Neuroeconomics and Big Data

In the Capital Markets

By

Gordon H Dash

Nina Kajiji

D R A F T

Affiliations

Gordon H Dash, Ph.D.
University of Rhode Island, USA
The NKD-Group, Inc., USA
Istituto Superiore per Imprenditori e Dirigenti di Azienda, Italy (ISIDA)

Nina Kajiji, Ph.D.
University of Rhode Island, USA
The NKD-Group, Inc., USA

Acknowledgements:
The authors wish to thank the many comments, suggestions, edits, and artwork, provided by the following individuals and groups: Laurel Dwyer, Miriam Dash, John Messere, Francesa Shaw, Joe Piecuch, and the students of the University of Rhode Island and ISIDA.

Part IV: Trading by Neuroeconomics: Policy and Performance

This page intentionally left blank

Chapter

16

Neuroeconomics and Big Data in the Capital Markets

There are some very good reasons why the business world is so enamored with the term "Big Data." In the not so distant past financial firms collected their own data using techniques like survey design, questionnaires and sampling methods. Under this tried and proven historical approach the main issues for management fell into a veracity and validity. Today, internal data is now collected electronically – a process that reduces the emphasis on veracity and validity. It also means that the focal point for this new treasure trove of electronically collected data has shifted to analytics. When data service firms integrate software to make it easy (or easier) for consumers and business analyst to have relatively easy user-access to the vast amounts of public data available in the "cloud" and elsewhere, we find that new service markets are formed. For example, data sampling alone takes on an increasing importance as management seeks out new ways to create decision making models that represent the wealth generating process of the firm. The White House view on "Big Data" has, in part, been recognized and partly described as follows:

> *"The data age has arrived. From crowd-sourced product reviews to real-time traffic alerts, "big data" has become a regular part of our daily lives. In 2013, researchers estimated that there were about 4 zettabytes of data worldwide: That's approximately the total volume of information that would be created if every person in the United States took a digital photo every second of every day for over four months! The vast majority of existing data has been generated in the past few years, and today's explosive pace of data growth is set to continue. In this setting, data science -- the ability to extract knowledge and insights from large and complex data sets -- is fundamentally important."*

Read the entire release with audio here: http://www.whitehouse.gov/blog/2015/02/19/memo-us-chief-data-scientist-dr-dj-patil-unleashing-power-data-serve-american-people

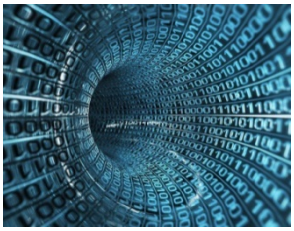
Chief Data Scientist (CDS) or Chief Data Officer (CDO).

Under President's Obama administration the use of data to improve the operation of the U.S. government and its interactions with those who depend on such data has taken on a formal role. On 09-May-2013 the President signed Executive Order 13642 – an order to made open and machine-readable data the new default for government information. The President's office was the first crate the U.S. Chief Data Scientist (CDS). This position is described as one that is designed: *"…to reasonably source, process and leverage data in a*

timely fashion to enable transparency, provide security, and foster innovation for the benefit of the American public, in order to maximize the nations return on its investment in data."

The Reality of "Big Data"

But, the move to "Big Data" is not without complexities. Immediately, analysts are thrust into the process of sampling and selection in order to reduce the scope of any subsequent analytical work. If that were not enough, "back-in-the-day" traditional data analysts easily reflect on how numeric data came packaged in a tidy structured format. But today, some of the most important data needed for business decision models is only available in an unstructured format. Unstructured data is not necessarily stored in the traditional row-column format (e.g. a spreadsheet of historical stock price activity). Instead, the data may be free-flowing (e.g., video feeds, irregular high-frequency trades of capital market instruments) and may not have a uniformly defined begin and end (e.g., a market open and close). The question naturally arises, in this world where "Big Data" is the norm rather than the exception, as contemporary analyst how should we proceed to process this data in order to maximize its value to the decision unit?



New Professional Opportunities in "Big Data"

Ah, it used to be so well-defined and straightforward. What was so well-defined? How data occupied space and, subsequently, the associated analytics of this well-structured data. Owing to the fast growing nature of today's data (and databases) in linear and nonlinear (or unstructured) formats, there is an increasing need to train new analysts who possess the skills to extract what is relevant, important, and challenging in the data. The impact on the modern post-secondary education process is immediate. Professors and teachers need to seize the opportunity to train tomorrow's data scientist. Of course, that opportunity comes with a cost – the cost of disruption to the traditional approach of teaching students how to analyze data sets that are borne by years of a structured presentation.

For the student in search of a profession with a future, it is in their best interest to seek out data science programs that have (or, are in the process) of meeting the challenge to teach "Big Data" skills. These future scientist "expect" courses to cover "Big Data," so the market driven directive is already in place: re-engineer the basic analytics course or, at a minimum, reconstitute the existing analytics course to incorporate "Big Data" concepts. But, either way, the preparation of students for current and future jobs requires a dedication to "Big Data."

The "Big Data" Impact on Capital Market Models

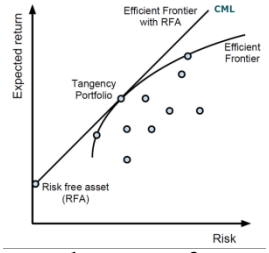
As we turn our attention to how these issues impact the capital markets, it is useful to turn our attention back to the study of the capital markets returns-generation process. To do this we first focus on a review of the venerable capital asset pricing model (CAPM) and its empirical estimation form generally referred to as the 'market model.' One important consequence of these models is the assumption of "market equilibrium." In the capital markets economic equilibrium can be considered a state where the market prices of securities are balanced by their measurable risk (of generating an expected return) vis á vis the overall market's expected return. Stated differently, under the assumption of equilibrium, the competitive price of a security is strictly related to the overall characteristics of the market. In the early 1960's this was an appealing thought – a capital market that tended toward equilibrium conditions. Later in this chapter, we ask a simple question – is this a valid assumption when data is "Big" and collected in (near-) real-time?

The CAPM in Review

The Capital Asset Pricing Model, or CAPM, is an equilibrium model that describes the relationship between risk and expected return. Since the early 1960s the model is used extensively in the pricing of risky assets. The fact that the reference point for this model dates back to the 1960s (actually, a bit earlier) and intersects with the manual ticker tape (in use from 1870 to 1970) suggests that the CAPM may be somewhat inappropriate for contemporary markets that are defined by global access to markets, high-frequency and automated trading. For example, a simple comparison of volume for the S&P 500 over two time periods, the first being Jan, 1960 to Dec, 1964 and the second being Jan, 2009 to Dec, 2014 finds that average monthly volume increased by a factor of 1,200 times! (See Excel table in folder – put all, or some, in an appendix).

The history of equilibrium pricing theory is rooted in the derivation of capital asset pricing theory and its empirical estimation component the capital asset pricing model (CAPM). CAPM is constructed upon the following assumptions:

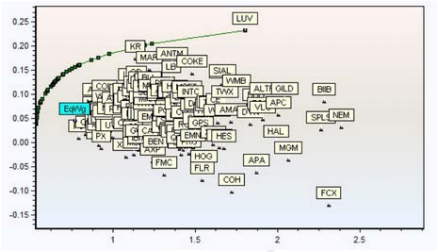
1. All investors "Markowitz" efficient. This assumption means that all investors determine the same set of efficient portfolios and that the universally desired efficient portfolio is the one that is a combination of the risk-free asset and the tangency portfolio on the efficient set; a relationship that produces the linear Capital Market Line (CML).



2. There are a large number of investors in the market and each is a price taker.
3. Markets are frictionless; a condition that also includes an absence of taxes and transactions costs.
4. Investors can lend and borrow at the same risk-free rate over the planned investment horizon.
5. As rational utility-maximizing decision makers, investors only focus on expected return and variance. All investors desire investment returns but dislike variance (risk).
6. All investors plan for the same single holding period and have the same information and beliefs about the distribution of returns (i.e., homogeneous expectations and beliefs).
7. The market portfolio (tangency portfolio) consists of all publicly traded assets.

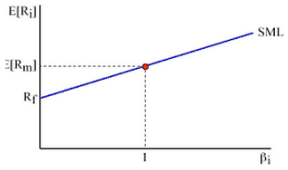
What do these assumptions mean to individual investors who operate in today's high frequency 24/7 trading environments? From the introductory years of the early 1960s, market deregulation alone leads one to ponder the effectiveness of assumptions about capital market efficiency (as stated above) and the implications for individual investor diversification decisions. The implications behind the market and model assumptions are fairly easy to state; however, as we turn our attention to the return-generating models relied upon by investors we will find it beneficial to enhance our study to include some of the latest findings from neuroscience and behavioral finance. To begin this line of thinking, let's re-state the simplifying assumptions to meet the needs of individual market participants:

1. Investors implement the Markowitz mean-variance algorithm (or a Sharpe simplified model) to enumerate the efficient frontier (set). The efficient set is comprised of both "corner" and "intermediate" portfolios. A corner portfolio (shown by the dots) is one on the efficient set that is uniquely different in risk-return characteristics to the adjoining corner portfolio. An intermediate portfolio (the line between two adjacent corner portfolios) is a portfolio between two corner portfolios. It can be shown that an intermediate portfolio is a linear weighted average of two corner portfolios.



2. The equilibrium CML allows us to state that there is a linear relationship between risk and return. By definition, the market (tangency) portfolio is mean-variance efficient and universally desired. Hence, the expected return for an individual asset (security or portfolio) can be estimated based on the relationship between the returns of the asset and those of the market index – Beta.

3. The relationship between an individual asset's returns and those of the market portfolio is commonly referred to as the "security-market-line" or SML. The pricing relationship of the SML can bet expressed through the market model. The various forms of the market model are as follows:



Model Definition	Estimating Equation	Interpretation
Single-Index Model	$E[R_i] = r_f + \beta_i\left(E[R_{M,t}] - r_f\right)$	$E[R_i] - r_f = \beta_i\left(E[R_{M,t}] - r_f\right)$
Expected vs. Realized Returns	$R_i - r_f = \beta_i\left(R_M - r_f\right) + \varepsilon_i$	The difference between expected and realized returns is attributable to the error term, ε_i.
The Market Model	$R_i = \alpha_i + \beta_i\left(R_M\right) + \varepsilon_i$	The intercept, α_i, and the slope coefficient, β_i, can be estimated by using historical security and market returns.

In the series of equations presented above, R_i, denotes the return on any asset (or portfolio) *i*. $R_{M,t}$ denotes the return on the market portfolio over *t* periods and β_i is simply the $\frac{cov(R_i,R_M)}{var(R_M)}$. What you have learned in the introductory course is that the SML explicitly states that there is a linear relationship between the expected return on an asset and the beta, β, of that asset with the market portfolio. If we define the market risk premium for the ith security as $E[R_i] - r_f$, then whenever $E[R_i] - r_f > 0$, the higher the beta is for an asset which, in turn, implies a higher expected return for the asset (and, of course, vice-versa).

Estimating Security Returns by Regression Analysis

This section presents the results of using the CAPM and Market Model to predict asset returns. To enhance the comparison of these two important models, the estimation is performed across three security tickers: IBM, STX and UTX.

CAPM

Using the CAPM model, where Beta is the primary determinant of expected return, it is not unusual to find predictions for annual expected returns calculated using only the computed beta and the observed risk-free rate.

Asset	Model Results
CAPM	$E[R_i] = r_f + \beta_i \left(E[R_{M,t}] - r_f \right)$
International Business Machines	$E[R_{IBM}] = 1.0\% + 0.86\left(9.0\% - 1.0\%\right) = 7.88\%$
Seagate Technologies	$E[R_{STX}] = 1.0\% + 2.32\left(9.0\% - 1.0\%\right) = 18.56\%$
United Technologies Corporation	$E[R_{UTX}] = 1.0\% + 1.10\left(9.0\% - 1.0\%\right) = 9.80\%$

For the calculations above, the risk-free rate is stated at 1.0% with the annualized beta coefficients provided by *Yahoo!* Finance.

Market Model

Let us state the regression equation for the market model as follows:

$R_{i,t} = \alpha_i + \beta_{i,M} R_{M,t} + \varepsilon_{i,t}$, where $\varepsilon_{i,t} \sim iid\ N(0,\sigma^2)$, and $\varepsilon_{i,t}$ is independent of $R_{M,t}$.

The notation *iid* implies that the error term from the regression analysis is independent and identically distributed. You ask, distributed in what way? The answer to that question is in the next part of the definition. $N(0,\sigma^2)$ tells us that the distribution is the normal (bell-shape) distribution with a mean of zero (0.0) and a measurable variance, σ^2.

When the market model is employed to predict a security's return, a lot of focus is placed on the metric alpha, α_i. Alpha captures the difference between the predicted returns based solely on beta and the total predicted return level. Technically, before using alpha for policy inference, one would subtract the risk-free rate from alpha. The result of this difference yields the excess return expected from the security over the risk free rate. An investment strategy followed by many is to search out high-alpha investment instruments. But, here is where we need to interject some of the quirks in relying on alpha as a policy tool. First, alpha is only as good as the asset's beta measurement. Second, alpha cannot distinguish why an asset underperforms (e.g., incompetence, market anomalies, etc.). Third, alpha may or may not truly reflect managerial skill.

Asset	Model Results
Market Model	$R_{i,t} = \alpha_i + \beta_{i,M} R_{M,t} + \varepsilon_{i,t}$
International Business Machines	$E[R_{IBM}] = -1.2\% + 0.97(9.0\% - 1.0\%) = 6.56\%$
Seagate Technologies	$E[R_{STX}] = 0.0\% + 2.31(9.0\% - 1.0\%) = 18.48\%$
United Technologies Corporation	$E[R_{UTX}] = 0.0\% + 1.14(9.0\% - 1.0\%) = 9.12\%$

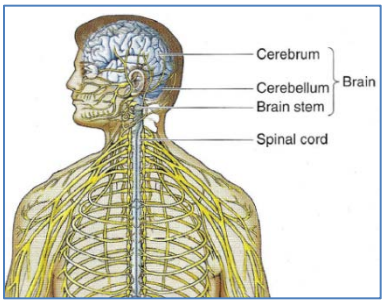
The market model results above show that alpha is a non-factor for both STX and UTX. By contrast, IBM produced a negative alpha estimate of -1.2%, or -0.2% on a risk-adjusted basis. To reiterate, for the CAPM results the published beta from Yahoo Finance was used. The Yahoo beta is estimated by using three-years of monthly return observations. For the market model the stock beta coefficients were estimated by solving a regression analysis using monthly returns for calendar years 2012 to 2014, inclusive. All estimated beta coefficients are statistically significant at the 95% confidence level when computed against the S&P500 market index. The alpha estimates for STX and UTX were not significantly significant at any confidence level.

By observation it is clear that the CAPM and Market Model approaches yield different, but similar results. We should never forget that these are models, not reality. The usefulness of any model is ultimately determined by its ability to closely replicate reality (or actual outcomes). In the next section we turn our attention to empirical issues when "Big Data" defines the observation period. Specifically, we ask – does the market model remain the modeling technique of choice when observations are observed on a high frequency basis? To answer this question the analytic process turns to recent advances in the field of neuroscience.

Capital Markets in Transition: Neuroscience Influences

The study of neuroeconomics (and neurofinance) is an outgrowth of research stemming from the relatively young but expanding field of neuroscience. While questions about how we hear, feel, hurt, move, learn, remember and forget are as old as the study of philosophy itself, the ability to tie it all together around the human brain remains rather novel and new. It is well known that philosophers and scientist from Ancient Greece to the Renaissance to the Nineteenth Century have proposed scientific inquiry into the inner workings of the human brain. Today, the evolution of this study has brought us to a modern day characterization of neuroscience and the professionals who are dedicated to the study of the brain – neuroscientist.

Because understanding the workings of the brain and central nervous system is such a vast field of study, the modern-day neuroscientist is generally a specialist. The recognized neuroscience medical specialties include the: neurologist, psychiatrist, neurosurgeon and neuropathologist. As you might imagine, the field also encompasses a great deal of discovery research. To that end, there are different career paths in the alternative types of experimental neuroscience. The list here is a bit more expansive than just the medical sub-divisions. More closely tied to the role of the modern capital market analyst is the field of computational neuroscience. But, within the realm of computational sciences, the list of other experimental neuroscience practitioners includes, but is not limited to: the Psychophysicist, the Molecular Neurobiologist, and the Neuropharmacologists.



As an interdisciplinary science, Neuroeconomics joins the study of economics, neuroscience and psychology in a manner that focuses on how individuals make economic decision. This objective can be met in many ways. For example, neuroscientist often attempt to build biological models of decision-making and then test such models in economic environments. The neuroscience approach in economic decision making has gained traction with each and every modern-day financial and economic crisis. Many pundits have asked: "If financial decisions and capital markets are driven by rational decisions in an equilibrium-based market, then what explains the apparent irrationality of the developing crisis?

Early on the approach was to focus on behavioral economics as an approach designed to incorporate psychological and emotional factors into models of individual decision-making. By the decade of the 1990s, the tools of neurobiology spurred on the study and analysis of molecular and physiological mechanisms on decision-making. Hence, neuroeconomics served as a convergence point between the neural and social sciences in an attempt to classify and predict financial decisions involving, at a minimum, risk-reward profiles. The scientific process of neuroeconomics emerged in parallel with the scientific approach relied upon by the biological scientist: observation, replication followed by interpretation.

In the next section we take some of what we have learned and apply it to the always necessary study of financial time series analysis. Students of the capital markets must have a keen understanding of how to process a time series of items such as: pair trading, financial returns, realized volatility, credit risk

modeling, and more. The section begins with a short view of contemporary approaches using the well known and easily applied exponential smoothing family of techniques. This includes the application of a moving average. Quickly the section advances to the use of more modern nonlinear methods known as artificial neural networks (ANN). Greater definition of this neuroscience inspired approach is provided later.

Applied Neuroeconomics: Modelling with Artificial Neural Networks

An artificial neural network (ANN) is an information-processing paradigm that is designed to emulate some of the observed properties of the mammalian brain. First proposed in the early 1950's it was not until the technology revolution of the 1980's that a multitude of alternative ANN methods were spawned to solve a wide variety of complex real-world problems. Today, the contemporary literature on ANNs is replete with successful reports of applications to problems that are too complex for conventional algorithmic methods or for which an algorithmic specification is too complex for practical implementation. The robustness of the ANN method under difficult modeling conditions is also well documented. For example, ANNs have proven extremely resilient against distortions introduced by noisy data. In short, the ANN paradigm has developed a track-record as a good pattern recognition engine, a robust classifier, and an expert functional agent in prediction and system modeling where the physical processes are not easily understood or are highly complex.

At the heart of these impressive results is the learning process. The algorithmic learning process is achieved during an iterative training phase where adjustments are made to the synaptic connection weights that exist between the neurons. These connection weights represent the imputed knowledge that is required to solve specific problems. The radial basis function artificial neural network (RANN) is a nonlinear method that presents its output as a linear combination of the radial basis functions and neuron parameters. RANNs are used for: a) function approximation; b) time series prediction; c) classification; and, c) system control.

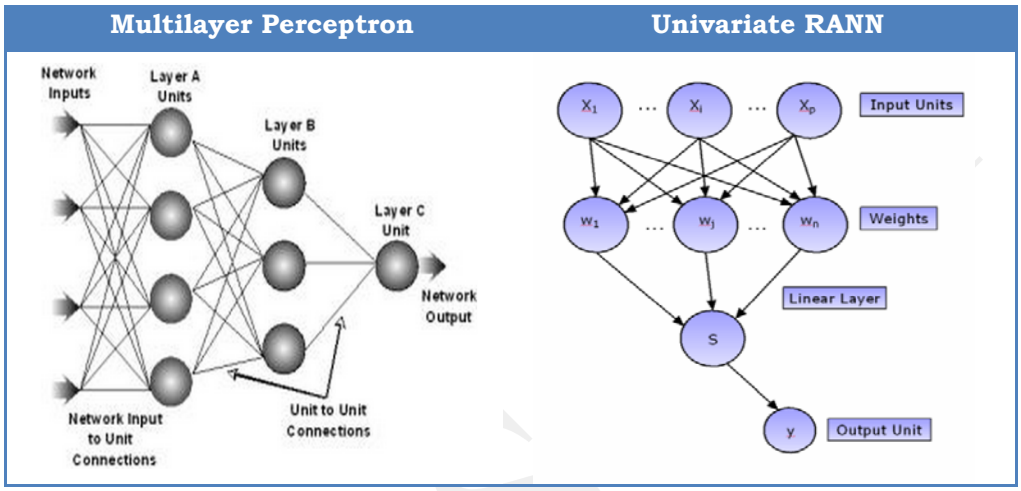
RANN: Radial Basis Artificial Neural Network

WinORS supports two basic neural network algorithms. First, there is a basic implementation of the venerable Backpropagation method. The featured ANNs in WinORS are the four Kajiji-RANNs (with emphasis on the K4-RANN) and the Kohonen self-organizing map (K-SOM). The K-SOM method is used solely for classification problems (i.e., can be classify period returns into industry groupings). Because our focus is to identify a suitable replacement for the 'market model' in the face of Big Data (function approximation and time series prediction of expected returns), in the next section we specifically address the four versions of the supported RANN algorithms with a detailed look at the senior most approach, the K4-RANN.

Radial Basis Function Artificial Neural Network in WinORS$_{e\text{-}AI}$

ANNs represent a class of machine learning algorithms that are inspired by biological neural networks. We note that a biological neural network is, at its core, the brain and central nervous system of animals. This machine learning approach is designed to accept (or estimate) functions that rely upon some number of inputs that produce one or more outputs. As you might imagine, there are a number of different ways to "conceptualize" just how the brain (that

is, the nervous system) might pool together all of these inputs, the implied data communication and implied weightings to generate a specific output(s). The intent of this chapter is not designed to introduce and compare alternative design methodologies; however, it is typical to hear of networks defined as: Radial Basis Function (RBF) networks; Kohonen self-organizing maps (K-SOMs); Recurrent neural networks (RNN); and more. To meet the specific purposes of this study, the only network topology of interest is the RBF-ANN. From this point forward we refer to this network topology as the RANN.



By review of figure XX, we show how a multilayer perceptron network can have several hidden layers whereas the RANN has but a single hidden-layer.

High Frequency Finance

While we do not explore the defining research over the decades since the introduction of the theory, it is safe to say that such an assumption is dissipated by many factors – high frequency and "Big Data" measurements being just one of those reasons. So if CAPM, while remaining an important contributor to low-frequency analysis is not dead, what do you do when you are obligated to work in a high-frequency (second-by-second, minute-by-minute, etc.) "Big Data" environment?

Today, high frequency finance is "Big Data" finance. The use of the term "Big Data" conjures up many different opinions about just what constitutes the demarcation from large data to "Big Data." To get a uniform set of definitions in place this chapter, as other chapters in this book, will turn primarily to the free encyclopedia, Wikipedia. To begin, Wikipedia provides us with a good succinct definition of Big Data: "*Big Data usually includes data sets with sizes beyond the ability of commonly used software tools to capture, curate, manage, and process data with a tolerable elapsed time.*" While the Wikipedia description goes on to develop the characteristics of "Big Data", when we apply the concept to financial returns, we find that the following 3-V's characterize what is most important to the market analyst:

Characteristic	Meaning
Volume	The quantity of data generated
Velocity	The speed of generation of data
Veracity	The quality of the data being captured

In our earlier econometric modeling experiment (estimating the market model with 61 observations) the data set was stored in an Excel file and the actual data represented monthly period returns. As we turn our attention to high frequency finance where observations are recorded as they occur, the time-scale of the data may be reduced to seconds (or smaller) time scale. As you can imagine, the (Vol)ume of the data will be magnitudes greater than the monthly observations downloaded from Yahoo Finance. Further, the (Vel)ocity of increase will be determined by the already demonstrated increasing volume of activity in stock trades. Lastly, like any system that goes faster and faster, there is an opportunity for a glitch (or two) to occur. That is, the (Ver)acity of the data may be called into question. To demonstrate these points consider the near high frequency 20-minute quotes on IBM equity in table xx. The following quotes are observed approximately every 20-minute over one trading day: 05-January-2015. The normal trading hours on the New York Stock Exchange are 9:30 a.m. to 4:00 p.m.

Day Counter	Date	Time	AM/PM	Price	Volume (x100)
1	1/5/2015	9:45:17	AM	162.06	2039
2	1/5/2015	10:00:16	AM	160.27	422872
3	1/5/2015	10:20:16	AM	159.54	697813
4	1/5/2015	10:40:19	AM	159.80	988173
5	1/5/2015	11:00:21	AM	159.66	1215705
6	1/5/2015	11:20:20	AM	159.88	1379944
7	1/5/2015	11:40:19	AM	159.73	1557971
8	1/5/2015	12:00:18	PM	160.18	1725527
9	1/5/2015	12:20:19	PM	159.88	1904213
10	1/5/2015	12:40:23	PM	159.80	2010757
11	1/5/2015	1:00:20	PM	159.64	2205159
12	1/5/2015	1:20:16	PM	159.47	2344129
13	1/5/2015	1:40:22	PM	159.43	2520571
14	1/5/2015	2:00:19	PM	159.62	2633047
15	1/5/2015	2:20:56	PM	159.52	2771483
16	1/5/2015	2:40:16	PM	159.53	2921194
17	1/5/2015	3:00:16	PM	159.36	3065426
18	1/5/2015	3:20:19	PM	159.26	3212732
19	1/5/2015	3:40:19	PM	159.52	3462765
20	1/5/2015	4:05:16	PM	159.71	3856652

An observation period of 20-minutes intuitively means that the database will record 20 trades per day. With approximately 250 trading days per year, a one-year data set for IBM would measure approximately 5,000 entries per data element. Three- and ten-year data count would require 15,000 and 50,000 entries, respectively. Clearly, in a commercial scale system where literally thousands of securities across multiple types (ETFs, equities, derivatives, etc.) are being tracked the volume of data is enormous and the velocity of its change is often measured by the second (or less) over a 24/7 trading cycle. As an important contributing issue, with data reception from various global markets, today's "Big Data" database systems all must check the veracity of the received data. That is, for example, the managing software behind these systems must

ask question such as: is every price recorded within an acceptable statistical range? Referring back to table xx, what should the system do if, say, an erroneous price of $500 was recorded at 02:20:56? Clearly, a set of verification routines would signal an outlier and subsequently "clean" the data based on a set of proprietary algorithms.

Compared to some marketing data, like the collection of mouse clicks and Internet URLs visited, financial data like that in table xx does not appear to be overly complex (or unstructured). The data is well organized; each observation comes at a pre-defined time-scale; and, the data seems to be relatively clean. This is in contrast to data that comes at irregular intervals and reflects the impulses of both expert and non-expert human beings. But, in our financial studies we learned that the real issue to study lies in understanding the time-series properties of individual time series as well as the correlated structure among a sample of financial assets where each is measured over similar time slices. This makes for a relatively interesting and complex modeling study.

Big Data and the Market Model for Predicting Asset Returns

The choice of which models to use when the data scientist is faced with Big Data is divergent and growing more complex by the day. When data volume grows, as it has with a move to 20-minute time scale aggregation, the use of the simple regression model becomes suspect to inadequate for the purpose of estimating future returns. To exemplify this statement, in table xx the results of estimating the market model over 15,373 observations is presented. Note how both the intercept is negative and close to zero (-3.5401E-05). The regression parameter is positive and small (0.0284). Importantly, both are statistically insignificant (not significantly different than 0.00). For completeness we should also note that the explanatory power of the model is almost nil. That is, the adjusted R-Square is 5.0538E-05 (which is pretty close to zero).

A	B	C	D	E	F
SUMMARY OUTPUT					
Regression Statistics					
Multiple R	0.017013962				
R Square	0.000289475				
Adjusted R Square	5.05383E-05				
Standard Error	0.002985218				
Observations	4186				
ANOVA					
	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	1.07964E-05	1.08E-05	1.211514	0.271095879
Residual	4184	0.037285821	8.91E-06		
Total	4185	0.037296617			
	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*
Intercept	-3.54014E-05	4.61505E-05	-0.76709	0.443073	-0.000125881
X Variable 1	0.02840967	0.025810834	1.100688	0.271096	-0.022193274

Statistically Insignificant

$$R_{i,t} = -3.5401E-05 + 0.0284 R_{M,t} + \varepsilon_{i,t}$$

In the absence of a statistically reliable market model, we must ponder if there is an alternative modelling methodology available to help define a "new normal" of high-frequency "Big Data" finance. We turn to neuroscience, specifically, neuroeconomics for some possible answers.

Big Data and RANN Estimation of the Market Model

The Kajiji-4 radial basis function artificial neural network (K4-RANN) provides an excellent tool by which to replace the OLS estimation of the market model. By definition, RANNs do not depend upon parametric assumptions (e.g., a Gaussian distribution of residual error terms). The network-based RANN is a nonlinear mapping method that produces linear and additive weights. In short, the K4-RANN is a nonlinear regression method that returns a linear-additive function. The question of importance in this chapter is whether the K4-RANN can do a better job at estimating the high-frequency market model.

To answer this question it is important to learn how to measure the estimating performance of the machine learning RANN algorithm. Table xx shows the results of solving eight different RANNs each with a slightly different combination of parameter controls. Without delving into all of the details, model 6, is the K4-RANN model of choice to estimate the high frequency market model (at least for ticker IBM). The K4-RANN model parameters of importance here are: a) data transformation using standardize-1 method (see Appendix A for details); b) a radius value of 2.5; and c) the choice of the inverse multiquadric transfer function. Alternative model results are also shown in table xx.

	Model 1	Model 2	Model 3	Model 4	Model 5
Target	IBM	IBM	IBM	IBM	IBM
Computed Measures					
Lambda	1026.306	479.238	99.078	243.747	8500339.626
Actual Error	8.39E-01	8.40E-01	8.39E-01	8.39E-01	8.39E-01
Training Error	7.70E-11	6.91E-10	1.68E-09	1.63E-09	3.19E-09
Validation Error	9.96E-09	1.56E-08	6.35E-09	4.46E-09	3.32E-09
Fitness Error	6.70E-09	1.07E-08	4.81E-09	3.52E-09	3.28E-09
Performance Measures					
Direction	0.954	0.947	0.953	0.957	0.944
Modified Direction	0.944	0.937	0.944	0.949	.
TDPM	0.000	0.000	0.000	0.000	0.000
R-Square	99.98%	99.96%	99.98%	99.99%	100.00%
AIC	-78778.995	-76806.661	-80162.806	-81463.920	-81770.536
Schwarz	-78766.316	-76793.982	-80150.127	-81451.241	-81757.857
Theil	0.01372	0.01737	0.01163	0.00995	0.00959
MAPE	24.43637	20.48652	24.81819	22.84862	22.93782
Model Characteristics					
Training (N)	1380	1380	1380	1380	1380
Training (%)	33.0%	33.0%	33.0%	33.0%	33.0%
Transformation	STD:1	STD:1	STD:1	STD:1	STD:1
Min/Max/SD	n/a	n/a	n/a	n/a	n/a
Radius	1.000	0.500	1.500	2.000	2.500
Algorithmic Settings					
Method	Kajiji-4	Kajiji-4	Kajiji-4	Kajiji-4	Kajiji-4
Error Min. Rule	GCV	GCV	GCV	GCV	GCV
Transfer Function	Multiquadric	Multiquadric	Multiquadric	Multiquadric	Multiquadric
Solution Range	A!B4:D4189	A!B4:D4189	A!B4:D4189	A!B4:D4189	A!B4:D4189

	Model 6	Model 7	Model 8
Target	**IBM**	IBM	IBM
Computed Measures			
Lambda	21.142	190.612	29.704
Actual Error	8.38E-01	8.38E-01	8.40E-01
Training Error	3.60E-09	3.41E-09	1.09E-08
Validation Error	1.50E-10	7.30E-11	6.30E-09
Fitness Error	1.28E-09	1.17E-09	7.84E-09
Performance Measures			
Direction	0.968	0.968	0.960
Modified Direction	.	.	.
TDPM	0.000	0.000	0.000
R-Square	99.99%	99.99%	99.96%
AIC	-85680.288	-86074.138	-78120.179
Schwarz	-85667.609	-86061.459	-78107.500
Theil	0.00601	0.00574	0.01484
MAPE	11.38721	10.10767	26.76225
Model Characteristics			
Training (N)	1380	1380	1380
Training (%)	33.0%	33.0%	33.0%
Transformation	STD:1	STD:1	STD:1
Min/Max/SD	n/a	n/a	n/a
Radius	2.500	1.500	0.500
Algorithmic Settings			
Method	Kajiji-4	Kajiji-4	Kajiji-4
Error Min. Rule	GCV	GCV	GCV
Transfer Function	Inv. Multiquadric	Inv. Multiquadric	Inv. Multiquadric
Solution Range	A!B4:D4189	A!B4:D4189	A!B4:D4189

Although different in magnitude, the weights, which are comparable to the OLS regression estimates, correspond in sign to the OLS solution. Using the AIC and Schwartz criterions, this model also shows superiority in overall fitting. The three models are only differentiated by the radius setting (2.50, 1.50 and 0.50, respectively). Like the earlier market model estimates on monthly data, the parameter weights are negative for the K4-RANN intercept (-0.018 vs. -0.012) and positive for the slope parameter (0.014 vs. 0.97).

Variables	Weights	Variables	Weights	Variables	Weights
Model 6	Model 6	Model 7	Model 7	Model 8	Model 8
GSPC	0.014	GSPC	0.008	GSPC	0.126
Time	-0.018	Time	-0.011	Time	-0.150

$$R_{i,t} = -0.018 + 0.014 R_{M,t} + \varepsilon_{i,t}$$

Based on an interpretation of performance measures commonly used in nonparametric statistical analysis, the K4-RANN market model using high frequency data is highly significant and representative of the modeled data. The estimated coefficients of the model are plausible and well within the scope of the modeled data. In summary, the K4-RANN has provided an extremely reliable and reasonable estimate of the two important pricing terms in a high frequency application – alpha and beta (systematic risk). To close this section, we remind the reader that the OLS results for the high-frequency experiment produced a statistically insignificant model by all parametric measurements.

APPENDIX A

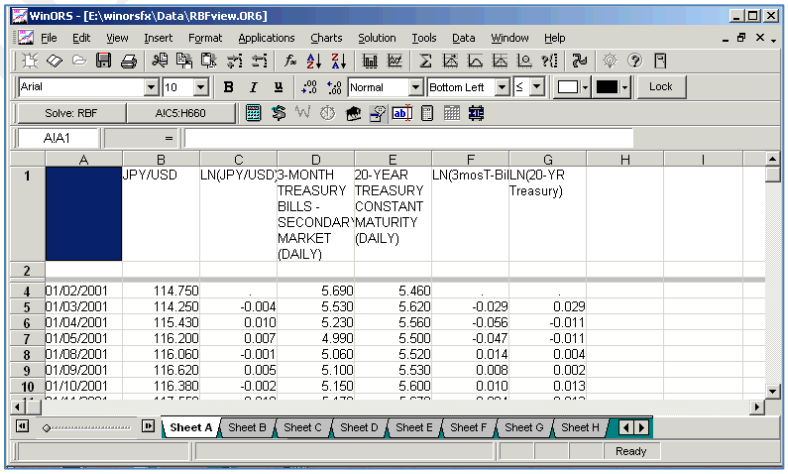
How to use RBF in WinORS*e-AI*

RANNs derive from the theory of function approximation. The network is known as supervisory as its use is divided into two stages. First, the network is trained on a subset of the data. Then, the output weights extracted from the training exercise are applied to the remaining data, known as the test set. One advantage of this approach is the RANNs are very fast machine learning tools. This is a property that we often exploit in "Big Data" function mapping – especially when modeling financial time series data.

The single layer of hidden nodes within the RANN implement of set of radial basis functions (e.g. Gaussian functions). The output nodes implement linear summation functions. In modeling terms, this means that to achieve a function approximation of, for example, the returns of a stock, it is necessary to have inputs (e.g., the market index, the risk-free rate, and possibly more) and one output target (the returns of the modeled stock). The appointed task of the RANN is to use the volatility of the input variables to replicate the volatility of the target variable. In the next section we use the K4-RANN to empirically test whether this neuroscience based approach can handle "Big Data" estimation of market returns. We already know that the CAPM is not ideally suited for this task.

Open A Sample File

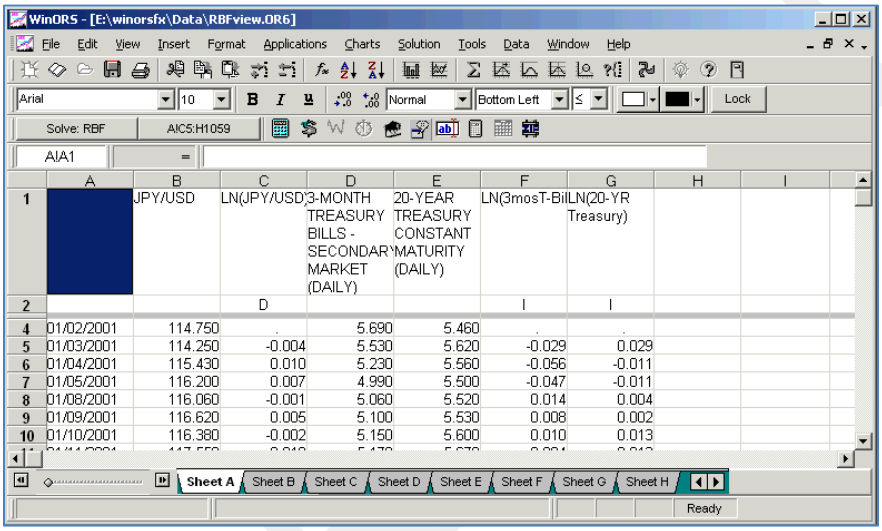
You may open the data file for the example presented here by using the WinORS web folder (File/Open/Web Data-icon on the outlook bar). The data in this file was imported from WinORS supported sites (see the FX and Economics data sub-menu off of the main Data menu). Column B presents the JPY/USD exchange rate where the log rate of return for this same series is presented in column C. Missing values in the yen-dollar exchange rate were estimated by using the menu tree /ACE. Similarly, following a similar menu tree created the ln rate of return data in column C: /ACR. Columns D and E are obtained from the menu tree: /DE. As with the FX data, the log (ln) rates of returns were created by the menu tree /ACR.



	A	B	C	D	E	F	G	H	I
1		JPY/USD	LN(JPY/USD)	3-MONTH TREASURY BILLS - SECONDAR MARKET (DAILY)	20-YEAR TREASURY CONSTANT MATURITY (DAILY)	LN(3mosT-Bill	LN(20-YR Treasury)		
2									
4	01/02/2001	114.750	.	5.690	5.460	.	.		
5	01/03/2001	114.250	-0.004	5.530	5.620	-0.029	0.029		
6	01/04/2001	115.430	0.010	5.230	5.560	-0.056	-0.011		
7	01/05/2001	116.200	0.007	4.990	5.500	-0.047	-0.011		
8	01/08/2001	116.060	-0.001	5.060	5.520	0.014	0.004		
9	01/09/2001	116.620	0.005	5.100	5.530	0.008	0.002		
10	01/10/2001	116.380	-0.002	5.150	5.600	0.010	0.013		

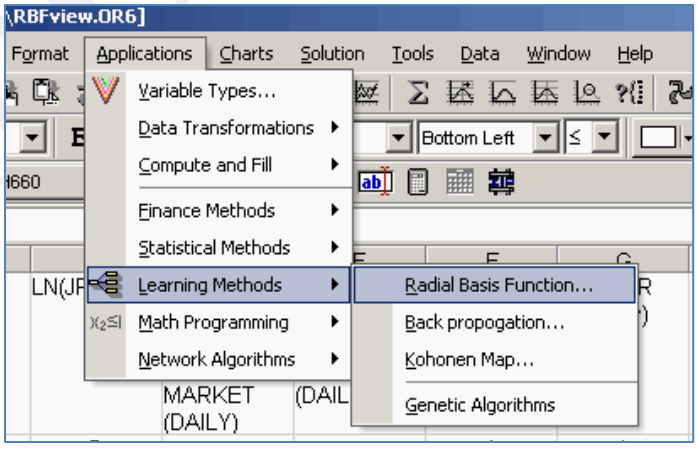
WinORS Variable Types

Row 2 of the WinORS spreadsheet is often referred to as the 'variable type' row. This row requires a single character identifier that defines the variable (column) to the chosen application. In the case of the radial basis function ANN, WinORS requires two variable types to be set. The dependent variable (sometimes referred to as a target variable in neural network studies) is identified by a variable type of 'D' in row 2. The predictor variables require the variable type of 'I' along this row. Below, the other required setting to obtain a solution is discussed -- the range of the data. However, we first introduce the dialog box associated with the radial basis function method.



RBF Model Execution

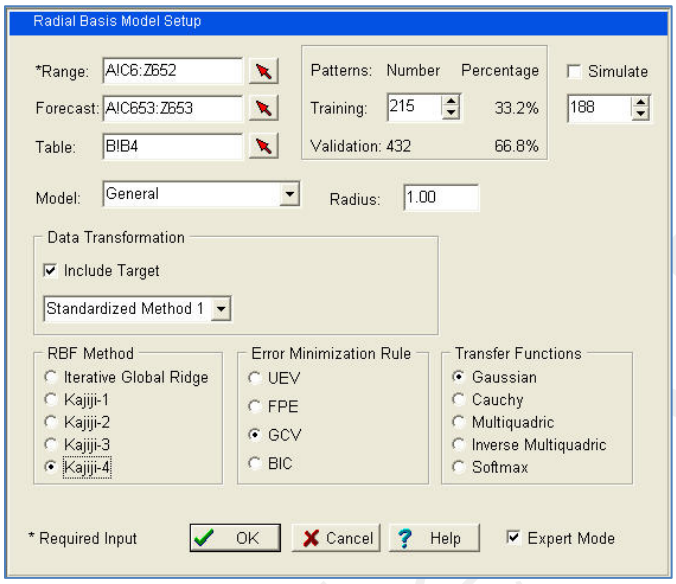
To execute the RBF application on the data set given the model specification (the variable types that are set) as specified, follow this menu tree: /ALR



The RBF Dialog Box

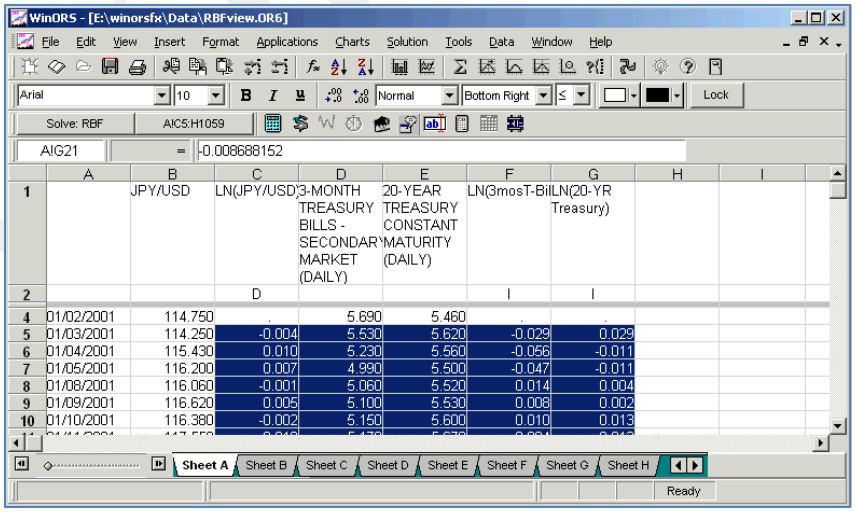
The first action to take on the RBF dialog box is to set the data range (*Range) correctly. In the following case, the data range is incorrect for the problem developed as part of this example. Use the backward facing red arrow to roll-up

the screen. When in the rolled-up state, use the mouse (or the keyboard) to highlight all cells that should be considered by the solution method.



Highlighting the Range

Here we see that, at a minimum, columns C through Z are included in the highlighted range. This breadth of range setting includes both the target (dependent) variable in column C, as well as the two independent variables in columns F and G, respectively. NOTE: only variables that are in the range with valid 'variable types' along row two will be considered in the analysis. Highlight down to the last row desired for inclusion in the analysis.

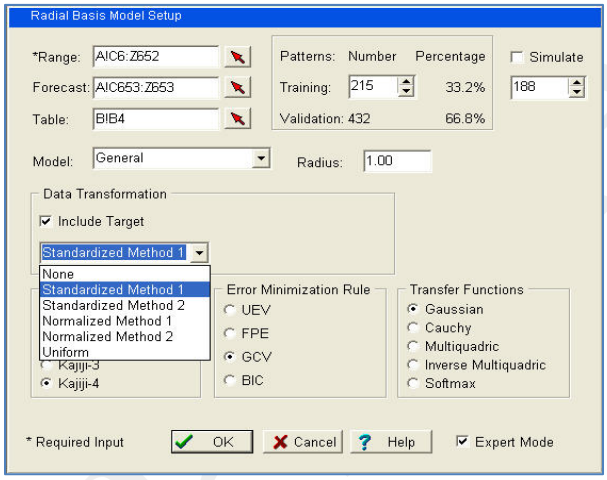


The RBF Dialog Box - Revisited

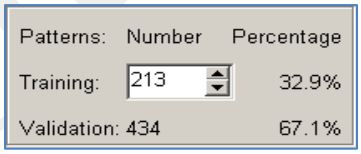
Once the highlighting is done and you roll-down the form, the dialog box will show the following results. First, notice that the model will be constructed over the range from cell C6 on tab A to cell Z652, tab A, inclusive.

Expert Mode

In order to set the ranges for the ***Forecast*** and ***Table***, the check box for Expert Mode must be in the 'on' state (see bottom right corner). Although it was not shown, the form was rolled-up for a second time using the backward facing red arrow associated with the ***Forecast*** range. In this example the forecast range was set to start at the next available row (653) and only this row (a one-period ahead forecast). NOTE: the forecast range can be blank. That is, it is acceptable to only model the target data (not model and forecast). Finally, the ***Table*** is the range over which to build the output table. While it is possible to direct the output to any specific location, it is strongly suggested to leave this setting at its default value.



Data Set Training and Validation Ranges

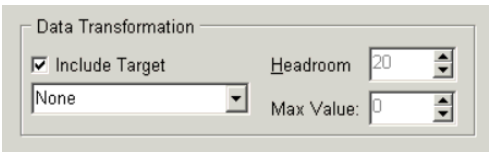


Start the neural network analysis by apportioning approximately 1/3 (33%) of the data set to the supervised training phase of the analysis. The remaining 2/3 (67%) of the data is automatically assigned to the validation phase. Use the provided spin control to increase (decrease) the allocation between the training and validation sets. Is the default proportion optimal? Unfortunately, this question is still open to debate. In fact, it is a major research issue in the field of study. The objective is to properly train the neural network. An improper training of the neural network may lead to an over-trained network. An over-trained network models both the true variability as well as the noise (error) in the variability of the data. In the current implementation, WinORS does not support techniques like cross-validation to automate the process of finding the best demarcation point between the training and validation sets.

Data Scaling by Transformation

Data scaling is another important issue in neural network modeling. Currently, WinORS supports the following five data transformation (scaling) methods: Standardized – Method 1; Standardized – Method 2; Normalized – Method 1;

Normalized – Method 2; and, Uniform.



It is possible to choose whether to apply the chosen transformation method to the target variable. Use the check-box to indicate Yes/No to this decision. The default is Yes - apply the transformation to the target variable.

Standardized Method 1

Method 1 is the well-known approach to creating a standardized value:

$$Z = \frac{(X - \mu)}{\sigma}.$$

Standardized Method 2

This method is invoked by subtracting the mean from all the data observations and multiplying by the square root of the inverse of the covariance matrix ($\sum$). The square root can be found since the covariance matrix is symmetric and can be diagonalized. This method essentially performs a coordinate transformation of a distribution to a different one where the new distribution has zero mean and the covariance matrix is the identity matrix. Let X be any random vector. The whitening transform of $X \rightarrow Y$ is given by: $Y = \sum^{-1/2}(X - \mu)^T$, where μ is the mean vector.

Data Transformations: Normalized

Using alternative data transformation functions can produce noticeably different RBF solutions. WinORS supports two data normalizing methods. The following notation is common to the four supported techniques.

Term	Definition
D_L	Lower scaling value
D_U	Upper scaling value
D_{min}	Minimum data value
D_{max}	Maximum data value
S_L	Lower headroom (%)
S_U	Upper headroom (%)

Where the values DL and DU are defined by for all supported normalized transformations:

$$D_L = D_{min} - ((D_{max} - D_{min}) \times S_L) / 100)$$

$$D_U = D_{max} + ((D_{max} - D_{min}) \times S_U) / 100),$$

Normalized Method 1

This approach computes a normalized data input value (D_V) from the actual data point (D) by:

$$D_V = (D - D_L) / (D_U - D_L).$$

The upper and lower headroom percent, S_L and S_U, are set separately during the model-building exercise. Each is controlled by an associated spin control on the dialog form.

Normalized Method 2

Data is scaled to a range required by the input neurons of the RBF ANN. Contemporary settings for the range are -1 to 1, or, 0 to 1. As with *normalized method 1*, both D_L and D_U are computed as shown above. The data range over which the network models the transformed data (D_V) is based on the settings for S_L and S_U. Specifically, the transformation is:

$$D_V = S_L + (S_U - S_L)*(D- D_L)/(D_U - D_L).$$

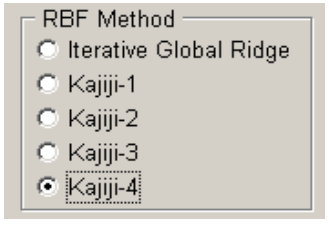
The values for S_L and S_U are controlled by the use of an associated spin control on the dialog form.

Uniform Transformation

The uniform method is designed to increase data uniformity during the process of scaling the input data into an appropriate range for neural network analytics. The method utilizes a statistical measure of central tendency and variance to remove outliers and spread out the distribution of the data. First, the mean and standard deviation for the input data associated with each input are determined. Next, S_L is then set to the mean minus some number of standard deviations. The number of standard deviations is set by a spin control on the dialog box. Similarly, S_U is set to the mean plus two standard deviations. By way of example, assume that the calculated mean and standard deviation are 50 and 3, respectively. Further, assume the user arbitrarily scales with a standard deviation of 2. Under these settings the S_L value would be 44, or (50-2*3). The S_U value is 56, or (50+2*3). Finally, all data values less than $S_L$are set to S_L and all data values greater than S_U are set to S_U. The linear scaling described under Method 2 is applied to the truncated data. By clipping off the ends of the distribution this way, outliers are removed, causing data to be more uniformly distributed.

Algorithmic Method

Choose the desired solution method



Kajiji-1

The Kajiji-1 algorithm is an ANN extension of explicit solution for generalized ridge (EGR) as proposed by Hemmerle [5] for parametric regression models. The Hemmerle method itself is based upon the iterative global ridge regression (IGR) as initially proposed by Orr [81]. The Orr method, like that of Poggio and Giorsi

[38, 39], is an adaptation of Tikhonov's regularization theory to the RBF ANN topology.

Kajiji-2

The Kajiji-2 method is an RBF ANN method constructed upon the unbiased ridge estimation (URE) of Crouse, Jin, and Hanumara [6] for parametric models.

Kajiji-3

Following Swindel [4], the Kajiji-3 method extends existing RBF implementations by incorporating a prior information matrix as means by which to augment the Tikhonov's regularization method as adapted to the RBF algorithmic design. The Kajiji-3 method is the Kajiji-1 algorithm augmented by the prior information matrix.

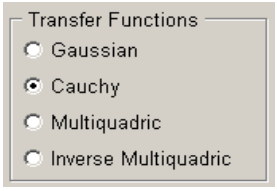
Kajiji-4

The Kajiji-4 method is the Kajiji-2 algorithm augmented by the prior information matrix (see Kajiji-3 for an important reference).

RBF Framework	*Description*
Kajiji-1	The Hemmerle method of closed form regularization parameter estimation applied to a RBF ANN
Kajiji-2	The Crouse et. al. method of closed form regularization parameter applied to a RBF Neural Network
Kajiji-3	Kajiji-1 with prior information
Kajiji-4	Kajiji-2 with prior information

Radial Functions

The special class of radial functions is what makes the RBF method unique. Radial functions decreases (increases) monotonically with distance from a central point. WinORS provides support for four radial functions: Gaussian, Cauchy, multiquadric, and inverse multiquadric. A Gaussian RBF monotonically decreases with distance from the center. By contrast, a multiquadric transfer function monotonically increases with distance from the center. Multiquadric-type RBFs have a global response (in contrast to the Gaussian which has a local response) and tend to be more plausible in biological research owing to its characteristic of a finite response.



Consider the following generalized statement of the RBF: $h(x) = \theta\left((x-c)^{\mathrm{T}}\Re^{-1}(x-c)\right)$, where θ is the function used (e.g., Gaussian), c is the center and $\Re$ is the metric. $\left((x-c)^{\mathrm{T}}\Re^{-1}(x-c)\right)$ is the distance between the input (x) and the center, c, in the metric defined by $\Re$. For the Gaussian, $\theta(z) = e^{-x}$; for the multiquadric, $\theta(z) = (1+z)^{0.5}$; for the inverse multiquadric;

$\theta(z) = (1+z)^{-0.5}$; and, for the Cauchy, $\theta(z) = (1+z)^{-1}$.

Radius

Under the radial basis function approach, a radius scales transfer functions. For example, the following is a one-dimensional example given by a Gaussian transfer function centered at *c*, and scaled by a 'radius' *r*:

$$h(x) = \exp\left(-\frac{(x-c)^2}{r^2}\right)$$

Error Minimization Rules

WinORS supports four alternative algorithmic error minimization rules. GVC is the default method.

Error Minimization Method	Description
UEV	Unbiased estimate of variance
FPE	Final prediction error
GVC	generalized cross validation
BIC	Bayesian information criterion

RBF Parameter Tab

The solution to the RBF modeling and forecasting application is placed on the next available tab. The output is split into five (5) dimensions. The first dimension is presented on rows 1 and 2 of the RBF Parameter Tab shown below in figure xx.xx

	A	B
1		Model 1
2	*Target*	LN(JPY/USD)
4	*Computed Measures*	
5	Lambda	0.000
6	Actual Error	3.83E-05
7	Training Error	3.50E-07
8	Validation Error	3.11E-07
9	Fitness Error	3.24E-07
11	*Performance Measures*	
12	Direction	0.904
13	Modified Direction	0.910
14	TDPM	0.000
15	R-Square	99.10%
17	*Model Characteristics*	
18	Training (N)	215
19	Training (%)	33.2%
20	Transformation	Standardize
21	Headroom	n/a
22	Radius	1.000
24	*Algorithmic Settings*	
25	Method	Kajiji-4
26	Error Min. Rule	GCV
27	Transfer Function	Gaussian
28		

Dimension 1 begins on row 1 indicates the model number. Row 2 restates the name of the target variable.

Descriptor	Model Characteristic
Blank	Model
Target	Variable name from the Data Sheet

The second dimension is displayed from row 4 through row 8. The variables displayed are:

Parameter	Description
Lambda	The regularization parameter
Actual Error	MSE for the entire data set
Training Error	MSE of the training data set
Validation Error	MSE of the validation set
Fitness Error	MSE of the training and validation sets

The third dimension reports on the overall performance of the RBF modeling application. The following characteristics are reported.

Accuracy Measure	Description
Direction	Direction is defined as the number of times the prediction followed the up and down movement of the underlying index.
Modified Direction	Modified Direction = ((# of correct up predictions / # of times index up) + (# of correct down predictions / # times index down)) – 1
TDPM	TDPM is a correction weight that compensates for incorrect directional forecasts by overall magnitude of the movement. The smaller the weight, the more accurate the training phase. Large weights are indicative of a missed direction, an incorrect magnitude adjustment, or some combination of the two.
R-squared	Traditional coefficient of determination

The fourth dimension summarizes the model input.

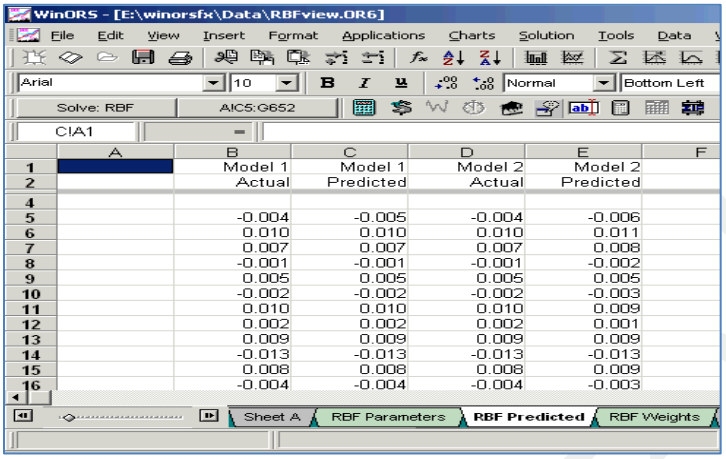
Item	Description
Training (*N*)	The number of cases in the training set
Training (%)	Percentage of *N* used in supervised training
Transformation	Method used to scale the actual data
Headroom	Scaling control for the normalize data transformation
Radius	Scaling used in the radial function

The fifth dimension summarizes the settings for the chosen algorithmic method:

Algorithmic Settings	User Choice
Method	*Kajiji-4*
Error Min. Rule	*GCV*
Transfer Function	*Gaussian*

RBF Predicted Tab

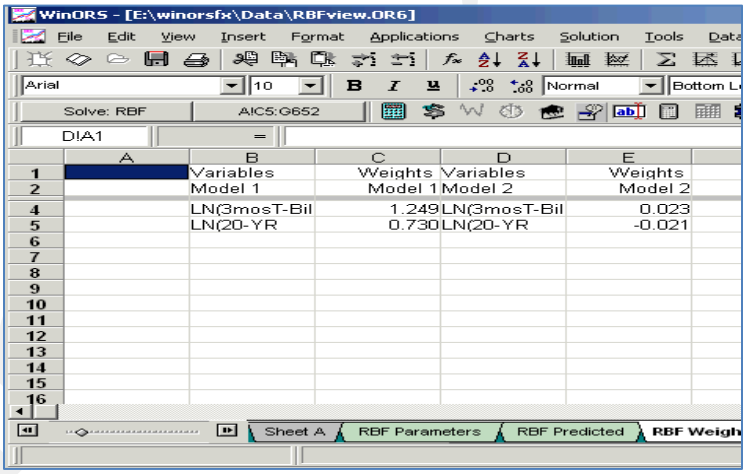
The RBF predicted tab presents the actual and predicted data for the current model. This data tab forms the foundation of the RBF diagnostic graphs.



	A	B	C	D	E	F
1		Model 1	Model 1	Model 2	Model 2	
2		Actual	Predicted	Actual	Predicted	
4						
5		-0.004	-0.005	-0.004	-0.006	
6		0.010	0.010	0.010	0.011	
7		0.007	0.007	0.007	0.008	
8		-0.001	-0.001	-0.001	-0.002	
9		0.005	0.005	0.005	0.005	
10		-0.002	-0.002	-0.002	-0.003	
11		0.010	0.010	0.010	0.009	
12		0.002	0.002	0.002	0.001	
13		0.009	0.009	0.009	0.009	
14		-0.013	-0.013	-0.013	-0.013	
15		0.008	0.008	0.008	0.009	
16		-0.004	-0.004	-0.004	-0.003	

RBF Weights Tab

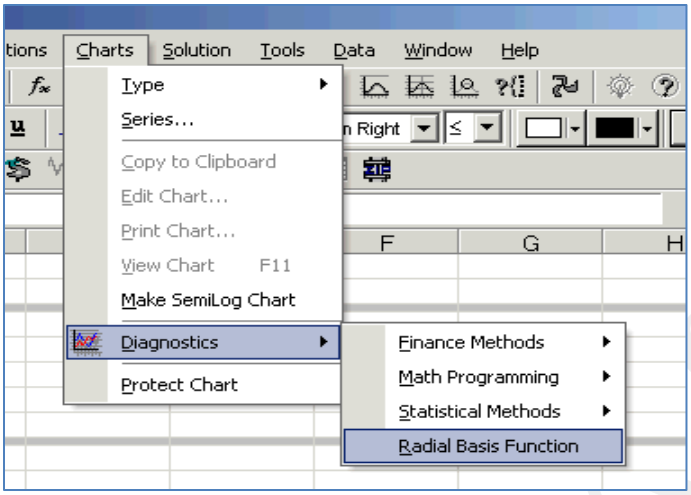
The final weights determined by the training phase of the solution are presented on this tab.

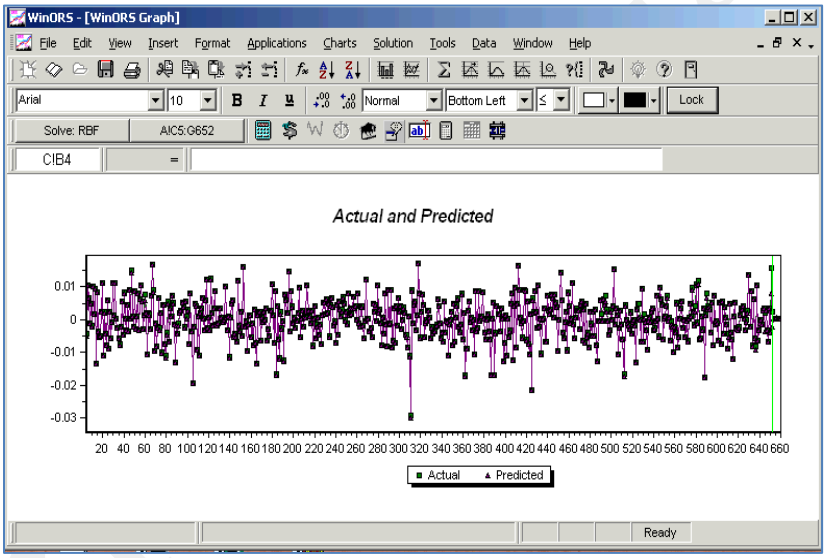


	A	B	C	D	E
1		Variables	Weights	Variables	Weights
2		Model 1	Model 1	Model 2	Model 2
4		LN(3mosT-Bil	1.249	LN(3mosT-Bil	0.023
5		LN(20-YR	0.730	LN(20-YR	-0.021
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

RBF Diagnostic Graphs

The WinORS Charts menu supports pre-programmed graphs for many of the WinORS applications. These graphs are referred to as diagnostic graphs. Choose the menu tree: /CDR to display the RBF predictive ability graph.

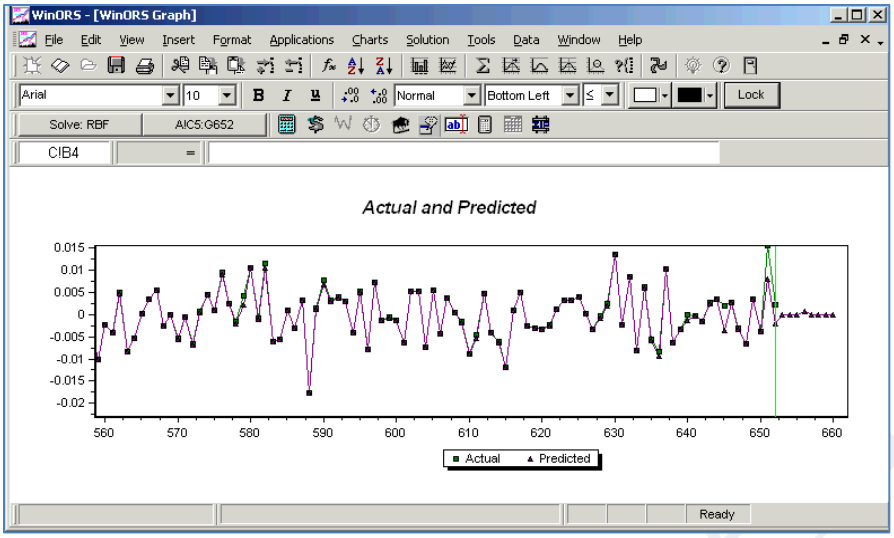




Zoom-In and Zoom-Out

For long time series, it is often useful to zoom-in and inspect a smaller part of the modeled relationship. To zoom the WinORS graph, follow these steps:

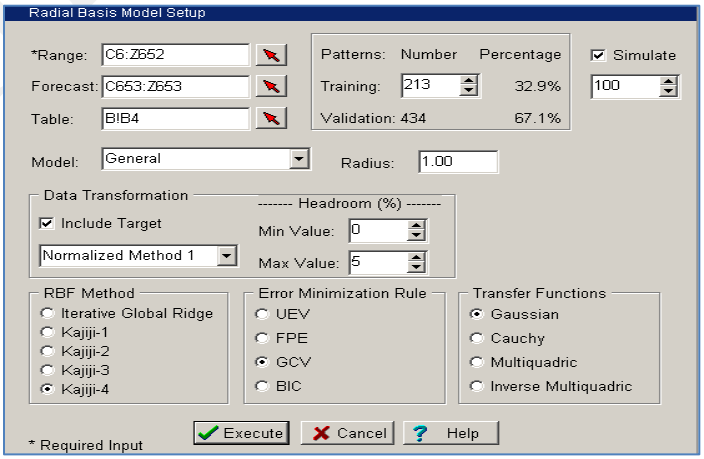
1. Place the mouse in the graph window. For example, place it about 3/4 from the right edge of the displayed graph.
2. Press and hold the left mouse button.
3. Move the mouse to the right.
4. Release the mouse button to see the zoomed image.
5. Repeating this action with a left-oriented movement will zoom-out the graph.



RBF Simulation

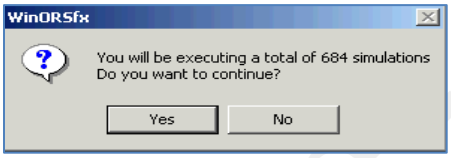
One of the difficult modeling issues all neural network researchers must confront is to determine how much of the data series should be used to train the network. WinORS provides a simulation option that focuses directly on locating the settings that produce the smallest network MSE within the simulation range. NOTE: The simulation option is not available when no transformation method is selected or it is standardized 1 or 2.

1. Begin by selecting the data transformation method, the RBF Method, Error Minimization Rule, and the desired Transfer Function.
2. Click the checkbox next to the Simulate option.
3. Use the spin control to set the observation from which to start the simulation procedure. The simulation begins from this value up to the last training observation (in the graphic, simulation will start at observation 100 and continue up to 213).
4. Click Execute to start the simulation and produce an answer.



Simulation Warning

WARNING! Simulation is a time-intensive procedure. For example, in the case of the settings as displayed above, for each data row in the simulation range six RBF solutions are computed -- one for headroom percent beginning at zero percent up to five percent. For the exhibit above where simulation is set for data observations 100 to 213 (114 observation rows), a total of 684 RBF solutions will be computed. While the solution time for each individual execution of the RBF algorithm will depend on a number of factors, assume that one solution requires 5 seconds to compute (including screen updates). Under these assumptions the simulation would require approximately 57 minutes to complete all operations (3,420 seconds). The final solution will report, the observation number and headroom percent combination that produced the smallest model MSE. Fortunately, the simulation option is only needed once (per selected transformation method).

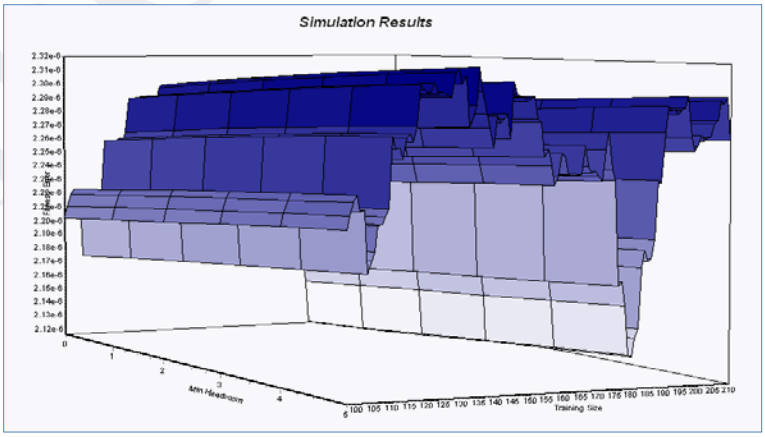


Simulation Results

The simulation results are summarized in two alternative formats; one graphical and one tabular on the spreadsheet tab 'RBF Simulate'.

Simulation: Graphical Results

A graphical result of the simulation effort is presented below. This chart is produced by using the menu tree: CHARTS/DIAGNOSTICS/RADIAL BASIS FUNCTION… Choose the Simulation Results option on the pick list. A review of the 3-D chart shows that the smallest MSE measures are produced around observation 180 with a low headroom % value. The 'best' solution is found exactly on the 'RBF Simulate' tab.

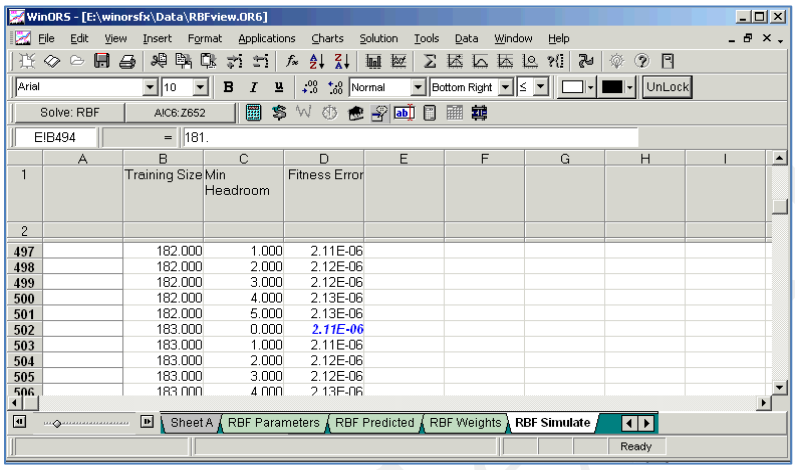


Simulation Results: Tabular

The RBF Simulate Tab presents three columns for your review. First, the number of observations used in the current simulation is presented under the column titles 'Training Size'. In the specific case of the simulation discussed in this document, the next column focuses on the 'Headroom' simulation

parameter. Finally, the MSE model fitness error is presented in column D.

The lowest fitness MSE value is highlighted in the color blue. In the case of the scripted analysis presented in this chapter, the lowest fitness MSE occurs with when solving a model with 183 training rows and minimum headroom of zero percent (0.0%).



APPENDIX B

Early and Contemporary Time Series Modeling

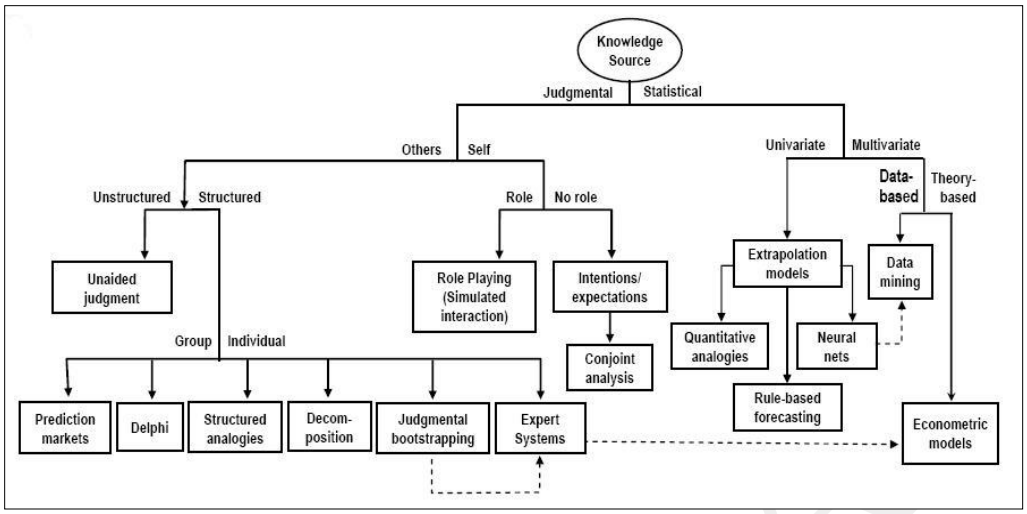
The elements that come into play in all forecasting methods is the concept of the future and time; uncertainty; and reliance on historical data. Below are some well-known quotes.

> *Forecasting involves making the best possible judgment about some future event. In other words, "forecasts are numerical estimates of an event for some future date that can be achieved with a specified level of support and are reproducible."*
> *"I often say that when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a very meagre and unsatisfactory kind."*
> *~ William Thomson, Lord Kelvin, 1824-1907*
>
> *"If we could first know where we are, then whither we are tending, we could then decide what to do and how to do it."*
> *~ Abraham Lincoln, 1809-1865*

Major Types of Forecasting Methods

- Subjective Methods
  - Sales Force Composites
  - Customer Surveys
  - Jury of Executive Opinions
  - Delphi Method

- Quantitative Methods
  - Exponential smoothing family
  - ARMA (Autoregressive-moving average) and ARIMA (Autoregressive integrated moving average)
  - Artificial Neural Networks (ANN)

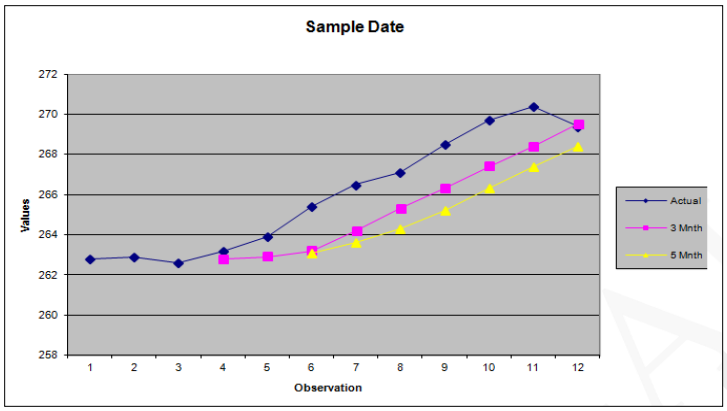


Moving Averages

The moving average approach calculates an average of the sample observations and then employs that average as the forecast for the next period. The number of sample observations included in the calculation of the average is specified at the start of this process. The term MOVING average means that as a new observation becomes available a new average is calculated by dropping the oldest observation in order to include the newest one.

Month	Period	Observed Values	3-Month MA	5-Month MA	Log Obs Values	Growth Series: 0.001%
Jan	1	262.8			2.420	2.410
Feb	2	262.9			2.420	2.412
Mar	3	262.6			2.419	2.415
Apr	4	263.2	262.8		2.420	2.417
May	5	263.9	262.9		2.421	2.420
Jun	6	265.4	263.2	263.1	2.424	2.422
Jul	7	266.5	264.2	263.6	2.426	2.424
Aug	8	267.1	265.3	264.3	2.427	2.427
Sep	9	268.5	266.3	265.2	2.429	2.429
Oct	10	269.7	267.4	266.3	2.431	2.432
Nov	11	270.4	268.4	267.4	2.432	2.434
Dec	12	269.4	269.5	268.4	2.430	2.437

Source: Business Forecasting Methods, by Jarrett, (Basil Blackwell, 1991).



Advantages:

1. Data requirements are small.
2. Better than using a simple arithmetic mean because it can be adjusted to reflect the observable patterns in the data.

Disadvantages:

1. The past n sample observations must be available.
2. Equal weights are given to all past observations and no weight is given to observations earlier than period t-n+1.
3. Assumes that the data has a stationary distribution (not always true).

Single Exponential Smoothing

Single parameter exponential smoothing (Unadjusted) is an easy to implement method of smoothing that overcomes some of the problems associated with moving averages. In contrast to moving averages, exponential smoothing permits the researcher to weight observations. It is not unusual for recent observations to contain more relevant information for forecasting purposes than older ones. The method also generates self-correcting forecasts through its ability to produce forecast values which reflect adjustment for earlier errors.

Advantages:

1. Simplifies forecasting calculations
2. Has small data requirements
3. Produces self-correcting forecasts with built-in adjustments that regulate forecast values by changing them in the opposite direction of earlier errors.
4. Simple! Only the last period's forecast must be saved.

Disadvantages:

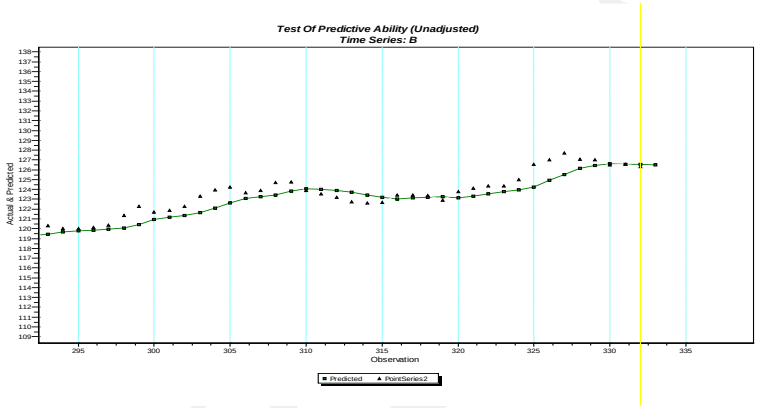
1. Specification of the smoothing constant is a problem. Alpha close to 1 implies that the new forecast includes a substantial adjustment for the error in the previous forecast. If alpha is close to zero, the new forecast will include only a small adjustment for error. Generally, it is suggested that if the smoothing constant is greater than 0.30 an alternative model should be used.
2. In general the forecasts trail the pattern in the sample data.

Notation for Single Unadjusted Exp. Smoothing

Notation	Description
$\mathbf{D_t}$	Actual value at time t
$\mathbf{F_{t+1}}$	Forecast value for time t+1
$\boldsymbol{\alpha}$	Smoothing constant ($0.0 \leq \alpha \leq 1.0$)

$$F_{t+1} = \alpha D_t + (1 - \alpha)F_{t-1} \text{ where } F_0 = D_1 \quad \text{or user input}$$

From the above equation it is apparent that there are two specific data input required for the unadjusted option. These are: the smoothing constant and the time series base (D_1)



Adaptive Rate of Response Single Exponential Smoothing (ARRSES)

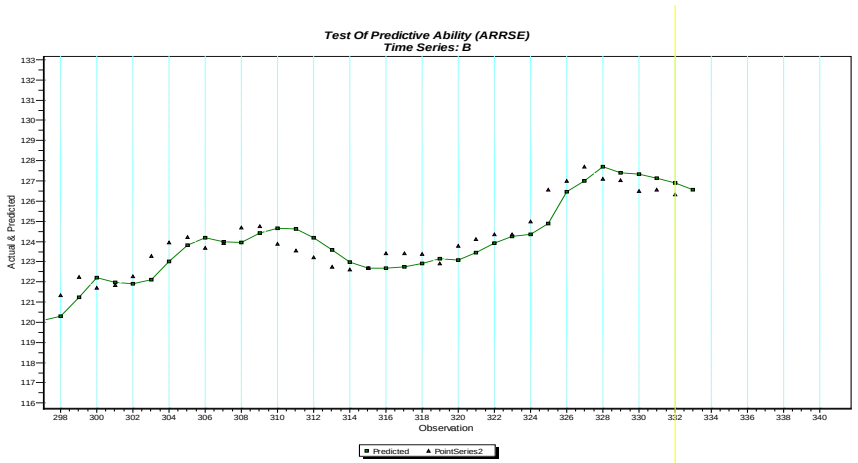
This method does not require the decision-maker to specify the alpha smoothing constant. ARRSES automatically changes the value of the unspecified alpha by a predetermined weight on an on-going basis; that is, whenever there is a change in data pattern. The only smoothing parameter that is needed is the Beta term. The Beta term is the weighting factor.

Advantages:

- Very useful when a large number of unique items have to be predicted.
- Excellent performance when there are a large number of observations (daily, monthly, etc.).

Disadvantages:

- Unknown smoothing constant.
- It may not be possible to replicate the solution.



Brown's Linear Exponential Smoothing (Double)

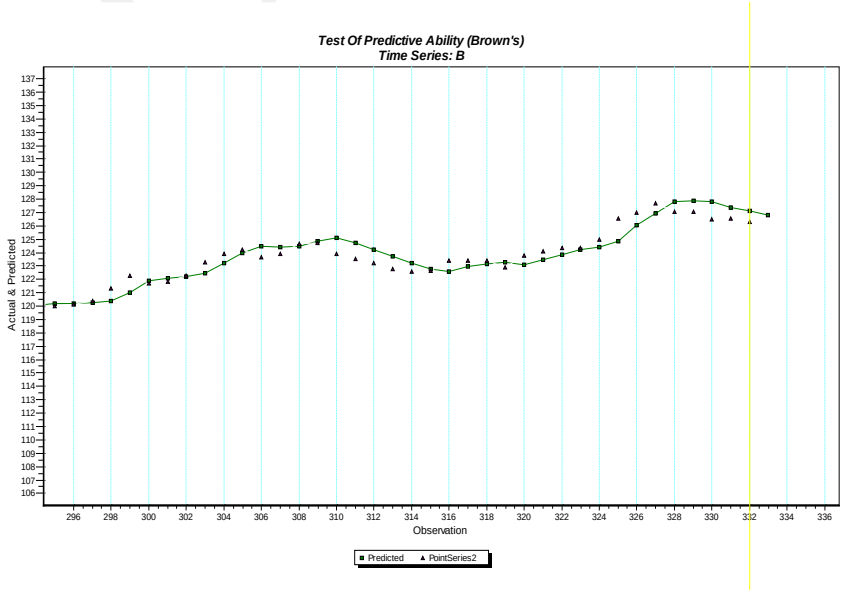
Double exponential smoothing is the application of exponential smoothing to the single exponential values. Brown's method provides an additional correction method; an approach which resembles the application of a moving average. Brown's method uses the difference between the single and double smoothed values as an additive factor to the single smoothed value. The method further adjusts for the pattern in the data.

Advantages:

- Provides an additional correction method for a data time series.
- Useful to account for linear trend in the data.

Disadvantage:

- The forecasts trail the pattern in the sample data.



Holts' Two Parameter Linear Exponential Smoothing

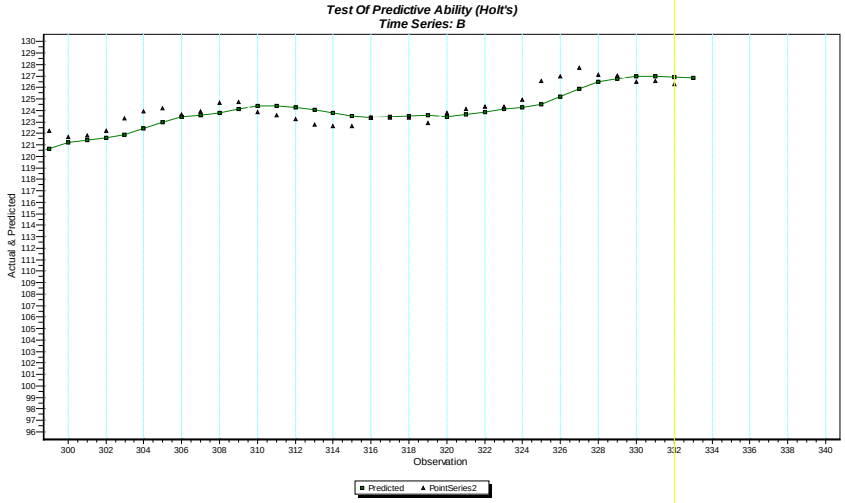
The Holt method is an extension to Brown's method. The Holt approach adds a growth factor to the smoothing equation. The method smoothes the trend values directly. When growth exists in the observed values of a time series, new observations will be greater than the previously observed values.

Advantages:

- Adds a growth factor to the smoothing equation.
- Trend values are smoothed directly (unlike the implied method in Brown)
- Eliminates the lag in smoothing.

Disadvantage:

- The forecast accuracy depends on determining the correct alpha and beta smoothing parameters.



End of Chapter 16