1 Introduction

In today's economy, financial asset bubbles are a trending topic linked with enthusiasm and interest. In keeping a watchful eye on recent market trends, it has been noted that there has been an escalation in the prices of Gold. Financial entrepreneurs and other interested parties are maintaining their posture, curios as to the future changes in these market prices. How are we able to detect or estimate the future changes of any asset, such as stock, gold, housing, or various commodities? How quickly will an asset price will jump, or fall? These are the questions which we will maintain our focus throughout this study, as we center our approach and define our reasonings. If a computer program existed that accurately estimated the stock market and gave a real time minute by minute update, this would benefit not only day traders and investors but this tool would increase the quality of life in every dynamic for anyone who consistently made a fortune in the stock market. In this paper we will study non-parametric estimator done by Florens Zmirou and Interpolation by cubic spline. Therefore, we will reveal how to determine whether any asset is experiencing a price bubble in real time.

How will we detect asset bubbles in real time? Our research problem will be in deciding whether an asset price is experiencing a price bubble in a finite, or an infinite time period.

We will be able to determine the volatility of asset prices. Using some helpful techniques to determine an asset bubble in real time will help financial corporations, banks, and money markets. They can lower their money damaging risks by using our methodology. According to "There is a bubble" paper paragraph 2, indeed, in 2009 the federal reserve chairman Ben bernanke said in congress testimony[1] "It is extraordinarily difficult in real time to know if an asset price is appropriate or not". Our goal is to estimate stock price volatility by way of the Florens Zmirou estimator, and then we will extrapolate the volatility tail in order to check the integral. We will then be able to determine whether the integral is finite or infinite. The process for bubble detection depends on a mathematical analysis that determines when an asset is undergoing speculative pricing, i.e., its market price is greater than its fundamental price. The difference between the market and the fundamental price, is a price bubble.

As stated above, we will use a nonparametric estimator Florens -Zmirou which is based on the local time of the diffusion process. The biggest challenge we have faced in using a non-parametric estimator, is that we can only estimate $\sigma(x)$ volatility function on the points which are visited by

the process. Only a finite number of data points are used which is a compact subset of R+, therefore, we are not able to estimate the tail of the volatility. However, by determining the tail of the volatility, we can then reveal whether the integral is finite or infinite. We do not, as of yet, know the asymptotic behavior of the volatility. Only finite data points are used which is a compact subset of $R+$. /therefore we will not able to estimate the tail of volatility.

After an estimation of the volatility function $\sigma(x)$, we will then interpolate the function using cubic splines and Reproducing Kernel Hilbert Spaces. Since a Hilbert Space is an inward item space that is finished and distinct regarding the standard deÞned by the internal item. The stock market is unpredictable, unstable and an insecure source to grow one's net worth however; despite posing a risk, the stock market still provides an avenue to increase one's net worth and increase value in assets. My program once debugged and completed will accurately estimate the volatility of stock and will interpolate the volatility function. We estimate the the volatility of asset prices using the Floren Zmirou estimator and then we will extrapolate the volatility tale in order to test if the prices exhibits a martin angle or submartinangle. If the volatility is large enough then a bubble exists. A non-parametric estimator Florens Zmirou, which is based on local tine of the diffusion process to estimate the volatility, function on the points which are visited by the process. The Divarication in parameter decisions ultimately decides if the regression capacity overÞts, underÞts, or Þts ideally. The association between characteristic spaces and smoothness is not self-evident, and subsequently, must be inspected utilizing a various methodologies. Characteristic feature spaces may further be utilized to compare or synchronize articles which have a much greater complex arrangement.

In chapter 2, we will present an overview of previous work, literature reviews, and alternate methodologies and their outcome measures. We will then present the background of the problem, and we will present how the problem is connected to finance and mathematics. We will further present the methods we have uncovered to solve the problem, complete with our best possible solution to the problem. In chapter 3, we will discuss the details of our algorithm, and we will further examine the probable and the extent of its implementation. In chapter 4, we will present several numerical examples and variables. We will finally present our conclusion and future work.

2 Previous Work

There are many methods which were reviewed by many researchers Since 1986.

2.1 Variance Bounds Tests of an Asset Pricing Model

The " Asset Price Volatility, Bubbles, and Process Switching" artile was published in 1986. This test will show that bubbles could in theory lead to excess volatility,but it shows that certain variance bounds tests preclude bubbles as an explanation.

2.2 A Panal Data Approach

In 2007,"Testing for Bubbles in Housing Markets" Vyacheslav Mikhed and Petrz Zemcik developed cross-sectionally robust panel data tests for unit roots and cointegration to find whether house prices reflect house related earnings.

2.3 Reproducing Kernal Hilbert Space

In 2011,"How to detect Asset Bubble" by Jarrow Robert, Kchia Younes, and Protter Philip used Reproducing kernal Hilbert Space and Optimization to detect asset Bubbles.

2.4 A monte Carlo Simulation

In 2011, "Bootstrapping Asset Price Bubbles" by Luciano Gutierrez used Monte Carlo Simulation.bootstrap procedures often provide better finite sample critical values for test-statistics than asymptotic theory does, bootstrap values are still approximations and are not exact. Davidson and Mackinnon (1999, 2006) show that it is possible to estimate the size and power of bootstrap tests by running Monte Carlo simulations, when the condition of asymptotic independence of the bootstrapped statistic and the bootstrap Data Generating Process (DGP) holds.

2.5 State Space Model With Markov-Switching

In 2011, "A Financial Engineering Approach to Identify Stock Market Bubble" by Guojin Chen and Chen Yan adopt a state space model with Markov-switching to identify the stock market bubbles both in China and US. Theoretical Background Let's consider the Stochastic Differential Equation:

$$dS_t = \mu(S_t)dt + \sigma(S_t)dW_t S_0 = s_0 \tag{1}$$

In this chapter, we will focus on numerical methods of Stochastic Differential Equations (SDE). SDE's are used in biology, physics, mathematics and ofcourse finance. The basic knowledge of SDE's comes from probability, random variables, variance and Stochastic Process. These useful

keywords are related to numberical methods of SDE's. Our problem focuses on finance, SDE's are used to model asset price with Brownian motion.

For the numerical solutions we will assume drift μ and diffusion σ coeficients are constants. Now Euler Muruyama method can be used to model asset price. By studying numberical solutions of SDE's, we will have better understanding toward the theory of SDE. SDE's are combined with continous Brownian Motion with different methods such as: Euler -Maruyama method, strong and weak convergence , milstein method. Let's start with finance background.

- What is Asset Bubble?
- How does one determine if an asset price is experiencing a bubble?

These are the common questions which will raise in every mind. Our methodology will clearly answer these questions.

3 Definations

[Asset Bubble] *Suppose the market price of an asset increases significantly. How can one determine if the market price is inflated above the actual price of an asset? This price behavior is know as a bubble.* [Market Price] *The current price of an asset.* [Fundamental Price] *The actual value of an asset based on an underlying perception of its true value.* [Risk] *Variance of return on an asset.* [Portfolio] *Set of Assets.* [Asset Bubble] *The difference between the market and fundamental price, if any, is a price bubble.* [Volatility] *Rate at which the price of security moves up and down.* Our work is a combination of finance and mathematics. From here we will introduce mathematical definations which are the connection between price to bubble. [Probability Space] *$(\Omega, \mathcal{F}, P)$ where Ω is a set (sample space), $\mathcal{F}$ is a sigma algebra of subsets (events) of Ω, and P is a Probability Measure.* [Random Variable] *Measurable functions of real analysis $X : \Omega \mapsto \mathcal{R}$ map $X : (\Omega, \mathcal{F}) \mapsto (\mathcal{R}, \mathcal{B})$ and X is random variable if*

$$X^{-1}(A)\epsilon\mathcal{F}, \forall A\epsilon\mathcal{B}$$

where $X^{-1}(A) := \omega\epsilon\Omega \mid X(\omega)\epsilon A$

- We treat the asset price as a stochastic process.

[Stochastic Process] *Given a probability space* $(\Omega, \mathcal{F}, P)$*, a stochastic process with state space* X *is a collection of* X*-valued random variables,* S_t*, on* Ω *indexed by a set* T *(e.g. time).*

$$S = \{S_t : t \in T\} \tag{2}$$

One can think of S_t as a asset price at time t. [Stochastic Differential Equation] *A differential equation with one or more terms is a stochastic process.* [Brownian Motion] $\{S_t : 0 \leq t \leq T\}$*:*

$$\begin{aligned} dS_t &= \sigma(S_t)dW_t + \mu(S_t)dt \\ S_0 &= 0 \end{aligned} \tag{3}$$

- W_t denotes the standard Brownian Motion.
- $\mu(S_t)$ called the drift coefficient.
- $\sigma(S_t)$ called the volatility coefficient.

[Brownian Motion] *A continuous-time stochastic process* $\{S_t : 0 \leq t \leq T\}$ *is called a Standard Brownian Motion on* $[0, T]$ *if it has the following four properties:*

(i) $S_0 = 0$

(ii) The increment of S_t *are independent; given*

$$0 \leq t_1 < t_2 < t_3 < \cdots < t_n \leq T$$

the random variables $(S_{t_2} - S_{t_1})$*,* $(S_{t_3} - S_{t_2})$*,* $\cdots$*,* $\left(S_{t_n} - S_{t_{n-1}}\right)$ *are independent.*

(iii) $(S_t - S_s)$*,* $0 \leq s \leq t \leq T$ *has the Gaussian distribution with mean zero and variance* $(t - s)$

(iv) $S_t(W)$ *is a continuous function of* t*, where* $W \in \Omega$*.*

[Martingales]

(a) $E[\mid S_n \mid] < +\infty$*, for all* n*.*

(b) S_n *is said to be adapted if and only if* S_n *is* $\mathcal{F}_n$*-measurable.*

The stochastic process $S = \{S_n\}_{n=0}^{\infty}$ *is a martingale with respect to* $(\{\mathcal{F}_n\}, P)$ *if* $E[S_{n+1} \mid \mathcal{F}_n] = S_n$*, for all* n*, almost surely and:*

- *S satifies (a) and (b).*

[Supermartingale] *The stochastic process* $S = \{S_n\}_{n=0}^{\infty}$ *is a supermartingale with respect to* $(\{\mathcal{F}_n\}, P)$ *if* $E[S_{n+1} \mid \mathcal{F}_n] \leq S_n$*, for all* n*, almost surely and:*

- *S satifies (a) and (b).*

[Local Martingale] *If* $\{S_n\}$ *is adapted to the filtration* $\{\mathcal{F}_n\}$*, for all* $0 \leq t \leq \infty$*, then* $\{S_n : 0 \leq t \leq \infty\}$ *is called a local martingle provided that there is nondecreasing sequence* $\{\tau_k\}$ *of stopping times with the property that* $\tau_k \to \infty$ *with probability one as* $k \to \infty$ *and such that for each* k*, the process defined by*

$$S_t^{(k)} = S_{t \wedge \tau_k} - S_0$$

for $t \in [0, \infty)$ *is a martingale with respect to the filtration*

$$\{\mathcal{F}_n : 0 \leq t < \infty\}$$

- Remark A strict local martingale is a non-negative local martingale.

If for any strict local martingale

$$\{S_t : 0 \leq t \leq T\}$$

with $E\left[|S_0|\right] < \infty$ is also a supermartingale and $E\left[S_T\right] = E\left[S_0\right]$, then $\{S_t : 0 \leq t \leq T\}$ is in fact a martingale. Remarks

- $\{S_t : 0 \leq t \leq T\}$ is a supermartingale and a martingale if and only if it has constant expectation.

- For a strict local martingale its expectation decreases with time.

Connection between martingale and bubble as follows:

$$\{S_t : 0 \leq t \leq T\}$$

is a strict local martingale if and only if

$$\int_{\alpha}^{\infty} \frac{x}{\sigma^2(x)} dx < \infty \tag{4}$$

for all $\alpha > 0$.

- A bubble exists if and only if (4) is finite.
- We shall call (4) the volatility of asset return.
- In this scope, the difference between a martingale and a strict local martingale is whether the volatility of asset return, (4) is finite or not finite.

4 Numerical Methods of SDE

For $t \in [0, T]$ equation (3) can be represented in an integral form in the following way:

$$\begin{aligned} dS_t &= \sigma(t)dW_t + \mu(t)dt \\ \int_0^t dS_t &= \int_0^t \sigma\left(S_t\right) dW_t + \int_0^t \underbrace{\mu\left(S_t\right)}_{\in \mathcal{R}^+} dt \\ S_t - S_0 &= \int_0^t \sigma\left(S_t\right) dW_t + \left(\underbrace{\mu\left(S_t\right) \cdot t}_{x_0} - \mu\left(S_t\right) \cdot 0 \right) \end{aligned}$$

$$S_t = x_0 + \int_0^t \sigma\left(S_t\right) dW_t \tag{5}$$

(1.5) is the price model which will be used in SDE's and Floren Zmirou Estimator

4.1 The Euler-Maruyama Method

SDE can be written into integral form as:

$$S_t = S_0 + \int_0^t f\left(S_s\right) ds + \int_0^t g\left(S_s\right) dW(s) \qquad t \in [0, T] \tag{6}$$

f, g are scalar function with S_0 is a random variable.

$$\begin{cases} dS_t = \mu(S_t)dt + \sigma(S_t)dW_t \\ S(0) = S_0 \end{cases}$$

Computing solution by Euler Maruyama method as follows:

$$w_0 = S_0$$

$$w_{i+1} = w_i + \mu(t_i, w_i) \triangle t_{i+1} + \sigma(t_i, w_i) \triangle W_{i+1}$$

$$w_{i+1} = w_i + \mu w_i \triangle t_i + \sigma w_i \triangle W_i$$

$$\triangle t_{i+1} = t_{i+1} - t_i$$

$$\triangle W_{i+1} = W(t_{i+1} - W(t_i)$$

The crucial question is how to model the Brownian motion $\triangle W_i$. Define N(0,1) to be the standard random variable that is normally distributed with mean 0 and standard deviation 1. Now drift coefficient μ and diffusion coefficient σ are constants, the SDE has an exact solution:

$$S(t) = S_0 \cdot Exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma W(t)\right) \tag{7}$$

For an example, we use the Euler-Maruyama Approximation Method on the SDE where the constants $\mu = 2$, $\sigma = 1$, and $S_0 = 1$ are given.

Put figure

There are other methods such as Strong and weak convergence of the Euler Muruyama method, Milstein's Higher Order Method, Linear Stability and Stochastic Chain Rule are also used for numerical solutions for SDE.

- What will happen if σ is not constant. Our problem will focus on the scenario when σ values will be approximate with non parametric estimator Floren Zmirou method on local real time.

5 Methodology

In this classical setting, Jarrow, Protter, and Shimbo [19], [20] show that there are three types of asset price bubbles possible. Two of these price bubbles exist only in infinite horizon economies, the thirdÑcalled type 3 bubblesÑexist in finite horizon settings. Consequently, type 3 bubbles are those most relevant to actual market experiences. For this type of bubble, saying whether or not a bubble exists amounts to determining whether the price process under a risk neutral measure is a martingale or a strict local martingale: if it is a strict local martingale, there is a bubble. Stock price is strict local martingale if and only if

$$\int_{\alpha}^{\infty} \frac{x}{\sigma\,(x)} dx < \infty \forall \alpha > 0 \tag{8}$$

Floren Zmirou's non parametric estimator is based on the local time of the Diffusion Process. [Diffusion Process] *In probability theory, a branch of mathematics, a diffusion process is a solution to a stochastic differential equation. It is a continuous-time Markov process with almost surely continuous sample paths.*

5.1 Floren Zmirou

Lets consider following equation:

$$S_t = S_0 + \int_0^t \sigma\left(S_t\right) dW_t \tag{9}$$

In Floren Zmirou, the drift coefficient $\mu(S_t)$ is null which is ignored without loss of generality. It is not involved in our problem. $\sigma(S_t)$ the volatility coefficient is unknown. We will follow steps for Floren Zmirou method:

- $(S_{t_1}, \ldots, S_{t_n})$ are the stock prices in the interval $t_1, \ldots, t_n \in [0, T]$
- Without loss of generality, we assume $T = 1$, therefore $t_i = \frac{i}{n}$

Estimator as follows:

$$l_T(x) = lim_{\epsilon \longrightarrow 0} \frac{1}{2\epsilon} \int_0^T 1_{\{|S_s - x| < \epsilon\}} d\langle S, S\rangle_s$$

where $d\langle S, S\rangle_s = \sigma^2(S_s)$

$$L_T(x) = lim_{\epsilon \longrightarrow 0} \frac{1}{2\epsilon} \int_0^T 1_{\{|S_s - x| < \epsilon\}} dS$$

$$\implies l_T(x) = \sigma^2(S_s) L_T(x)$$

$$\implies \frac{l_T(x)}{L_T(x)} = \sigma^2(S_s)$$

Let's assume that $nh_n \to \infty$ and $h_n \to 0$

$$L_T^n(x) = \frac{T}{2nh_n} \Sigma_{i=1}^n 1_{\{|S_{t_i} - x| < h_n\}}$$

$$l_T^n(x) = \frac{T}{2nh_n} \Sigma_{i=1}^n 1_{\{|S_{t_i} - x| < h_n\}} n(S_{t_i+1} - S_{t_i})^2$$

estimator of $\sigma^2(x)$ as follows:

$$S_n(x) = \frac{\Sigma_{i=1}^n 1_{\{|S_{t_i} - x| < h_n\}} n(S_{t_i+1} - S_{t_i})^2}{\Sigma_{i=1}^n 1_{\{|S_{t_i} - x| < h_n\}}} \tag{10}$$

If σ is bounded above and below from zero, has three continuous and bounded derivatives, and if $(h_n)_n{}^3$1satisfies $nh_n \to \infty$ and $nh_4n \to 0$, then $S_n(x)$ is a consistent estimator of $\sigma^2(x)$.
In Theorem 5.1 we say XYZ.... If $nh_n \to 0$, then $\sqrt{N_x^n}((\frac{S_n(x)}{\sigma^2(x)}) - 1)$ converges in distribution to $\sqrt{2}Z$ where Z is a standard normal ransom variable and $N_x^n = 1_{\{|S_{ti}-x|<hn\}}$ Floren Zmirou estimator requires a grid step which is $h_n \to 0$ such that $nh_n \to \infty$ and $nh_n^4 \to 0$. We will choose step size $h_n = \frac{1}{n^{\frac{1}{3}}}$ so that all the conditions will be satisfied.

5.2 Interpolation

Interpolation is a method of constructing new data points within the range of a discrete set of known data points. There are many methods to do interpolation. For example Linear Interpolation,polynomial interpolation, piecewise constant interpolation, spline interpolation. Here, we will interpolate an estimate of $\sigma^2(x_i)$ where $i \in [1.M]$ within the bounded finite interval D where we have observations. We used cubic spline interpolation and Reproducing Kernal Hilbert Spaces to get interpolation function.

5.2.1 Cubic Spline Interpolation

- **Piecewise-polynomial approximation** : An alternative approach is to divide the interval into a collection of subintervals and construct a different approximating polynomial on each subinterval. Approximation by functions of this type is called piecewise-polynomial approximation.

- The most common piecewise-polynomial approximation uses cubic polynomial between each sucessive pair of nodes and is called **cubic spline interpolation**.

- **Cubic Spline Interpolation**

  Given a function f defined on [a,b] and a set of nodes $a = x_o < x_1 < \ldots < x_n = b$ a **cubic spline interpolant S** for f is a function that satisfies the following conditions:

  a) $S(x)$ is a cubic polynomial, denoted $S_j(x)$ on the subinterval $[x_j, x_{j+1}]$ for each $j = 0, 1, 2, \ldots, n-1$;

  b) $S_j(x_j) = f(x_j)$ and $S_j(x_{j+1}) = f(x_{j+1})$ for each $j = 0, 1, 2, \ldots, n-1$;

  c) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ for each $j = 0, 1, 2, \ldots, n-2$;

  d) $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ for each $j = 0, 1, 2, \ldots, n-2$;

  e) $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$ for each $j = 0, 1, 2, \ldots, n-2$;

f) One of the following sets of boundry conditions is satisfied;

i) $S''(x_0) = S''x_n) = 0$;

ii) $S'(x_0) = f'x_0)$ and $S'(x_n) = f'x_n)$

- When the free boundry conditions occur, the spline is called **natural spline**.
- Interpolation is seen as inverse problem.
- We will have two types of solutions for inverse problem.
- Normal Solution It will allows an exact interpolation with minimal squared norm.
- Regularized Solution it will yields quasi interpolative results, accompained by an error bound analysis with Tikhonov Regularization produces an approximate solution f_α which belongs to $H(D)$and that can be obtained via the minimization of the regularization functional.

Regularized solution with Kernal function For interpolation, we denote kernal function $K_{n,\tau}^{a,b}$ for n =1 and n = 2. of $H^n(a,d)$ where $D = (a,b)$ Now one we have interpolation function and extended form of the estimated $\sigma^2(x)$ call this extended function $\sigma^b(x)$, we can decide if there is need of extrapolation.

- if the volatility $\sigma^2(x_i)$ doesn not diverge to ∞ when $x \to \infty$ and it remains bounded on $\mathcal{R}^+$. - No extrapolation is required. - $\int_\epsilon^\infty \frac{x}{\sigma^2(x)}$ is infinite. - The process is true martingale.
- If the volatility diverges to ∞ when $x \to \infty$ then we will extrapolate.

5.3 Extrapolation ,Optimization and Minimization

Now since we have extended function $\sigma^b(x)$, we will extrapolate this over $\mathcal{R}^+$.
Let's consider following optimization with minimazation problem:

$$\bar{m} = \underset{m \geq 0}{\operatorname{argmin}} \sqrt{\int_{[a,\infty \cap D]} |\sigma_m - \sigma^b|^2 dS} \tag{11}$$

- σ_m will interpolate the input data points when n = 2.
- $\sigma_{\bar{m}}$ has the asymptotic behavior that beast matches our function on the estimation interval. We will construct extrapolation by choosing the asymptotic weighting function parameter m such that $f_m = \frac{1}{\sigma^2 m}$.

- a = max S - $\frac{1}{3}$(maxS-min S)
- $H_{2,m}$ allows the best interpolation of M which is estimated points such that the extrapolation function remains as close as possible to $\sigma^b(x)$.
- We will plot function with different asymptotic weighting parameter m which is obtained from RKHS extrapolation method.
- The asymptotic weighting function's parameter $\bar{m}$ obtained by optimization and minimization is the most consistent which exactly match the input data within all the function in $H_{1,m}$

Implementation of an Asset Bubble In this chapter. we will provide valueable information about the methods used in our problem. We used sage python to solve this problem. We orgnize data and process in four classes. Each class is designed to store the information about Stock, Floren Zmirou ,Cubic Spline,and Run.

6 Stock

Let's consider following questions:

- What kind of Data will be used?
- How can someone obtain accurate historic data information?
- Will it be Mintue to mintue data information?
- How we will store and use data information?
- Can we get data information for any stock symbol in market?

Following Stock Class which will provide answers to all the above questions.

6.1 Stock Class

We used Google Finance and CSV file from Yahoo Finance methods to download stock data information. There are GetGoogleData, GetStockPrice and init are the four algorithms used in Stock Class which will decribe the process.

- S Stock Price.
- Smax Maximum Stock Price.

- Smin Minimum Stock Price.
- Row of Stock prices is numberical string.
- Ticker is Ticker symbol name.
- D Number of days.
- T time in seconds in intergers.

We will read, download and save stock data by following methods.

6.1.1 Google Finance Stock Prices

1. Get Google Data: This function obtains mintue to mintue stock prices in numberical string for any ticker from Google Finance and save in array.

Algorithm 1 Get Google Data

INPUT: Ticker, D ,T
OUTPUT: $S = (S_{t_1} \dots S_{t_n})$ in [0,T] where $t_i = \frac{i}{n}T$.

if Ticker length ≤ 3 **then**
    exchange Ticker with New York Stock Exchange (NYSE).
else
    exchange Ticker with Natinal Association of Securities Dealers (NASD)
end if
Open Google Finance link.
DataList = reads each line from Google Finance link.
TickerData = list of array of DataList.
Put stock Prices in a list.
for MinuteData in TickerData **do**
    Seperate MintueData with commas and store in a list S.
    Append S as float and save it.
return S = Stock prices

6.1.2 Yahoo Finance Stock Prices

2. Get Yahoo Data : This function obtains the stock prices which are downloaded from yahoo finance and stored in csv file by filename. It will provide the yahoo API based minute to minute data.

Algorithm 2 Get Yahoo Data

INPUT: Filename
OUTPUT: S from CSV file.

```
Open Yahoo Finace link.
Open and read the csv Filename as universal.
Skip the header.
Create a empty list.
for row in csv Filename do
    if the second row is numberical string then
        Add second row to empty list
Name this row S.
return S
```

6.1.3 Choosing either Google Ticker Or Yahoo CSV file

3. Class Constructor: This function will analyzie wheather to use csv filename for Yahoo or Ticker for Google finance.

Algorithm 3 –init–

INPUT: User Keyword
OUTPUT: S

```
if Filename is in keyward Usage then
    Obtain Stock Prices from Get Stock Prices. Ticker is used in keyward Usage
    Obtain Stock Prices from Get Google Data.
else Bad parameters
return S
```

6.2 Example

7 Floren Zmirou

This class will implement the following equation:

$$S_n(x) = \frac{\Sigma_{i=1}^{n} 1_{\{|S_{t_i} - x| < h_n\}} n(S_{t_i+1} - S_{t_i})^2}{\Sigma_{i=1}^{n} 1_{\{|S_{t_i} - x| < h_n\}}} \tag{12}$$

Now we have Stock Prices from Stock Class. Our goal is to get $\sigma(x)$ values from list of Stock Prices S. We will import sage and stock class to Floren Zmirou class. In order to use Floren Zmirou Estimator, we will need following:

- n = Number of stock prices.
- T = Time in seconds
- hn = $1/(n)^(1/3)$
- x-stepsize
- x values
- We will implement hn, x-stepsize, and x values so we can use the terms in Floren zmirou estimator equation (1.1).

7.1 Floren Zmirou Class

Algorithm 4 Derive hn

INPUT: Stock Prices S
OUTPUT: hn

```
1: n = length of Stock Prices S.
2: hn = 1.0/n**(1.0/3.0)
3: return hn =0
```

4. (x step size): This function will be used to create step size to generate x.

Algorithm 5 x_hn = x step size

INPUT: Stock Prices S
OUTPUT: x_hn

```
Double hn = 2 **hn
Difference = max S-min S
x_hn = Difference * Double hn
return x_hn
```

Algorithm 6 x Values

INPUT: Stock Prices S
OUTPUT: x

```
half h_n = x_hn /2.0
x = empty list.
Append x with Smin + half h_n
FirstElement = first element of x's list.
while FirstElement < Smax do
    FirstElement = FirstElement + x_hn
end while
Append FirstElement into x's list.
return x
```

- Now we will implement floren Zmirou Estimator. Floren Zmirou has Sublocal time, Local time, volatility Estimator and Indicator function.

5. Sublocal Time:This function will give $L_T^n(x) = (\frac{T}{2nhn}) \sum_{i=1}^{n} 1_{|S_{t_i} - x| < h_n}$

Algorithm 7 Sublocal Time

INPUT: T, S, x, n, hn.
OUTPUT: $L_T^n(x) = (\frac{T}{2nhn}) \sum_{i=1}^{n} 1_{|S_{t_i} - x|}$

```
sum = 0.0
scalar = T/(2.0*n*hn)
for i in range of the length of Stock Prices do
    Sti = i th element of the list of Stock prices
    absoluteValue = abs(Sti-x)
    indicatorValue to pass through indicator function
    if absoluteValue is less than hn then
        sum = sum+indicatorValue
return scalar*sum
```

6. Local Time: $L_T^n(x) = (\frac{T}{2nh_n}) \sum_{i=1}^{n} 1_{|S_{t_i} - x| < hn} * n(S(t(i+1)) - S(t(i))^2$

7. Volatility Estimator Sn(x): Volatility Estimator is $\sigma^2(x)$. This function will give Local Time/Sublocal Time.

Algorithm 8 Local Time

INPUT: T, S, x, n, hn.
OUTPUT:

```
sum = 0.0
scalar = T/(2.0*n*hn)
for i in range of the length of Stock prices - 1 do
    Sti = ith element of the list of Stock prices
    Stj = jth element of the list of Stock prices
    absoluteValue = abs(Sti-x)
    Difference = (Stj-Sti)**2
    if absoluteValue is less than hn then
        sum = sum+indicatorValue*n*Difference
return scalar*sum
```

Algorithm 9 Volatility Estimator

INPUT: T, S, x, n, hn.
OUTPUT:

```
Ratio = Local Time / Sublocal Time.
return Ratio =0
```

- We have $\sigma^2(x)$ for Stock Prices. Now we want to see how many stock prices are in each grid point. If there are less than 0 or 1 percent of stock prices in each grid then we will exclude that grid point from our calculation process.

8. (DoGridAnalysis): This function give us the list of usable grid points and for each usable grid point, the list of Stock prices.

8 Natural Cubic Spline

- To construct the cubic spline interpolant S for the function f, defined at the numbers $x_0 < x_1 < \ldots < x_n$ satisfying $S''(x_0) = S''(x_n) = 0$:

(Natural Cubic Spline)

9 Run the Program

References

[1] Philip Protter Robert A. Jarrow, Younes Kchia. How to detect an asset bubble. *Journal of Computing*, 2011.

Algorithm 10 DoGridAnalysis

INPUT: T,S,x,n,hn.
OUTPUT:

x = useableGridPoints.
d = empty dictionary
Si = length of stock prices.
for grid Points in x **do**
    create a dictionary where the keys are the grid points and the value is a list for each grid point
    for stockPrice in S **do**
        if |grid Points - stockPriceCount|≤ hn **then**
            Add x value to corresponding Si
            for grid Points in x **do**
                if if the list of data points corresponding to x values than Y percent of total grid points **then**
                    add it to the list of usable grid points
                    L = dictionary with key values of grid points.
                    N = Length of L in float.
                    Percent of Stock Prices = N/S
                    if Percent of Stock prices < Y **then**
                        Remove grid points from x
                        Delete L
return x = usable grid points and d

Algorithm 11 Get Grid Variance

INPUT: S,x_hn, x values
OUTPUT: Floren Zmirou estimation over usable grid points.

Points = Empty list.
half x_hn = x step size*(Stock Prices)/2.0
for x in usable grid points **do**
    y =Sn(x)**2
    Append (x,y)
return (x,y)

Algorithm 12 GetGridInverseStandardDeviation

INPUT:S,x_hn,x values.
OUTPUT: Standard Deviation of usable grid points.

Points = Empty list
for x in usable grid points **do**
    y = 1/sqrt(Sn(x))**2
    Append (x,y) in Points's list
return (x,y)

Algorithm 13 GetGridSigma

INPUT:S,x_hn,x values.
OUTPUT: Standard Deviation of usable grid points.

Points = Empty list
for x in usable grid points **do**
    y = Sn(x)
    Append (x,y) in Points's list
return (x,y)

Algorithm 14 Natural Cubic Spline

INPUT: $n, x_0, x_1, \ldots, x_n; a_0 = f(x_0), a_1 = f(x1), \ldots, a_n = f(x_n)$
OUTPUT: a_j, b_j, c_j, d_j for $j = 0, 1, \ldots, n-1$.
(note: $S_(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$ for $x_j \leq x \leq x_{j+1}$)

for $i = 0, 1, \ldots, n-1$ **do**
    $h_i = x_{i+1} - x_i$
end for
for $i = 1, 2, \ldots, n-1$ **do**:
    $\alpha_i = \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1})$
end for
Set $l_0 = 1$;
$\mu = 0$;
$z_0 = 0$.
for $i = 1, 2, \ldots, n-1$ **do**
    $l_1 = 2(x_{i+1} - x_{i-1}) - h_{i-1}\mu_{i-1}$;
$\mu_1 = \frac{h_1}{l_1}$;
$z_1 = \frac{(\alpha_1 - h_{i-1}z_{i-1})}{l_1}$.
Set $l_n = 1$;
$z_n = 0$;
$c_n = 0$.
    for $j = n-1, n-2, \ldots, 0$ **do**
        $c_j = z_j - \mu_j c_{j+1}$;
        $b_j = (a_{j+1} - a_j)/h_j - h_j(c_{j+1} + 2c_j)/3$;
        $d_j = (c_{j+1} - c_j)/(3h_j)$
Return $(a_j, b_j, c_j, d_j for j = 0, 1, \ldots, n-1)$;