

50 Shades of Stable

Derl Clausen, Bruno Komel, Rakeen Tanvir

5/11/2020

Outline

1. Exploratory Data Analysis
2. The Impact of “Rare” Events
3. Linear and Logistic Regression
4. Fourier Analysis
5. Random Walk
6. Normal Distribution
7. Fractal Analysis
8. Cauchy Distribution
9. Bootstrap
10. Partial Variance
11. Divergent Integration
12. Stable Distribution
13. KS Test / Chi-Squared Test (Chi-square Test on Varying Time Scales) / Complexities of Goodness of Fit Test
14. The Impact of Political Regimes - Hypothesis Testing With Permutation Test
15. Hypothesis Testing: Contingency table with chi-square test

Exploratory Data Analysis

The first differences of DJI Open price values present an interesting challenge. The histogram resembles a symmetric distribution with a sharp peak around the mean, heavy tails, and a left skew. The normal distribution does not seem to be a good fit at first glance, but a heavy-tailed distribution could help us model our data

```
#Import libraries
library('pracma')
library('fitdistrplus')

## Loading required package: MASS
## Warning: package 'MASS' was built under R version 3.6.2
```

```

## Loading required package: survival
## Loading required package: npsurv
## Loading required package: lsei
library('MASS')
library('ggplot2')
library('gganimate')
library('gifski')
library('fractaldim')

## Loading required package: abind
library('fBasics')

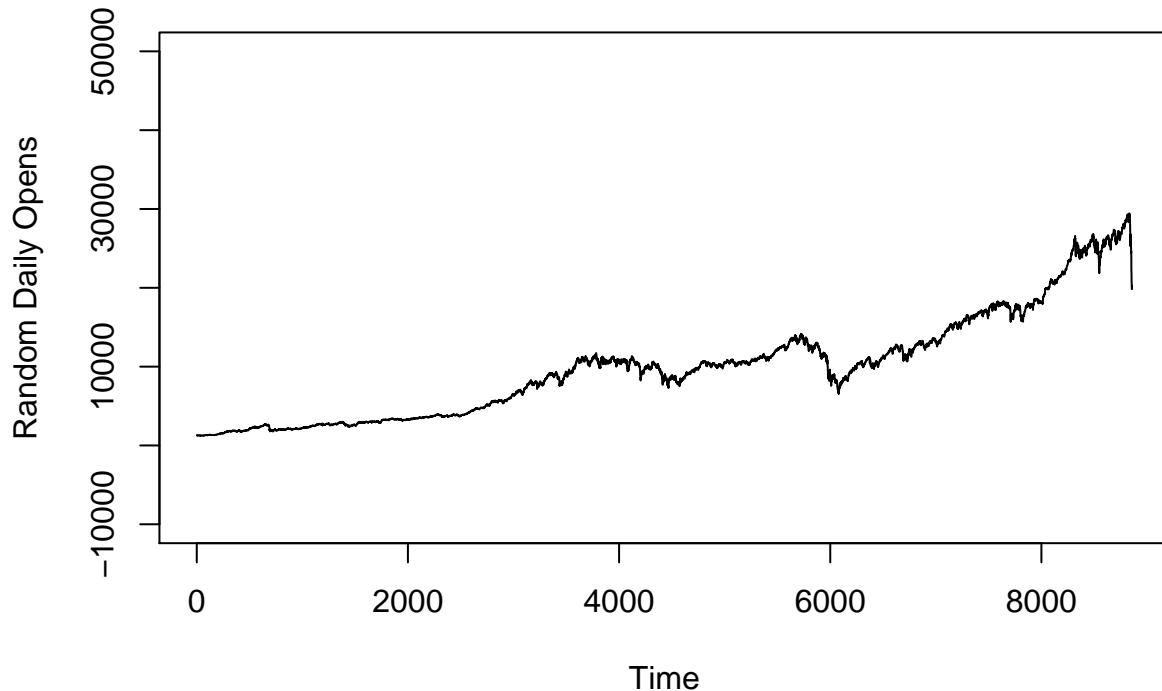
## Loading required package: timeDate
## Loading required package: timeSeries
##
## Attaching package: 'fBasics'
## The following objects are masked from 'package:pracma':
##       akimaInterp, inv, kron, pascal
library('stabledist')
library('car')

## Loading required package: carData
##
## Attaching package: 'car'
## The following object is masked from 'package:fBasics':
##       densityPlot
## The following object is masked from 'package:pracma':
##       logit
source('prj_DataPreparation.R')
source('prj_Functions.R')

#This is a graphical representation of the Dow Jones Index from 1985 to 2020
plot(DJI$Open, type = "l", xlab = "Time",
      ylab = "Random Daily Opens", main = "Random Walk Model",
      ylim = c(-10000,50000))

```

Random Walk Model



```
#Get the first difference of the time series data
diffs <- diff(DJI$Open)
```

The Impact of “Rare” Events

Let's begin by taking a look at presumably rare events, and how they're represented in our data. Let's consider each of the days where was a statistically significant price change assuming these price changes followed a normal distribution.

```
mu <- mean(diffs); mu
## [1] 2.142422
sigma <- sd(diffs); sigma
## [1] 117.3051
N <- length(diffs)
SDs <- numeric(N)
for (i in 1:N){
  SDs[i] <- (diffs[i]-mu)/sigma
}
head(SDs);head(diffs)

## [1]  0.14924841 -0.13871881 -0.07197018 -0.05969396  0.16911181 -0.09643625
## [1]  19.650024 -14.130005  -6.300049   -4.859985  21.980103  -9.170044
length(SDs)

## [1] 8857
```

```

SDs.data <- c(0,SDs[1:length(SDs)]); head(SDs.data)

## [1] 0.00000000 0.14924841 -0.13871881 -0.07197018 -0.05969396 0.16911181
DJI <- data.frame(DJI,SDs.data)
idx <- which(abs(SDs.data) > 5); head(idx)

## [1] 3846 4200 5971 5979 5981 5983
unusual <- DJI[idx,]; head(unusual)

##           Date      Open      High      Low     Close Adj.Close    Volume Regime
## 3846 2000-04-17 10303.29 10583.75 10232.55 10582.51 10582.51 247520000     BC
## 4200 2001-09-18  8922.70  9022.06  8861.05  8903.40  8903.40 372230000    GWB
## 5971 2008-09-30 10371.58 10868.90 10371.42 10850.66 10850.66 319770000    GWB
## 5979 2008-10-10  8568.67  8901.28  7882.51  8451.19  8451.19 674920000    GWB
## 5981 2008-10-14  9388.97  9794.37  9085.43  9310.99  9310.99 412740000    GWB
## 5983 2008-10-16  8577.04  9013.27  8197.67  8979.26  8979.26 422450000    GWB
##          Republican Recession      diffs   SDs.data
## 3846        FALSE       FALSE -619.5596 -5.299871
## 4200        TRUE        TRUE -657.6201 -5.624329
## 5971        TRUE        TRUE -768.0400 -6.565634
## 5979        TRUE        TRUE -693.0205 -5.926109
## 5981        TRUE        TRUE  926.5498  7.880367
## 5983        TRUE        TRUE -724.8701 -6.197620

```

As we can see, there are 32 days in which the price flux for the Dow Jones was larger than 5 standard deviations away from the mean. To show just how bad of a fit the assume of normal distribution is for our data consider the p-values for each of these events.

```

N <- nrow(unusual)
pvals <- numeric(N)
for (i in 1:(N)) {
  pvals[i] <- pnorm(abs(unusual$SDs.data[i]*sigma), mean = mu, sd = sigma, lower.tail = FALSE)
}
head(pvals)

## [1] 6.402775e-08 1.034892e-08 2.927955e-11 1.733056e-09 1.888683e-15
## [6] 3.218177e-10
rare <- max(pvals); rare #2.303366e-07, which is pretty much 0.

## [1] 2.306794e-07
(1/rare)/365

## [1] 11876.77

```

If we interpret the p-value as the probability of an event taking place, and our events are measured in days, then a given p-value tells us the probability of seeing that extreme of an event on any given days (i.e. a p-value of 0.05 would correspond to an event that we'd "expect" to see once every 20 days, since it has a 1/20 chance of arising). In other words, if the DJI first differences followed a normal distribution, we would expect to see the least rare of these rare events once in 11,894.45 years, but from the data it is clear that these events are far more common than that.

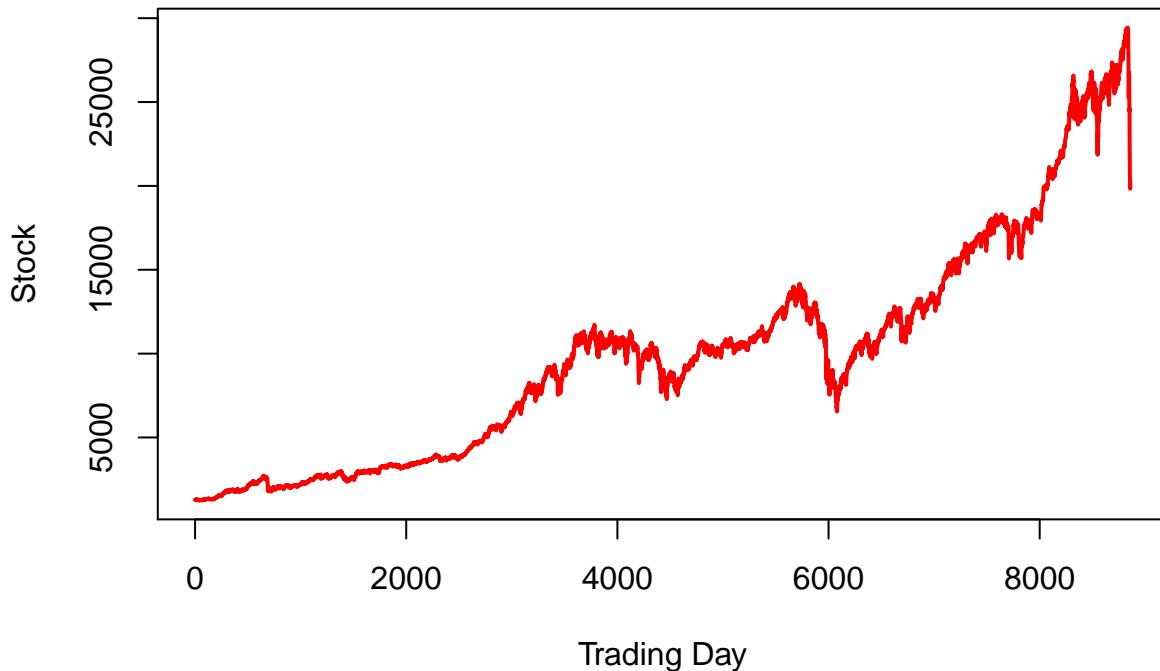
This raises two questions, does the stock market have properties of random walk? And, if so, what is the underlying distribution of the step size of the random walk?

Fourier Analysis

Before analyzing our data for properties of a random walk and infinite variance, we first need to run a Fourier Analysis to ensure there is no cyclic patterns in our data:

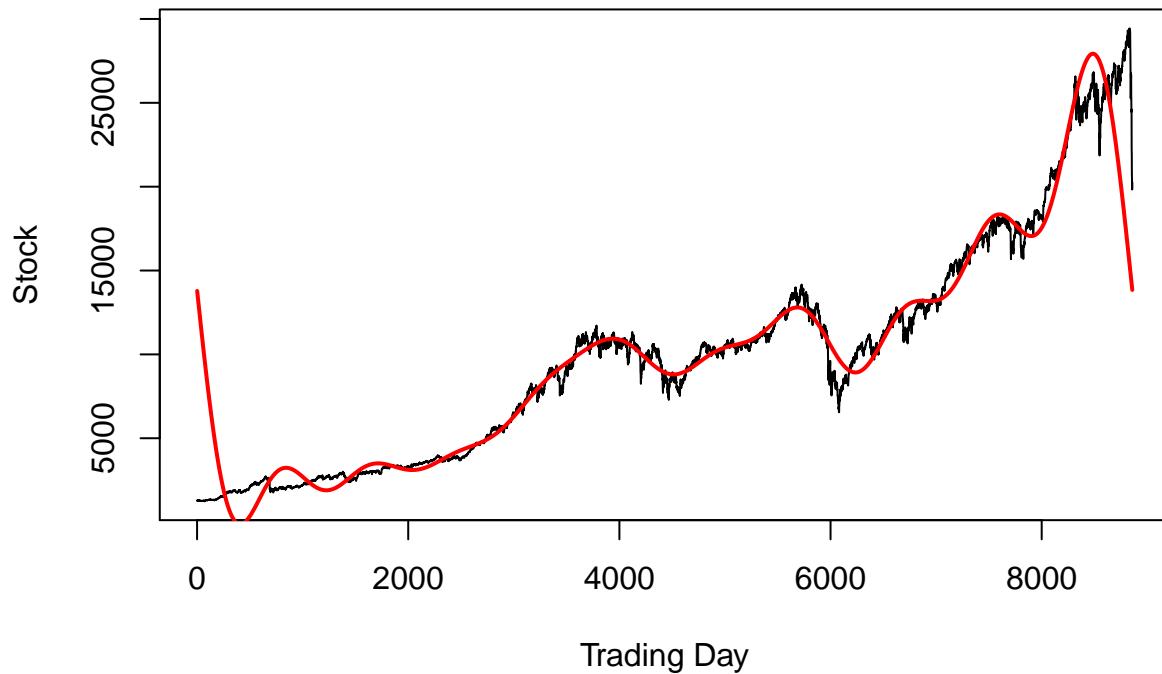
```
#We will begin analysing the Open Stock Price Data before shifting into it's first difference (or price
#Using our RunFourier() function defined in Main.R, we check that we can fully reconstruct our data from
RunFourier(length(DJI$Open)/2, DJI$Open, FALSE) #Perfect Reconstruction
```

Stock Price over Trading Days



```
RunFourier(10, DJI$Open, FALSE) #Using only 10 basis vectors
```

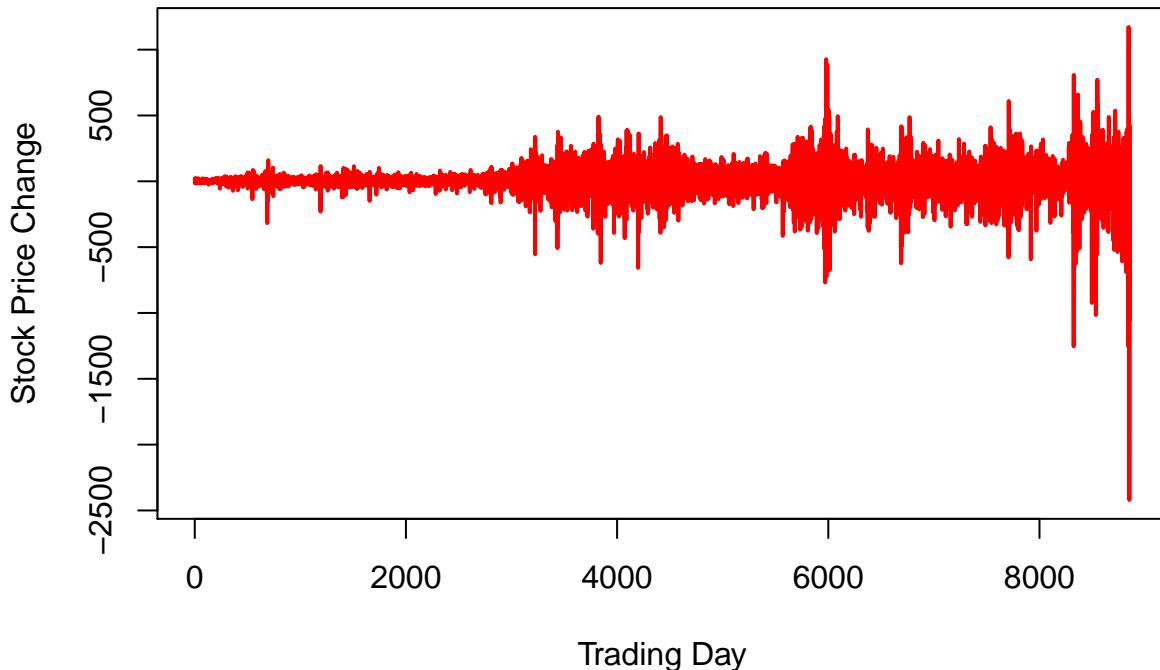
Stock Price over Trading Days



```
#We capture the general shifts of the market using very few basis vectors in our analysis.  
#As you can see, there is no discernable cyclical pattern.
```

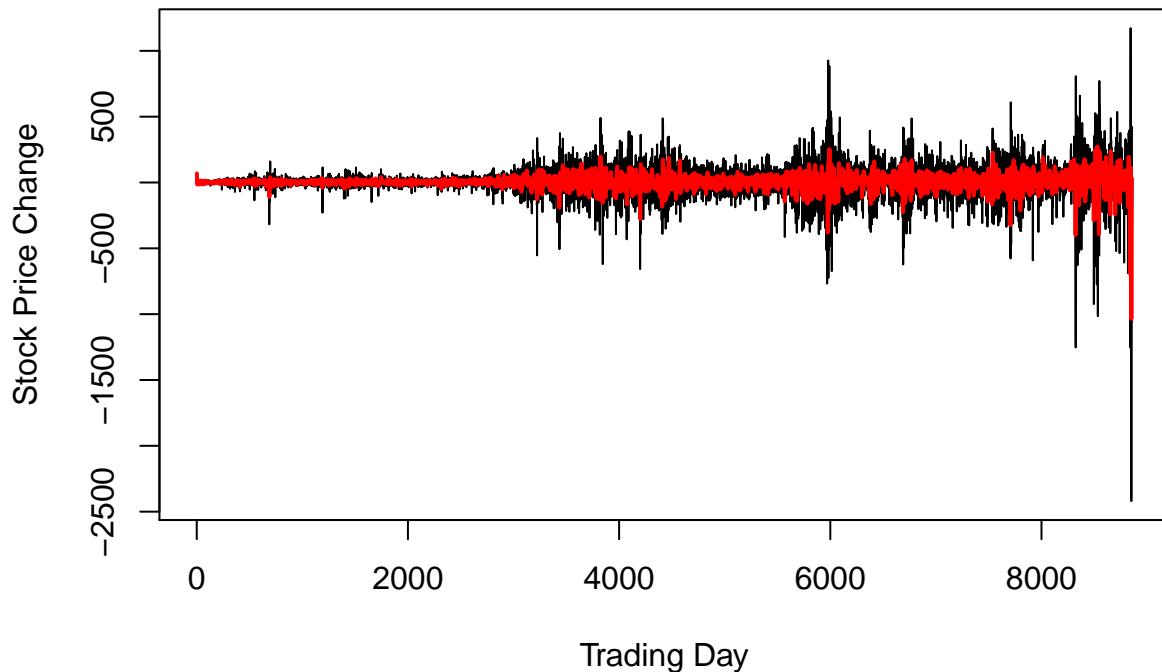
```
#Next run Fourier Analysis on the stock price change over trading days. Again, we look for a perfect re  
RunFourier(length(diffs) /2, diffs, TRUE) #Excellent
```

Stock Price Change over Trading Days



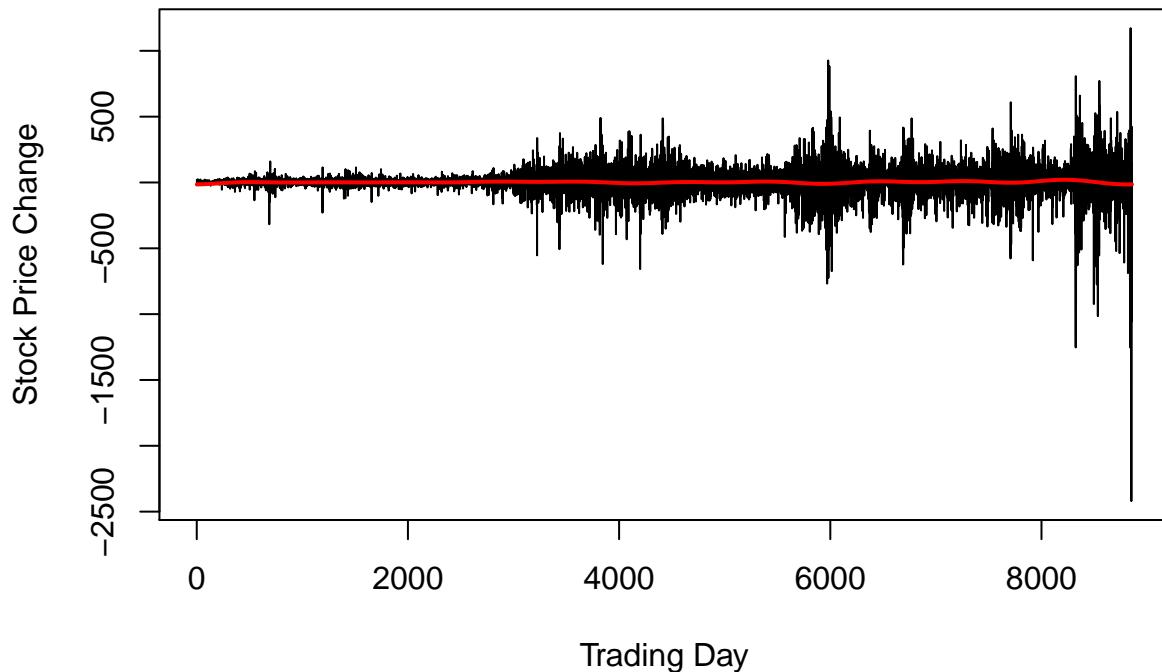
```
#Next we test it using 1000 basis vectors  
RunFourier(1000, diffs, TRUE) #No discernable pattern
```

Stock Price Change over Trading Days



```
#Check 10 basis vectors, again to see if there is any kind of pattern:  
RunFourier(10, diffs, TRUE)
```

Stock Price Change over Trading Days



```
#Using a small number of basis vectors yields no new information. Although we can
#perfectly reconstruct the data, the more basis vectors added to our analysis mostly just
#pick up on noise. Again, there is no noticeable trend or cycle which indicate we can
#proceed to examine random walks.
```

Random Walk with Infinite Variance Demonstration (Introduction)

As we will show, a simple plot of our data resembles a random walk. But if it is a random walk, what kind of random walk is it? This is a good moment to introduce Benoit Mandelbrot's research from 1963. While analyzing stock prices, he associated two to three different probability laws that describe the length of the step of a random walk: Gaussian, Levy Flight and the less popular Cauchy Flight. These probability laws are associated with the underlying distribution of the price changes over time. For instance, if a set of stock prices are a Gaussian random walk, their price change over time (first difference) follows a normal distribution, Levy flight follows a (Levy-Pareto) stable distribution, and Cauchy Flight follows a Cauchy distribution. Using our stock price data, we demonstrate the relationship between random walks, and the distribution of their first difference:

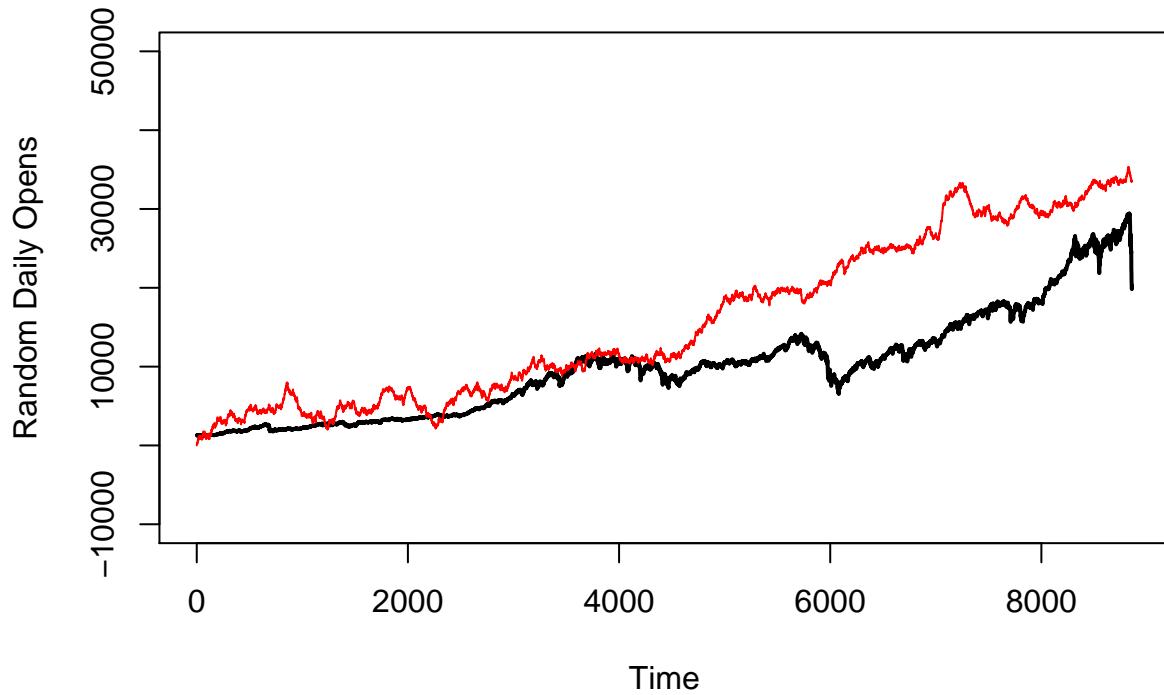
```
## We begin our test by assuming our data is a Gaussian random walk model created using a (0,1,0) Arima
#Because we are assuming our data is a Gaussian random walk, we assume our data has finite mean and var
mu.chg.open <- mean(diffs); mu.chg.open # 2.142422

## [1] 2.142422
sd.diff <- sqrt(mean(diffs^2) - mean(diffs)); sd.diff #117.3089

## [1] 117.3089

#Next, we plot our original data (black), with our random walk model (red):
plot(DJI$Open, type = "l", xlab = "Time",
      ylab = "Random Daily Opens", main = "DJI Stock Price Data with 1 Simulation of Random Walk Model",
      ylim = c(-10000,50000), lwd =2)
rw.drift <- arima.sim(model = list(order = c(0,1,0)),
                      length(DJI$Open), mean = mu.chg.open,
                      sd = sd.diff)
lines(rw.drift, col = "red")
```

DJI Stock Price Data with 1 Simulation of Random Walk Model

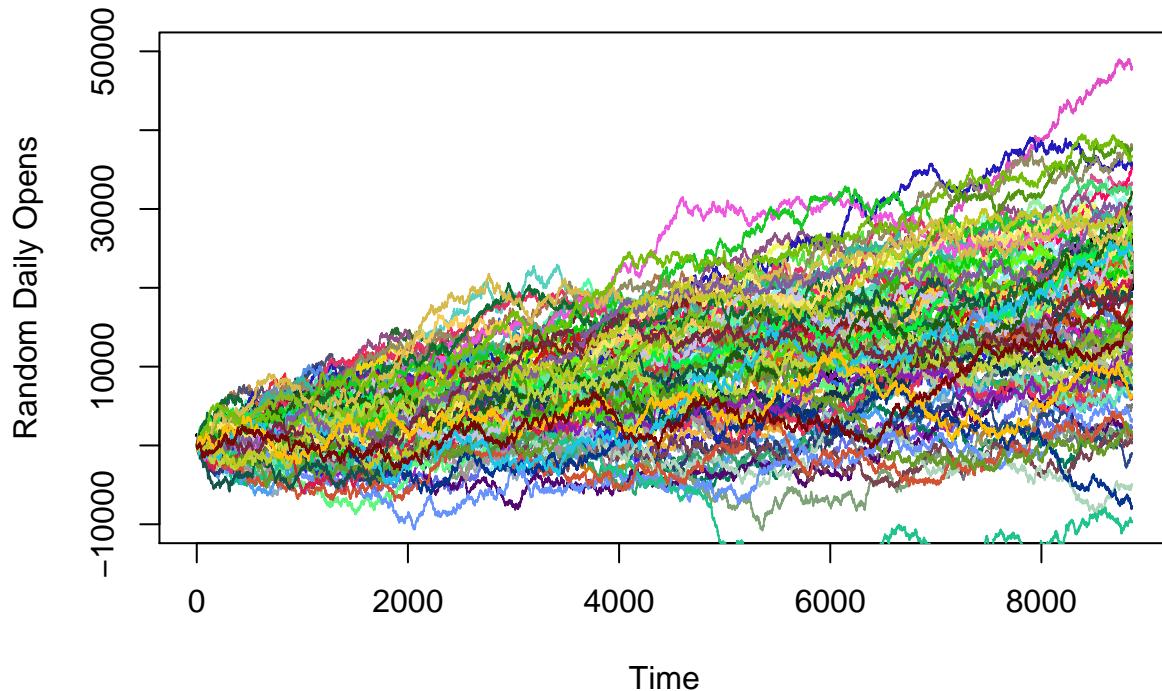


To show the full impact of the stock market as a random walk, we run a simulation of our model 100 times:

```
#Plot our original data
plot(DJI$Open, type = "l", xlab = "Time",
      ylab = "Random Daily Opens", main = "100 Simulations of Random Walk Model",
      ylim = c(-10000,50000), lwd =2)

#Plot of our 100 simulations:
for (i in 1:100) {
  rw.drift <- arima.sim(model = list(order = c(0,1,0)),
                        length(DJI$Open), mean = mu.chg.open,
                        sd = sd.diff)
  lines(rw.drift, col = rgb(runif(1,0,1),runif(1,0,1),runif(1,0,1)))
}
```

100 Simulations of Random Walk Model



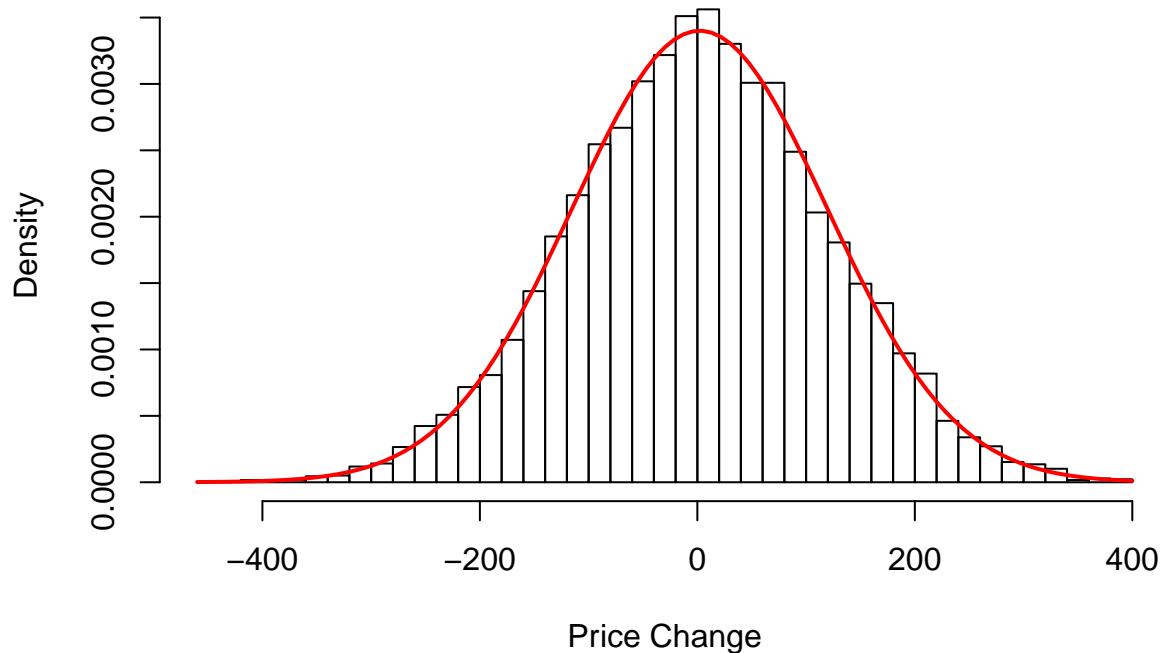
Normal Distribution

Under a gaussian random walk assumption with finite mean and variance, our model takes on a significant range of values across the 100 simulations. We even experience some fairly significant volatility in certain outlier simulations that could potentially represent a sharp drop in stock prices, perhaps, from a recession. Therefore, it is tempting to stop here and assume our data is a Gaussian random walk. However, our model is based on finite variance and mean calculated from a sample rather than the population. Therefore, to fully understand what probability laws our data is following, we need to examine the first difference histograms of our data and our model. Then, we will compare the distributions.

```
#Plot the distribution of our model and overlay a normal density curve:
```

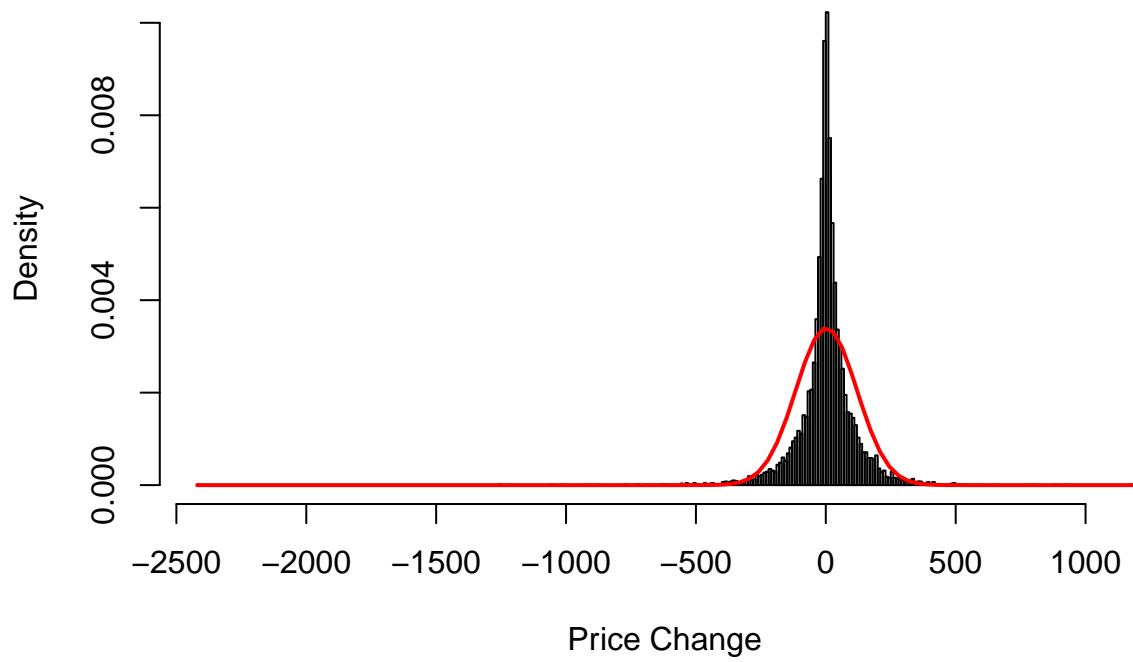
```
hist(diff(rw.drift), breaks = "FD", freq = FALSE, main = "Histogram of Random Walk Model First Difference")
curve(dnorm(x, mean(diffs), sd(diffs)), add = TRUE, lwd = 2, col = "red")
```

Histogram of Random Walk Model First Difference

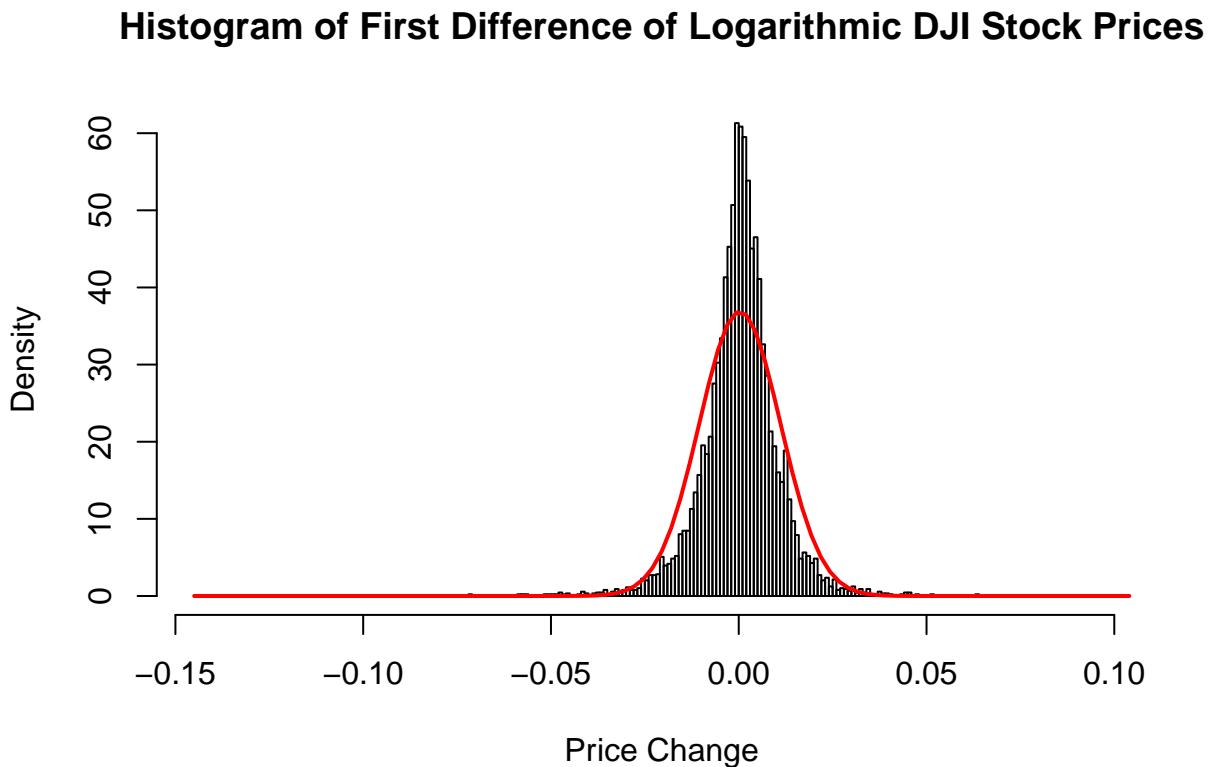


```
#Plot the distribution of our first differences and overlay a normal density curve:  
hist(diff, breaks = "FD", freq = FALSE, main = "Histogram of DJI First Difference", xlab = "Price Change",  
curve(dnorm(x, mean(diff), sd(diff)), add = TRUE, lwd = 2, col = "red")
```

Histogram of DJI First Difference



```
#Plot the distribution of the log of our first differences and overlay a normal density curve:
hist(diff(log(DJI$Open)), breaks = "FD", freq = FALSE, main = "Histogram of First Difference of Logarithmic DJI Stock Prices", xlab = "Price Change", ylab = "Density")
curve(dnorm(x, mean(diff(log(DJI$Open))), sd(diff(log(DJI$Open)))), add = TRUE, lwd = 2, col = "red")
```



Off of inspection alone, we can see that the normal distribution is an excellent fit for the first difference of our gaussian random walk model. However, the normal curve fails to capture the high peak and heavy tails of our first difference data. To add a layer of robustness, we attempted to model the first difference of the logarithm of our data. As you can see, it also fails to capture both the high peak and heavy tails of the distribution. Later, we will run three different goodness of fit tests when we explore some of the complexities of modeling our data with various distributions. For now, however, we can continue our investigation of stock prices as random walks by introducing fractal analysis and a new way of thinking about randomness.

Fractal Analysis

The brilliance of Mandelbrot was not merely his grasp of the technical mathematics of various probability laws and their application. He also popularized a new, more generalized, way to think about randomness. Rather than randomness being a binary, the idea is simply that there are ‘degrees’ of randomness. This was explored in detail through his work with fractal geometry, especially as it pertains to the famous Mandelbrot Sets. Through fractal geometry, we can think of randomness as a measure of self-similarity or lack thereof. More specifically, for our data we will utilize fractal analysis to calculate the hurst exponent as well as the fractal dimension since $H = 2 - D$ where H is the hurst exponent and D is the fractal dimension. (Note: hurst exponents are calculated on stationary time series such as our first differences whereas fractal dimension is computed on our original stock price data. This will become clearer below.) If the hurst exponent is near or equal to .5, then we are experiencing ‘wild randomness’ in our data. If the hurst exponent is significantly above or below .5, then the data is experiencing a lesser degree of randomness. We demonstrate this here:

```
#We will be using the Rescaled Range (R/S) Method to find the Hurst Exponent:
```

```
#We begin by setting the maximum number of divisions that will occur as we divide our data set in 1/2^n
```

```
N <- floor(log(length(diffs), base = 2)) - 1
```

```

#Create a table to store the size of each division (2^0, 2^1, 2^2, ..., 2^n)
n <- numeric(N)

#Create table for our R/S result from each
result.hexp <- numeric(N)

#Loop through each division
for(i in 1:N){
  #Set division size
  n[i] <- floor(length(diffs)/2^(i-1))

  #''Chunk'' our data into 2^n chunks
  ch <- split(diffs, cut_number(1:length(diffs), 2^(i - 1)), drop = TRUE)

  #Create table for R/S analysis average from each chunk for each division
  rs_avgs <- numeric(length(ch))

  #Loop through each chunk
  for (k in 1:length(ch)){

    #Set X = to the chunk
    X <- ch[[k]]

    #Get the mean of the chunk
    m <- mean(X)

    #Mean Adjusted Series:
    Y <- X - m; Y

    #Table for Cumulative Deviate Series:
    Z <- numeric(length(Y))

    #Calculate Cumulative Deviate Series
    for (j in 1:length(Z)){
      Z[j] <- sum(Y[1:j])
    }

    #Calculate range
    r <- max(Z) - min(Z); r

    #Calculate standard deviation of the the mean adjusted series:
    s <- sqrt(mean(Y^2))

    #Store R/S Analysis average
    rs_avgs[k] <- (r/s)
  }

  #Store the mean of the R/S analysis Average for each chunk
  result.hexp[i] <- mean(rs_avgs)
}

#Plot the log of the R/S analysis across the log of the size of the division

```

```

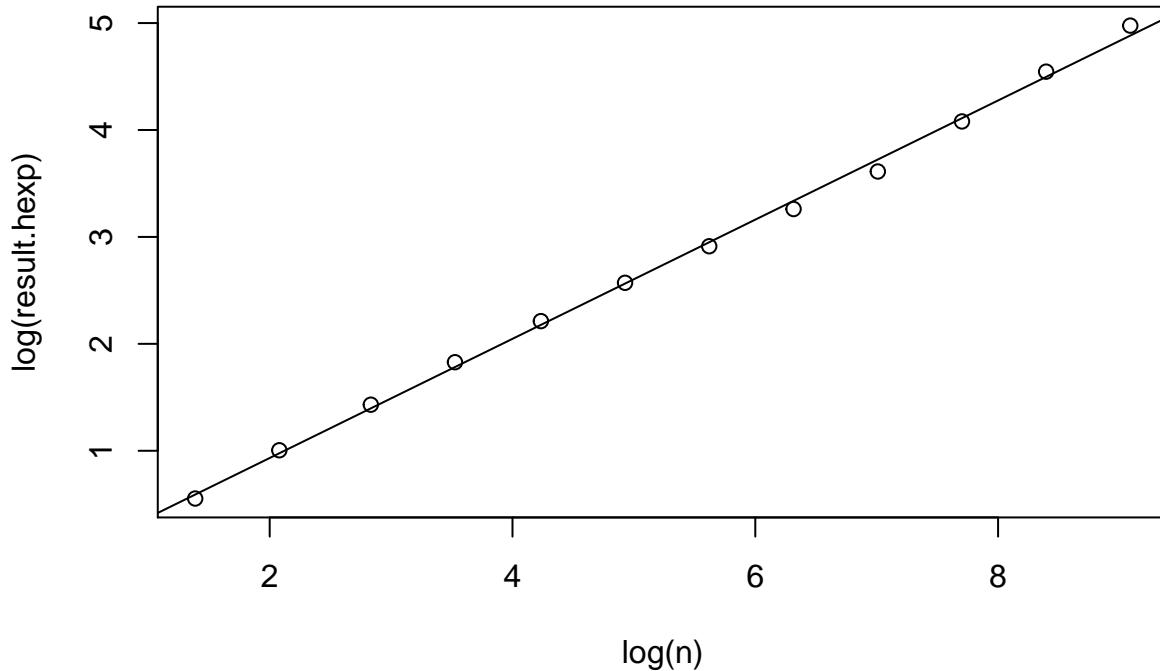
plot(log(result.hexp) ~ log(n), main = "Linear Regression Plot of our R/S Results (Slope is Hurst Exponent)

#Plot Regression line
rl <- lm(log(result.hexp) ~ log(n)); rl

##
## Call:
## lm(formula = log(result.hexp) ~ log(n))
##
## Coefficients:
## (Intercept)      log(n)
##       -0.1821      0.5573
abline(rl$coefficients[1], rl$coefficients[2])

```

Linear Regression Plot of our R/S Results (Slope is Hurst Exponent)



```

#Hurst Exponent:
hexp <- rl$coefficients[2]; cat("Hurst Exponent: ", hexp, "\n\n")

## Hurst Exponent: 0.5572579
#Sanity Test using the built in R Hurst Exponent:
cat("Sanity Check Hurst Exponent: \n"); hurstexp(diffs); cat("\n\n")

## Sanity Check Hurst Exponent:
## Simple R/S Hurst estimation:      0.5361279
## Corrected R over S Hurst exponent: 0.5467487
## Empirical Hurst exponent:        0.5094824
## Corrected empirical Hurst exponent: 0.4833059
## Theoretical Hurst exponent:      0.5257333

```

```

#Approximately .5 across all of the analysis. This indicates that our data is,
#in fact, a random walk.

#Since  $H = 2 - D$  where  $H$  is our Hurst Exponent and  $D$  is fractal dimension, our fractal dimension
#of our stock prices should be approximately 1.5
#Using the fractal dimensions package, we can see that our fractal dimension is approximately
#1.5 across 4 separate methods of fractal analysis:
#(For this analysis, we use the stock prices not the stationary first difference)

fracd.estimates <- fd.estimate(DJI$Open, methods = c("variogram", "madogram",
                                                       "hallwood")); cat("Fractal Dimension: ", fracd.est)

## Fractal Dimension: 1.497758 1.471524 1.481244
#One final check for our Hurst Exponent:
cat("\n\n 2 - D:", 2 - fracd.estimates$fd)

##
##
## 2 - D: 0.5022425 0.5284762 0.5187562
#Correct again. We get .5 across all 3 methods. The Dow Jones Index seems to be following a
#random walk.

```

As you can see, our values for our hurst exponent are hovering around .5 which indicates that our stock price changes are experiencing wild randomness. Let's compare this to hurst exponent we get from our random walk model:

```

hurstexp(diff(rw.drift))

## Simple R/S Hurst estimation:      0.5171415
## Corrected R over S Hurst exponent: 0.5462096
## Empirical Hurst exponent:        0.527856
## Corrected empirical Hurst exponent: 0.5021597
## Theoretical Hurst exponent:       0.5257333

```

As we might expect, our random walk model is also experiencing wild randomness. Therefore, we can conclude with near certainty that our stock price data is a form of a random walk but not a Gaussian Random Walk. Furthermore, we cannot conclude any details about the distribution of the first difference from the hurst exponent other than it is not normally distributed. Although this means more investigation, it is a beautiful problem to have because it signals that there is even more depth and richness to the concept of randomness. We have to deal with the step sizes of our random walk, which means we must return to our first difference distribution and attempt to incorporate the high peaks and heavy tails into our model. This will serve as the core test of whether or not our data has infinite variance but it requires that we investigate two new distributions. The first of which, is the Cauchy Distribution.

Cauchy Distribution

The Cauchy Distribution is best known for its undefined moment generating functions. As a result, the distribution has infinite variance which results in a high peak and long heavy tails.

Surprisingly, the distribution is only characterized by two parameters: location and scale which we esimated using the median and an innerquartile method used the *Goodness-of-Fit Testing for the Cauchy Distribution with Application to* by M. Mahdizadeh, and Ehsan Zamanzade. Given the shape of the our data's histogram, and the underlying theoretical basis, we can investigate whether our data or the log of our data is well modeled by a Cauchy distribution:

```

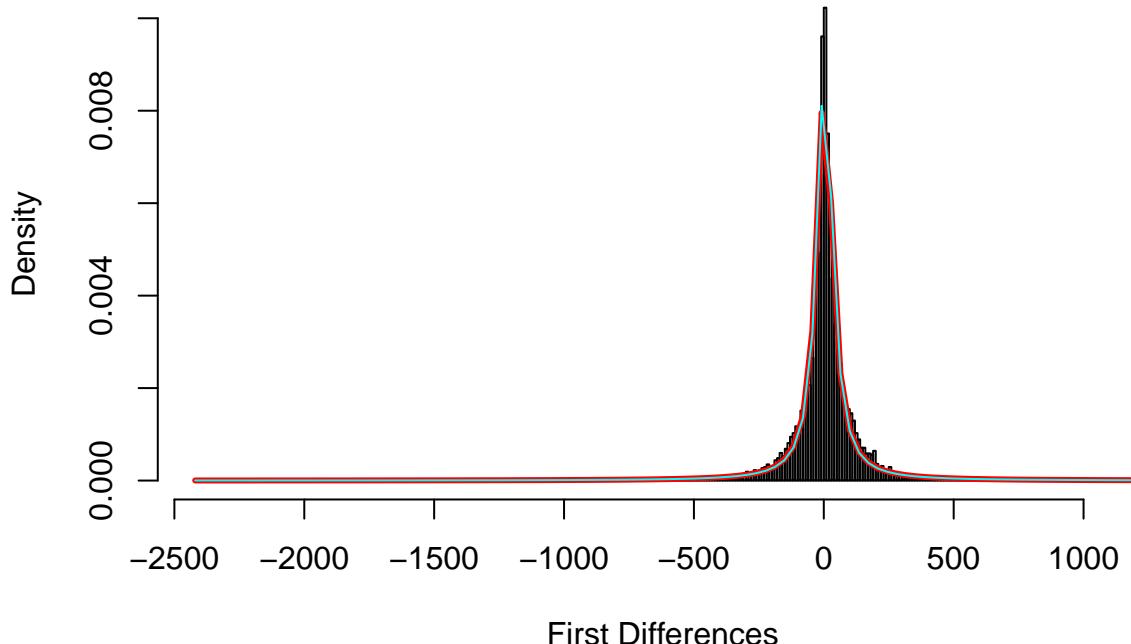
#First Difference of our data:
#Median:
diffs.median <- median(diffs) #3.429688
#Half Interquartile Range:
diffs.hiq <- (quantile(diffs)[[4]] - quantile(diffs)[[2]]) /2 # 36.41016

#Save the parameters for later in the report:
daily.cauchy.params <- c(diffs.median, diffs.hiq)

#Checking our paramters against the fitdist paramters (nearly equal). We will test both paramters to
fit.diffs <- as.vector(fitdist(diffs, "cauchy")$estimate)
hist(diffs, prob = TRUE, breaks = "FD", main = "Histogram of First Differences
    Cauchy Model", xlab = "First Differences")
curve(dcauchy(x, location = diffs.median, scale = diffs.hiq), add = TRUE, lwd = 3, col = "red")
curve(dcauchy(x, location = fit.diffs[1], scale = fit.diffs[2]), add = TRUE, lwd = 1, col = "cyan")

```

Histogram of First Differences Cauchy Model



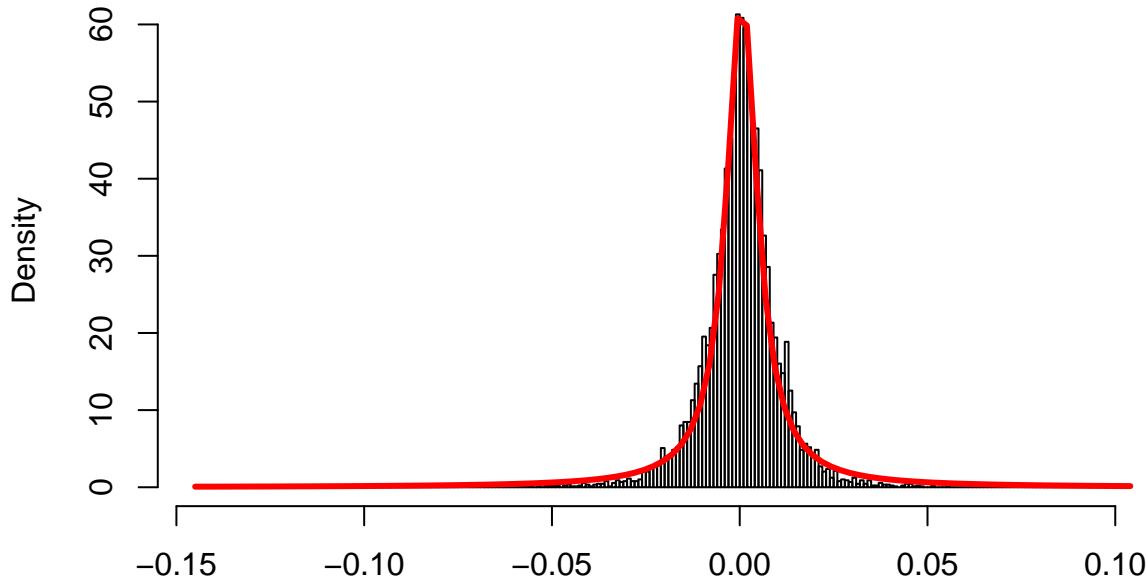
```

#First Difference of the Logarithm of our Data
#Median:
diffs.median <- median(diff(log(DJI$Open))) #0.0005814626
#Half Interquartile Range:
diffs.hiq <- (quantile(diff(log(DJI$Open)))[[4]] - quantile(diff(log(DJI$Open)))[[2]]) /2 # 0.00496212

hist(diff(log(DJI$Open)), prob = TRUE, breaks = "FD", main = "Histogram of Logarithm of the First Difference
    Cauchy Model", xlab = "Logarithm of First Differences")
curve(dcauchy(x, location = diff.medians[1], scale = diff.hiqs[1]), add = TRUE, lwd = 3, col = "red")

```

Histogram of Logarithm of the First Differences Cauchy Model



Logarithm of First Differences

As we

can see from the curves, the Cauchy distribution does a good job of fitting our original data visually, and an excellent job of modeling the first difference of the logarithm of our stock prices. This is not definitive proof that our data comes from a cauchy distribution. However, this is a good time to demonstrate that if our data is modeled by a Cauchy distribution with the location and scale parameters above, then it does indeed have infinite variance. To do that, we'll need to define functions that will describe the integrands that will allow us to use the tail-integral theorem. First, we'll define the integrand that will give us $E(X)$, as well as well as $E(X^2) - E(X)^2$:

Divergent Integrals, Divergent Variance - Cauchy Distribution

All we are doing here is using a given distribution to calculate $E(X)$ and $E(X^2)$ in this manner:

$$E(X) = \int_{-\infty}^{\infty} x \cdot \mu_X$$

$$E(X^2) = \int_{-\infty}^{\infty} x^2 \cdot \mu_X$$

Which gives us

$$Var(X) = E(X^2) - E(X)^2$$

```
## Divergent Integration For Calculating Variance
integrand <- function(x) dcauchy(x, location = diffs.median, scale = diffs.hiq)*x
#and now we have E(X):
exp.x <- integrate(f = integrand, lower = -Inf, upper = Inf)$value; exp.x

## [1] 0.0005814569

#In the same manner, we can try to calculate E(X^2) so that we can get Var = E(X^2) - E(X)^2
integrand2 <- function(x) dcauchy(x, location = diffs.median, scale = diffs.hiq)*x^2
```

```

#And  $E(X^2)$ 
exp.x2 <- integrate(f = integrand2, lower = -Inf, upper = Inf)$value

## Error in integrate(f = integrand2, lower = -Inf, upper = Inf): the integral is probably divergent
#And it appears that the integral is divergent! This means that
#Var =  $E(X^2) - E(X)^2$  also diverges, and thus Var = Inf!

```

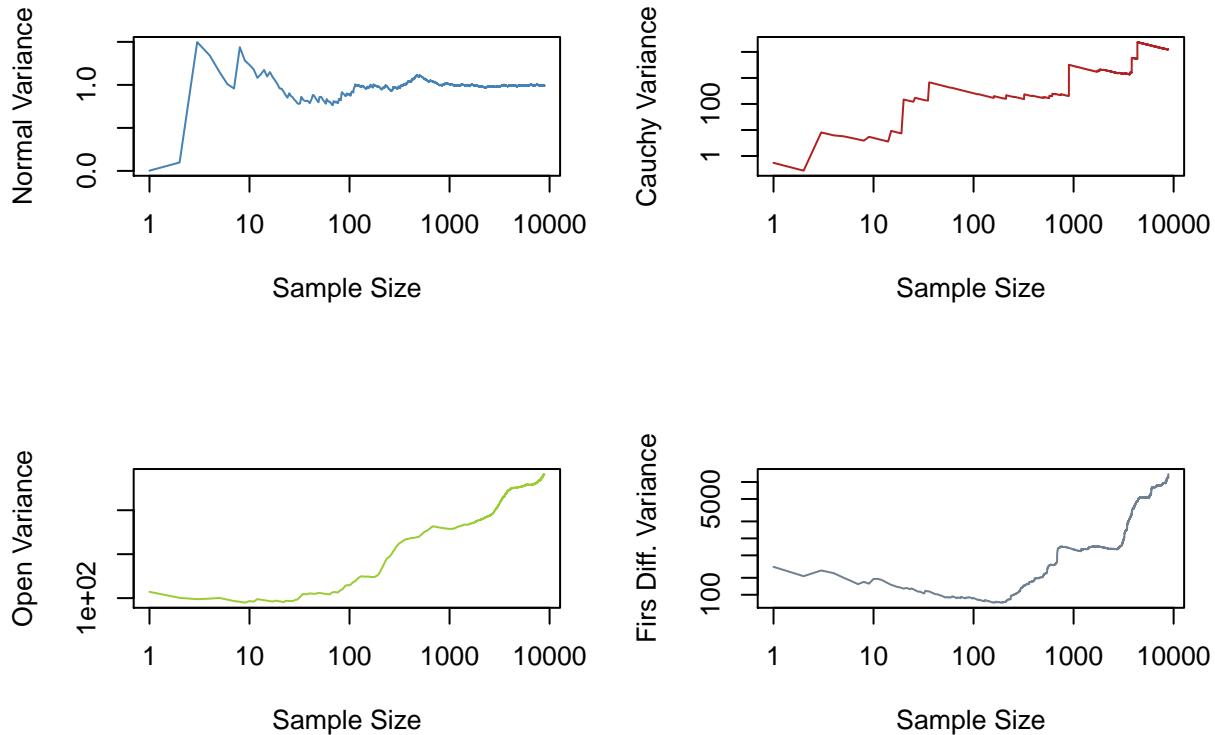
The result is as we expected, the integrals diverge. However, before we run a goodness-of-fit test, we need to demonstrate a few phenomenons we discovered while attempting to model our data with a cauchy distribution. The first of which occurs is a phenomenon that we found while modeling the partial variance of our data.

Partial Variance

```

# Partial Variance to test for convergence of variance
Open <- DJI$Open; diffs.Open <- diff(Open)
N <- length(Open) - 1;
variances.normal <- variances.cauchy <- variances.Open <- variances.diffs <- numeric(N)
sample.normal <- rnorm(N + 1); sample.cauchy <- rcauchy(N + 1)
index <- 1:N
for (i in 2:(N + 1)) {
  variances.normal[i - 1] <- var(sample.normal[1:i])
  variances.cauchy[i - 1] <- var(sample.cauchy[1:i])
  variances.Open[i - 1] <- var(Open[1:i])
  variances.diffs[i - 1] <- var(diffs.Open[1:i])
}
variances.diffs <- variances.diffs[-1]
par(mfrow = c(2,2)) # create 2x2 plot matrix
plot(index,variances.normal, type = "l", col = "steelblue", log = "x", ylab = "Normal Variance", xlab =
plot(index,variances.cauchy, type = "l", col = "firebrick", log = "xy", ylab = "Cauchy Variance", xlab =
plot(index,variances.Open, type = "l", col = "yellowgreen", log = "xy", ylab = "Open Variance", xlab =
plot(head(index,-1),variances.diffs, type = "l", col = "slategray", log = "xy", ylab = "Firs Diff. Vari

```



```

par(mfrow = c(1,1)) # revert to 1x1 plot matrix
summary(variances.normal) # data is centered closely around mean and median

##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## 0.0030  0.9886  0.9924  0.9929  0.9971  1.4968

summary(variances.cauchy) # seems to be large spread

##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## 0.273  1681.493 12448.393 9454.285 16154.159 23799.489

summary(variances.Open) # extremely large spread

##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## 62     434143 10600731 10185446 14778516 43724789

summary(variances.diffs) # spread is larger than it is for Cauchy but less than Open prices

##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.   NA's
## 72.4   695.8  4892.5  4540.3  8128.1 13760.5      1

```

As index increases, partial variance converges for the normal distribution, but it diverges in jagged jumps for the Cauchy distribution and in smoother curves for both Open values and first differences. This indicates that our data may have undefined or infinite variance. Although this provides more evidence for the Cauchy distribution, the next phenomenon we discovered provides competing evidence.

Bootstrap

If our first differences data are observations from independent and identically distributed random variables from a distribution with light tails, then the shape of a histogram of our standardized data should approximate that of a standard normal distribution with moderately large sample sizes. If however our data comes from a heavy-tailed distribution, the presence of extreme values will limit the approximation to a standard normal. We compare our first differences with the normal distribution and two Cauchy distributions parameterized

differently in a bootstrap. The first differences sample data are treated as a population and similarly sized data are randomly drawn from the reference distributions fitted to the first differences data. Then, we treat our samples as populations and draw bootstrap samples of size n from each N times. The bootstrap samples are standardized and plotted as empirical cumulative distributions (eCDF) to get a sense of their typical Kolmogorov-Smirnov test statistic, the max delta between the eCDF and the reference CDF which is plotted as a standard normal eCDF in our case.

The overall result is that our first differences data may lie somewhere between Gaussian and Cauchy on the parameter scale for stable distributions. However, the data is a sample from an underlying population, so the limited accessibility to a sample size of only 8857 limits the capture of extreme values from possibly heavy right tails. This phenomenon will be even more pronounced when we analyze monthly and yearly price change averages. But first, we need to introduce a distribution that can capture both normal and cauchy features depending on it's paramters. For that, we turn to the (Pareto-Levy) Stable distribution.

(Pareto-Levy) Stable Distribution

The (Pareto-Levy) Stable Distribution or simply, the stable distribution is considered a generalization of both cauchy and normal distributions. It is paramaterized by four paramaters $s(\alpha, \beta, \gamma, \delta)$. When $\alpha = 1$ the distribution is normal and when $\alpha = 2$ and $\beta = 0$ the distribution is Cuachy. A small proof of this is demonstrated as such.

Consider the characteristic function of the stable distribution:

$$s(\alpha, \beta, \gamma, \delta) = \exp(it\mu - |ct|^\alpha(1 - i\beta \text{sgn}(t)\Phi))$$

where Φ is defined as:

$$\Phi = \begin{cases} \tan(\frac{\pi\alpha}{2}) & \alpha \neq 1 \\ -\frac{2}{\pi} \log |t| & \alpha = 1 \end{cases}$$

The standard stable distribution characteristic function can be written as:

$$s(\alpha, \beta, 0, 1) = \exp(-|t|^\alpha(1 + i\beta \text{sgn}(t)\Phi))$$

Normal Distribution ($\alpha = 2$): With $\alpha = 2$, $\Phi = \tan(\pi) = 0$, therefore we are left with: $s(2, \beta, 0, 1) = \exp(-t^2)$ which the characteristic function of the standard normal distribution which can be further generalized to the characteristic function of the normal distribution.

Cauchy Distribution ($\alpha = 1$ and $\beta = 0$): With $\beta = 0$, we are left with: $s(1, 0, 0, 1) = \exp(-t^\alpha)$. However, since $\alpha = 1$, we get $s(1, 0, 0, 1) = \exp(-t)$ which is the characteristic function of a standard cauchy distribution which can be further generalized to the characteristic function of the Cauchy distribution. ■

Implementing the Stable Distribution

Although powerful, the stable distribution is tedious to paramaterize from a data set. There are many methods to investigate on how to efficiently and accurately esimate parameters but we will be relying on a fairly consistent inner quartile method popularized by J. Huston McCulloch, PHD (Ohio State University). Our implmentation using our first differences and our log first differences is as follows:

```
#Stock Price Changes:
stable.Xs <- quantile(diffs, c(.05, .25, .5, .75, .95))

#Calculate V's
stable.V_a <- (stable.Xs[[5]] - stable.Xs[[1]]) / (stable.Xs[[4]] - stable.Xs[[2]])
stable.V_b <- (stable.Xs[[5]] + stable.Xs[[3]] - (2*stable.Xs[[3]])) / (stable.Xs[[5]] - stable.Xs[[1]])
```

```

#Using McCulloch's table we calculate alpha and beta
stable.a <- 1.13
stable.b <- 0

#Calculate Phi_3 from the table as well
stable.phi_3 <- 2.312

#Use phi_3 to calculate scale.
stable.c <- (stable.Xs[[4]] - stable.Xs[[2]]) / stable.phi_3

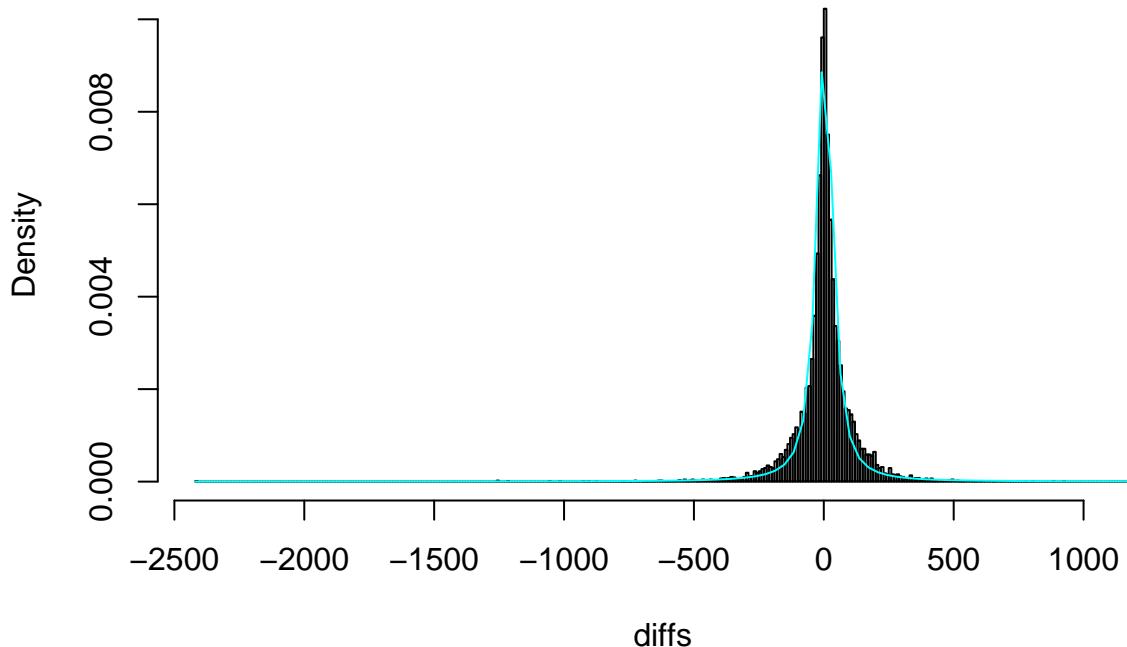
#Since beta = 0, the location is simply the median. (We will also use this method with any small value)
stable.location <- median(diffs)

#Save these parameters for later in the report:
daily.stable.params <- c(stable.a, stable.b, stable.c, stable.location)

#Plot:
hist(diffs, breaks = "FD", freq = FALSE, main = "Stock Price Changes")
curve(dstable(x, (stable.a),(stable.b),stable.c, stable.location), add = TRUE, col = "cyan")

```

Stock Price Changes



```

#Log Stock Price Changes:
logDiffs <- diff(log(DJI$Open))

stable.Xs <- quantile(logDiffs, c(.05, .25, .5, .75, .95))
#Calculate V's
stable.V_a <- (stable.Xs[[5]] - stable.Xs[[1]]) / (stable.Xs[[4]] - stable.Xs[[2]])
stable.V_b <- (stable.Xs[[5]] + stable.Xs[[1]] - (2*stable.Xs[[3]])) / (stable.Xs[[5]] - stable.Xs[[1]])
#Using Table we calculate alpha and beta
stable.a <- 1.484

```

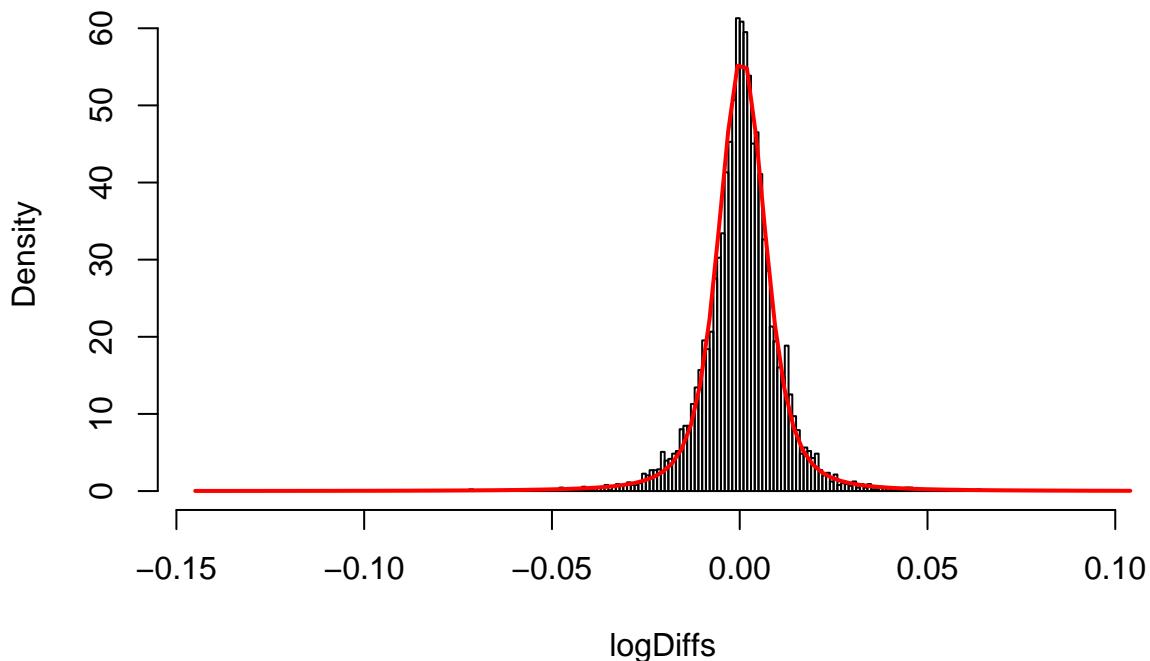
```

stable.b <- 0
#Calculate Phi_3
stable.phi_3 <- 1.939
#Use phi_3 to calculate scale and then location is found from the table
stable.c <- (stable.Xs[[4]] - stable.Xs[[2]]) / stable.phi_3
stable.location <- median(logDiffs)

hist(logDiffs, breaks = "FD", freq = FALSE, main = "Histogram of Logarithmic Stock Price Changes")
curve(dstable(x, stable.a, stable.b, stable.c, stable.location), add = TRUE, lwd = 2, col = "red")

```

Histogram of Logarithmic Stock Price Changes



As you can see, the stable distribution is a good fit for our Stock Price Changes and an excellent fit for our logarithmic stock price changes. We feel that with better parameter estimation techniques, we could more accurately fit the distribution.

This relationship between normal, stable, and cauchy distributions provides insight into the complexity that was raised in the bootstrapping phenomenon that we explored earlier. More specifically, it answers the question of why certain time frames of stock returns (Daily, Monthly, and Yearly) demonstrate properties of different distributions. First, let's look at the graphs of the three distributions across all three time frames:

```

#Histogram:
hist(diff, breaks = "FD", freq = FALSE, main = "Histogram of Stock Price Changes (Daily)", xlab = "Price"
#Plot Curves:
curve(dnorm(x, mean(diff), sd(diff)), add = TRUE, lwd = 2, col = "blue")

curve(dcauchy(x, location = daily.cauchy.params[1],
              scale = daily.cauchy.params[2]),
      add = TRUE, lwd = 3, col = "red")

curve(dstable(x, daily.stable.params[1], daily.stable.params[2],

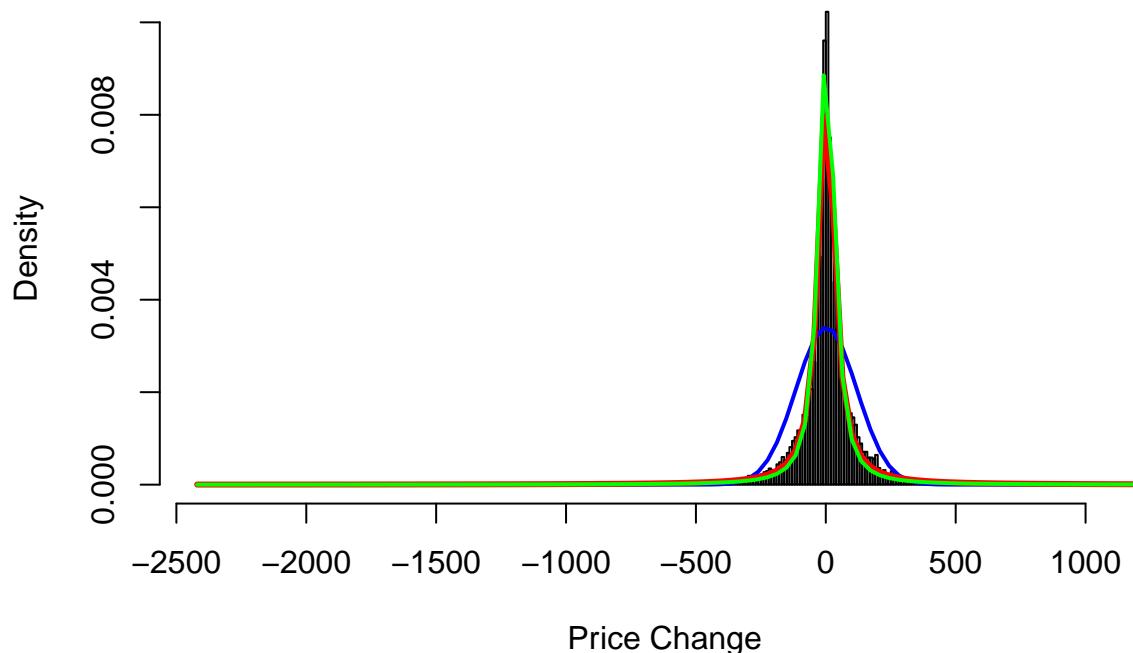
```

```

    daily.stable.params[3], daily.stable.params[4]),
add = TRUE, lwd = 2, col = "green")

```

Histogram of Stock Price Changes (Daily)



```

#Monthly
#Get Data:
monthly <- DJI; monthly$Date <- as.Date(monthly$Date)
monthly$Date <- format(as.Date(monthly$Date, format = "%d/%m/%Y"), "%Y-%m")
monthly <- aggregate(monthly[, 2:4], list(monthly$Date), mean, drop = TRUE)
colnames(monthly)[1] <- "Date"
monthlyDiffs <- diff(monthly$Open)

#Histogram
hist(monthlyDiffs, breaks = "FD", freq = FALSE, main = "Histogram of Stock Price Changes (Monthly)", xla

# Set Parameters for Stable and Cauchy Distributions:
#Stable
stable.Xs <- quantile(monthlyDiffs, c(.05, .25, .5, .75, .95))
#Calculate V's
stable.V_a <- (stable.Xs[[5]] - stable.Xs[[1]]) / (stable.Xs[[4]] - stable.Xs[[2]])
stable.V_b <- (stable.Xs[[5]] + stable.Xs[[1]] - (2*stable.Xs[[3]])) / (stable.Xs[[5]] - stable.Xs[[1]])
#Using Table we calculate alpha and beta
stable.a <- 1.279
stable.b <- 0
#Calculate Phi_3
stable.phi_3 <- 1.955
#Use phi_3 to calculate scale and then location is found from the table
stable.c <- (stable.Xs[[4]] - stable.Xs[[2]]) / stable.phi_3
stable.location <- median(monthlyDiffs)

```

```

#Save parameters for later in the report:
monthly.stable.params <- c(stable.a, stable.b, stable.c, stable.location)

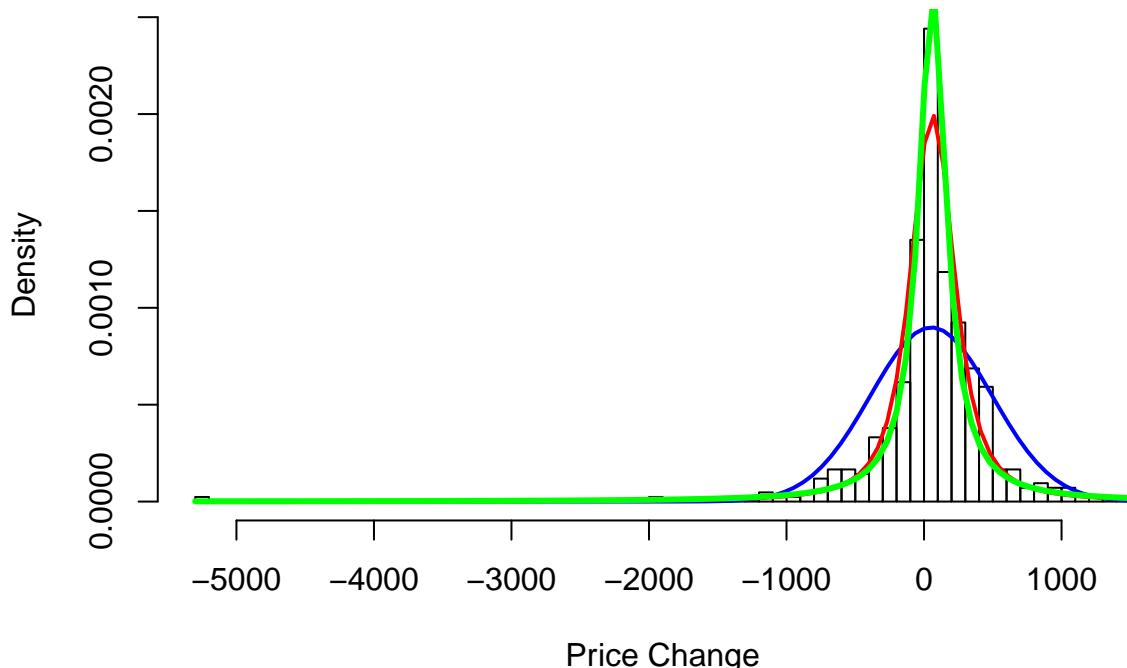
#Cauchy
cauchy.median <- median(monthlyDiffs)
cauchy.hiq <- (quantile(monthlyDiffs)[[4]] - quantile(logDiffs)[[2]]) / 2

#Save parameters for later in the report:
monthly.cauchy.params <- c(cauchy.median, cauchy.hiq)

#Plot Curves:
curve(dnorm(x, mean(monthlyDiffs), sd(monthlyDiffs)), col = "blue", lwd = 2, add = TRUE)
curve(dstable(x, stable.a, stable.b, stable.c, stable.location), add = TRUE, lwd = 2, col = "red")
curve(dcauchy(x, cauchy.median, cauchy.hiq), add = TRUE, col = "green", lwd = 3)

```

Histogram of Stock Price Changes (Monthly)



```

#Yearly
#Get Data:
yearly <- DJI; yearly$Date <- as.Date(yearly$Date)
yearly$Date <- format(as.Date(yearly$Date, format = "%d/%m/%Y"), "%Y")
yearly <- aggregate(yearly[, 2:4], list(yearly$Date), mean, drop = TRUE)
colnames(yearly)[1] <- "Date"
yearlyDiffs <- diff(yearly$Open)

#Histogram:
hist(yearlyDiffs, breaks = "FD", freq = FALSE, main = "Histogram of Stock Price Changes (Yearly)", xlab

#Set Parameters for Stable and Cauchy Distribution:
#Stable

```

```

stable.Xs <- quantile(yearlyDiffss, c(.05, .25, .5, .75, .95))
#Calculate V's
stable.V_a <- (stable.Xs[[5]] - stable.Xs[[1]]) / (stable.Xs[[4]] - stable.Xs[[2]])
stable.V_b <- (stable.Xs[[5]] + stable.Xs[[1]] - (2*stable.Xs[[3]])) / (stable.Xs[[5]] - stable.Xs[[1]])
#Using Table we calculate alpha and beta
stable.a <- 1.388
stable.b <- -0.165
#Calculate Phi_3
stable.phi_3 <- 1.795
#Use phi_3 to calculate scale and then location is found from the table
stable.c <- (stable.Xs[[4]] - stable.Xs[[2]]) / stable.phi_3
stable.location <- median(yearlyDiffss)

#Save parameters for later in the report:
yearly.stable.params <- c(stable.a, stable.b, stable.c, stable.location)

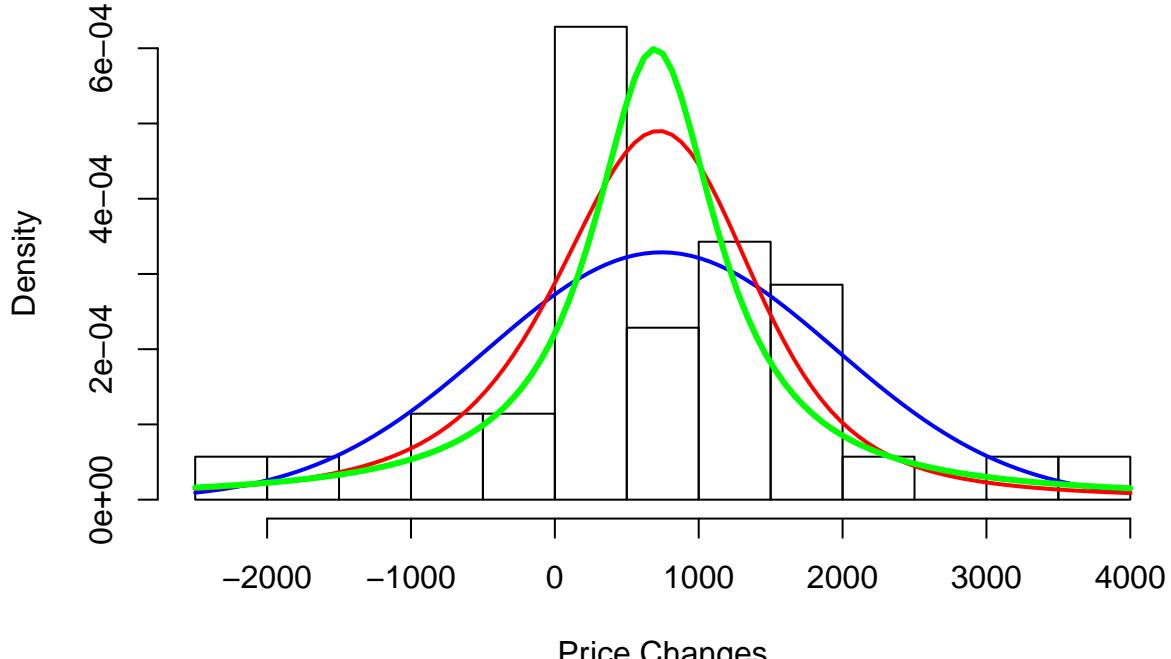
#Cauchy
cauchy.median <- median(yearlyDiffss)
cauchy.hiq <- (quantile(yearlyDiffss)[[4]] - quantile(yearlyDiffss)[[2]]) / 2

#Save parameters for later in the report:
yearly.cauchy.params <- c(cauchy.median, cauchy.hiq)

#Plot the Curves:
curve(dnorm(x, mean(yearlyDiffss), sd(yearlyDiffss)), col = "blue", lwd = 2, add = TRUE)
curve(dstable(x, stable.a, stable.b, stable.c, stable.location), add = TRUE, lwd = 2, col = "red")
curve(dcauchy(x, cauchy.median, cauchy.hiq), add = TRUE, col = "green", lwd = 3)

```

Histogram of Stock Price Changes (Yearly)



As you can see, the normal distribution becomes a better fit as the time frame accounts for longer periods of price change. This is sometimes considered a contradiction in the use of cauchy distributions to model price changes.

This is why using a Stable distribution, as a generalized version of both Cuachy and Normal distributions, is so important to capturing the behavior of stock price changes over time.

A simple observation of the hurst exponent at each time frame gives a better understanding of what is happening:

```
cat("Daily Price Change Hurst Exponent:\n")

## Daily Price Change Hurst Exponent:
hurstexp(diffs)

## Simple R/S Hurst estimation:      0.5361279
## Corrected R over S Hurst exponent: 0.5467487
## Empirical Hurst exponent:        0.5094824
## Corrected empirical Hurst exponent: 0.4833059
## Theoretical Hurst exponent:       0.5257333

cat("\n\nMonthly Price Change Hurst Exponent:\n")

##
##
## Monthly Price Change Hurst Exponent:
hurstexp(monthlyDiffs)

## Simple R/S Hurst estimation:      0.5945887
## Corrected R over S Hurst exponent: 0.6531656
## Empirical Hurst exponent:        0.6505578
## Corrected empirical Hurst exponent: 0.6020064
## Theoretical Hurst exponent:       0.5466761

cat("\n\nYearly Price Change Hurst Exponent:\n")

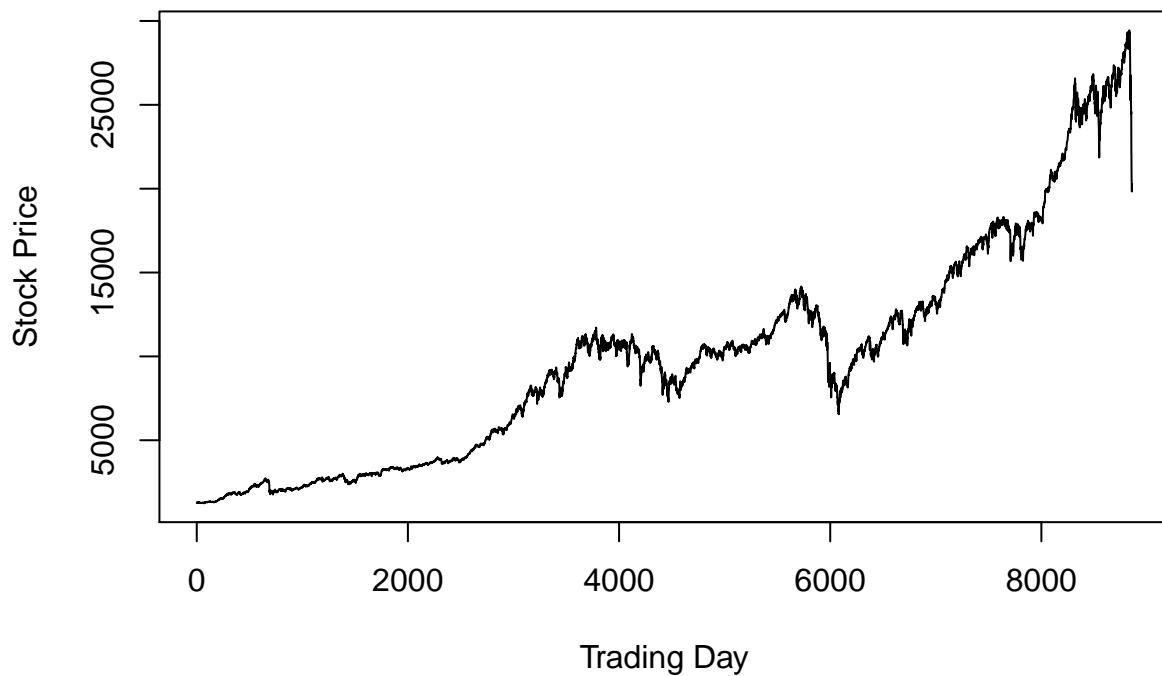
##
##
## Yearly Price Change Hurst Exponent:
hurstexp(yearlyDiffs)

## Simple R/S Hurst estimation:      0.6057436
## Corrected R over S Hurst exponent: 0.74063
## Empirical Hurst exponent:        0.6878735
## Corrected empirical Hurst exponent: 0.5944356
## Theoretical Hurst exponent:       0.6036964
```

The hurst exponent is moving away from .5 as the data set accounts for broader periods of time. In other words, the stock prices show more mild randomness as we go from daily stock price changes to yearly stock price changes. A graphical representation of our stock prices over time explains why this is occurring:

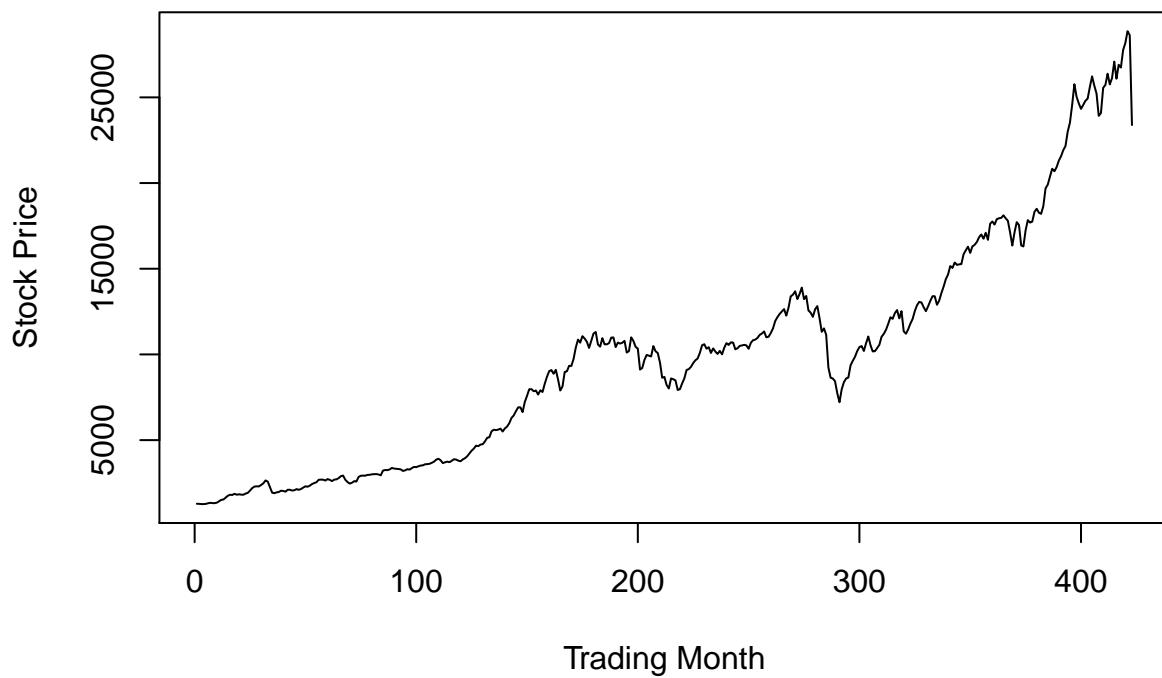
```
plot(DJI$Open, type = "l", xlab = "Trading Day", ylab = "Stock Price", main = "Daily Stock Prices Over 1000 Days")
```

Daily Stock Prices Over Time



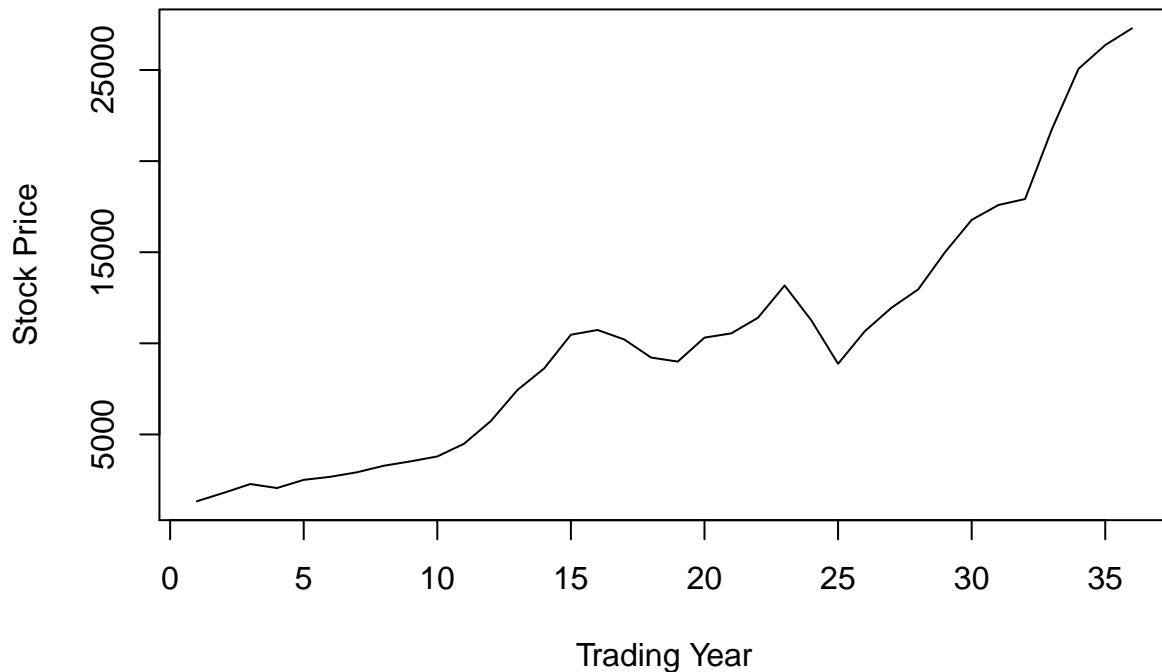
```
plot(monthly$Open, type = "l", xlab = "Trading Month", ylab = "Stock Price", main = "Monthly Stock Price Averages Over Time")
```

Monthly Stock Prices Averages Over Time



```
plot(yearly$Open, type = "l", xlab = "Trading Year", ylab = "Stock Price", , main = "Yearly Stock Price Averages Over Time")
```

Yearly Stock Prices Averages Over Time



Thinking about each a graph as a fractal, we see that curve is losing it's "roughness", and therefore it experiences less wild randomness. To think about this in concrete terms, a 2000 point drop in Daily Stock Prices will appear as a more extreme outlier when compared to a 2000 point drop in Yearly Stock Prices.

Does this mean that yearly stock prices follow a normal distribution? Not necessarily. In fact, we hypothesize the opposite. For instance, we have 8857 data points (or days) in our Daily Stock Price Data compared to 35 data points (or years) in our yearly averages. We believe that if were to check stock market prices of our yearly averages in 8000 years, they would be better fit by a stable distribution with $\alpha \rightarrow 2$ or "more cauchy".

Goodness-of-Fit Tests - Kolmogorov-Smirnov

Another way to consider whether our data is well modeled by a given distribution is via the Kolmogorov-Smirnov test, which measures the maximum vertical difference between the CDF's of 2 functions. Let us then explore what this tells us about the DJI data. First, we'll consider our results from using the estimated interquartile parameters:

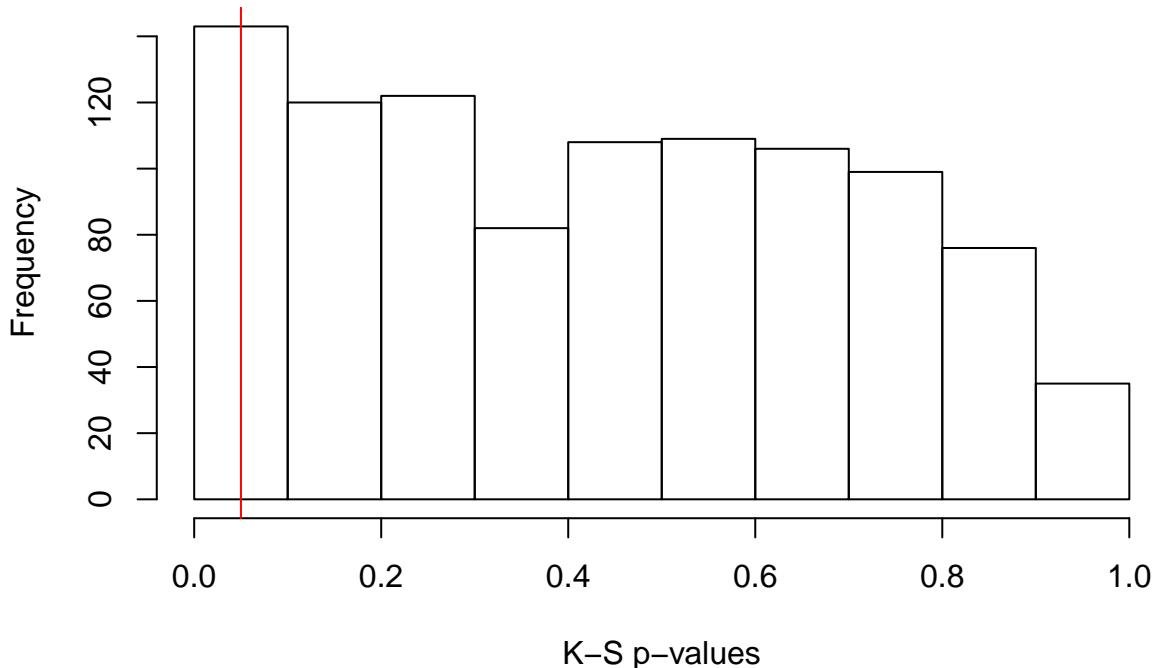
```
nn <- 500 #sample size.

N <- 10^3
ks.stats2 <- numeric(N)
for (i in 1:N) {
  rand2 <- rcauchy(nn, location = fit.diffs[1], scale = fit.diffs[2]); head(rand)
  diff.samp2 <- sample(diffs,nn, replace = TRUE)
  ks.stats2[i] <- ks.test(diff.samp2,rand2)$p.value
}
mean(ks.stats2) #mean pvalues = 0.4248467

## [1] 0.4299057
sum(ks.stats2 > 0.05)/length(ks.stats2) #About 92% of the time our random samples generate a p-value gre
```

```
## [1] 0.925
{hist(ks.stats2, breaks = "fd", main = "Histogram of K-S p-values", xlab = "K-S p-values")
abline(v = 0.05, col = "red")}
```

Histogram of K-S p-values



About 92% of p-values are greater than 0.05, thus we have some convincing evidence in favor of the null hypothesis that the samples come from similar distributions. (refer to main R script for similar analysis using interquartile range and median as estimated Cauchy parameters)

Further investigation is needed to confirm this hypothesis, and would require access to an abundance of intra-day data. But, the a deeper question that is not answered in this report is whether there exists a concrete relationship between the volume of data and our α parameter.

That being said, we feel that there is enough evidence from the relationship between our data and the normal, cauchy, and stable distributions to suggest that our data more than likely has infinite variance.

To clearly demonstrate this relationship we will run a series of goodness-of-fit tests and explain some of the complexities and challenges we faced when attempting to confirm the fit of our model.

As a robustness test, let's compare samples taken from the data to samples taken from a Cauchy distribution

Taking N samples from diff and using fitdist to fit each of those. Then comparing each of those samples to a Cauchy distribution with those same parameters, and then running a ks-test:

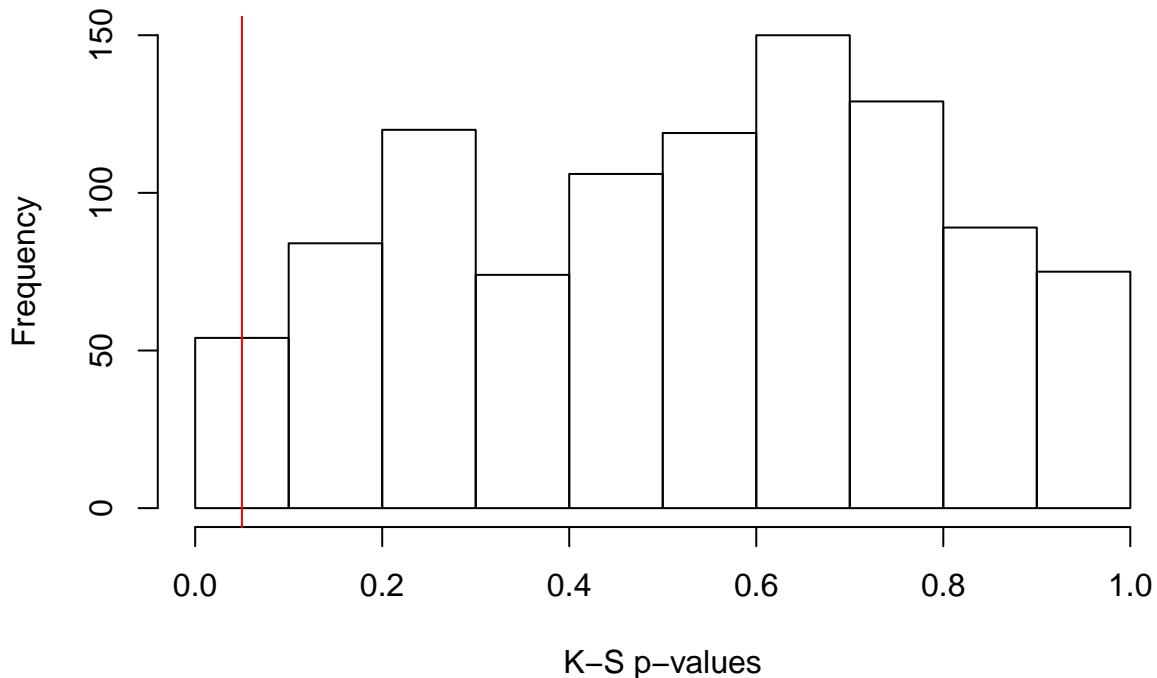
```
N <- 10^3
samp.rand <- numeric(N)
nn <- 500
ks.samp.rand <- numeric(N)
for(i in 1:N){
  diff.samp <- sample(diff, nn, replace = TRUE)
  fit.samp <- fitdist(diff.samp, "cauchy", "mle")
```

```

cauchy.samp <- rcauchy(nn, location = fit.samp$estimate[1], scale = fit.samp$estimate[2])
ks.samp.rand[i] <- ks.test(diff.samp, cauchy.samp)$p.value
}
sum(ks.samp.rand > 0.05)/length(ks.samp.rand) #About 99% of the time our random samples generate a p-val
## [1] 0.98
{hist(ks.samp.rand, breaks = "fd", main = "Histogram of K-S P-values", xlab = "K-S p-values")
abline(v = 0.05, col = "red")}

```

Histogram of K-S P-values



For further robustness tests, as well as for an analysis of sensitivity to sample sizes, refer to the main R script. In summary what we can see is that the K-S test statistic is sensitive to the size of the random sample being drawn from the random Cauchy distribution. We conclude that this is a result of the fact that the diff samples are drawn from a sample that has finite data (DJI), whereas the rcauchy samples are drawn from the true distribution, which has infinite variance and infinitely many data points. We note however, that samples with as many as 500 elements have provided good evidence that we should not reject the null hypothesis.

Goodness-of-Fit Tests - QQ Plots

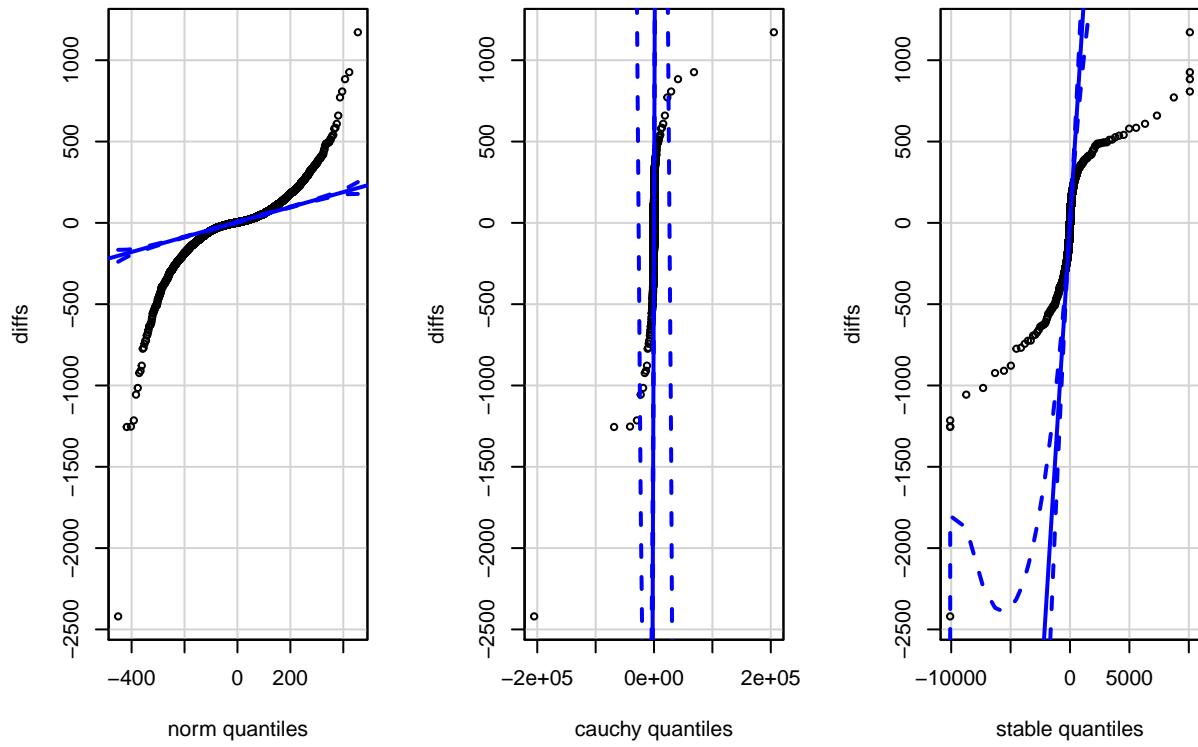
The issue that seems to arise from our goodness-of-fit tests is that the quantity of data we are working with produces large error statistics that result in rejecting the null hypothesis. Therefore, we used QQ Plots to better understand the fit of our 3 distributions across our 3 different time frames:

```

#Daily:
par(mfrow = c(1,3))
qqPlot(diffs, "norm", mean = mean(diffs), sd = sd(diffs), id = FALSE)
qqPlot(diffs, "cauchy", location = daily.cauchy.params[1], scale = daily.cauchy.params[2], id = FALSE)
qqPlot(diffs, "stable", alpha = daily.stable.params[1], beta = daily.stable.params[2], gamma = daily.stable.params[3])
mtext("QQ-Plots of Daily Stock Price Changes", side = 3, line = -1.2, outer = TRUE)

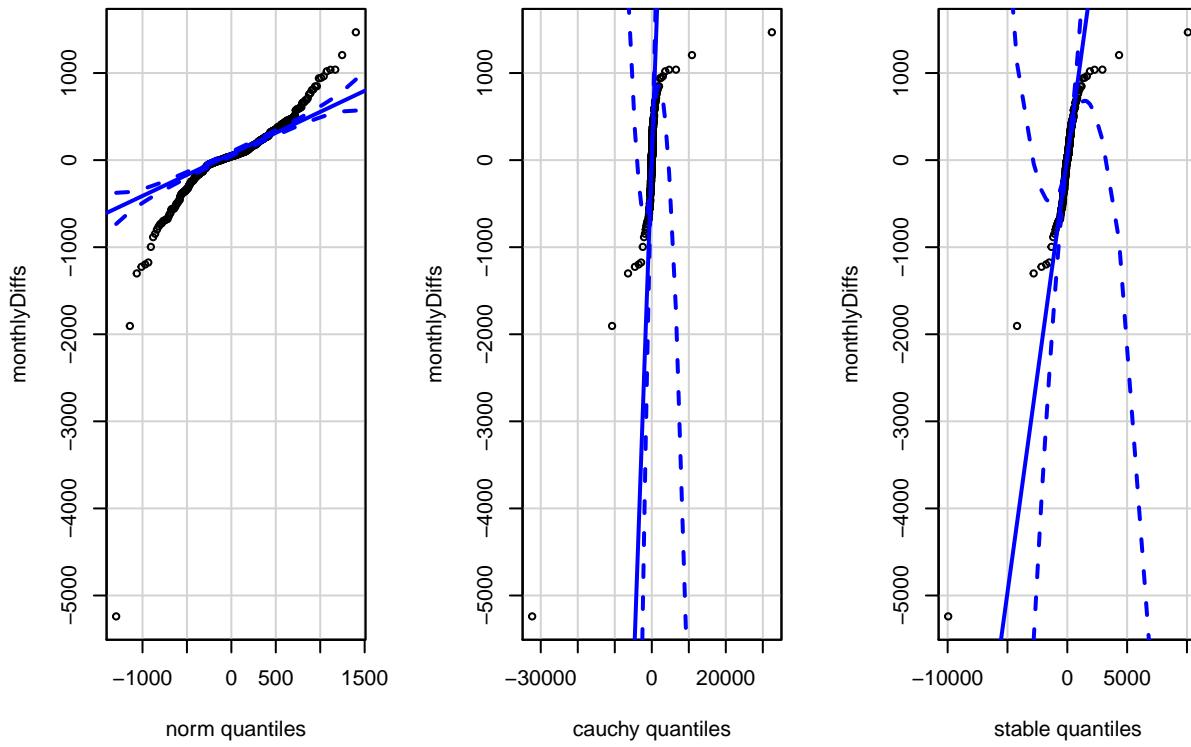
```

QQ-Plots of Daily Stock Price Changes



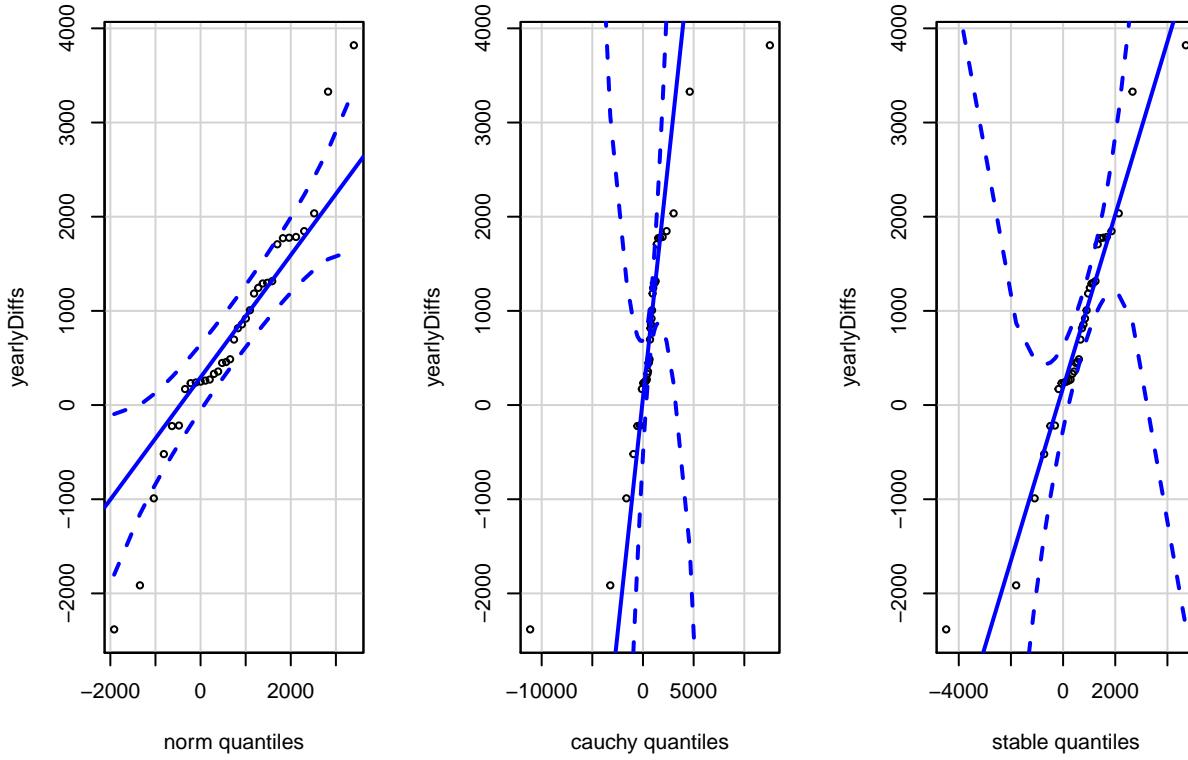
```
#Monthly
par(mfrow = c(1,3))
qqPlot(monthlyDiffs, "norm", mean = mean(monthlyDiffs), sd = sd(monthlyDiffs), id = FALSE)
qqPlot(monthlyDiffs, "cauchy", location = monthly.cauchy.params[1], scale = monthly.cauchy.params[2], id = FALSE)
qqPlot(monthlyDiffs, "stable", alpha = monthly.stable.params[1], beta = monthly.stable.params[2], gamma = monthly.stable.params[3], id = FALSE)
mtext("QQ-Plots of Monthly Stock Price Changes", side = 3, line = -1.2, outer = TRUE)
```

QQ–Plots of Monthly Stock Price Changes



```
#Yearly
par(mfrow = c(1,3))
qqPlot(yearlyDiffss, "norm", mean = mean(yearlyDiffss), sd = sd(yearlyDiffss), id = FALSE);
qqPlot(yearlyDiffss, "cauchy", location = yearly.cauchy.params[1], scale = yearly.cauchy.params[2], id =
qqPlot(yearlyDiffss, "stable", alpha = yearly.stable.params[1], beta = yearly.stable.params[2], gamma =
mtext("QQ–Plots of Yearly Stock Price Changes", side = 3, line = -1.2, outer = TRUE)
```

QQ–Plots of Yearly Stock Price Changes



Across all three time frames we notice that the longer the time period, the better the normal distribution fits the data. Inversely, the worse the Cauchy distribution fits the data. However, the QQ Plot shows the power of the stable distribution as it changes with our time frame to keep a relatively consistent fit. And, since the data seems to be ‘converging’ to a stable distribution with $\alpha \rightarrow 2$, this suffices to demonstrate that our data has infinite variance. Therefore, our stock prices follow a random walk with Levy-flight.

Surveying the impact of the White House on the Dow Jones

Now that we have seen that the data follows a random walk model with infinite variance, there still remains questions about autoregression in the stock market as it relates to fiscal and monetary policy. Let us consider the impact of such policies by examining the hypothesis that political regimes have an impact on the market, here clearly represented by the Dow Jones Industrial Average.

```
regime <- lm(diff(DJI$Open) ~ DJI$Republican[1:length(diff(DJI$Open))] + DJI$Recession[1:length(diff(DJI$Open))])
summary(regime)
```

```
##
## Call:
## lm(formula = diff(DJI$Open) ~ DJI$Republican[1:length(diff(DJI$Open))] +
##     DJI$Recession[1:length(diff(DJI$Open))])
##
## Residuals:
##      Min        1Q    Median        3Q       Max
## -2412.61   -32.54     1.14    40.48   1179.27
##
## Coefficients:
##             Estimate Std. Error t value
## (Intercept)  100.0000   10.0000  10.000
## DJI$Recession  1.0000   0.5000   2.000
## DJI$Republican -0.5000   0.5000  -1.000
```

```

## (Intercept)           5.302    1.858   2.854
## DJI$Republican[1:length(diff(DJI$Open))]TRUE -2.538    2.600  -0.976
## DJI$Recession[1:length(diff(DJI$Open))]TRUE -10.069   3.396  -2.965
##                                         Pr(>|t|)
## (Intercept)           0.00433 ** 
## DJI$Republican[1:length(diff(DJI$Open))]TRUE  0.32895
## DJI$Recession[1:length(diff(DJI$Open))]TRUE  0.00303 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 117.2 on 8854 degrees of freedom
## Multiple R-squared:  0.00138,   Adjusted R-squared:  0.001154
## F-statistic: 6.117 on 2 and 8854 DF,  p-value: 0.002214

```

At first glance, it looks like republican regimes tend to be negatively correlated with growth in the Dow. But let us look a little closer. Given the large standard error for the Republican coefficient, we cannot safely conclude that republican administrations correlate with losses in the Dow. (i.e. the p-value is .3, so we fail to reject the null hypothesis that this control variable has no impact on the response variable.)

Let's see if we can get a statistically significant result for any of the individual presidents:

```

GHWB <- DJI$Regime == "GHWB"
BC <- DJI$Regime == "BC"
GWB <- DJI$Regime == "GWB"
BO <- DJI$Regime == "BO"
DJT <- DJI$Regime == "DJT"

pres.binary <- data.frame(GHWB, BC, GWB, BO, DJT)

regress.data <- data.frame(DJI,pres.binary)

#since we omitted the Ronald Reagan variable, each of the coefficients for the various presidents
#represents the incremental surplus or deficit in the DJIA that occurred during each of the other
#presidents' terms
ind.pres <- lm(regress.data$diffs ~ regress.data$Recession + regress.data$GHWB + regress.data$BC + reg
ind.pres

## 
## Call:
## lm(formula = regress.data$diffs ~ regress.data$Recession + regress.data$GHWB +
##     regress.data$BC + regress.data$GWB + regress.data$BO + regress.data$DJT)
## 
## Coefficients:
##             (Intercept)  regress.data$RecessionTRUE
##                     0.9587                  -11.4885
## regress.data$GHWBTRUE      regress.data$BCTRUE
##                     5.8136                  2.7822
## regress.data$GWBTRUE      regress.data$BOTRUE
##                     2.1005                  6.1711
## regress.data$DJTTRUE
##                     0.3856

####These results are interesting because they seem out of line with the recent (2018-2019) rhetoric of
####Trump administration's success in boosting the DJIA to new heights.
summary(ind.pres)

```

```

## 
## Call:
## lm(formula = regress.data$diffs ~ regress.data$Recession + regress.data$GHWB +
##      regress.data$BC + regress.data$GWB + regress.data$B0 + regress.data$DJT)
## 
## Residuals:
##    Min     1Q   Median     3Q    Max 
## -2409.78 -32.81    1.16  40.05 1182.11 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.9587    3.6984   0.259  0.79547    
## regress.data$RecessionTRUE -11.4885   3.7094  -3.097  0.00196 ** 
## regress.data$GHWBTRUE       5.8136   5.5428   1.049  0.29427    
## regress.data$BCTRUE        2.7822   4.5256   0.615  0.53872    
## regress.data$GWBTRUE       2.1005   4.7369   0.443  0.65746    
## regress.data$BOTRUE        6.1711   4.5494   1.356  0.17499    
## regress.data$DJTTRUE       0.3856   5.5670   0.069  0.94477    
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 117.2 on 8851 degrees of freedom
## Multiple R-squared:  0.001562, Adjusted R-squared:  0.0008856 
## F-statistic: 2.308 on 6 and 8851 DF, p-value: 0.03144

####here again, we find that no presidents' presence in the White House had a significant impact on the ...

```

These results are interesting because they seem out of line with the recent (2018-2019) rhetoric of the Donald Trump administration's success in boosting the DJIA to new heights. Here again, we find that no presidents' presence in the White House had a significant impact on the Dow.

Let us now exclude the days when 5 sigma + events took place, that way we'll keep only the data for "normal/typical" days.

```

regress.data.ne <- regress.data[-idx,]
ind.pres.2 <- lm(regress.data.ne$diffs ~ + regress.data.ne$Recession + regress.data.ne$GHWB + regress.data.ne$BC + regress.data.ne$GWB + regress.data.ne$B0 + regress.data.ne$DJT)

summary(ind.pres.2)

```

```

## 
## Call:
## lm(formula = regress.data.ne$diffs ~ +regress.data.ne$Recession +
##      regress.data.ne$GHWB + regress.data.ne$BC + regress.data.ne$GWB +
##      regress.data.ne$B0 + regress.data.ne$DJT)
## 
## Residuals:
##    Min     1Q   Median     3Q    Max 
## -583.97 -33.44    0.93  38.58 567.08 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.95874   3.25813   0.294  0.768567    
## regress.data.ne$RecessionTRUE -3.57645   3.28551  -1.089  0.276380    
## regress.data.ne$GHWBTRUE       1.85366   4.88588   0.379  0.704406    
## regress.data.ne$BCTRUE        3.03985   3.98714   0.762  0.445833    
## regress.data.ne$GWBTRUE       -0.01293   4.17506  -0.003  0.997529    
## regress.data.ne$BOTRUE        5.53480   4.00908   1.381  0.167448

```

```

## regress.data.ne$DJTTRUE      16.51069    4.93978   3.342 0.000834 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 103.3 on 8819 degrees of freedom
## Multiple R-squared:  0.002272,  Adjusted R-squared:  0.001594
## F-statistic: 3.348 on 6 and 8819 DF,  p-value: 0.002694

```

These results seem to indicate that, excluding days with “extremely” events, and controlling for recessions (and nothing else), when we compare the performance of the DJIA during the previous 6 US presidents, the results indeed appear to be most favorable to Donald Trump, and least favorable to George W Bush. It appears that the negative coefficient related to GWB comes from the effects of the 2008 housing crisis, which took place in the final months of his second term. Additionally, we should note that the only statistically significant coefficient was that of Donald Trump, which had a p-value well below 0.01.

Logistic Regression: Recessions and Results

Given the earlier results, it seems that recessions have a large impact on first differences in daily Open values for the Dow Jones Industrial Average. Let us inspect how economic recessions correlate with the performance of the DJIA.

```

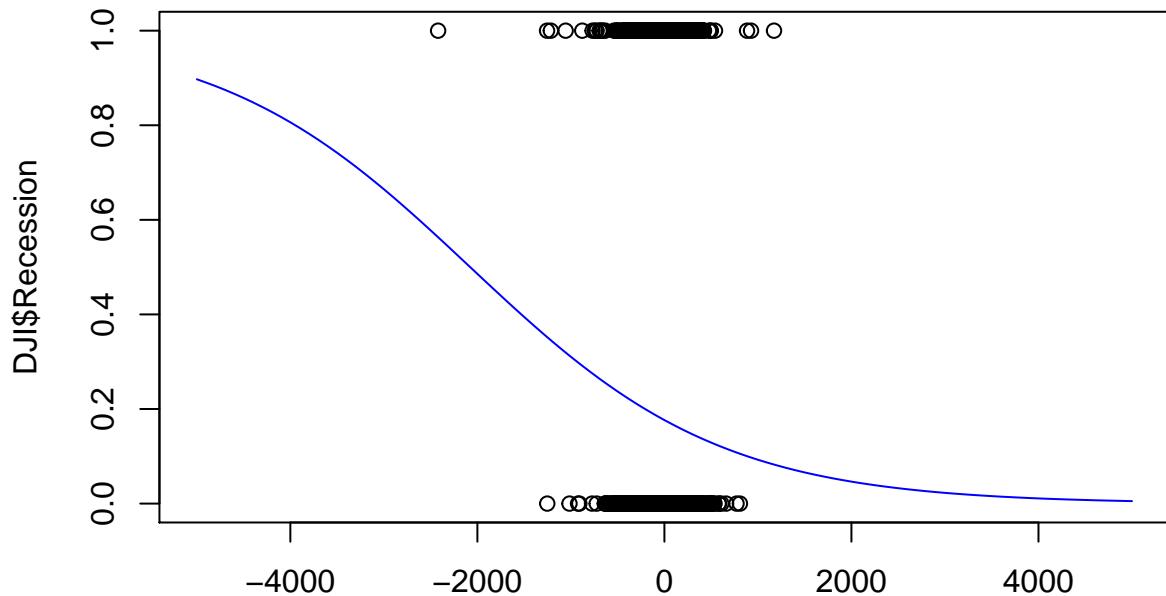
plot(DJI$diffs,DJI$Recession, xlim = c(-5000,5000))
MLL <- function(alpha, beta) {
  -sum( log( exp(alpha+beta*DJI$diffs)/(1+exp(alpha+beta*DJI$diffs)) ) *DJI$Recession
        + log(1/(1+exp(alpha+beta*DJI$diffs)))*(1-DJI$Recession) )
}

#R has a function that will maximize this function of alpha and beta
#install.packages("stats4")  #needs to be run at most once
library(stats4)
results <- mle(MLL, start = list(alpha = 0, beta = 0)) #an initial guess is required
results@coef

##           alpha          beta
## -1.5398359230 -0.0007414154

curve( exp(results@coef[1] + results@coef[2]*x) / (1 + exp(results@coef[1] + results@coef[2]*x)), col =

```



DJI\$diffs

This is

a fairly interesting result because its graph looks different than the normal logistic curve. Of course, this becomes obvious when one considers the nature of the regression, namely that recessions are events that are expected to correlate with negative values of first differences (i.e. price drops). In any case, this provides some evidence to support the hypothesis that negative fluxes in the Dow Jones correlate with economic recessions.

The Impact of Political Regimes - Hypothesis Testing With Permutation Test

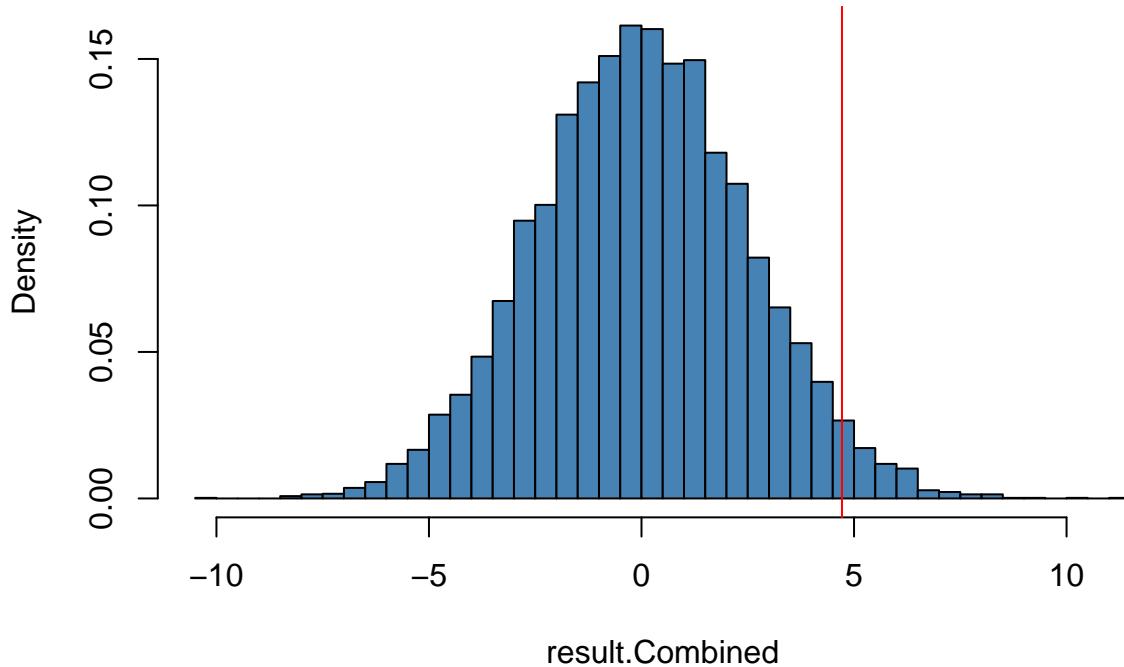
```

RepAvg <- sum(DJI$diffs*(DJI$Republican == TRUE))/sum(DJI$Republican == TRUE) ; RepAvg
## [1] -0.006957805
DemAvg <- sum(DJI$diffs*(DJI$Republican == FALSE))/sum(DJI$Republican == FALSE) ; DemAvg
## [1] 4.709857
Obs <- DemAvg - RepAvg; Obs
## [1] 4.716814
#
N <- 10^4 #number of simulations
result.Combined <- numeric(N) #this is the vector that will store our simulated differences
for (i in 1:N) {
  Rep <- sample(DJI$Republican) #This is our permuted party column
  RepMu <- sum(DJI$diffs*(Rep == TRUE))/sum(Rep == TRUE) ; RepMu
  DemMu <- sum(DJI$diffs*(Rep == FALSE))/sum(Rep == FALSE) ; DemMu
  result.Combined[i] <- DemMu - RepMu
}
mean(result.Combined) #inspecting that these are indeed close to zero
## [1] 0.01476177

```

```
hist(result.Combined, breaks = "FD", probability = TRUE, col = "steelblue")
abline(v = Obs, col = "red")
```

Histogram of result.Combined



```
pvalue <- (sum(result.Combined >= Obs) + 1)/(N + 1) ; pvalue
```

```
## [1] 0.03149685
```

Given a p-value of 2.87% chance that this extreme of an observed difference would arise by chance, so it appears that the DJI performed better during democratic regimes, a result that is statistically significant.

Hypothesis Testing: Contingency table with chi-square test for political party and recession.

```
p <- sum(DJI$Recession)/length(DJI$Recession) # 17.67% of observations are in recession years
obs.tbl <- table(DJI$Republican,DJI$Recession) # Republican has more Recession
colnames(obs.tbl) <- c("Expansion", "Recession")
rownames(obs.tbl) <- c("Democrat", "Republican")
exp.tbl <- outer(rowSums(obs.tbl), colSums(obs.tbl))/sum(obs.tbl)
colnames(exp.tbl) <- c("Expansion", "Recession")
rownames(exp.tbl) <- c("Democrat", "Republican")
obs.tbl ; exp.tbl

##
## Expansion Recession
## Democrat      3782      254
## Republican    3511     1311
##
## Expansion Recession
## Democrat   3322.934  713.0662
```

```

## Republican 3970.066 851.9338
chisq.test(DJI$Republican,DJI$Recession)

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: DJI$Republican and DJI$Recession
## X-squared = 657.98, df = 1, p-value < 2.2e-16

```

As we can see from this contingency table, Republicans had more days in office during recessions, but they also had more days in office during expansions. The result of the chi-square test is a p-value is less than 2.2e-16, far below our .05 threshold, so there would be a very small chance that the observed contingency table would arise by chance. Thus, the observations provide sufficient evidence to reject the null hypothesis that Republican and Democratic regimes are equally likely to be associated with recession years from 1985 to early 2020.

Let us try running this as chi-square test of contingency table including all regimes:

```

obs.tbl <- table(DJI$Recession, DJI$Regime); rownames(obs.tbl) <- c("Expansion", "Recession"); obs.tbl

##
##          BC   BO   DJT   GHWB   GWB   RR
## Expansion 2008 1774  742  505 1259 1005
## Recession   13  241   55  506  750    0
exp.tbl <- outer(rowSums(obs.tbl), colSums(obs.tbl))/sum(obs.tbl); rownames(exp.tbl) <- c("Expansion",
##          BC        BO        DJT        GHWB        GWB        RR
## Expansion 1663.9369 1658.997 656.1889 832.3801 1654.057 827.4402
## Recession  357.0631 356.003 140.8111 178.6199 354.943 177.5598
#This table allows us to see a breakdown of how long each president was in office in terms of recessions
#expansions
chisqvalue <- sum((obs.tbl - exp.tbl)^2/exp.tbl)

```

Here we can see that GWB had the most recession days. We can also see how long each president was in office in terms of recessions and how many recession days we would've expected, if they occurred evenly through the years.

```
P.Value <- pchisq(chisqvalue, df = (2 - 1) * (6 - 1), lower.tail = FALSE); P.Value
```

```
## [1] 0
```

And we get a p-value of zero, thus we reject the null hypothesis that recession years are equally likely to arise across regimes.

Lastly, we can run this analysis as chi-square test specific to each regime with p the observed probability of recession:

```
q <- 1 - p; q # 0.8233235 probability of not being in a recession
```

```
## [1] 0.8233235
```

```
prob <- (DJI$Recession*p + (!DJI$Recession)*q) / sum(DJI$Recession*p + (!DJI$Recession)*q)
min(prob) ; max(prob) ; sum(prob)
```

```
## [1] 2.812873e-05
```

```
## [1] 0.0001310817
```

```
## [1] 1
```

```

for (i in unique(DJI$Regime)) {
  print(chisq.test(DJI$Recession, DJI$Regime == i, p = prob))
}

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: DJI$Recession and DJI$Regime == i
## X-squared = 241.89, df = 1, p-value < 2.2e-16
##
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: DJI$Recession and DJI$Regime == i
## X-squared = 820.18, df = 1, p-value < 2.2e-16
##
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: DJI$Recession and DJI$Regime == i
## X-squared = 520.2, df = 1, p-value < 2.2e-16
##
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: DJI$Recession and DJI$Regime == i
## X-squared = 688.97, df = 1, p-value < 2.2e-16
##
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: DJI$Recession and DJI$Regime == i
## X-squared = 57.903, df = 1, p-value = 2.754e-14
##
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: DJI$Recession and DJI$Regime == i
## X-squared = 68.984, df = 1, p-value < 2.2e-16

```

Null hypothesis is that each regime has the observed probability p of recession across regimes. Note: There could exist carryover/lingering effects of recession or otherwise from one regime to the next. Each p-value is less than $2.2e-16$, far below the .05 threshold. This indicates that no individual regime is equally likely to be associated with recessions from the years 1985 to early 2020.

As you can see, fiscal and monetary policies seem to have an effect on the stock market. However, they are still subject to the volatility of the system and carry-over effects. To finish off, this raises questions about the independence of our random variables which is a topic for further research.