Math 458 - Math for Actuaries II - Project # 3

Due: Midnight on Sunday, December 10, 2023.
Objective: Use R, its graphing package `ggplot2`, the concept of a random walk, and the logarithmic model of stock prices to simulate the evolution of stock prices over time.

1. Introduction: The lognormal model is often used to represent the future random behavior of stock prices. We can use this combined with the concept of a random walk to generate possible stock price "paths" over time.

2. Model: This model will reference the logarithmic model of stock prices from Project #4 and the new concept of a random walk. We need two major concepts, one old & one new:

   - Relative Stock Price: $S_t > 0$ represents the stock price at time t. S_{t+h} for $h > 0$ is the stock price at a later time $t + h$. We will use $h = 1/250$, representing the idea that the stock can change at most once each day, and that there are approximately 250 trading days in a year. If you know S_t then S_{t+h}, the next day price, is computed using the model

   $$\underbrace{S_{t+h}}_{\text{new price}} = S_t e^{\left(r-\delta-\sigma^2/2\right)h+\sigma\sqrt{h}Z}$$

   - Random Walk: For the purposes of our work, a random walk is simply a sequence of numbers $\{Z_1, Z_2, \ldots, Z_{250}\}$ created using the R function `rnorm(250,0,1)`, where each is drawn from a standard normal distribution (e.g., the "bell curve").

   If we take $t = jh$ for some j in the list $\{1, 2, \ldots, 249, 250\}$ so that $S_{t+h} = S_{(j+1)h}$, then we can write

   $$\underbrace{S_{(j+1)h}}_{\text{new price}} = S_{t+h} = S_t e^{\left(r-\delta-\sigma^2/2\right)h+\sigma\sqrt{h}} = \underbrace{S_{jh}}_{\text{last price}} e^{\left(r-\delta-\sigma^2/2\right)h+\sigma\sqrt{h}Z_{j+1}}.$$

   Some basic precalculus applied to the above (extra credit!) shows that in general for any $t = jh$, $j = 1, 2 \ldots, 250$, the price $S_t = S_{jh}$ of the stock on day j is given by

   $$S_t = S_{jh} = S_0 e^{\left(r-\delta-\sigma^2/2\right)t+\sigma\sqrt{h}(Z_1+Z_2+\cdots+Z_j)}. \tag{1}$$

   Note that $Z_1 + Z_2 + \cdots Z_j$ is the accumulated value of the random walk.

3. Method:

   A) Open RStudio and create a new script that starts by loading the libraries `ggplot2` and `reshape` which we will use to format format the simulated stock price "trajectories". Note that you may have to install the package `reshape`for the library to be loaded.

B) Write an R script that

i) assigns the numerical values $\delta = 2\%$, $r = 7\%$, $\sigma = 25\%$, and $S_0 = 100$ to similarly named parameters;

ii) creates parameters `timesteps` with assigned value 250 and `simulationnumber` with assigned value 20;

iii) uses the above parameters to create a data frame called `stockdata` with the following properties:

- The first column is named "time" and contains a list of 250 times, starting with 1/250 and then increasing by 1/250 per time until the list ends with 1;
- To the right of the "time" column are twenty different columns, each with column headings `stockdata_`i, where i ranges from 1 to 20;
- Each of these 20 stock data columns contains a *different* list of stock prices that were created using the lognormal / random walk model described above in part 2) above.

NOTE:

- Creating the prices to populate each `stockdata_`i requires the use of `rnorm(timesteps,0,1)` and `cumsum` functions in R, each of which were discussed in class.
- To create the data frame `stockdata`, use a `for` loop with the counter `(i in 1:simulationnumber)` and the `cbind` function to append each list of stock prices to the data frame. Use the loop and the `paste` & `cbind` functions to both create the each of `stockprices_`$i$column names and then to bind them as a vector to the final data frame using the `colnames` command.

iv) Use the `melt` command

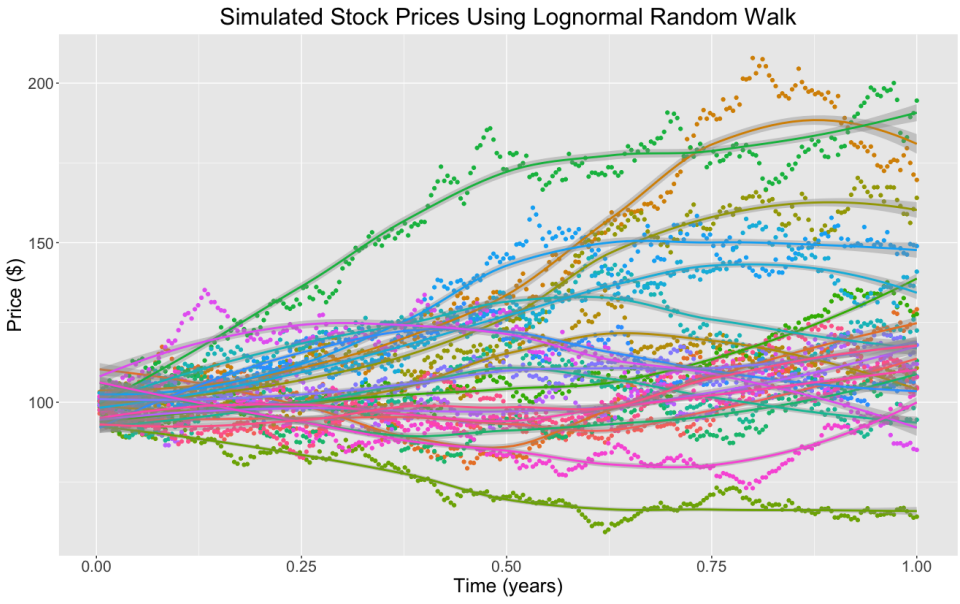
```
stockdatamelted<-melt(stockdata,id.vars="time")
```

to transform the data frame into a form suitable for graphing all of the stock price trajectories together on a single plot.

v) Uses the `ggplot` command

```
ggplot(stockdatamelted,aes(time,value
,col=variable))+ggtitle("Simulated
Stock Prices Using Lognormal Random
Walk")+xlab("Time (years)")+ylab("Price
($)")+geom_point()+theme(legend.position = "none",
text=element_text(size=20),
plot.title=element_text(hjust=0.5))+stat_smooth()
```

to generate an image of the form



REMARK: The `stat_smooth()` command computes a LOESS (LOcally Estimated Scatterplot Smoothing) to computes a localized polynomial regression curve to the data. See Math 311 for more details about this technique!