The following is from Chapter 2 of my book *Expert Trading Systems: Modeling Financial Markets with Kernel Regression.* The book was published by Wiley in 2000.

Chapter 2: Data Modeling of Time Series

2.1 The Time Series Problem

Time series present some unique modeling problems. When dealing with time series the analyst should ask the following questions:

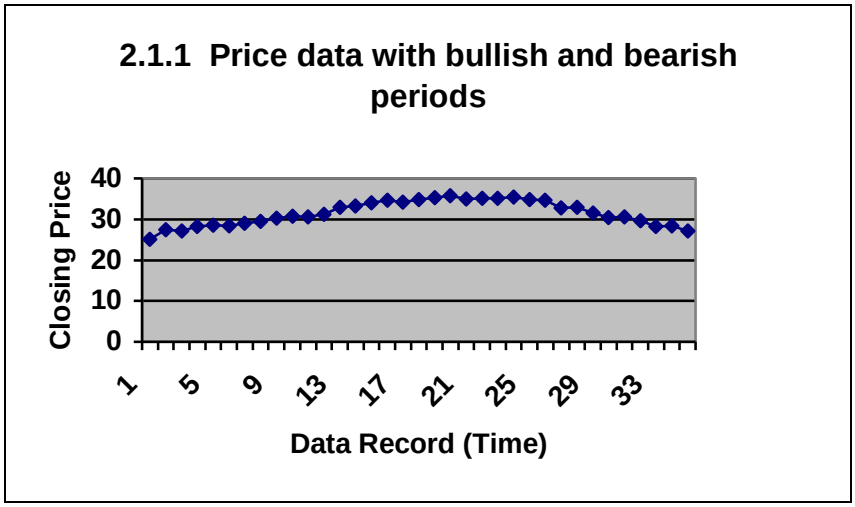
1) Is there enough data available to develop and test a model?
2) For the proposed candidate predictors does the available data adequately populate the various spaces?
3) Is serial correlation a major problem?
4) Does the model hold up over time? (In other words, can we be reasonably assured that the model itself doesn't change with time? Or if it does change, is the change gradual?)

1) Is there enough data available? When modeling financial markets, the availability of data is dependent upon the time scale of the data. For example, if one is using daily data, then there are only about 250 data points per year. Ten years of data would thus include about 2500 points but then question number (4) becomes relevant: Does the market behavior ten years ago have any relevance today? One must develop and use testing procedures to ascertain the relevance of models. If the modeling is based upon a much more rapid time scale (say 5 minute time periods), then much more data is available per day. For example, for markets that open at 8:30 A.M. and close at 16 P.M., there are 90 five minute time periods per day. So in one year there are approximately 90*250 (i.e., 22500) available data points. Clearly it is much easier to develop models when there is a lot more available data.

One method for increasing the amount of available data is to include data from a group of financial series rather than just a single series. For example consider a common stock database in which daily prices are available. Rather than trying to model the fractional price changes for one particular stock, it is much easier to try to model a group of similar stocks (for example, from a particular industrial group). An application in which monthly data from over 7000 different stocks are analyzed as a group is discussed in Chapters 6 and 7.

2) Does the data adequately populate the spaces under consideration? Coverage of the space is indeed an important concept in modeling. When modeling financial markets one first must make sure that there is adequate representation of the different types of markets in both the data used to develop the model and the data used to test the model. For example, consider a situation in which the modeling data is taken only

from time periods which could be described as "bullish" (i.e., the price changes of the financial instrument of interest are generally positive). If the testing of the model is based upon data from "bearish" periods (i.e., falling prices), then one should not be surprised if the testing yields disappointing results. In Figure 2.1.1 the trend is essentially *bullish* up to record 20. From about record 25 the trend becomes *bearish*. Clearly it is preferable to use data which is representative of the different types of market conditions that one is likely to encounter.



The same argument holds true for the independent variables. For example consider an indicator *x* which is some sort of oscillator about zero. If the values of *x* are mostly positive during the modeling time periods, and then mostly negative during the testing time periods, once again one might expect poor results. Clearly it is useful to devise statistical tests to measure the similarity of the modeling and testing data.

3) Is serial correlation a problem? Serial correlation is a problem unique to time series. Serial correlation relates to the independence of adjoining points. Are the adjoining points independent or are they somehow related? We can define ***r*** (the correlation coefficient) between two variables (lets say ***x*** and ***y***) as follows:

$$r = \frac{SS_{xy}}{\sqrt{SS_{xx} SS_{yy}}} \qquad \textbf{(2.1.1)}$$

where:

$$SS_{xy} = \sum_{i=1}^{i=n} \left(x_i - \bar{x}\right) * \left(y_i - \bar{y}\right) \qquad \textbf{(2.1.2)}$$

$$SS_{xx} = \sum \left(x_i - \bar{x}\right)^2 \qquad \textbf{(2.1.3)}$$

$$SS_{yy} = \sum_{i=1}^{i=n} \left(y_i - \bar{y}\right)^2 \qquad \textbf{(2.1.4)}$$

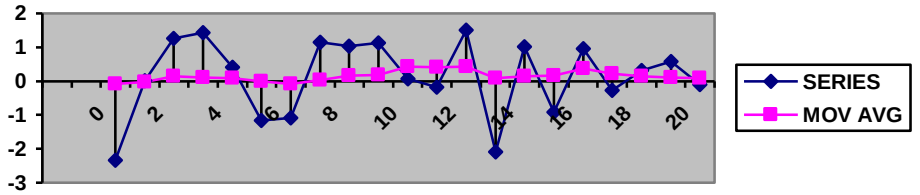
The notation ***SS*** is used for "sum of the squares". Consider the case where *x* is a candidate predictor and ***y*** is the same predictor lagged by one time period. Serial correlation implies that the value of ***r*** is significantly larger than 0. To illustrate this point assume that ***x*** and ***y*** are defined as follows:

X **= MA(SERIES,10)**
(2.1.5)

Y **= LAG(MA(SERIES,10), 1)**
(2.1.6)

In the Figure 2.1.2, 21 values of SERIES and ***X*** (10 day Moving Average) computed from SERIES are shown. The serial correlation of SERIES is –0.175 which is fairly close to zero, while the serial correlation of ***X*** is 0.623.

Figure 2.1.2 - A time series and its moving average



The high degree of correlation between adjoining values of ***X*** should not be surprising. The values of ***Y*** (i.e., ***X*** lagged by one time period) are determined from almost the same 10 points as the values of ***X***. What does this imply? Since the values of ***X*** are serially correlated, we really don't have as much independent data as we might expect from just considering the number of data points. Serial correlation is not necessarily a major problem. It is merely a fact-of-life when modeling time series data and it should be considered when developing a modeling strategy.

The Durbin-Watson test can be used to determine if the serial correlation of a time series is significant [Du71]. Descriptions of this test and Durbin Watson tables are included in many texts [e.g., Mc94]. A ***d*** statistic is computed which ranges from 0 to 4:

$$d = \frac{\sum_{t=2}^{n} (x_t - x_{t-1})^2}{\sum_{i=1}^{n} x_t^{\,2}}$$

(2.17)

.If the series does not exhibit serial correlation then ***d*** is approximately 2. If the values are highly positively correlated, then ***d*** is close to 0 and if they are highly negatively correlated than ***d*** is close to 4.

4) Does the model hold up over time? The *persistency* of models is an important concept when using the models to make future predictions. In financial markets one can discover many examples of models that work quite well up to a certain point in

time and then cease to perform acceptably. The user is then faced with trying to decide if the failure is temporary or is due to some underlying fundamental change in the market behavior. . There is no simple answer to this question. However, it can be said that this problem is much more relevant for models developed using data from a longer time period. For example, models based upon daily data are much more likely to become obsolete than models based upon 5 minute bar data. The most obvious check for any model's persistency is to save the most recent data for final testing. If the model holds up at this point, then at least one can start using it with some degree of certainly that it is still a useful tool.

2.2 Classical Methods of Time Series Modeling

The history of time series analysis predates the computer age. H. Wold wrote a book in 1938 which summarized the knowledge of the subject to that point in time: H.Wold, *A Study in the Analysis of Stationary Time Series*, Almqvist and Wiksell, Stockholm, 1938. Another early book on the subject was written by the well-known cyberneticist Norbert Weiner (N. Weiner, *The Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications*, Wiley, 1949). Wold and Weiner refer to the contributions of G. U. Yule, A, Khinchine and A. Kolmogoroff to time series analysis. In the introduction to a fairly recent book edited by A. S. Weigend and N. A. Gershenfeld (A. S. Weigend and N. A. Gershenfeld,: *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison Wesley, 1994), the authors reflect upon the early work of Yule: "The beginning of 'modern' time series prediction might be set in 1927 when Yule invented the autoregressive technique in order to predict the annual number of sunspots. His model predicted the next value as a weighted sum of previous observations of the series…For the half century following Yule, the reigning paradigm remained that of linear models driven by noise."

A summary of classical time series analysis is included in the introduction of a well-known book by M. B. Priestley: *Non-Linear and Non-Stationary Time Series Analysis*, Harcourt Brace, 1988. "During the past fifty years or so, time series analysis has become a highly developed subject, and there are now well-established methods for fitting a wide range of models to time series data – as described in the books by Anderson [1971], Box and Jenkins [1970], Brillinger [1975], Chatfield [1975], Hannon [1970], Jenkins and Watts [1968], Koopmans [1975] and Priestley [1981]. However, virtually all the established methods rest on two fundamental assumptions; namely that (i) the series is *stationary* (or can be reduced to stationarity by some transformation, such as differencing), and (ii) the series conforms to a *linear* model. Assumption (i) means, in effect, that the main statistical properties of the series remain constant over time, and (ii) means that the values of the observed series can be represented as linear combinations of present and past values of a 'strictly random' (or 'independent') series. Needless to say, both of these assumptions are mathematical idealizations which, in some cases, may be valid only as approximations to the real situation. In practical applications the most one could hope for is that, for example, over the observed time interval the series would not depart 'too far' from stationarity for the results to be invalid."

One of the most well-known texts on the subject of time series analysis is the book by Box and Jenkins (*Time Series Analysis: forecasting and control*, Holden Day, Revised edition, 1976). The original version of this book was written in 1969 and reflected the growing power of computers for solving time series problems. The widespread availability of computers influenced many people to apply computers to the modeling of time series data. It was obvious to most analysts that the ability to predict the future based upon past history could be incredibly valuable. The techniques popularized by Box and Jenkins were applied to many areas of activity. Analysts on Wall Street were among the early users of these techniques as they attempted to make predictions related to the financial markets.

The classical approach was influenced by the computing power of the available machines. Most of the emphasis at that time was to consider only the time series itself as the data source. The two most popular model classes considered then were:

1) 1)Linear Stationary Models called ARMA models (for Auto Regressive Moving Average).

2) 2)Linear Non-stationary Models called ARIMA models (for Auto Regressive Integrated Moving Average).

The ARMA models are based upon an assumption that the time series is generated by a *linear aggregation of random shocks*. Since the random shocks may be positive or negative, we would expect the time series to vary about some mean value. The word *stationary* refers to the mean value of the time series. The basic equation for this class of models is:

$$z_t = a_t + \psi_1 * a_{t-1} + \psi_2 * a_{t-2} + \ldots \qquad \textbf{(2.2.1)}$$

In this equation z_t is a deviation at time t from the mean value of the variable of interest. The process a_t may be regarded as a random shock of white noise (i.e., its expected value is zero and its standard deviation is a constant over time). Theoretically the summation may go back to the first value of the time series. However, practically it is limited to several steps backwards in time. Box and Jenkins discuss at length how one goes about choosing the optimal number of time steps to be included in the model. The ψ's in the model are unknown constants which can be determined as part of the modeling process. It can be shown that under suitable conditions equation (2.2.1) reduces to the following:

$$z_t = a_t + \pi_1 * z_{t-1} + \pi_2 * z_{t-2} + \ldots \qquad \textbf{(2.2.2)}$$

This equation relates. z_t (the deviation at time t) to the deviations at previous times with the addition of a random shock at time t. The π's are unknown constants. Using the data available for modeling, the π's are determined and then the ARMA model can be used to predict the change from time t to t+1 using the changes from t-1 to t, t-2 to t-1, etc. If the model is a true representation of the values of z_t then the uncertainty introduced by using (2.2.2) to compute z_t is simply a_t (the unknown random shock). This is the beauty of ARMA modeling but alas, the financial markets are not that simple!

The ARIMA models differ from ARMA models in that there is no assumption that the deviations are from a fixed mean value. In other words, the values of z may drift up or down as they are not tied to a fixed mean. Daily changes in stock prices are an example of a non-stationary time series. For ARIMA models, the basic assumption is that some difference of the process is stationary. In other words, if we look at the first order difference (or second, or third, etc.) of z, we can use an ARMA process. Using only the time series itself, one can gain some insight into the nature of the model. When trying to develop a model to predict the change in the current value of a financial instrument, ARMA and ARIMA models are useful in determining the importance of past changes and moving averages of the series. Often candidate predictors based upon differences and moving averages of the time series itself turn out to be among the most relevant predictors. However, with the massive power of today's computers (as compared to the machines of the 1960's and 1970's), there is no need to limit the search to models based solely upon the time series itself.

There is also no need to limit the search to linear models. The ARMA and ARIMA models and the many variations spawned from these techniques are parametric in nature: they are based upon linear equations with unknown parameters. Weigend and Gershenfeld trace the first example of the recognition of the limits of linear modeling to one of the early pioneers in time series analysis: S. Ulam. In 1957 Ulam discussed the problems associated with predicting the next values of the time series generated with the following simple equation:

$$x_{t+1} = \lambda\, x_t \left(1 - x_t\right) \qquad \textbf{(2.2.3)}$$

Linear modeling fails for this problem. In 1980 Tong and Lim described a technique called the threshold autoregressive model (TAR) which is considered the first globally nonlinear method for modeling time series. A description of the method is included in Tong's book: H. Tong, *Threshold Models in Nonlinear Time Series Analysis*, Springer Verlag, 1983. Since then a lot of interesting work on nonlinear time series analysis has been reported. Contributions from many of the leaders in the field are included in a book edited by T. Subba Rao: *Developments in Time Series Analysis (in honor of M. B. Priestley)*, Chapman and Hall, 1993.

The general subject of *data mining* has received considerable attention in recent years. The combination of huge databases and very powerful computers has led to a thriving modeling industry. Many academics and industrialists have realized that benefits can be obtained if tools are available for extracting information from their databases. As a result, commercial software based upon well-known techniques is readily available. There are three basic categories into which most popular techniques fall (S. M. Weiss, N. Indurkhya, *Predictive Data Mining: A Practical Guide*, Morgan Kaufman, 1998):

1) Techniques based upon derivation of mathematical equations (e.g., classical regression and neural networks).

2) Techniques based upon development of logical rules.

3) Techniques based upon similarity (or distance).

Within each category there are variations and hybrid technologies are becoming popular. All of these techniques can be used to develop non-linear models in which a variety of candidate predictors are considered. . Access to many publicly and commercially available data mining software products can be found at the Internet site *www.kdnuggets.com.*

Currently within the financial community modeling based upon neural networks is receiving considerable interest. A number of books have recently been written on the subject (A. P. Refenes, *Neural Networks in the Capital Markets*, Wiley, 1995, E. Gately, *Neural Networks for Financial Forecasting*, Wiley, 1996, J. S. Zirilli, *Financial Prediction using Neural Networks*, International Thompson Publishing, 1996). However, experience with neural networks is making users aware of the major problem with this technique: computational speed decreases dramatically as the number of input variables increases. In addition, the need for data records is also strongly dependent upon the number of input variables. (S. Haykin, *Neural Networks – A Comprehensive Foundation*, Prentice Hall, 1994). One method of enhancing the neural network approach is to preprocess the data in an attempt to eliminate some of the input variables.

Techniques directed towards development of a set of rules are popular in many data mining applications. The appeal of these methods is that the output is a model that is easily understandable. For example, in credit scoring applications, it is comforting to be able to explain why a person is given or refused credit. Contrast these sorts of models to the output of a neural network in which a prediction is just a number derived from a non-linear transformation based upon many computed weights. Probably the most well know rule generating technique is CART (L. Breiman, J. Friedman, R. Olshen, C. Stone, *Classification and Regression Trees*, Pacific Grove: Wadsworth, 1984, also see *www.salford-systems.com*). In the financial community use of genetic algorithms to generate rules is gaining in popularity (R. J. Bauer, *Genetic Algorithms and Investment Strategies*, Wiley, 1994).

Similarity or distance based techniques use the data as the model. No attempt is made to replace the data with a set of equations. When making a prediction, the basic concept is to find previous examples of similar (i.e., nearby) cases. If the database is large (as is usually the case in financial market modeling), the need for online memory is large. However, the available RAM in today's machine is much greater than the machines of a few years ago and will be much less than the available RAM a few years down the line. The problem, of course, is locating the nearby cases in a multi-dimensional space. If these cases can be located efficiently, then the computational speed can be orders of magnitude faster than neural networks. Kernel regression (KR) is a technique fitting into this category and rapid implementations are possible (see Chapter 4). Thus KR can be used either as a preprocessor to neural networks or as an alternative stand-alone modeling technique.

2.3 The Curse of Dimensionality

A well known concept in financial market modeling is *the curse of dimensionality*. This concept actually relates to all very complex systems: the amount of data required increases exponentially with the number of variables appearing in the model (R.E. Bellman, *Adaptive Control Processes*, Princeton University Press, 1961). It is also well known that the greater the system complexity, the greater the number of dimensions required to adequately describe the system behavior. (This concept is

sometimes referred to as the *law of requisite variety*.) Financial markets are indeed very complex systems so we require the ability to develop models with a fairly large number of dimensions (for example, four or more) and therefore there is the need for a large number of data points. Probably there is a real model that if known could predict financial market movement with absolute certainly! This model is often called *God's model* and he alone knows the true structure of the model. We live in a very complex world and there are many phenomena which drive financial markets and often in a highly non-linear fashion. Very few people seriously believe it is possible to determine *God's model.*

The result of the curse of dimensionality is that we go into the modeling of financial markets knowing that there is no hope of completely understanding the causes of market behavior. The markets are just too complex to be able to develop models that give consistently accurate predictions. What can be done however is to take a statistical perspective of the task. Can we make predictions that have some utility? The predictions have utility if we can use them in such a manner that we gain some edge on random guessing. If the models explain some of the movement in the data then they might have utility.

To have some hope of developing useful models of very complex systems (like financial markets), we need to propose models with a large number of variables (i.e., dimensions). One hopes that if there are enough proposed candidate predictors, some of them might turn out to be useful. Unfortunately, there is no guarantee that a useful model can be determined even if the number of candidate predictors is large (for example, several hundred). Typically one chooses candidate predictors based upon solid economic theory. For example, we might expect that changes in interest rates from yesterday to today might have some influence upon the change in the price of gold from today to tomorrow. The choice of useful candidate predictors is a subject that continues to receive considerable attention in the literature and is discussed at greater length in Section 2.4.

From the preceding paragraph one should *not* get the impression that all the candidate predictors will appear in the final model. A typical strategy is to attempt to find a subset of the candidate predictors upon which the final model is based. The concept of *data sparseness* was introduced in Section 1.3. As the dimensionality of the model increases by one dimension, the density of the data is halved. As the number of dimensions increases to a value required to develop useful models, often the amount of data becomes too small to support the model! In Section 1.3 it was emphasized that the maximum dimensionality of the model is governed by the data available for modeling. If the amount of data is too small, then strategies to partially compensate for this lack of data are required. There are several methods for combating this *curse of dimensionality*:

1) Combine variables in such a manner that one new variable includes effects from several of the original variables.
2) Use a multistage modeling strategy (i.e., the outputs of one stage are the inputs to the next stage).
3) Increase the amount of data by combining similar data sets.

There are well known techniques for combining variables (e.g., *Principle Components*). Multistage modeling is an alternative strategy for accomplishing the same effect. This subject is discussed in a later chapter. Increasing the amount of data can often be accomplished in a straightforward manner. For example, if one is trying to model securities, it might be possible to use groups of similar stocks (i.e., from a particular sector of the stock market). By combining the data from a number of stocks, one can develop a larger database available for modeling. Similarly, modeling of the commodities markets can be based upon grouping similar commodities (e.g., currencies).

2.4 Candidate Predictors

One of the first tasks required to model financial markets is to propose a set of *candidate predictors.* If we really knew what drives the market of interest, then there would be no need for candidate predictors. In reality, no one really knows with certainty what drives financial markets. We might have some idea which time series have relevance regarding the market of interest, but that is usually the limit of our knowledge. Knowing that series A, series B and series C might have some relevance regarding market X is a long way from knowing the connection between A and X, B and X

and C and X. What can be done is to develop a set of candidate predictors based upon each of these series and add these predictors to the overall set of predictors that will form the basis of the modeling process.

2.4.1 Differences

The most obvious candidate predictors are simple differences. Assume we are trying to model price changes one time unit into the future. Some obvious choices for candidate predictors are the last changes over N times units (where several values of N might be selected). For example: N = 1, 2, 3 and 5. As an example consider the following table:

DATE	PRICE	Y	X1	X2	X3	X4
970610	865.8	2.95	3.3	19.7	23.7	19.8
970611	868.75	1.6	2.95	6.25	22.65	25.95
970612	870.35	24.35	1.6	4.55	7.85	28.25
970613	894.7	8.65	24.35	25.95	28.9	48.6
970616	903.35	0.65	8.65	33	34.6	40.85
970617	904	-1.35	0.65	9.3	33.65	38.2
970618	902.65	-3.6	-1.35	-0.7	7.95	33.9
970619	899.05	8.55	-3.6	-4.95	-4.3	28.7
970620	907.6	-3.5	8.55	4.95	3.6	12.9

Table 2.4.1 - S & P prices and Price Differences

In Table 4.2.1 The Y column is future looking: tomorrows closing S & P price index minus today's price index. The X columns are all backwards looking. The X1 column is the one day price difference, X2 is the two day price difference, X3 is the 3 day price difference and X4 is the 5 day price difference. Each of the four X columns can be considered as candidate predictors for Y. At the end of each day the X values can be computed. If we could determine a model based upon these four X's, then we could predict Y.

To quickly test whether or not these X's by themselves are powerful predictors of Y, we can compute correlations as described in Section 2.1. Using 1250 days of data (920722 to 970630) the results are not very encouraging. The correlation coefficients for Y versus X1, X2, X3 and X4 are –0.0548, -0.0554, -0.0726, and –0.0658 respectively. These differences are slightly negatively correlated with Y but the values are close to zero. (Using a statistical *t-test* we can show that these values are not significantly different from zero. W. Mendenhall & T. Sincich, *Statistics for Engineering and Science* – 3rd Edition, Maxwell MacMillan, 1992[1]) It can be concluded from these numbers that a simple **ARMA** or **ARIMA** model will not be sufficient to predict the changes in the S&P price index. However, these X's might combine well with other candidate predictors to create a model that does have a greater degree of predictive power.

Differences from one series can also be used as candidate predictors for another series. For example, changes in the short term and long term interest rates might be used as candidate predictors for an S&P model. Alternatively, one might prefer differences of the ratios of the series. There are many variations to this theme and the number of differences that can be proposed is only limited by the analyst's imagination.

2.4.2 Moving Averages

[1] To test the significance of *r*, the *t* statistic is computed*: t = r*sqrt(n – 2) / sqrt(1 – r²)* where *n* is the number of data points used to compute *r*. If the true value of *r* is zero (i.e., there is no correlation), then *t* should be distributed according to the *Student's t* distribution with *n*-2 degrees of freedom. This distribution rapidly approaches a standard normal distribution: with n-2>10 the values are less than 15% greater than the standard normal. The largest value of *r* above (i.e., -0.0726) yields a value of *t* = -2.56. The probability of a value of *r* being at least this negative (if the true value is zero) is approximately 0.5%.

Moving averages by themselves are not useful candidate predictors. If, for example, a time series is trending upwards and then suddenly shifts direction, moving averages of the series will also change direction but at a later time. As the time period of the moving average increases, the time lag in this change of direction also increases. This lag in response time makes them quite useless as predictors. However, when used in conjunction with other series they can be quite powerful.

A typical predictor based upon moving averages is the ratio of a series to its moving average. Many trading systems are based on rules concerning such ratios. For example, if the ratio of the closing price of a series to its moving average minus one changes sign, then buy if the change is positive or sell if it is negative. The problem can be stated in mathematical terms: time series such as market prices and interest rates are non-stationary. Moving averages of these time series are also non-stationary, but the ratio of the series to its moving average is a stationary series about a mean value of one. Candidate predictors are only useful if they are based upon stationary time series. The need for stationarity of the candidate predictors is due to a basic modeling assumption: past history is relevant to future behavior. If a candidate predictor is non-stationary then there is a distinct possibility that current values of the candidate predictor are not within the range of past values. If this is the case, then we cant make a prediction based upon similar past values as none exist!

There are other series that are similar to moving averages but serve slightly different purposes. Two of these are *moving medians* and *exponential smoothings*. Series based upon moving medians take longer to compute than moving averages but they are less sensitive to outliers in the data. The additional computing time is due to the need to sort the data to compute a median value. Exponential smoothings of data serve a similar purpose as moving averages and moving medians but require less computing time. Instead of **N** (the number of time periods) used to compute a moving average or a moving median, exponential smoothing uses a smoothing parameter α which is assigned a value between 0 and 1. We define the exponential smoothing of **SERIES** as follows:

$$\mathbf{ES\_SERIES(t) = SERIES(t) * \alpha + ES\_SERIES(t-1) * (1 - \alpha)} \qquad \mathbf{(2.4.1)}$$

The exponentially smoothed value of the series at time **t** is computed using only the value of the series at time **t** and the exponentially smoothed value at time **t-1**. It can be seen that the contribution of **SERIES(t)** is the same for both moving average and exponential smoothing if $\alpha = 1/N$. However, all values of **SERIES** up to time **t** affect the exponentially smoothed value. For moving averages and moving medians only the last **N** values are used to compute these series.

2.4.2 Moving Slopes

Moving slopes of time series are similar to moving averages. They can be used to identify a trend but they change direction rather slowly. For example, if a time series is in an upward trend, the value of a moving slope of the series is positive. If the series suddenly changes direction, the moving slope will eventually turn negative but at a later time. The lag in response time increases as the time period used to determine the slope increases. Like moving averages, moving slopes are also non-stationary time series. The value of a moving slope is that it is a measure of the "trendiness" of a time series. It can therefore be used to create stationary candidate predictors that capture this quality.

Once again the use of moving slopes to create candidate predictors is only limited by the imagination of the analyst. A very simple use of a moving slope is to remove the trend from the difference series. Differences are not exactly stationary series but they are close to being stationary. The series created by subtracting a moving slope from a difference is stationary. Whether it is a useful candidate predictor is another matter.

As an example of a more interesting candidate predictor based upon a moving slope, consider two time series: SERIES1 is the N day moving slope of a price series and SERIES2 is the change in volume of the market in question. The product of these two series might be interesting. This product is not exactly stationary but since it is based upon one day changes in volume it is fairly close to being stationary. If the volume change is positive and the one day difference in the price is in the opposite direction to SERIES1, then the market might be changing direction. This candidate predictor is an

example of a series that is probably quite meaningless on its own. However, when examined conjointly with other series, it might be useful.

2.5 The Equity Curve

When modeling financial markets, at some point there is a need to generate equity curves. The purpose of modeling financial markets is to develop trading strategies and the evaluation of any trading strategy requires analyses of the equity curves. A number of parameters can be computed from an equity curve and these parameters are then used to evaluate the *quality* of the equity curve. Some of the most popular measures of the quality of an equity curve are:

1) The rate of return
2) The Sharpe ratio
3) The maximum drawdown
4) The average drawdown

The rate of return: Clearly, the purpose of a trading system is to generate a return on one's capital. Thus the rate of return (ROI) is probably the most important single measure of performance. Usually it is stated on an annualized basis. To compute ROI, one looks at the initial and final equity. However, there is another issue that must be considered: is the equity curve based upon *compounding*? In other words, if the positions are allowed to grow as the equity grows, then we assume compounding. Alternatively, if a standard position size is used then there is no compounding. Without compounding the computation of ROI is:

ROI=100*(((equity[final]/equity[initial])*RECS_PER_YEAR/RECS)–1) **(2.5.1)**

With compounding the computation is:

ROI=100*((pow(equity[final]/equity[initial],RECS_PER_YEAR/RECS)–1) **(2.5.2)**

In these equations, RECS are the number of records in the equity curve. The pow function is a standard C function which returns the first argument raised to the power of the 2nd argument. As an example, assume that the equity curve is based upon 500 daily records and RECS_PER_YEAR is 250 (i.e., 2 years of data). Assume that the ratio of equity[final]/equity[initial] is 1.5. Without compounding, ROI is 25%. With compounding, ROI is 22.5%.

As a more realistic example, consider a portfolio of stocks. If one were to maintain a buy-and-hold strategy, average performance comparable to the S&P index could be expected. Looking at the closing prices for the S&P index from 600104 to 980113 the index rose from 59.91 to 952.12 (i.e., a factor of 15.9 in 38.03 years. On a compounded basis, the ROI for this period is 7.54%.

The Sharpe Ratio: The Sharpe Ratio as a measure of performance was suggested by William F. Sharpe. His book (W. F. Sharpe, *Investments*, Prentice-Hall, 1979) summarizes his many contributions to the analysis of investment strategies. There are several different definitions of the Sharpe Ratio in use today, but usually it is defined as the ratio of ROI (expressed as a fraction and not a percent) to σ (the standard deviation of the equity changes). Clearly, for this measure of performance to be a dimensionless number, the units for ROI and σ must be the same. To annualize σ, the value of σ generated from the daily

fractional equity changes must be multiplied by sqrt(RECS_PER_YEAR). The resulting equation is:

Sharpe_ratio = (ROI/100) / (σ * sqrt(RECS_PER_YEAR)) **(2.5.3)**

Using the S&P data (from 600104 t0 980113), a value of 0.553 is obtained. This is not a very impressive number! A value less than one implies that on the average equity swings are larger than the ROI. After all, the alternative is to invest money in a "risk-free" vehicle (like a CD or a short term Treasury Bill). When modeling financial markets, the objective is to develop a portfolio that significantly out-performs the risk-free interest rate (which changes over time) but do it in such a way as to minimize risk. The Sharpe Ratio takes both of these factors into consideration.

The Maximum Drawdown: The maximum drawdown is a measure of "pain" as well as performance. The drawdown at any given moment is the fractional decrease in equity from the previous equity high point. As an investor watches an investment lose value, he or she begins to feel pain. The maximum drawdown is a very important measure of this phenomenon. Using a vector language like **TIMES** (Zmanim Inc., Walnut Creek, Calif), the drawdown can be computed as follows:

```
MaxEquity = scan max(Equity);
EquityRatio = Equity / MaxEquity;
Drawdown = 1 – EquityRatio;
```

The scan operator creates a series based upon the max operator. The first term is Equity[1], the second term is Equity[1] max Equity[2], the third term is Equity[1] max Equity[2] max Equity[3], etc. The input series is **Equity** and the resulting series **Drawdown** is a series with the same length as **Equity** and with values from zero to the Maximum Drawdown. For the S&P series, the maximum observed drawdown was 0.482 which occurred on Oct 3, 1974. The S&P index fell from a high of over 120 on Jan 11, 1973 to 62.28 (i.e., a decrease of almost 50%). This time period was almost two years in duration and included the mid-East oil crisis.

The Average Drawdown: This measure of performance is similar to the Maximum Drawdown. It is the average value of drawdown over the entire period. For the S&P data, the Average Drawdown in this period was 0.087. This means that the S&P index over the 38 year period analyzed was on the average down 8.7% from its previous high.

A really interesting question is: can drawdown be predicted? Clearly, if anyone could predict *when* a serious drawdown was about to occur, he or she would be able to amass a considerable fortune. This is a very difficult prediction problem. However, a relatively simple problem is predicting *the probability of a drawdown of size P% or greater* at some time in the future. A simple model of financial markets is that the daily fractional changes in equity are normally distributed around a non-zero mean value. If the mean value is positive, then one would expect the equity curve to gradually rise but exhibit periodic upward and downward swings. If we denote the mean value as μ and the standard deviation as σ, then we can say that the equity changes are distributed as follows:

(Equity[I+1] – Equity[I]) / Equity[I] = μ ± σ **(2.5.4)**

Distributions based upon this model are often called Inverse Gaussian or Wald distributions (N. L. Johnson and S. Kotz, *Continuous Univariate Distribution*s, Houghton Mifflin, 1970). The "motion" derived from this distribution is called *Brownian Motion*. It can be shown (Private Communication, P. Feigin) that the probability of a drawdown of P% or greater can be related to μ and σ as follows:

$$\text{Prob(Drawdown} >= \text{P)} = \left(1 - \boldsymbol{P}/100\right)^{2\mu/\sigma^2} \qquad \textbf{(2.5.5)}$$

Using the S&Pdata, the values of μ and σ for the 38.03 years of data are 0.0003263 and 0.008601. The exponent in Equation 4.3.5 is thus 8.8226. In Table 4.3.1 the probabilities of drawdowns for various values of P are compared to the actual values obtained from the data.

P-VALUES	10%	20%	30%	40%	50%
Actual	0.511	0.347	0.236	0.13	0.026
Eq 2.5.5	0.636	0.395	0.238	0.14	0.043

Table 2.5.1 – Actual and predicted fraction of days with drawdown >= P (S&P price index).

For this simple Brownian motion model of the S&P price index, the results are surprisingly accurate. For example, Equation 2.5.5 predicts the probability of a 40% drawdown as slightly greater than 1%. The actual observed number of days with drawdowns of this or greater magnitude was slightly less than 1%. Even for a P of 5% the results are still within statistical accuracy. To prove this, a random number generator was used to create a series of the same length (i.e., 9572 records) using the same values of μ and σ. This experiment was repeated 10 times. The average fraction of days with drawdowns greater or equal to 5% was 0.654 with a standard deviation of 0.093. The value of 0.511 is thus well within the 2 sigma range.

From Equation 2.5.5 we see that the exponent $2\mu/\sigma^2$ is a powerful parameter which can be used to predict the probabilities of drawdowns of varying magnitude. In Table 2.5.2 the exponent required to achieve various probability levels are listed for a variety of P values.

P-VALUES	10%	20%	30%	40%	50%
0.0001	87.4	41.3	25.8	18.0	13.30
0.001	65.6	30.9	19.4	13.5	9.96
0.01	43.7	20.6	12.9	9.0	6.64

Table 2.5.2 – Values of $2\mu/\sigma^2$ required to achieve varying drawdown probabilities

As an example of how Table 2.5.2 is used, consider a requirement that a trading system achieve a probability of less than 0.001 for a 30% drawdown. From the table we see that an exponent of at least 19.4 is required. If, for example, we could achieve a 20% annual ROI, then the daily μ would be 0.00073 (assuming 250 trading days per year, $0.00073 = 1.2^{\frac{1}{250}} - 1$). The target value of σ would thus be sqrt(2μ/19.4) which is 0.00867.

2.6 Measuring the Efficiency of a Modeling Method

Regardless of the method used for modeling, it is often useful to be able to measure how well the method succeeds. Measurements of efficiency are useful when comparing different methods or when doing parameter studies within the domain of a particular method. For example, if one is using a neural network approach to modeling, one might want to study the effect of the number of neurons upon performance of the network. A very powerful measurement technique is based upon the use of artificial data that has been constructed in such a manner as to have the sort of properties that one anticipates in real data sets. If, for example, a modeling method fails using artificial data, it is important to understand the reason for failure before attempting to model real data. Using real data it is only possible to detect failure if one can be assured that some underlying model drives the data. For financial application this is often not the case. The starting point in the generation of artificial data sets is to first list the properties that one would expect in such data and then build the artificial data sets accordingly.

To build an artificial data set, one first has to propose a model. Typically *Y* (the dependent variable) is created as a function of several *X* variables (the independent variables) plus random noise. If, for example, the modeling method will include a search through a large candidate predictor space, the data set would include many *X* variables, most of which are unrelated to *Y*. The noise component may be generated in a variety of different ways. For example, it might be pure Gaussian random noise (mean of zero and σ of some specified amount), it might be a random value between XMIN and XMAX or it might be generated by some sort of chaotic process [Ma99]. For financial market modeling, one problem often encountered is a change in market volatility over time. To simulate volatility changes one might use a Gaussian random noise generator but vary σ over time.

The most straightforward measure of efficiency of the modeling process is the fraction or percentage of the variance of the true signal (i.e., the pure function without noise) that is captured by the process. For example, lets say a data set of 15000 records has been created in which the *Y* column is 10% signal and 90% random noise. Lets assume that 10000 records are used to create the model and the remaining 5000 are used to test the model. By comparing the actual values of *Y* with the values of *Ycalc* (the calculated values of *Y* for the test set), the VR (Variance Reduction) can be computed using Equation 1.4.1. If, for example, a value of VR = 7.25 is computed, then we can say that the modeling process was 72.5% efficient for the particular example under consideration. In other words, the modeling process captured 72.5% of the actual variance in the pure signal. Note that a perfect model would yield a VR = 10 because the *Y* column is 10% signal and 90% noise. Due to the random component in the generated data, it is possible to obtain a value of VR slightly in excess of 10, but if VR significantly exceeds 10 for this example, one would immediately suspect that the process is somehow overfitting the data.

In Section 5.1 a TIMES program for generating artificial data [Ti99] is included in Figure 5.1.1. This program illustrates the generation of a data set with 10 columns of *X* variables and five *Y* columns. The *X* columns are created using a Gaussian random number generator. The first *Y* column (i.e., column 11) is the pure signal column, and the next four columns are created by adding varying degrees of noise to column 11. Column 12 has a 50% noise component, column 13 has 75% noise, column 14 has 90% noise and column 15 has 95% noise. The pure signal is created using a nonlinear function of X2, X5 and X9. The equations for generating measured levels of noise (i.e., Equations 5.1.1 through 5.1.4) are also included in Section 5.1. The results included in Chapter 5 are based upon artificial data generated in this manner. In addition, the comparison of KR (kernel regression) and NN (neural networks) included in Appendix D are also based upon a similar data set.