Agronomy Journal

Software Code from
Symposium on Statistical Concepts

The March–April, 2015 edition of *Agronomy Journal* contains 11 articles that provide practical background information and advice for statistical concepts and procedures that our scientists often need to use in their research. In 10 of these articles, the authors provided software code that supports analyses discussed in their articles. That same code is provided here in this document to help others properly analyze their data. Most of the code uses SAS, although some authors provided code in GenStat or R in addition to or instead of SAS.
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fundamentals of Experimental Design: Guidelines for Designing Successful Experiments</td>
<td>3</td>
</tr>
<tr>
<td>Can Analysis of Variance Be More Significant?</td>
<td>8</td>
</tr>
<tr>
<td>“Is, or is not, the two great ends of Fate:” Errors in Agronomic Research</td>
<td>10</td>
</tr>
<tr>
<td>Evaluation and Interpretation of Interactions</td>
<td>15</td>
</tr>
<tr>
<td>Analysis and Interpretation of Interactions in Agricultural Research</td>
<td>22</td>
</tr>
<tr>
<td>Analysis of Combined Experiments Revisited</td>
<td>29</td>
</tr>
<tr>
<td>The Design and Analysis of Long-term Rotation Experiments</td>
<td>31</td>
</tr>
<tr>
<td>Non-linear Regression Applications in Agricultural Research</td>
<td>37</td>
</tr>
<tr>
<td>Multivariate Analysis: Greater Insights into Complex Systems</td>
<td>42</td>
</tr>
<tr>
<td>Rethinking the Analysis of Non-Normal Data in Plant and Soil Science</td>
<td>47</td>
</tr>
</tbody>
</table>
Fundamentals of Experimental Design:
Guidelines for Designing Successful Experiments

Michael D. Casler

ABSTRACT

We often think of experimental designs as analogous to recipes in a cookbook. We look for something that we like, something that satisfies our needs, and frequently return to those that have become our long-standing favorites. We can easily become complacent, favoring the tried-and-true designs (or recipes) over those that contain unknown or untried ingredients or those that are too complex for our tastes and skills. Instead, I prefer to think of experimental designs as a creative series of decisions that are meant to solve one or more problems. These problems may be real or imagined - we may have direct evidence of a past or current problem or we may simply want insurance against future potential problems. The most significant manifestation of a "problem" or a "failed" design is unsatisfactory p-values that prevent us from developing inferences about treatment differences. Four basic tenets or pillars of experimental design - replication, randomization, blocking, and size of experimental units - can be used creatively, intelligently, and consciously to solve both real and perceived problems in comparative experiments. Because research is expensive, both in terms of grant funds and the emotional costs invested in grant competition and administration, biological experiments should be designed under the mantra, "failure is not an option." Guidelines and advice provided in this review are designed to reduce the probability of failure for researchers who are willing to question, evaluate, and possibly modify their decision-making processes.

Michael D. Casler, USDA–ARS, U.S. Dairy Forage Research Center, 1925 Linden Dr., Madison, WI 53706-1108 USA. *Corresponding author (mdcasler@wisc.edu).

Abbreviations: RCBD, randomized complete block design; CRD, completely randomized design; \( V_{\text{DTM}} \), variance of a difference between treatment means; \( B \times T \), block \( \times \) treatment
SAS CODE FOR TUKEY’S TEST FOR NON-ADDITIVITY

SAS code to compute Tukey’s test for nonadditivity in a randomized complete block design. The test is computed in two parts: (1) output the predicted values from the mixed models analysis, then (2) square the predicted values and include this term as a single-degree-of-freedom regressor variable (covariate) as a test for multiplicative block and treatment effects (Sahai and Ageel, 2000).

```sas
data a; infile 'tukey.dat';
input rep block trt y;
proc mixed; class trt;
model pcb = trt / outpred=x;
random block; run;
data x; set x;
p2=pred*pred;
proc mixed; class trt;
model pcb = trt p2;
random block; run;
```

SAS CODE FOR PREDICTING THE POWER OF AN EXPERIMENT WITH TWO SOURCES OF ERROR: EXPERIMENTAL ERROR AND SAMPLING ERROR

SAS code to estimate power of a hypothesis test with the following parameters: treatment means = 95 and 100, variance components = 5 and 10 (experimental error and sampling error, respectively), \( r = 4 \) replicates, \( s = 2 \) observational units per experimental unit, and the assumption of normally distributed errors (adapted from Gbur et al., 2012). The code creates an exemplary data set with fixed treatment means, which is then analyzed by proc glimmix using fixed parameter values. The non-centrality parameter of the F-distribution is then used to estimate power. The user must set the following values as input data: treatment means or detection level, e.g. 95 vs. 100; prior estimates of experimental error and sampling error; known distribution of the variable of interest; type I error rate (alpha); and experimental design structure.

```sas
options nocenter mprint;
data a; input trt y;
do rep=1 to 4 by 1;
do samples=1 to 2 by 1;
output;
end;
end;
datalines;
1 95
```
SAS CODE FOR PREDICTING THE POWER OF AN EXPERIMENT EMBEDDED WITHIN A MACRO TO ALLOW COMPUTATION ACROSS A WIDE RANGE OF CONDITIONS

SAS code from the previous example is embedded in a macro that allows power to be estimated for a range of experimental designs with 4 to 8 experimental units per treatment (rep, repl, repmax) and 2 to 4 observational units per experimental unit (obs, obsv, obsmax). Values of the variables – rep, repl, repmax, obs, obsv, obsmax – can be changed to any values desired for different research projects.

options nocenter mprint;
%macro one(obsmax,repmax);
data a;
%do obsv=2 %to &obsmax;
group1=&obsv;
%do repl=2 %to &repmax;
group2=&repl;
do obs=2 to &obsv by 1;
do rep=4 to &repl by 1;
do trt=0 to 1 by 1;
output;
end;
end;
end;
%end;
%end;
SAS CODE FOR PREDICTING THE POWER OF A RANDOMIZED COMPLETE BLOCK EXPERIMENT CONDUCTED AT MULTIPLE LOCATIONS

SAS code to estimate power of a hypothesis test with the following parameters: treatment means = 9 and 10, variance components = 0.02 and 0.2 (treatment \times location interaction and experimental error, respectively), \( r = 4 \) to 20 replicates, \( l = 2 \) to 6 locations, and the assumption of normally distributed errors (adapted from Gbur et al., 2012).

options nocenter mprint;
%mend one;
%one(4,8); /* <--- change values here */
run;
proc sort; by group1 group2;
data b; set a; by group1 group2;
if trt=0 then y=95;
if trt=1 then y=100;
run;
proc glimmix; class trt rep; by group1 group2;
model y = trt;
random rep(trt);
parms (5)(10) / hold=1,2;
ods output tests3=power_terms;
data power;
set power_terms;
alpha=0.05;
cparm=numdf*Fvalue;
F_critical=finv(1-alpha, numdf, dendf, 0);
power=1-probf(F_critical, numdf, dendf, ncparm);
proc print;
run;
end;
end;
end;
%end;
%end;
%mend two;
%two(6,20); /* <--- change here */
run;
proc sort; by group1 group2;
data b; set a; by group1 group2;
if trt=0 then y=9;
if trt=1 then y=10;
run;
proc glimmix; class location trt rep; by group1 group2;
model y = trt;
random trt*location;
parms (0.02)(0.2) / hold=1,2;
ods output tests3=power_terms;
data power;
set power_terms;
alpha=0.05;
ncparm=numdf*Fvalue;
F_critical=finv(1-alpha, numdf, dendf, 0);
power=1-probf(F_critical, numdf, dendf, ncparm);
proc print;
run;
Can Analysis of Variance Be More Significant?

Marla S. McIntosh*

ABSTRACT

Recent widespread criticism of the lack of statistical rigor in science journals has focused attention on the need to improve the standards for statistical design and analysis in research. This study examined the role of analysis of variance (ANOVA) in the context of current concerns regarding the validity and appropriateness of statistics in scientific publications. One objective was to suggest how ANOVA tables can be constructed to enhance the transparency and scientific integrity of scientific journals and better assist the interpretation of data. The broader goal of this study was to generate new discussion, debate, and ideas regarding ANOVA. The history and current status of ANOVA as the context for assessing the practical and statistical relevance of ANOVA tables for students, authors, reviewers, editors, and readers of scientific journals is discussed. Each component of an ANOVA table (sources of variation, degrees of freedom, sums of squares, mean squares, $F$ values, and $P$ values) is critiqued for its information and value. Using a criterion of including the components that provide essential information on key details of the experimental design and validating the appropriateness of the analysis, guidelines are provided for constructing an ANOVA table that is SIMPLE—Simple, Informative, Meaningful, Powerful, Logical, and Effective. A prototype SIMPLE ANOVA table is presented to encourage further consideration and debate regarding best practices for ANOVA tables.

Abbreviations: MS, mean squares; RCBD, randomized complete block design; SOV, source of variation; SS, sums of squares.

M.S. McIntosh, 6117 Plant Science Building, Univ. of Maryland, College Park, MD 20742. Received 2 Apr. 2014. Accepted 19 Nov. 2014. *Corresponding author (mmcintos@umd.edu).
SAS CODE FOR THE DEFAULT MIXED PROCEDURE ANALYSIS OF A RANDOMIZED COMPLETE BLOCK EXPERIMENT COMBINED OVER LOCATIONS

The SAS code using PROC MIXED to test significance of fixed effects and estimate variance and covariance of random effects for a two-factor randomized complete block experiment combined over locations. The blocks (BLK) are random effects and the locations (LOC) and factors (A, B) are fixed effects. Using the default mixed procedure, the output does not include the degrees of freedom or F-tests for effects that include random factors.

Proc mixed;
Class loc blk A B;
Model dependent = loc A loc*A B loc*B A*B loc*A*B;
Random blk(loc);
run;

SAS CODE FOR THE MIXED PROCEDURE ANALYSIS AND TYPE3 METHOD OPTION OF A RANDOMIZED COMPLETE BLOCK EXPERIMENT COMBINED OVER LOCATIONS USING PROC MIXED

The SAS code using PROC MIXED to produce an ANOVA table with all terms in the model for a two-factor randomized complete block experiment combined over locations. The blocks (BLK) are random effects and the locations (LOC) and factors (A, B) are fixed effects. The Type3 option generates a comprehensive ANOVA table with sources of variation, degrees of freedom, expected mean squares, and F-tests of random as well as fixed effects in the model.

Proc mixed method=type3;
Class loc blk A B;
Model dependent = loc A loc*A B loc*B A*B loc*A*B;
Random blk(loc);
run;
“Is, or is not, the two great ends of Fate:” Errors in Agronomic Research

Kimberly Garland Campbell,* Yvonne M. Thompson, Stephen O. Guy, Marla McIntosh, and Barry Glaz

ABSTRACT

Agronomic research results include Type 1 (α) and Type 2 (β) errors. Results are often reported using α ≤ 0.05 while β is ignored. Our objective was to discuss whether a false positive was more serious than a false negative in agronomic research. For comparison, current statistical methods used in Agronomy Journal were tabulated. Most papers used null hypothesis tests with α ≤ 0.05, reporting results based on the LSD among all treatment pairs. Current practices do not account for the relative costs of false positive vs. false negative errors. A case study from the Washington State Wheat Extension trials was analyzed using mixed models with specific contrasts. While the overall effect for cultivar was significant, the β error rate for the contrast was 40% and additional replications were needed to increase the power of this contrast. A second case study analyzed trials evaluating wheat (Triticum aestivum L.) resistance to Fusarium crown rot. Optimal α and β error rates were estimated for two to eight replications with the Type1/Type2 error cost ratio set at 1:1 and 1:5. An average error rate (α + β) ≤ 0.05 could be achieved with four replications when a reduction in the β error was critical and α errors could be corrected in future experiments. Effective experimental design requires estimation of the acceptable magnitude and cost ratio of false positive and false negative errors and critical effect sizes. To be truly informative, reports of results should include this information plus observed effect sizes and variances.

K.G. Campbell, USDA–ARS, Wheat Genetics, Quality, Physiology, and Disease Research Unit, and Dep. of Crop and Soil Sciences, Washington State Univ., Pullman, WA 99164; Y.M. Thompson and S.O. Guy, Dep. of Crop and Soil Sciences, Washington State Univ., Pullman, WA 99164; M. McIntosh, Dep. of Plant Science and Landscape Architecture, Univ. of Maryland, College Park, MD 20742; and B. Glaz, USDA–ARS, Sugarcane Production Research Unit, Canal Point, FL 33438 (retired). Title is a quote from “Upon Nothing” by John Wilmot Rochester, 2nd Earl of Rochester Received 21 Mar. 2014. Accepted 26 Nov. 2014. *Corresponding author (kgcamp@wsu.edu).

Abbreviations: CV, coefficient of variation; H₀, null hypothesis; H₁, alternative hypothesis; LSD, least significant difference.
The SAS code for calculating the power of an experiment when the experimental design and critical F value can be estimated. Items in **bold** are provided by the user. The area of the noncentral F distribution that exceeded the critical F value was calculated using the PROBF function. In this code, the alpha level, the numerator and denominator degrees of freedom, and the F value (Alpha, Numdf, Dendf, and Fvalue, respectively) are provided from similar experiments. Alternatively, the coding can include a new data set created from the output of an example mixed model analysis as described below for Numdf, Dendf, and Fvalue.

```sas
Data POWER;
Alpha=0.10;
Numdf=124;
Dendf=1157;
Fvalue=5.13;
Noncent=Numdf*Fvalue;
Fcrit=Finv(1-Alpha, Numdf, Dendf, 0);
Power=1-Probf(Fcrit, Numdf, Dendf, Noncent);
Proc Print Data=POWER;
```

The SAS code for mixed models analysis of two environments with 58 entries in each and three replications and 12 incomplete blocks (**ENV, CULTIVAR, REP, IBLOCK**), with data provided by the user in a data set named **AAA**. The contrast requests a test of the difference between Cultivar 3 and the check Cultivar 11. The output statement creates two data sets, **TEST3** and **CONTRAST**, which hold the Type 3 tests of fixed effects for the main effects and interactions and the contrast statements, respectively. These statistics are used in the data set **POWER** to estimate the noncentrality parameter, the critical F value for a given alpha level, and the power of the experiment to test the various effects. The power calculations are similar to those above, except that the data for degrees of freedom and Fvalue are read from the data sets **TEST3** and **CONTRAST**.

```sas
Proc mixed data=AAA method=reml;
  class ENV CULTIVAR REP IBLOCK;
  model yield=CULTIVAR ENV CULTIVAR*ENV/solution ddfm=satterth;
  Contrast ‘C3 vs. Check C11’ CULTIVAR
    0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```
random REP(ENV) ibloc(REP*ENV);
lsmeans CULTIVAR ENV CULTIVAR*ENV;
ods output tests3=TEST3 contrasts=CONTRAST; run;
data POWER; set TEST3 CONTRAST;
alpha=0.10;
noncent=numdf*fvalue;
fcrit=finv(1-alpha, numdf, dendf, 0);
power=1-probf(fcrit, numdf, dendf, noncent);
proc print data=POWER;

SAS CODE TO CALCULATE THE POWER OF EXPERIMENTS OVER A RANGE OF STANDARD DEVIATIONS AND OBSERVATION NUMBERS

Effect are estimated from cultivar mean disease scores provided in the data set BBB. The analysis is coded to evaluate the influence of alpha error levels from 0.05 to 0.15 and experimental standard deviation from 1.5 to 3.5. These values were chosen based on analysis of previous similar trials. The experiment size is varied from 100 to 800 observations. The missing data point for power requests that it be calculated by the program.
Proc glmpower data=BBB;
class CULTIVAR;
model MEANSCORE=CULTIVAR;
power
alpha=0.05 0.10 0.15
stddev=1.5 2.5 3.5
ntotal=100 150 200 250 300 350 400 500 600 700 800
power=. ;
plot x=n ;
run;
CALCULATING THE OPTIMAL ALPHA LEVEL FOR AN EXPERIMENT USING THE R SCRIPT:

ANOVA OPTIMAL ALPHA R CODE VERSION 1.1

The R script ANOVA optimal.α.R code version 1.1 downloaded from http://www.plosone.org/annotation/listThread.action?root=67535 (Mudge et al., 2012a) was run several times with varying scenarios, holding numerator degrees of freedom (u) at 91 but increasing the denominator degrees of freedom (v) to simulate larger numbers of replications, varying the effect size (Cohen’s $F^2$) from 0.1 to 0.3, and testing the T1/T2 ratio at 0.2 and at 1.

```r
> optab.anova(u=91,v=91,f2=0.1,T1T2cratio=.2,HaHopratio=1) #2 reps f2=.1.
> optab.anova(u=91,v=182,f2=0.1,T1T2cratio=.2,HaHopratio=1) #3 reps f2=.1.
> optab.anova(u=91,v=273,f2=0.1,T1T2cratio=.2,HaHopratio=1) #4 reps f2=.1.
> optab.anova(u=91,v=364,f2=0.1,T1T2cratio=.2,HaHopratio=1) #5 reps f2=.1.
> optab.anova(u=91,v=455,f2=0.1,T1T2cratio=.2,HaHopratio=1) #6 reps f2=.1.
> optab.anova(u=91,v=546,f2=0.1,T1T2cratio=.2,HaHopratio=1) #7 reps f2=.1.
> optab.anova(u=91,v=91,f2=0.2,T1T2cratio=.2,HaHopratio=1) #2 reps f2=.2.
> optab.anova(u=91,v=182,f2=0.2,T1T2cratio=.2,HaHopratio=1) #3 reps f2=.2.
> optab.anova(u=91,v=273,f2=0.2,T1T2cratio=.2,HaHopratio=1) #4 reps f2=.2.
> optab.anova(u=91,v=364,f2=0.2,T1T2cratio=.2,HaHopratio=1) #5 reps f2=.2.
> optab.anova(u=91,v=455,f2=0.2,T1T2cratio=.2,HaHopratio=1) #6 reps f2=.2.
> optab.anova(u=91,v=546,f2=0.2,T1T2cratio=.2,HaHopratio=1) #7 reps f2=.2.
> optab.anova(u=91,v=91,f2=0.2,T1T2cratio=1,HaHopratio=1) #2 reps f2=.3.
> optab.anova(u=91,v=182,f2=0.2,T1T2cratio=1,HaHopratio=1) #3 reps f2=.3.
> optab.anova(u=91,v=273,f2=0.2,T1T2cratio=1,HaHopratio=1) #4 reps f2=.3.
> optab.anova(u=91,v=364,f2=0.2,T1T2cratio=1,HaHopratio=1) #5 reps f2=.3.
> optab.anova(u=91,v=455,f2=0.2,T1T2cratio=1,HaHopratio=1) #6 reps f2=.3.
> optab.anova(u=91,v=546,f2=0.2,T1T2cratio=1,HaHopratio=1) #7 reps f2=.3.
> optab.anova(u=91,v=91,f2=0.2,T1T2cratio=1,HaHopratio=1) #2 reps f2=.2.
> optab.anova(u=91,v=182,f2=0.2,T1T2cratio=1,HaHopratio=1) #3 reps f2=.2.
> optab.anova(u=91,v=273,f2=0.2,T1T2cratio=1,HaHopratio=1) #4 reps f2=.2.
> optab.anova(u=91,v=364,f2=0.2,T1T2cratio=1,HaHopratio=1) #5 reps f2=.2.
> optab.anova(u=91,v=455,f2=0.2,T1T2cratio=1,HaHopratio=1) #6 reps f2=.2.
```
> optab.anova(u=91, v=546, f2=0.2, T1T2cratio=1, HaHopratio=1) #7 reps f2=.2.
Evaluation and Interpretation of Interactions

Jose Crossa, Mateo Vargas, C. Mariano Cossani, Gregorio Alvarado, Juan Burgueño, Ky L. Mathews, and Matthew P. Reynolds

Abstract

Understanding the factors that define a given interaction is important in agricultural, agronomic, and plant breeding research, where agronomic treatments and/or genotypes are evaluated in several environmental conditions and where interactions usually complicate a researcher’s decisions. The existence of interaction in agricultural experiments precedes the development of the analysis of variance and in general a multiplicative operator for assessing interactions gives better fit to the differential responses than the sum formula. In this article we give examples of how interactions in common agricultural experiments can be examined and studied to make use of the rich information available on the interaction term of the model. Examples with different levels of interaction complexity are used to illustrate how to analyze and interpret interactions and how interaction components can be partitioned into comparisons with sensible biological interpretations that will offer researchers a greater understanding of how to exploit interaction information beyond the standard interaction statistical tests performed in the usual analysis of variance. Some simple SAS codes for performing standard interaction contrasts and defining interaction covariables are provided.


Abbreviations: NID: Normally and Independently Distributed; GE: genotype × environment interaction; FR: Factorial regression; PLS: Partial Least Squares
SAS Code for calculating 1 df sum of square contrasts for main effects and their interactions in Factorial Experiments

In Factorial experiments the structure of the treatment combinations suggests several sets of orthogonal contrasts for the main effects and for the interactions. For example assume you have an agronomical experiment with two cultivars (C1 and C2) planted at two different plant densities (D1 and D2) and evaluated under three levels of water availability, namely: severe water stress (SS), intermediate water stress (IS), and well-watered conditions (WW) with three replicates for grain yield measured in Mg ha\(^{-1}\). Factor cultivar-density (CultDen) suggests three orthogonal comparisons (contrasts 1-3) of single degree of freedom each. For factor type of stress, two orthogonal one degree of freedom contrasts are suggested (contrasts 4 and 5). These three and two orthogonal contrasts for the main effects suggest six interaction contrasts, contrasts 6-11.

The SAS code using **PROC GLM** and the **CONTRAST** statements of the main effects and interactions are given in the next code. Other comparisons can be made and/or individual contrasts can be combined in more than one degree of freedom comparison.

**PROC GLM:**

```sas
PROC GLM;
  Class CultDen Stress Rep;
  Model Yield = CultDen Stress CultDen*Stress;
  * Contrasts for Main Effect CultDen
    C1D1 C1D2 C2D1 C2D2;
    Contrast "C1D1 vs. C1D2" CultDen 1 -1 0 0;
    Contrast "C2D1 vs. C2D2" CultDen 0 0 1 -1;
    Contrast "C1 vs. C2" CultDen 1 1 -1 -1;
  * Contrasts for Main Effect Stress
    SS IS WW;
    Contrast "SS vs. IS" Stress 1 -1 0;
    Contrast "Stress vs. WW" Stress 1 1 -2;
  * Contrasts for Interactions CultDen*Stress
    M11 M12 M13 M21 M22 M23 M31 M32 M33 M41 M42 M43;
    Contrast "C1*Stress" CultDen*Stress 1 -1 0 -1 1 0 0 0 0 0 0 0;
    Contrast "C2*Stress" CultDen*Stress 0 0 0 0 0 0 1 -1 0 -1 1 0;
    Contrast "C1&C2*Stress" CultDen*Stress 1 -1 0 1 -1 0 -1 1 1 0 -1 1;
    Contrast "C1*TypeStress" CultDen*Stress 1 1 -2 -1 1 2 0 0 0 0 0 0;
    Contrast "C2*TypeStress" CultDen*Stress 0 0 0 0 0 0 1 1 -2 -1 -1 2;
    Contrast "C1&C2*TypeStress" CultDen*Stress 1 1 -2 1 1 -2 -1 2 -1 -1 2;
```
SAS Code for calculating Interaction Contrasts using covariables and the Factorial Regression (FR) Model

It might be useful for the researcher to have comparisons that combines other contrasts in order to have a more specific idea of the interaction response trends. For instance, one might be interested in testing the interaction of cultivars 1 and 2 in both planting densities with the type of stress. This can be done by combining contrasts 9-11 in only one comparison. These comparisons are defined as a \texttt{Covar\_1} and the result gives a comparison with 3 degrees of freedom.

**** Defining the covariables for the Stress factor ****;

* Average (SS + IS) vs. WW;
  if Stress = 'WW' then Covar\_1 = -2;
  else Covar\_1 = 1;

* SS vs. IS;
  if Stress = 'SS' then Covar\_2 = 1;
  else if Stress = 'IS' then Covar\_2 = -1;
  else Covar\_2 = 0;

**** Using the covariables ****;

PROC GLM;
  Class Stress CultDen;
  Model Yield = Stress CultDen CultDen*Covar\_1;
PROC GLM;
  Class Stress CultDen;
  Model Yield = Stress CultDen CultDen*Covar\_2;

SAS Code for the Factorial Regression Model in a Genotype × Environment Interaction (GEI) Framework using Covariables

Assume a field experiment including 40 wheat (\textit{Triticum aestivum} \textit{L.}) genotypes evaluated in two consecutive years under four environmental conditions: drought (D), irrigated (I) and two heat (H) stress conditions (intermediate heat, IH, and severe heat, SH).

The use of the covariables in the FR is indeed useful for a rapid and efficient partition of the interactions into contrasts that can explain sizeable portions of its complexity. Furthermore, when several covariables are defined, the \texttt{PROC GLMSELECT}, which uses a stepwise variable selection procedure, is useful for determining the most important interaction comparisons.

** Defining environments as combinations of year and environmental condition **;
if Env_0 = "Intermediate_Heat" and Year = 2011 then Env = "IH_1";
if Env_0 = "Intermediate_Heat" and Year = 2012 then Env = "IH_2";
if Env_0 = "Severe_Heat" and Year = 2011 then Env = "SH_1";
if Env_0 = "Severe_Heat" and Year = 2012 then Env = "SH_2";
if Env_0 = "Irrigation" and Year = 2011 then Env = "I_1";
if Env_0 = "Irrigation" and Year = 2012 then Env = "I_2";
if Env_0 = "Drought" and Year = 2011 then Env = "D_1";
if Env_0 = "Drought" and Year = 2012 then Env = "D_2";

** Defining the covariables for the contrasts **;

** Contrast Heat (H) vs. Irrigated (I): covar_1 **;
  if Env in ( "I_1" "I_2") then Covar_1 = 2;
  else if Env in ( "D_1" "D_2") then Covar_1 = 0;
  else Covar_1 = -1;

** Contrast Drought (D) vs. Irrigated (I): covar_2 **;
  if Env in ( "I_1" "I_2") then Covar_2 = 1;
  else if Env in ( "D_1" "D_2") then Covar_2 = -1;
  else Covar_2 = 0;

** Contrast Drought (D) vs. Heat (H): covar_3 **;
  if Env in ( "I_1" "I_2") then Covar_3 = 0;
  else if Env in ( "D_1" "D_2") then Covar_3 = 2;
  else Covar_3 = -1;

** Contrast [Drought (D) + Heat (H)] vs. Irrigated (I): covar_4 **;
  if Env in ( "I_1" "I_2") then Covar_4 = 3;
  else Covar_4 = -1;

Title1"ANOVA complete two-way model including GEI";
PROC GLM;
  Class Env Rep Block Gen;
  Model Yield = Env Rep(Env) Block(Rep Env) Gen Env*Gen;

Title1 "Interpreting GEI using Covariable 1: Contrast Irrigated vs. Heat";
PROC GLM;
  Class Env Rep Block Gen;
  Model Yield = Env Rep(Env) Block(Rep Env) Gen Gen*Covar_1;

Title1 "Interpreting GEI using Covariable 2: Contrast Irrigated vs. Drought";
PROC GLM;
  Class Env Rep Block Gen;
  Model Yield = Env Rep(Env) Block(Rep Env) Gen Gen*Covar_2;

Title1 " Interpreting GEI using Covariable 3: Contrast Drought vs. Heat";
PROC GLM;
  Class Env Rep Block Gen;
  Model Yield = Env Rep(Env) Block(Rep Env) Gen Gen*Covar_3;

Title1 " Interpreting GEI using Covariable 4: Irrigated vs. Drought & Heat";
PROC GLM;
Class Env Rep Block Gen;
Model Yield = Env Rep(Env) Block(Rep Env) Gen Gen*Covar_4;

*** Interpreting GEI with Factorial Regression using GLMSELECT: all covariables and stepwise variable selection ***;

PROC GLMSELECT;
Class Env Rep Block Gen;
Model Yield = Env Rep(Env) Block(Rep Env) Gen
    Gen*Covar_1 Gen*Covar_2 Gen*Covar_3 Gen*Covar_4
    / include = 4 selection = stepwise (select = SL SLE = 0.10 SLS = 0.10 choose = AIC);
ODS Listing Exclude ParameterEstimates;

*** Interpreting GEI final Factorial Regression model using GLM and only the SELECTED covariables ***;

PROC GLM;
Class Env Rep Block Gen;
Model Yield = Env Rep(Env) Block(Rep Env) Gen
    Gen*Covar_4 Gen*Covar_3;

SAS codes for Interpreting Genotype × Environment Interaction using Partial Least Squares (PLS) Analysis

When environment or genotype covariables show high co-linearity, the regression coefficients are estimated very imprecisely. Noise in the response variable also complicates interpretation of the FR parameters; least squares estimation of the parameters in the FR models is not unique when the number of covariables is larger than the number of observations. An alternative estimation method is the partial least squares (PLS) regression, which overcomes some of these problems.

*** This code was initially prepared for 29 genotypes (2 to 30) evaluated in 8 environments (E1 to E8), and using 9 environmental covariables ***;
*** Complete code and data can be obtained by requesting to the first author ***;

*** Reading phenotypic data ***;
Data Genotypes;
    Infile'c:\yield data for pls.csv' dlm=';' firstobs=3;
    Input env gen rep block yld;
Datalines;

*** Reading environmental data ***;
Data Environmental;
    Infile'c:\environmental data for pls.csv' dlm=';' firstobs=3;
    Input env tmax1 tmax2 tmax3 tmin1 tmin2 tmin3 rh1 rh2 rh3;
Datalines;
*** computing lsmeans by environment using mixed models ***;
Proc MIXED data=crops method=reml ; by env; class rep block gen ;
   model yld= gen; random rep block(rep); lsmeans gen;
   ods output lsmeans = ls1; run;

*** recovering the lsmeans data ***;
Data medias; set ls1; yld=estimate; keep env gen yld;
Proc Sort data = medias; by env gen;
Proc Means data = medias noprint; by env gen; var yld; output out=envgeno mean=yld;

*** Generating the residual matrix using the Additive Main effect and Multiplicative Interaction model ***;
Proc GLM data=envgeno noprint; class env gen ; model yld = env gen;
   output out=outrer r=resid; run;
Data residual; set outrer; keep env gen resid;
Proc Transpose data=residual out=residual; by env; id gen;
Data residual; set residual;
   keep _2 - _30; * modify depending on the number of genotypes;

*** Standardizing data for use in the Partial Least Square algorithm ***;
Proc Standard data=residual mean=0 std=1 out=residua2;
Proc Standard data=envgeno mean=0 std=1 out=environ3;

*** Merging phenotypic and environmental standardized data ***;
Data plsdatalast; merge residua2 environ3; by env;

*** Using Proc PLS for calculating the Scores ***;
Proc PLS data=plsdatalast method=pls (algorithm=svd) outmodel=est1
   cv=one cvtest ( stat=press pval=1.0 ) lv=3;
   Model _2- _30 = tmax1 tmax2 tmax3 tmin1 tmin2 tmin3 rh1 rh2 rh3;
   * modify depending on number and name of genotypes and environmental variables;
   output out = scores xscore=xscore yscore=yscore;

*** Recovering the relevant information from the PLS results ***;
Data scores; set scores; keep xscore1-xscore2;
Data xloading; set est1; if _type_ = 'wb';
   keep env tmax1 tmax2 tmax3 tmin1 tmin2 tmin3 rh1 rh2 rh3 _lv_;
   * modify only the name of the environmental variables;
Proc Transpose data=xloading out=xloadin2; id _lv_;
Data yloading; set est1; if _type_ = 'pq';
   keep _2 - _30 _lv_; * modify only the number of genotypes;
Proc Transpose data=yloading out=yloadin2; id _lv_;

*** Managing the scores using IML preparing for graphing the biplot ***;
Proc IML;
   use scores; read all into mscores; factor1=max(abs(mscores));
   mscores2=(1/factor1)*mscores; namecol1={'dim1','dim2'};
   create scores3 from mscores2[colname=namecol1]; append from mscores2;
   close scores3;
   use xloadin2; read all into xload;xload=xload[,1:2]; factor2=max(abs(xload));
   xload2=(1/factor2)*xload; namecol2={'dim1','dim2'};
   create xload3 from xload2[colname=namecol2]; append from xload2;close xload3;
use yloadin2; read all into yload; yload=yload[,1:2];factor3=max(abs(yload));
yload2=(1/factor3)*yload; namecol3={'dim1','dim2'};
create yload3 from yload2[colname=namecol3]; append from yload2;close yload3;
quit;

*** Creating the name and type variables for graphing the biplot ***;

Data envname; * modify depending on the names of environments;
   Input name $ @@; type='env'; datalines;
      E1 E2 E3 E4 E5 E6 E7 E8

Data genname; * modify depending on the names of genotypes;
   Input name $ @@; type='gen'; datalines;
      2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Data varname; * modify depending on the names of environmental variables;
   Input name $ @@; type='var'; datalines;
      tmxt tmxsi tmxb tmnt tmnsi tmnb rht rhsi rhb

Data scores4; merge envname scores3;
Data xload4; merge varname xload3;
Data yload4; merge genname yload3;

*** Joining all the information for the biplot ***;

Data biplot; set scores4 xload4 yload4; run;

title1'all adjusted data'; proc print data = biplot; run;
quit;

*** NOTE: SAS codes for graphing the biplot are not included, but the complete code and data examples can
be obtained by requesting to the first author;
Analysis and Interpretation of Interactions in Agricultural Research

Mateo Vargas, Barry Glaz, Gregorio Alvarado, Julian Pietragalla, Alexey Morgounov, Yuriy Zelenskiy, and Jose Crossa*

ABSTRACT

When reporting on well-conducted research, a characteristic of a complete and proper manuscript is one that includes analyses and interpretations of all interactions. Our purpose is to show how to analyze and interpret interactions in agronomy and breeding research by means of three data sets comprising random and fixed effects. Experiment 1 tested wheat (Triticum aestivum L.) at two N and four P fertilizer rates in two soil types. For this data set, we used a fixed-effect linear model with the highest order (three-way) interaction considered first and then worked down through the lower order interactions and main effects to illustrate the importance of interactions in data analysis. Experiment 2 evaluated maize (Zea mays L.) hybrids under four rates of N for 3 yr. For this data set, we used a linear mixed model and partitioned the four N rates into orthogonal polynomials. Experiment 3 evaluated genotypes in six environments where the objective was to show how to study genotype × environment interactions. Researchers must analyze all interactions, determine if they are due to changes in rank (crossover) or only to changes in scale, and then judge whether reporting on significant main effects or interactions would best explain the biological responses in their experiments. In an experiment with more than one factor, complete and correct analysis of interactions is essential for reporting and interpreting the research properly.

M. Vargas, G. Alvarado, and J. Crossa, Biometrics and Statistics Unit, CIMMYT, Apdo. Postal 6-641, 06600 México DF, Mexico; M. Vargas, Univ. Autónoma Chapingo, Chapingo, Mexico; J. Glaz (retired), USDA–ARS Sugarcane Field Station, 12990 U.S. Highway 441, Canal Point, FL 33438; J. Pietragalla, Global Wheat Breeding Program, CIMMYT, Apdo. Postal 6-641, 06600 México, DF, Mexico; and A. Morgounov, CIMMYT, P.O. Box 39, Emek, 06511 Ankara, Turkey, and CIMMYT, P.O. Box 1446, Astana, 010000, Kazakhstan. Received 23 Aug. 2013.

*Corresponding author (j.crossa@cgiar.org).

Abbreviations: COI, crossover interaction; GE, genotype × environment interaction; GY, grain yield; HWCI, half width confidence interval; NCOI, non-crossover interaction; TE, treatment × environment interaction.
SAS Code for calculating the coefficients of orthogonal polynomials for unequally spaced levels

The coefficients for orthogonal polynomials when the levels of factors are equally spaced are easy to obtain directly from tables in books on statistics. However, when the factor levels are unevenly spaced it is helpful to use the following SAS code to compute those coefficients. The SAS code shown is also useful for equally spaced factor levels.

It is necessary to have installed the Interactive Matrix Language (IML) procedure for calculating the orthogonal polynomial coefficients. In the $P_{levels}$ statement, you express how many and which levels of the variable you wish to calculate. For instance if you have four unevenly spaced P rates: 0, 50, 150, and 250 kg ha$^{-1}$. The $Orpol$ function is used for obtaining the required coefficients; since there are three degrees of freedom, we can calculate three polynomial coefficients—linear, quadratic, and cubic—which are assigned to macro variables $PosCoefGrade&K$ and $NegCoefGrade&K$, which are later used in the data step in the three-way analysis of variance and contrasts.

In the main effects contrasts, we need only positive values of the coefficients, but in the two- and three-way contrasts, a combination of positive and negative coefficients are required, depending on the levels of each of the factors involved in those interactions. These positive and negative coefficients are assigned to the $PosCoefGrade$ and $NegCoefGrade$ macro variables, respectively, using the Call Symput function. The order of the polynomial contrasts is obtained using the %do – %end cycle. Note that the appropriate number and grade of the coefficients are automatically determined in the $ncof = ncol(coef) - 1$ statement. This program can be easily modified for different numbers and/or levels of factors.

SAS macro program: Items in bold are provided by the user according to his/her own data

**** Reading data ****;
Data Raw;
   Infile “C:\Yield & Gluten Data 2007.csv” dlm=”,“ firstobs=3;
   Informat Soil$ 10 .;
   Input Soil N P Rep Yield Gluten;
Datalines;
Run;
**** Here begins the macro code for calculating the orthogonal polynomial contrasts in an automatic way ****;

%Macro Coefficients;
   Proc IML;
   Plevels={0, 50, 150, 250};
   coeff=Orpol(Plevels,3);
   ncof=ncol(coeff) - 1;
   call symputx(“ncof”, ncof);
   %do K=1 %to &ncof;
      CoefGrade&K=t(Coeff[,&K+1]);
      PosCoefGrade&K=rowcat(char(CoefGrade&K, 15,10));
      NegCoefGrade&K=rowcat(char(-(CoefGrade&K), 15,10));
      call symputx(“PosCoefGrade&K”, PosCoefGrade&K);
   %end
Page 23 of 55
call symputx(“NegCoefGrade&K”, NegCoefGrade&K);

**%end;**

**** Here begins the macro code for the Three-way ANOVA, decomposing df for P into three contrasts: linear, quadratic, and cubic****;

Proc GLM data=Raw;
   class Soil N P Rep;
   model Yield = Soil N P Soil*N Soil*P N*P Soil*N*P / ss3;
   **** P main effects contrasts ****;
      contrast “Linear P” P &PosCoefGrade1;
      contrast “Quadratic P” P &PosCoefGrade2;
      contrast “Cubic P” P &PosCoefGrade3;
   **** Soil x P two way interaction contrasts ****;
      contrast “Linear Soil*P” Soil*P &PosCoefGrade1 &NegCoefGrade1;
      contrast “Quadratic Soil*P” Soil*P &PosCoefGrade2 &NegCoefGrade2;
      contrast “Cubic Soil*P” Soil*P &PosCoefGrade3 &NegCoefGrade3;
   **** N x P two way interaction contrasts ****;
      contrast “Linear N*P” N*P &PosCoefGrade1 &NegCoefGrade1;
      contrast “Quadratic N*P” N*P &PosCoefGrade2 &NegCoefGrade2;
      contrast “Cubic N*P” N*P &PosCoefGrade3 &NegCoefGrade3;
   **** Soil x N x P three way interaction contrasts ****;
      contrast “Linear Soil*N*P” Soil*N*P &NegCoefGrade1 &PosCoefGrade1
                      &PosCoefGrade1 &NegCoefGrade1;
      contrast “Quadratic Soil*N*P” Soil*N*P &NegCoefGrade2 &PosCoefGrade2
                      &PosCoefGrade2 &NegCoefGrade2;
      contrast “Cubic Soil*N*P” Soil*N*P &NegCoefGrade3 &PosCoefGrade3
                      &PosCoefGrade3 &NegCoefGrade3;
%Mend;
%Coefficients;
Run;

SAS code for Computing and Graphing Least Significant Difference for Interactions:

Because the \( LSD = \left( t_{\alpha/2}\right)^2 / 2 \sqrt{2\text{MSE}/V} \) and the half width of the confidence interval \( \text{HWCI} = \left( t_{\alpha/2}\right)/\sqrt{\text{MSE}/V} \), where \( t_{\alpha/2} \) is Student’s \( t \) quantile with error degrees of freedom and significance level \( \alpha/2 \),
MSE is the mean square error from the analysis of variance, and \( V \) is the number of replications multiplied by the number of factor levels not involved in that LSD value (or the number of observations for those means), it is easy to see that \( \text{LSD} = \sqrt{(2) \times \text{HWCI}} \).

For the main effects, SAS Proc GLM as well others procedures, directly calculates and outputs the LSD value. However, SAS does not directly calculate the LSD values for the interactions. Using the above relationship between the LSD and HWCI, it is possible to obtain the confidence interval for the least squares adjusted means (lsmeans).

The above relationship is valid only for balanced data. For unbalanced data, the LSD expression becomes \( \text{LSD} = (t_{\alpha/2}) \sqrt{[\text{MSE}(1/v_i + 1/v_j)]} \), where \( v_i \) and \( v_j \) are the number of observations involved in the corresponding levels \( i \) and \( j \) of the factor being analyzed. If the cell sizes are unequal, SAS uses the harmonic mean of the cell sizes to compute the critical ranges in the mean statement and for the confidence limits for the individual lsmeans. This approach is reasonable if the cell sizes are not too different, but it can lead to liberal tests if the cell sizes are highly disparate. Because it is difficult to automatically compute the harmonic mean in the SAS data step, we have included a correction factor given by the ratio of the number of real observations used (\( OU \)) in the analysis of variance divided by the total number of observations read (\( OR \)). This ratio is the coefficient \( CF = OU/OR \). When the data is balanced, \( CF = 1 \).

The output delivery system (\( ODS \), \( ODS \) Output \( \text{LSMeanCL}=\text{LsmeansCI} \) statement is useful for creating a virtual file with only the information that will be needed later. The \( ODS \) listing exclude is for suppressing the additional information that SAS writes by default, which isn’t necessary. For generating the different variables containing information for the regression lines associated with each combination of the soil × P interaction (Fig. 3), we created the new variables \( Y_{\text{black}} \) and \( Y_{\text{chest}} \) from the yield values. Similarly, cycles \( \text{if – then do – end} \) are used for obtaining the information needed in the upper and lower LSD bars. The codes can be adapted if four regression lines need to be depicted with their corresponding LSD bars.

If we are interested in the ANOVA for all the main effects and two- and three-way interactions; thus all those terms should be included in the model statement when computing the correct LSD and/or confidence interval. In the \( \text{LSMeans} \) statement, we have included only the interactions that are of interest for graphing the LSD values, thus simplifying the output.

**** Data Reading from an external CSV File ****;

Data Raw;
    Infile “C:\Yield & Gluten Data.csv” dlm=”,”, firstobs=3;
    Informat Soil$ 10.;
    Input Soil N P Rep Yield Gluten;
Datalines;

**** Three-way ANOVA, including all two-way and three-way interactions; ****;
Proc GLM Data=Raw;
    Class Soil N P Rep;
    Model Yield=Soil Rep(Soil) N P Soil*N Soil*P N*P Soil*N*P / ss3;
    LSMeans Soil*P/Adjust=T Lines CL;
    ODS output LSMeanCL=LsmeansCI NObs=Nobserv;
    ODS listing exclude CLMeansInfo Diff CLMeans LSMeans LSMeanCL
        LSMeanDiffCL;
**** Calculating the correction factor for an unbalanced data set ****;
Data Nobs;
    Set Nobserv;
    CF0 = NobsUsed/NobsRead;
    Call symput("CF", CF0);
**** LSD for interactions through HWCI, generating a macro variable using the call symput function ****;
Data LSD;
    Set LSMeansCI;
    HWCI=LSmean – LowerCL;
    LSD=Sqrt(2*&CF)*(HWCI);
    P2=P + 0.0;
    Drop P;
Proc Sort Data=LSD;
    By Soil P2;
**** Calculating the means of the soil \times P interaction to be used in the regression curves and graph ****;
Proc Sort Data=Raw;
    By Soil P;
Proc Means Data=Raw noprint;
    By Soil P;
    output out=Raw_Means mean=;
    var Yield;
**** Generating the different curves to be used in graphing the LSD bars ****;
Data Graph;
    Merge LSD (rename=(P2=P)) Raw_Means;
    By Soil P;
    HLSD=LSD/2;
    if Soil='black' then Y_black=Yield;
    if Soil='chestnut' then Y_chest=Yield;
    if Soil='black' then do;
      Yield1=Yield; output;
      Yield1=Yield – HLSD; output;
      Yield1=Yield + HLSD; output;
    end;
    if Soil='chestnut' then do;
Yield2=Yield; output;
Yield2=Yield – HLSD; output;
Yield2=Yield + HLSD; output;
end;
Drop _Type_ _Freq_; 
Run;

**** Graphic options for creating a CGM file as output ****;
**** Here you must to change the route and the name of the CGM file ****;
FileName Figures ‘C:\Multi Factorial Interactions\Fig. 3.CGM’;
Goptions Device=CGMOF97L GSFName=Figures GSFMode=Replace;

Proc GPlot Data=Graph;
   Plot (Y_black Y_chest)*P (Yield1 Yield2)*P / frame overlay vaxis=axis1 haxis=axis2
      nolegend;
   Symbol1 v=dot cv=black h=2.3 l=1 w=15 i=RL ci=black;
   Symbol2 v=dot cv=red h=2.3 l=1 w=15 i=RL ci=red;
   Symbol3 l=1 w=2 i=hiloct ci=black;
   Symbol4 l=1 w=2 i=hiloct ci=red;
   axis1 length=6.0 in order=(1.2 to 2.8 by 0.4)
      label=(f=hwcgm002 h=2.0 a=90 ‘Grain yield (Mg ha-1)’)
      value=(f=hwcgm002 h=2.0) offset=(1) minor=none;
   axis2 length=9.0 in order=(0 to 250 by 50)
      label=(f=hwcgm002 h=2.0 ‘P fertilizer rate (kg ha-1)’)
      value=(f=hwcgm002 h=2.0) offset=(3) minor=none;
   Title1 f=hwcgm002 h=0.2 ‘ ‘;
Run;

SAS Code for Graphing the Interaction Profiles in Factorial Experiments using the GLIMMIX Procedure:

An easy way to obtain the graphs for the mean response profiles for main effects and interactions in a factorial experiment is through the generalized linear mixed models procedure (GLIMMIX). Also, the response profile and 95% confident interval (CI) for the combination level of lsmeans for factors are included. The difference between the intervals for the interactions obtained using G PLOT and those computed using the GLIMMIX procedure are twofold: (i) GLIMMIX does not allow a curvilinear polynomial fit to the data but rather only depicts the lsmean profile (and its 95% confident interval) at each level of the factor plotted in the x axis, while the G PLOT procedure will fit the curvilinear polynomial and the LSD at each lsmean point; and (ii) the G PLOT produces a computer graphics metafile (CGM) that can be easily edited and modified; the graphs produced by GLIMMIX are not easy to edit and modify.
**** Data Reading from an external CSV File ****

Data Raw;

   Infile “C:\Yield & Gluten Data.csv” dlm=”,“ firstobs=3;
   Informat Soil$ 10.;
   Input Soil N P Rep Yield Gluten;

Datalines;

**** Three-way ANOVA using GLIMMIX procedure, including all two-way and three-way interactions ****

ODS graphics on;
ODS Select CovParms Tests3 MeanPlot;
Proc GLIMMIX Data=Raw;
   Class Rep Soil N P;
   Model Yield=Soil Rep(Soil) N P Soil*N Soil*P N*P Soil*N*P;
   Lsmeans Soil N P / plot=mean (join cl);
   Lsmeans Soil*N Soil*P / plot=mean (sliceby=Soil join cl);
   Lsmeans N*P / plot=mean (sliceby=N join cl);
   Lsmeans Soil*N*P / plot=mean (sliceby=Soil*N join cl);

ODS graphics off;
Run;

The ODS Graphics is used to generate by default several graphics related to the Lsmeans statement. The meanplot option or simply mean requests displaying the least squares means. For example, in the line Lsmeans Soil N P / plot=mean (join cl), the Lsmeans response profiles are requested for all the main effects soil, N, and P, simultaneously. The meanplot-options controls the display of the least squares means. Join or connect connects the least squares means with lines, while CL displays upper and lower confidence limits for the least squares means. By default, 95% limits are drawn. The confidence levels can be changed with the \(\alpha\) = option.

The statement Lsmeans Soil*N Soil*P / plot=mean (sliceby=Soil join cl) is used for simultaneously requesting the response profiles for the soil \(\times\) N and soil \(\times\) P interactions. Sliceby=fixed-effect specifies the soil effect by which to group the means in a single plot and the levels for the N and P effects to be drawn in the horizontal axis because soil is a qualitative factor, while N and P are quantitative factors.

Similarly, the statement Lsmeans N*P / plot=mean (sliceby=N join cl) is used to draw the individual response profiles for each N level for each P level in the horizontal axis.

Finally, the statement Lsmeans Soil*N*P / plot=mean (sliceby=Soil*N join cl) is useful for drawing the four response profiles for the combination of the levels of factors soil and N, both of them with two levels, and for each level of the P factor on the horizontal axis.
Agronomic experiments are often replicated over time and space to evaluate how treatments perform over a range of environments. The analysis of experiments conducted over more than one growing season (years) and/or places (locations) is commonly referred to as analysis of combined experiments. Common analyses of these studies treat some effects as fixed, treat others as random, and usually include interactions between fixed and random effects, which we call mixed interactions. Recommendations for how to treat mixed interactions has changed. In the traditional practice, the effects of interactions between fixed and random effects were assumed to sum to zero within each level of a fixed factor. Contemporary practice considers these effects to be mutually independent. This latter assumption is used to construct F-tests by many of the statistical analysis programs that are widely used to analyze data from agronomic experiments, but is inconsistent with that used in many previously published studies. The assumptions made about mixed interactions in the analysis of variance can result in very different interpretations and can potentially lead to different conclusions. This article addresses the discrepancy between the analyses that were formerly recommended and those that are currently implemented by popular software programs and provides recommendations for analyzing data from combined experiments.
SAS CODE FOR ANALYSIS OF COMBINED EXPERIMENTS

SAS Code for analyzing combined experiments using the GLM, MIXED, and GLIMMIX procedures. This code accepts all default assumptions associated with the procedure and can and should be augmented in some situations. In general, the more a dataset departs from the traditional assumptions, the more arguments will need to be included to perform an optimal analysis. We follow Stroup and Littell (2002) where possible. They recommend method=type3 rather than the default method=reml because method=type3 provides better control of type-I error rates for tests of fixed effects. The GLIMMIX procedure provides restricted maximum likelihood (REML) and other estimation methods, but not method=type3. The GLM procedure uses ordinary least squares to calculate the mean squares used in the analysis.

Treatment (Trt) and Location (Loc) fixed, Year (Yr) random:

```
proc mixed method=type3;
  class Yr Loc Blk Trt;
  model Yield = Loc Trt Trt*Loc;
  random Yr Yr*Loc Blk(Yr*Loc) Trt*Yr Trt*Yr*Loc;
```

```
proc glm;
  class Yr Loc Blk Trt;
  model Yield = Yr Loc Yr*Loc Blk(Yr*Loc) Trt*Yr Trt*Loc Trt*Yr*Loc;
  random Yr Yr*Loc Blk(Yr*Loc) Trt*Yr Trt*Yr*Loc / test;
```

```
proc glimmix;
  class Yr Loc Blk Trt;
  model Yield = Loc Trt Trt*Loc;
  random Yr Yr*Loc Blk(Yr*Loc) Trt*Yr Trt*Yr*Loc;
```

Treatment fixed, Year and Location random:

```
proc mixed method=type3;
  class Yr Loc Blk Trt;
  model Yield = Trt;
  random Yr Loc Yr*Loc Blk(Yr*Loc) Trt*Yr Trt*Loc Trt*Yr*Loc;
```

```
proc glm;
  class Yr Loc Blk Trt;
  model Yield = Yr Loc Yr*Loc Blk(Yr*Loc) Trt*Yr Trt*Loc Trt*Yr*Loc;
  random Yr Loc Yr*Loc Blk(Yr*Loc) Trt*Yr Trt*Loc Trt*Yr*Loc / test;
```

```
proc glimmix;
  class Yr Loc Blk Trt;
  model Yield = Trt;
  random Yr Loc Yr*Loc Blk(Yr*Loc) Trt*Yr Trt*Loc Trt*Yr*Loc;
```

Treatment, Year and Location random:

```
proc mixed method=type3;
  class Yr Loc Blk Var;
  model Yield = ;
  random Yr Loc Yr*Loc Blk(Yr*Loc) Var Var*Yr Var*Loc Var*Yr*Loc;
```

```
proc glm;
  class Yr Loc Blk Var;
  model Yield = Yr Loc Yr*Loc Blk(Yr*Loc) Var Var*Yr Var*Loc Var*Yr*Loc;
  random Yr Loc Yr*Loc Blk(Yr*Loc) Var Var*Yr Var*Loc Var*Yr*Loc / test;
```

```
proc glimmix;
  class Yr Loc Blk Var;
  model Yield = ;
  random Yr Loc Yr*Loc Blk(Yr*Loc) Var Var*Yr Var*Loc Var*Yr*Loc;
```
The Design and Analysis of Long-term Rotation Experiments
Roger William Payne

ABSTRACT

Rotation experiments are intended to compare different sequences of crop (and possibly husbandry) combinations. To avoid the conclusions being dependent on a specific sequence of years, it is advantageous to phase the start of the experiment, with new replicates of the rotations starting in successive years. Once a complete cycle has taken place, comparisons can then be made between the rotations in every subsequent year. If sufficient resources are available to have more than one replicate in each year, it will be possible to do an interim analysis with the data from a single year. Otherwise meaningful analyses will need several years data and the assumption e.g. that higher-order interactions can be ignored, or that responses over years can be modeled by low-order polynomials. Other analysis complications are that the within-year variances may be unequal, and that the correlation between observations on a plot may differ according to the distance in time between them. The old-fashioned method of analysis, feasible if the data are balanced, would be to do a repeated-measurements analysis of variance. A more recent, and more satisfactory, alternative is to do a mixed model analysis by REML, possibly fitting a model to the between-year correlation structure. The issues are illustrated using data from the Woburn Ley-Arable experiment.

Appendix 1. GenStat commands to analyze the Woburn Ley-arable experiment

" suppress messages and echoing of command lines "
SET [INPRINT=*, DIAGNOSTIC=warnings]
IMPORT [PRINT=*,] 'Table3.xls'
" calculate orthogonal polynomial contrasts over years "
CALCULATE X = Year
ORTHPOLYNOMIAL [MAXDEGREE=4] X; POLYNOMIAL=YearPol
CALCULATE LinYear,QuadYear,CubYear,QuartYear = YearPol[]
POINTER [VALUES=LinYear,QuadYear,CubYear,QuartYear] YearPol
CAPTION 'First analysis: do we need different variances in each year?';
STYLE=meta
VCOMPONENTS [FIXED=Rotation*N*YearPol[]] Year
REML [PRINT=*] Yield
VRACCUMULATE [PRINT=*, METHOD=restart] 'Constant variance'
VCOMPONENTS [FIXED=Rotation*N*YearPol[]; EXPERIMENTS=Year] Year
REML [PRINT=*] Yield
VRACCUMULATE [PRINT=deviance,dfrandom,aic]\n'Different variance in each year'
CAPTION 'Conclusion: yes we do we need different variances. ';
STYLE=stress
CAPTION 'Second analysis: can we simplify the fixed model?';
STYLE=meta
VDISPLAY [PRINT=Wald]
CAPTION 'Conclusion: drop the cubic and quartic polynomials. ';
STYLE=stress
CAPTION 'Third analysis: any further simplification of the fixed model?';
STYLE=meta
VCOMPONENTS  [FIXED=Rotation*N*YearPol[1,2]; EXPERIMENTS=Year] Year
REML  [PRINT=Wald] Yield
CAPTION 'Conclusion: drop Rotation.N.LinYear and Rotation.N.QuadYear.';
    STYLE=stress
CAPTION  'Fourth analysis: final model.'; STYLE=meta
VCOMPONENTS  [FIXED=Rotation*N*YearPol[1,2]- Rotation.N.YearPol[1,2]; EXPERIMENTS=Year] Year
REML  [PRINT=components,Wald] Yield
" form predicted means assuming quadratic year trends "
VARIATE  [VALUES=1981...2000] xlin
ORTHPOLYNOMIAL  [MAXDEGREE=2] xlin; POLYNOMIAL=xpred
VPREDICT  [PRINT=*,PREDICTIONS=predictedmeans; SED=sed]\n    LinYear,QuadYear,Rotation,N; LEVELS=xpred[],*,*;\n    PARALLEL=*,LinYear,*,*; NEWFACTOR=*,year,*,*\n    [LEVELS=!(1981...2000); LABELS=\n        t('81','82','83','84','85','86','87','88','89','90',\n        '91','92','93','94','95','96','97','98','99','00');\n        MODIFY=yes] year
CALCULATE  averageded = MEAN(sed)
    & maxsed = MAX(sed)
    & minsed = MIN(sed)
CAPTION  'Predicted means'; STYLE=minor
PRINT  [IPRINT=*,PREDICTIONS=predictedmeans]
CAPTION  'Standard errors of differences'; STYLE=minor
PRINT  ![t('average:','maximum:','minimum:'),\n        !(averageded,maxsed,minsed); JUST=left,right
" plot predicted means "
PEN  2...5; SYMBOL=0; LINESTYLE=2,3,5,7; THICKNESS=2
PEN  -1,-2; THICKNESS=2
DTABLE  [METHOD=line; XFREPRESENTATION=label] predictedmeans;\n    XFACTOR=year; GROUPS=N; TRELLIS=Rotation; PEN=!(2...5);\n    TITLE=''; YTITLE='yield t ha^{-1}'
Appendix 2. R and ASReml-R commands to analyze the Woburn Ley-arable experiment

# load asreml library
library(asreml)

# read data
Table3 <- asreml.read.table("Table3.txt", header=T)
summary(Table3)

# get factor versions of year and n
Table3$Year <- as.factor(Table3$year)
Table3$N <- as.factor(Table3$n)
head(Table3)

# 1: full fixed model with constant variance
model1 <- asreml(fixed=yield ~ Rotation*N*pol(year,4), random=~Year,
data=Table3)
summary(model1)
wald(model1, denDF="algebraic")
# calculate AIC for this model (on deviance scale - smaller = better)
aic1 <- -2*(model1$loglik - length(model1$gammas))
aic1

# 2: full fixed model with separate variances across years
model2 <- asreml(fixed=yield ~ Rotation*N*pol(year,4), random=~Year,
rcov=~at(Year):id(units), data=Table3)
summary(model2)
wald(model2)
# calculate AIC
aic2 <- -2*(model2$loglik - length(model2$gammas))
aic2

# compare AIC across models 1 and 2: model 2 better (smaller AIC)
aic1 - aic2

# construct individual vectors to separate out polynomial orders
matpol <- poly(Table3$year, degree=4)
Table3$linyear <- matpol[1:480,1]
Table3$quadyear <- matpol[1:480,2]
Table3$cubyear <- matpol[1:480,3]
Table3$quaryear <- matpol[1:480,4]

# 3: model 2 with polynomial components separated
model3 <- asreml(fixed=yield ~ Rotation*N*(linyear+quadyear+cubyear+quaryear),
random=~Year,
rcov=~at(Year):id(units), data=Table3)
summary(model3)
wald(model3, denDF="default")

# 4: drop cubic and quartic polynomial components
model4 <- asreml(fixed=yield ~ Rotation*N*(linyear+quadyear), random=~Year, rcov=~at(Year):id(units), data=Table3)
summary(model4)
wald(model4,denDF="default")

# 5: drop 3-way interaction and return to pol function (easier prediction)
model5 <- asreml(fixed=yield ~ Rotation*N*pol(year,2)-(Rotation:N:pol(year,2)), random=~Year, rcov=~at(Year):id(units), data=Table3)
summary(model5)
wald(model5,denDF="default")

# get predictions from final model
model5.pv <- predict(model5, classify=c("Rotation:N:year"), levels=list(Rotation=1:6,N=1:4, year=1981:2000))
model5.pv$predictions

# extract results
model5.pred <- model5.pv$predictions$pvals$predicted.value
model5.pR <- model5.pv$predictions$pvals$Rotation
model5.pN <- model5.pv$predictions$pvals$N

# make data frame containing predictions
model5.predict <- data.frame(pred=model5.pred, Rotation=model5.pR, N=model5.pN, year=model5.py)
model5.predict

# plot predictions
require(lattice)
xyplot(pred ~ year | Rotation, data=model5.predict, groups=N, auto.key=T)
# save to pdf file
pdf(file="xyplot.pdf")
xyplot(pred ~ year | Rotation, data=model5.predict, groups=N, auto.key=T)
dev.off()
Appendix 3. SAS commands to analyze the Woburn Ley-arable experiment

PROC IMPORT OUT = rotation
   DATAFILE = "&pathname.\long-term rotation\Table3.xlsx";
   SHEET = "Sheet1";
RUN;

* Centre covariates;

/*
DATA rotation; SET rotation;
   Year_num = Year_num - 1991;
   N = N - 105;
RUN;
*/

* Origin at Year = 1980;

DATA rotation; SET rotation;
   Year_num = Year_num - 1980;
RUN;

* Obtain the orthogonal polynomial, and merge it with the rest of the data;

PROC IML;
USE rotation;
READ ALL VAR {Year_num} INTO y;
yp = ORPOL(y,4);
cname = {"YearPol0" "YearPol1" "YearPol2" "YearPol3" "YearPol4"};
CREATE yp_data FROM yp [ COLNAME = cname ];
APPEND FROM yp;
RUN;
QUIT;

DATA yp_data2; SET yp_data;
   row_no = _N_;
RUN;

DATA rotation2; SET rotation;
   row_no = _N_;
   ysq = Year_num ** 2;
RUN;

PROC SQL;
   CREATE TABLE rotation3 AS
      SELECT a.*, b.*
      FROM rotation2 AS a, yp_data2 AS b
      WHERE a.row_no eq b.row_no;
QUIT;

* Fit model with heterogeneity of residual variance
* among years, with YearPol3 and YearPol4 omitted,
* with 3-way interactions also omitted
* and with terms in same order as in GenStat;

ODS RTF FILE = "sasrtf het year, no cub quart or 3-way.rtf";
PROC MIXED ASYCHOV DATA = rotation3 ;
   CLASS Year Rotation N Plot;
   MODEL Yield = Rotation N YearPol1 YearPol2 Rotation*N
   Rotation*YearPol1 N*YearPol1
   Rotation*YearPol2 N*YearPol2
   / DDFM = KENWARDROGER HType=1 3;
   RANDOM Intercept / subject=Year;
parms 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
repeated intercept / subject = Plot*year type=vc group=year;
RUN;
ODS RTF CLOSE;
Abstract

Non-linear regression models are important tools as many crop and soil processes are better represented by non-linear than by linear models. Fitting nonlinear models is not a single step procedure, but an involved process that requires careful examination of each individual step. Depending on the objective and the application domain, different priorities are set when fitting nonlinear models and these include obtaining acceptable parameter estimates, and a good model fit while meeting standard assumptions of statistical models. We propose steps in fitting nonlinear models as described by a flow diagram and discuss each step separately providing examples and updates on procedures used. The following steps are considered: 1) choose candidate models, 2) set starting values, 3) fit models, 4) check convergence and parameter estimates, 5) find the “best” model among competing models, 6) check model assumptions (residual analysis), and 7) calculate statistical descriptors, and confidence intervals. The associated feedback mechanisms are also addressed (i.e. model variance homogeneity). In particular, we emphasize the first step (choose candidate models) by providing an extensive library of nonlinear functions (77 equations with the associated parameter meaning) and typical application examples in agriculture. We hope that this contribution will clarify some of the difficulties and confusion with the task of using nonlinear models.

Sotirios V Archontoulis, Iowa State University, Department of Agronomy, 1206 Agronomy Hall, Ames, Iowa, 50011; Fernando E Miguez, Iowa State University, Department of Agronomy, 1206 Agronomy Hall, Ames, Iowa, 50011. *Corresponding author (femiguez@iastate.edu).

Keywords: non-linear model, goodness of fit, model comparison, agriculture, residual analysis

### code chunk number 3: nlraa.Rnw:84-86

```r
sm$eu <- with(sm, factor(Block):factor(Input):factor(Crop))
sm2 <- subset(sm, DOY != 141)
```

### code chunk number 4: nlraa.Rnw:92-93

```r
smG <- groupedData(Yield ~ DOY | eu, data = sm2)
```

### code chunk number 5: nlraa.Rnw:100-101

```r
fit.lis <- nlsList(Yield ~ SSbgf(DOY, w.max, t.e, t.m), data = smG)
```

### code chunk number 6: nlslisResid

```r
print(plot(fit.lis))
```

### code chunk number 7: nlslisIntervals

```r
print(plot(intervals(fit.lis)))
```

### code chunk number 8: nlraa.Rnw:135-136

```r
fit.me <- nlme(fit.lis, control = list(minScale =1e-50, pnlsTol=0.01))
```

### code chunk number 9: nlmeResid

```r
print(plot(fit.me))
```

### code chunk number 10: nlmeAugPred
```r
print(plot(augPred(fit.me, level = 0:1)))
```

```r
### code chunk number 11: nlraa.Rnw:172-175
###
fit.lis2 <- nlsList(Yield ~ bgf2(DOY, w.max, w.b = 0, t.e, t.m, t.b = 141),
    data = smG,
    start = c(w.max = 30, t.e = 280, t.m = 240))
```

```r
### code chunk number 12: nlslisResid2
###
print(plot(fit.lis2))
```

```r
### code chunk number 13: nlraa.Rnw:193-196
###
fit.me2 <- nlme(fit.lis2)
fit2.me2 <- update(fit.me2, random = pdDiag(w.max + t.e + t.m ~ 1))
anova(fit.me2, fit2.me2)
```

```r
### code chunk number 14: nlraa.Rnw:203-219
###
fe <- fixef(fit2.me2) ## Some starting values with visual help
fit3.me2 <- update(fit2.me2, fixed = list(w.max + t.e + t.m ~ Crop),
    start = c(fe[1], -10, 20, fe[2], -40, 0, fe[3], -40, 0))
## We next include the Input
fe2 <- fixef(fit3.me2)
fit4.me2 <- update(fit3.me2, fixed = list(w.max + t.e + t.m ~ Crop + Input),
    start = c(fe2[1:3], 0, fe2[4:6], 0, fe2[7:9], 0))
## and the interaction
fe3 <- fixef(fit4.me2)
fit5.me2 <- update(fit4.me2,
    fixed = list(w.max + t.e + t.m ~ Crop + Input + Crop:Input),
    start = c(fe3[1:4], 0, 0, fe3[5:8], 0, 0),
```

Page 39 of 55
fe3[9:12, 0, 0))

#######################################################
### code chunk number 15: nlmeResid2
#######################################################
print(plot(fit5.me2))

#######################################################
### code chunk number 16: nlraa.Rnw:241-247
#######################################################
fit6.me2 <- update(fit5.me2,
        weights = varPower(form = ~ fitted(.) | Crop))

fit7.me2 <- update(fit6.me2, weights = varPower(form = ~ fitted(.)))

anova(fit6.me2, fit7.me2)

#######################################################
### code chunk number 17: nlraa.Rnw:253-254
#######################################################
fit6.me2

#######################################################
### code chunk number 18: nlraa.Rnw:260-268
#######################################################
## Random effects are almost zero
fit8.me2 <- gnls(Yield ~ bgf2(DOY, w.max, t.e, t.m, w.b=0, t.b=141),
        data = smG,
        params = list(w.max + t.e + t.m ~ Crop + Input + Crop:Input),
        weights = varPower(form = ~ fitted(.) | Crop),
        start = fixef(fit7.me2))

anova(fit6.me2, fit8.me2)

#######################################################
### code chunk number 19: nlraa.Rnw:273-274
#######################################################
anova(fit8.me2)
### code chunk number 20: nlraa.Rnw:283-284
print(plot(fit8.me2))

### code chunk number 21: nlraa.Rnw:289-298
smG$prds <- fitted(fit8.me2)

dois <- 168:303
ndat <- expand.grid(DOY=dois, Crop= unique(smG$Crop), Input=c(1,2))
ndat$preds <- predict(fit8.me2, newdata = ndat)

ndat2 <- ndat
ndat2[ndat2$Crop == "M" & ndat2$DOY > 270,"preds"] <- NA
ndat2 <- na.omit(ndat2)
ABSTRACT

Many agronomic researchers measure and collect multiple response variables in an effort to understand the more complex nature of the system being studied. Multivariate (MV) statistical methods encompass the simultaneous analysis of all random variables measured on each experimental or sampling unit. Many agronomic research systems studied are, by their very nature, MV; however, most analyses reported are univariate (analysis of one response at a time). The objective of this review is to outline a statistical foundation of applications of MV methods and techniques for the agronomic sciences. By utilizing two agronomic data sets, both typical in dimension and structure, we discuss three classes of MV techniques based on the research question and characteristics of the data: (i) hypothesis driven, such as MV analysis of variance; (ii) dimension reduction, such as principal components analysis; and (iii) classification and discrimination, which includes canonical discriminant analysis. Several advantages and disadvantages of the MV tools are explained. This review will provide researchers with a beginning framework of MV generalizations of univariate techniques, and methods that are unique to MV dimension analysis. It is important for researchers to capture the concept of variability within a MV data set to better understand the complex system.

K.M. Yeater, USDA–ARS, Southern Plains Area, 1001 Holleman Dr. E., College Station, TX 77840-4117; S.E. Duke, USDA–ARS, Southern Plains Agricultural Research Center, 2765 F&B Rd., College Station, TX 77845; and W.E. Riedell, USDA–ARS, North Central Agricultural Research Lab., 2923 Medary Ave., Brookings, SD 57006. Received 10 Jan. 2014. *Corresponding author (kathleen.yeater@ars.usda.gov).

Abbreviations: CCA, canonical correlation analysis; CDA, canonical discriminant analysis; DA, discriminant analysis; MANOVA, multivariate analysis of variance; MV, multivariate; PC, principal component; PCA, principal components analysis; RV, random variable.
SAS code examples for generating Multivariate Analyses with Experimental Data

Description of SAS program for MANOVA

The following SAS program example employs the multivariate analysis of variance (MANOVA) to measure differences in the leaf mineral nutrients (N, P, K, Ca, Mg, Zn, Fe, Mn, and Cu), chlorophyll (chl), leaf area index (lai), and grain yield (GY). The following statements invoke the GLM PROCEDURE to perform the MANOVA that generated the output as presented in Tables 1 and 2. Unlike in univariate ANOVA, all dependent response variables tested are included on the left side of the equation in the MODEL statement. The inclusion of a MANOVA statement, following the model statement, tells SAS to perform MV hypothesis tests for the model effects. All four multivariate tests are computed via ‘h=_all_’ within the MANOVA statement. Exact F tests are calculated via the MSTAT=EXACT option. The SAS output includes univariate analysis (each variable analyzed alone) unless a <nouni> option (no univariate analysis) is used in the PROC GLM statement.

SAS code program for MANOVA example

ods graphics on;
proc glm data=mvms.oat_aphid_study;
   class Year Treatment Rep;
   model N P K Ca Mg Zn Fe Mn Cu chl lai GY = Treatment Year Treatment*Year;
   manova h = _all_ / mstat=exact summary;
run;
quit;
ods graphics off;

Description of SAS program for PCA

This example program analyzes and generates the first five principal components for grain measures (yield, oil, starch, N, P, K, S, Ca, Mg, Fe, Mn, and Zn), plant measures (N, P, K, Ca, Mg, Fe, Mn, Zn) and soil measures (NO3, P, K, Ca, Mg, Fe, Mn, and Zn) using the PRINCOMP PROCEDURE. First, each set of variables (grain, plant, soil) was investigated with separate PCA assessments (data not shown). Then all variables were combined and data step programming was used to re-label and merge the corresponding output. The generated output is presented in Tables 3 and 4. The default plots are generated from the ‘plots=all’ option and scree plot and corresponding variance plot provided as Figure 1. The observations on plots are labeled with the ‘ID’ statement calling for nitrogen input and crop rotation (variable name: in_rot).

SAS code program for PCA example and plots of principal components

ods graphics;
ods rtf file='C:\pca_grain.rtf';
proc princomp data=mvms.all_data_source_a out=mvms.pcag n=5 plots=all;
id in_rot;
var Yield Oil Starch Grain_N Grain_P Grain_K Grain_S Grain_Ca Grain_Mg Grain_Fe Grain_Mn Grain_Zn;
run;
ods rtf close;
ods rtf file='C:\pca_plant.rtf';
proc princomp data=mvms.all_data_source_a out=mvms.pcap n=5 plots=all;
id in_rot;
var Plant_DW Plant_N Plant_P Plant_K Plant_Ca Plant_Mg Plant_Fe Plant_Mn Plant_Zn;
run;
ods rtf close;
ods rtf file='C:\pca_soil.rtf';
proc princomp data=mvms.all_data_source_a out=mvms.pcas n=5 plots=all;
id in_rot;
var Soil_NO3 Soil_P Soil_K Soil_Ca Soil_Mg Soil_Fe Soil_Mn Soil_Zn;
run;
ods rtf close;
proc princomp data=mvms.all_data_source_a out=mvms.pca_all n=5 plots=all;
id in_rot;
var Soil_NO3 Soil_P Soil_K Soil_Ca Soil_Mg Soil_Fe Soil_Mn Soil_Zn Plant_DW Plant_N Plant_P Plant_K Plant_Ca Plant_Mg Plant_Fe Plant_Mn Plant_Zn Yield Oil Starch Grain_N Grain_P Grain_K Grain_S Grain_Ca Grain_Mg Grain_Fe Grain_Mn Grain_Zn;
run;
data mvms.PL_pca (drop=Soil_NO3 Soil_P Soil_K Soil_Ca Soil_Mg Soil_Fe Soil_Mn Soil_Zn Plant_DW Plant_N Plant_P Plant_K Plant_Ca Plant_Mg Plant_Fe Plant_Mn Plant_Zn Yield Oil Starch Grain_N Grain_P Grain_K Grain_S Grain_Ca Grain_Mg Grain_Fe Grain_Mn Grain_Zn);
set mvms.pcaP;
rename prin1=PL_pca1 prin2=PL_pca2 prin3=PL_pca3;
run;
data mvms.S_pca (drop=Soil_NO3 Soil_P Soil_K Soil_Ca Soil_Mg Soil_Fe Soil_Mn Soil_Zn Plant_DW Plant_N Plant_P Plant_K Plant_Ca Plant_Mg Plant_Fe Plant_Mn Plant_Zn Yield Oil Starch Grain_N Grain_P Grain_K Grain_S Grain_Ca Grain_Mg Grain_Fe Grain_Mn Grain_Zn);
set mvms.pcaS;
rename prin1=S_pca1 prin2=S_pca2 prin3=S_pca3;
run;
data mvms.G_pca (drop=Soil_NO3 Soil_P Soil_K Soil_Mg Soil_Fe Soil_Mn Soil_Zn Plant_DW Plant_N Plant_P Plant_K Plant_Ca Plant_Mg Plant_Fe Plant_Mn Plant_Zn Yield Oil Starch Grain_N Grain_P Grain_K Grain_S Grain_Ca Grain_Mg Grain_Fe Grain_Mn Grain_Zn); set mvms.pcaG; rename prin1=G_pca1 prin2=G_pca2 prin3=G_pca3; run;

data mvms.all_princ; merge mvms.G_pca mvms.PL_pca mvms.S_pca; by obs year input rotation rep; run;

Description of SAS program for CDA

This example program of the CANDISC PROCEDURE for canonical discriminant analysis created the output in Tables 5 and 6. The soil measures of NO3, Ca, Fe, Mn, plant measures of N, P, K, Ca, Zn, and grain measures of N, P, K, S, and Mg are used to discriminate and classify among the crop rotations (variable name: rotat). The DISTANCE option displays squared Mahalanobis distances between the class group means, F statistics, and the corresponding probabilities of greater squared Mahalanobis distances between the class group means. Additionally, SAS coding in the Graph Template Language using the TEMPLATE PROCEDURE and SGRENDER PROCEDURE is provided, which generated the plot of the first two canonical variables in Figure 2.

SAS code program for CDA example and plot of canonical variables

proc candisc data=mvms.all_data_source_a out=mvms.outcan distance; class rotation; var Soil_NO3 Soil_Ca Soil_Fe Soil_Mn Plant_N Plant_P Plant_K Plant_Ca Plant_Mg Plant_Fe Plant_Mn Plant_Zn Grain_N Grain_P Grain_K Grain_S Grain_Ca Grain_Mg Grain_Fe Grain_Mn Grain_Zn ; run;

proc template; define statgraph sgdesign; dynamic _CAN1 _CAN2 _ROTATION; begingraph; entrytitle halign=center 'Influence from Crop Rotation'; layout lattice / rowdatarange=data columndatarange=data rowgutter=10 columngutter=10; layout overlay / xaxisopts=( label=('Canonical Variable 1') linearopts=( viewmin=-4.0 viewmax=5.0)) yaxisopts=( label=('Canonical Variable 2') linearopts=( viewmin=-3.5 viewmax=4.25)); scatterplot x=_CAN1 y=_CAN2 / group=_ROTATION name='scatter';
discretelegend 'scatter' / opaque=false border=true halign=right valign=top displayclipped=true across=1 order=rowmajor location=inside;
endlayout;
endlayout;
endgraph;
end;
run;

proc sgrender data=WORK.OUTCAN template=sgdesign;
dynamic _CAN1="CAN1" _CAN2="CAN2" _ROTATION="ROTATION";
run;
Rethinking the Analysis of Non-Normal Data in Plant and Soil Science

Walter W. Stroup

ABSTRACT

The introduction of high-quality, useable generalized linear mixed model (GLMM) software in the mid-2000s changes the conversation regarding the analysis of non-normal data from designed experiments. For well over half a century, the reigning paradigm called for using analysis of variance (ANOVA), either assuming approximate normality of the original data or applying a variance-stabilizing transformation. The appearance of GLMMs creates a dilemma. ANOVA based analyses and GLMM-based analyses often yield mutually contradictory results. What results should a researcher report, and how should the choice be justified? If GLMM-based analysis is preferred—and there is increasing evidence that this is the case—approaches to data analysis ingrained while learning ANOVA must be unlearned and relearned. In this paper, the basic issues associated with the analysis of non-normal data are reviewed, thought processes required for GLMMs, and how they differ from traditional ANOVA are introduced, and three examples are presented giving an overview of GLMM-based analysis. The three examples include discussions of what is known to date about the relative merits of GLMM- and ANOVA based analysis of non-normal data.

In SAS programs: lower case means you need to type exactly what is shown; UPPER CASE are names that are users’ option.

Example 1: Randomized Complete Block Design, Binomial Data

The data:

/* RCBD: 8 blocks, 2 treatments */
/* Response: Y ~ binomial(N=100,prob_trt) */
/* at each block*treatment combination */

Data INTRO_BINOMIAL;
Input BLOCK TREATMENT Y N;
/* N = number of observations at each experimental unit */
/* (i.e. block*treatment combination */
/* Y = number of “favorable outcomes” */
Datalines;
1 0 98 100
1 1 94 100
2 0 95 100
2 1 36 100
3 0 93 100
3 1 85 100
4 0 94 100
4 1 88 100
5 0 99 100
5 1 91 100
6 0 61 100
6 1 82 100
7 0 84 100
7 1 43 100
8 0 92 100
8 1 71 100
;
SAS Statements for Usual Analysis:

```sas
proc glimmix data=INTRO_BINOMIAL;
  class BLOCK TREATMENT;
  model Y/N=TREATMENT;
  random intercept TREATMENT / subject=BLOCK;
  lsmeans TREATMENT / cl ilink;
run;
```

SAS Statements for GEE Analysis (see example for when this would be appropriate instead of the “usual analysis.”)

```sas
proc glimmix data=INTRO_BINOMIAL;
  class BLOCK TREATMENT;
  model Y/N=TREATMENT;
  random TREATMENT / type=CS subject=BLOCK residual; /* this line => GEE */
  lsmeans TREATMENT / ilink cl;
run;
```
Example 2: Split-Plot with Counts

The data:

```plaintext
data COUNT_SPLT_PLT;
  input BLOCK A B COUNT;
datalines;
1 1 1 7
1 1 2 3
1 2 1 49
1 2 2 22
2 1 1 6
2 1 2 17
2 2 1 61
2 2 2 8
3 1 1 1
3 1 2 17
3 2 1 6
3 2 2 4
4 1 1 1
4 1 2 1
4 2 1 5
4 2 2 2
5 1 1 1
5 1 2 12
5 2 1 0
5 2 2 97
6 1 1 4
6 1 2 3
6 2 1 5
6 2 2 17
;
/* (1) mean plots of counts                */
/* the ONLY useful output here is the plot */
/* THE ANALYSIS FROM THIS RUN IS INVALID */
/* but DOES get plot of raw sample means */
proc glimmix data=COUNT_SPLT_PLT;
  class BLOCK A B;
  model COUNT=A|B;
  random BLOCK BLOCK*A;
  lsmeans A*B / plot=meanplot(sliceby=A join);
run;
/* NAIVE Poisson: shows overdispersion diagnostic and bogus ANOVA p-values */
/* Pearson chi-square/DF is ONLY valid output from this run */
proc glimmix data=COUNT_SPLT_PLT method=quad;
  class BLOCK A B;
  model COUNT=A|B/d=poi;
  random intercept A / subject=BLOCK;
run;
/* (2) Negative Binomial GLMM    */
/* this is the correct analysis to use */
proc glimmix data=COUNT_SPLT_PLT method=quad;
  class BLOCK A B;
  model COUNT=A|B/d=negbin;
```
random intercept A / subject=BLOCK;
lsmeans A*B / ilink plot=meanplot(sliceby=A join ilink);
run;
Example 3: Repeated Measures, Binomial Data

The data:
```
data EX3_RPTM;
  input TRT PLOT TIME Y N;
datalines;
0 1 1 15 50
0 1 2 12 50
0 1 3 21 50
0 1 4 9 50
0 1 5 12 50
1 1 1 20 50
1 1 2 18 50
1 1 3 22 50
1 1 4 35 50
1 1 5 27 50
0 2 1 11 50
0 2 2 9 50
0 2 3 15 50
0 2 4 12 50
0 2 5 12 50
1 2 1 9 50
1 2 2 14 50
1 2 3 13 50
1 2 4 15 50
1 2 5 17 50
0 3 1 13 50
0 3 2 14 50
0 3 3 18 50
0 3 4 27 50
0 3 5 32 50
1 3 1 12 50
1 3 2 15 50
1 3 3 14 50
1 3 4 14 50
1 3 5 14 50
0 4 1 3 50
0 4 2 4 50
0 4 3 12 50
0 4 4 8 50
0 4 5 14 50
1 4 1 19 50
1 4 2 16 50
1 4 3 16 50
1 4 4 21 50
1 4 5 19 50
0 5 1 6 50
0 5 2 10 50
0 5 3 5 50
0 5 4 14 50
0 5 5 8 50
1 5 1 8 50
1 5 2 12 50
```
SAS statements begin next page
SAS Statements for Binomial Repeated Measures

/* 1 - get AICC for indep model and mean plot of PCT        */
/* DO NOT use this analysis unless it has the smallest AICC */
/* of all models considered                               */
proc glimmix data=EX3_RPTM method=laplace;
  class TRT PLOT TIME;
  model Y/N=TRT|TIME;
  random intercept TRT / subject=PLOT;
  lsmeans TRT*TIME / ilink plot=meanplot(sliceby=TRT join ilink);
run;

/* 2 - get AICC for AR(1) model */
proc glimmix data=EX3_RPTM method=laplace;
  class TRT PLOT TIME;
  model y/n=TRT|TIME;
  random intercept / subject=PLOT;
  random TIME / subject=PLOT*TRT type=ar(1);
run;

/* 3 rerun above replacing, in turn */
random intercept / subject=PLOT;
random TIME / subject=PLOT*TRT type=ar(1);
with the following
/* 3a */
random TIME / subject=PLOT*TRT type=cs;
/* 3b */
random TIME / subject=PLOT*TRT type=arh(1);
/* 3c */
random TIME / subject=PLOT*TRT type=ante(1);

/* for step 4 use the type=___ that gives you the smallest AICC */
/* in this example, AR(1) had the lowest AICC */
/* therefore */
/* 4 - proceed with the analysis */
proc glimmix data=EX3_RPTM;
  class TRT PLOT TIME;
  model Y/N=TRT|TIME/ddfm=kr2;
  random intercept / subject=PLOT;
  random TIME / subject=PLOT*TRT type=ar(1);
  lsmeans TRT*TIME / slicediff=TIME slice=TRT ilink
    plot=meanplot(sliceby=TRT join ilink);
  /* add contrasts that seem reasonable here */
  /* one size-fit-all recommendation not possible and unwise */
run;
SAS statements in cases for which GEE is preferable (rare in agronomic research applications, but it does happen – see article):

```sas
proc glimmix data=EX3_RPTM empirical=mbn;
  class TRT PLOT TIME;
  model Y/N=PLOT TRT|TIME;
  random TIME / subject=PLOT*TRT type=ar(1) residual;
  lsmeans TRT*TIME / slicediff=TIME slice=TRT ilink
    plot=meanplot(sliceby=TRT join ilink);
run;
```