Gaussian White Noise Return Model for Asset Returns

Eric Zivot

4/12/2021

Gaussian White Noise (GWN) Model

$$r_{it} = \text{ cc return on asset } i \text{ in month } t$$
$$i = 1, \cdots, N \text{ assets; } t = 1, \cdots, T \text{ months}$$

Assumptions (normal distribution and covariance stationarity)

$$\begin{aligned}
r_{it} &\sim iid\ N(\mu_i,\ \sigma_i^2) \text{ for all } i \text{ and } t \\
\mu_i &= E[r_{it}] \ \text{ (constant over time)} \\
\sigma_i^2 &= \mathrm{var}(r_{it}) \ \text{ (constant over time)} \\
\sigma_{ij} &= \mathrm{cov}(r_{it},\ r_{jt}) \ \text{ (constant over time)} \\
\rho_{ij} &= \mathrm{cor}(r_{it},\ r_{jt}) \text{ (constant over time)}
\end{aligned}$$

Regression Model Representation (GWN Model)

Since $r_{it} \sim iid\ N(\mu_i,\ \sigma_i^2)$, we can equivalently express r_{it} as

$$
\begin{aligned}
r_{it} &= \mu_i + \epsilon_{it} \quad t = 1, \cdots, T; \quad i = 1, \cdots N \\
\epsilon_{it} &\sim \text{iid } N(0, \sigma_i^2) \text{ or } \epsilon_{it} \sim GWN(0, \sigma_i^2) \\
\text{cov}(\epsilon_{it},\ \epsilon_{jt}) &= \sigma_{ij},\ \rho_{ij} = \text{cor}(\epsilon_{it},\ \epsilon_{jt}) \\
\text{cov}(\epsilon_{it},\ \epsilon_{js}) &= 0 \quad t \neq s, \text{for all } i,\ j
\end{aligned}
$$

Interpretation

- ϵ_{it} represents random news that arrives in month t
- News affecting asset i may be correlated with news affecting asset j
- News is uncorrelated over time

Interpretation

$$\underset{\substack{\text{unexpected}\\ \text{news}}}{\epsilon_{it}} = \underset{\substack{\text{Actual}\\ \text{return}}}{r_{it}} - \underset{\substack{\text{expected}\\ \text{return}}}{\mu_i}$$

$$\begin{aligned} \text{No news } \epsilon_{it} = 0 &\Longrightarrow r_{it} = \mu_i \\ \text{Good news } \epsilon_{it} > 0 &\Longrightarrow r_{it} > \mu_i \\ \text{Bad news } \epsilon_{it} < 0 &\Longrightarrow r_{it} < \mu_i \end{aligned}$$

GWN Model Regression with Standardized News Shocks

$$
\begin{aligned}
r_{it} &= \mu_i + \epsilon_{it} \quad t = 1, \cdots, T; \quad i = 1, \cdots N \\
&= \mu_i + \sigma_i \times z_{it} \\
z_{it} &\sim \text{iid } N(0,1) \\
\text{cov}(z_{it},\ z_{jt}) &= \text{cor}(z_{it},\ z_{jt}) = \rho_{ij} \\
\text{cov}(z_{it},\ z_{js}) &= 0 \quad t \neq s, \text{ for all } i,\ j
\end{aligned}
$$

Here, $z_{it} \sim \text{iid } N(0,1)$ is a standardized news shock and σ_i is the volatility of news.

Implied Model for Simple Returns

$$
\begin{aligned}
R_{it} &= \exp(r_{it}) - 1 \\
&\Rightarrow 1 + R_{it} \sim \text{lognormal}(\mu_i, \sigma_i^2)
\end{aligned}
$$

where

$$
\begin{aligned}
E[R_{it}] &= \exp\left(\mu_i + \frac{1}{2}\sigma_i^2\right) - 1 \\
\operatorname{var}(R_{it}) &= \exp(2\mu_i + \sigma_i^2)(\exp(\sigma_i^2) - 1)
\end{aligned}
$$

However, if r_{it} is close to zero (typical for daily, weekly, or monthly returns) then it's safe to assume $R_{it} \sim iid\ N(\mu_i,\ \sigma_i^2)$

GWN Model in Matrix Notation

Define the $N \times 1$ vectors $r_t = (r_{1t}, \ldots, r_{Nt})'$, $\mu = (\mu_1, \ldots, \mu_N)'$, $\epsilon_t = (\epsilon_{1t}, \ldots, \epsilon_{Nt})'$ and the $N \times N$ symmetric covariance matrix

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1N} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1N} & \sigma_{2N} & \cdots & \sigma_N^2 \end{pmatrix}.$$

Then the GWN model matrix notation is

$$\begin{aligned} \mathbf{r}_t &= \mu + \epsilon_t, \\ \epsilon_t &\sim GWN(\mathbf{0}, \Sigma), \end{aligned}$$

which implies that $r_t \sim iid\ N(\mu, \Sigma)$.

Monte Carlo Simulation

Idea: Use computer random number generator to create simulated values from assumed model

- Reality check on proposed model
- Create *what if?* scenarios
- Study properties of statistics computed from proposed model

Simulating Random Numbers from a Distribution

Goal: simulate random number x from pdf $f(x)$ with CDF $F_X(x)$

Method: Inverse CDF technique

- Generate $U \sim$ Uniform $[0, 1]$
- Generate $X \sim F_X(x)$ using inverse CDF technique:

$$
\begin{aligned}
x &= F_X^{-1}(u) \\
F_X^{-1} &= \text{inverse CDF function (quantile function)} \\
F_X^{-1}\left(F_X(x)\right) &= x
\end{aligned}
$$

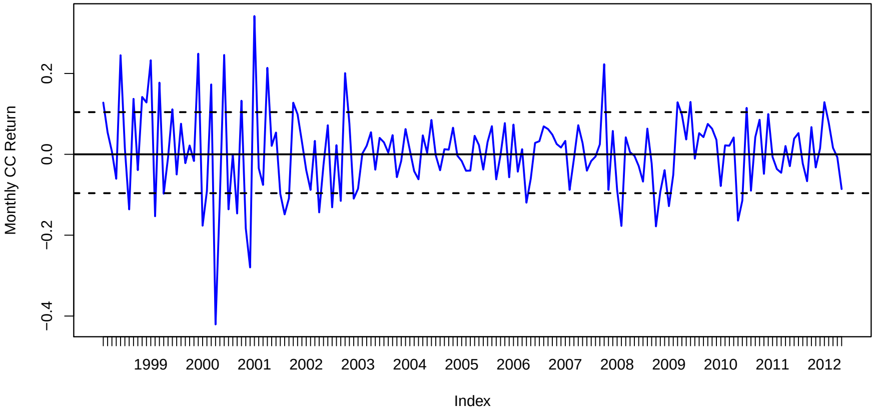
Example: Monthly MSFT CC Returns

Calibrate GWN model to observed $T = 172$ cc returns on MSFT from Jan 1998 through May 2012

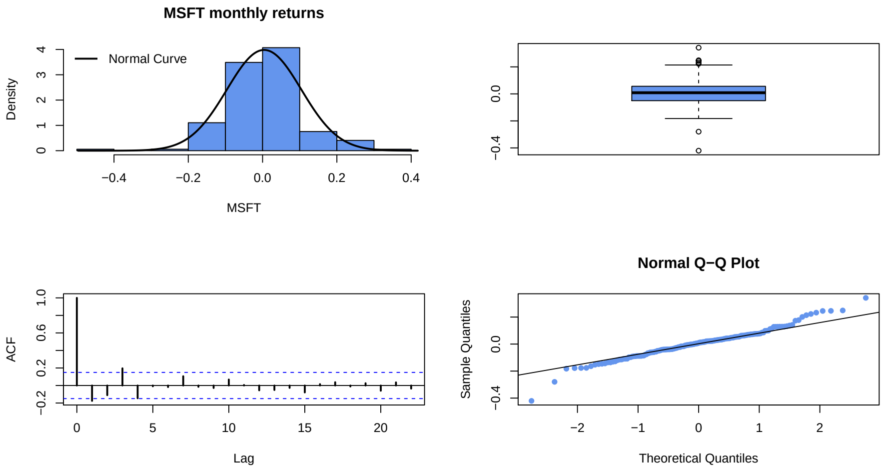
```
data(msftDailyPrices)
msftPrices = to.monthly(msftDailyPrices, OHLC=FALSE)
smpl = "1998-01::2012-05"
msftPrices = msftPrices[smpl]
msftRetS = Return.calculate(msftPrices, method="simple")
msftRetS = msftRetS[-1]
msftRetC = log(1 + msftRetS)
```

Mean is 0.004 and SD is 0.1

Example: Monthly MSFT CC Returns



Example: Monthly MSFT CC Returns



Example: Monte Carlo Simulation

- Specify parameters based on sample statistics

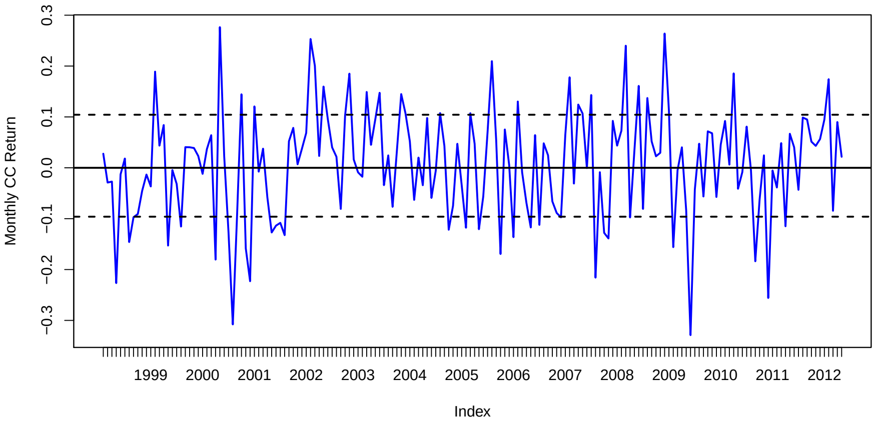
$$
\begin{aligned}
\mu_i &= 0.004 \text{ (sample mean return)} \\
\sigma_i &= 0.10 \text{ (sample SD/volatility)} \\
r_{it} &= 0.004 + \epsilon_{it}, \ t = 1, \ldots, 172 \\
\epsilon_{it} &\sim \text{iid } N(0, (0.10)^2)
\end{aligned}
$$

- Simulation requires generating random numbers from a normal distribution. In R use `rnorm()`.

Example: Monte Carlo Simulation

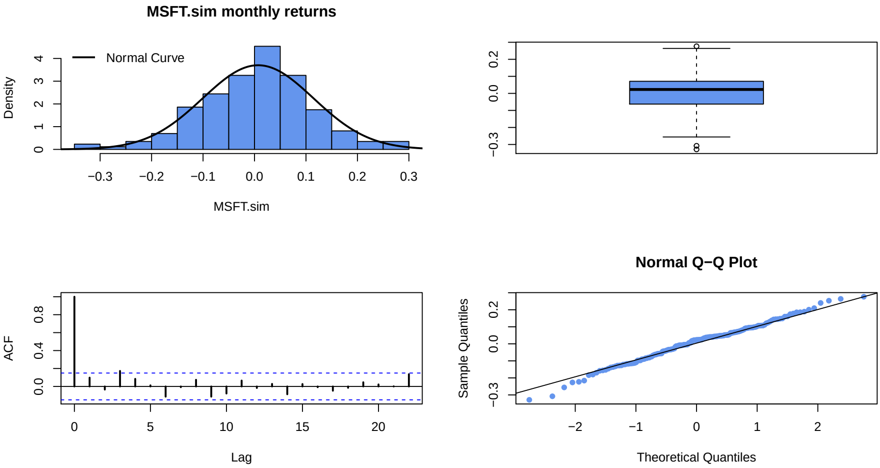
```
mu = mean(msftRetC)
sd.e = sd(msftRetC)
n.obs = length(msftRetC)
set.seed(111)
sim.e = rnorm(n.obs, mean=0, sd=sd.e)
sim.ret = mu + sim.e
sim.ret = xts(sim.ret, index(msftRetC))
colnames(sim.ret) = "MSFT.sim"
```

Example: Simulated MSFT CC returns



Simulated cc returns look like MSFT cc returns except they have constant volatility

Example: Simulated MSFT CC returns



Simulated cc returns share similar stylized facts as MSFT cc returns

Multiple Simulations

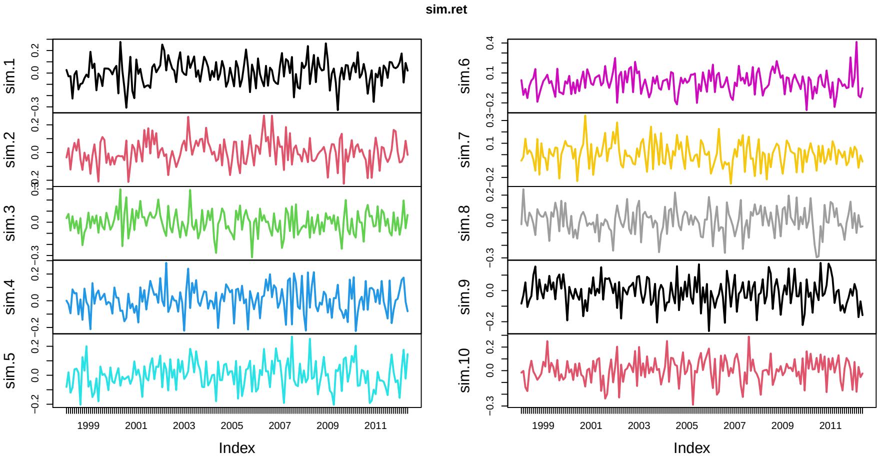
- The power of Monte Carlo Simulation comes when you create many simulated samples from your model
  - View many *alternative realities* of your model
  - Compute approximate probabilities from simulated samples
- Can be used to analyze statistical properties of sample estimates

Example: Multiple Simulations

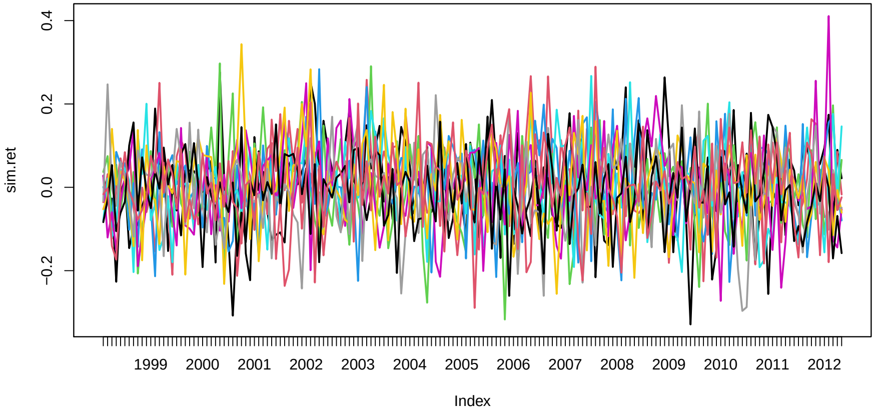
Create 10 simulated samples from the GWN model for Microsoft returns

```
sim.e = matrix(0, n.obs, 10)
set.seed(111)
for (i in 1:10) {
  sim.e[,i] = rnorm(n.obs, mean=0, sd=sd.e)
}
sim.ret = mu + sim.e
sim.ret = xts(sim.ret, index(msftRetC))
colnames(sim.ret) = paste("sim", 1:10, sep=".")
```

Example: Multiple Simulations

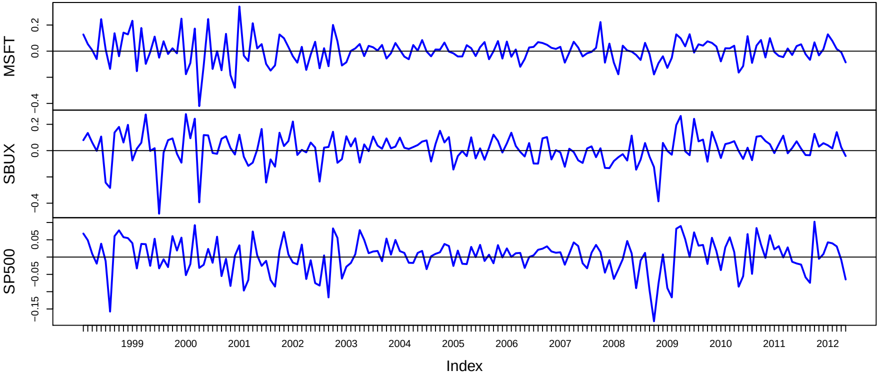


Example: Multiple Simulations

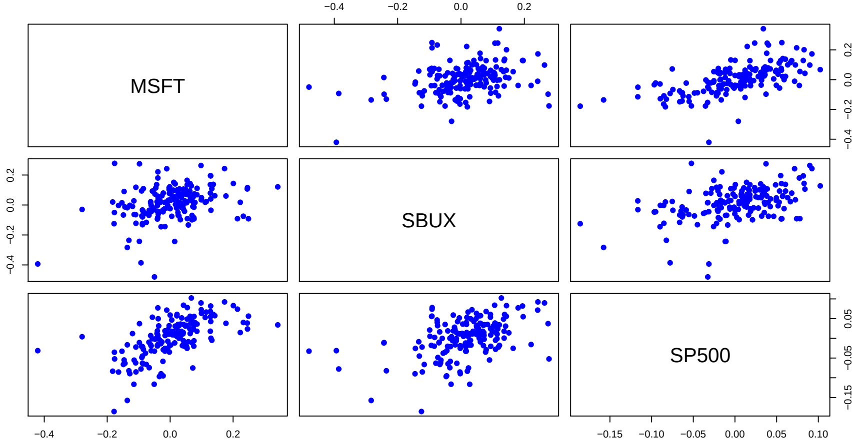


Example: Multivariate Simulation

Calibrate GWN model to observed $T = 172$ cc returns on MSFT, SBUX and SP500 from Jan 1998 through May 2012



Example: Multivariate Simulation



Assets are positively correlated

Example: Multivariate Simulation

Sample statistics to calibrate multivariate GWN model.

Sample mean vector:

```
##    MSFT    SBUX   SP500
## 0.00413 0.01466 0.00169
```

Sample covariance matrix

```
##          MSFT    SBUX   SP500
## MSFT  0.01004 0.00381 0.00300
## SBUX  0.00381 0.01246 0.00248
## SP500 0.00300 0.00248 0.00235
```

Example: Multivariate Simulation

- Specify parameters based on sample statistics

$$\mu = \begin{pmatrix} .004 \\ .015 \\ .002 \end{pmatrix}, \Sigma = \begin{pmatrix} .010 & .004 & .003 \\ .004 & .012 & .002 \\ .003 & .002 & .002 \end{pmatrix}$$

$$\mathbf{r}_t = \mu + \epsilon_t, \ \ t = 1, \ldots, 172$$

$$\epsilon_t \sim \text{iid } N(\mathbf{0}, \Sigma)$$

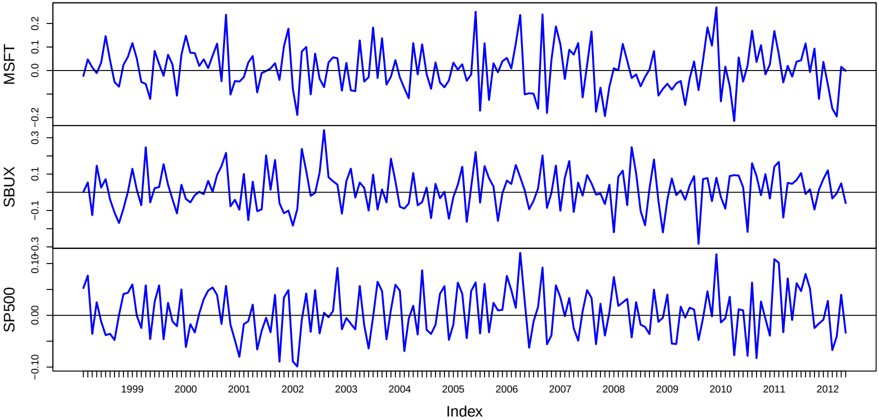
- Simulation requires generating random numbers from a multivariate normal distribution.
- R package **mvtnorm** has function `mvnorm()` for simulating data from a multivariate normal distribution.

Example: Multivariate Simulation

Simulate GWN model returns calibrated to sample statistics using **mvtnorm** function `rmvnorm()`

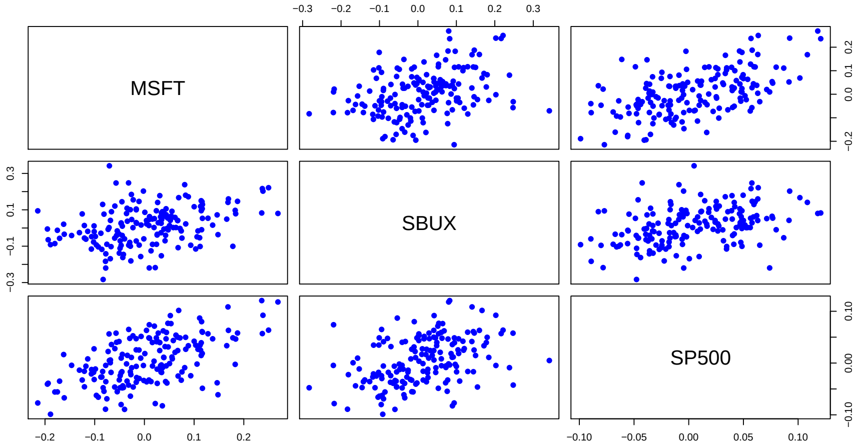
```
muVec = apply(gwnRetC, 2, mean)
covMat = cov(gwnRetC)
set.seed(123)
returns.sim = rmvnorm(n.obs, mean=muVec, sigma=covMat)
colnames(returns.sim) = colnames(gwnRetC)
returns.sim = xts(returns.sim, index(gwnRetC))
```

Example: Simulated Returns



Simulated returns look like actual returns except they have constant volatility

Example: Simulated Returns



Pairwise correlations in simulated returns match actual returns

GWN Model and Multi-period cc Returns

$$
\begin{aligned}
r_t &= \mu + \epsilon_t, \;\; \epsilon_t \sim GWN(0, \sigma^2) \\
r_t(k) &= r_t + r_{t-1} + \cdots + r_{t-k+1} = \sum_{j=0}^{k-1} r_{t-j} \\
&= (\mu + \epsilon_t) + (\mu + \epsilon_{t-1}) + \cdots + (\mu + \epsilon_{t-k+1}) \\
&= k\mu + \sum_{j=0}^{k-1} \epsilon_{t-j} \\
&= \mu(k) + \epsilon_t(k)
\end{aligned}
$$

GWN Model and Multi-period cc Returns

Here,

$$\mu(k) = k\mu$$
$$\epsilon_t(k) = \sum_{j=0}^{k-1} \epsilon_{t-j} \sim GWN\left(0, k\sigma^2\right)$$

GWN Model and Multi-period cc Returns

Result: In the GWN model

$$
\begin{aligned}
E[r_t(k)] &= \mu(k) = k\mu \\
\mathrm{var}\left(r_t(k)\right) &= \sigma^2(k) = k\sigma^2 \\
\mathrm{SD}\left(r_t(k)\right) &= \sigma_k(k) = \sqrt{k}\sigma
\end{aligned}
$$

and

$$
\epsilon_t(k) = \sum_{j=0}^{k-1} \epsilon_{t-j} = \text{accumulated news shocks}
$$

The Random Walk Model

The GWN model for cc returns is equivalent to the random walk (RW) model for log stock prices

$$
\begin{aligned}
r_t &= \ln\left(\frac{P_t}{P_{t-1}}\right) = \ln P_t - \ln P_{t-1} \\
&= \ln P_t - \ln P_{t-1}
\end{aligned}
$$

which implies

$$\ln P_t = \ln P_{t-1} + r_t$$

The Random Walk Model

Recursive substitution starting at $t = 1$ gives

$$\begin{aligned}
\ln P_1 &= \ln P_0 + r_1 \\
\ln P_2 &= \ln P_1 + r_2 \\
&= \ln P_0 + r_1 + r_2 \\
&\vdots \\
\ln P_t &= \ln P_{t-1} + r_t \\
&= \ln P_0 + \sum_{s=1}^{t} r_s
\end{aligned}$$

Interpretation: Price at t equals initial price plus accumulation of cc returns

The Random Walk Model

In GWN model, $r_s = \mu + \epsilon_s$ so that

$$\begin{aligned} \ln P_t &= \ln P_0 + \sum_{s=1}^{t} r_s \\ &= \ln P_0 + \sum_{s=1}^{t} (\mu + \epsilon_s) \\ &= \ln P_0 + t \cdot \mu + \sum_{s=1}^{t} \epsilon_s \end{aligned}$$

Interpretation: Log price at t equals initial price $\ln P_0$, plus expected growth in prices $E[\ln P_t] = t \cdot \mu$, plus accumulation of news $\sum_{s=1}^{t} \epsilon_s$.

The Random Walk Model

The price level at time t is

$$P_t = P_0 \exp\left(t \cdot \mu + \sum_{s=1}^{t} \epsilon_s\right) = P_0 \exp\left(t \cdot \mu\right) \exp\left(\sum_{s=1}^{t} \epsilon_s\right)$$

$$\exp\left(t \cdot \mu\right) = \text{expected growth in price}$$

$$\exp\left(\sum_{s=1}^{t} \epsilon_s\right) = \text{unexpected growth in price}$$

GWN Model for Simple Returns

- GWN Model can also be used for simple returns

$$R_t = \mu + \epsilon_t$$
$$\epsilon_t \sim \mathrm{GWN}(0, \sigma^2)$$

- The justification is that when R_t is close to zero, $R_t \approx r_t$ so that the model for r_t can be used as the model for R_t
- Main drawbacks:
  - Normal distribution allows $R_t < -1$;
  - Multi-period simple returns are not normally distributed

GWN Model for Simple Returns

For multi-period simple returns

$$\begin{aligned} R_t(k) &= (1+R_t)(1+R_{t-1})\cdots(1+R_{t-k+1}) - 1 \\ &\nsim N(k\mu, k\sigma^2) \end{aligned}$$

However, it can be shown that

$$\begin{aligned} E[R_t(k)] &= (1+\mu)^k - 1 \\ \mathrm{var}(R_t(k)) &= (1+\sigma^2+2\mu+\mu^2)^k - (1+\mu)^{2k} \end{aligned}$$

GWN Model for Simple Returns

- If $\mu \approx 0$ then $(1+\mu)^k - 1 \approx k \times \mu$,
- If $\mu \approx 0$ and σ^2 is not too large,

$$(1 + \sigma^2 + 2\mu + \mu^2)^k - (1+\mu)^{2k} \approx k \times \sigma^2$$