

**Background and Implementation of Proposed Methodology for
Projecting Domestic Percent Crop Treated**

Economic Analysis Branch
Biological and Economic Analysis Division
Office of Pesticide Programs
U.S. Environmental Protection Agency
August 2002

Table of Contents

<u>1</u>	<u>Introduction</u>	1
<u>2</u>	<u>Exponential Smoothing Models</u>	2
<u>2.1</u>	<u>Fitted Values and Point Forecasts</u>	3
<u>2.1.1</u>	<u>Simple Exponential Smoothing</u>	3
<u>2.1.2</u>	<u>Origin of the Term Exponential Smoothing</u>	7
<u>2.1.3</u>	<u>Linear Exponential Smoothing</u>	8
<u>2.1.4</u>	<u>Damped-Trend Exponential Smoothing</u>	10
<u>2.2</u>	<u>Interval Forecasts</u>	14
<u>2.2.1</u>	<u>Interval Forecasts for Single Point Forecasts</u>	14
<u>2.2.2</u>	<u>Interval Forecasts for Multiple Point Forecasts</u>	16
<u>3</u>	<u>Forecasting Methodology</u>	19
<u>3.1</u>	<u>Initial Values, Parameter Estimation, and Model Selection</u>	19
<u>3.1.1</u>	<u>Initial Values</u>	19
<u>3.1.2</u>	<u>Estimating Parameters and Model Selection</u>	20
<u>3.2</u>	<u>Outline of Forecasting Methodology</u>	22
<u>3.3</u>	<u>Coded Version of the Forecasting Methodology</u>	22
<u>3.3.1</u>	<u>Preliminary Data Manipulation</u>	22
<u>3.3.2</u>	<u>Linear Regression Parameter Estimation</u>	23
<u>3.3.3</u>	<u>Model Initialization and Parameter Estimation</u>	23
<u>3.3.4</u>	<u>Forecast Variance and Point Forecast Calculations</u>	25
<u>3.3.5</u>	<u>Interval Forecast Calculations</u>	26
<u>3.3.6</u>	<u>Model Selection</u>	27
<u>3.3.7</u>	<u>Data Set Construction</u>	28
<u>3.3.8</u>	<u>Graphical Display</u>	28
	<u>References</u>	30
	<u>Appendix</u>	31

1 2 Introduction

Pesticides are regulated in the U.S. under the Federal Insecticide, Fungicide, and Rodenticide Act (FIFRA) and the Federal Food, Drug and Cosmetics Act (FFDCA). In 1996, Congress passed the Food Quality Protection Act (FQPA), which amended both FIFRA and FFDCA by requiring that aggregate and cumulative risks be considered by the Environmental Protection Agency (EPA) in granting pesticide tolerance petitions and in assessing whether pesticides can be reregistered for use. Through these statutes, EPA evaluates risks posed by the use of each pesticide to make a determination of safety. Only if the Agency determines that such residues would be “safe”, may it authorize a tolerance to allow a pesticide residue in food.

One of the responsibilities of EPA’s Office of Pesticide Programs (OPP) is to assess the potential risks from pesticide residues for food consumption. The size of the potential risks depends on a variety of factors including the toxicity of the pesticide (how much harm, if any, is caused by specific amounts of the pesticide) and the magnitude of the exposure to the pesticide. In turn, exposure to a pesticide in the food supply depends on two factors: the amount of the pesticide present in food and how much food a person eats.

To develop estimates of such exposure, the Agency must use available and reliable, representative data for such risk assessments. These data include pesticide use statistics such as the percent of a crop treated (PCT) with a particular pesticide.

The FQPA-amended FIFRA also requires that OPP re-evaluate risks on a continuing basis. Specifically, the act permits the Agency to consider the percent of a crop that is treated with a pesticide (PCT), but requires that this information be re-evaluated (and, if necessary, the risk assessment be adjusted) after five years. Thus, estimates of PCT may need to reflect *future* pesticide use based on information from OPP data sources. As such, OPP may *forecast* PCT so as to ensure that pesticide exposure estimates are not being underestimated over a given period.

The Office of Pesticide Programs (OPP) has developed a methodology for forecasting the percent of a crop that is treated with a particular pesticide (PCT). PCT information for several pesticide crop combinations is made available to OPP in the form of reports and/or databases on an annual basis. The PCT values for a particular pesticide crop combination can be arranged as a time series of annual observations, and forecasting PCT has focused on extrapolation techniques. Specifically, exponential smoothing (ES) has been identified as the forecasting method of choice (please see the companion paper to this for more details). The term “forecasting method” refers to the technique employed to forecast PCT values. The first section of this document will provide formulae and examples for calculating fitted and forecast values, and provide descriptions of the model parameter(s) for each of the models. The second section will provide an overview of the forecasting methodology. The term “forecasting methodology” refers to the systematic steps and decision rules applied to automate the forecasting procedure. The steps and rules associated with the methodology include estimating the model parameter(s), choosing the “best” model for the data, and calculating the forecasts. The primary focus of the section will be the encoding of this methodology using the SAS IML[®] software.

3 Exponential Smoothing Models

The proposed forecasting methodology will employ three of the fifteen exponential smoothing (ES) models presented in Hyndman et al. (forthcoming). Nine of the ES models contain a seasonality component to account for within-year patterns. These models were immediately excluded because PCT information is reported on an annual basis. The six remaining models can be categorized by their trend component (none, additive, and damped), with two models in each category. The models in each category differ only in how the error terms are specified. Three of the models are specified as having additive error terms, while the other three are specified as having multiplicative error terms. The error terms of a model represent the random fluctuations one expects to find in the time series of interest. Additive error terms imply that the variance of the error terms is *constant* throughout the time series (i.e. independent of the magnitude of the observations). This characteristic is referred to as homoscedasticity. Multiplicative error terms are specified to *vary* over time since they are proportional to the magnitude of the observations. This is an example of heteroscedasticity. The three ES models that specify additive error terms were selected because fluctuations in PCT time series are expected to be independent of the magnitude of the observations (i.e. large PCT values are not expected to fluctuate more so than small PCT values). Based on classification of their trend component, these three models are referred to as simple exponential smoothing, linear exponential smoothing, and damped-trend exponential smoothing throughout this paper.¹

The three ES models being considered can provide both *point* and *interval* forecasts. A point forecast is the *expected value* or mean of the time series for a specific time period. An interval forecast on the other hand provides a range of values within which one can be assured, with a certain degree of confidence, future observations will lie. Interval forecasts account for uncertainty or variability associated with the point forecasts: greater uncertainty results in a larger interval. Generally, OPP is interested in obtaining two different estimates of PCT. These two pesticide usage statistics are referred to as “likely average PCT” and “likely maximum PCT.” The “likely average PCT” is an estimate of the percent of a crop that is treated in a typical year. The “likely maximum PCT” is an upper bound for the “likely average PCT” that is not expected to be exceeded. This upper bound is intended to reflect the maximum percent of the crop expected to be treated based on the historic observations. This value can be seen as “protective” and account for any trend or variability in the data. Both the “likely average PCT” and the “likely maximum PCT” refer to pesticide use at the national level and are estimated using *point* and *interval* forecasts respectively. More specifically, the upper bound of the interval forecast would serve as an estimate for the “likely maximum PCT.”

The remainder of this section will provide descriptions, equations, and examples for the three ES models. The first part will discuss fitting the models to the data and calculating the point forecasts. The second part will focus on calculating the interval forecasts for each of the models.

3.1 Fitted Values and Point Forecasts

3.1.1 Simple Exponential Smoothing

The simple exponential smoothing model (SES) is the least complex ES model. SES models the time series as having a “level.” The level of the time series may shift from year to year, but

these changes are not expected to follow a consistent pattern. If the variations in the level *do* consistently increase or decrease, then the time series should be modeled with a trend component in addition to the level component. The equation for the model is

$$Y_t = l_{t-1} + \hat{\epsilon}_t \quad (1)$$

Y_t represents the observation of the time series Y_1, Y_2, \dots, Y_n at time t , l_{t-1} is the level at time $t-1$, and ϵ_t is the model error (i.e. random fluctuations associated with the underlying data) at time t . These error terms are assumed to be independent and from the *same* normal distribution with a mean of zero and a variance denoted as σ^2 [i.e. $\{\epsilon_t\}$ are i.i.d. $N(0, \sigma^2)$]. The estimate of the variance term, σ^2 is dependent upon whether the error terms are assumed to be heteroscedastic or homoscedastic. The assumption of normality of the error terms has no effect on the *point* forecasts of the ES models, since a violation of this assumption would not affect the estimation of the model parameters as long as the mean of the error terms is zero. However, mischaracterizing the distribution of the error terms may affect the *interval* forecasts whose calculation relies upon critical values of the normal distribution. Since the mean of the error terms is assumed to be zero, the *estimated* or *fitted* value for the observation at time t is

$$Y_t = l_{t-1} \quad (2)$$

To compute the level, the following recursive equation is used:

$$l_t = l_{t-1} + \alpha \epsilon_t \quad (3)$$

where the level smoothing parameter, α is bounded between zero and one. Since the value of the error terms is unknown, estimates are used. The estimate for the model error is the forecast error denoted as

$$\epsilon_t = Y_t - \hat{Y}_t \quad (4)$$

Using (2) and (4), (3) can be written in its “error-correction” form,

$$Y_{t+1} = Y_t + \alpha(Y_t - \hat{Y}_t) \quad (5)$$

In this form the role of the smoothing parameter α can be better understood. From the above equation, the fitted value at time $t+1$ is the previous fitted value plus some percentage of the previous forecast error. If α is large (i.e. close to one), then the forecast error greatly affects the next fitted value. If α is small (i.e. close to zero), then the forecast error does not greatly affect the next estimate. To help illustrate this relationship, (5) can be examined with respect to the

extreme values of α . When $\alpha = 1$, then (5) becomes $Y_{t+h} = Y_t$. Thus the fitted value for time $t+1$ is the value of the observation from the previous year. This method of forecasting is equivalent to the “random walk model” or “naïve model.” When $\alpha = 0$, then (5) becomes $Y_{t+h} = \bar{Y}$. In other words, the fitted values do not change from year to year. In the case of $\alpha = 0$, the common fitted value that would minimize the forecast errors would be the algebraic mean (i.e. average) of the time series; this is sometimes referred to as the “constant model” or “mean model.”

The extremes of the α parameter for the simple model reflect the extreme characteristics one might endeavor to model: a time series so volatile that the best estimate of an observation from the time series is the value of the previous observation (i.e. no relationship between subsequent observations); and a time series so tame and invariable that the best estimate of an observation is average of the previous observations. Thus SES would serve as an adequate model for time series with a level that may change irregularly over time or whose observations are fairly constant with only random fluctuations about the mean. The introduction (or eradication) of a pest in a region would be example where the “level” of the percent crop treated values would shift. An example where the percent crop treated values might fluctuate about a mean would be a pest that consistently presents a problem to growers; the percent crop treated values would be reflective of the consistent pest pressure and would tend to be the same from year to year. For time series with levels that increase or decrease in a consistent manner over time (i.e. have an upward or downward trend), linear exponential smoothing or damped-trend exponential smoothing may be a more appropriate model. To illustrate the computation of the fitted values for the simple model, some examples are shown below in spreadsheet format. The procedure for choosing the appropriate value of the smoothing parameter will be discussed in next section.

Table 1. Examples of Simple Exponential Smoothing (SES)

(a)

(b)

So far the discussion has pertained to calculating the *fitted* values for the SES model. Now the focus will be calculating the *forecast* values for the SES model. The notation used for the

forecast values is similar to that used for the fitted values. For a time series Y_1, Y_2, \dots, Y_n , the h -period-ahead SES forecast is denoted as

$$Y_n(h) = l_n \quad (6)$$

where l_n is the last estimated value of the level for the time series. The similarity between (2) and (6) can be understood if $Y_n(l)$ is viewed as the fitted value at time $n+1$ (i.e. one time period beyond the last observation). From this perspective, (2) would become

$$Y_{n+1} = l_n \quad (2a)$$

The same equations used for fitting the model to the time series are used to calculate the point forecasts. To do so, first use (2) to calculate Y_n in the time period $(n+2)$:

$$Y_{n+2} = l_{n+1} \quad (2b)$$

In turn, to calculate the value of l_{n+1} , (3) would be used to obtain

$$l_{n+1} = l_n + \hat{a}_{n+1} \quad (3a)$$

Again the value of the error term is unknown so the estimate must be calculated using (4):

$$\hat{a}_{n+1} = Y_{n+1} - Y_{n+1} \quad (4a)$$

Herein lies the problem: the observed value at time $n+1$, Y_{n+1} , is unknown since it has yet to be observed. What is known is that the mean of the error term is equal to zero. [Recall that the underlying assumption of the model is that $\{\varepsilon_t\} \sim N(0, \sigma^2)$]. Therefore, if the mean of the error term is substituted into (3a), the equation becomes $l_{n+1} = l_n$, which is exactly the case with (6) (i.e. the level is the same for all years being forecasted). Thus, regardless of the number of years

being forecasted, the *point* forecast does not change [i.e. $Y_n(1) = Y_n(2) = \dots = Y_n(h) = l_n$]. This is not the case, however, with the *interval* forecasts, which will be discussed in the second part of this section. Forecasts for the previous examples are given below. The highlighted values represent the point forecasts.

Table 2. Examples of forecast values for the simple (SES) model

(a)

(b)

3.1.2 Origin of the Term Exponential Smoothing

Before discussing the linear exponential smoothing (LES) model, the origin of the term “exponential smoothing”

will be explained. Now that the underlying equations of the simple form of ES have been examined, consider another perspective of the method. To facilitate the process, (5) is rewritten as

$$Y_{t+h} = \alpha Y_t + (1-\alpha)Y_t \quad (5a)$$

As with the earlier example, (5a) can be defined in a recursive manner (i.e., the equation is defined in terms of itself). Therefore if (5) is used again to define the Y_t term, then (5a) becomes

$$Y_{t+h} = \alpha Y_t + (1-\alpha)[\alpha Y_{t-1} + (1-\alpha)Y_{t-1}] \quad (5b)$$

which can be rearranged as

$$Y_{t+h} = \alpha Y_t + \alpha(1-\alpha)Y_{t-1} + (1-\alpha)^2 Y_{t-1} \quad (5c)$$

If this process is repeated, (5) can eventually be rewritten as

$$Y_{t+h} = \alpha Y_t + \alpha(1-\alpha)Y_{t-1} + \alpha(1-\alpha)^2 Y_{t-2} + \dots + \alpha(1-\alpha)^{t-1} Y_1 \quad (5d)$$

As the above equation implies, the fitted value for a given time period is a *weighted* average of the previous observations. The following graphs indicate the value of the weights for specific values of α .

Figure 1. Magnitudes of weights for various values of values of alpha

Notice that the weights for the more recent observations are larger than those of the distant observations and that the closer α is to the value of one, the more rapidly the magnitude of the weights decline. Actually, the values of the weights decay *exponentially*. Using these weights one calculates a forecast that is a *smoothed* estimate of the historic observations. Together these terms provide the origin of the name *exponential smoothing*.

3.1.3 Linear Exponential Smoothing

In addition to an estimate for level, the linear exponential smoothing (LES) model includes a trend for the time series. For LES, the model equation is

$$Y_t = l_{t-1} + b_{t-1} + \hat{\varepsilon}_t \quad (7)$$

As before, Y_t denotes the observation at time t , l_{t-1} denotes the level, and ε_t denotes the error term. Additionally, b_{t-1} represents the slope or trend associated with the time series. Again the underlying assumption is that $\{\varepsilon_t\}$ are i.i.d. $N(0, \sigma^2)$. The fitted value of the observation at time t is

$$Y_t = l_{t-1} + b_{t-1} \quad (8)$$

The level and slope of the fitted values are calculated using the following equations respectively

$$l_t = l_{t-1} + b_{t-1} + \alpha \hat{\varepsilon}_t \quad (9)$$

$$b_t = b_{t-1} + \beta \hat{\varepsilon}_t \quad (10)$$

where α and β represent the level smoothing parameter and trend smoothing parameter respectively. Both α and β are bounded between the values zero and one. As with the SES model, the forecast errors serve as estimates of the model errors. Again, to better understand the roles of the smoothing parameters, the above equations will be written in their “error-correction” form. By substituting (4) and (8) into (9), one obtains the error-correction form of (9)

$$l_t = Y_t + \alpha(Y_t - Y_t) \quad (11)$$

Keeping in mind that the level from time t will be used to calculate the fitted value at time $t+1$, equation (11) demonstrates that the level for a fitted value is equal to the previous fitted value plus some percentage of the forecast error. To write (10) in its error-correction form is not as straightforward. Without changing the balance of the equation, (10) can be rewritten as

$$b_t = b_{t-1} + \hat{\alpha}[l_{t-1} + b_{t-1} + \hat{\alpha}\hat{\varepsilon}_t - l_{t-1} - b_{t-1}] \quad (10a)$$

by adding and subtracting the two terms l_{t-1} and b_{t-1} . Now (10a) can be written in its error-correction form by substituting (9) into (10a).

$$b_t = b_{t-1} + \beta[l_t - l_{t-1} - b_{t-1}] \quad (12)$$

The parenthesis around the terms l_t and l_{t-1} are used to emphasize the concept that the difference between levels of subsequent fitted values (i.e., the levels from time t and $t-1$) should be approximately equal to the slope estimate at time $t-1$. The terms within the brackets can be

thought of as representing the change in the slope (i.e. what the slope was estimated to be at time $t-1$, b_{t-1} and the change in the level from time $t-1$ to t). Therefore (12) can be interpreted as follows: the slope for the next fitted value is equal to the slope from the previously fitted value plus some percentage of the change in the slope. The following examples illustrate the results of the fitted values of a linear ES model for two different crop/chemical combinations.

Table 3. Examples of fitted values for the linear model (LES)

(a)

(b)

The linear model provides useful forecasts because it models a time series as having a level and slope that can vary over time. With the addition of a slope component, the forecasts for the linear model are more sophisticated than those of the simple ES model and provide a mechanism for estimating (and forecasting) trends in data. In cases where the percent crop treated is steadily increasing (or decreasing) over time, the linear model can be very useful for estimating future PCT values. For the linear model, the notation for the forecasted values is the same as the simple model. However, the equation for calculating the point forecasts differs. For an h -period-ahead forecast,

$$Y_n(h) = l_n + hb_n \quad (13)$$

The last estimated values for the level and trend of the time series are denoted as l_n and b_n respectively. Similar to (8), the forecasted value for a one-period-ahead forecast is the sum of the level and slope. However for every additional time period, an additional slope factor is added to the forecast. Thus the forecasts for the linear model all lie on the same line with slope b_n . The forecasts for the previous examples are given below. In the first series (Table 4a), the PCT increases up to the forecast horizon (1995), and the forecast reflects this upward trend. In the second time series (Table 4b), there is a slight upward trend up to the forecast horizon and the forecasts reflect this.

Table 4. Examples of forecast values for the linear model (LES)

(a)

(b)

3.1.4 Damped-Trend Exponential Smoothing

In some cases where percent crop treated is increasing, a linear model may not be the best choice. For example, when percent crop treated is increasing, it cannot continue increase indefinitely; ultimately it is bounded by an upper value of 100%. In many cases this upper bound is much lower. For example, perhaps only 70% of the area where a crop is grown may provide a favorable environment for the pest. In this case the percent crop treated would be limited by the 70% value. For cases in which the trend is showing signs of leveling off, the damped-trend model would be very useful. The damped-trend exponential smoothing (DES) model is very similar to the linear model. Both model the time series as having level and trend components. In fact, the equation that specifies the damped-trend model is exactly the same as that of the LES model.

$$Y_t = l_{t-1} + b_{t-1} + \hat{a}_t \quad (7)$$

Even the equations for the fitted values and the level remain the same.

$$Y_t = l_{t-1} + b_{t-1} \quad (8)$$

$$l_t = l_{t-1} + b_{t-1} + \alpha \varepsilon_t \quad (9)$$

The calculation of the slope component is where the two models differ. The equation for the slope of the DES model includes an additional parameter, ϕ , a damping coefficient, which ranges over the values β to one.

$$b_t = \phi b_{t-1} + \alpha \beta \varepsilon_t \quad (14)$$

The error-correction form of the level remains the same and has the same interpretation as the linear model. The error-correction form of the slope differs slightly, but is obtained in a similar manner as its counterpart in the linear model.

$$b_t = \phi b_{t-1} + \beta [(l_t - l_{t-1}) - b_{t-1}] \quad (15)$$

Again the only difference is the damping coefficient, ϕ . This damping coefficient allows the damped-trend model to specify a trend component that decays over time by a constant factor, namely $(1 - \phi)$; thus the name *damped-trend*. The similarity between the two models is not coincidental. The linear model is actually a special case of the damped-trend model, when ϕ equals one. Some examples of the fitted values for the damped-trend model are given below.

Table 5. Examples of fitted values for damped-trend model (DES)

Like the linear model, the point forecasts for the damped-trend model have level and trend components. However ϕ is also included in the calculation. To calculate an h -period-ahead point forecast, the following equation is used:

$$Y_n(h) = l_n + \phi_{h-1} b_n \quad (16)$$

where

$$\phi_{h-1} = \sum_{i=0}^{h-1} \phi^i = 1 + \phi^1 + \phi^2 + \dots + \phi^{h-1} \quad (17)$$

Thus the first forecast is given by the equation

$$Y_n(1) = l_n + \phi_0 b_n \quad (16a)$$

or

$$Y_n(1) = l_n + b_n \quad (16b)$$

The calculations for the 2- and 3-period-ahead forecasts would be

$$Y_n(2) = l_n + (1 + \phi) b_n \quad (16c)$$

and

$$Y_n(3) = l_n + (1 + \phi + \phi^2) b_n \quad (16d)$$

respectively. Thus the additional slope component between subsequent forecasts will become progressively *damped* [i.e. decrease by a factor of $(1 - \phi)$]. Forecasts for the previous examples are given below.

Table 6. Examples of forecast values for damped-trend model (DES)

3.2 Interval Forecasts

3.2.1 Interval Forecasts for Single Point Forecasts

Interval forecasts or prediction intervals are important for establishing the upper and lower bound for a point forecast. However, OPP is interested primarily in the upper bound of a point forecast, because of the need to estimate the “likely maximum PCT.” The “likely maximum PCT” values for a pesticide may be used in dietary exposure assessments (as described in a companion paper). For all ES models, the general form of a prediction interval with a $100\% \times (1-\delta)$ confidence level is

$$\mu_h \pm z_{\delta/2} \sqrt{v_h} \quad (18)$$

where μ_h is the forecast mean (i.e. the point forecast), v_h is the forecast variance, and $z_{\delta/2}$ is the critical value of the standard normal distribution (i.e. the value for which $100\% \times (1-\delta)$ of the normal distribution, with a mean of zero and a variance of one, lie between the values $-z_{\delta/2}$ and $+z_{\delta/2}$). The subscript h in (18) indicates that the forecast mean and variance are specified by the number of time period beyond the last observation being forecasted. A commonly used confidence level is 95%, for which the critical value is $z_{0.05/2} = 1.96$. However as mentioned earlier, OPP is interested only in the upper bound of the prediction interval. For such cases, one-sided prediction intervals are appropriate. The critical value for a one-sided prediction interval with a 95% confidence level would be $z_{0.05} = 1.645$. The equation for a prediction interval that incorporates this critical value is

$$\mu_h + 1.645\sqrt{v_h} \quad (19)$$

Note that the 95% confidence level used here is for exemplary purposes only; ultimately the selection of a confidence level is an OPP policy decision. Again, μ_h is an h -period-ahead point forecast, which was denoted earlier as $Y_n(h)$. Therefore the only term of (19) that has yet to be specified is the forecast variance, v_h . The following general formulae for the forecast variance are applicable to all three ES models:

$$v_1 = \sigma^2 \quad (20)$$

$$v_h = \sigma^2 \left[1 + \sum_{j=1}^{h-1} c_j^2 \right] \text{ for } h \geq 2 \quad (21)$$

The term σ^2 refers to the error variance that specifies the normal distribution of the model errors mentioned earlier in this paper. The formula for calculating the error variance is

$$\sigma^2 = \sum_{i=1}^n (\varepsilon_i - \bar{\varepsilon})^2 \quad (22)$$

The mean of the error terms, $\bar{\varepsilon}$ is assumed to be equal to zero, but the individual error terms are unknown. Therefore their estimates, the ε_i 's are used. Using the estimates for the error terms, an estimate for σ^2 can be calculated.

$$\sigma^2 = \sum_{i=1}^n \varepsilon_i^2 \quad (23)$$

This estimate for the error variance, σ^2 is also referred to as the mean square error (MSE), which will play a key role in parameter estimation later.

The last term of the error variance that needs to be specified is the c_j term. Calculation of the c_j term is model specific. For the simple, linear, and damped-trend models the equations for calculating the c_j term are, respectively,

$$c_j = \alpha \quad (24)$$

$$c_j = \alpha(1 + j\beta) \quad (25)$$

$$c_j = \alpha(1 + \phi_j\beta) \quad (26)$$

Note that (17) is used to calculate the ϕ_j term in (26). Substituting (23), (24), (25), and (26) into (21) yields the model specific forecast variances.

$$v_h = \sigma^2 \left[1 + \sum_{j=1}^{h-1} \alpha_j^2 \right] \text{ for } h \geq 2 \quad (27)$$

$$v_h = \sigma^2 \left[1 + \sum_{j=1}^{h-1} \alpha^2 (1 + j\beta) \right] \text{ for } h \geq 2 \quad (28)$$

$$v_h = \sigma^2 \left[1 + \sum_{j=1}^{h-1} \alpha^2 (1 + \phi_j \beta) \right] \text{ for } h \geq 2 \quad (29)$$

The forecast variance for a one-period-ahead forecast is the same for all models and is specified in equation (20). Utilizing equations (19), (20), (27), (28), and (29), the upper prediction interval for any h -period-ahead point forecast can be calculated. The formulae for the forecast variances indicate that the interval forecasts depend upon many factors: the forecast errors, the value of the model parameters (α , β , and ϕ), and the number of time periods beyond the last observation being forecasted. Generally the larger these factors, the larger the prediction interval.

3.2.2 Interval Forecasts for Multiple Point Forecasts

So far the discussion of interval forecasts has pertained to calculating an upper bound for a single point forecast. For a one-sided prediction interval, there is a $100\%X(1-\delta)$

probability that the *individual* point estimate, μ_h will be less than or equal to $\mu_h + z_\delta \sqrt{v_h}$. However OPP is interested in obtaining an upper bound for *multiple* point forecasts.

Calculating a multiple-year prediction interval begins with computing the forecast variance for each year that a forecast has been calculated. For OPP's purposes, the number of years forecasted will generally be between three and five. Once the forecast variance of each point forecast has been calculated, the simultaneous UPI can be obtained. The UPI is chosen such that the product of the confidence levels is equal to 95% (approximately). The probability that each point forecast *does not exceed* the UPI is calculated, using the forecast variance for each forecast point. In order to compute these individual probabilities, one must first

standardize the UPI. This is done for each point forecast $[Y_n(i)]$ by subtracting it from the UPI (Y_{UPI}) and dividing by the square root of its forecast variance (v_i),

$$z_i = \frac{Y_{UPI} - Y_n(i)}{\sqrt{v_i}}$$

Here the term i refers to the i^{th} point forecast of the time series. These standardized values, the z_i 's are assumed to be associated with a normal distribution. Each z_i has an associated probability denoted as $P_i(Z \leq z_i)$. That is, the probability of observing a value less than or equal to z_i , the standardized value of the UPI. This probability represents the confidence level of Y_{UPI} associated with the $Y_n(i)$ forecast.

Assuming independence, the probability (P^c) that PCT does not exceed the upper prediction interval *in any of the forecasted years* is equal to the product of these individual probabilities,

$$P_c(Z \leq z_1 | \dots | Z \leq z_h) = \prod_{i=1}^h P_i(Z \leq z_i)$$

Here h denotes the number of point forecasts being made. Thus P^c is the combined probability of observing values less than or equal to z_i for all h z_i 's, which represents the confidence level of the simultaneous UPI for all h point forecasts. Since the forecast variance and the point forecasts are fixed values, the UPI can be found by varying its value until the combined probability is approximately equal 0.95. It is worth noting that this method of computing simultaneous UPIs relies on the assumption that the individual probabilities are independent. The simultaneous upper prediction intervals were calculated for all of the previous examples and are shown below.

Table 7. Examples of the interval forecasts for all three models (SES, LES, and DES)

Table 7f

crop:	rowcrop2					
chemical:	insecticide4					
$\alpha =$	0.10					
$\beta =$	0.60					
$\phi =$	0.90					
	<i>Actual</i>	<i>Forecast</i>	<i>Forecast</i>			<i>UPI</i>
<i>Year</i>	<i>PCT</i>	<i>PCT</i>	<i>Error</i>	<i>Level</i>	<i>Trend</i>	<i>PCT</i>
				1.44	0.64	
1987	0.77	2.08	-1.31	1.95	0.49	
1988	3.13	2.44	0.69	2.51	0.49	
1989	4.93	2.99	1.94	3.19	0.55	
1990	3.59	3.74	-0.16	3.73	0.49	
1991	4.59	4.22	0.38	4.25	0.46	
1992	6.08	4.72	1.37	4.85	0.50	
1993	5.69	5.35	0.34	5.38	0.47	
1994	4.62	5.85	-1.24	5.73	0.35	
1995	8.18	6.08	2.10	6.29	0.44	
1996	6.76	6.73				20.08
1997	12.70	7.12				20.08
1998	11.11	7.48				20.08
1999	11.86	7.80				20.08
2000	10.16	8.09				20.08

Table 7e: Damped model (DE,S)

crop:	rowcrop1					
chemical:	herbicide3					
$\alpha =$	0.20					
$\beta =$	0.80					
$\phi =$	0.90					
	<i>Actual</i>	<i>Forecast</i>	<i>Forecast</i>			<i>UPI</i>
<i>Year</i>	<i>PCT</i>	<i>PCT</i>	<i>Error</i>	<i>Level</i>	<i>Trend</i>	<i>PCT</i>
				11.21	1.38	
1987	11.80	12.59	-0.79	12.43	1.12	
1988	14.40	13.55	0.85	13.72	1.14	
1989	17.60	14.86	2.74	15.41	1.47	
1990	17.11	16.87	0.23	16.92	1.36	
1991	15.64	18.28	-2.64	17.75	0.80	
1992	19.30	18.55	0.75	18.70	0.84	
1993	19.47	19.54	-0.07	19.52	0.74	
1994	23.53	20.27	3.27	20.92	1.19	
1995	24.18	22.11	2.07	22.53	1.40	
1996	22.46	23.93				50.29
1997	23.93	25.20				50.29
1998	20.18	26.33				50.29
1999	21.32	27.36				50.29
2000	22.35	28.28				50.29

4 Forecasting Methodology

This section describes the automated procedure proposed by OPP for forecasting “likely average PCT” and “likely maximum PCT” values of pesticide crop combinations for which OPP has historic use information. The first part of this section will explain the process for estimating model parameters, selecting the “best” model, and initializing the model components (i.e. the level and trend components). The second part section will provide a brief outline of the steps in the forecasting methodology. This outline serves as a framework for the subsequent encoding of the methodology. The encoding of the forecasting methodology was done using the SAS IML® software. The third part of this section will therefore describe the purpose of each line of code by explaining the syntax. It is important to keep in mind that this forecasting methodology was developed to provide reasonable estimates of the “likely average PCT” and the “likely maximum PCT.” The methodology is meant to serve as an objective and automated procedure for estimating PCT values based on historical information that is already available to OPP. Frequently these PCT values will be needed for a pesticide that is used on several crops. Therefore the forecasting procedure must be reliable and reproducible method. However the forecasts provided by any such automated forecasting procedure will be considered a preliminary step in the process of providing estimate of PCT, which will be supplemented (and revised if necessary) by extensive professional judgment on the part of OPP analysts. Ultimately, the analyst assigned to the specific pesticide will verify that the forecasts are reasonable and will be given print-outs of the historic observations to identify any anomalies, such as outlying, missing, or inaccurate observations.

4.1 Initial Values, Parameter Estimation, and Model Selection

The previous section of this paper was devoted to providing descriptions, formulae, and examples for the three exponential smoothing models. By doing so, explanations to many steps of the forecasting methodology were given; these include calculating fitted values, point forecasts, and interval forecasts. However there are other important steps of the forecasting procedure that were not covered; these include initializing the level and trend component, estimating the model parameters, and selecting the appropriate model from the three available. This part of the section will discuss these additional details.

4.1.1 Initial Values

In order to use any exponential smoothing techniques, initial values for the level and trend must be obtained due to the recursive nature of the equations. If the PCT values are regressed on the time metric (year of the observation), the linear regression parameters can be used to find reasonable initial values for the level and trend. The regression parameters, slope and intercept, are calculated using the ordinary least squares method. For the simple ES model, the initial level, l_0 is set equal to the “predicted” value of the initial observation as calculated from the

regression model. From (2) we see that the first fitted value, \hat{Y}_1 , will therefore be the predicted value of the linear regression model. For the linear and damped-trend ES models, the first fitted value will also be the predicted value from the regression model. However the slope of the regression model, β_1 , will serve as the initial trend, b_0 for both models. From (8) we see that the fitted value is the sum of the level and trend. Since the goal is to have the first fitted value be equal to the predicted value, the initial level is set equal to the difference of the predicted value and the trend. Using the OLS regression parameters, initial values for the level and trend are calculated. These values are then used to calculate the first or initial fitted value for each model.

4.1.2 Estimating Parameters and Model Selection

Forecast errors, the mean square error, and the Bayesian information criterion are important concepts for understanding how one goes about choosing the “best” smoothing parameters and the “best” model. Exponential smoothing is a forecasting technique that uses data arranged in a time series, which can be denoted as Y_1, Y_2, \dots, Y_n . For each of the observations in the time series, ES

provides a fitted value denoted as $\hat{Y}_1, \hat{Y}_2, \dots, \hat{Y}_n$. Although typically associated with cross-sectional data for which the goal is to interpolate unobserved values, certain aspects of the familiar linear regression model provide useful analogies to exponential smoothing. For example, the fitted values of an exponential smoothing model are analogous to the “predicted” values associated with linear regression models. Additionally, forecast errors are analogous to the “residuals” of a linear regression model. Each forecast error is the difference between an actual

value (observation) and its fitted value, symbolically $\varepsilon_i = Y_i - \hat{Y}_i$ for $i = 1, 2, \dots, n$. As with residuals, large forecast errors indicate a poor fit to the data. Just as the estimation of regression parameters is based on minimizing the

residuals, obtaining estimates for the smoothing parameters and damping coefficients is based on minimizing the forecasting errors.

The mean square errors (MSE) is a widely used statistics that measures the relative “goodness-of-fit” of a model. The term “relative” is employed here to emphasize the fact that the MSE should not be used to compare the performance of models fitted to different data sets, since its magnitude is determined in part by the observations’ unit of measurement. As its name implies, the MSE is the algebraic mean of the squared forecast errors,

$$MSE = \frac{1}{n} \sum_{i=1}^n \varepsilon_i^2$$

The parameters of the ES models parameters α , β , and are bounded between the values zero and one. For a given ES model (simple, linear, and damped trend), the parameter value(s) that produces the smallest MSE is/are selected as the “best” or optimal parameter(s).

Not only it useful in developing parameter estimates, the MSE also serves as an important component in developing prediction intervals or interval forecasts. To obtain interval forecast, the forecast variance must be calculated. The forecast variance is dependent upon how many time periods (years) beyond the last observation are being forecasted, the value of the model’s parameters, and the MSE. For all of the models, the variance for a forecast that is one-time-period ahead is simply the MSE, $v_1 = MSE$. In general, for an h-time-period ahead forecast the variance is

$$v_h = MSE \left[1 + \sum_{j=1}^{h-1} c_j \right]$$

The formula for the c_j term is model specific and generally increases in magnitude as the number of time periods being forecasted increases. However, the formula for the forecast variance attests to the fact that the MSE is essential to the process of calculating interval forecasts. It is worth noting that the larger the MSE, the larger the prediction interval. Thus, a model with a better fit to the historical data will tend to produce smaller prediction intervals.

Once the parameters for each model have been optimized for a given data set, the next step is to select the “best” ES model out of the three. One could use the MSE to select the model with the best fit. However, consider a particular data set for the damped-trend ES model, which requires the estimation of three separate parameters, has an MSE that is slightly less than that of the simple ES model, which only requires the estimation of one parameter. In such a situation where the MSE is used as the selection criterion, the more complex model would be selected, even though with the added benefit of having two more parameters, only marginally improves the fit to the data. An improvement upon the MSE would be a selection criterion that “penalizes” a model for incorporating more parameters. As mentioned earlier, a smaller MSE indicates a better fit to the data. Therefore a more appropriate selection criterion would generate a larger value for a model that has more parameters. The Bayesian Information Criterion (BIC) is one such selection criterion. The equation for the BIC is

$$BIC = n \ln(MSE) + k \ln(n)$$

For the above equation, n is the number of observations in the time series, k is the number of parameter estimates for the model, and $\ln(\cdot)$ is the natural logarithm function. All other factors (n and MSE) equal, the BIC would be smaller for the model with the least number of parameters. There the BIC is the model selection criterion that is used in the forecasting methodology.

4.2 Outline of Forecasting Methodology

The basic outline of the forecasting methodology is as follows:

1. Initialize the level and trend for each model using OLS linear regression.
2. Estimate the parameters (α , β , and ϕ) for each model by minimizing the MSE.
3. Calculate the forecast variance for each model.
4. Calculate the point forecasts for each model.
5. Calculate the interval forecast for each model.
6. Choose the “best” (i.e. optimal) model based on the BIC model selection criterion.
7. Save each of the models as SAS data sets.
8. Provide a graph for each model containing point forecasts, the interval forecast, the historic observations, and the BIC.

4.3 Coded Version of the Forecasting Methodology

The following part of this section will describe each line of code by explaining the syntax and predefined functions of the SAS IML® code. A print-out of the code will be included in the appendix of this paper.

4.3.1 Preliminary Data Manipulation

The *libname* command is used to assign a library name to the working directory. The next step involves importing a Microsoft Excel® spreadsheet file as a SAS® data file using the *proc import* command. The spreadsheet file and the subsequent data file contain the PCT values, the year of the observation, the name of the active ingredient (pesticide), and the name of the crop arranged in column format with the variable names at the top: *pct*, *year*, *ai*, and *crop*, respectively. After importing the data file, the *proc iml* command is used to invoke the SAS IML® software. The *use* command specifies the target SAS® data file and the *read* command specifies which variable names are to be used from the data file. By default these variables are read in as column vectors. The *show names* command is used simply to verify that the column vectors were properly read with the correct number of elements. The number of observations is stored as the scalar *n* by counting the number elements in the variable *pct* using the *nrow* function. Although the elements of the *pct* variable are counted, all of the column vectors have the same number of elements at this point. Next the number of years to be forecasted is hard-coded as the scalar *h*. The variables *ai* and *crop* are then reshaped from column vectors of length *n* to column vectors of length *n + h* using the *shape* function. Utilizing the *j* function, the variable *year_for* is created as a column vector (*n* x 1) with each element initialized to a value of one. A *do* loop is then employed to populate *year_for* with the values of the years to be forecasted. These values are generated to account for regular observation intervals other than annual. The column vector *year_h* is created by vertically appending *year_for* to *year* using the *//* operator. The (*n* x 2) matrix *year_one* is made by horizontally appending a column vector of ones (1's) to the column vector *year*.

4.3.2 Linear Regression Parameter Estimation

The purpose of creating the *year_one* variable with a column of ones is to have the intercept parameter estimated in the ordinary least squares (OLS) calculations. If the *year* variable were used in the calculations, only the slope parameter would be estimated. The *year_one* matrix and the *pct* column vector are arranged in a linear algebraic equation to yield the OLS parameters utilizing the matrix inversion function *inv* and the matrix transpose operator *'*. The result of the algebraic equation is a (2 x 1) column vector containing the regression parameter estimates (slope and intercept), which is stored in the column vector *b*. These parameters will be used to initialize the level and trend components of the exponential smoothing models.

4.3.3 Model Initialization and Parameter Estimation

After computing the regression parameters, the column vectors that will contain the fitted values, forecast error terms, the level component, and trend component are initialized to contain the value -999 for every element. This initialization scheme is performed repeatedly for other variables throughout the program in order to facilitate the debugging process. The names of these initialized variables are *pct_fit*, *error*, *level*, and *trend* respectively. These variables will be reused and consequently overwritten throughout the program. However, the values stored in these variables will be conditionally saved to other variables. The scalar variables *opt_ses_mse*, *opt_les_mse*, and *opt_des_mse* are initially assigned the value 99999. These variables will contain the mean square error (MSE) of the models with smallest MSE for the three exponential

smoothing models (simple, linear, and damped-trend). These variables are assigned a large value to facilitate the model selection process, which is based on minimizing the MSE within each of three classes.

The process of estimating the parameters for the ES models begins with the simple exponential smoothing (SES) model. First the fitted or predicted value for the first PCT observation as estimated by the linear regression model, is stored in the first element of the level component, *level[1]*. This value is calculated using the regression parameters stored in the variable *b*. Thus for the first observation, the fitted value of the simple ES model will be equal to that of the linear regression model, since the first fitted value of the simple model, *pct_fit[1]*, is equal to *level[1]*. The SES model has only one parameter that needs to be estimated: the level smoothing parameter, *alpha*. Utilizing *do* loops, the SES model is fitted to the PCT values and the MSE is calculated for every value of *alpha* with a two decimal place precision. The first *do* loop cycles through the possible values of *alpha* (zero to one) in 0.01 increments. The next *do* loop is used to calculate the fitted values, the forecast error, and the level component for all *n* observations.

Once the model has been fitted for a specific value of *alpha*, the MSE is calculated using the *ssq* function, which sums the square of the elements of the function argument, in this case *error*. This MSE value is stored in the variable *mse*. Next, an *if-then* statement tests if *mse* is the minimal MSE for the simple model. If the MSE for the calculated model is the minimal value, the model components are stored and the Bayesian information criterion (BIC) for the model is calculated and also stored. The variables *mse*, *alpha*, *pct_fit*, *level*, and *error* are stored in the variables *opt_ses_mse*, *opt_ses_alpha*, *opt_ses_fit*, *opt_ses_level*, and *opt_ses_error*, respectively. Additionally, the number of parameters of the model is stored in the variable *p*. For the SES model this value is one, since the only parameter is the level smoothing parameter, *alpha*. The BIC for the model is calculated utilizing the *log* function, which calculates the natural logarithm of its argument. To be consistent with the other ES models, the values zero, one, and one are stored in the variables *opt_ses_beta*, *opt_ses_phi*, and *opt_ses_trend*, respectively.

The process of conditionally saving the model components may be repeated several times before all 101 values of *alpha* have been cycled through. Ultimately however, these variables will contain the model components of the SES model that results in the minimal MSE (i.e. best fits the PCT data). Thus the *alpha* parameter will have been estimated and stored in the variable *opt_ses_alpha*.

The procedure for estimating the parameters for the linear and damped-trend models is very similar to that used for the simple model. However, one aspect in which the procedures do differ is the initialization of model components, namely the level and trend. This difference is primarily due to the fact that the simple model does not have a trend component. As with the simple model, the initial fitted value, *pct_fit[1]*, is to be equal to the fitted or predicted value of the first observation as estimated by the linear regression model. The fitted value of the simple model is equal to the level, whereas the fitted value for the linear and damped-trend models is equal to the sum of the level and trend. Thus for both the linear and damped-trend models, the trend component is initialized to the value of the regression slope parameter. Additionally the initial level component is set equal to the predicted value of the first observation as estimated by

the linear regression model minus the slope parameter. Thus for the linear and damped-trend models, initial fitted value, which is the sum of the first level component ($level[1]$) and the first trend component ($trend[1]$) is equal the predicted value of the regression model.

In addition to the *do* loop for $alpha$, there are two more *do* loops: one for the trend smoothing parameter, $beta$ and one for the damping coefficient, phi . By nesting the *do* loops, all the values for the model parameters are cycled through with a two decimal precision. Since the calculations for the level and trend components are identical for both models, the same nested *do* loops can be used for the linear and damped-trend models. Note that when the damping coefficient is equal to one, the equation for the trend component of the damped-trend model, (14) reduces to the corresponding equation of the linear model, (10). Following the example of Hyndman et al (2002), phi is restricted to be greater than $beta$.

As with the simple model, once a model has been fitted to the PCT value with a set of parameter values, the mean square error (MSE) is calculated. A series of *if-then* loops are then used to select the models with optimal parameter values (i.e. those that minimize the MSE). The first *if-then* statement ensures that phi is equal to one (i.e. meets the criterion for being a linear model) and that the MSE is minimal before storing the linear model components. Once the model parameters have been verified to produce the minimal MSE for the appropriate model, the components for the model are saved. The procedure for creating variable names for the various components is similar to that used for the simple model. The values of the variables mse , $alpha$, $beta$, phi , pct_fit , $level$, $trend$, and $error$ are stored as opt_les_mse , opt_les_alpha , opt_les_beta , opt_les_phi , opt_les_fit , opt_les_level , and opt_les_error , respectively. Additionally, the number of parameters for the model is stored in the variable p , which is then used to calculate the model's Bayesian information criterion (BIC), which in turn is stored in the variable opt_les_bic . Although the damping coefficient, phi , is stored for the linear model, this parameter does not contribute to the number of parameter for the model since its value is always equal to one and thus need not be calculated. Out of the 101^2 combinations of $alpha$ and $beta$, eventually the components of the linear model that produces the smallest MSE will be stored.

The second *if-then* statement verifies that phi is not equal to one or zero¹ (i.e. meets the criterion for being a damped-trend model) and that the MSE value is minimal, before saving the components of the damped-trend model. The values for mse , $alpha$, $beta$, phi , pct_fit , $level$, $trend$, and $error$ are stored in opt_des_mse , opt_des_alpha , opt_des_beta , opt_des_phi , opt_des_fit , opt_des_level , and opt_des_error , respectively. For the damped-trend model, the number of parameters, p , is set equal to three and the BIC is calculated and stored in the variable opt_des_bic . After cycling through all $(101^2)(102)/2$ unique combinations of $alpha$, $beta$, and phi , the model with the parameter values that produces the minimal MSE will have its components stored. Thus the parameters of the linear and damped-trend models will be estimated.

4.3.4 Forecast Variance and Point Forecast Calculations

The ($h \times 1$) variables that will contain the forecast variances for the simple, linear, and damped-trend models are *ses_for_var*, *les_for_var*, and *des_for_var*, respectively. Each element of these variables is initialized to the value -999 for debugging purposes. The first element of each of these variables will contain the one-step-ahead forecast variance; the second element will contain the two-step-ahead forecast variance; and generally the h^{th} element will contain the h -step-ahead forecast variance. Consistent with equations (20) and (23), the forecast variance for a one-step-ahead forecast is set equal to the mean square error for each of the three models. The values of the remaining elements of these variables will be calculated using a *do* loop. The calculations of the forecast variances for these models are consistent with Hyndman et al (forthcoming). The ($h \times 1$) variables *geo_sum_phi* and *sum_geo_sum_phi* will be used to calculate the forecast variance for the damped-trend model. The elements of both of these variables are also initialized to the value -999 to aid in the debugging process. Generally, the h^{th} element of *geo_sum_phi* will contain the sum of an h -term geometric series, where the first term is one and the rate is equal to *phi* (i.e. $a = 1$ and $r = \phi$). For example, the third element of

geo_sum_phi would be equal to $1 + \phi + \phi^2$. Generally, the h^{th} element of *sum_geo_sum_phi* will contain the sum of the first h elements of *geo_sum_phi*. For example, the third element of *sum_geo_sum_phi* would be equal to $(1) + (1 + \phi) + (1 + \phi + \phi^2)$. The first element of both these variables are always equal to one and are thus initialized to this common value. Since the first element of *ses_for_var*, *les_for_var*, *des_for_var*, *geo_sum_phi* and *sum_geo_sum_phi* were already specified, the *do* loop calculates the values for the remaining elements of these variables starting with the second element. The forecast variances contained in *ses_for_var*, *les_for_var*, and *des_for_var* will be used to calculate the interval forecast later in the program.

Next the point forecasts for the three models will be calculated. The ($h \times 1$) variables that will contain the point forecasts for the simple, linear, and damped-trend models are *ses_for*, *les_for*, and *des_for*, respectively. The elements of these variables are also initialized to the value -999. The h point forecasts for each of models are calculated using a *do* loop. The calculations of the point forecasts are consistent with equations (6), (13), and (16).

4.3.5 Interval Forecast Calculations

Once the point forecasts and forecast variances have been calculated for the various exponential smoothing models, the interval forecast for each model is calculated. Since the basic principle for computing the interval forecast is the same for all three models, a line-by-line explanation of the interval forecast calculation will be given for only one of the models. As explained earlier in this paper, OPP would like to calculate an interval forecast that would serve as upper bound for multiple point forecasts. For an interval forecast, there is an associated probability that refers to the likelihood that the specified future observations will less than the upper bound (or greater than, if in reference to a lower bound). For this program, that probability is calculated for each future observation for which a point forecast was computed; then the product of these individual probabilities is calculated. Under the assumption of mutual independence, the product of these individual probabilities represents the joint probability that all of the future observations will be less than the interval forecast.

First, the variable *delta* is assigned the value 0.0001. This variable is used to adjust the value of the interval forecast. For the simple model *temp_upi* is set equal to last point forecast for the SES model, *ses_for[h]* plus twice the standard deviation of the point forecast (i.e. twice the square root of the forecast variance). This value provides a reasonable starting point for finding the interval forecast. Eventually, the variable *sim_prob* will contain the product of the individual probabilities associated with each point forecast. Since its value will be determined multiplicatively, *sim_prob* is initially assigned the value 1.0. A *do-while* loop is used to test if *sim_prob* is approximately equal to the 0.95 (i.e. 95%). Within the *do-while* loop, *sim_prob* is assigned the value 1.0 again and

ses_sim_upi is set equal to the previously calculated *temp_upi*. Next, a *do* loop is used to calculate the value of *sim_prob*. The *probnorm* function is used to compute the probability that the future observation will be less than the

ses_sim_upi for each of the $h=5$ point forecasts. Calculating the probabilities in this manner, assumes that the forecast errors are normally distributed, since the *probnorm* function is used to calculate the probability of observing a value as large as *ses_sim_prob* from a normal distribution with a mean of *ses_for[k]* and a standard deviation of

$\sqrt{ses_for_var[k]**0.5}$ (i.e. $\sqrt{ses_for_va[k]}$). After calculating the probability for a specific point forecast, *sim_prob* is reassigned to be equal to the product of this probability and its previous value. After cycling through all *h* point forecasts, *sim_prob* will contain the product of all *h* probabilities. Utilizing an *if-then-else* statement, the value of *temp_upi* is adjusted. If *sim_prob* is less than .95, then is set equal to *ses_sim_upi* plus *delta*; otherwise *temp_upi* is set equal to *ses_sim_upi* minus *delta*. In other words, if the simultaneous probability is less than the sought probability, the interval forecast is slightly increased (by a 0.0001 increment); otherwise the interval forecast is slightly decreased. This process of calculating the simultaneous probability for an interval forecast and then adjusting the value of the interval forecast is repeated until the simultaneous probability is approximately equal to 95%. Utilizing this procedure, the interval forecast for the simple, linear, and damped-trend models will be stored in the variables *ses_sim_upi*, *les_sim_upi*, and *des_sim_upi*, respectively.

4.3.6 Model Selection

After calculating the interval forecasts for the exponential smoothing models, the “best” of the three models is selected. The model with the smallest Bayesian information criterion (BIC) is selected as the “best” model and its components (smoothing parameters, fitted values, and forecast values) are saved to another set of variables. A series of *if-then-else* statements is used to determine which of the models (simple, linear, and damped-trend) results in the minimal BIC. The first *if-then* statements tests if the BIC of the simple model is less than or equal to that of the linear and damped-trend models. Recall that the BICs of the simple, linear and damped-trend models are stored in the variables *opt_ses_bic*, *opt_les_bic* and *opt_des_bic* respectively. If the BIC of the simple model is determined to be the minimum of the three, then its components, stored in the variables *opt_ses_bic*, *opt_ses_alpha*, *opt_ses_beta*, *opt_ses_phi*, *opt_ses_fit*, *opt_ses_level*, *opt_ses_trend*, *ses_sim_upi* and *ses_for* are stored in the variables *opt_es_bic*, *opt_es_alpha*, *opt_es_beta*, *opt_es_phi*, *opt_es_fit*, *opt_es_level*, *opt_es_trend*, *es_sim_upi* and *es_for*. This process is repeated for linear model using a similarly constructed *if-then* statement. If the BICs of these models are not the minimum, then the damped-trend model is selected as the “best”. As with the other two models, if the damped-trend BIC is the minimal value, then the model components are saved to similarly named variables with an *opt_es* prefix.

4.3.7 Data Set Construction

Before constructing the SAS® data sets that will contain the components from each of the models, some data manipulation is performed to prepare the data to be graphical displayed. First the variables *ses_for_h*, *les_for_h*, *des_for_h* and *es_for_h* are created. For each of these variables the *pct* is vertically appended to the variables containing the point forecasts of the various models. Recall that the *pct* variable contains the historic percent crop treated observations and the point forecasts for the simple, linear and damped-trend models are stored in *ses_for*, *les_for* and *des_for* respectively. Additionally the variable *es_for* will contain a duplicate set of point forecasts of the model with minimal Bayesian information criterion (BIC). Next for each model, the variables that contain the BIC, alpha, beta and phi are reshaped from scalars (i.e. 1X1 matrices) to column vectors of length $n + h$ using the *shape* function. Similarly the variables containing the interval forecasts from the various models are reshaped from scalars to column vectors of length *h*.

Once the above variables have been reshaped, the data sets to contain the model components are constructed using the *create* command. The first data set constructed contains the model

components for the simple exponential smoothing model. The term *es.ses_sum* specifies the name of the data set being created, *ses_sum* and the library to which the data set will be saved, *es*. Recall that *es* refers to the working directory specified at the beginning of the program. The term *var* specifies that the data set will contain the variables listed within the braces. Similar data sets are created for the linear and damped-trend model in addition to the data set *es_sum*, which will be a duplicate data set of the components of the model with the minimal BIC. The *append* option is used after each *create* command to ensure that these data sets are not mistakenly overwritten. After the four data sets have been created the *iml* procedure is terminated with the *quit* command. The *run* command specifies that the code contained within the *iml* procedure is to be executed.

4.3.8 Graphical Display

Once the data sets for the exponential smoothing models have been constructed, a graphical display of the historic and forecasted percent crop treated (PCT) values will be provided. The *gplot* procedure is used to create the graphical representation of these values. Since this segment of the code pertains to cosmetic details of the graphical displays, an explanation of options and commands for the *gplot* procedure will not be given. The output of this section of the code will be three graphs, one for each of the exponential smoothing models. Each graph will have a title indicating the pesticide crop combination and model type (simple, linear or damped-trend) being represented. A legend will also be included that specifies that the historic PCT values are represented by solid circles connected by a solid line, the point forecasts are represented by unfilled circles connected by a dashed line and the interval forecasts are represented by dashed line. Additionally the value of Bayesian information criteria will be reported beneath the legend. Appropriate labels for the x-axis and y-axis will also be included that specify the year and the PCT value for the observations/forecasts. These graphs will be supplied to the analyst assigned to the pesticide as a tool for verifying that the forecasts are reasonable and appropriate.

References

- Gardner, E.S. Jr. (Feb 1987), "Short-Range Forecasting", *Lotus*, Vol. 3, No. 2, 54-58.
- Gardner, E.S. Jr. (Mar 1988), "Forecasting with Exponential Trends", *Lotus*, Vol. 4, No. 3, 27-30.
- Gardner, E.S. Jr. (May 1988), "How to Detect Trends and Seasonal Cycles", *Lotus*, Vol. 4, No. 5, 44-50.
- Gardner, E.S. Jr. (Mar 1987), "Using Exponential Smoothing", *Lotus*, Vol. 3, No. 3, 61-66.
- Hyndman, R.J., A.B. Koehler, R.D. Snyder, and S. Grose (2002) "A state space framework for automatic forecasting using exponential smoothing methods". *International J. Forecasting*, 18(3), 439-454.
Full text at <<http://www-personal.buseco.monash.edu.au/~hyndman/papers/>>
- Hyndman, R.J., A.B. Koehler, J.K. Ord, R.D. Snyder (forthcoming), "Prediction intervals for exponential smoothing state space models."
Full text at <<http://www-personal.buseco.monash.edu.au/~hyndman/papers/>>

Appendix

```
libname es 'c:\home\AQUA_Project\SAS_ES';

PROC IMPORT OUT= ES.temp_pct
DATAFILE= "C:\Home\AQUA_Project\SAS_ES_Macro\temp_pct.xls"
          DBMS=EXCEL2000 REPLACE;
          GETNAMES=YES;
          format ai $15.
          crop $15.;
RUN;

proc iml;
  use es.temp_pct;
  read all var{pct,year,ai,crop};
  show names;

n = nrow(pct);          /*Number of Observations*/
h = 5;                 /*Forecast Horizon*/

  ai = shape(ai,n+h,1);
  crop = shape(crop,n+h,1);

  year_for = j(h,1,1);

  do i = 1 to h;
    /*In case every other year*/
    year_for[i] = i * (year[n] - year[n-1]) + year[n];
  end;

  /*Append forecast years*/
  year_h = year // year_for;
```

```

/*Append column vector to estimate intercept*/
year_one = year || j(n,1,1);

/*OLS parameter estimates*/
b = inv(year_one`*year_one)*year_one`*pct;

pct_fit = j(n,1,-999); /*Specify dimensions of fitted*/
error = j(n,1,-999); /*values, error, level, and trend*/
level = j(n+1,1,-999); /*Use -999 for debugging*/
trend = j(n+1,1,-999);

/* Forecasting Process */

opt_ses_mse = 99999;
opt_les_mse = 99999;
opt_des_mse = 99999;

/* Simple Exponential Smoothing */

level[1] = b[1]*year[1] + b[2];

do alpha = 0 to 1 by 0.01;
do i = 1 to n;
pct_fit[i] = level[i];
error[i] = pct[i] - pct_fit[i];
level[i+1] = level[i] + alpha*error[i];
end;

mse = ssq(error)/n;

if mse < opt_ses_mse then do;
opt_ses_mse = mse;
/*number of parameters*/
p = 1;
/*BIC for SES*/
opt_ses_bic = n*log(opt_ses_mse) + p*log(n);
opt_ses_alpha = alpha;
opt_ses_beta = 0;
opt_ses_phi = 1;
opt_ses_fit = pct_fit;
opt_ses_level = level;
opt_ses_trend = 0;
opt_ses_error = error;

end;
end;

/* Linear and Damped-Trend Exponential Smoothing */

trend[1] = b[1];
level[1] = b[1]*year[1] + b[2] - b[1];

do beta = 0 to 1 by 0.01;
do phi = beta to 1 by 0.01;
do alpha = 0 to 1 by 0.01;
do i = 1 to n;
pct_fit[i] = level[i] + trend[i];
error[i] = pct[i] - pct_fit[i];
level[i+1] = level[i] + trend[i] + alpha*error[i];

```

```

trend[i+1] = phi*trend[i] + alpha*beta*error[i];
end;

mse = ssq(error)/n;

if (int(phi) = 1 & mse < opt_les_mse) then do;
    opt_les_mse = mse;
    /*number of parameters*/
    p = 2;
    /*BIC for LES*/
    opt_les_bic = n*log(opt_les_mse) + p*log(n);
    opt_les_alpha = alpha;
    opt_les_beta = beta;
    opt_les_phi = phi;
    opt_les_fit = pct_fit;
    opt_les_level = level;
    opt_les_trend = trend;
    opt_les_error = error;

end;

if (int(phi) ^= 1 & phi ^= 0 & mse < opt_des_mse) then do;
    opt_des_mse = mse;
    /*number of parameters*/
    p = 3;
    /*BIC for DES*/
    opt_des_bic = n*log(opt_des_mse) + p*log(n);
    opt_des_alpha = alpha;
    opt_des_beta = beta;
    opt_des_phi = phi;
    opt_des_fit = pct_fit;
    opt_des_level = level;
    opt_des_trend = trend;
    opt_des_error = error;
end;
end;
end;

ses_for_var = j(h,1,-999);
les_for_var = j(h,1,-999);
des_for_var = j(h,1,-999);

ses_for_var[1] = opt_ses_mse;
les_for_var[1] = opt_les_mse;
des_for_var[1] = opt_des_mse;

geo_sum_phi = j(h,1,-999);
sum_geo_sum_phi = j(h,1,-999);

geo_sum_phi[1] = 1.0;
sum_geo_sum_phi[1] = 1.0;

/* From "Prediction intervals for exponential smoothing state space
models" (Hyndman, et al) */
do k = 2 to h;
    geo_sum_phi[k] = geo_sum_phi[k-1] + opt_des_phi**(k-1);
    sum_geo_sum_phi[k] = sum_geo_sum_phi[k-1] + geo_sum_phi[k];
end;

```

```

ses_for_var[k] = (ses_for_var[1])*(1 + (opt_ses_alpha**2)*(k-1));
les_for_var[k] = (les_for_var[1])*(1 + (opt_les_alpha**2)*(k-1)*
(1 + 2*opt_les_beta*k + (opt_les_beta**2)*
k*(2*k-1)/6));
des_for_var[k] = (des_for_var[1])*(1 + (opt_des_alpha**2)*(k-1) +
(opt_des_alpha**2)*opt_des_beta*
(2*sum_geo_sum_phi[k-1] + opt_des_beta*
(sum_geo_sum_phi[k-1]**2)));
end;

/*Optimal (smallest MSE) ES forecast for each class
(simple,linear,and damp-trend)*/

ses_for = j(h,1,-999);
les_for = j(h,1,-999);
des_for = j(h,1,-999);

do k = 1 to h;
ses_for[k] = opt_ses_level[n+1];
les_for[k] = opt_les_level[n+1] + k*opt_les_trend[n+1];
des_for[k] = opt_des_level[n+1] +
geo_sum_phi[k]*opt_des_trend[n+1];
end;

delta = 0.0001;

temp_upi = ses_for[h] + 2*(ses_for_var[h]**0.5);
sim_prob = 1.0;

do while(abs(sim_prob - .95) > .001);
sim_prob = 1.0;
ses_sim_upi = temp_upi;

do k = 1 to h;
sim_prob = sim_prob * probnorm((ses_sim_upi - ses_for[k])/
(ses_for_var[k]**0.5));
end;

if(sim_prob < .95) then temp_upi = ses_sim_upi + delta;
else temp_upi = ses_sim_upi - delta;
end;

temp_upi = les_for[h] + 2*(les_for_var[h]**0.5);
sim_prob = 1.0;

do while(abs(sim_prob - .95) > .001);
sim_prob = 1.0;
les_sim_upi = temp_upi;

do k = 1 to h;
sim_prob = sim_prob * probnorm((les_sim_upi - les_for[k])/
(les_for_var[k]**0.5));
end;

if(sim_prob < .95) then temp_upi = les_sim_upi + delta;
else temp_upi = les_sim_upi - delta;

```

```

        end;

temp_upi = des_for[h] + 2*(des_for_var[h]**0.5);
sim_prob = 1.0;

do while(abs(sim_prob - .95) > .001);
    sim_prob = 1.0;
    des_sim_upi = temp_upi;

    do k = 1 to h;
sim_prob = sim_prob * probnorm((des_sim_upi - des_for[k])/
    (des_for_var[k]**0.5));
        end;

if(sim_prob < .95) then temp_upi = des_sim_upi + delta;
    else temp_upi = des_sim_upi - delta;
    end;

if ( (opt_ses_bic <= opt_les_bic) & (opt_ses_bic <= opt_des_bic) )
    then do;
        opt_es_bic = opt_ses_bic;
        opt_es_alpha = opt_ses_alpha;
        opt_es_beta = opt_ses_beta; /*Should be zero*/
        opt_es_phi = opt_ses_phi;
        opt_es_fit = opt_ses_fit;
        opt_es_level = opt_ses_level;
        opt_es_trend = opt_ses_trend; /*Should be zero vector*/
        es_sim_upi = ses_sim_upi;
        es_for = ses_for;
    end;
else
if ( (opt_les_bic <= opt_ses_bic) & (opt_les_bic <= opt_des_bic) )
    then do;
        opt_es_bic = opt_les_bic;
        opt_es_alpha = opt_les_alpha;
        opt_es_beta = opt_les_beta;
        opt_es_phi = opt_les_phi;
        opt_es_fit = opt_les_fit;
        opt_es_level = opt_les_level;
        opt_es_trend = opt_les_trend;
        es_sim_upi = les_sim_upi;
        es_for = les_for;
    end;
else do;
        opt_es_bic = opt_des_bic;
        opt_es_alpha = opt_des_alpha;
        opt_es_beta = opt_des_beta;
        opt_es_phi = opt_des_phi;
        opt_es_fit = opt_des_fit;
        opt_es_level = opt_des_level;
        opt_es_trend = opt_des_trend;
        es_sim_upi = des_sim_upi;
        es_for = des_for;
    end;

ses_for_h = pct // ses_for;
les_for_h = pct // les_for;
des_for_h = pct // des_for;
es_for_h = pct // es_for;

```

```

    opt_ses_bic    = shape(opt_ses_bic,n+h,1);
    opt_ses_alpha = shape(opt_ses_alpha,n+h,1);
    opt_ses_beta  = shape(opt_ses_beta,n+h,1);
    opt_ses_phi   = shape(opt_ses_phi,n+h,1);
    ses_sim_upi   = shape(ses_sim_upi,h,1);

    opt_les_bic    = shape(opt_les_bic,n+h,1);
    opt_les_alpha  = shape(opt_les_alpha,n+h,1);
    opt_les_beta   = shape(opt_les_beta,n+h,1);
    opt_les_phi    = shape(opt_les_phi,n+h,1);
    les_sim_upi    = shape(les_sim_upi,h,1);

    opt_des_bic    = shape(opt_des_bic,n+h,1);
    opt_des_alpha  = shape(opt_des_alpha,n+h,1);
    opt_des_beta   = shape(opt_des_beta,n+h,1);
    opt_des_phi    = shape(opt_des_phi,n+h,1);
    des_sim_upi    = shape(des_sim_upi,h,1);

    opt_es_bic     = shape(opt_es_bic,n+h,1);
    opt_es_alpha   = shape(opt_es_alpha,n+h,1);
    opt_es_beta    = shape(opt_es_beta,n+h,1);
    opt_es_phi     = shape(opt_es_phi,n+h,1);
    es_sim_upi     = shape(es_sim_upi,h,1);

create es.ses_sum var {ai crop opt_ses_bic opt_ses_alpha
                        opt_ses_beta opt_ses_phi opt_ses_fit
                        opt_ses_level opt_ses_trend ses_for
                        ses_sim_upi ses_for_h pct year year_for
                        year_h};
                        append;

create es.les_sum var {ai crop opt_les_bic opt_les_alpha
                        opt_les_beta opt_les_phi opt_les_fit
                        opt_les_level opt_les_trend les_for
                        les_sim_upi les_for_h pct year year_for
                        year_h};
                        append;

create es.des_sum var {ai crop opt_des_bic opt_des_alpha
                        opt_des_beta opt_des_phi opt_des_fit
                        opt_des_level opt_des_trend des_for
                        des_sim_upi des_for_h pct year year_for
                        year_h};
                        append;

create es.es_sum var {ai crop opt_es_bic opt_es_alpha opt_es_beta
                        opt_es_phi opt_es_fit opt_es_level
                        opt_es_trend es_for es_sim_upi es_for_h pct
                        year year_for year_h};
                        append;

                        quit;
                        run;

                        options nobyline;
symbol1 color=black line=1 interpol=join value=dot height=1;

```

```

symbol2 color=black line=3 interpol=join value=circle height=1;
symbol3 color=black line=3 interpol=join value=none;

axis1 minor=none label=none value=(angle=90);
legend1 value=(h=.5
tick=1 j=c 'Historic'
      j=c 'PCT'
tick=2 j=c 'Forecasted'
      j=c 'PCT'
tick=3 j=c '95% UPI'
      j=c 'PCT')
label=none
across=1
shape=symbol(4.5,1)
position=(top right outside);

title1 h=2 '#byval(crop)-#byval(ai)';
title2 h=1 'Simple Exponential Smoothing';
title3 h=1 move=(65,10)
      'bic = #byval3';

proc gplot data=es.ses_sum;
  by ai crop opt_ses_bic;
plot pct*year ses_for_h*year_h ses_sim_upi*year_for
  / overlay haxis=axis1 legend=legend1;
run;

title2 "Linear Exponential Smoothing";
title3 h=1 move=(65,10)
      'bic = #byval3';
proc gplot data=es.les_sum;
  by ai crop opt_les_bic;
plot pct*year les_for_h*year_h les_sim_upi*year_for
  / overlay haxis=axis1 legend=legend1;
run;

title2 "Damped-Trend Exponential Smoothing";
title3 h=1 move=(65,10)
      'bic = #byval3';
proc gplot data=es.des_sum;
  by ai crop opt_des_bic;
plot pct*year des_for_h*year_h des_sim_upi*year_for
  / overlay haxis=axis1 legend=legend1;
run;
quit;

```

