Internship Report

Implementation of a Python-R interface for the assessment of simulation models

An internship report presented in
partial fulfillment of the requirement
of the Professional Science Master's
in Computational Biosciences

Alisha Rossi
Computational Biosciences Program,
Arizona State University

Jeffrey W. White
Internship advisor
US Arid Land Agricultural Research Center (ALARC) of
the United States Department of Agriculture
Agricultural Research Service (USDA ARS)

Internship:
January 10th 2006 : May 10th 2006

NOT CONFIDENTIAL

Technical Report Number: 06-10
May 1st, 2006

ABSTRACT

Simulation models for agriculture and natural resource management are widely promoted as powerful tools for research and decision support. Before applying a model, users should understand how well the model represents the processes of interest and how accurate the outputs are likely to be. Many model validation procedures are based on comparison of observed and simulated data. Advances in statistical methods and computer capabilities offer numerous options for strengthening model validation. This report describes the development of an interface that permits the analysis of simulation models using statistical methods. A number of statistical methods are currently available, including regression techniques, such as linear regression (r^2, slope, and intercept), quantile regression, and the simultaneous F-Test. The PyRamid application has been created as a prototype for producing a convenient way for researchers to apply these regression techniques. PyRamid was developed in Python and exploits the RPy interface to R to execute R functions and access R's graphing capabilities through an easy to use graphical interface. Future work includes adding visual techniques and deviance measures, such as mean absolute error, root mean squared deviation, and modeling efficiency, into PyRamid's list of capabilities.

ACKNOWLEDGEMENTS

I would like to express my appreciation for the support and help of my internship advisor Dr. Jeffrey White. I would also like to thank Dr. Gerrit Hoogenboom for his helpful critique and review of my work. My research was funded by the Biological and Agricultural Engineering Department at the University of Georgia and supported by the US Arid Land Agriculture Research Center of the USDA Agriculture Research Service. I would also like to thank Dr. Rosemary Renaut and Dr. Zoé Lacroix for consenting to be on my committee and for their expedient and timely review of my work. I am deeply appreciative of the support and love of my husband Michael, and my sisters, Aja, Araina, Ariel, and Avia.

TABLE OF CONTENTS

1.0 INTRODUCTION

Simulation modeling is the development of a computerized mathematical model of a system (Chung, 2004). Simulation is among the most widely used operation- research and management science techniques available (Law and Kelton, 2000). Simulation models allow practitioners, from a variety of backgrounds, to analyze a large variety of systems or system conditions in less time and with reduced analytic requirements. Simulation results are usually in the form of statistics that can be used during validation. Validation is the process of determining whether the model provides an accurate representation of the real- word system. This report presents various validation techniques and introduces a tool that has been designed to facilitate the validation of simulation models.

The most appropriate simulation model validation method to use depends on several factors. The type of data is important when choosing a verification or validation method, but this is often overlooked (Kleijnen, 1999). Agricultural data usually appear in a form where the input (or trace) is known and is used to perform correlated inspection simulation (trace-driven analysis). The popular way to validate a trace- driven simulation is to make a scatter plot with real and simulated outputs, fit a line, and test whether the line has a unit slope and passes through the origin (Kleijnen, 1999). Commonly used tests of statistical significance and correlation measures such as r and r^2 are often used inappropriately when analyzing observed vs. simulated data (Willmott, 1982).

1.1 SIMULATION MODELS

A system is a collection of entities (i.e., crops, people or machines) that interact to perform a particular task. As outlined by Law and Kelton (2000), a system can be studied in different ways (Fig. 1).

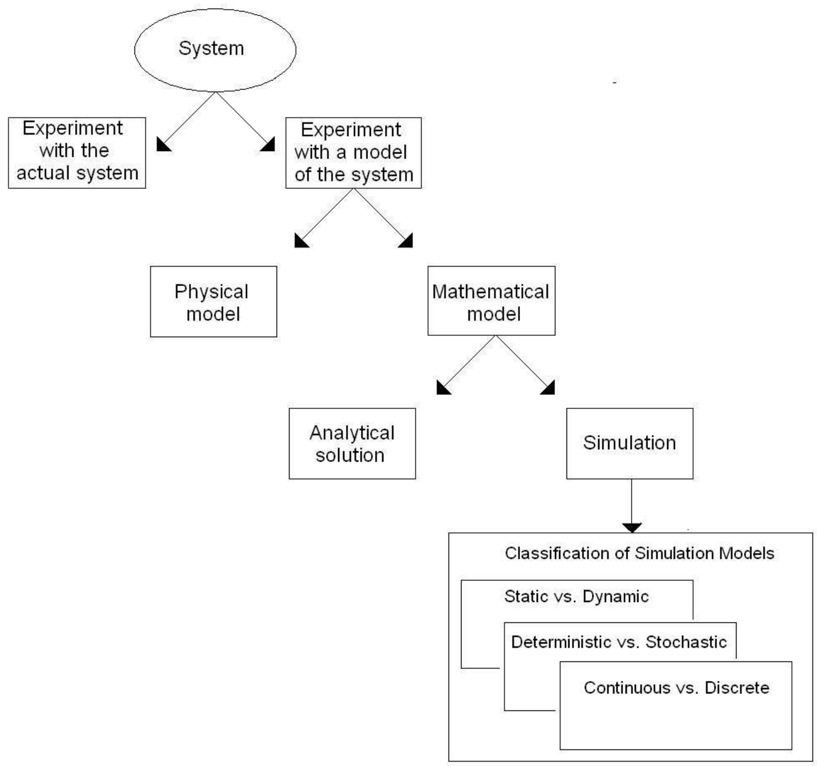

Figure 1: Ways to study a system (modified from Law and Kelton, 2000).

Validation is the process of determining whether a simulation model is acceptable for its intended use given the specified performance requirements (Rykiel, 1996). There is no question of validity when experimenting with an actual system, but experimental approaches may be too costly or too disruptive to the system to be feasible. When using a model, however, validity becomes a key issue because the model may not reflect the system for the purposes of the decisions to be made (Law and Kelton, 2000). Two types of models can be recognized, physical and mathematical. A physical model may be used in engineering and management systems, but for the vast majority of systems, mathematical models are the most useful. A mathematical model is defined as a set of equations that collectively describe the dynamics of the system (how the system reacts, or will react, under certain circumstances if the model is valid). Analytic methods can be used in the case of simple mathematical models, but in more complex systems, analytic solutions are rarely feasible and simulation is required.

1.1.1 Classification of a simulation

Simulation allows for the analysis of a system in less time and with reduced analytic requirements. Performing a simulation is defined as the process of creating and experimenting with a computerized mathematical model of a system by numerically excising the inputs to analyze the affect on system output (Chung, 2004; Law and Kelton, 2000). The simulation can be further classified by answering the following three questions:

Classification of a Simulation

1. Is the model a representation of a system at a given time (static simulation) or does the model represent the system as the system evolves (dynamic simulation)?
2. Does the model contain probabilistic components (stochastic simulation) or not (deterministic simulation)?
3. Do the state variables change instantaneously at separate points in time (discrete simulation) or continuously with respect to time (continuous simulation)?

The first question, asks whether the model is *static* or *dynamic*. In Monte Carlo Models, random numbers are used for solving certain problems where the passage of time plays no substantial role. Most simulations involving a Monte Carlo Model would therefore be identified as static. Conversely, when time does play a role in the model, the simulation is referred to as dynamic. The second question asks whether the model is *stochastic* or *deterministic*. In a complicated system of differential equations, the output is "determined" once the set of input quantities and relationships have been identified. A simulation of this system would therefore be identified as deterministic. A model containing at least some probabilistic components, however, would be recognized as stochastic. The third question asks whether the model is *discrete* or *continuous*. Continuous simulations often involve differential equations that give relationships for the rates of change of the response variable with time. These models can be investigated analytically or with numerical analysis methods such as Runge- Kutta integration. When the modeling of a system does not concern variables that change continuously with respect to time, the simulation is considered discrete. With this definition, a discrete model is not always used to describe a discrete system.

This report focuses on simulations of agricultural systems using models that typically are dynamic, continuous, and deterministic. If multiple years of weather data are used as inputs or other inputs are obtained through sampling, the models can be viewed as stochastic. Although simulations that are dynamic, continuous, and deterministic could conceptually be performed using hand calculations, the amount of data that must be manipulated in most real- world systems requires the use of a computer. The importance of validating a simulation model can be seen when examining the steps in a simulation study.

1.2 VALIDATION OF A SIMULATION

The simulation life cycle, provided in Figure 2, illustrates that conceptual model and operational (program) validation steps must occur before the model can be used for its intended purpose. Without validation, the utility of any results produced by a simulation model cannot be judged, thus, validation is a fundamental component of any simulation process. The first step in the simulation life cycle is to formulate the problem and plan the study. The overall objectives and scope of the model must be specified in this initial step. Next, the data is collected and the model is defined. Collecting high quality information and data on a system helps to define the problem. If the conceptual model is shown to be invalid, the model must be redefined after additional data collection on the existing system. Data collection is important, because an accurate model can appear invalid if the inputs have large errors.

A conceptual model is the mathematical, logical, or verbal representation of the problem (Sargent, 2000). In parallel, the conceptual model mimics reality within a limited set of assumptions. During *conceptual model validation*, the assumptions of the model are reviewed. If the assumptions appear correct and complete, programming will begin. The program is the conceptual model translated onto a computer. Verification of the program is then performed in order to determine whether the model assumptions and mathematical formalisms are correctly translated into the computer program (Rykiel, Jr. 1996).

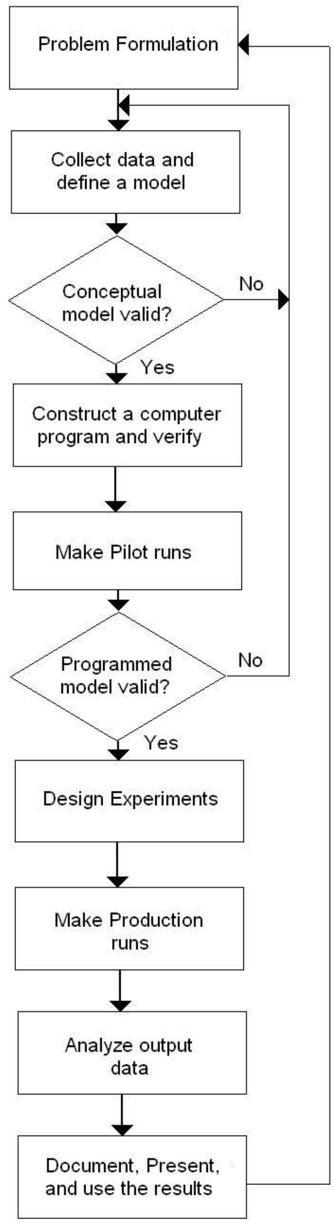

Figure 2: The Simulation Life Cycle (modified from "Steps of a simulation study," Law and Kelton, 2000).

Verification can include, for example, the analysis of model components, visualization techniques, or debugging. After several runs of the computer program, the accuracy of the computerized model output is analyzed in a process called *operational validation*. Regression techniques used in operational validation are the focus of this report.

After experimentation and several production runs, the output is analyzed. Output analysis is used to estimate a model's true measures of performance (i.e., simulation run length and warm up time). Law and Kelton (2000) provide a contrast between validation and output analysis when discussing the mean of the system (μ_S) and the mean of the simulation (μ_M). After a simulation run, an estimation of the mean of the simulation is obtained ($\hat{\mu}_M$). The error of the simulation mean estimate is given in the following equation:

$$\text{Error in } \hat{\mu}_M = |\hat{\mu}_M - \mu_S|$$

$$= |\hat{\mu}_M - \mu_M + \mu_M - \mu_S|$$

$$\le |\hat{\mu}_M - \mu_M| + |\mu_M - \mu_S| \quad \text{(by the triangle inequality)}$$

Output analysis is concerned with minimizing the first absolute value, whereas validation is concerned with minimizing the second (Law and Kelton, 2000).

It is important to note that validation is not required for the initial development or exploration of the model (i.e., problem formulation). Models can initially be defined with the wrong elements, and these elements can be removed as the development and understanding of the system improve. An overemphasis of model validation can stifle model development, and thus, validation should only limit the simulation process when it is required during the conceptual model validation and operational validation steps.

1.2.1 Development of a Tool

Here we describe the development of a tool that can be used to perform linear, quantile, and multiple regressions. As an initial test case, a classic model for grain yield as a function of crop water use was examined (French and Schultz, 1984). A heterogeneous response distribution with unequal variances results in a mean of the measured factors (water use) having little predictive value for the mean of the response variable (grain yield) distribution. In other words, the dataset provides an example of when least squares regression and correlation methods fail to give a complete understanding of the effect of the variables involved. Other parts of the response variable distribution, however, provide more useful predictive relationships. In our report, we will describe how these possible causal relationships can be provided by estimating the conditional quantiles of the grain yield distribution, an approach called quantile regression. Agriculturalists and other scientists can perform quantile regression using our tool.

2.0 LITERATURE REVIEW

There is considerable confusion about the meaning of validation in the context of simulation models. Validation is an integral part of the simulation building process, and yet there are no standards for validation criteria or even on the meaning of validation itself. The scientific and semantic differences between model builder and model user terminology make model credibility difficult to assess. The first step in model validation is to settle on a single definition. This section begins with a definition for validation and related terms. The second section provides a detailed look at how three specific statistical validation methods are utilized in the assessment of simulation models, including: visual techniques, regression techniques, and deviance measures.

2.1 GENERAL OVERVIEW OF VALIDATION

This report focuses on building a tool for statistical validation. Hence, it is important to understand the terminology related to validation and what statistical validation is. Statistical validation techniques can be used for conceptual and programmed models. This section provides a summary of the terms and techniques associated with general and statistical validation, primarily for the programmed model case.

2.1.1 Verification, Calibration, and Qualification

Validation is the process of determining whether the model is an accurate representation of the real- word system. The purpose of validation is to build model *credibility*. Credibility is the degree of belief in the validity of a model which is a subjective, qualitative judgment referring to whether the manager or other project personnel accepts the model as correct. Validation can also be applied to *calibration* and *qualification*. The parameters and constants of a model are typically given values in order to produce numerical results. Calibration is the estimation and adjustment of model parameters and constants to improve the agreement between model output and a data set (Rykiel, Jr., 1996). A model is only valid over the domain for which it has been validated; therefore, it is important to describe the conditions under which a model has been validated. Qualification is aimed at discovering this domain by revalidating the model for new cases.

The Department of Defense (DoD) is the largest user of modeling and simulation applications in the world. Balci and Ormsby (2000) proposed that the DoD should recognize three processes in assessing a given model: verification, validation, and accreditation, VV&A. *Accreditation* is an official determination, as given by the DoD, that a simulation model is acceptable for a specific purpose. The evaluation for accreditation is usually conducted by a third party. While ultimately a subjective decision, accreditation often includes formal documentation of model verification, and conceptual and programmed model validation.

2.1.2 Conceptual vs. Programmed Model Validation

The form of validation described in this paper is statistical validation. Statistical techniques can be used during both conceptual model validation and programmed model (operational) validation. Tests of statistical significance are often subject to assumptions, and these assumptions must be validated. The F-test for lack of fit could be applied to determine whether a linear model is appropriate to use, and the omission of important predictor variables could be assessed by plotting residuals against the omitted variable and checking for whether the residuals vary systematically (Neter et al., 1996). In the case of least squares regression, normality, homogeneity of variance and independence are assumed, and each assumption must be validated. Normality can be tested by preparing a normal probability plot of the residuals. Non-normality and lack of constant error variance often go hand in hand;

homogeneity of variance can be tested using the Modified Levene test or Brusch- Pagan test (Neter et al., 1996).

In data from agricultural research, error terms are often *autocorrelated*. For time series data, lack of independence can be determined by plotting the error terms against time; if a positive relationship exists, then the data is autocorrelated. This has a number of consequences on using the least squares approach (Neter, et. al, 1996):

Problems Using Least Squares Approach on Autocorrelated Data

1. Estimated regression coefficients no longer have the minimum variance property.
2. MSE may seriously underestimate the variance of the error terms.
3. The standard deviation, $s\{b_k\}$, may seriously underestimate the true standard deviation of the estimated regression coefficient.
4. Confidence intervals and tests using the F distribution are no longer strictly applicable.

Although the problem of autocorrelation can be revealed during conceptual model validation, there is no simple solution. The problem usually persists throughout the entire simulation process. When autocorrelated data are present, the two principal remedial measures are to add one or more predictor variables to the regression model or to use transformed variables. This depends on the cause of the autocorrelation. Often, a major cause of autocorrelation of the error terms arises from omission of one or more key predictor variables. In agricultural settings, it may be difficult to capture all of the long-term persistent effects in a response variable, and a trend component can be added to the model (such as the use of an indicator variable for seasonal effects). For this reason, autocorrelation can influence the outcome of p*rogrammed model validation*, or *operational validation*. Programmed model validation is comparing real-world observations and simulated output. There are four main categories that can be used in programmed model validation, namely, subjective assessment, visual techniques, deviance measures and statistical tests (Mayer and Butler, 1993).

2.2 STATISTICAL VAIDATION OF A SIMULTAION

Statistical validation techniques used in programmed model validation are the main focus of this report. Visual techniques and deviance measures are briefly discussed. For a more detailed look at other validation procedures (i.e., face validity and Turing techniques) see Rykiel (1996) and Sargent (2000).

2.2.1 Visual Techniques

Time series plots can be used as the basis for comparison between system and model. The most widely used visual technique is to plot observed data (usually as discrete points) and simulated data (usually as a continuous line) against a common independent variable (time). This method, however, is not ideal because it does not relate the observed data to the 'perfect fit' line; it equates the observed values to a recalibration of the model (Mayer and Butler, 1993). A preferred visual technique for operational validation is to plot the observed vs. predicted data.

2.2.1.1 Plotting Observed vs. Simulated Values

Figure 3 shows a plot of observed vs. simulated days to anthesis, which indicates a strong positive relationship. The y = x (or 1:1) line is included as a visual guide to judge bias and goodness of fit. Different plot symbols are used for each stratum to indicate possible clustering of the data, which may reflect lack of independence.

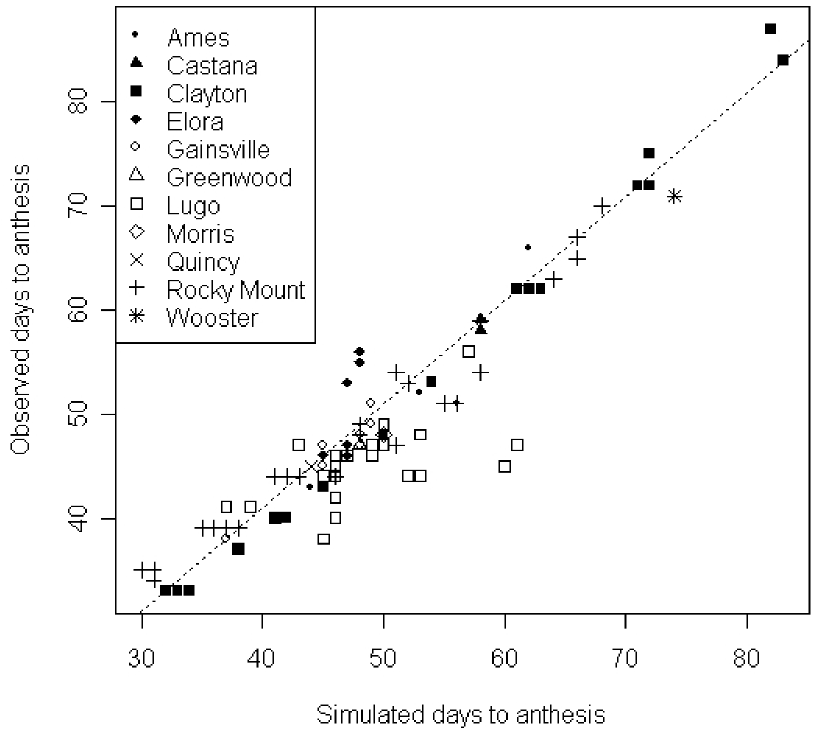

Figure 3: Observed vs. Simulated Days to Anthesis. The Lugo data appear slightly below the 45-degree dotted line which could be an early indication of bias or unexplained error. The data was taken from the CMS-CROPGRO-Soybean model version 4.0.2.0 (Jones et al. 2003, Hoogenboom et al. 2004). The plot was created using the R language and environment for statistical computing (R version 2.3.0 for windows).

In this example, a different symbol was used for each location where data were collected. There is a possible under-prediction (bias or unexplained error) in the data obtained from Lugo, Spain. Also, fewer data points appear between the 70 to 80 day intervals than for less than 70 days. This shows that the application of visual techniques can be useful prior to the application of regression techniques for two reasons: (1) it may indicate that a certain statistical test should be applied, such as, the simultaneous F-test for bias; (2) it may point out insufficient data sampling, indicating that more data should be collected or that validity for some locations has greater uncertainty.

2.2.2 Regression Techniques

There are two types of tests for statistical validation that will be discussed in this section: (1) the model produces output that has the same statistical properties as the observations obtained form the real system; and (2) the error associated with the critical output variables falls within specified or acceptable limits. The first type is referred to as "lack-of-fit analyses" and will be addressed in the following discussion of regression techniques. The second will be addressed in the discussions of deviance measure techniques (Section 3.1.2).

2.2.2.1 Linear Regression

The linear regression approach is often used when assessing the "goodness of fit" between observed vs. simulated values, or between the output variable against one or more predictor variables. The equation for linear regression is: $y_i = b_0 + b_1 x_i + \epsilon_i$, where y_i is the observed data value and x_i is the corresponding simulated value or a single predictor variable. The b_0 and b_1 values estimate the unknown coefficients ($\beta_0 + \beta_1$) in the "true" equation.

Regression analysis of the output variable on a predictor variable tests whether a relationship exists between the two. No linear relationship exists when the slope is zero and a strong linear relationship exists when the slope is near one. The statistical test for the relationship of the output variable against one or more predictor variables is the F-test which is provided in Section 2.2.2.3. The test criteria, however, can be too severe. The proportion of variation that is explained should be high, but agricultural data will often deviate from the 45-degree line. A 95% confidence interval provides a less restrictive alternative. Additionally, less restrictive hypotheses can be used (Rose, 1995).

For linear regression of the output variable against the predicted variable, the slope should be equal to one and the intercept should be zero, indicating a "perfect fit". The statistical test for this case is the simultaneous F-test. These criteria may also be too severe. Theoretically, the expected slope of the relationship on observed vs. predicted data is actually less than one, and the expected intercept is greater than zero (Harrison, 1990). This will result in a systematic departure below the 45-degree line in the plot of observed vs. simulated values. When the assumption of independence is invalid, estimates of the model parameters and associated statistics may be biased. Many validation datasets are time-series autocorrelated, including most farming systems models. Mayer and colleagues (1994) suggested that averaging the subsequent pairs (or triplets, quadruplets, etc.) within the time series will minimize correlations.

This, however, usually results in rejection rates as high as 20%, which is considerably higher than the 5% rate that is often expected (Mayer, et al, 1994). In their Monte Carlo study, Mayer and colleagues showed that the autocorrelated data results in high rejection rates (as high as 47%) for valid models. For some cases, however, autocorrelation results in an inflated correlation coefficient (r^2), causing the slope to appear closer to one. This can cause an invalid model to appear valid. Therefore, for time series autocorrelated data, it is important to estimate the magnitude of the deviation from the mean (bias), as the actual the degree of the relationship may be evaluated poorly.

The magnitude of the deviation from the mean, however, should be considered for all cases, not just for models with time series autocorrelated data (Rose, 1995). If error exists in the observed (y) values, even with a perfect model (exclusive of the error), the slope will be less than one, and the more error that exists, the lower the slope will be (Rose, 1995; Kleijnen, 1999). Mayer et al. (2004) explains why the observed values should be taken as the response variable (y). To illustrate, a dataset was obtained from the CMS-CROPGRO-Soybean model version 4.0.2.0 (Jones et al. 2003, Hoogenboom et al. 2004), and the residuals vs. observed values and residuals vs. simulated values were plotted. In the plot of residuals vs. observed values (Fig. 4), the residuals become more positive as the observed values increase. This trend is not seen in the plot of residual vs.

simulated values, because the simulated values were created in a way that calculatedly causes no systematic tendencies in the residuals.

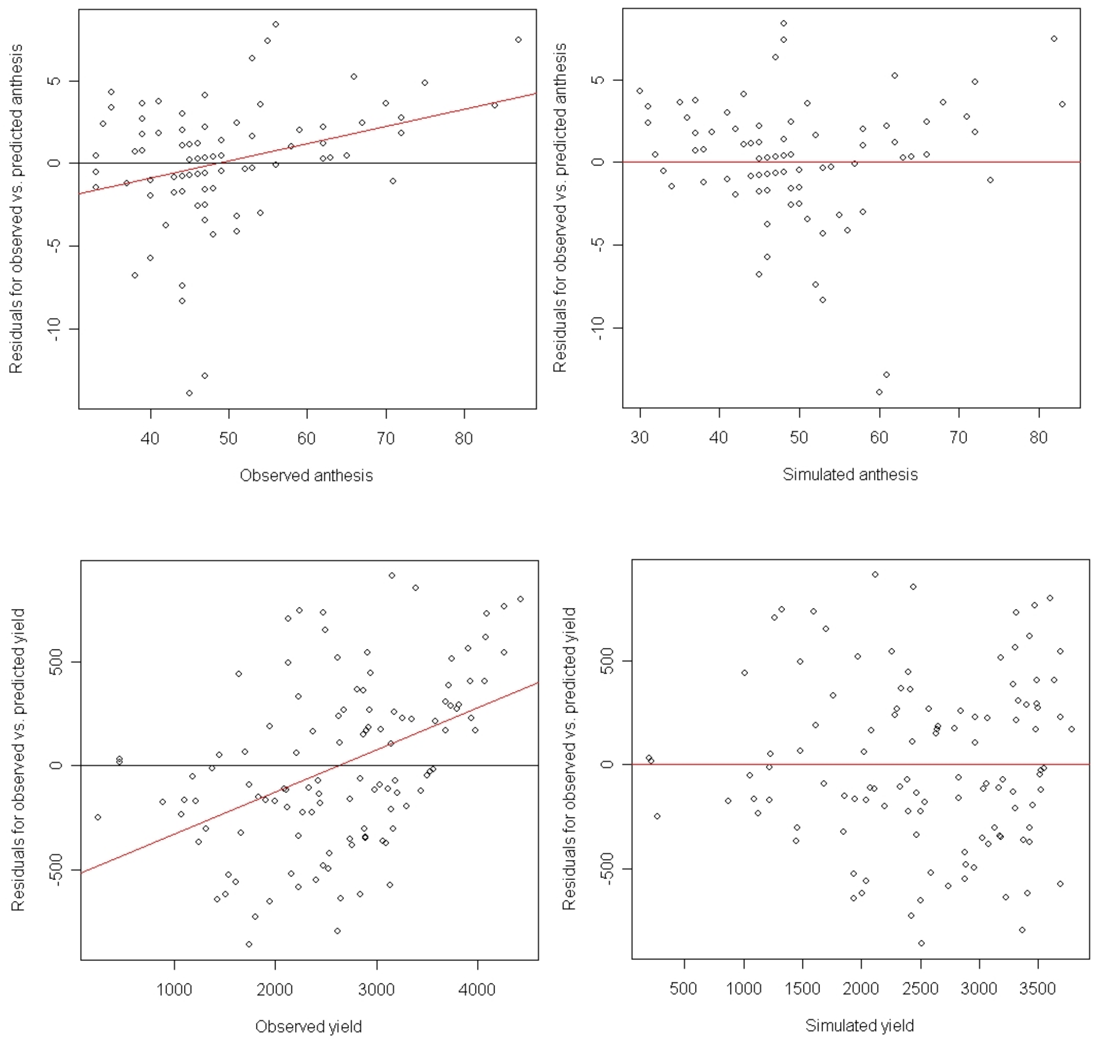

Figure 4: Residuals vs. observed and residuals vs simulated plots for days to anthesis and grain yield data obtained from the CMS-CROPGRO-Soybean model version 4.0.2.0 (Jones et al. 2003, Hoogenboom et al. 2004). The plots were created using the R language and environment for statistical computing (R version 2.3.0 for windows).

Mayer et al. (1994) explained that the relativity between overall variation of the x-data and the error they contain (σ^2) can be theoretically approximated as:

$$E[b_1] = 1 - \frac{v_1}{v_1 + v_2} = \frac{1}{1 + \frac{v_1}{v_2}}$$

The variance of the x means is given as:

$$v_0 = \sum \frac{(\mu_{xt} - \mu_x)^2}{n} \text{, where } e_{xt} = x_t - \mu_{xt}$$

$$v_1 = E\left[\sum \frac{(e_{xt} - e_x)^2}{n}\right] = \sigma^2 \left(1 - \frac{1}{n}\frac{1+p}{1-p} + \frac{2}{n^2}\frac{p(1-p^n)}{1-p^2}\right)$$

When the x data contain no random variation, $v_1 = 0$ and $E[b_1] = 1$. Thus, for a valid model, the slope of the line should be equal to one and the amount of variation should be near zero. This supports the use of a simultaneous F-test for observed vs. simulated values. The evidence does not, however, support the opposite case. The variation of the error terms for the y-data is calculated in a way that produces consistent variation around the mean ($v_1 = 0$, regardless of the appropriateness of the model), and thus provides no testable hypotheses.

Although this visual technique can be used to illustrate the deviation of the residuals of the x and y values from zero, it provides no real suggestions on how to improve the model. The observed values in Figure 4 do not deviate in a way that indicates linear regression is inappropriate to use. The appropriateness of linear regression can be seen using other residual plots (i.e., standardized residuals vs. fitted value and other diagnostic plots).

In summary, the mean of the response variable distribution as a function of a set of predictor variables is given by linear regression. It has been suggested that regression is not ideal for validating simulated models because the fitting of the model to its measurement is not of concern; it is the comparison of calculated values and measured values that is important (Kobayashi and Salam, 2000). It is also important, however, to know about portions of variation that are explained. For this reason, regression may be useful in the validation process. Linear statistical methods should be used as descriptive techniques rather than inferential devices. These methods should be combined with other techniques (visual, and deviance measure), when assessing the validity of a model.

2.2.2.2 Quantile Regression

In the 1970's, quantile regression was developed by econometricians (Koenker and Bassett, 1978) as an extension of the linear model. The quantile regression approach assumes no particular parametric form for the error distribution (i.e., binomial). The quantile (τ) of a dataset is a value with an approximate fraction of the data less than or equal to that percentile (Koenker and Hallock, 2001). The sample median, (corresponding to τ = 0.5), is a measure of the center of a distribution. It is the middle value of the ordered data. The difference between the upper and lower quartiles, or interquartile range, is a measure of *variation*. The quantile plot (for $x_1, x_2, \ldots x_i, \ldots x_n$) is the graph of x_i vs. f_i, where $f_i = i/(n + 1)$. Quantile regression has been recommended for estimating limits in various types of ecological analyses (Scharf et al., 1998), but has not been applied to validation of agricultural models.

In modeling crop production, a common problem is that many complex factors can reduce growth or yield below the potential level that the model describes. For example, soil borne diseases may limit water and nutrient uptake causing a crop to suffer water and nitrogen deficits that ultimately limit growth and the final economic yield. The problem for model validation is that the observed data are biased toward lower values than the simulated values. Thus, in the validation process, the expectation is that a model will describe the upper limit of the observed values. Therefore, the statistical problem is to estimate this upper limit rather than the mean tendency.

For the French and Schulz dataset, the heterogeneous response distribution with unequal variances results in a mean measured factor (water use) with little predictive value for the mean response variable (grain yield) distribution (Fig. 5a and 5b). A more complete view of the relationship between grain yield and water use can be seen by estimating the rates of change of all parts of the distribution of the response variable using quantile estimates (gray) rather than limiting the focus to the mean and its corresponding confidence intervals.

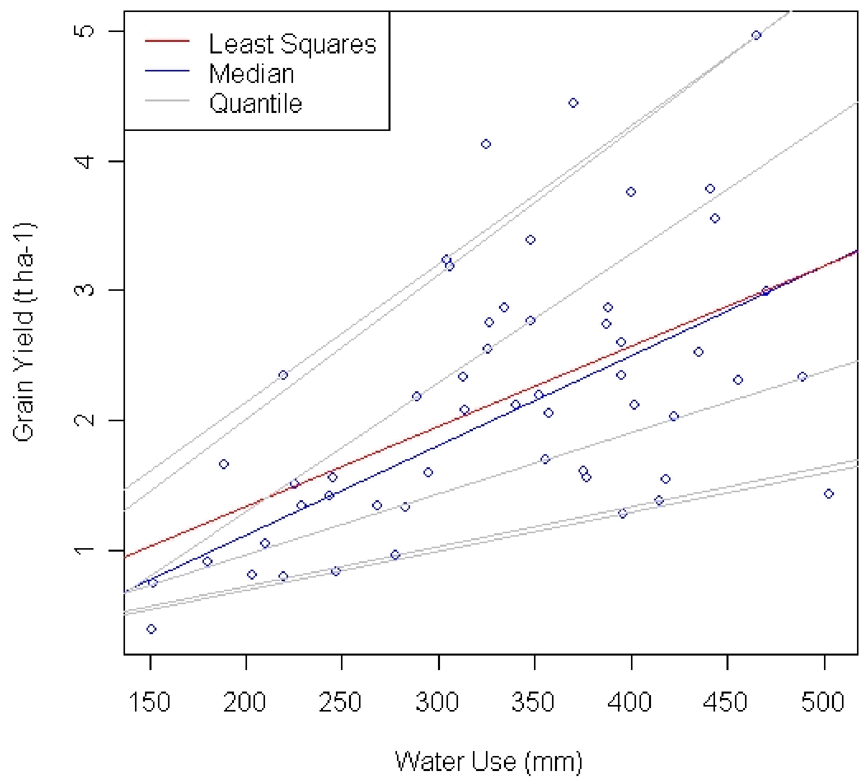

Figure 5a

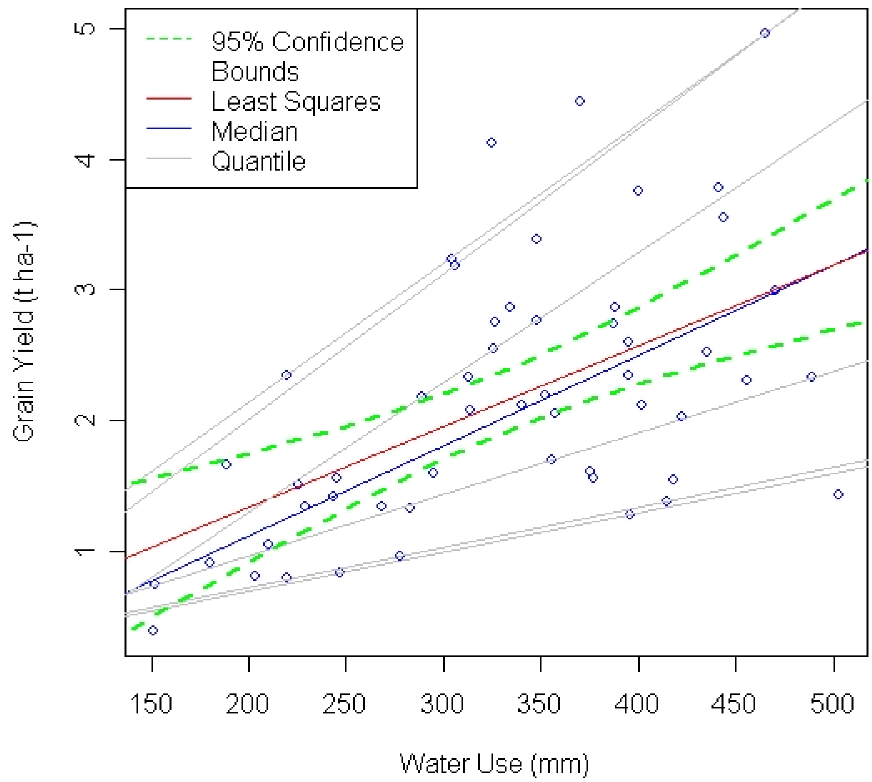

Figure 5b

Figure 5a: The relation between yield of wheat and water use (soil water plus rainfall). Superimposed on the plot are the {τ = 0.05, 0.10, 0.25, 0.75, 0.90, 0.95} regression quantile estimates (gray), the median fit (blue), and the least sq. estimate in red. Figure 5b: Quantile regression plot superimposed with 95% confidence intervals (green) and least the squares regression line (red). The variation in the output variable is better represented by the quantile regression lines than by the 95% confidence bounds. Both plots were created using the R language and environment for statistical computing (R version 2.3.0 for windows).

Heterogeneous response distributions with unequal variances are commonly seen in both ecological and agricultural data. Although quantile regression is becoming more commonly applied in ecological research (Scharf, 1998), the method is seldom used in the agricultural community. This is perhaps due to the familiarity and convenience of approaches such as the least squares method.

Quantile regression is discussed here as a method of explaining the variation within a model that contains bias, which appears to be of marginal use. This technique was, however, explored as a possible alternative to the current validation methods. For example, for the model $y_i = b_0 + b_1 x_i + e$, consider the case where the y_i values are affected by some undeterminable bias. This bias will reduce the y_i - values by a factor between 1 and 0, that we will call B_i. This bias would be represented in the model as

$$B_i y_i = b_0 + b_1 X_i + e_i,$$

and thus, the equation becomes:

$$y_i = \frac{b_o + b_1 x_i + e_i}{B_i}$$

One trouble with this equation is that bias would be difficult to determine as B approaches zero. So suppose the bias reduced the y_i- values simply by an indeterminable amount and not a factor. Then the equation,

$$B + y_i = b_0 + b_1 X_i + e,$$

could be written as:

$$Y_i = (b_0 - B_i) + b_1 X_i + e_i$$

This equation, however, is useless without information on this bias B_i. The quantile regression approach might then be used to estimate B_i. For example, by comparing the upper and lower quantiles of a real- world system to the upper and lower quantiles of a simulated system, it might be feasible to obtain a value for the bias.

2.2.2.3 ANOVA and F-Test

The Analysis of Variance (ANOVA) approach to regression analysis is based on the partitioning of the sums of the squares of the degrees of freedom with respect to the response variable. The differences that exist between each response value (y_i) and the overall mean ($\bar{y}$) are conventionally referred to as *variation*. The measure of total variation is described by the sum of the squared deviations from the mean, or the *total sum of squares* (SSTO).

$$\text{SSTO} = \sum (y_i - \bar{y})^2$$

The Analysis of Variance derives its name from the fact that the quadratic form of the SSTO is decomposed into component parts and analyzed. The formulas for these components, the *error sum of squares* (SSE) and the *residual sum of squares* (SSR), are given as follows:

$$\text{SSR} = \sum (\hat{y}_i - \bar{y})^2$$

$$\text{SSE} = \sum (y_i - \hat{y}_i)^2$$

In Figure 5a, water use is the independent variable, and observed grain yield is the dependent variable (y). The y-hat ($\hat{y}_i$) represents the points on the least squares regression line (red). The mean of the observed values ($\bar{y}$, “y-bar”), does not appear on the graph, but if it did, it would appear as a horizontal line going through the point where the median line and the least squares regression line meet. Thus the SSR represents the deviation form the regression line to the horizontal line for the mean of the y-values, and the SSE represents the vertical variation of the data points from the regression line along the y-axis.

SSTO has n-1 degrees of freedom associated with it. The degrees of freedom of the SSR are given by the number of parameters minus one (p-1). The number of data values minus the number of parameters (n-p) represents the degrees of freedom associated with the SSE. The mean squared regression (MSR) is the SSR divided by its associated degrees of freedom; the mean square error (MSE) is the SSE divided by its associated degrees of freedom. For the simple linear regression case, analysis of variance provides us with a test where null (H_0) and alternative (H_a) hypothesis are:

$$H_0: \beta_1 = 0$$

$$H_a: \beta_1 \neq 0$$

The test statistic for this approach is denoted by F* and is defined as follows:

$$F^* = \frac{\dfrac{\sum(\hat{y}_i-\bar{y})^2}{p-1}}{\dfrac{\sum(y_i-\hat{y}_i)^2}{n-p}} = \frac{\dfrac{SSR}{p-1}}{\dfrac{SSE}{n-p}} = \frac{MSR}{MSE}$$

Low values of F* report H_0 (no linear relationship exists between x and y), and values of F* near 1 report H_a (reject the null hypothesis). A one- way ANOVA (a.k.a. simple ANOVA, single classification ANOVA, univariate ANOVA, or one- factor ANOVA) can be performed in a situation where there is one dependent and one independent variable. A one- way ANOVA tests whether the groups formed by the categories of the independent variable have the same pattern of dispersion by measuring the group variances. If the groups are different, then it is concluded that the independent variable has an impact on the dependent variable. In the two- way ANOVA for $x_1, x_2, \ldots, x_i, \ldots x_n$ the null and alternative hypotheses become:

$$H_0: \beta_1 = \beta_2 = \beta_3 = \ldots = \beta_i = \ldots \beta_n = 0$$

$$H_a: \text{At least one } \beta_i \neq 0$$

These tests are useful for determining which variables are needed to explain the overall variation. The goal of a model builder, however, is to accept a null hypothesis of no difference between real world and modeled data with a known level of confidence. When investigating observed vs. simulated values, the previous hypotheses are inappropriate. For the simple linear regression case of observed vs. simulated data, we are testing to see whether observed values and predicted values vary together constantly over their ranges. The other test of interest is the test of zero intercept. The null and alternative hypotheses become:

$$H_0: \beta_1 = 1 \text{ and } \beta_0 = 0$$

$$H_a: \beta_1 \neq 1, \text{ or } \beta_0 \neq 0, \text{ or both}$$

This is referred to as the *simultaneous F test*. According to Mayer et al. (1994), the statistic for this test becomes:

$$F^* = \frac{nb_1^2 + 2\sum x_i b_1 (b_2 - 1) + \sum x_i^2 (b_2 - 1)^2}{2s^2}$$

The standard assumptions for regression are made, including: (1) the samples are normally distributed, (2) the samples are independent, and (3) the variances are homogenous. As discussed earlier, many validation sets are time series autocorrelated, and using the F-test on observed vs. simulated data can have detrimental effects. Two effects are known to occur, even in data with little or no autocorrelation, which include: (1) models with a high correlation coefficient (r) have smaller rejection percentages; and (2) increasing the sample size of n increases the likelihood of a false rejection of the null hypothesis (Harrison, 1990; Thornton and Hansen, 1996). Because this analysis tends to reject valid simulation models, and smaller rejection percentages occur in models with a high correlation coefficient (resulting in not rejecting invalid simulation models of this type), other statistical techniques should be shown to produce comparable results before rejecting or accepting a model as valid.

2.2.3 Deviance Measures

There are several deviance measure techniques that can be used in programmed model validation. This section reviews three commonly used measures, namely, Modeling Efficiency (EF), Root Mean Squared Deviation (RMSD), and the Mean Absolute Error (MAE).

2.2.3.1 Modeling Efficiency

A dimensionless statistic which is said to directly relate model predictions to the observed data is *modeling efficiency* (Mayer and Butler, 1993). Modeling efficiency (EF) is a simple index of performance.

The equation for EF is given as follows:

EF = $$\frac{(\text{SS about } y = \hat{y})}{(\text{Corrected SS of } y)} = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y}_i)^2}$$

It can be seen that EF is analogous to the coefficient of determination:

$$r^2 = 1 - \frac{(\text{SS about the line of bestfit})}{(\text{Corrected SS of } y)} = \frac{SSR}{SSTO} = 1 - \frac{SSE}{SSTO} = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y}_i)^2}$$

The coefficient of determination is interpreted as the proportion of variation explained by the fitted regression line. Since $0 \le SSR \le SSTO$, it follows that $0 \le r^2 \le 1$. The modeling efficiency, however, can be negative ($-1 \le EF \le 1$) because the data are compared with a fixed line. For EF, 1 indicates a perfect fit, 0 reveals that the model is no better than a simple average, and a negative value indicates a poor model (Vanclay and Skovsgaard, 1997).

2.2.3.2 Root Mean Squared Deviation

It has been previously noted that the relationship between r^2 and model performance for linear regression are not consistent, and a linear relationship between the simulated value and the mean of the observed value must be assumed (although this is not always certain), and thus, approaches based on mean squared deviation (MSD) and related measures have been described as the best overall measures of model performance (Willmott, 1982; Kobayashi, 2000). The mean deviation (MD), sometimes referred to as *mean bias*, is given in the following equation:

$$\text{MD}=\frac{1}{n}\sum_{i=1}^{n}(x_i-y_i)$$

The observed value is the response variable (y) and the simulated value is the predicted variable (x). The difference between a simulation and its measurement can be calculated as the mean squared deviation.

$$\text{MSD}=\frac{1}{n}\sum_{i=1}^{n}(x_i-y_i)^2$$

The mean squared deviation is the square of the root mean squared deviation (RMSD). The equation for the RMSD is given as follows:

$$\text{RMSD}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-y_i)^2}$$

The MD provides a measure of bias, whereas the MSD or RMSD can be used to determine the variability of observed values from simulated values. The lower the MSD value, the closer the simulation is to its measurement.

2.2.3.2 Mean Absolute Error

The RMSD, also known as root mean squared error (RMSE), is sensitive to extreme values. The mean absolute error (MAE) has been suggested as an alternative (Willmott, 1982). MAE is less sensitive to extreme values because it does not weigh each difference (observed – simulated) by its square, but instead substitutes the absolute value of the difference.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |x_i - y_i|$$

No single criterion can incorporate all aspects of statistical validation. It is desirable to have an array of easy to use tests. Modeling efficiency and MSD-based analysis are two methods that provide an alternative to linear regression with results that are easy to interpret.

3.0 Tool Development

Python was chosen for the user interface for various reasons. The DSSAT group wished to move from strictly Windows-based software to platform independent applications. Furthermore, they sought a language that would facilitate providing open source code. Python is platform-independent and is widely used in open source development (Lutz, 2001). R was chosen as the statistical package primarily because it combines advanced statistical methods with powerful graphical display capabilities (Maindonald and Braun, 2003). A further incentive is that an initial survey of options revealed that a Python library existed for calling R functions, namely, Rpy.

Using RPy, R objects were managed and R functions were executed in Python. All errors in the code were converted to Python extensions. Code that was too difficult to convert, or not strictly applicable, were left in a script form that could be may be added into the PyRamid program in future work. Besides R and Python, four libraries (RPy, Pywin, NumPy, and PMW) were required to build the validation tool prototype. Tool development began with the creation of scripts. Scripts were combined into one program called PyRamid. The different features of PyRamid, and a display of its capabilities, are given in the following section.

3.1 Creation of Scripts

Window gadgets, termed widgets, were created using the Tkinter GUI development option (Lutz, 2001). Tkinter is an open- source portable GUI library that is used as the standard for development in Python. The underlying library used by Tkinter is the Tk library. The Tk library is also used in perl and Tcl scripting languages. Python Mega Widgets (PMW) is an extension toolkit that was used to create the combobox for the multivariate regression options. Most of the scripts created were used in the PyRamid application in some way. A list of reusable widgets and scripts that illustrate GUI options and that may eventually be utilized in the PyRamid Program are provided in Table A.2 of the Appendix.

3.2. PyRamid Demo

The PyRamid application was created as a potential statistics program for the Decision Support System for Agrotechnology Transfer (Hoogenboom et al., 2004). PyRamid was created in Python and exploits the RPy interface to R to execute R functions and access R's graphing capabilities through an easy to use graphical interface. PyRamid allows users to perform linear, multivariate linear, and quantile regressions. The results can be viewed in the form of diagnostic and data plots, or basic statistics tables. There are also options for dataset selection and modification.

3.2.1 Main Window

When the PyRamid Application is run, a small window with various buttons will appear. To start, select the "Open Dataset Selector" button at the top of this window. A "PyRamid Dataset Selector" popup window will appear.

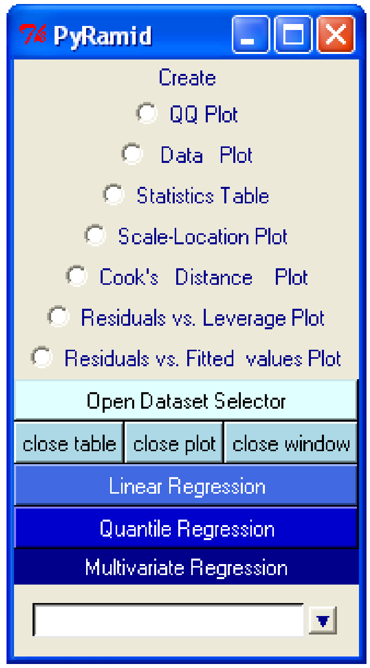

Figure 6: The PyRamid Application

3.2.2 Dataset Selector

Under the file menu, there are currently four options (Fig. 7a). The first is a dotted line that allows the user to tear the menu away from the window and create a popup window. The same selections can be made whether the file menu is attached or unattached.By clicking on “Choose dataset” under the file menu, PyRamid will automatically open a dialog window. From here the user can select a dataset. The file must be column formatted with spaces separating values. Variables may be numeric or character strings, but strings should have no spaces. Once the text file is opened, the data appears in the entry boxes of the Dataset Selector window. This indicates that the data have been successfully loaded and saved in a "temp.txt" file. The data are always read from this file. If a dataset is not selected, the last dataset to be used remains in the file and is used in the regression.

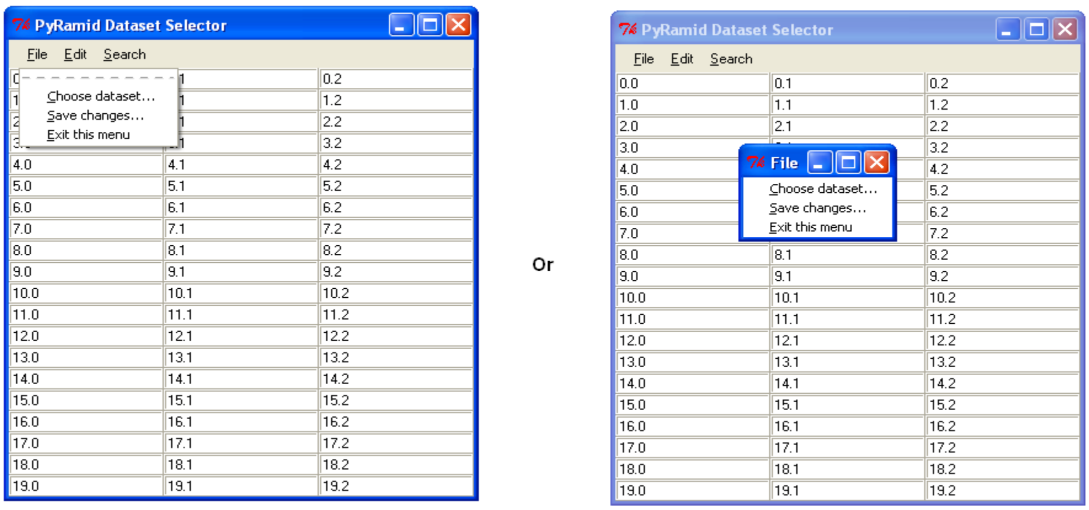

Figure 7a

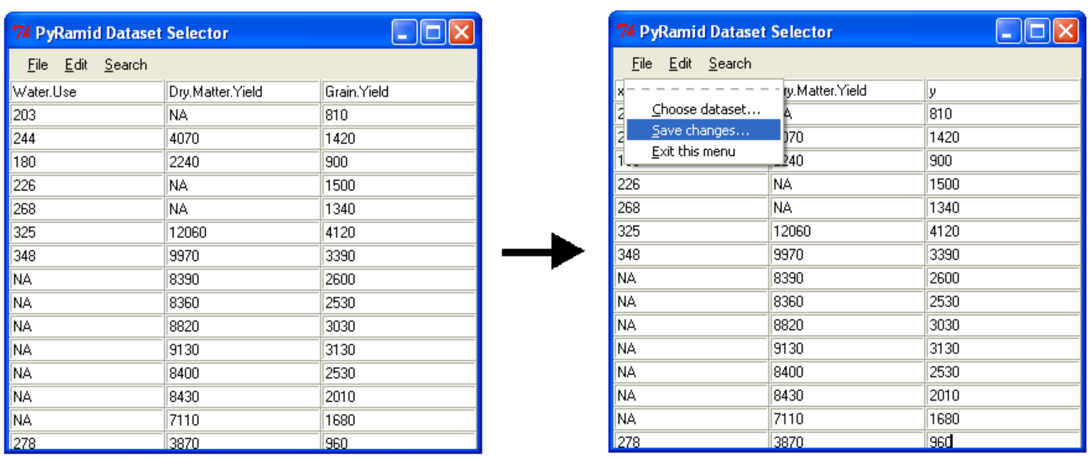

Figure 7b

Figure 7a: The PyRamid Dataset Selector. Figure 7b: Choosing a dataset and choosing the variables is simple. Label the variables and select the "Save changes" option under the file menu.

Currently, the variables must be named with consideration as to which regression method will be applied (Fig. 7b). For this reason, water use will be labeled “x” and grain yield will be labeled “y”. The user can close the “PyRamid Dataset Selector” window by choosing the “Exit this menu” option within the file menu, or by clicking the “close window” button or the “close table” button on the main window of the PyRamid application.

3.2.3 Data Plots

Once the dataset has been selected and the variables have been labeled, the user can choose to create a plot or a statistics table and then choose which regression method to use (linear, quantile, or multivariate). Selecting the “Data Plot” radiobutton and the “Linear Regression” button will cause a plot of data points with the linear regression line to appear in an R graphics window.

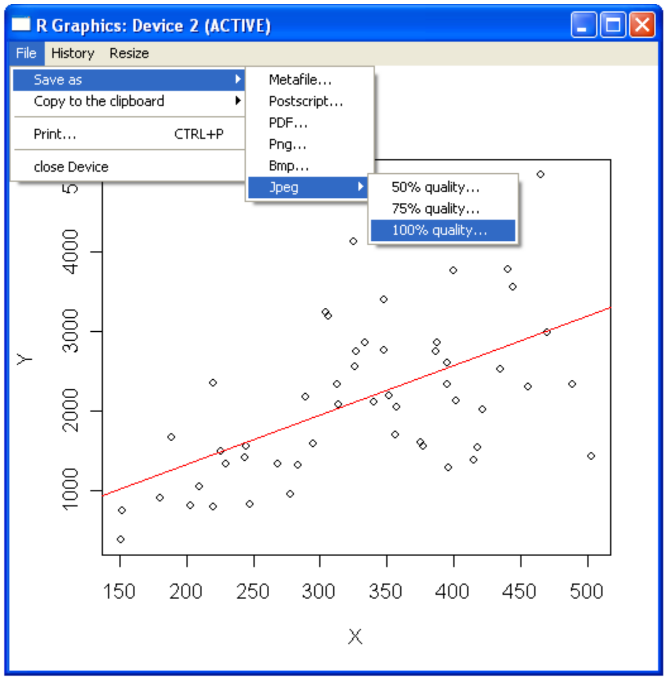

Figure 8: A data plot of grain yield vs. water use. The least squares estimate appears in red. PyRamid plots appear in an R graphics window, which provides the user with the option of saving the plot in a variety formats (Metafile, Postscript, PDF, PNG, BMP, or JPEG).

The user has the option of saving the plot in a variety formats including metafile, postscript, PDF, PMG, BMP, and JPG (Fig. 8). The windows graphic metafile (EMF) format may be imported into many graphics editors for further modification. A "label plot axes" button for the PyRamid main window is under development, which will allow the user to enter a title and x and y axes labels.

By choosing the "Quantile Regression" button on PyRamid's main window, a quantile data plot will appear (Fig. 9). To make the plot disappear, the user can choose the "close device" option in the R graphics device file menu, or simply click on the "close plot" button on the PyRamid main window.

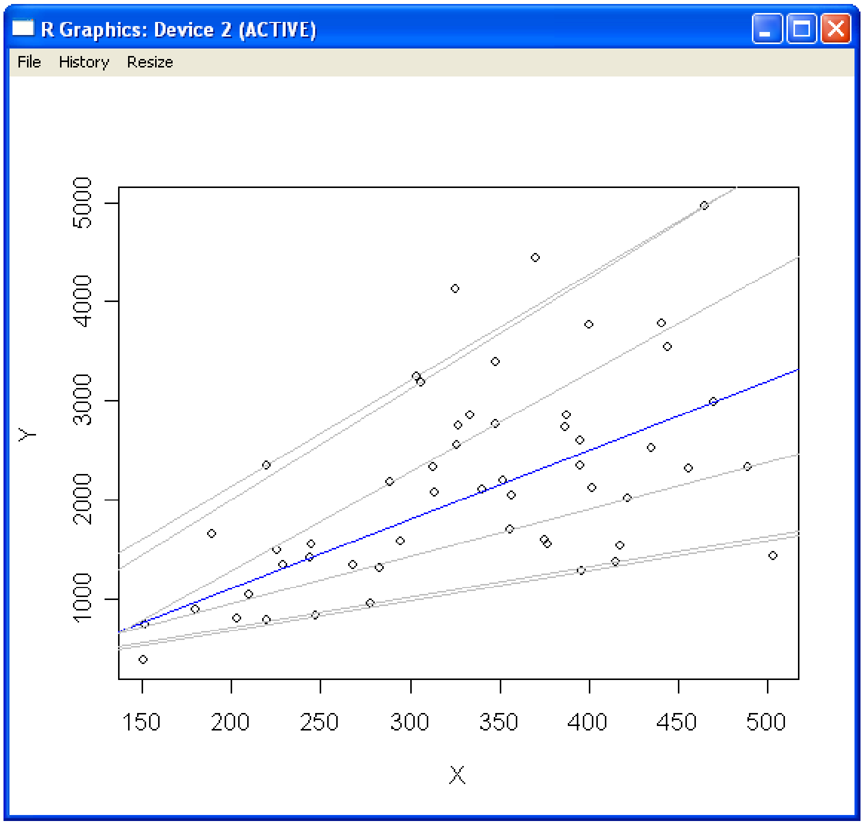

Figure 9: A quantile regression plot of grain yield vs. water use created using PyRamid.

Unfortunately, not all options of the graphics window work as expected because the actual window is produced from R. For example, by choosing the “get from variable” option under the “History” menu, typing in a variable name, and pressing ok, Python will crash. This could be a problem with the RPy interface, where a simple line of code is all that needed to correct this problem. To avoid such problems, PyRamid may eventually have to utilize a Python canvas window, rather than the R graphics window, for plots to be displayed and saved.

3.2.4 Diagnostic Plots

On the PyRamid main window, there are five diagnostic plots that can be generated at the press of a radiobutton. These plots are standardized residuals vs. theoretical quantiles (QQ), standardized residuals vs. fitted values (Scale- Location), Cook's distance vs. observation number, standardized residuals vs. leverage, and the residuals vs. fitted values. An example of the diagnostic plots and statistics table produced using PyRamid can be seen in the appendix (Fig. A.1 and Fig. A.2). Diagnostic plots can be used to determine the existence of outliers. A fitted line may be pulled disproportionately toward the outlier, and therefore it may be beneficial not to include the outlying value. The outlier may be the result of a mistake, but there is also the possibility that the outlier can convey important information. A Cook's distance plot can determine the level of influence that the outlier has on a fitted regression line, but unless the outlying observation can be traced back to a miscalculation or an error in recording, it is best not to discard the outlier.

Currently, the quantile diagnostic plots come up as "not yet implemented". To add quantile diagnostic plots, an option for the user to specify the quantile of interest will have to be created first. Since quantile diagnostic plots appear to have limited use at this time, they have not been included in the current prototype.

3.2.5 Statistics Tables

By pressing the “Statistics Table” radio button, a table is created which contains an ANOVA table as well as the slope, the intercept, and the number of observations. In the case where a factor (a list of discrete values as strings) is used as a predictor variable, such as for geographic locations, the program will automatically match up the strings and enumerate them. The slope and intercept of regressions involving factors were not included in the statistics table to save time. These values can be added in later but have limited use in statistical validation. The statistics are printed to the console and to a tabletemp.txt file where they are placed into a statistics table window. By Selecting the “Save as” option under the file menu, the results can be saved. This window is closed by choosing “Exit this Menu” in the file menu or by selecting the “close table” button on the main PyRamid window. The current statistics table for quantile regressions will display the number of observations, slope, and intercept values for the corresponding quantile estimates, but not the ANOVA tables. This can be easily added after the program has been modified to allow the user to select the quantile regression estimate of interest.

3.2.6 Demonstration

Multivariate regression can be performed on a dataset containing y, x1, and x2 values (x3 is optional). The CMS-CROPGRO-Soybean model dataset was opened and simulated anthesis was labeled as "x1", anthesis observed was labeled "y", environment was labeled "x2", and cultivar was changed to "x3". The data was saved, and from the multivariate regression dropdown list, the fourth multivariate regression equation was chosen.

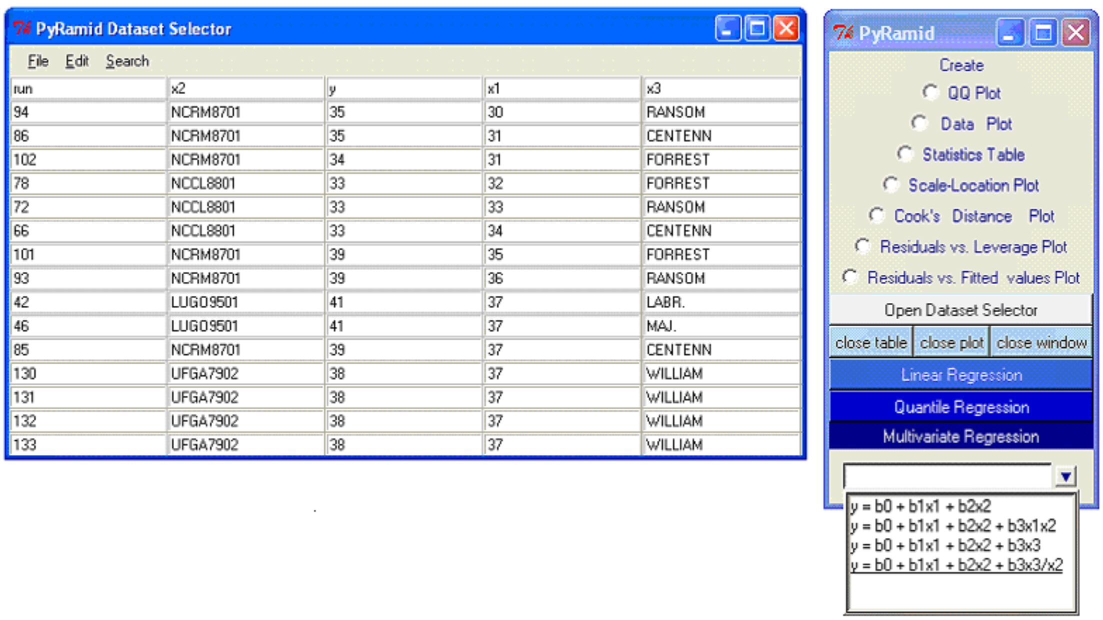

Figure 10: How to perform multivariate regression using PyRamid. The Y = b0+b1x1+b2X2 + b3X3/X2 option was chosen, where x3/x2 sybolizes x3 is nested in x2.

After selecting the fourth equation (which includes the term x3/x2 indicating in this case that cultivar is nested within the environments), the ANOVA table appears. The application has checked for strings and automatically enumerated them. The coefficients are not given in the table as a result. Data plots can also be displayed (one at a time, click the first plot to display the next plot). The results of the statistics table (shown in Fig. A.3 of the appendix) are the same as those given by SAS and R directly.

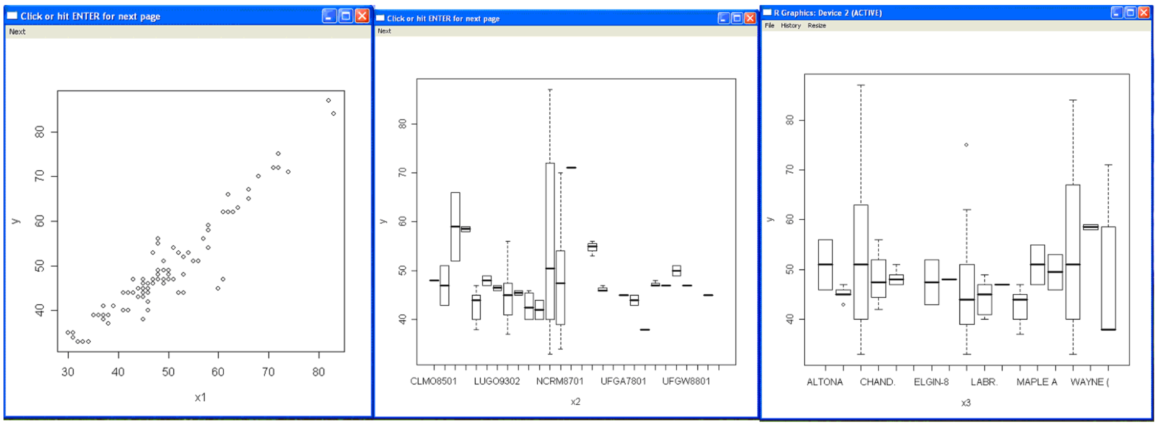

Figure 11: Data plots for observed days to anthesis days to anthesis vs. simulated days to anthesis, environment, and cultivar, respectively.

In summary, the PyRamid application allows the user to: (1) choose and modify a dataset, (2) display diagnostic and data plots, (3) create and save basic statistics tables and ANOVA tables, and (4) perform linear, multivariate linear, and quantile regressions. There are five diagnostic plot options to choose from. The basic statistics table includes the slope, intercept, and number of observations. ANOVA tables are available for all regressions except quantile regressions.

4.0 CONCLUSIONS

After reviewing the literature, the most appropriate statistical techniques to use for validating simulation models of the continuous, dynamic, and deterministic type were specified. These methods were categorized into three groups: 1) visual techniques, 2) regression techniques, and 3) deviance measures. The most appropriate visual technique was recommended to be the plotting of the observed vs. simulated values, which allows a model user to determine which regression techniques to apply next, and may point out insufficient data sampling. Linear regressions, quantile regressions, and F-tests were the suggested regression techniques. No other studies, to the best of my knowledge, have proposed the use of quantile regression during validation. These visual and regression techniques were implemented in a tool to assist validation. This tool utilizes the RPy interface to R, creating a GUI interface in Python that allows model users to easily apply these statistical techniques. Certain deviance measures were suggested to be included as future options in the PyRamid program. These measures are modeling error, root mean squared error, and mean absolute value.

4.1 FUTURE WORK

Although deviance measures can be easily derived in R, statistics are saved as R objects that cannot be retrieved with Python using RPy. Thus, deviance measures were not included in the current demo. Future work for

PyRamid will be to include deviance measures in the statistics table by explicitly coding the equations in Python. Table A.1 was created in R and illustrates a possible format for future statistics tables. In addition, more options that will aid in visual analysis will be included in the program. The ability to superimpose the least squares and 95% confidence interval estimates and to select different colors and characters are practical options.

The current PyRamid program only includes a limited number of multivariate equations to choose from. An equation builder option, which would allow users to enter their own multivariate regression equations, is currently under development. Additional options for quantile regression will also be made available, which allow the user to choose a quantile and view the corresponding diagnostic plots and ANOVA tables.

The prototype software created was intended to be used as an application within the Decision Support System for Agrotechnology Transfer (DSSAT). The next step in PyRamid development is to make the program executable for use within the DSSAT program. For example, the "freeze" tool (available with Python) will create a C-language file that contains all of the Python modules in an application. These files are compiled to a DLL that can be shipped with the application. This is more convenient for the user because they will not have to download Python separately to use the software. It is unclear at this point whether the other

Python library options and extensions (RPy, Pywin, NumPy, and PMW) can be included in the executable version.

Although many types of simulation models exist, DSSAT creates simulations which are primarily dynamic, continuous and deterministic. As one would expect, validation methods for this type of simulation became a focus of the literature review early on. Comparing the different statistical methods and deciding which ones were important in the assessment of DSSAT models was a key factor when selecting methods to include in the current prototype. Another factor was the level of difficulty in adding the technique as an option into PyRamid, however, the addition of such options will be considered in later development.

REFERENCES

1. Balci, O. and W. Ormsby. 2000. Planning for Verification, Validation, and Certification of Modeling and Simulation Applications. *Proceedings of the 2000 Winter Simulation Conference*. Eds. J. A. Joines, R. R. Barton, K. Kang, and P. A Fishwick, pp. 829-839.
2. Chung, C. A. *Simulation modeling handbook: a practical approach*. CRC Press, New York. (2004)
3. French, R. J. and J. E. Schultz. (1984) Water use efficiency of Wheat in a Mediterranean-type Environment: The Relation between Yield, Water Use, and Climate. *Aust. J. Agric. Res.* 35:743-764.
4. Harrison, S. R., 1990. Regression of a model on real-system output: an invalid test of model validity. *Agricultural Systems* 34:183-190.
5. Hoogenboom, G., J. W. Jones, P. W. Wilkens, C. H. Porter, W. D. Batchelor, L. A. Hunt, K. J. Boot, U. Singh, O. Uryasev, W. T. Bowen, A. J. Gijsman, A. du Toit, J. W. White, and G. Y. Tsuji. 2004. *Decision Support System for Agrotechnology Transfer Version 4.0*. CD-ROM Publication. University of Hawaii, Honolulu, HI.
6. Jones, J. W., G. Hoogenboom, C. H. Porter, K. J. Boot, W. D. Batchelor, L. A. Hunt, P. W. Wilkens, U. Singh, A. J. Gijsman, and J. T. Ritchie. 2003. The DSSAT Cropping System Model. *Eur. J. Agron.* 18:235-265.
7. Kleijnen, J. P. C., 1999. Validation of Models: Statistical techniques and Data Availability. *Proceedings of the 1999 Winter Simulation Conference*. Eds. P.A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, pp. 647-654
8. Kobayashi, K., and M. U. Salam. 2000. Comparing Simulated and Measured Values Using Mean Squared Deviation and its Components. *Agronomy Journal* 92:345-352.
9. Koenker, R., and G. Bassett, 1978, "Regression Quantiles", *Econometrica*, 46 (1): 1-26.
10. Koenker, R., and K. F. Hallock. 2001. Quantile regression. *Journal of Economic Perspectives* 15(4): 143-156.
11. Law, A. and D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill. Third edition. (2000) pp. 1-9
12. Lutz, M. *Programming Python*. O'reilly, Second edition. Cambridge. 2001. pp. 249-425.
13. Mayer, D. G. and D. G. Butler, 1993. Statistical Validation. *Ecological Modeling* 68:21-32.
14. Mayer, D. G., M. A. Stuart, and A. J. Swain. 1994. Regression of a model on real-system output: an appropriate overall test of model validity. *Agricultural Systems* 45:93-104.
15. Maindonald, J. and J. Braun. 2003. *Data analysis and graphics using R*. Cambridge University Press, New York.
16. Neter, J., M. H. Kutner, C. J. Nachtsheim, and W. Wasserman. *Applied Linear Statistical Models*. WCN/McGraw-Hill, Fourth edition. (1996) pp. 95-143
17. Rose, K. A., and E. P. Smith. 1995. Model goodness of fit analysis using regression and related techniques. *Ecological Modeling* 77:49-64.
18. Rykiel, Jr., E. J., 1996. Testing Ecological Models: The Meaning of Validation. *Ecological Modeling*.90:229-224.
19. Sargent, R. G., 2000. Verification, Validation, and Accreditation of Simulation Models. *Proceedings of the 2000 Winter Simulation Conference*. Eds. J. A. Joines, R. R. Barton, K. Kang, and P. A Fishwick, pp. 50-59.
20. Scharf, F. S., F. Juanes, and M. Sutherland. 1998. Inferring Ecological Relationships from the Edges of Scatter Diagrams: Comparison of Statistical techniques. *Ecology* 79(2):448-460.
21. Thornton, P. K., and J. W. Hansen. 1996. A Note on Regressing Real World Data on Model Output. *Agricultural Systems* 50: 411-414.

22. Vanclay, J. K., and J. P. Skovsgaard. 1997. Evaluation of Forest Growth Models. *Ecological Modeling*. 98:1-12

23. Willmott, C. J., 1982. Some comments on the evaluation of model performance. *Bulletin American Meteorological Society*, pp. 1309-1313

APPENDIX

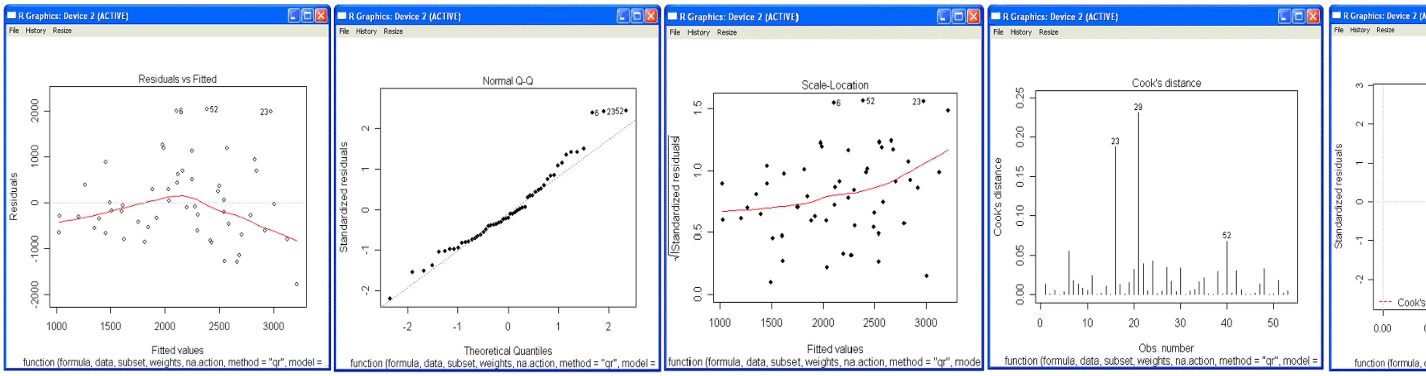

Figure A.1: Diagnostic plots produced using PyRamid.

PyRamid Statistics Table

File Edit

Formula.Table	-	-	-	-	-
Slope	Intercept	Number.of.observations	-	-	-
6.212151	88.291960	53	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
Anova.Table	-	-	-	-	-
-	Df	Sum.Sq	Mean.Sq	F.Value	Pr(>F)
X	1	16332378.095219	16332378.095219	22.562158	0.000017
Residuals	51	36918067.187800	723883.670349	-	-

Figure A.2: Statistics table produced using PyRamid's linear regression option

PyRamid Statistics Table

File Edit

Anova.Table	-	-	-	-	-
-	Df	Sum.Sq	Mean.Sq	F.Value	Pr(>F)
X1	1	10420	10420.433890	1905.493618	0.000000
X2	23	785.663002	34.159261	6.246405	0.000000
X3\X2	18	77.405314	4.300295	0.786357	0.708004
Residuals	65	355.460756	5.468627	-	-

Figure A.34: ANOVA Table for y=bo+b1x1 + b2x2 + b3x3/x2 where y is the observed days to anthesis, x1 is the simulated days to anthesis, x2 is the environment, x3 is the cultivar, and x3/x2 indicates nesting.

Variable. Name	Mean. Observed	Mean. Simulated	Mean. Absolute. Difference	Root. Mean. Squared. Error	Intercept	Slope	R. Squ
Days.to. Anthesis	48.51852	49.22124	2.333333	3.412127	2.76174	0.952239	0.89

Table A.1: Basic Statistics for comparisons of observed and simulated data for days to anthesis. The data was taken from the CMS-CROPGRO-Soybean model version 4.0.2.0 (Jones et al. 2003, Hoogenboom et al. 2004). The table was created using the R language and environment for statistical computing (R version 2.3.0 for windows). Equivalent values are given in SAS.

Name	**Library**[1]	**Other**[2]	**Purpose**
Bind.py		Δ	Callback handlers are triggered when bound events occu receives an event object argument that gives details abo
Button.py	Δ		make an increment button
canvasDraw.py	Δ		draw on canvas: delete with double left click, drag to dra drawn object to new spot with a click
Counter.py	Δ		Creates a counter widget
CustomDLG.py	Δ		Create a pop-up window in either modal or non-model on the makemodal global value
demoScale.py		Δ	Links one variable with two scales
dlg1.py		Δ	Buttons to trigger pop-ups
ECQDF.py		Δ	Estimates conditional quantile and density functions usi
Entry.py	Δ		The Entry widget is a single line input field that support bindings for editing, and text selections
Functions.py	Δ		Contains linear, quadratic, cubic, sine, exponential, expo and hyperbolic math functions
imgButton.py	Δ		Puts an image on a button
Loops.py		Δ	Loops for finding a specific integer in a series (i.e., large by seven,)

Table A. 2: List of scripts and their uses

[1] Part of Library of reusable classes and widgets (scripts used in PyRamid are excluded from the list)

[2] Scripts for learning Tkinter gui concepts