - In the `for` loop, what variable is changing? What are it's possible values?
   - What does the `readline()` function do?

Questions

1. The default axis labels aren't great. So we can change the x-axis label using `xlab`:

   ```
   plot(group$time, group$values, xlab="Time")
   ```

   Use the `ylab` argument to alter the y-axis label.

2. To add a title to a plot we use the `main` argument, viz:

   We can combine strings/characters using the `paste()` function,

   Rather than have a static title, make the title of each plot display the treatment type.

3. The y-axis range should really be the same in all graphics. Add a `ylim` argument to fix the range. **Hint**: Work out the range before the `for` loop.

4. At each iteration, use the `message()` function to print the average measurement level across all time points.

5. On each graph, highlight any observations with a blue point if they are larger than the mean + standard deviations or less than the mean - standard deviations. Use the `points()` function to highlight a point. **Hint**: You don't need `if` statements here. Just subset your data frame and pass this new data frame to the points function. For example, to highlight the points (1,2) and (3, 4) we use the command:

```
points(c(1, 3), c(2, 4), col = 2)
```

6. Suppose we wanted to save individual graphs in a pdf file. Add the `pdf()` function to your code save the resulting graph. To get unique file names, use the paste command:

7. Put your code, i.e. the `for` loop and plotting commands, in a function which takes the data frame as an argument.

Solutions

Solutions are contained within this package:

```
vignette("solutions2", package = "jrProgramming")
```