Homework 3: An Insufficiently Random Walk Down Wall Street

36-402, Advanced Data Analysis

Due at 11:59 pm on 4 February 2013

> INSTRUCTIONS: Submit *one* PDF file containing your written answers and all your figures and tables; the filename should include your Andrew ID and the assignment number. Submit one plain text file (`.txt` or `.R`) containing all your R code, commented if possible, also named with your Andrew ID and the assignment number. If you cannot manage to do this, you can include the R as an appendix to your PDF file.

In this assignment, you will work with a data set of historical values for the S& P 500 stock index, which also features in the notes. You will need to download `SPhistory.short.csv` from the class website. This data set records the actual *prices* of the index, say P_t on day t, but in finance we actually care about the returns, $\frac{P_t}{P_{t-1}}$, or about the logarithmic returns,

$$R_t = \log \frac{P_t}{P_{t-1}}$$

since we care more about whether we're making 1% on our investment than $1 per share. In this assignment, "returns" always means "logarithmic returns".

Problems 2 and 3 are about estimating the first percentile of the return distribution, $Q(0.01)$, under various assumptions. The returns will be larger than this 99% of the time, so $Q(0.01)$ gives an idea of how bad the bad performance will be, which is useful for planning. Note that a calendar year contains about 250 trading days, and so should average two or three days when returns are even worse than $Q(0.01)$. Problems 4 and 5 are about predicting future returns from historical returns, and the uncertainty in this. Doing all the bootstrapping for problem 5 may be time-consuming, and should not be left to the last minute.

Include code for all problems. Clearly indicate which block of code is for which problem. Comment your code when at all possible; it is much easier to give partial credit when we can tell what you are doing.

1. (5) Load the data file, take the last column (containing the daily closing price), and calculate the logarithmic returns. Note that the file is in reverse chronological order (newest first). When you are done, if everything worked right, running `summary` on the returns series should give

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-0.094700 -0.006440  0.000467 -0.000064  0.006310  0.110000
```

Hint: `help(rev)` and Recipe 14.8 in *The R Cookbook*.

2. In finance, it is common to model daily returns as independent Gaussian variables.

   (a) (5) Find the mean and standard deviation of the returns. What is $Q(0.01)$ of the corresponding Gaussian distribution? *Hint:* `qnorm`.

   (b) (5) Write an expression which will generate a series of independent Gaussian values of the same length as the returns, with the mean and standard deviation you found in 2a. Check that the mean and standard deviation of the output is *approximately* right, and that their histogram looks like a bell-curve.

   (c) (10) Write a function which takes in a data vector, calculates its mean and standard deviation, and returns $Q(0.01)$ according to the corresponding Gaussian distribution. Check that it works by seeing that it matches what the answer you got in 2a when run on the actual returns.

   (d) (10) Using the code you wrote in 2b and 2c, find a 95% confidence interval for $Q(0.01)$ from 2a. *Hint*: Look at the examples in the notes of parametric bootstrapping.

   (e) (5 points) What is the first percentile of the data? Is it within the confidence interval you found in 2d? *Hint:* `quantile`.

3. (a) (5) Use `hist` to plot the histogram of returns. Also plot, on the same graph, the probability density function of the Gaussian distribution you fit in problem 2a. Comment on their differences.

   (b) (5) Write a function to resample the returns; it should generate a different random vector of the sample length as the data every time it is run. Check that running `summary` on these vectors produces results close to those on the data. *Hint:* Look at the examples in the notes of resampling.

   (c) (5) Write a function to calculate $Q(0.01)$ from an arbitrary vector, *without* assuming a Gaussian distribution. Check that it works by seeing that its answer, when run on the real data, matches what you found in 2e.

   (d) (10) Using the code you wrote in 3b and 3c, find a 95% confidence interval for $Q(0.01)$. Compare this to your answer in 2d. Which is more believable, and why? *Hint:* Look at the examples in the notes of non-parametric bootstrapping.

4. (10) Using `npreg`, fit a kernel regression of R_{t+1}, tomorrow's returns, on R_t, today's returns. (Use the automatic bandwidth selector.) Report

the selected bandwidth and the in-sample mean-squared error. Make a scatter-plot with R_t on the horizontal axis and R_{t+1} on the vertical axis, and add the estimated kernel regression function. Comment on the shape of the curve. *Hints:* Make a data frame with R_t as one column and R_{t+1} as another column. Also, see examples in the notes of plotting fitted models from `npreg`.

5. (25) *Uncertainty in the kernel regression*

   (a) (5) Write a function which resamples (R_t, R_{t+1}) pairs from the returns series, and produces a new data frame of the same size as the original. Check that it works by running `summary` on it, and seeing that both columns *approximately* match the summaries of the data. *Hint:* look at the examples of resampling cases for regression in the notes.

   (b) (10) Write a function which takes a data frame with appropriately-named columns, and runs a kernel regression of R_{t+1} on R_t. It should return fitted values at 30 evenly-spaced values of R_t which span its observed range.

   (c) (10) Using your code from 5a and 5b, add 95% confidence bands for the kernel regression to your plot from problem 4. *Hint:* See the examples of plotting bootstrapped nonparametric regressions in the notes.