Homework 2 – Due Tuesday, Feb. 24

STAT-GB.2302, STAT-UB.0018: Forecasting Time Series Data

In Problems 1–3, we will analyze the Russell 2000 Stock Index, adjusted daily closing price, recorded from 10 September 1987 to 11 February 2015. Specifically, we will work with the mean-adjusted series $x_t = \text{Russell}_t - \overline{\text{Russell}}$. To create x_t first read in the data:

```
data <- read.csv("http://ptrckprry.com/course/forecasting/data/russell.csv")
date <- as.Date(data$date)
russell <- data$russell
```

Next, run the command

```
today <- russell - mean(russell)
```

The `today` variable contains x_t, Today's Russell.

Problem 1

(A) On a single plot, draw Today's Russell versus time, as well as Yesterday's Russell versus time. To create Yesterday's Russell, run the command

```
yesterday <- c(NA, today[-length(today)])
```

Use different line types for the two series.

Next, on a single plot, draw Today's Russell versus time, as well as (0.5)(Yesterday's Russell) versus time.

(B) Based on these two plots, which seems to be a better forecast of Today's Russell: Yesterday's Russell, or (0.5)(Yesterday's Russell)?

(C) Calculate the average squared forecast errors for the two forecasts. Based on this, which one was better?

Problem 2

(A) Plot Today's Russell versus Yesterday's Russell. Describe any patterns you see.

(B) Run a linear regression of Today's Russell (dependent variable / response) on Yesterday's Russell (independent variable / predictor). What is the prediction of Today's Russell implied by the regression coefficients? Is this consistent with your answers to Problem 1, parts B and C?

(C) Is the slope in your fitted regression significantly different from 1? Briefly comment on the intercept as well. (Unfortunately, as we will see later, the p-values for the slope and intercept cannot necessarily be trusted when we regress a time series on a lagged version of itself, that is, x_t on x_{t-1}. Furthermore, the p-values cannot necessarily be trusted when we regress x_t on t.)

(D) Based on everything you have done so far, do you see any strong evidence that the Russell is *not* a random walk?

(E) Compute the correlation coefficient between Today's Russell and Yesterday's Russell. (This is equal to the square root of R^2 if the slope in the fitted regression is positive; it is equal to $-\sqrt{R^2}$ if the fitted slope is negative.) Based on this, how strong is the linear association between Today's Russell and Yesterday's Russell? (Note: The correlation coefficient you get here should be quite close to the value of the slope you got in part C.)

Problem 3

(A) Returning now to the non-mean-adjusted data, compute and plot the Russell returns, defined as

$$\text{return}_t = \frac{\text{Russell}_t - \text{Russell}_{t-1}}{\text{Russell}_{t-1}},$$

versus time. Compute the sample average and standard deviation of the returns. Based on an ordinary t-test, are the mean returns significantly different from zero? Interpret your findings.

(B) Plot a histogram and boxplot of the Russell returns. Also, try a normal quantile-quantile plot (also called a normal probability plot), which should reveal an approximately straight-line pattern under normality. Do you think that the Russell returns are normally distributed? Explain.

(C) Plot today's returns versus yesterday's returns. Does this plot appear very different from the one in Problem 2(A)? Which seems to be easier to predict: Today's Russell, or Today's return?

(D) Run a linear regression of today's returns (dependent variable) on yesterday's returns (independent variable). What is the prediction of today's return implied by the regression coefficients? Are the coefficients statistically significantly different from zero?

Problem 4

If $\{x_t\}$ is stationary with $E[x_t] = 0$ and $\text{corr}(x_t, x_{t-1}) = \rho_1$, show that the best linear predictor of x_t based on x_{t-1} is $\rho_1 x_{t-1}$. You will need to use calculus to do this problem.

Here are some hints: First, define the random variables $Y = x_t$ and $X = x_{t-1}$. Consider any linear predictor $\hat{Y} = a + bX$, where a and b are any numbers. Consider the mean squared forecasting error, $\text{MSE} = E[Y - \hat{Y}]^2 = E[Y - (a + bX)]^2$. Take the derivative of MSE with respect to a and set it equal to zero. Similarly, take the derivative of MSE with respect to b and set it equal to zero. Let's assume that the solution to these two equations for a and b gives us the coefficients which *minimize* MSE. By solving these two equations, you should conclude that the best a and b are given by $a = 0$ and $b = E[XY]/\text{Var}[X]$. Now, use the fact that $\{x_t\}$ is stationary with $E[x_t] = 0$ to show that the above expression for b is the same is ρ_1 in this case.

New R commands used in this assignment

- `boxplot`. Produce a boxplot. See the "Introduction to R" handout for examples.
- `c`. Concatenate two or more vectors together. Examples:

```
x <- c(1, 3, 4, 9)
y <- c(2, -2, 8)
z <- c(x, y)
```

- `confint`. Compute a confidence interval for a regression coefficient. Examples:

```
  model <- lm(weight ~ age + height)

  confint(model, "age") # 95% confidence interval for the age coefficient

  confint(model, "height", level = 0.99) # 99% confidence interval

  confint(model, "(Intercept)") # confidence interval for intercept
```

- `cor`. Compute a correlation coefficient. Examples:

```
  # sample correlation between vectors x and y:
  cor(x, y)

  # compute correlation after removing missing observations
  cor(x, y, use="complete.obs")
```

- `hist`. Produce a histogram. See the "Introduction to R" handout for examples.
- `lines`. Add lines to an existing plot.
- `mean`. Compute the mean of a vector. Note: if the vector contains missing values (`NA`), then you need to specify `na.rm=TRUE` to remove these values. Examples:

```
x <- c(1, 2, 3, NA, 5)
y <- c(3, 5, 8, 1, 2)

mean(x) # gives NA

mean(x, na.rm=TRUE) # gives 2.75

mean((x - y)^2, na.rm=TRUE)
```

- `qqnorm`. Produce a normal quantile-quantile plot. See the the "Introduction to R" handout for an example.
- `sd`. Compute the sample standard deviation of a vector. As with the `mean` function, if the vector contains missing values, you need to specify `na.rm=TRUE` to remove these values.
- `t.test`. Perform a t test on a population mean. See the "Introduction to R" handout for examples.