Scientific Research Publishing

American Journal of Plant Sciences, 2017, 8, 1672-1698
http://www.scirp.org/journal/ajps
ISSN Online: 2158-2750
ISSN Print: 2158-2742

# RGxE: An R Program for Genotype x Environment Interaction Analysis

**Mahendra Dia[1], Todd C. Wehner[1*], Consuelo Arellano[2]**

[1]Department of Horticultural Science, North Carolina State University, Raleigh, USA
[2]Statistics Department, North Carolina State University, Raleigh, USA
Email: *tcwehner@gmail.com

**How to cite this paper:** Dia, M., Wehner, T.C. and Arellano, C. (2017) RGxE: An R Program for Genotype x Environment Interaction Analysis. *American Journal of Plant Sciences*, **8**, 1672-1698.
https://doi.org/10.4236/ajps.2017.87116

**Received:** March 31, 2017
**Accepted:** June 20, 2017
**Published:** June 26, 2017

## Abstract

Genotype x environmental interaction (GxE) can lead to differences in performance of genotypes over environments. GxE analysis can be used to analyze the stability of genotypes and the value of test locations. We developed an Rlanguage program (RGxE) that computes univariate stability statistics, descriptive statistics, pooled ANOVA, genotype *F* ratio across location and environment, cluster analysis for location, and location correlation with average location performance. Univariate stability statistics calculated are regression slope ($b_i$), deviation from regression ($S^2_d$), Shukla's variance ($\sigma_i^2$), S square Wricke's ecovalence ($W_i$), and Kang's yield stability ($YS_i$). RGxE is free and intended for use by scientists studying performance of polygenic or quantitative traits over multiple environments. In the present paper we provide the RGxE program and its components along with an example input data and outputs. Additionally, the RGxE program along with associated files is also available on GitHub at https://github.com/mahendra1/RGxE, http://cucurbitbreeding.com/todd-wehner/publications/software-sas-r-project/ and http://cuke.hort.ncsu.edu/cucurbit/wehner/software.html.

## Keywords

Genotype x Environment Interaction, R Programming Language, RGxE, Univariate, Multivariate

## 1. Introduction

Genotype x environmental interaction (GxE) refers to the modification of genetic factors by environmental factors, and to the role of genetic factors in determining the performance of genotypes in different environments. GxE can occur for quantitative traits of economic importance and is often studied in plant and animal breeding, genetic epidemiology, pharmacogenomics and conservational biology research. The traits include reproductive fitness, longevity, height, weight, yield, and disease resistance.

Selection of superior genotypes in target environments is an important objective of plant breeding programs. A target environment is a production environment used by growers [1] [2] [3] [4] [5]. In order to identify superior genotypes across multiple environments, plant breeders conduct trials across locations and years, especially during the final stages of cultivar development. GxE is said to exist when genotype performance differs over environments. Performance of genotype can vary greatly across environment because of the effect of environment on trait expression. Cultivars with high and stable performance are difficult to identify, but are of great value [6] [7].

Since it is impossible to test genotypes in all target environments, plant breeders do indirect selection using their own multiple-environment trials, or test environments. GxE reduces the predictability of the performance of genotypes in target environments based on genotype performance in test environments [8]. An important factor in plant breeding is the selection of suitable test locations, since it accounts for GxE and maximizes gain from selection [9]. An efficient test location is discriminating, and is representative of the target environments for the cultivars to be released. Discriminating locations can detect differences among genotypes with few replications. Representative locations make it likely that genotypes selected will perform well in target environments [9].

The analysis of variance (ANOVA) is useful in determining the existence, size and significance of GxE. In order to determine GxE for a group of elite cultivars, genotypes are often considered to be fixed effects and environments random. However, for the purpose of estimating breeding values using best linear unbiased prediction (BLUP), genotypes are considered to be random and environments fixed. Some statisticians consider genotypes random effect, provided that the objective is to select the best ones [10]. If GxE is significant, additional stability statistics can be calculated.

Several statistical methods have been proposed for stability analysis. These methods are based on univariate and multivariate models. The present paper focuses on univariate models for the analysis of stability measured using R programming, so a brief description of each stability measure is provided below.

The most widely used methods are univariate stability models based on regression and variance estimates. According to the regression model, stability is expressed in terms of the trait mean ($M$), the slope of regression line ($b_i$) and the sum of squares for deviation from regression $\left(S_d^2\right)$. High mean of a genotype performance is a precondition of stability. The slope ($b_i$) of regression indicates the response of genotype to the environmental index, which is derived from the average performance of all genotypes in each environment. If $b_i$ is not significantly different from unity, the genotype is adapted in all environments. A $b_i$ greater than unity describes genotypes with higher sensitivity to environmental change (below average stability), and greater specificity of adaptability to high yielding environments. A $b_i$ less than unity provides a measure of greater resistance to environmental change (above average stability), and therefore increasing specificity of adaptability to low yielding environments.

The variance parameters that measure stability statistics include stability ecovalence $\left(W_i^2\right)$ proposed by [11], stability variance $\left(\sigma_i^2\right)$ proposed by [12], and yield stability ( $YS_i$) proposed by [13].

Ecovalence stability index $\left(W_i^2\right)$ of a genotype is its contribution to the GxE squared and summed across all environments. Since the value of $W_i^2$ is expressed as a sum of squares, a test of significance for $W_i^2$ is not available. [12] proposed an unbiased estimate $\left(\sigma_i^2\right)$ of the variance of GxE plus an error term associated with genotype. Shukla's stability variance $\left(\sigma_i^2\right)$ is a linear combination of Wricke's ecovalence $\left(W_i^2\right)$. Shukla's stability statistic measures the contribution of a genotype to the GxE and error term, therefore a genotype with low $\sigma_i^2$is regarded as stable. According to [13], $W_i^2$ and $\sigma_i^2$ are equivalent in ranking genotypes for stability.

The [14] stability statistic ( $YS_i$) is a nonparametric stability procedure in which both the mean ($M$) and [12] stability variance $\left(\sigma_i^2\right)$ for a trait are used as selection criteria. This method gives equal weight to $M$ and $\sigma_i^2$. According to this method, genotypes with $YS_i$ greater than the mean $YS_i$ are considered stable [14] [15] [16].

Genotype $F$ ratio for each test location and correlation of test location with average location are important measures of location value. When the mean of all genotypes are equal, then the $F$ ratio will be close to 1. If analysis of variance is run by location, then high genotype $F$ ratio indicates high discriminating ability for that location. High and significant value of Pearson correlation of each location with the mean of all locations indicates strong representation of mean location performance.

Our objective was to develop an Rlanguage program (RGxE) that gives an output for genotype stability and location value using univariate models, descriptive statistics, genotype $F$ ratio across location and environment, cluster analysis for location, and location correlation with average location performance. In addition to the RGxE program, [17] provided a SAS program (SASGxE) that computes multivariate stability statistics using R program along with univariate stability statistics and location value using SAS programming. These multivariate stability statistics include the additive main effects and multiplicative interaction (AMMI) model, and genotype main effects plus GxE (GGE) model. RGxE uses R software (version 3.1.3 and higher). RGxE is freely available, annotated, and intended for scientists studying performance of polygenic or quantitative traits under different environmental conditions. In the present paper we provide the general features of RGxE program and along with the functionality of each module and their outputs. A supplemental file is provided with the RGxE program, instructions for the user-enetered fields required in RGxE program, interpretation of univariate stability statistics, example input data, and output from example input data. The RGxE program along with associated files is also available on GitHub at https://github.com/mahendra1/RGxE, http://cucurbitbreeding.com/todd-wehner/publications/software-sas-r-project/ and http://cuke.hort.ncsu.edu/cucurbit/wehner/software.html.

## 2. General Features and Functionality of the RGxE Program

### 2.1. Overview of the RGxE Program

RGxE is a user friendly and annotated R program that will allow user to analyze genotype stability and evaluate test location value of balanced mult-location replicated trial data. This program generates output (.csv or .txt) into the same folder from where it reads input dataset and Console window of helper application "R studio" [18] of R statistical software [19]. A schematic representation of RGxE is presented in Figure 1. Below are the key components of RGxE program which user can independently run.

### 2.2. Installing and Loading Packages

RGxEuses **dplyr** [20], **tidyr** [21], **broom** [22], **agricolae** [23], **lme4** [24], **afex** [25], **cluster** [26], and **grDevices** [19] packages. The **dplyr**, **tidyr**, **broom**, **agri-**

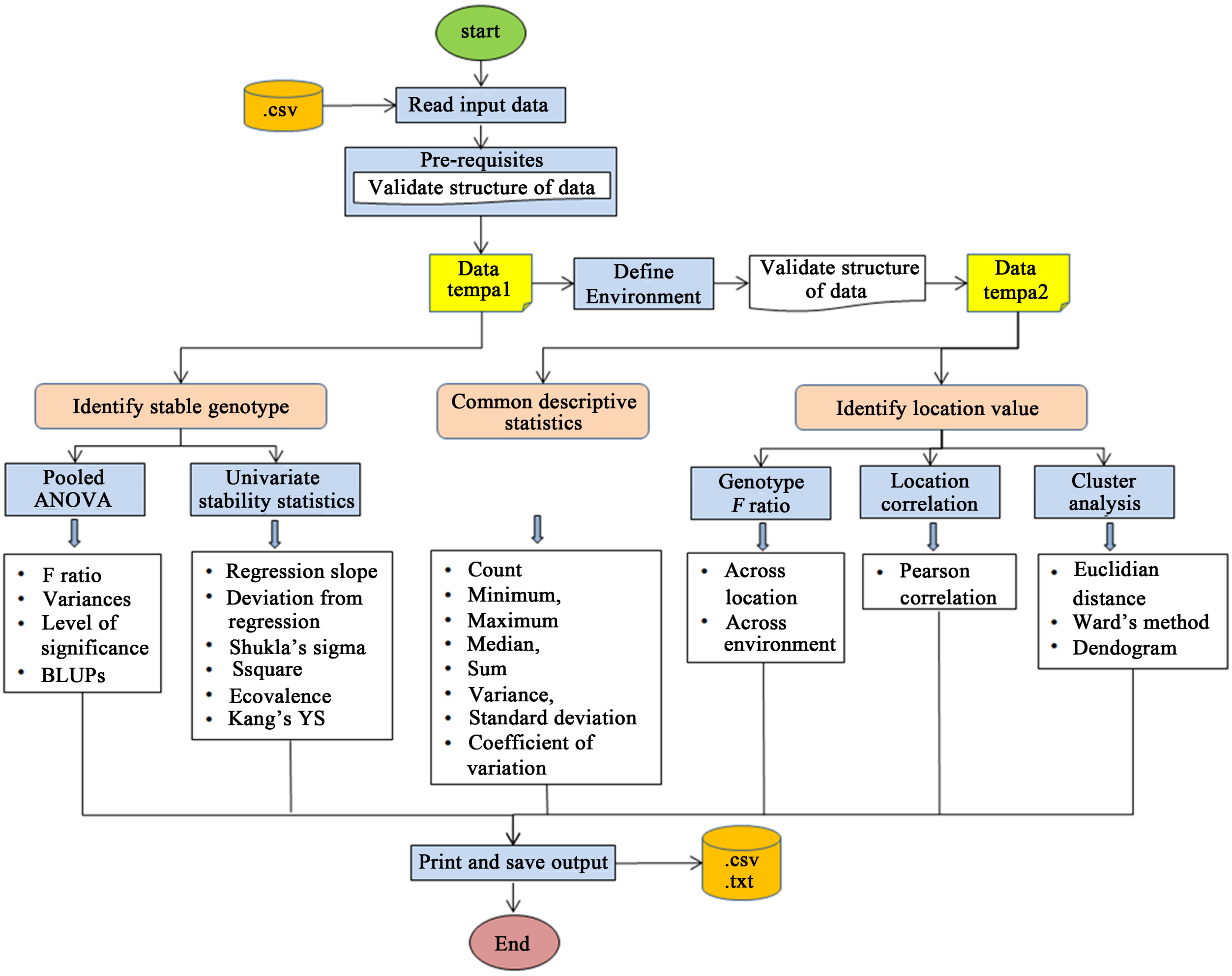

Figure 1. Overview of overall process of RGxE program for genotype stability and location value.

**colae**, **lme4**, **afex**, **cluster**, and **grDevices** packages are available from the Comprehensive R Archive Network (CRAN), therefore they can be installed as any other packages, by simply typing:

```
install.packages("dplyr")
install.packages("tidyr")
install.packages("broom")
install.packages("agricolae")
install.packages("lme4")
install.packages("afex")
install.packages("cluster")
install.packages("grDevices")
```

Once installed, the packages have to be loaded before they can be used. This can be done through the `library()` or `require()` command, as shown below.

```
library(tidyr)
library(dplyr)
library(sqldf)
library(lme4)
library(afex)
library(broom)
library(agricolae)
library(cluster)
library(grDevices)
```

## 2.3. Input Data and Validation

RGxE starts with user-entered field to read input data. Instructions on user enetered fields are presented in Supplemental Material. The user is required to set current working directory using `setwd()`, which is input data file location, and pass input data file name. RGxE requires an input data file in .csv (comma separated value) format. Highlighted fields are user entered in the code shown below for Windows and iOS (Mac) operating system, respectively.

```
setwd("E:/PhD Research Work/PhD Articles")
#### For Windows user ####
tempa<- read.csv("RGxEInputData2_2016_02_15.csv", header = TRUE)
#### For iOS or Mac user ####
file.name <- "E:/PhD Research Work/PhD Articles/RGxEInputData2_2016_02_15.csv"
out.name <- "E:/PhD Research Work/PhD Articles/GxEROutput.csv"
tempa<- read.csv(file.name)
```

The input data file is comprised of column names including **YR** (year), **LC** (location), **RP** (replication), **CLT** (cultigen or genotype), and dependent variable (**Trait**). Sample input data is presented in Supplemental Material. User is required **not** to change the column names as program takes same variable name for the analysis. Dependent variable in example input data is yield (Mg·ha$^{-1}$) of watermelon. Hereafter, a word "genotype" is used to indicate cultigen, cultivar, variety or genotype. RGxE validates the structure of input data, with below arguments, so that correct column types (numeric, logical, factor, or character) are used for statistical analysis.

```
tempa$YR<- as.factor(tempa$YR)
tempa$RP<- as.factor(tempa$RP)
tempa$LC<- as.factor(tempa$LC)
tempa$CLT<- as.factor(tempa$CLT)
tempa$Trait<- as.numeric(tempa$Trait)
```

To access the structure of data, the `str()` command can be used.

```
str(tempa)
'data.frame': 400 obs. of 5 variables:
 $ YR   : Factor w/ 2 levels "2009","2010": 1 1 1 1 1 1
1 1 1 1 ...
 $ LC   : Factor w/ 5 levels "CI","FL","KN",..: 3 3 3 3
3 3 3 3 3 3 ...
 $ RP   : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1
1 1 1 1 1 ...
 $ CLT  : Factor w/ 10 levels "CalhounGray",..: 3 1 9 2
5 4 7 10 6 8 ...
 $ Trait: num 56.2 74.2 32.6 74.2 64.8 ...
```

Top 6 rows of example input data can be viewed using `head()` command.

```
head(tempa)
    YR LC RP              CLT  Trait
1 2009 KN  1EarlyCanada 56.236
2 2009 KN  1CalhounGray 74.167
3 2009 KN  1        StarbriteF1 32.601
4 2009 KN  1CrimsonSweet 74.167
5 2009 KN  1GeorgiaRattlesnake 64.794
6 2009 KN  1          FiestaF1 70.907
```

## 2.4. Genotype Stability Statistics

### 2.4.1. Analysis of Variance (ANOVA)

In multi-location replicated trial data, combined ANOVA is performed with the objectives to identify the significance of different effects; estimate and compare mean for levels of fixed factors; and estimate the size of genotype and GxE variance components. The ANOVA model comprises four factors: genotype (CLT), location (LC), year (YR), and replication or block (RP) nested within locations and year. The response of the genotype *i* in the location *j*, year *k* and replication *r* is presented as:

$$\begin{aligned}\text{Response} = {} & m + CLT_i + LC_j + YC_k + RP_r\left(LC_j * YR_k\right) + CLT_i * LC_j \\ & + CLT_i * YR_k + LC_j * YR_k + CLT_i * LC_j * YR_k + \text{Error}_{ijkr}\end{aligned}$$

where $m$ = grand mean. Depending on the objectives of the analysis, the genotype, location and year are defined as random or fixed effect, which gives five different ANOVA models (Table 1). The genotype is random when the aim is to estimate variance components, genetic parameters, genetic gains expected from selection or different breeding strategies etc. Conversely, genotype is fixed factor when aim is to make comparison of test material for selection or recommendation. Similarly, location is considered as random when the main interest is to estimate variance components for sites that are representative of the relevant population within target region. Location is fixed when interest is to make explicit comparison of one level another and each location represents a well-defined area with relative to crop management. The year and replication are usually treated as random factor.

Different combinations of random and fixed effects in ANOVA model have implications for the expectations of mean square (MS) values with the possible modification of the error term to be adopted in the $F$ test. Therefore, sometimes the $F$ test is not as straightforward as the ratio between two mean squares.

RGxE computes five different cases of ANOVA:

- case 1: CLT, YR, LC and RP–all random
- case 2:CLT, YR and LC – fixed; RP–random
- case 3:CLT–fixed; LC, YR and RP–random
- case 4: LC–fixed; CLT, YR and RP–random
- case 5: CLT and LC–fixed; YR and RP–random

For random effect RGxE computes estimates of variance components using `lmer()` function of **lme4** package. The significance of random effects is computed using likelihood ratio test to attain p-values. Likelihood is the probability of the data given a model. The logic of the likelihood ratio test is to compare the likelihood of two models with each other. The model *without* the factor that you are interested in (null model) is compared with model *with* the factor that you are interested in (full model) using `anova()` function. It gives a Chi-Square

Table 1. ANOVA models including the factors genotype (CLT), location (LC), year (YR), and replication (RP) for multi-location replicated trials across years in a randomized complete block design.

| Source of variation | DF | Fixed vs. random effects | | | | |
|---|---|---|---|---|---|---|
| | | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 |
| CLT | $g-1$ | **Random** | **Fixed** | **Fixed** | **Random** | **Fixed** |
| LC | $l-1$ | **Random** | **Fixed** | **Random** | **Fixed** | **Fixed** |
| YR | $y-1$ | **Random** | **Fixed** | **Random** | **Random** | **Random** |
| RP(LC*YR) | $(r-1)ly$ | Random | Random | Random | Random | Random |
| CLT*LC | $(g-1)(l-1)$ | Random | Fixed | Random | Random | Fixed |
| CLT*YR | $(g-1)(y-1)$ | Random | Fixed | Random | Random | Random |
| LC*YR | $(l-1)(y-1)$ | Random | Fixed | Random | Random | Random |
| CLT*LC*YR | $(g-1)(l-1)(y-1)$ | Random | Fixed | Random | Random | Random |
| Pooled error | $(r-1)(g-1)ly$ | | | | | |

value, the associated degrees of freedom and p-value. According to Wilk's theorem, the negative two times the log likelihood ratio of two models approaches a Chi-Square distribution with k degrees of freedom, where k is number of random effects tested. RGxE create user defined `anova_lrt()` function to compute likelihood ratio test and it is stored in ANOVA model Case I code.

The type III sum of squares (SS), MS, *F*value of fixed effects are computed by fitting model in `anova()` function of **lme4** package. The significance (p-value) of fixed effects is computed using `mixed()` function of **afex** package. The `mixed()` function computes type III like p-values using default method via Kenward-Roger approximation for degrees of freedom.

To identify each experimental unit (EU) uniquely a distinct value must be assigned to EU. RGxE assign a distinct value to each combination of replication (RP) nested within location (LC) x year (YR) and use this new term (`RPid`) in model. After installing and calling packages, user can independently compute five different ANOVA models while feeding input data (`tempa`) in below code. User friendly output is generated in "`data.frame`" class using **dplyr** and **tidyr** packages.

```
  #####################################################
###################
  ##                ANOVA: Compute analysis of variance
##
  #####################################################
###################
  #Generate unique id for replication for anova
  tempa$RPid<-as.factor(paste(tempa$YR,  tempa$LC,  tem-
pa$RP, sep="."))
  #####################################################
###################
  ###       ANOVA Case 1: CLT, YR, LC and RP - All Random
###
  #####################################################
###################
  #full model
  fit.f1<-lmer(Trait~ 1 + (1|YR)  + (1|LC) + (1|CLT) +
(1|YR:LC) +
          (1|YR:CLT) + (1|LC:CLT) + (1|YR:LC:CLT) +
          (1|RPid), data=tempa)
  #model summary
  summary1 <- summary(fit.f1)
  #variance of random factors
  variance<- as.data.frame(summary1$varcor)
  #drop rownames
  rownames(variance) <- NULL
  variance1 <- variance %>% select (-var1, -var2) %>%
  rename(sov=grp, Variance=vcov, stddev=sdcor)
  #Type 3  test of hypothesis
  #Type III Wald chisquare tests
  anova(fit.f1, type="III")
  #Type 1  test of hypothesis
  anova(fit.f1, type="marginal", test="F")
  #model fitness
  anovacase1 <- plot(fit.f1,
  main="Model fitness Case 1: CLT, YR, LC and RP - All
Random", xlab="Predicated Value", ylab="Residual")
  #LRT - likelihood ratio test for computing significance
of random effect
  #create function (anova_lrt) for Likelihood ratio test,
where parameters
  #a=outputdatasetname; example-anova1r
  #b=full model name; example-fit.f1
  #c=reduced model name; example-fit.f1r
  #d=effect  name;  example-  "RPid",  NOTE:  call  it  in
quotation
  anova_lrt<- function (a,b,c,d){
  #level of significance
    a <-anova(b,c)
  #convert anova into data frame
    a <- data.frame(a)
  #convert rownames into column
  a$name<- rownames(a)
  # droprownames
  rownames(a) <- NULL
    a <- a %>% filter(name=="b") %>%
  mutate(sov=d) %>% select(sov, Pr_Chisq =
starts_with("Pr..Chisq."))
    # return the result
  return(a)
  }
  #null model for YR
  fit.f1y<-lmer(Trait~ 1 + (1|LC) + (1|CLT) + (1|YR:LC) +
(1|YR:CLT) +
               (1|LC:CLT)  +  (1|YR:LC:CLT)  +  (1|RPid),
data=tempa)
  #level of significance
  #call function anova_lrt
  anova1y <- anova_lrt(anova1y,fit.f1,fit.f1y,"YR")
  #null model for LC
  fit.f1l<-lmer(Trait~ 1 + (1|YR) + (1|CLT) + (1|YR:LC) +
(1|YR:CLT) +
               (1|LC:CLT)  +  (1|YR:LC:CLT)  +  (1|RPid),
data=tempa)
  #level of significance
  #call function anova_lrt
  anova1l <- anova_lrt(anova1l,fit.f1,fit.f1l,"LC")
  #null model for CLT
  fit.f1c<-lmer(Trait~ 1 + (1|YR) + (1|LC) + (1|YR:LC) +
(1|YR:CLT) +
               (1|LC:CLT)  +  (1|YR:LC:CLT)  +  (1|RPid),
data=tempa)
  #level of significance
  #call function anova_lrt
  anova1c <- anova_lrt(anova1c,fit.f1,fit.f1c,"CLT")
  #null model for YR:LC
  fit.f1yl<-lmer(Trait~ 1 + (1|YR) + (1|LC) + (1|CLT) +
(1|YR:CLT) +
                (1|LC:CLT)  +  (1|YR:LC:CLT)  +  (1|RPid),
data=tempa)
  #level of significance
  #call function anova_lrt
  anova1yl <- anova_lrt(anova1yl,fit.f1,fit.f1yl,"YR:LC")
  #null model for YR:CLT
  fit.f1yc<-lmer(Trait~ 1 + (1|YR)  + (1|LC) + (1|CLT) +
(1|YR:LC) +
                (1|LC:CLT) + (1|YR:LC:CLT) +  (1|RPid),
data=tempa)
  #level of significance
  #call function anova_lrt
  anova1yc <- anova_lrt(anova1yc,fit.f1,fit.f1yc,"YR:CLT")
  #null model for LC:CLT
  fit.f1lc<-lmer(Trait~ 1 + (1|YR)  + (1|LC) + (1|CLT) +
(1|YR:LC) +
                (1|YR:CLT) +
                (1|YR:LC:CLT) + (1|RPid), data=tempa)
  #level of significance
  #call function anova_lrt
  anova1lc <- anova_lrt(anova1lc,fit.f1,fit.f1lc,"LC:CLT")
  #null model for YR:LC:CLT
  fit.f1ylc<-lmer(Trait~ 1 + (1|YR)  + (1|LC) + (1|CLT) +
(1|YR:LC) +
                 (1|YR:CLT) +
                 (1|LC:CLT) + (1|RPid), data=tempa)
  #level of significance
  #call function anova_lrt
  anova1ylc <-
anova_lrt(anova1ylc,fit.f1,fit.f1ylc,"YR:LC:CLT")
  #null model for RP
  fit.f1r<-lmer(Trait~ 1 + (1|YR) + (1|LC) + (1|CLT) +
(1|YR:LC) +
                  (1|YR:CLT) +
                  (1|LC:CLT) + (1|YR:LC:CLT), data=tempa)
  #level of significance
  #call function anova_lrt
  anova1r <- anova_lrt(anova1ylr,fit.f1,fit.f1r,"RPid")
  #Merge anova and level of significance
  anova1 <- bind_rows(anova1y, anova1l)%>%
bind_rows(anova1c)%>%
  bind_rows(anova1yl)%>% bind_rows(anova1yc)%>%
  bind_rows(anova1r)%>%
bind_rows(anova1lc)%>%bind_rows(anova1ylc)
  anova1 <- as.data.frame(anova1)
  #Merge final output
  anova_randall<-  variance1%>%  left_join(anova1  ,  by
="sov")
  anova_randall$Pr_Chisq[anova_randall$stddev == 0] <- NA
  #Print final output
  print(anova_randall)
  sov  VariancestddevPr_Chisq
   YR:LC:CLT  49.72994  7.051946 8.894184e-03
      LC:CLT   0.00000  0.000000           NA
  RPid  73.91368  8.597306 4.145811e-07
      YR:CLT   0.00000  0.000000           NA
       YR:LC   57.81311  7.603494 7.872463e-02
         CLT  111.69687 10.568674 1.386709e-03
          LC  699.56950 26.449376 9.083568e-03
          YR    0.00000  0.000000           NA
    Residual 327.52638 18.097690           NA
```

Where `sov` = source of variance, `stddev` = standard deviation, `Pr_Chisq` = Chi-Square probability

In this example, the estimate of variance of random effects location x genotype (`LC:CLT`), year x genotype (`YR:LC`) and year (`YR`) is zero. It represent overfitted model, meaning model is more complex than the data can support. Random effect variance estimated as zero is common with those random effects that have too few or small number of levels. The alternate option is to use Markov Chain Monte Carlo (MCMC) simulation using **MCMCglmm** package to get probability of random effects.

Fitness of ANOVA model for case 1 can be plotted using command `print(anovacase1)`, where x-axis is model predicted value and y-axis is residual value (Figure 2). The uniform distribution of fitted residuals on both side of the reference line (value = 0) confirms the goodness of fit.

The best linear unbiased predictor (BLUP) of random effects can be extracted using `ranef()` function of **lme4** package. BLUPs are estimates of random effects. They allow us to account environmental factors in our model and missing data; and can be used for making selection. BLUP tend to "shrunk" towards the population mean relative to their fixed effects estimates. RGxE computes BLUP of individual genotypes and generate user friendly output in "`data.frame`" class using **dplyr** and **tidyr** packages (see below code).

```
  #Compute BLUP for CLT
  #BLUP - Best linear unbiased predictor
  randeffect1 <- ranef(fit.f1)
  #BLUP for clt
  BLUP_CLT <- as.data.frame(randeffect1$CLT)
  #convert rownames into column
  BLUP_CLT$genotype<- rownames(BLUP_CLT)
  #drop rownames
  rownames(BLUP_CLT) <- NULL
  #rename variable name
  BLUP_CLT <- BLUP_CLT %>% select(genotype,Blup =
starts_with("(Intercept)"))
  #return estimate of fixed effect from full model summary
to compute BLUP
  fixestimate1 <- as.data.frame(summary1$coefficients)
  #compute BLUP value
  BLUP_CLT1 <- BLUP_CLT %>%
```

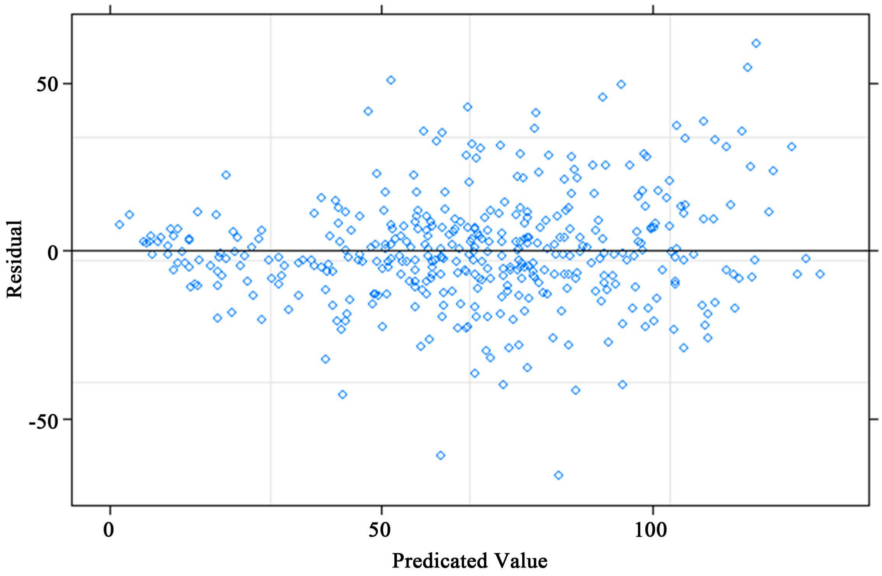

Figure 2. Residual plot for Case 1 of ANOVA model where genotype, location, year and replication are treated as random effect.

```
mutate(Blup = Blup + fixestimate1$Estimate)
#final output for BLUP for cultivars
BLUP_CLT1 <- as.data.frame(BLUP_CLT1)
#Print final output
print(BLUP_CLT1)
genotypeBlup
CalhounGray 76.255
CrimsonSweet 62.535
EarlyCanada 59.840
          FiestaF1 77.283
GeorgiaRattlesnake 70.710
            Legacy 68.417
Mickylee 61.300
Quetzali 58.744
        StarbriteF1 83.422
SugarBaby 51.169
```

Where BLUP = best linear unbiased predictor

Similarly, remaining four cases of ANOVA model can be independently computed using code presented in Supplemental Material.

### 2.4.2. Descriptive Statistics

A new additional variable environment (ENV) is created and quality check of missing value is performed in dataset "tempa2" using **dplyr** package. Environment is location-year combination, which is highlighted is below code. RGxE validates structure of dataset "tempa2" as it serves input data for descriptive and other statistics (Figure 1).

```
#Compute environment - Location by year combination
tempa2 <- tempa %>%
mutate (ENV = paste(LC,YR, sep='-')) %>%
  #remove missing records
na.omit()
#validate data
tempa2$YR <- as.factor(tempa2$YR)
tempa2$RP <- as.factor(tempa2$RP)
tempa2$LC <- as.factor(tempa2$LC)
tempa2$CLT <- as.factor(tempa2$CLT)
tempa2$ENV <- as.factor(tempa2$ENV)
tempa2$Trait <- as.numeric(tempa2$Trait)
```

Top 6 rows of input data 'tempa2' can be viewed using `head()` command.

```
head(tempa2)
   YR LC RP              CLT  TraitRPid    ENV
1 2009 KN  1EarlyCanada 56.236 2009.KN.1 KN-2009
2 2009 KN  1CalhounGray 74.167 2009.KN.1 KN-2009
3 2009 KN  1       StarbriteF1 32.601 2009.KN.1 KN-2009
4 2009 KN  1CrimsonSweet 74.167 2009.KN.1 KN-2009
```

```
5 2009 KN  1GeorgiaRattlesnake 64.794 2009.KN.1 KN-2009
6 2009 KN  1          FiestaF1 70.907 2009.KN.1 KN-2009
```

Descriptive statistics including count, minimum (min), maximum (max), mean, sum, median, variance (var), standard deviation (sd), and coefficient of variation (cv) are computed using **dplyr** package. Using **tidyr** package results of descriptive statistics are transposed in user friendly layout so that researchers can interpret them easily (see Supplemental Material for descriptive statistics outputs). RGxE generates following descriptive statistics.

- Trait mean over genotype and environment
- Trait mean and sd over genotype and year
- Trait mean, cv, sd and sum over genotype and location
- Trait mean, sd, and sum over genotype, location and year
- Trait mean over genotype, location and replication
- Trait mean over location and year
- Trait mean over location and replication
- Trait count, min, max, mean, sum, median, var, and sd over location
- Trait count, min, max, mean, sum, median, var, and sd over year
- Trait count, min, max, mean, sum, median, var, and sd over genotype
- Trait count, min, max, mean, sum, median, var, and sd over environment

### 2.4.3. Univariate Stability Statistics

Among univariate stability statistics, RGxE generates output of regression slope ($b_i$), deviation from regression ($S^2_d$), Shukla's sigma ($\sigma_i^2$), ssquares, Wricke's ecovalence ($W_i$) and Kang's statistics ($YS_i$). RGxE regresses the response of genotype against the environmental index to compute regression slope ($b_i$), deviation from regression ($S^2_d$), $T$-test on regression slope ($H_0$: $b_i = 1$) and $F$-test on deviation from regression ($H_0$: $S^2_d = 0$). The level of significance of $T$-test and $F$-test is computed using `lm()` function of R [19], and **dplyr** and **tidyr** packages. The level of significance at 0.05, 0.01 and 0.001 is represented by "*", "**", "***"; respectively. Environmental index is average performance of all genotypes in each environment. `Stability.par()` function of **agricolae** package is used to compute Shukla's sigma ($\sigma_i^2$), ssquares, Wricke's ecovalence ($W_i$) and Kang's statistics ($YS_i$). For selection of stable genotype, user can independently compute univariate stability statistics while feeding required input data (`tempa2`) in below code. Top 6 rows of input data "`tempa2`" is presented in section 4.2 descriptive statistics.

```
  ##################################################################################
  ##   Compute univariate stability statistics - regression analysis   ##
  ##################################################################################
  #Compute regression (slope) and deviation from regression
  #compute environmental index
  dsterm<- tempa2 %>%
  group_by (ENV, RP, YR, LC) %>%
  summarize (ENVTrait = mean(Trait,na.rm=FALSE))
  dst02 <- tempa2 %>%
  left_join(dsterm, by=c("ENV", "RP")) %>% #Left join on
multiple columns
  arrange (CLT) %>%
  rename (YR= YR.x, LC = LC.x )
  #fit model
  fit_model<- dst02 %>%
  group_by(CLT) %>% #group regression analysis by cultivar
  do (model=lm(Trait~ENVTrait + ENV + RP, data=.))
  #parameter estimates
  paramlm<- as.data.frame(fit_model %>% tidy(model))
  glancelm<- as.data.frame(fit_model %>% glance(model))
  augmentlm<- as.data.frame(fit_model %>% augment(model))
  outmsed<- lapply(fit_model$model, anova) #anova output
  outmsed2 <- as.data.frame(do.call(rbind, outmsed))
#convert list into data.frame
  #convert rownames into column
  outmsed2$SOV <- rownames(outmsed2)
  # droprownames
  rownames(outmsed2) <- NULL
  #remove numeric values from string of rownames using
function gsub
  outmsed2 <- outmsed2 %>% mutate(SOV = gsub("\\d+","",SOV))
  #extract unique cultivar name and merge to outmsed2
dataset
  genotypes<- dst02 %>% select(CLT) %>% distinct (CLT) %>%
arrange(CLT)
  #Stack 4 times to match number of rows with outmsed2
dataset
  genotypes1 <- genotypes %>%bind_rows(genotypes) %>%
  bind_rows(genotypes) %>% bind_rows(genotypes) %>%
arrange(CLT)
  #attach list of cultivars to outmsed2
  outmsed3 <- as.data.frame(outmsed2 %>%
bind_cols(genotypes1))
  #transpose outmsed3
  outmsed4 <- outmsed3 %>%
  select (CLT, SOV, MS = starts_with("Mean")) %>%  #rename
variables
  filter (SOV != "RP")
  #transpose MS values
  MSDS <- outmsed4 %>%
  spread (SOV, MS) %>% #transpose using library tidyr
  arrange (CLT)
  #Transpose degress of freedoms for F-test
  FDS3 <- outmsed3 %>%
  filter (SOV != "RP")%>%
  select (CLT, SOV, Df) %>%
  spread (SOV, Df) %>% #transpose using library tidyr
  rename (DF_ENVTrait = ENVTrait, DF_Residuals = Residuals,
          DF_ENV = ENV)
  #Subset parameters - paramlm dataset
  REGCOEFGS <- paramlm %>%
  filter (term == "ENVTrait") %>%
  select (-statistic, -p.value)
  #Test  and  level  of  significance  of  regression  and deviation from regression
  #Merge MSDS, FDS3, REGCOEFGS
  slope<- MSDS %>% inner_join (REGCOEFGS, by = "CLT") %>%
  inner_join (FDS3, by = "CLT") %>%
  rename (MSE = Residuals, LREGMS=ENVTrait, DEVLMS = ENV,
          BI= estimate, STDERR = std.error)
  #test significance levels
  slope1 <- slope %>%
  mutate (T_H01 = (BI-1)/STDERR , #Null Hypothesis for slope = 1
          PT_H01 = 2*pt(-abs(T_H01), DF_Residuals),
          F_DEVREG=DEVLMS/MSE, #NULL HYPOTHESIS: PREDICTED-ACTUAL = 0
          PF_HO0= 1-pf(F_DEVREG, DF_ENV, DF_Residuals))
  #add legend for level of significance
  slope2 <- slope1 %>%
  mutate (SIG_SLOPE = ifelse(PT_H01 <= 0.001, "***",
  ifelse(PT_H01 <= 0.01, "**",
  ifelse(PT_H01 <= 0.05, "*","")))) %>%
  mutate (SIG_DEVREG = ifelse(PF_HO0 <= 0.001, "***",
  ifelse(PF_HO0 <= 0.01, "**",
  ifelse(PF_HO0 <= 0.05, "*",""))))
  #final regression output
  options(digits=5)
  univariate2 <- slope2 %>%
  mutate (SLOPE = paste(BI,SIG_SLOPE, sep=""),
          DEVREG = paste(DEVLMS,SIG_DEVREG, sep="") ) %>%
  select (CLT,SLOPE, DEVREG )
  #####################################################
###################
  #   Compute univariate stability statistics - shukla,
ecovalence, YS   #
  #####################################################
###################
  #Compute Shukla, WrickeEcovalense, Kangs YS
  repno<- tempa2 %>%
  summarise (total_rep = n_distinct(RP)) #count total
number of rep
  dstgl<- tempa2 %>%
  group_by (CLT, LC) %>%
    #Summarize genotype performance across locations
  summarize (Trait = mean(Trait,na.rm=FALSE))
  dstgl1 <- dstgl %>%
  spread (LC, Trait) #transpose values
  #convert into data frame so that row containing structure
information is deleted
  dstgl2 <- as.data.frame(dstgl1)
  #create rownames
  rownames(dstgl2) <- dstgl2[ ,1]
  shukla<- dstgl2[,-1]
  #compute MS error term
  tempa3 <- glm(Trait ~ LC + YR + LC:YR + RP %in% (LC:YR)
+ CLT + CLT:LC +
                CLT:YR + CLT:LC:YR, family = gaussian ,
data= tempa2 )
  #model summary
  summary1 <- summary.glm(tempa3)
  #Error SS
  error_ss<- as.data.frame(summary1$deviance)
  error_ss1 <- error_ss %>%
    #rename variable
  select  (Deviance  =  starts_with("summary"),  every-
thing())
  #Error DF
  error_df<- as.data.frame(summary1$df.residual)
  error_df1 <- error_df %>%
    #rename variable
  select (Df = starts_with("summary"), everything())
  #MS of error
  mse1 <- as.data.frame(error_ss1/error_df1)
  mse<- mse1 %>%
    #rename variable
  rename (MS = Deviance)
  # MSError is used populated from ANOVA
  univariate1a <- stability.par(shukla, rep=
repno$total_rep , MSerror=mse$MS,
  alpha=0.1, main="Genotype")
  #pool results into individual columns
  univariate1b <- univariate1a$statistics
  #create column genotype from rownames
  univariate1b$genotype <- rownames(univariate1b)
  rownames(univariate1b) = NULL #remove rownames
  names(univariate1b)  [3]  <-  "significane_sigma"     #
rename duplicate name dot
  names(univariate1b) [5] <- "significane_s2"   # rename
duplicate name dot
  names(univariate1b) [2] <- "sigma"   # rename
  names(univariate1b) [4] <- "ssquare"   # rename
  univariate1c <- univariate1a$stability
  #create column genotype from rownames
  univariate1c$genotype <- rownames(univariate1c)
  rownames(univariate1c) = NULL #remove rownames
  names(univariate1c) [8] <- "legend"   # rename varia-
ble ... to legend
  #Merge
  univariate1d <- univariate1b %>%
  inner_join (univariate1c , by = "genotype") %>%
   # deselect all columns between Yield and Stab.rating
  select (-Yield: -Stab.rating) %>%
   # arrange the column order for final output
  select (CLT=genotype, Mean, sigma,
  significane_sigma, ssquare,
          significane_s2, Ecovalence,YSi, legend)
  #Final stability statistics
  #Merge Univariate2 and Univariate1d
  univariate<- univariate2 %>%
  inner_join(univariate1d, by = "CLT") %>%
  mutate (SIGMA=paste(sigma,significane_sigma, ""),
          SIGMA_SQUARE=paste(ssquare,significane_s2,
""),
  YS_Kang =paste(YSi,legend, "")) %>%
  select (Genotype = CLT, Mean, SLOPE, DEVREG, SIGMA,
  SIGMA_SQUARE,Ecovalence, YS_Kang)
  print(univariate)
  Genotype  Mean SLOPE  DEVREG      SIGMA   SIGMA_SQUARE
EcovalenceYS_Kang
```

```
  CalhounGray 77.349 1.32  124.67     61.35 ns   15.76 ns
279.75     10 +
  CrimsonSweet 62.013 1.36  1450.04*** 439.12 ns  567.99 ns
1488.64      4
  EarlyCanada 59.000 0.32* 686.25*    253.23ns  285.37 ns
893.77      2
            FiestaF1  78.498  1.58   657.87       300.29  ns
385.99 ns   1044.35     11 +
  GeorgiaRattlesnake  71.151  0.92   220.06       52.21  ns
44.86 ns    250.52      8 +
              Legacy  68.588  1.13   428.07       287.39  ns
262.49 ns   1003.10      7 +
  Mickylee  60.632  0.59   705.48*    188.11 ns  195.13 ns
685.37      3
  Quetzali  57.775  0.97   96.53       82.81 ns   86.10 ns
348.42      1
         StarbriteF1  85.360  1.29   221.14       157.24  ns
78.37 ns    586.61     12 +
  SugarBaby  49.307  0.50*  332.18*    264.19ns  308.43  ns
928.84     -1
```

Where `SLOPE` = regression slope, `DEVREG` = deviation from regression, `SIGMA` = Shukla's sigma, `SIGMA_SQUARE` = ssquares, `Ecovalence` = Wricke's ecovalence, `YS_Kang` = Kang's statistics, `ns` = non-significant, `+` = indicate stable genotype according to Kang's stability statistics

## 2.5. Location Value Statistics

Input data "`tempa2`" is used to calculate genotype *F* ratio across location and environment; correlation of location with average location performance; and location cluster analysis.

### 2.5.1. Genotype *F* Ratio across Location and Environment; and Correlation among Location and Average Location

RGxE computes analysis of variance by location using `lm()` function to get the genotype F values across location. When the mean of all genotypes are equal then the *F* ratio will be close to 1. The high genotype *F* value indicates high discriminating ability for that location. Similarly, Pearson's test of correlation of locations with average location is computed using `cor.test()` function of R built in **stats** package [19]. Function `cor.test()` provide level of significance of correlation and the level of significance at 0.05, 0.01 and 0.001 is represented by "*", "**", "***"; respectively. RGxE generates user friendly output for genotype *F* ratio across location and environment; and correlation of location with average location performance using **dplyr** and **tidyr** packages as shown in below code.

```
  ##########################################################
##################
```

```
  ##     Compute location statistics - genotype F ratio
across      ##
  ##       location and environment; location correlation
##
  #####################################################
#################
  #Location values
  #F-value of genotype across location
  #fit model
  fit_modellc<- tempa2 %>%
  group_by(LC) %>% #group regression analysis by location
  do (model1=lm(Trait~CLT + YR + CLT:YR + RP%in%YR ,
data=.))
  #parameter estimates
  paramlmlc<- as.data.frame(fit_modellc %>% tidy(model1))
  glancelmlc<- as.data.frame(fit_modellc %>%
glance(model1))
  augmentlmlc<- as.data.frame(fit_modellc %>%
augment(model1))
  outmsedlc<-  lapply(fit_modellc$model1,  anova) #anova
output
  #convert list into data.frame
  outmsedlc2 <- as.data.frame(do.call(rbind, outmsedlc))
  #convert rownames into column
  outmsedlc2$SOV <- rownames(outmsedlc2)
  #drop rownames
  rownames(outmsedlc2) <- NULL
  #remove numeric values from string of rownames using
function gsub
  outmsedlc2 <- outmsedlc2 %>% mutate(SOV =
gsub("\\d+","",SOV))
  #extract unique location name and merge to outmsedlc2
dataset
  location<- dst02 %>% select(LC) %>% distinct (LC) %>%
arrange(LC)
  #Extract gentype F value for each location
  locvalue<- outmsedlc2 %>%
  filter(SOV == "CLT") %>%
  select (FRatioGenotype = starts_with("F value")) %>%
  bind_cols (location) %>% select (LC, FRatioGenotype)
  locvalue<- as.data.frame (locvalue)
  #Correlation between location and average location for
each genotype
  #compute genotype mean at each location
  glcmean1 <- tempa2 %>%
  group_by (CLT, LC) %>%
  summarize (glcmean = mean(Trait,na.rm=FALSE)) %>%
  as.data.frame(select (CLT, LC, glcmean))
  #compute genotype mean across all location -average
location
  gmean1 <- tempa2 %>%
  group_by (CLT ) %>%
  summarize (gmean = mean(Trait,na.rm=FALSE)) %>%
  as.data.frame(select (CLT, gmean))
  #merge location mean with average location for each
genotype
  lgmean<- glcmean1 %>%
  left_join(gmean1, by="CLT") %>%
  arrange(LC) %>% select (-CLT)
  #compute correlation with level of significance
  lcgcorr<- lgmean %>%
  group_by(LC) %>%
  do(tidy(cor.test(.$glcmean, .$gmean, method =
c("pearson"))))
  lcgcorr1 <- lcgcorr %>%
  select (LC, Corr_Value = starts_with ("estimate"),
  Pvalue = starts_with("p.value"))
  #post process correlation value
  lcgcorr2 <- lcgcorr1 %>%
  mutate (SIG_CORR = ifelse(Pvalue<= 0.001, "***",
  ifelse(Pvalue<= 0.01, "**",
  ifelse(Pvalue<= 0.05, "*",""))))
  #concatenate p value symbol with correlation value
  lcgcorr3 <- lcgcorr2 %>%
  mutate     (LocCorrelation=paste(Corr_Value,SIG_CORR,
sep="")) %>%
  select (LC, LocCorrelation)
  lcgcorr3 <- as.data.frame(lcgcorr3)
  #Final location value table for output
  #compute location mean
  Locmean<- tempa2 %>%
  group_by (LC ) %>%
  summarize (Trait = mean(Trait,na.rm=FALSE))%>%
  select (LC, Mean = starts_with("Trait"))
  Locmean<- as.data.frame(Locmean)
  #merge all location value outputs for print
  LocationValue<- Locmean %>%
  inner_join (locvalue, by = "LC") %>%
  inner_join(lcgcorr3, by = "LC") %>%
  rename (Location = LC)
  print(LocationValue)
  Location   Mean FRatioGenotypeLocCorrelation
        CI  61.040       4.1804 0.95***
  FL 100.153       2.2579  0.86**
        KN   63.786       4.6964      0.90***
        SC  82.685       6.8813  0.88***
        TX  27.173       2.9966  0.89***
  WhereFRatioGenotype = genotype F ratio, LocCorrelation
= correlation of location with average location
  #####################################################
###################
  ###             F-value of genotype across environmen
##
  #####################################################
###################
  #fit model
  fit_modelen<- tempa2 %>%
  group_by(ENV) %>% #group regression analysis by location
  do (model2=lm(Trait~CLT + RP , data=.))
  #parameter estimates
  paramlmen<- as.data.frame(fit_modelen %>% tidy(model2))
  glancelmen<- as.data.frame(fit_modelen %>%
glance(model2))
  augmentlmen<- as.data.frame(fit_modelen %>%
augment(model2))
  outmseden<-  lapply(fit_modelen$model2,  anova)  #anova
output
  #convert list into data.frame
  outmseden2 <- as.data.frame(do.call(rbind, outmseden))
  #convert rownames into column
  outmseden2$SOV <- rownames(outmseden2)
  # droprownames
  rownames(outmseden2) <- NULL
  #remove numeric values from string of rownames using
function gsub
  outmseden2 <- outmseden2 %>% mutate(SOV =
gsub("\\d+","",SOV))
  #extract unique environment name and merge to outmseden2
dataset
  environment<- dst02 %>% select(ENV) %>% distinct
(ENV) %>% arrange(ENV)
  #Extract gentype F value for each location
```

```
locvalue2 <- outmseden2 %>%
filter(SOV == "CLT") %>%
select (FRatioGenotype = starts_with("F value")) %>%
bind_cols (environment) %>% select (ENV, FRatioGenotype)
locvalue2 <- as.data.frame (locvalue2)
print(locvalue2)
     ENV FRatioGenotype
  CI-2009        2.4015
  CI-2010        3.3665
  FL-2009        2.6914
  FL-2010        1.7231
  KN-2009        1.9999
  KN-2010        6.3971
  SC-2009        2.8729
  SC-2010        8.0454
  TX-2009        2.5003
TX-2010        1.4619
```

Where ENV = environment, FRatio Genotype = genotype *F* ratio

### 2.5.2. Location Cluster Analysis

Hierarchical cluster analysis for location relatedness is computed using `hclust()` function of R built in **stats** package [19]. The arguments passed to `hclust()` function include Euclidean distance computed from `dist()` function and Ward's method. Function `dist()` of R built in **stats** package [19] computes and return the distance matrix between rows of a data matrix. Tree or dendogram of cluster analysis is generated using `plot()` function, of R built in **graphics** package [19], as shown in below code.

```
  #####################################################
###################
  ###              Compute cluster analysis of location
###
  #####################################################
###################
  #location cluster analysis
  #Euclidean distance
  #Ward Hierarchical Clustering
  #trait mean over location
  mean_l<- tempa2 %>%
  group_by (LC ) %>%
  summarize (Trait = mean(Trait,na.rm=FALSE))
  mean_l1 <- as.data.frame(mean_l)
  clusterdata<- mean_l1 %>% select (Trait)
  clusterdata<- na.omit(clusterdata)
  distance<- dist(clusterdata, method = "euclidean") #
```

```
distance matrix
  hcluster<- hclust(d=distance, method="ward.D")
  locationcluster<- plot(hcluster, labels=mean_l1$LC) #
display dendogram
```

## 3. Final Output

After all computation is over, RGxE clears the Console Window of R studio then saves the output using `sink()` function along with the system date and time using `Sys.time()` function of R built in **base** package [19]. RGxE auto saves the output (output name = "`RGxEOutput`") in folder which is defined in-`setwd()` command in the beginning of the program. Program gives user option to save results in .csv (`RGxEOutput.csv`) or .txt (`RGxEOutput.txt`) format. Output, in .txt format, from sample input data generated by RGxE is presented in **Supplemental Material**. Additionally, RGxE prints the output in Console Window of R studio.

## 4. Result Interpretation

Interpretation of univariate stability statistics is presented in **Supplemental Material**. Additionally, studies published on genotype stability [27] and location value [28] used SASGxE program [17], which is equivalent to RGxE program. Similarly, research study on stability of watermelon fruit quality traits used RGxE program [29]. Thus, these studies can serve as source of RGxE output interpretation. Also, interpretation of RGxE and SASGxE program is available at available at http://cuke.hort.ncsu.edu/cucurbit/wehner/software.html.

## References

[1] Dia, M., Weindorf, D., Thompson, C., Cummings, H., Cacovean, H. and Rusu, T. (2009) Spatial Distribution of Heavy Metals in the Soils of Erath County, Texas. *Studia Universitatis Babes-Bolyai, Geographia*, No. 2, 99-114.

[2] Li, Y., Gibson, J.M., Jat, P., Puggioni, G., Hasan, M., West, J.J., Vizuete, W., Sexton, K. and Serre, M. (2010) Burden of Diseases Attributed to Anthropogenic Air Pollution in the United Arab Emirates: Estimates Based on Observed Air Quality Data. *Science of the Total Environment*, **408**, 5784-5793. https://doi.org/10.1016/j.scitotenv.2010.08.017

[3] Jat, P. and Serre, M.L. (2016) Bayesian Maximum Entropy Space/Time Estimation of Surface Water Chloride in Maryland Using River Distances. *Environmental Pollution*, **219**, 1148-1155. https://doi.org/10.1016/j.envpol.2016.09.020

[4] Weindorf, D.C., Sarkar, R., Dia, M., Wang, H., Chang, Q., Haggard, B., McWhirt, A. and Wooten, A. (2008) Correlation of X-Ray Fluorescence Spectrometry and Inductively Coupled Plasma Atomic Emission Spectroscopy for Elemental Determination in Composted Products. *Compost Science & Utilization*, **16**, 79-82. https://doi.org/10.1080/1065657X.2008.10702361

[5] Weindorf, D., Rinard, B., Zhu, Y., Johnson, S., Haggard, B., McPherson, J., Dia, M., Spinks, C. and McWhirt, A. (2008) High Resolution Soil Survey of Capulin Volcano National Monument, New Mexico. *Soil Horizons*, **49**, 55-62. https://doi.org/10.2136/sh2008.3.0055

[6] Dia, M., Wehner, T.C., Hassell, R., Price, D.S., Boyhan, G.E., Olson, S., King, S., Davis, A.R., Tolla, G.E., Bernier, J., Juarez, B., Sari, N., Solmaz, I. and Aras, V. (2012) Stability of Fruit Yield in Watermelon Genotypes Tested in Multiple US Environments. *Proceedings of the Xth EUCARPIA Meeting on Genetics and Breeding of Cucurbitaceae*, Antalya, Turkey, 15-18 October 2012, 84-88.

[7] Dia, M., Wehner, T.C., Hassell, R., Price, D.S., Boyhan, G.E., Olson, S., King, S., Davis, A.R., Tolla, G.E., Bernier, J., Juarez, B., Sari, N., Solmaz, I. and Aras V.(2012) Mega-Environment Identification for Watermelon Yield Testing in the *USProceedings of the Xth EUCARPIA Meeting on Genetics and Breeding of Cucurbitaceae*, Antalya, Turkey, 15-18 October 2012, 385-390.

[8] Kumar, R., Dia, M. and Wehner, T.C. (2013) Implications of Mating Behavior in Watermelon Breeding. *HortScience*, **48**, 960-964.

[9] Yan, W., Pageau, D., Frégeau-Reid, J. and Durand, J. (2011) Assessing the Representativeness and Repeatability of Test Locations for Genotype Evaluation. *Crop Science*, **51**, 1603-1610. https://doi.org/10.2135/cropsci2011.01.0016

[10] Smith, A.B., Cullis, B.R. and Thompson, R. (2005) The Analysis of Crop Cultivar Breeding and Evaluation Trials: An Overview of Current Mixed Model Approaches. *Journal of Agricultural Science*, **143**, 449-462. https://doi.org/10.1017/S0021859605005587

[11] Wricke, G. (1962) Evaluation Method for Recording Ecological Differences in Field Trials. *Z Pflanzenzücht*, **47**, 92-96.

[12] Shukla, G.K. (1972) Genotype Stability Analysis and Its Application to Potato Regional Trails. *Crop Science*, **11**, 184-190.

[13] Kang, M.S., Miller, J.D. and Darrah, L.L. (1987) A Note on Relationship between Stability Variance and Ecovalence. *Journal of Heredity*, **78**, 107. https://doi.org/10.1093/oxfordjournals.jhered.a110322

[14] Kang, M.S. (1993) Simultaneous Selection for Yield and Stability in Crop Performance Trials: Consequences for Growers. *Agronomy Journal*, **85**, 754-757. https://doi.org/10.2134/agronj1993.00021962008500030042x

[15] Mekbib, F. (2003) Yield Stability in Common Beans (*Phaseolus vulgaris* L.) Genotypes. *Euphytica*, **130**, 147-153.

[16] Fan, X.M., Kang, M., Chen, H., Zhang, Y., Tan, J. and Xu, C. (2007) Yield Stability of Maize Hybrids Evaluated in Multi-Environment Trials in Yunnan, China. *Agronomy Journal*, **99**, 220-228. https://doi.org/10.2134/agronj2006.0144

[17] Dia, M., Wehner, T.C. and Arellano, C. (2016) Analysis of Genotype × Environment Interaction (G × E) Using SAS Programming. *Agronomy Journal*, **108**, 1838-1852. https://doi.org/10.2134/agronj2016.02.0085

[18] RStudio Team (2015) RStudio: Integrated Development for R. RStudio, Inc., Boston, MA (Computer Software v0.98.1074). http://www.rstudio.com/

[19] R Core Team (2015) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org/

[20] Wickham, H. and Francois, R. (2015) dplyr: A Grammar of Data Manipulation. R Package Version 0.4.3. http://CRAN.R-project.org/package=dplyr

[21] Wickham, H. (2016) tidyr: Easily Tidy Data with 'spread()' and 'gather()' Functions. R package Version 0.4.1. http://CRAN.R-project.org/package=tidyr

[22] Robinson, D. (2015) broom: Convert Statistical Analysis Objects into Tidy Data Frames. R Package Version 0.4.0. http://CRAN.R-project.org/package=broom

[23] Mendiburu, F.D. (2015) agricolae: Statistical Procedures for Agricultural Research.

R Package Version 1.2-3. http://CRAN.R-project.org/package=agricolae

[24] Bates, D., Maechler, M., Bolker, B. and Walker, S. (2015) Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, **67**, 1-48. https://doi.org/10.18637/jss.v067.i01

[25] Singmann, H., Bolker, B. and Westfall, J. (2015) afex: Analysis of Factorial Experiments. R Package Version 0.15-2. http://CRAN.R-project.org/package=afex

[26] Maechler, M., Rousseeuw, P., Struyt, A. and Hubert, M. (2015) cluster: Cluster Analysis Basics and Extensions. R Package Version 2.0.1.

[27] Dia, M., Wehner, T.C., Hassell, R., Price, D.S., Boyhan, G.E., Olson, S., King, S., Davis, A.R. andTolla, G.E. (2016) Genotype × Environment Interaction and Stability Analysis for Watermelon Fruit Yield in the U.S. *Crop Science*, **56**, 1645-1661. https://doi.org/10.2135/cropsci2015.10.0625

[28] Dia, M., Wehner, T.C., Hassell, R., Price, D.S., Boyhan, G.E., Olson, S., King S., Davis, A.R. and Tolla, G.E. (2016) Values of Locations for Representing Mega-Environments and for Discriminating Yield of Watermelon in the U.S. *Crop Science*, **56**, 1726-1735. https://doi.org/10.2135/cropsci2015.11.0698

[29] Dia, M., Wehner, T.C., Perkins-Veazie, P., Hassell, R., Price, D.S., Boyhan, G.E., Olson, S., King, S., Davis, A.R., Tolla, G.E., Bernier, J. and Juarez, B. (2016) Stability of Fruit Quality Traits in Diverse Watermelon Cultivars Tested in Multiple Environments. *Horticulture Research*, **3**, Article No. 16066. https://doi.org/10.1038/hortres.2016.66

## Supplemental Material

The supplemental material available online includes the RGxE program, instructions for user enetered field needed in RGxE program, independent module of ANOVA model case 2 to 5 (Table 1), interpretation of univariate stability statistics, example input data and output from example input data generated from RGxE program. Additionally, interpretation of univariate and multivariate statistical analysis is provided in [17].

http://cucurbitbreeding.com/wp-content/uploads/2016/05/RGxE17Supplement.pdf

## List of Abbreviations

AMMI = Additive main effects and multiplicative interaction model

ANOVA = Analysis of variance

BLUP = Best linear unbiased prediction

CLT = Cultigen or genotype

CRAN = Comprehensive R Archive Network

CSV = Comma Separated Value

CV = Coefficient of variation

DF = Degrees of freedom

ENV = Environment (location - year combination)

EU = Experimental unit

GGE = Genotype main effects plus Genotype x environmental interaction effect model

GxE = Genotype x environmental interaction

$H_0$ = Null hypothesis

LC = Location

Max = Maximum

MCMC = Markov Chain Monte Carlo

Min = Minimum

MS = Mean square

RGxE = R program for the analysis of genotype stability and location value

RP = Replication

RPid = Replication id, which is an experimental unit

Sd = Standard deviation

SS = Sum of square

Var = Variance

YR = Year

$b_i$ = Regression slope

$S_d^2$ = Deviation from regression

$\sigma_i^2$ = Shukla's variance

$W_i$ = Wricke'secovalence

$YS_i$ = Kang's yield stability