STAT 505: Project 1

Bruce D. Marron

May 14, 2014

Some Initial Observations

Law[1] clearly recognizes the nature of simulation work as a scientific exercise with this definition: "a simulation is a computer-based statistical sampling experiment" (p.485). He goes on, "if the results of a simulation study are to have any meaning, appropriate statistical techniques must be used to design and analyze the simulation experiments." Acknowledgement of the validity of these statements launched my study of statistics given my interest in simulation modeling.

The general nature of simulation work thus clarified, Law notes various reasons why computer-based simulations have not been subject to the statistical rigor of design and analysis afforded other scientific experiments. He also alerts the would-be model builder to potential statistical pitfalls and establishes the axioms and assumptions that are necessary for the application of statistical techniques that, in turn, can lead to valid scientific inference from simulation experiments. Law does not, however, explicitly discuss the relationship between simulation modeling and time series analysis.

I have come to recognize that simulation modeling and time series analysis are the flip sides of the same coin. Simulation modeling is a bottom-up approach that seeks to understand a process generatively, by creating a stochastic model that mimics the observational output of the process. Time series analysis is a top-down approach that seeks to understand a process empirically by using statistical methods to characterize the observed stochasticity in the observational output. Epistemology is best served when the approaches are harnessed in tandem and the project discussed below is a first attempt at that coupling.(cf. Law, Shumway and Stoffer[2], and Luenberger[3]).

[1]Law, Averill M. 2006. Simulation Modeling and Analysis. 4th ed. McGraw-Hill Series in Industrial Engineering and Management Science. Boston: McGraw-Hill.

[2]Shumway, R. and Stoffer, D. 2011. Time Series Analysis and Its Applications. Springer

[3]Luenberger, D. 1979. Introduction to Dynamic Systems: Theory, Models, and Applications. John Wiley and Sons

Law's Methods, Axioms and Assumptions

Law offers suites of methods for estimating measures of performance, θ, which are taken as point estimates and their associated confidence intervals for means, probabilities, and quantiles. Law only briefly makes mention of a jackknife method for ratio estimators (p 542). Methodologies are keyed to the stochastic output of the two major classes of simulation models; namely, models with a 'natural' terminating event (terminating simulations) and models without such an event (non-terminating simulations). Law acknowledges that transient, steady-state and cyclic behavior can (and does) occur within each of the major classes of models. Statistical methodologies and recommendations for their use are grouped under the general categories of *fixed-sample-size procedures* and *sequential procedures* for both terminating and non-terminating simulations. Law's recommendations are based on the literature and on the results of his own experiments. For terminating simulations there is basically just one, fixed-sample-size method and one, sequential method (pp 503 - 507). For non-terminating simulations there are six, fixed-sample-size methods (pp 520 - 529) and over eight sequential methods (pp 529 - 534). Note that many of the non-terminating procedures which are presented have direct analogs in time series analysis.

As the basis for all of the methods presented, Law states two axioms and maintains two assumptions. The first axiom of simulation modeling underscores the role of stochastic processes; namely, that stochastic processes are the fundamental drivers in both simulation models and in the real-world systems that such models seek to represent. An important corollary is that outside of some simple physical systems most real-world systems (especially complex adaptive systems) do not have stationary outputs. That is, the probability distribution of most stochastic processes $\{X_i\}$ are not invariant under a shift in time. The second axiom is that virtually all simulations are autocorrelated. Thus classical statistics (i.e., point estimators for means, totals, proportions, and variances) are not directly applicable because (a) the covariances are nonstationary, and (b) the observations or samples that are derived from simulation outputs are not independent and identically distributed (IID).

Law makes two fundamental assumptions that allow the subsequent application of his various methods for statistical inference. For the first assumption, let $\{y_{11}, y_{12}, ..., y_{1m}\}$ be a realization of the random variables$\{Y_1, Y_2, ..., Y_m\}$ that result from generating a single run of a simulation of length m using the set of random numbers $\{u_{11}, u_{12}, ..., u_{1m}\}$. If a different set of random numbers is used for a second run (still of length m), then the realization of the same random variables $\{Y_1, Y_2, ..., Y_m\}$ would be $\{y_{21}, y_{22}, ..., y_{2m}\}$. Law assumes that while samples *within* any single run are not IID, samples *between* independent replications (runs) are, in fact, IID. The second assumption is that the stochastic processes embedded in simulation models will be sufficiently covariance-stationary such that the models will, in general and ultimately, exhibit steady-state behavior. Even with these assumptions, Law states that the methods presented for estimating measures of

performance or parameters θ, are still likely to have problems. Specifically, $\hat{\theta}$ will not be an unbiased estimator of θ, and $v\hat{a}r(\theta)$ will not be an unbiased estimator of $var(\theta)$. Regrettably, the second assumption has substantially constrained my use of Law's methods and approaches because the models of greatest interest to me are decidely nonstationary and are unlikely to exhibit steady-state behavior.

General Project Description

The project begins with a question: What results obtain if a simulation is constructed from a known, dynamic linear model and the simulation output is subjected to time series analysis? This question seems an obvious inquiry into the synthesis of the realms of stochastic model building, simulation, state-space (or dynamic linear) models, and statistical analysis. As a first-step in exploring this question I have adapted a problem from Shumway and Stoffer (Problem 2.3, p. 78). I constructed a simple, 'random walk with drift' model called, "RandWalk1" using the R statisitcal computing package (version 3.1.1 (2014-04-10)) freely available from The R Foundation for Statistical Computing. The annotated code for RandWalk1 is presented in Figure 1. This model, which could be considered as a fundamental archetype of the models used in time series analysis for analyzing trends (Shumway and Stoffer, 2011), is absolutely equivalent to

- a univariate, state-space (or dynamic linear) model in its most basic form, $\mathbf{Y} = \mathbf{Ax} + \varepsilon$
- a linear, first-order difference equation: $y(k+1) = ay(k) + b + e$ (where a = 1, b = forcing term, e = random shock)
- an AR(1) model

The RandWalk1 model has the form,

$$\begin{pmatrix} y_{t+1} \\ \delta \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} y \\ \delta \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} (AF)(\varepsilon \sim N(0,1))$$

and maps the one-dimensional, stochastic displacement of a single point to $\mathbf{R}^1$ through time. The dynamics of RandWalk1 are generated recursively with the 2 x 2 transition matrix. Note that the trend or forcing term (δ) is added as a constant value through the transition matrix. This does not change the fact that the model is univariate. Stochasticity in the model is generated by Gaussian white noise via random variates which are pulled at every recursive iteration from the normal distribution with $\mu = 0$ and $\sigma^2 = 1$. The model is designed so that the random variates can be multiplied by an augmention factor (AF), if desired.

Experimental

An experiment was designed to evaluate classical, univariate least squares regression in the time series context where the time series data are generated by a simulation. The

Figure 1: A 'random walk with trend' simulation model.

```
RandWalk1_Model.txt                                                    1 / 1


Model:RandWalk1
Author:B.Marron
OriginalDate:25 Apr 2014
CurrentDate:01 May 2014

--- About the Model ---
This model is a 'random walk with drift' model, often useful for analyzing trends
(Shumway and Stoffer, 2011).
The model is equivalent to,
        * a univariate state-space (or dynamic linear) model in its most basic form
        * a linear, first-order difference equation: y(k+1) = ay(k) + b + e
[a=1, b=forcing term, e=random shock]
        * an AR(1) model


The dynamics are generated recursively with the n x n transition matrix, GG. Note
that the trend or forcing term (FT)
is added as a constant value through the transition matrix. Stochasticity in the
model is generated from a single
random variate pulled (at every iteration) from the normal distribution with
mean=0 and var=1.  The model is designed
so that the random variate can be multiplied by an augmention factor (AF), if
desired.

--- About the Simulation ---
Independent replications (runs) of the model (using an outer 'for' loop) allow for
simulation of the model where,
        *'reps' is the number of replications of the simulation
        *'tiks' is the (arbitrarily determined) length of the simulation

The simulation could be considered as non-terminating but by ending it at tiks=
(some positive integer>0),
a 'natural' terminating event is imposed (ie the number of tiks).  The various
simulations can be plotted nicely with
xyplot (pkg 'lattice').  Note that outputting data from loops in R requires that a
data frame or a matrix or a vector
be set up outside of the loop. Thus, the 'sim1data1' data frame is defined outside
of the replications loop and
the 'X' matrix is defined inside the replications loop but outside of the
iterations loop.

--RandWalk1 --
sim1data1<-data.frame(x=1:1000)
reps<-5

for (j in 1:reps){

    tiks<-999
    FT<-.1
    af<-1
    X0<-c(0,FT)
    p <- length(X0)
    X <- rbind(X0, matrix(0, tiks, p))
    GG<-matrix(c(1,1,0,1), 2, byrow=TRUE)

        for (i in 1:tiks) {
                X[i+1, ] <- GG %*% X[i, ] + (AF*matrix(c(rnorm(1, 0, 1), 0)))
                }

    sim1data1[ ,j]<-X[ ,1]
        }


library("lattice")
xyplot(sim1data1[,1] + sim1data1[,2] + sim1data1[,3] + sim1data1[,4] +
sim1data1[,5] ~ 1:1000, type="l")
```

experiment consisted of a set of five (5) trials where each trial generated data from a different number of independent replications (n = 1, 5, 25, 100, 500) of the RandWalk1 model. All replications (or runs), regardless of trial, used identical initial conditions ($y_{t=0} = 0; \delta = .02$ and AF =1) and an identical termination point (500 model tiks; t=500). Each individual run of the simulation created a univariate, time-series data set $\{ y_1, y_2, ...y_{500}\}$ or equivalently, 500 observations per run. For the trials where n>1, the multiple time-series data sets that were produced could be considered as repeated measures, analogous to replicated observations from a chemical analysis. Independence of the replications was assumed to follow from the use of different sets of random variates for each run. The experiment, and all subsequent evaluations, were performed using R.

The following procedure was used to derive a single, ordinary least-squares (OLS) linear regression model for each experimental trial. First, the sets of y_t from each replication, n, of the simulation were fitted to a simple linear regression model in R using the lm() function. The OLS regression model as determined for each individual run is given as,

$$y_t = \beta_1 z_{t1} + \beta_2 z_{t2}$$

where $z_{t1} = 1$ and $z_{t2} = t$. Note that under this procedure a unique regression model would be generated for each replication. Owing to the fact that n = 1 for Trial 1, this first step was sufficient to generate the single, OLS regression model for Trial 1.

Next, where the number of replicatons was greater than one (i.e., Trial 2 (n = 5), Trial 3 (n = 25), Trial 4 (n = 100)), and Trial 5 (n = 500)), the sample means β_1 and β_2 were determined as,

$$\hat{\mu}_{\beta_{ia}} = \frac{\sum_1^n \beta_{ia}}{n} \qquad for\ i = 1, 2\ and\ a = 1, 2...5$$

where a indexes the trials and n indexes the number of replications per trial. The values $\hat{\mu}_{\beta_{1a}}$ and $\hat{\mu}_{\beta_{2a}}$ as derived above were subsequently taken as the coefficients for each of the single, OLS regression models. In order define each of the single, OLS regression models (one per trial) and to generate predicted values from these models, the appropriate coefficients $\hat{\mu}_{\beta_{1a}}$ and $\hat{\mu}_{\beta_{2a}}$ were inserted into an lm() object file.

As a last step, a *500* x n matrix of residuals was generated per trial as the difference between the y_t values originally outputted by the simulation (per trial) and the predicted values from the single, regression model (per trial). An example of the R code used to realize each trial of the experiment is shown in Figures 2 and 3. Plots of the simulation outputs (per trial) as well as plots of residuals (per trial) are presented in the Appendix.

Figure 2: Trial 2 of the simulation experiment.

```
Sim1Exp1Trial2.txt                                                              1 / 2


Simulation:     Sim1
Model:          RandWalk1
Experiment:     Exp1
Title:          Trial2

Author:B.Marron
OriginalDate:08 May 2014
CurrentDate:11 May 2014
Cross-Ref.s: Sim1Exp1.txt; RandWalk1_Model.txt


--------------------Trial 2: n=5 ---------------------------------
set.seed(29)
sim1exp1tr2.d<-data.frame(x=1:500)                      #.d = data
reps<-5

for (j in 1:reps){

    tiks<-499
    FT<-.02
    AF<-1
    X0<-c(0,FT)
    p <- length(X0)
    X <- rbind(X0, matrix(0, tiks, p))
    GG<-matrix(c(1,1,0,1), 2, byrow=TRUE)

        for (i in 1:tiks) {
                X[i+1, ] <- GG %*% X[i, ] + (AF*matrix(c(rnorm(1, 0, 1), 0)))
                }

    sim1exp1tr2.d[ ,j]<-X[ ,1]
}


#Full Data Graphics (saved as sim1exp1trial2.jpeg)
library("lattice")
time<-c(1:500)

xyplot(sim1exp1tr2.d[,1] + sim1exp1tr2.d[,2] + sim1exp1tr2.d[,3] +
sim1exp1tr2.d[,4] + sim1exp1tr2.d[,5] ~ time,

type="l", xlab="Model Tiks", ylab="Displacement", main="Trial 2 (n = 5)",

panel = function(...) {
           panel.abline(a=0, b=.02, lty = 2)
           panel.text(450,14,labels="y1 = .02t")
           panel.abline(a=(-1.3), b=(.0053), lty=3)
           panel.text(350, -3, labels="y2 = -1.3 +.0053t")

           panel.text(10, 40, labels="y1 = expected")
           panel.text(20, 35, labels="y2 = experimental")
           panel.xyplot(...)
       }
)


#Ordinary Least Squares Regressions
time<-c(1:500)
sim1exp1tr2.mc<-matrix(0, 2)                    #.mc = model coefficients
sim1exp1tr2.mr<-matrix(0,500)                   #.mr = model residuals

for (k in 1:reps){
sim1exp1tr2.m<-lm(sim1exp1tr2.d[ ,k] ~ time, na.action=NULL)            #.m = model
sim1exp1tr2.mc<-cbind(sim1exp1tr2.mc, as.matrix(sim1exp1tr2.m$coefficients))
```

Figure 3: Trial 2 of the simulation experiment (con't).

```
Sim1Exp1Trial2.txt                                                              2 / 2

sim1exp1tr2.mr<-cbind(sim1exp1tr2.mr, as.matrix(sim1exp1tr2.m$residuals))
}


#Determine the mean values for each of the regression coefficients
rowMeans(sim1exp1tr2.mc[,2:6])          #exclude sim1exp1tr2.mc[ ,1] b/c [0,0]'
 (Intercept)         time
-1.294796036  0.005297731


#Define a new, OLS model [as an lm() object] using the mean values
#of the coefficients and create a set of 'fitted' values from
#the new model by using predict()
sim1exp1tr2.m$coefficients<-c(-1.295, .005298)
sim1exp1tr2.mp<-as.matrix(predict(sim1exp1tr2.m))       #.mp = model predictions

#Add 'generated' residuals as (simulation output - model prediction values)

for (j in 1:reps){
sim1exp1tr2.mr<-cbind(sim1exp1tr2.mr, (sim1exp1tr2.d[ ,j] - sim1exp1tr2.mp))
}


#Test for homoscedasticity
#Plot of residuals against time (saved as sim1exp1tr2_mr.jpeg)
#b/c time series, plotting against 'fitted' data same as plotting against time

xyplot(sim1exp1tr2.mr[,7] + sim1exp1tr2.mr[,8] + sim1exp1tr2.mr[,9] +
        sim1exp1tr2.mr[,10] +sim1exp1tr2.mr[,11] ~ time,

type="l", xlab="Model Tiks", ylab="Residuals", main= "Trial 2 (n = 5)",

panel = function(...) {
           panel.abline(a=0, b=0, lty = 2)
           panel.abline(a=10, b=.03, lty = 2)
           panel.abline(a=-10, b=-.03, lty = 2)

           panel.text(60, 40, labels="A visual test for homoscedasticity")
           panel.xyplot(...)
        }
)


#Plot of residuals generated from each rep's OLS model
#Compare to above

xyplot(sim1exp1tr2.mr[,2] + sim1exp1tr2.mr[,3] + sim1exp1tr2.mr[,4] +
sim1exp1tr2.mr[,5] +sim1exp1tr2.mr[,6] ~ time,

        type="l", xlab="Time", ylab="Residuals",

        panel = function(...) {
            panel.abline(a=0, b=0, lty = 2)
            panel.abline(a=10, b=.01, lty = 2)
            panel.abline(a=-10, b=-.01, lty = 2)
            panel.xyplot(...)
        }
)
```

Results and Discussion

Simulations of the type done in this experiment create repeated measures of time series data. Extracting statistical inferences from these situations is not trivial. I first considered using Law's recommendations for deriving a point estimator and its confidence interval for the displacement (the single random variable in the experiment) but realized that none of Law's statistics would be valid because the 'random walk with drift' model is not a covariance-stationary stochastic process. Next I considered a time-series analysis approach (i.e., classical regression in the time-series context) but discovered that the 'random walk with drift' model does not even meet Shumway and Stoffer's definition for a weakly stationary time series (p. 23). In fact, the autocovariance function of the 'random walk with drift' model is not dependent on the time lag between any two time points (for example, t_i and t_{i+2}), but rather is dependent on the values of the time points themselves. In addition there is the compounding problem of repeated measures. I hoped to overcome the repeated measures problem with the internet-published function make.rm() which manipulates a data matrix to allow the function lm() to operate on repeated measures data. (This still may be an option but needs more work.) In the end, I opted to simply validate the published mean function of a 'random walk with drift' model and perform a very simple visual test for homoscedasticity on the simulation outputs.

Shumway and Stoffer (p.18) state that the mean function of a random walk with drift model is given as,

$$E(y_t) \;=\; \delta t + \sum_{j=1}^{t} E(w_j) \;=\; \delta t \qquad\qquad where\ w_j \;=\; Gaussian\ white\ noise$$

A working hypothesis for validating this statement (based on the Central Limit Theorem) is that as the number of replications $n \longrightarrow \infty$, $\beta_1 \longrightarrow 0$ and $\beta_2 \longrightarrow \delta$. Certainly a visual review of the plots in the Appendix would seem to confirm this hypothesis: the regression model for Trial 5 differs only slightly from the expected value. A t-test or an F-test should be done to objectively assess the hypothesis but I was uncertain as to how to proceed with the determination of these test statistics given the nature of the data (i.e., repeated measures of time series).

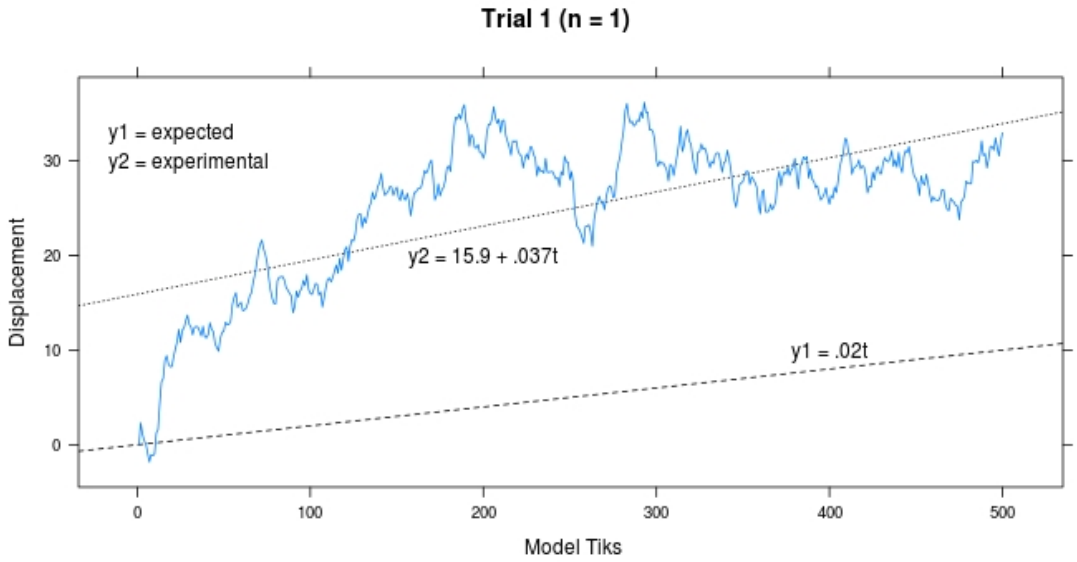
Generating a statistically solid diagnostic of the regressions also proved challenging. For example, would it be possible to manipulate the data sets so as to perform a cross-validation (preferred)? If not, could Durbin-Watson's test (autocorrelation of the residuals) be done? If Durbin-Watson's test is not possible, could the residuals be plotted against the fitted values, and if so, which residuals should be used? Standardized residuals? Studentized residuals? Is it possible to generate either standardized residuals or studentized residuals for a stochastic process with known heteroscedasticity?

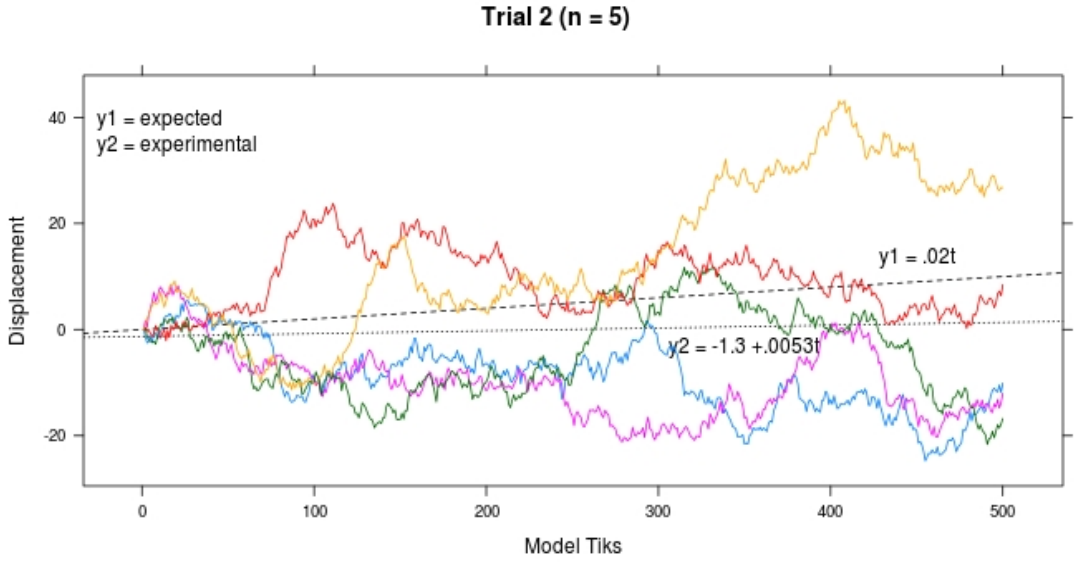
The diagnostic residual plots in the Appendix use only 'raw' residuals knowing full well that studentized residuals are preferred. Yet again I was uncertain as how to proceed. Knowing that calculation of studentized residuals requires the determination of the "leverages" (the values of h_{ij} in the so-called "hat" matrix, $\mathbf{H}$), I was uncertain how to obtain these values given my OLS procedure and the nature of the simulation data.

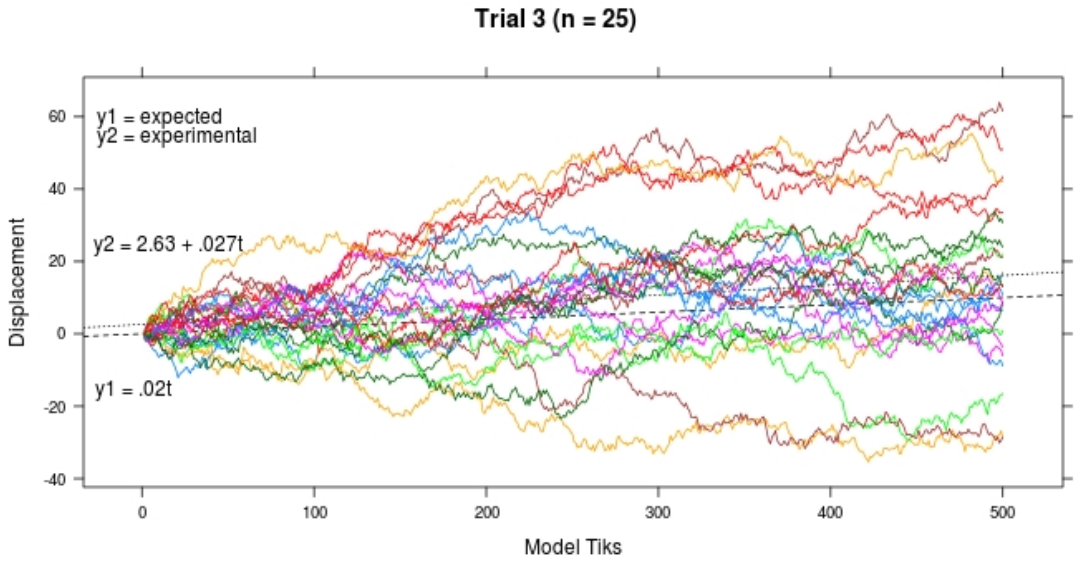
Conclusions

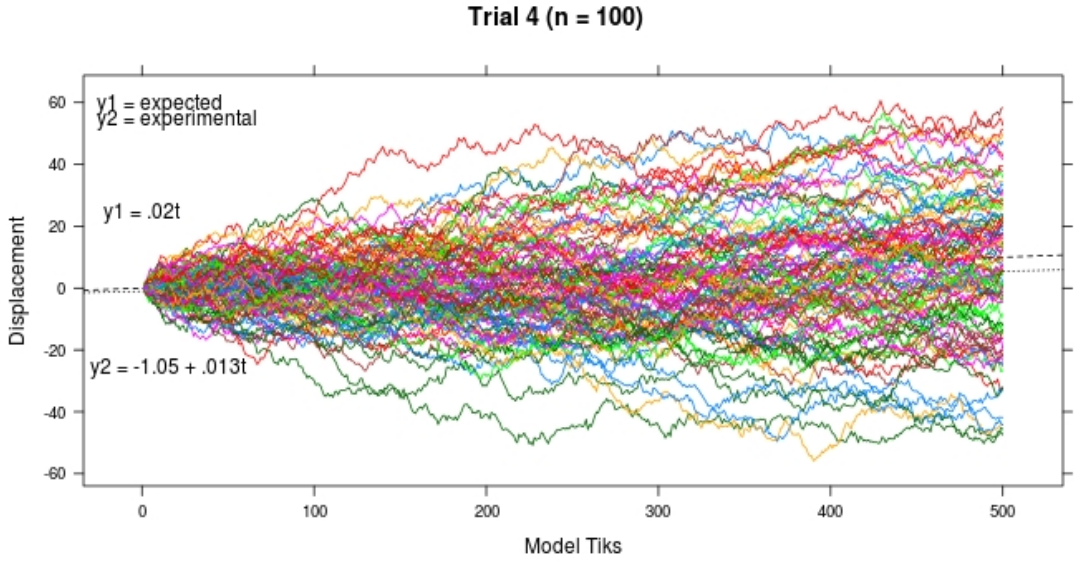
Based on this experiment, I suspect that at least 100 replications are needed for valid inference from any simulation that models a non-stationary, stochastic process. In the case of heteroscedasticity, obtaining model diagnostics and valid inferences is not a trivial task, even with the more-than-sufficient-amount of data from a simulation. Because my research interests lie in the realm of generative models such as the 'random walk with drift' model, I need to find valid statistical methodologies or algorithms that can provide valid inferences. Meanwhile, it is refreshing to validate (albeit qualitatively) that the published expected mean function and heteroscedasticity of the 'random walk with drift' model both deliver as promised.

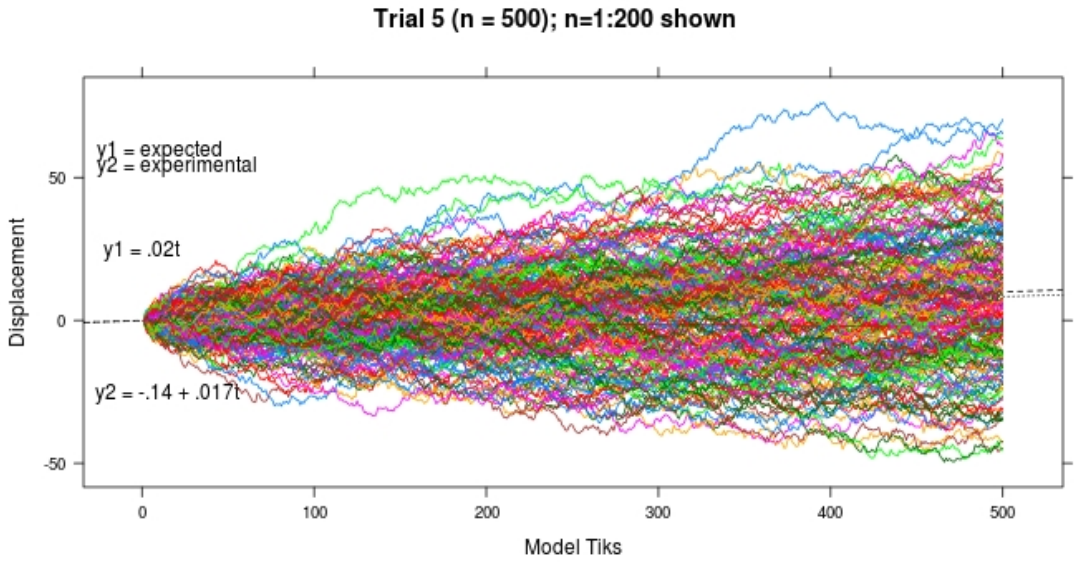
Appendix

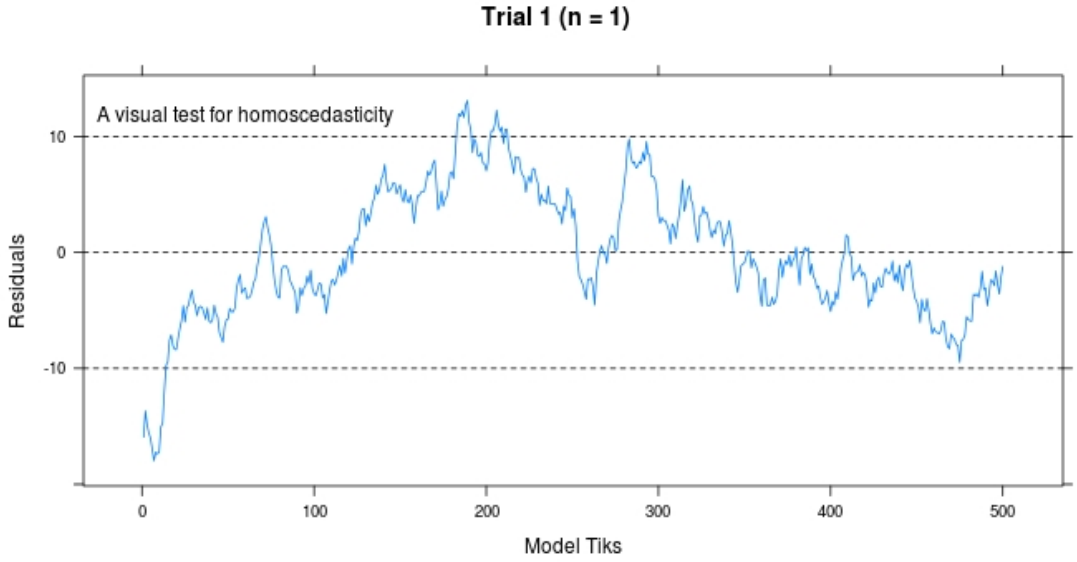

Trial 1 (n = 1)
y1 = expected
y2 = experimental
y2 = 15.9 + .037t
y1 = .02t
Displacement
Model Tiks
0
100
200
300
400
500
0
10
20
30

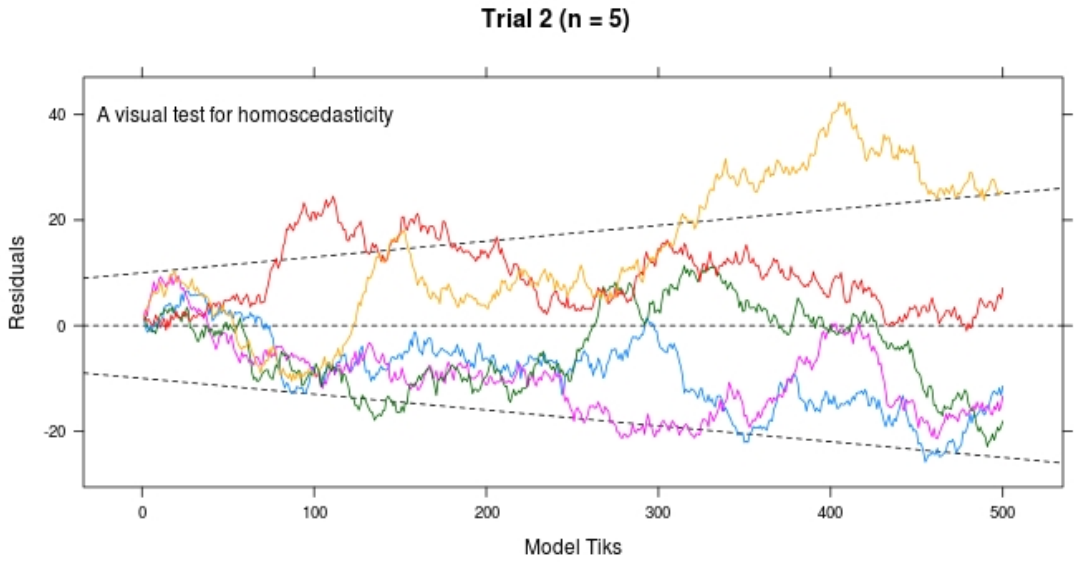

Trial 2 (n = 5)
y1 = expected
y2 = experimental
y1 = .02t
y2 = -1.3 +.0053t
Displacement
Model Tiks
0
100
200
300
400
500
-20
0
20
40

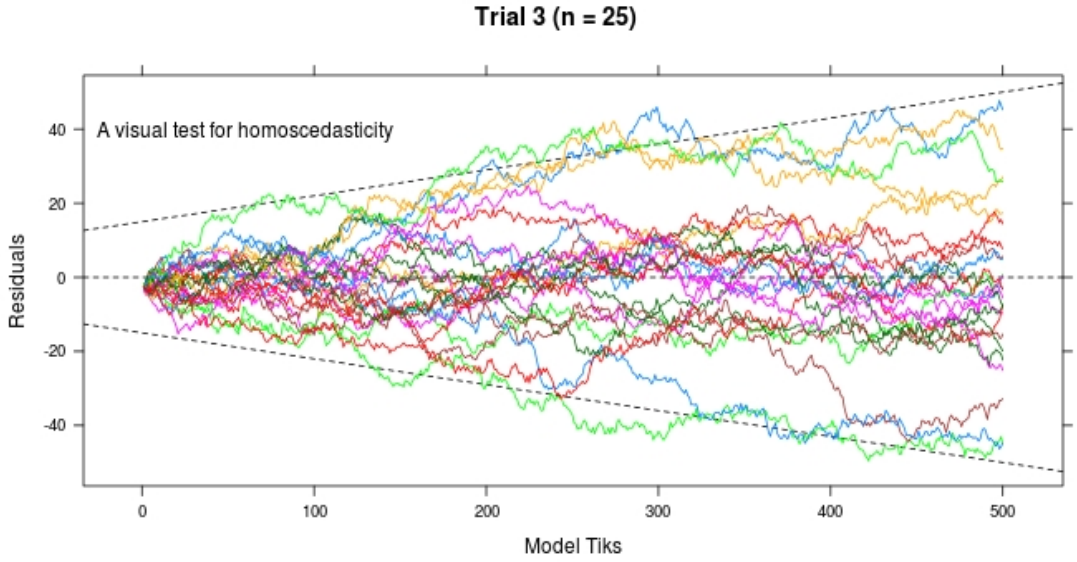

Trial 3 (n = 25)
y1 = expected
y2 = experimental
y2 = 2.63 + .027t
y1 = .02t
Displacement
Model Tiks
60
40
20
0
-20
-40
0
100
200
300
400
500

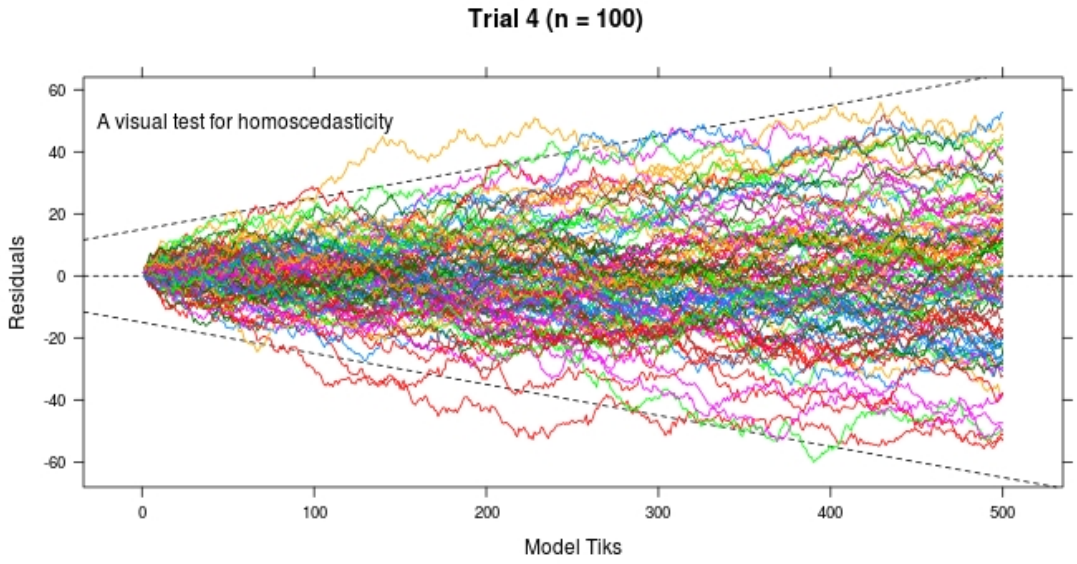

Trial 4 (n = 100)
y1 = expected
y2 = experimental
y1 = .02t
y2 = -1.05 + .013t
Displacement
Model Tiks
60
40
20
0
-20
-40
-60
0
100
200
300
400
500

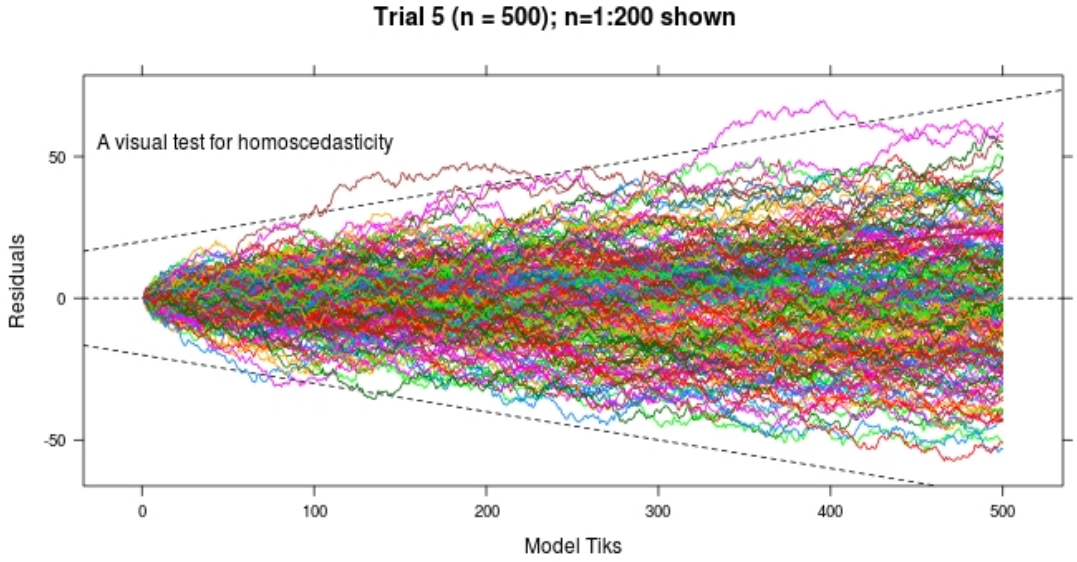

Trial 5 (n = 500); n=1:200 shown
y1 = expected
y2 = experimental
y1 = .02t
y2 = -.14 + .017t
Displacement
50
0
-50
0
100
200
300
400
500
Model Tiks


Trial 1 (n = 1)
A visual test for homoscedasticity
Residuals
10
0
-10
0
100
200
300
400
500
Model Tiks


Trial 2 (n = 5)
A visual test for homoscedasticity
Residuals
40
20
0
-20
0
100
200
300
400
500
Model Tiks


Trial 3 (n = 25)
A visual test for homoscedasticity
Residuals
40
20
0
-20
-40
0
100
200
300
400
500
Model Tiks


Trial 4 (n = 100)
A visual test for homoscedasticity
Residuals
Model Tiks
60
40
20
0
-20
-40
-60
0
100
200
300
400
500


Trial 5 (n = 500); n=1:200 shown
A visual test for homoscedasticity
Residuals
Model Tiks
50
0
-50
0
100
200
300
400
500