



FACULTAD DE CIENCIAS AGRARIAS
UNIVERSIDAD NACIONAL DE ROSARIO

Paquete de R y aplicación Web Shiny para el análisis de datos
provenientes de ensayos multiambientales

JULIA ANGELINI

TRABAJO FINAL PARA OPTAR AL TÍTULO DE ESPECIALISTA EN
BIOINFORMÁTICA

DIRECTOR: Dr. Gerardo Cervigni
CO-DIRECTOR: Mgs. Marcos Prunello

AÑO: 2021

Paquete de R y aplicación Web para el análisis de datos provenientes de ensayos multiambientales

Julia Angelini

Licenciada en Estadística – Universidad Nacional de Rosario

Este Trabajo Final es presentado como parte de los requisitos para optar al grado académico de Especialista en **Bioinformática**, de la Universidad Nacional de Rosario y no ha sido previamente presentada para la obtención de otro título en ésta u otra Universidad. El mismo contiene los resultados obtenidos en investigaciones llevadas a cabo en el **Centro de Estudios Fotosintéticos y Bioquímicos (CEFOBI)**, durante el período comprendido entre los años **2017 y 2021**, bajo la dirección del **Dr. Gerardo Cervigni** y **Mgs. Marcos Prunello**.

Nombre y firma del autor

Nombre y firma del Director

Nombre y firma del Co - Director

Defendida: _____ de 20____.

Agradecimientos

En este trabajo final, directa o indirectamente, participaron muchas personas a las que les quiero agradecer.

En primer lugar al Dr. Gerardo Cervigni por confiar en mí y permitirme explorar el mundo de la Bioinformática durante mi tesis doctoral, para que hoy sea parte de mis conocimientos. Al Mgs. Marcos Prunello por acompañarme en el desarrollo del trabajo final, por su dedicación y sus consejos.

Todo esto nunca hubiera sido posible sin el apoyo y el cariño de mis padres, de mi hermano, de Otto, de Segundo y Kalita. Siempre estuvieron a mi lado, las palabras nunca serán suficientes para agradecerles.

A mis compañeras Jor y Lu, por su ayuda y por compartir excelentes momentos.

A Gaby y Euge mis compañeras de CEFOBI, gracias a ustedes este camino ha sido más fácil!

A mis amigos, por estar siempre presentes.

Muchas gracias a todos!

Abreviaturas y Símbolos

ACP: análisis de componentes principales.

ANOVA: análisis de la variancia, del inglés *analysis of variance*.

AMMI: modelo de efectos principales aditivos e interacción multiplicativa, del inglés *Additive Main effects and Multiplicative Interaction*.

COI: interacción con cambio de rango, del inglés *crossover interaction*.

CRAN: *Comprehensive R Archve Network*

DVS: descomposición de valores singulares

EMA: ensayos multiambientales.

G: Efecto genotípico.

GE: Genotipo-ambiente, del inglés *Genotype-Environment*.

GGE: Genotipo más Genotipo-ambiente, del inglés *Genotype plus Genotype-Environment*.

IGA: interacción genotipo ambiente.

NCOI: interacción sin cambio de rango, del inglés *no crossover interaction*.

SREG: modelo de regresión por sitio, del inglés *Site Regression model*.

Resumen

Las variedades mejoradas de cultivos vegetales son el resultado del trabajo de desarrollo genético llevado a cabo en los programas de fitomejoramiento, los cuales se extienden a lo largo de varios años y requieren cuantiosas inversiones. En etapas avanzadas, los ensayos multiambientales (EMA), que comprenden experimentos en múltiples ambientes, son herramientas fundamentales para incrementar la productividad y rentabilidad de los cultivos. La vigencia comercial de las variedades puede extenderse durante varias décadas, por lo que su elección es crítica para que el productor evite pérdidas económicas por malas campañas y el suministro al mercado sea constante. Consecuentemente, un análisis adecuado de la información de los EMA es indispensable para asegurar el éxito del programa de mejoramiento de cultivos. Actualmente, R es uno de los lenguajes de programación más utilizados para el análisis de datos debido a su distribución como software libre y a la gran variedad de herramientas que ofrece. Sin embargo, los mejoradores que no están familiarizados con la programación tienden a utilizar otros tipos de programas que responden a instrucciones por menú en lugar de escribir líneas de código, a pesar de los costos económicos derivados del pago de sus licencias. Mientras que, aquellos que sí tienen afinidad con el uso de código para el análisis de datos se enfrentan con dificultades a la hora de identificar las herramientas apropiadas entre el gran número de instrumentos disponibles. Por lo tanto, en este trabajo se presenta el desarrollo de dos herramientas informáticas para asistir en el análisis de datos provenientes de EMA. Por un lado, se creó un nuevo paquete de R que incluye metodología recientemente publicada que no se encuentra disponible en el software y al mismo tiempo reúne todas aquellas de mayor utilidad, de modo que aquellos usuarios que posean un manejo del lenguaje puedan simplificar su tarea. Por otro lado, se confeccionó una interfaz gráfica de usuario mediante una aplicación web Shiny que permite realizar los principales análisis implementados en el paquete sin necesidad de programar y se encuentra publicada en internet para su libre acceso.

Palabras Clave: análisis estadístico, ensayos multiambientales, interfaz gráfica, lenguaje R, programación.

Abstract

Crop improvement is the result of genetic development which requires several years and large investments. In advanced stages of breeding programs, multi-environment trials (MET), which consist of evaluating different cultivars in multiple environments, are essential tools to increase crop productivity. Since varieties remain on market for decades, their choice is essential to avoid economic losses due to bad seasons and to ensure a constant supply. Consequently, an adequate analysis of MET data is essential to guarantee the success of a breeding program. Currently, R is one of the most widely used programming language for data analysis due to its distribution as free software and the wide variety of tools it offers. However, breeders who are unfamiliar with programming tend to use other types of programs that respond to menu prompts instead of writing lines of code, despite the financial costs of their licenses. Whereas, those who have an affinity with the use of code for data analysis face difficulties in identifying the right tools from the large number of instruments available. Therefore, in this work two tools are develop for MET data analysis. On one hand, a new R package that includes new methodology not available in the software and at the same time brings together all those most useful created to facilitate the users task. On the other hand, a graphical user interface was created using a Shiny web application that allows the main analyzes implemented in the package to be carried out without the need for programming and is published on internet for free access.

Keywords: multi-environment trials, programming, statistical analysis, user interfaz, R language.

Índice general

Capítulos	Página
1. Introducción	1
2. Objetivos	5
2.1. Objetivo general	5
2.2. Objetivos específicos	5
3. Métodos	6
3.1. Métodos estadísticos	6
3.1.1. Modelo AMMI y SREG	6
3.1.2. Modelo AMMI robusto	8
3.1.3. Métodos de imputación	9
3.2. Creación de un paquete de R	10
3.2.1. <i>geneticae</i>	10
3.2.2. Estructura general del paquete	11
3.2.3. Archivo DESCRIPTION	12
3.2.4. Archivos de código	13
3.2.5. Documentación	14
3.2.6. Uso de funciones de otros paquetes	15
3.2.7. Testeos	16
3.2.8. Datasets	17
3.2.9. Archivo README	18
3.2.10. Archivo NEWS	18
3.2.11. Viñetas	19

3.2.12. R CMD check e instalación	20
3.2.13. Publicación y difusión	20
3.3. Aplicación web Shiny	21
3.3.1. Estructura de la aplicación web Shiny	21
3.3.2. Desarrollo de la aplicación web Shiny	22
3.3.3. Compartiendo una aplicación web Shiny	23
4. Resultados	24
4.1. Paquete de R <i>geneticae</i>	24
4.1.1. Conjuntos de datos en <i>geneticae</i>	24
4.1.2. Modelo AMMI	25
4.1.3. Modelo de Regresión por Sitio	28
4.1.4. Métodos de imputación	37
4.2. Geneticae Shiny Web App	38
4.2.1. Preparación de un archivo de datos	38
4.2.2. Análisis descriptivo	40
4.2.3. Modelo de regresión por sitio	42
4.2.4. modelo AMMI	44
4.2.5. Ayuda	45
5. Conclusión	46
Bibliografía	47

Índice de figuras

Figura 1.1: Representación gráfica de tipos de interacción genotipo - ambiente: (A) <i>crossover</i> , (B) <i>no crossover</i> y (C) sin interacción.	2
Figura 3.1: Chequeo de disponibilidad del nombre <i>geneticae</i> elegido para el paquete en desarrollo mediante la función <code>available()</code> del paquete <i>available</i>	11
Figura 3.2: Creación del paquete <i>geneticae</i> mediante la función <code>create_package()</code> del paquete <i>usethis</i>	12
Figura 3.4: Fragmento de función <code>GGEmodel()</code> del paquete <i>geneticae</i>	13
Figura 3.3: Archivo DESCRIPTION de <i>geneticae</i>	14
Figura 3.5: Fragmento de los comentarios roxygen de la función <code>GGEmodel()</code> del paquete <i>geneticae</i>	16
Figura 3.6: Resultado de correr la función <code>test()</code> del paquete <i>devtools</i> en <i>geneticae</i>	17
Figura 3.7: Porcentaje del código de <i>geneticae</i> que es evaluado durante los tes- teos obtenido mediante la función <code>test_coverage()</code> del paquete <i>covr</i>	17
Figura 3.8: Fragmento del archivo README mostrado en el repositorio GitHub del paquete <i>geneticae</i>	19
Figura 3.9: Archivo NEWS de <i>geneticae</i>	19
Figura 4.1: Biplot GE obtenido del modelo AMMI clásico basado en los datos de rendimiento de trigo de invierno obtenidos en Ontario en 1993. El 71,66 % de la variabilidad de la IGA se explica por los dos prime- ros términos multiplicativos. Los cultivares se muestran en letras minúsculas y los ambientes en mayúsculas.	27

Figura 4.2: Biplot GGE basado en datos de rendimiento de trigo de invierno obtenido de Ontario en 1993. El método de partición de valores singulares utilizado es el simétrico (opción por defecto). El 78 % de la variabilidad de $G + GE$ se explica por los dos primeros términos multiplicativos. Los cultivares se muestran en minúsculas y los entornos en mayúsculas.	30
Figura 4.3: A: Ranking de cultivares en el ambiente OA93. B: Ranking de ambientes para cultivar Kat, basado en datos de rendimiento de trigo de invierno obtenido de Ontario en 1993. El método de partición de valores singulares utilizado es el simétrico (opción por defecto). El 78 % de la variabilidad de $G + GE$ se explica por los dos primeros términos multiplicativos. Los cultivares se muestran en minúsculas y los entornos en mayúsculas.	31
Figura 4.4: comparación de los cultivares Kat y Cas. El método de partición de valores singulares utilizado es el simétrico (opción por defecto). El 78 % de la variabilidad de $G + GE$ se explica por los dos primeros términos multiplicativos. Los cultivares se muestran en minúsculas y los entornos en mayúsculas.	32
Figura 4.5: Vista poligonal del biplot GGE, que muestra qué cultivares presentaron mayor rendimiento en cada ambiente/mega-ambiente. El método de partición de valores singulares utilizado es el simétrico (opción por defecto). El 78 % de la variabilidad de $G + GE$ se explica por los dos primeros términos multiplicativos. Los cultivares se muestran en minúsculas y los entornos en mayúsculas.	33
Figura 4.6: A: Evaluación de los cultivares con base en el rendimiento promedio y la estabilidad y B: Clasificación de genotipos con respecto al genotipo ideal, basado en el método de partición de la descomposición en valores singulares enfocado en los genotipos.	35
Figura 4.7: A: Relación entre ambientes y B: Clasificación de ambientes con respecto al ambiente ideal, basado en el escalado centrado en los genotipos.	37
Figura 4.8: Conjunto de datos (A) <i>plrv</i> (B) <i>yanwinterwheat</i> disponible en Geneticae Shiny Web APP	39
Figura 4.9: Importar el conjunto de datos <i>yanwinterwheat</i> en Geneticae Shiny Web APP	40

Figura 4.10: Diagrama de caja de (A) genotipos y (B) ambientes para el conjunto de datos <i>yanwinterwheat</i>	41
Figura 4.11: Gráfico de correlación (A) y matriz (B) entre genotipos para conjunto de datos <i>yanwinterwheat</i>	41
Figura 4.12: Gráfico de interacción para (A) ambientes a través de genotipos y (B) genotipos a través de los ambientes para conjunto de datos de <i>yanwinterwheat</i>	42
Figura 4.13: Vistas del biplot GGE usando la partición en valores singulares simétrica.	43
Figura 4.14: Vistas del biplot GGE usando la partición de valores singulares enfocada en los genotipos.	44
Figura 4.15: Vistas del biplot GGE usando la partición de valores singulares enfocada en los ambientes.	44
Figura 4.16: Biplot GE obtenido del modelo AMMI clásico basado en datos de rendimiento de trigo de invierno obtenido de Ontario en 1993. . . .	45

Capítulo 1

Introducción

A lo largo de la historia de la agricultura, el hombre ha desarrollado el mejoramiento vegetal en forma sistemática y lo ha convertido en un instrumento esencial para incrementar la producción agrícola en términos de cantidad, calidad y diversidad. Las variedades mejoradas son el resultado del trabajo llevado a cabo en los programas de fitomejoramiento, los cuales se extienden a lo largo de varios años y requieren cuantiosas inversiones. En etapas avanzadas de estos programas, comúnmente se llevan a cabo ensayos multiambiales (EMA) de comparación de rendimientos, donde un conjunto de variedades se evalúan en múltiples ambientes. Estos son esenciales debido a la presencia de interacción genotipo - ambiente (IGA) la cual es inevitable debido a las variaciones en las condiciones climáticas y de suelo de los distintos ambientes analizados. La IGA es considerada casi unánimemente por los fitomejoradores como el principal factor que limita la selección de cultivares superiores y, en general, afecta la eficiencia de los programas de mejoramiento (Crossa et al., 1990; Cruz Medina, 1992; Kang y Magari, 1996). Cuando los ambientes son muy diferentes, la IGA usualmente gana importancia porque cambia el rango de las líneas de mejoramiento. Gauch y Zobel (1997) explicaron que si no hubiera interacción, una sola variedad o híbrido rendirían al máximo en todo el mundo, además los materiales podrían evaluarse en un solo lugar y proporcionarían resultados universales.

Peto (1982) ha distinguido las interacciones cuantitativas, conocidas también como sin cambio de rango o *no crossover interaction* (NCOI), de las cualitativas, denominadas a su vez como con cambio de rango o *crossover interaction* (COI). Cuando dos genotipos G_1 y G_2 tienen una respuesta diferencial en dos ambientes, se dice que la IGA es del tipo COI si hay cambios en el orden de los genotipos según su rendimiento (Figura 1.1(A)) y del tipo NCOI si su ordenamiento permanece sin cambios (Figura 1.1(B)). Por otro lado, se dice que la IGA es inexistente cuando los genotipos responden de manera similar en ambos ambientes (Figura 1.1(C)).

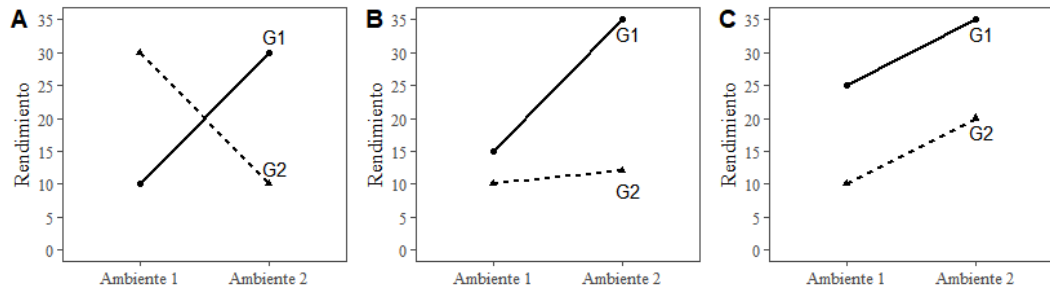


Figura 1.1: Representación gráfica de tipos de interacción genotipo - ambiente: (A) *crossover*, (B) *no crossover* y (C) sin interacción.

Entre las implicancias negativas de la IGA en los programas de mejoramiento vegetal se encuentra el impacto negativo sobre la heredabilidad, cuanto menor sea la heredabilidad de un carácter, mayor será la dificultad para mejorarlo. Conceptos importantes tales como regiones ecológicas, ecotipos, mega-ambientes, adaptación específica y estabilidad se pueden analizar a partir de la IGA (Yan y Hunt, 2001).

Un análisis adecuado de la información de los EMA es indispensable para el éxito del programa de mejoramiento genético de los cultivos. El rendimiento medio en los ambientes es un indicador suficiente del rendimiento genotípico sólo en ausencia de IGA (Yan y Kang, 2003). Sin embargo, la aparición de IGA es inevitable y no basta con la comparación de las medias de los genotipos, sino que se debe recurrir a una metodología estadística más apropiada. Las más difundidas para analizar los datos provenientes de EMA se basan en modificaciones de los modelos de regresión, análisis de variancia (ANOVA) y técnicas de análisis multivariado.

Particularmente, para el estudio de la IGA y los análisis que de ella se derivan, dos modelos multiplicativos han aumentado su popularidad entre los fitomejoradores como herramientas de análisis gráfico: el modelo de los efectos principales aditivos e interacción multiplicativa (AMMI de las siglas en inglés *Additive Main effects and Multiplicative Interaction*) (Kempton, 1984; Gauch, 1988) y el de regresión por sitio (SREG de las siglas en inglés *Site Regression model*) (Cornelius et al., 1996; Crossa y Cornelius, 1997). Estos modelos se ajustan en dos etapas. Primero, se realiza un ANOVA para obtener estimaciones de los efectos principales aditivos de ambientes y genotipos (G) en AMMI y sólo de los ambientes en SREG. En segundo lugar, los residuos del ANOVA se ordenan en una matriz con genotipos en las filas y ambientes en las columnas y se aplica una descomposición de valores singulares (DVS), representando los patrones de IGA presentes en los residuos en AMMI, y de G e IGA conjuntamente en SREG. El resultado de los dos primeros términos multiplicativos de la DVS a menudo se presentan en un biplot llamado

GE (de las siglas en inglés *Genotype-Environment*) para el modelo AMMI y GGE (de las siglas en inglés *Genotype plus Genotype-Environment*) para SREG. Sin embargo, estos modelos no siempre son lo suficientemente eficientes para analizar la estructura de datos provenientes de EMA de programas de mejoramiento vegetal (de Oliveira et al., 2016; Jarquín et al., 2016; Hadasch et al., 2018). Por un lado, tienen serias limitaciones frente a información faltante y, a pesar de que los EMA están diseñados para que todos los genotipos se evalúen en todos los ambientes, la presencia de valores perdidos es muy común (Woyann et al., 2017; Aguate et al., 2019). Esto ocurre, por ejemplo, debido a errores de medición o destrucción de plantas por presencia de animales, inundaciones o problemas durante la cosecha, además de la dinámica propia de las evaluaciones en las que se incorporan y se descartan genotipos debido a su mal desempeño (Hill y Rosenberg, 1985). Numerosas metodologías de imputación se han estado desarrollando en los últimos años para solventar esta limitación (Arciniegas-Alarcón et al., 2010, 2014; Josse y Husson, 2016; Arciniegas-Alarcón et al., 2020). Por otro lado, ambos modelos son sensibles a la presencia de observaciones atípicas, lo cual es una regla más que una excepción cuando se consideran datos reales. Para superar esta fragilidad, recientemente distintas metodologías robustas se han desarrollado para el modelo AMMI (Rodrigues et al., 2016).

En este contexto, el análisis de datos provenientes de EMA requiere metodología estadística cuyas rutinas informáticas no se encuentran disponibles en programas comerciales debido a su reciente desarrollo o bien se deben utilizar varios de ellos para cumplir un único objetivo. Esto último genera el inconveniente de tener que disponer de todos los programas necesarios para los distintos análisis, atender los requerimientos de formatos de datos usados por cada uno de ellos, y comprender los diversos tipos de salidas en las que se presentan los resultados obtenidos. Además, los costos de las licencias algunos programas pueden resultar muy elevados.

Ante estas dificultades, una alternativa para el análisis es el empleo de algún lenguaje de programación de distribución libre y gratuita, que le confiera al analista la flexibilidad necesaria para cumplir con su objetivo. En este contexto, R es uno de los lenguajes de programación desarrollados para el análisis de datos de mayor uso en la actualidad. R es un software de uso libre y distribuido bajo los términos de la *General Public Licence*. Este programa se descarga de un repositorio mantenido por *The R Foundation for Statistical Computing* conocido como CRAN (*Comprehensive R Archive Network*), en el cual también se encuentran disponibles miles de paquetes adicionales que consisten en conjuntos de funciones desarrolladas con fines específicos que se distribuyen con un protocolo determinado, garantizando su correcto funcionamiento. Cualquier desarrollador puede producir su propio paquete y publicarlo en CRAN, siempre que cumpla con los requisitos establecidos y pase correctamente por los procedimientos de control. Además, hay paquetes que

pueden obtenerse de otros repositorios como Github, Bioconductor, rOpenSci, entre otros. R es propicio para el análisis de datos de EMA puesto que se ha desarrollado metodología específica para este entorno computacional.

A pesar de estas ventajas, el análisis de datos de EMA en R presenta algunos desafíos. Por un lado, existen numerosos paquetes con funcionalidad afín que hay que identificar cómo combinar adecuadamente. Por otro lado, el software puede resultar dificultoso para aquellos analistas no familiarizados con la programación. Atendiendo a estas dos necesidades, se crea un paquete que incluya metodología recientemente publicada y reúna las funciones más útiles a fin de solventar la primera de ellas. Para la segunda, se crea una aplicación web Shiny de libre acceso mediante conexión a internet que permita realizar los principales análisis implementados en el paquete sin necesidad de escribir líneas de código.

Capítulo 2

Objetivos

2.1. Objetivo general

Desarrollar un paquete de R para el análisis de datos provenientes de EMA y una interfaz gráfica de usuario para el mismo a través de la aplicación web Shiny.

2.2. Objetivos específicos

- Mostrar un flujo de trabajo reproducible para la construcción de paquetes de R.
- Programar e incluir en el paquete metodología para el análisis de datos provenientes de EMA recientemente publicada y no disponible en R.
- Añadir en el paquete de R funciones ya existentes con modificaciones o agregados para favorecer su uso.
- Desarrollar una aplicación web Shiny que sirva como interfaz gráfica de usuario para el paquete.
- Publicar el paquete y la aplicación web para su libre uso.

Capítulo 3

Métodos

Este capítulo tiene como objetivo focalizar al lector en los aspectos fundamentales sobre los que se apoya este trabajo. Se compone de tres secciones, en la primera se presenta la metodología estadística que se incluirá en el paquete de R. En la segunda y tercera sección, se presenta un flujo de trabajo reproducible para el desarrollo del paquete y la aplicación web Shiny, respectivamente.

3.1. Métodos estadísticos

3.1.1. Modelo AMMI y SREG

Para el estudio de la IGA y los análisis que de ella se derivan, dos modelos multiplicativos han aumentado su popularidad entre los fitomejoradores como herramientas de análisis gráfico: el modelo de los efectos principales aditivos e interacción multiplicativa (AMMI de las siglas en inglés *Additive Main effects and Multiplicative Interaction*) (Kempton, 1984; Gauch, 1988) y el de regresión por sitio (SREG de las siglas en inglés *Site Regression model*) (Cornelius et al., 1996; Crossa y Cornelius, 1997). Estos modelos se ajustan en dos etapas. Primero, se realiza un ANOVA para obtener estimaciones de los efectos principales aditivos de ambientes y genotipos en AMMI y sólo de los ambientes en SREG. En segundo lugar, los residuos del ANOVA se ordenan en una matriz con genotipos en las filas y ambientes en las columnas y se aplica una DVS, representando los patrones de IGA presentes en los residuos en AMMI, y de G e IGA conjuntamente en SREG.

Las ecuaciones de los distintos modelos son:

$$\text{AMMI: } y_{ij} = \mu + G_i + A_j + \sum_{k=1}^K \lambda_k \alpha_{ik} \gamma_{jk}$$

$$\text{SREG: } y_{ij} = \mu + A_j + \sum_{k=1}^K \lambda_k \alpha_{ik} \gamma_{jk}$$

donde

- y_{ij} es el carácter fenotípico evaluado (rendimiento o cualquier otro carácter de interés) del i -ésimo genotipo en el j -ésimo ambiente,
- μ es la media general,
- G_i es el efecto del i -ésimo genotipo con $i = 1, \dots, g$,
- A_j es el efecto del j -ésimo ambiente con $j = 1, \dots, a$,
- $\sum_{k=1}^K \lambda_k \alpha_{ik} \gamma_{jk}$ es la sumatoria de términos multiplicativos utilizadas para modelar la IGA en AMMI o de G e IGA conjuntamente en SREG. K es el número de términos multiplicativos retenidos en el modelo con $K \leq \min(g - 1, a - 1)$ en AMMI y $K \leq \min(g, a - 1)$ en SREG; λ_k es el k -ésimo valor singular y α_{ik} y γ_{jk} son los elementos de los autovectores asociados con el i -ésimo genotipo y el j -ésimo ambiente para el k -ésimo término multiplicativo, respectivamente. En general, los dos primeros términos multiplicativos ($K = 2$) son suficientes para explicar los patrones de la IGA en AMMI y de G e IGA en SREG; la variabilidad remanente se interpreta como ruido aleatorio.

El resultado de los dos primeros términos multiplicativos de la SVD se presenta a menudo en un biplot llamado GE (de las siglas en inglés *Genotype-Environment*) para el modelo AMMI y GGE (de las siglas en inglés *Genotype plus Genotype-Environment*) en SREG y representan una aproximación de dos rangos de los efectos multiplicativos. Dado que para seleccionar cultivares el G e IGA debe considerarse simultáneamente, el modelo SREG resulta superior a AMMI para visualizar patrones en datos provenientes de EMA. Un biplot GGE que explica suficiente variabilidad debida a G e IGA de un conjunto de datos provenientes de EMA permite, entre otras cosas, visualizar tres aspectos importantes:

- (i) las relaciones entre los genotipos y ambientes representadas por el patrón “cuál-ganadónde” (*which-won-where*), que facilitan la investigación de megaambientes (Gauch y Zobel, 1997);
- (ii) las interrelaciones entre los ambientes de prueba, que facilitan la identificación de mejores ambientes para la evaluación de cultivares (Cooper et al., 1997) y de aquellos que son redundantes y pueden descartarse (Yan y Rajcan, 2002);
- (iii) las interrelaciones entre genotipos que posibilita la comparación entre ellos y la clasificación de los mismos considerando tanto en el rendimiento medio como la estabilidad (Yan et al., 2001).

En un biplot la puntuación del i -ésimo genotipo en la k -ésima componente principal se muestra como un punto definido por $g_{ik} = \lambda_k^s \alpha_{ik}$, y la correspondiente al j -ésimo ambiente

en la k -ésima componente por $e_{kj} = \lambda_k^{1-s} \gamma_{jk}$ donde $k = 1, 2$ para un biplot bidimensional y s es el factor de partición de los valores singulares. Teóricamente, el factor de partición puede tomar cualquier valor entre 0 y 1. Dentro de este rango, la elección de s no altera las relaciones o interacciones relativas entre los genotipos y los ambientes, aunque la apariencia del biplot será diferente. Cuando $s = 1$, $g_{ik} = \lambda_k \alpha_{ik}$, y $e_{kj} = \gamma_{jk}$, los valores singulares se dividen por completo en los autovectores de los genotipos. En esta escala la unidad de las puntuaciones de los genotipos (g_{ik}) es la unidad original del carácter fenotípico evaluado y las puntuaciones ambientales (e_{kj}) están normalizadas (es decir no tienen unidad). Cuando $s = 0$, $g_{ik} = \alpha_{ik}$, y $e_{kj} = \lambda_k \gamma_{jk}$, los valores singulares se dividen por completo en los autovectores de los ambientes. En esta escala las puntuaciones ambientales están en la unidad original del carácter fenotípico evaluado y las de los genotipos no tienen unidad. Cuando $s = 0,5$, $g_{ik} = \lambda_k^{0,5} \alpha_{ik}$, y $e_{kj} = \lambda_k^{0,5} \gamma_{jk}$, la partición es simétrica. En esta escala las puntuaciones de los genotipos y las ambientales tienen la misma unidad que es la raíz cuadrada de la unidad original. El valor de $s = 0,5$ es empleado en el biplot GE y el más utilizado en GGE, aunque dependiendo de los intereses de la investigación, se pueden construir numerosas vistas del biplot GGE derivado de SREG. Independientemente del factor de partición en valores singulares utilizado, los biplots GGE revelan el mismo patrón *which-won-where*. Sin embargo, difieren en su precisión al mostrar la interrelación entre ambientes y genotipos. La partición centrada en los genotipos ($s = 1$) muestra la interrelación entre genotipos con mayor precisión que cualquier otro método; la partición enfocada en los ambientes ($s = 0$) es la más informativa sobre las interrelaciones entre los ambientes; y la simétrica ($s = 0,5$) permite visualizar la magnitud relativa tanto de la variación de los genotipos como de los ambientes.

3.1.2. Modelo AMMI robusto

El modelo AMMI, en su forma estándar, asume que no hay valores atípicos en el conjunto de datos. Sin embargo, la presencia de *outliers* es más una regla que una excepción cuando se consideran datos agronómicos debido características inherentes a los genotipos que se evalúan, errores de medición o el efecto inesperado de plagas o enfermedades que pueden afectar el rendimiento de algunos genotipos.

Rodrigues et al. (2016) proponen una generalización robusta del modelo AMMI, que resulta de ajustar un modelo lineal robusto basado en el estimador M-Huber (Huber, 1981) y luego utilizar un procedimiento de DVS o de análisis de componentes principales (ACP) robusto. Para la DVS o el ACP los autores consideraron varios métodos, dando lugar a un total de cinco modelos robustos llamados: R-AMMI, H-AMMI, G-AMMI, L-AMMI, PP-AMMI.

El empleo de la versión robusta del modelo AMMI puede ser extremadamente útil debido a que una mala representación de genotipos y ambientes puede resultar en una mala decisión con respecto a qué genotipos seleccionar para un conjunto dado de ambientes (Gauch y Zobel, 1997; Yan et al., 2000). A su vez, la elección de los genotipos incorrectos pueden provocar grandes pérdidas en términos de rendimiento. Los biplots obtenidos de los modelos robustos mantienen las características e interpretación estándar del modelo AMMI clásico (Rodrigues et al., 2016).

3.1.3. Métodos de imputación

Una limitación importante que presentan los modelos multiplicativos descriptos previamente es que requieren que el carácter fenotípico bajo estudio se encuentre registrado para todas las combinaciones entre genotipos y ambientes, es decir no admiten valores perdidos. Aunque los EMA están diseñados para que todos los genotipos se evalúen en todos los ambientes, la presencia de valores faltantes es muy común debido a errores de medición o pérdidas de plantas por animales, inundaciones o problemas durante la cosecha, además de la dinámica propia de las evaluaciones en las que se incorporan o descartan genotipos debido a su pobre desempeño (Hill y Rosenberg, 1985).

Se han propuesto numerosas metodologías para superar el problema de valores ausentes en el conjunto de datos, entre las cuales se encuentran:

- EM-AMMI: Gauch y Zobel (1990) desarrollaron un procedimiento iterativo que utiliza el algoritmo de maximización de la esperanza (algoritmo EM de las siglas en inglés *Expectation Maximization*) incorporando el modelo AMMI.
- EM-SVD: (Perry, 2015) propusieron un método de imputación que combina el algoritmo EM con DVS.
- EM-PCA: Josse y Husson (2016) propusieron imputar los valores faltantes de un conjunto de datos mediante un ACP.
- GabrielEigen: Arciniegas-Alarcón et al. (2010) presentaron un método de imputación que combina regresión y aproximación de rango inferior usando DVS.
- WGabriel Eigen: Arciniegas-Alarcón et al. (2014) plantearon una extensión ponderada del método GabrielEigen.

3.2. Creación de un paquete de R

Un paquete de R es un conjunto de funciones programadas en este lenguaje que comparten fines específicos y se distribuyen con un protocolo estandarizado, garantizando su correcto funcionamiento. Para la creación de un paquete se deben seguir ciertas convenciones referidas a la creación y almacenaje de carpetas y archivo con código de programación, documentación e instrucciones de sistema. La gestión de todos estos documentos puede ser manual, pero existen paquetes de R que asisten en la tarea del desarrollo de nuevos paquetes automatizando ciertas fases del proceso. En este trabajo se usaron los paquetes: *devtools* (Wickham et al., 2021), *usethis* (Wickham y Bryan, 2021), *roxygen2* (Wickham et al., 2020), *testthat* (Wickham, 2011) y *available* (Ganz et al., 2019).

Una vez finalizada esta etapa, dichas carpetas y archivos se compilan y comprimen para su distribución. Si esto se realiza con Windows como sistema operativo, se debe descargar e instalar el software Rtools disponible en CRAN.

Todo este proceso se realizó utilizando Git y GitHub. Git es un sistema de control de versiones, una herramienta que toma inicialmente una versión de un documento y luego registra los cambios que sufre el mismo a lo largo del tiempo. Esto facilita el trabajo colaborativo entre distintas personas ya que si más de una persona trabaja en el mismo documento, el sistema de control de versiones las puede integrar en una nueva. Git es más útil cuando se combina con GitHub. Este último es el servicio de hosting que se utiliza para que el proyecto tenga una presencia en la web permitiéndole a otras personas explorar los archivos, su historia, sincronizarse con la versión actual, proponer y realizar cambios, etc. Git + GitHub es el entorno más popular para los desarrolladores de paquetes de R ya que permite a cualquier persona descargar e instalar un paquete e incluso realizar aportes, detectar *bugs*, incluir sugerencias, etc.

A continuación se detallan los distintos pasos que componen la creación de un paquete en R bajo un enfoque de trabajo reproducible, lo cual significa que los mismos pueden usarse de ejemplo para el desarrollo de nuevos paquetes o para imitar la creación del paquete *geneticae* que es objeto de desarrollo de este trabajo.

3.2.1. *geneticae*

En primer lugar se debe elegir el nombre del paquete cumpliendo con ciertas reglas: solo puede contener letras, números o puntos; debe tener al menos dos caracteres y empezar con una letra y no terminar con un punto. Se debe chequear si el nombre elegido está disponible en los repositorios CRAN, Bioconductor y GitHub. Para ello, se utiliza

la función `available()` del paquete *available*, que además indicará si el nombre elegido tiene algún significado especial que podemos desconocer (revisa las webs de *Wikipedia*, *Wiktionary* y *Urban Dictionary*). El nombre elegido en este caso fue “geneticae” (Figura 3.1).

```
> library(available)
> available("geneticae")
Urban Dictionary can contain potentially offensive results,
should they be included? [Y]es / [N]o:
1: Y
— geneticae —
Name valid: ✓
Available on CRAN: ✓
Available on Bioconductor: ✓
Available on GitHub: ✓
Abbreviations: http://www.abbreviations.com/geneticae
Wikipedia: https://en.wikipedia.org/wiki/geneticae
Wiktionary: https://en.wiktionary.org/wiki/geneticae
Urban Dictionary:
  Not found.
Sentiment:???
```

Figura 3.1: Chequeo de disponibilidad del nombre *geneticae* elegido para el paquete en desarrollo mediante la función `available()` del paquete *available*.

3.2.2. Estructura general del paquete

Un paquete de R se construye creando y guardando diversos archivos y carpetas en un directorio cuyo nombre es igual al elegido en el paso anterior. Algunos elementos son de presencia obligatoria; entre ellos:

- Archivo DESCRIPTION: archivo de texto que describe el contenido del paquete, quienes son sus desarrolladores, establece como se va a relacionar con otros, el tipo de licencia con el que se distribuye, los requisitos de sistema, etc.
- Carpeta R: contiene el o los archivos de código (*script*) de R con las funciones del paquete.
- Carpeta man: incluye archivos con la documentación del paquete, funciones y data-sets.
- README: archivo de texto que brinda información sobre el proyecto.
- Archivo NAMESPACE: declara las funciones del paquete que se ponen a disposición de los usuarios y lista las funciones de otros paquetes de las cuales hace uso.

Existen otros elementos cuya inclusión es opcional, por ejemplo:

- Carpeta data: contiene objetos de R con los conjuntos de datos.
- Carpeta vignettes: contiene los tutoriales que muestran ejemplos de uso del paquete, generalmente escritos en Rmarkdown
- Carpeta tests: incluye código que permiten someter al paquete a diversos controles.

Si bien estas carpetas y archivos pueden crearse en forma manual, es conveniente utilizar la función `create_package()` del paquete *usethis* que se encarga de generar automáticamente un directorio con todas las componentes requeridas para el desarrollo del paquete que sirve de plantilla para facilitar la parte inicial de este proceso (Figura 3.2).

```
> # Cargar la librería devtools
> library(devtools)
Loading required package: usethis
> # Crear el paquete geneticae
> create_package("~/home/julia-fedora/Escritorio/geneticae")
✓ Creating '/home/julia-fedora/Escritorio/geneticae/'
✓ Setting active project to '/home/julia-fedora/Escritorio/geneticae'
✓ Creating '.R/'
✓ Writing 'DESCRIPTION'
Package: geneticae
Title: What the Package Does (One Line, Title Case)
Version: 0.0.0.9000
Authors@R (parsed):
 * First Last <first.last@example.com> [aut, cre] (YOUR-ORCID-ID)
Description: What the package does (one paragraph).
License: 'use_mit_license()', 'use_gpl3_license()' or friends to
        pick a license
Encoding: UTF-8
LazyData: true
Roxygen: list(markdown = TRUE)
RoxygenNote: 7.1.1
✓ Writing 'NAMESPACE'
✓ Writing 'geneticae.Rproj'
✓ Adding '.Rproj.user' to '.gitignore'
✓ Adding '^geneticae\\.Rproj$', '^\\.Rproj\\.user$' to '.Rbuildignore'
✓ Opening '/home/julia-fedora/Escritorio/geneticae/' in new RStudio session
✓ Setting active project to '<no active project>'
> |
```

Figura 3.2: Creación del paquete *geneticae* mediante la función `create_package()` del paquete *usethis*.

3.2.3. Archivo DESCRIPTION

El archivo DESCRIPTION provee toda la metadata sobre el paquete presentada a través de campos, algunos de los cuales tienen que estar de forma obligatoria y otros son opcionales. Los campos obligatorios son:

- Package: nombre del paquete.
- Title: título del paquete (hasta 65 caracteres).
- Version: número de la versión actual del paquete, en este caso 0.1.9000.
- Author, Maintainer o Authors@R: autores, contribuyentes y personas a cargo del mantenimiento del paquete.
- Description: descripción del paquete.
- License: nombre de la licencia bajo la cual se distribuye el paquete. Si se pretende que cualquiera lo puede usar, entonces se debe recurrir a los tipos más comunes de licencia para código abierto: CC0, MIT o GPL.

Los elementos no obligatorios son:

PONGO LOS QUE APARECEN EN LA FIGURA

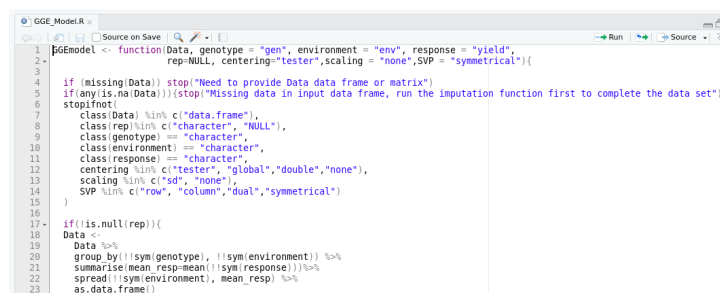
- Encoding: especifica la codificación que se utiliza para el archivo DESCRIPTION, los .R, el NAMESPACE y los archivos .Rd.

- LazyData: indica si los datos estan o no disponibles inmediatamente cuando se carga el paquete.
- RoxygenNote: versión de roxygen. ?????
- Date: fecha de publicación de la versión actual del paquete. **SACARIA PORQUE NO ESTÁ EN MI DESCRIPTION**
- Imports: en el caso de que el código desarrollado haga uso de funciones pertenecientes a otros paquetes, los mismos deben ser listados en este campo.
- Suggests: listado de paquetes que no son imprescindibles para el uso del nuevo paquete, pero que pueden ser útiles como herramientas secundarias para el mismo (por ejemplo, para seguir los tutoriales).
- VignetteBuilder: paquetes requeridos para la creación de las viñetas.
- Depends: mínima versión de R con la cual el paquete es compatible.
- URL: dirección de la página web del paquete.
- BugReports: dirección donde los usuarios pueden enviar avisos sobre los problemas que encuentren al utilizar el paquete.
- Language: indica el idioma en la que se escribe la documentación del paquete.

El archivo DESCRIPTION del paquete *geneticae* se muestra en la Figura 3.3.

3.2.4. Archivos de código

Una vez creada la estructura del paquete y el archivo DESCRIPTION se deben programar las funciones que el mismo contendrá. Estas deben ser guardadas en *scripts* con extensión .R, en el subdirectorio R/. Los *scripts* pueden contener código para una o más funciones y ser guardadas con cualquier nombre, aunque es recomendable que el mismo este relacionado con su contenido (Figura 3.4).



```

1 GGEmodel <- function(Data, genotype = "gen", environment = "env", response = "yield",
2   rep=NULL, centering="tester", scaling = "none", SVP = "symmetrical"){
3
4   if (missing(Data)) stop("Need to provide Data data frame or matrix")
5   if (any(is.na(Data))) stop("Missing data in input data frame, run the imputation function first to complete the data set")
6   stopifnot(
7     class(Data) %in% c("data.frame"),
8     class(rep) %in% c("character", "NULL"),
9     class(genotype) == "character",
10    class(environment) == "character",
11    class(response) == "character",
12    centering %in% c("tester", "global", "double", "none"),
13    scaling %in% c("sd", "none"),
14    SVP %in% c("row", "column", "dual", "symmetrical")
15  )
16
17  if (!is.null(rep)){
18    Data <-
19      Data %>%
20      group_by(!sym(genotype), !sym(environment)) %>%
21      summarise(mean_resp=mean(!sym(response))) %>%
22      spread(!sym(environment), mean_resp) %>%
23    as.data.frame()
  
```

Figura 3.4: Fragmento de función `GGEmodel()` del paquete *geneticae*

A medida que se desarrolla el paquete, se van generando relaciones complejas entre las funciones programadas. Algunas de ellas son de uso interno (son invocadas por otras


```

DESCRIPTION
1 Package: geneticae
2 Title: Statistical Tools for the Analysis of Multi Environment Agronomic Trials
3 Version: 0.1.9000
4 Authors@R: c(
5   person(given = "Julia",
6     family = "Angelini",
7     role = c("aut", "cre"),
8     email = "jangelini_93@hotmail.com",
9     comment = c(ORCID = "0000-0002-5815-1771")
10  ),
11   person(given = "Marcos",
12     family = "Prunello",
13     role = c("aut"),
14     email = "marcosprunello@gmail.com",
15     comment = c(ORCID = "0000-0002-9611-527X")
16  ),
17   person(given = "Gerardo",
18     family = "Cervigni",
19     role = c("aut")
20  )
21 )
22 Description: Data from multi environment agronomic trials, which are often
23 carried out by plant breeders, can be analyzed with the tools offered by this
24 package such as the Additive Main effects and Multiplicative Interaction model
25 or 'AMMI' ('Gauch' 1992, ISBN:9780444892409)
26 and the Site Regression model or 'SREG' ('Cornelius' 1996,
27 <doi:10.1201/9780367802226>). Since these methods present a poor performance
28 under the presence of outliers and missing values, this package includes
29 robust versions of the 'AMMI' model ('Rodrigues' 2016,
30 <doi:10.1093/bioinformatics/btv533>), and also imputation techniques
31 specifically developed for this kind of data ('Arciniegas-Alarcón' 2014,
32 <doi:10.2478/bile-2014-0006>).
33 License: GPL-2
34 Encoding: UTF-8
35 LazyData: true
36 RoxygenNote: 7.1.2
37 Imports:
38   stats,
39   GGEbiplots,
40   ggforce,
41   ggplot2,
42   scales,
43   MASS,
44   pcaMethods,
45   rrcov,
46   dplyr,
47   bcv,
48   missMDA,
49   calibrate,
50   graphics,
51   agrdat,
52   reshape2,
53   matrixStats,
54   tidyr,
55   prettydoc
56 Suggests:
57   spelling,
58   knitr,
59   rmarkdown,
60   testthat
61 VignetteBuilder: knitr
62 Depends: R (>= 2.12.0)
63 URL: https://github.com/jangelini/geneticae
64 BugReports: https://github.com/jangelini/geneticae/issues
65 Language: en-US
66

```

Figura 3.3: Archivo DESCRIPTION de *geneticae*

funciones del paquete para cumplir con alguna tarea específica) y otras son diseñadas para que estén disponibles para los usuarios (es decir se “exportan”). Además algunas funciones invocan a otras pertenecientes a paquetes escritos por terceros. Esto hace necesario realizar pruebas del funcionamiento del código producido durante todo el proceso de desarrollo para garantizar que el código realice lo que realmente se desea y para corregir errores en la programación (depuración o *debugging*). Para simular el proceso de construcción, instalación y carga del paquete durante su desarrollo se utiliza la función `load_all()` que permite acceder a las funciones del paquete para su evaluación.

3.2.5. Documentación

Uno de los aspectos más importantes de la creación de un paquete es la documentación donde se describe cómo se usa cada función, para qué sirven los argumentos, aclarar qué tipo de resultado devuelve, proveer ejemplos para el uso, etc. El paquete *roxygen2* provee

pautas para escribir comentarios con un formato especial que incluyan toda la información requerida antes de la definición de la función en el mismo archivo de código.

El flujo de trabajo para crear la documentación con el paquete *roxygen2* es el siguiente:

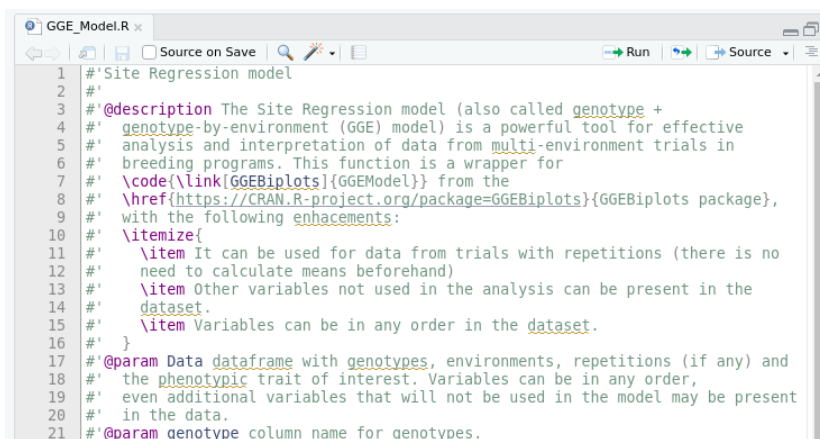
1. Agregar comentarios a los archivos .R. Estos deben comenzar con `#'`, para distinguirlo de los comentarios regulares, y preceden a cada función. La primera línea es el título y el párrafo que le sigue es su descripción. Para el resto de los campos de la documentación, se utilizan etiquetas listadas línea tras línea que comienzan con `@`, siendo las más importantes a incluir:
 - `@param`: detalla para qué sirve cada parámetro de la función, que tipo de objeto es y que valor toma por *default* (opcional).
 - `@return`: explica qué objeto devuelve la función.
 - `@details`: agrega cualquier aclaración que se considere necesaria.
 - `@examples`: incluye ejemplos de uso de la función.
 - `@export`: indica que esta función tiene que estar disponible cuando alguien cargue el paquete con `library()`.
 - `@references`: inserta referencias bibliográficas.
2. Ejecutar `document()` del paquete *devtools* para convertir los comentarios escritos en formato roxygen en los archivos que compondrán el manual de ayuda y que deben ir guardados en la carpeta `man`. Además, esto se encarga de generar el archivo `NAMESPACE`, que tiene como objetivo declarar cuáles son las funciones del nuevo paquete que son para exportar, así como también listar todas las funciones importadas de otros paquetes.

La figura 3.5 muestra un fragmento de los comentarios roxygen de la función `GGE-model()` del paquete *geneticae*.

3.2.6. Uso de funciones de otros paquetes

Cuando el código desarrollado invoca a funciones de otros paquetes, los mismos deben ser listados en el campo Imports del archivo `DESCRIPTION` como se mencionó en la sección 3.2.3. Esto puede hacerse de manera manual o mediante la función `use_package()` del paquete *devtools* indicando el nombre del paquete.

En el código se debe hacer uso de tales funciones anteponiéndolo a su nombre el del paquete y el operador `::`. Por ejemplo, `dplyr::group_by()` invoca la función `group_by()` del paquete *dplyr*. En el caso de que alguna función se aplicara con mucha frecuencia, se



```

1 #' Site Regression model
2 #'
3 #' @description The Site Regression model (also called genotype +
4 #' genotype-by-environment (GGE) model) is a powerful tool for effective
5 #' analysis and interpretation of data from multi-environment trials in
6 #' breeding programs. This function is a wrapper for
7 #' \code{\link[GGEbiplots]{GGEModel}} from the
8 #' \href{https://CRAN.R-project.org/package=GGEbiplots}{GGEbiplots} package,
9 #' with the following enhancements:
10 #'
11 #' \itemize{
12 #'   \item It can be used for data from trials with repetitions (there is no
13 #'     need to calculate means beforehand)
14 #'   \item Other variables not used in the analysis can be present in the
15 #'     dataset.
16 #'   \item Variables can be in any order in the dataset.
17 #' }
18 #' @param Data dataframe with genotypes, environments, repetitions (if any) and
19 #' the phenotypic trait of interest. Variables can be in any order,
20 #' even additional variables that will not be used in the model may be present
21 #' in the data.
22 #' @param genotype column name for genotypes.

```

Figura 3.5: Fragmento de los comentarios roxygen de la función `GGEModel()` del paquete *geneticae*.

puede prescindir del comando anterior si se agrega el nombre de la función y el paquete de procedencia en la etiqueta `@importFrom` en los comentarios roxygen. Para el ejemplo anterior se debería indicar `@importFrom dplyr group_by`. Esto permite mencionar a la función en el código sin el operador `::`.

Si se utilizan repetidamente muchas funciones de otro paquete, es posible importarlas todas indicando el nombre del mismo en la etiqueta `@import` en los comentarios roxygen. Sin embargo, esta es la solución menos recomendada porque hace que el código sea más difícil de leer, y si tiene muchos paquetes, aumenta la posibilidad de que entren en conflicto nombres de funciones.

3.2.7. Testeos

Probar el código desarrollado, sometiéndolo a casos particulares y a distintos ejemplos, es fundamental en la creación de paquetes ya que permite detectar y corregir errores y asegurarse que el código haga lo que realmente se desea.

Como se mencionó anteriormente, esto puede hacerse de manera dinámica e interactiva durante el desarrollo al instalar y cargar el paquete en gestación con `load_all()`. Sin embargo, si siempre se realizan los mismos controles es posible automatizar este proceso. Para ello, se generan unidades de testeo que ponen a prueba el código corriendo parte del mismo bajo distintas circunstancias y comparando el resultado obtenido con el esperado por el desarrollador. La función `use_test()` del paquete *testthat*, que agrega “testthat” al campo Suggest del archivo DESCRIPTION, crea un directorio `test/testthat` para ubicar los códigos con los testeos y un archivo `testthat.R` que se encarga de la ejecución de los mismos. Una vez escritos estos archivos que establecen cuales son los controles a realizar

automáticamente, se pueden evaluar los resultados con la función `test()` del paquete *devtools*. Ante cada error encontrado, se debe corregir el código y repetir este proceso hasta que todas las unidades de testeo pasen la prueba. En la Figura 3.6 se muestra el resultado de evaluar las unidades de testeo creadas para el paquete *geneticae*.

```
> test()
i Loading geneticeae
i Testing geneticeae
✓ | OK F W S | Context
✓ | 6         | GGE_Model [0.2 s]
✓ | 7         | GGE_Plot [0.8 s]
✓ | 4         | impute [0.2 s]
✓ | 3         | r_AMMI [0.2 s]

== Results ==
Duration: 1.4 s

[ FAIL 0 | WARN 0 | SKIP 0 | PASS 20 ]
```

Figura 3.6: Resultado de correr la función `test()` del paquete *devtools* en *geneticae*

Una medida de la calidad de un paquete está dada por el porcentaje de código que es evaluado durante los testeos. Esta se puede obtener mediante la función `test_coverage()` del paquete *covr* (Hester, 2020). El paquete *geneticae* tiene un porcentaje total de cobertura de los test igual a 24.75% (Figura 3.7).

geneticae coverage - 24.75%

Files		Source					
File	Lines	Relevant	Covered	Missed	Hits / Line	Coverage	
R/W_GabrielEigen.R	270	131	0	131	0	0.00%	
R/GabrielEigen.R	136	63	0	63	0	0.00%	
R/rAMMI.R	235	126	2	124	0	1.59%	
R/GGE_Plot.R	465	238	57	181	1	23.95%	
R/impute.R	186	49	37	12	1	75.51%	
R/GGE_Model.R	119	27	21	6	4	77.78%	
R/EM_AMMI.R	160	59	49	10	6	83.05%	

Figura 3.7: Porcentaje del código de *geneticae* que es evaluado durante los testeos obtenido mediante la función `test_coverage()` del paquete *covr*.

3.2.8. Datasets

A menudo es útil incluir conjuntos de datos en un paquete a fin de proporcionar ejemplos de uso de las funciones incluidas. Los datasets que se deseen añadir a un paquete deben ser guardados como archivos `.RData` en el directorio `data/`. La función `use_data()` del paquete *usethis* puede ser empleada para automatizar este proceso.

Los objetos de la carpeta `data` siempre se exportan, por lo cual se debe agregar documentación para los mismos. Esto se puede incluir en cualquier archivo de código `.R` dentro del directorio `R/`.

3.2.9. Archivo README

Un README es un archivo de texto plano que se utiliza para documentar o brindar información sobre alguna pieza de software o proyecto. Si un directorio contiene un archivo README, se espera que el usuario lo lea antes de explorar el resto del contenido. En el contexto de creación de paquetes de R, el README se suele escribir en RMarkdown y describe brevemente por qué y para qué alguien podría usar el paquete, junto con menciones para su instalación y un ejemplo introductorio. Además, este archivo se muestra en la página del paquete cuando este es publicado en plataformas como GitHub.

Para generar el README se utiliza la función `use_readme_rmd()` que crea un archivo de Rmarkdown con una plantilla donde se completa su contenido.

En el archivo README se suelen incluir insignias (“badges”) y el logo del paquete. Algunas funciones del paquete *usethis* permiten agregar las insignias al README. Por otro lado, muchos de los paquetes de R disponen de un logo con forma hexagonal conocido como *hexSticker*. Esto permite darle identidad al paquete. El logo se puede crear con ayuda del paquete *HexSticker* y ser añadido al README con la función `use_logo()`. La Figura 3.8 presenta un fragmento del archivo README mostrado en el repositorio GitHub del paquete *geneticae*.

3.2.10. Archivo NEWS

El archivo NEWS se encarga de contar los cambios presentes en cada versión nueva del paquete que se publica. Mientras que el README apunta a ser leído por nuevos usuarios, el archivo NEWS es para aquellos que ya usan el paquete.

Se sugiere usar RMarkdown para escribir este archivo y colocar un título principal para cada versión, seguido por títulos secundarios que describen lo realizado (cambios principales, bugs arreglados, etc.). Si se trata de cambios impulsados por otras personas, por ejemplo, a través de sugerencias hechas en GitHub, se los menciona para darles mérito. Una buena práctica es ir escribiendo este archivo cada vez que se realiza algo nuevo en el paquete. La función que permite crear este archivo automáticamente es `use_news_md()` del paquete *usethis*.

La figura 3.9 presenta el contenido del archivo NEWS del paquete *geneticae*.

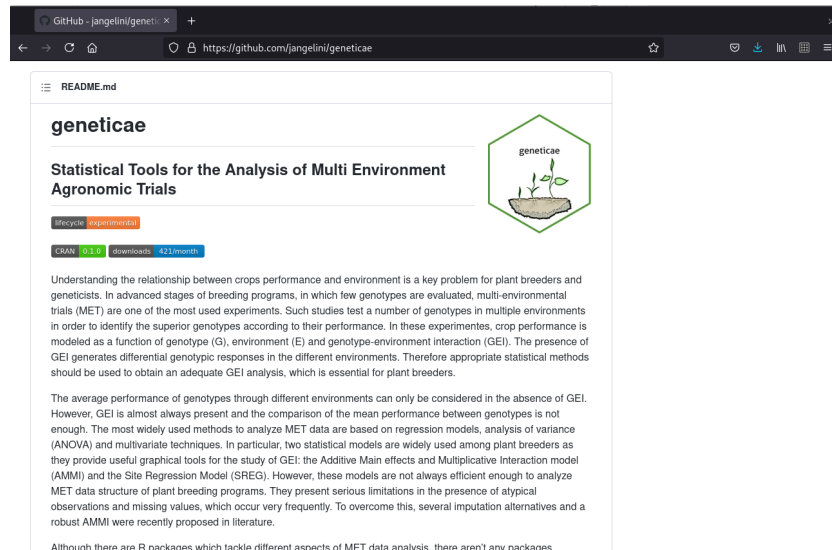


Figura 3.8: Fragmento del archivo README mostrado en el repositorio GitHub del paquete *geneticae*.

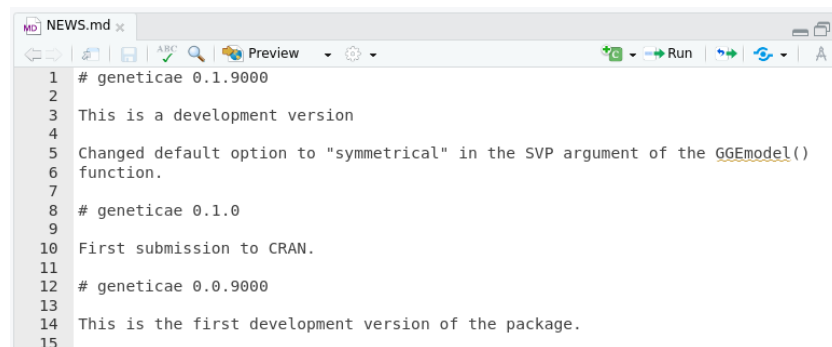


Figura 3.9: Archivo NEWS de *geneticae*

3.2.11. Viñetas

Una viñeta es un tipo especial de documentación que puede agregarse al paquete para dar más detalles y ejemplos sobre el uso del mismo. En ella se brinda una descripción del problema que el paquete está diseñado para resolver y muestra al lector cómo resolverlo. Se diferencian de las páginas de ayuda en que su adición es opcional y no sigue una estructura fija, dándole la libertad al autor de enseñar de la forma que más le guste cómo usar su paquete.

Muchos de los paquetes existentes tienen viñetas a las que se puede acceder utilizando la función `browseVignettes("nombre del paquete")` si el mismo se encuentra instalado o consultando en su página de CRAN, por ejemplo para el paquete *geneticae*: <https://cran.r-project.org/web/packages/geneticae/vignettes/a-tutorial.html>.

Generalmente, las viñetas son generadas en RMarkdown, un formato de escritura que facilita la presentación de texto entrelazado con código y resultados. Para crear viñetas se emplea la función `use_vignette("nombre del paquete")` del paquete *usethis* que crea un directorio `vignettes/`, agrega las dependencias necesarias a `DESCRIPTION` y genera una plantilla en RMarkdown para redactar la viñeta.

3.2.12. R CMD check e instalación

Además de los controles interactivos o automatizados que los desarrolladores realicen, existe un riguroso proceso de control conocido como R CMD check que debe ser superado sin errores, advertencias ni ningún tipo de nota si se desea publicar el paquete en algún repositorio oficial como CRAN. R CMD check esta compuesto por más de 50 chequeos individuales entre los cuales se encuentran: la estructura del paquete, el archivo `DESCRIPTION`, `NAMESPACE`, el código de R, los datos, la documentación, entre otros.

Se aconseja realizar verificaciones completas de que todo funciona a medida que se van incorporando funciones para detectar y solucionar problemas de forma temprana. Una vez que se desarrollaron todos los elementos necesarios para el paquete y no se detectan errores, advertencias o notas, se ejecuta la función `install()`, con el objetivo de instalar el paquete en la biblioteca.

3.2.13. Publicación y difusión

Por último para que otros usuarios puedan utilizar el paquete desarrollado es necesario publicarlo en algún sitio de cual pueda ser descargado e instalado. Esto se logra subiéndolo a un proyecto personal público en GitHub o enviándolo a repositorios oficiales como CRAN. Para ser aceptado en CRAN, el paquete además de atravesar con éxito los rigurosos controles de R CMD check debe superar una serie de políticas que son comprobadas manualmente por revisores.

El paquete *geneticae* se encuentra disponible tanto en CRAN (<https://cran.r-project.org/web/packages/geneticae/index.html>) como en GitHub (<https://github.com/jangelini/geneticae>).

Con el fin de favorecer a la difusión del paquete una vez que el mismo es finalizado, es conveniente publicarlo en una página web. Esta es una tarea reactivamente sencilla gracias a dos factores. En primer lugar, la función `build_site()` del paquete *pkgdown* toma todo el material creado para el paquete (documentación, README, tutoriales, NEWS, etc.) y crea automáticamente en un sitio web que puede ser posteriormente personalizado. En segundo lugar, GitHub ofrece un servicio gratuito de *web-hosting* que posibilita la

publicación del sitio en internet (<https://jangelini.github.io/geneticae/>).

3.3. Aplicación web Shiny

Shiny es un paquete R que permite construir aplicaciones web directamente desde RStudio sin necesidad de conocer en profundidad los lenguajes HTML / CSS / JavaScript . Estas aplicaciones constituyen una interfaz gráfica entre el usuario y R, que permiten realizar un análisis a través de un navegador web sin necesidad de programar.

Una característica importante de las aplicaciones web creadas mediante Shiny es que son dinámicas e interactivas. Para que shiny funcione correctamente, es necesario tener instalado R 3.0.2 o cualquier versión posterior.

3.3.1. Estructura de la aplicación web Shiny

Las aplicaciones están compuestas por la interfaz de usuario, ui (*user interfaz*), sección server y la función shinyApp().

Interfaz del usuario

La interfaz del usuario (user interface o ui, por sus siglas en inglés) controla el diseño de la aplicación, recibe los inputs y muestra los outputs en el navegador. En general, definir las características de la interfaz puede no resultar tan sencillo ya que muchas de sus herramientas están vinculadas a otros lenguajes de programación, por ejemplo HTML, CSS o JavaScript. Sin embargo, las funciones del paquete shiny facilitan la tarea sin necesidad de conocer en profundidad estos lenguajes.

Server

En la sección server se escribe el código de R que le indica a la app qué debe hacer y cómo debe funcionar, incluyendo la lectura y manipulación de datos, el armado de gráficos, el ajuste de modelos, etc. Para esto, se define una función que debe tener dos argumentos: input y output. Los mismos son listas que almacenan elementos de entrada (datos u opciones elegidas por el usuario a través de la ui) y elementos de salida para mostrar en la app (resultados, tablas, gráficos, mapas, etc.), respectivamente.

Ejecución

Por último, se llama a la función shinyApp(), cuyos dos argumentos principales son ui y server, es decir, cada uno de los elementos definidos anteriormente. Ejecutar esta función da como resultado el lanzamiento de la aplicación, la cual podremos utilizar dentro de RStudio o usando nuestro navegador (Google Chrome, Mozilla Firefox, Microsoft Edge,

etc.). Es importante destacar que, al seguir estos pasos, la aplicación sólo funcionará mientras la sesión de RStudio desde la cual se lanzó siga vigente.

3.3.2. Desarrollo de la aplicación web Shiny

Una forma de desarrollar una aplicación es a partir de un nuevo directorio con un sólo archivo llamado `app.R`, como se muestra a continuación.

```
library(shiny)
ui<- ...
server<- ...
shinyApp(ui = ui, server = server)
```

En este archivo se carga el paquete `shiny`, se define la interfaz de usuario, la función `server` y por último, se ejecuta función que permite construir e iniciar una aplicación. Al ejecutar la aplicación la misma aparecerá, de manera predeterminada, en una ventana emergente. Sin embargo, otras dos opciones se pueden configurar desde el menú desplegable de *Run App*. Una de ellas es la ejecución en el panel del visor que permite verla al mismo tiempo que ejecuta el código. La segunda opción es ejecutar en un navegador externo mostrando la aplicación como la mayoría de los usuarios la verán. Dado que la sesión de R estará monitoreando la aplicación y ejecutando las ordenes dadas por el usuario, no se podrá ejecutar ningún comando.

En cualquier lenguaje de programación tener el código duplicado genera un desperdicio computacional y, lo que es más importante, aumenta la dificultad de mantener o depurar el código. Cuando se programa en R, se utilizan dos técnicas para lidiar con el código duplicado: guardar un valor usando una variable o utilizar una función para almacenar un cálculo. Ninguno de estos enfoques son apropiados en una aplicación web Shiny, sino que se utilizan expresiones reactivas. Una expresión reactiva tiene una diferencia importante con una variable: sólo se ejecuta la primera vez que se llama y luego almacena en caché el resultado de la misma hasta que necesite actualizarse. La programación reactiva es un estilo de programación que enfatiza valores que cambian con el tiempo, y cálculos y acciones que dependen de esos valores. Esto es importante para las aplicación web Shiny porque son interactivas: los usuarios cambian los inputs, lo que hace que la lógica se ejecute en el servidor que finalmente resultan en actualización de los outputs/resultados.

Entre los problemas que pueden surgir al crear una aplicación web Shiny se encuentran los errores inesperados, no se obtiene ningún error pero el valor obtenido es incorrecto, o bien todos los resultados son correctos, pero no se actualizan cuando se deben. Una vez localizada la fuente del error, la herramienta más poderosa es el depurador interactivo, éste detiene la ejecución y brinda una consola interactiva donde puede se ejecutar cualquier

código para descubrir el error. Para iniciar el mismo, se puede agregar la función `browser()` en el código fuente, o bien agregar un punto de interrupción RStudio haciendo clic a la izquierda del número de línea.

Al modificar la aplicación, se la ejecuta para poder ver los cambios realizados, por lo tanto resulta esencial reducir la velocidad de iteración. La primera forma acelerar el proceso consiste en escribir el código, utilizar el atajo del teclado `Cmd/Ctrl+ Shift+ Enter` en lugar del botón “Ejecutar aplicación”, experimentar interactivamente con la aplicación y cerrar la aplicación, repitiendo este proceso al realizar cualquier cambio. Otra forma de reducir aún más la velocidad de iteración es activar la recarga automática (`options(shiny.autoreload = TRUE)`) y luego ejecutar la aplicación en un trabajo en segundo plano. Con este flujo de trabajo cuando se guarde un archivo, su aplicación se reiniciará: no es necesario cerrarla y reiniciarla, lo cual conduce a un flujo de trabajo aún más rápido. La principal desventaja de esta técnica es que debido a que la aplicación se ejecuta en un proceso separado, es considerablemente más difícil de depurar.

3.3.3. Compartiendo una aplicación web Shiny

Una vez creada la aplicación se la publica para su libre uso. En este caso Geneticae Shiny Web APP encuentra disponible momentaneamente en un servidor gratuito de uso limitado <https://geneticae.shinyapps.io/geneticae-shiny-web-app/>, sin embargo será instalado en el servidor de CONICET para su libre uso. También se puede acceder desde la página de CEFOTI-CONICET <https://www.cefobi-conicet.gov.ar/bases-de-datos-y-programas/> y desde GitHub <https://github.com/jangelini/Geneticae-Shiny-Web-APP> al código fuente.

Capítulo 4

Resultados

En esta sección se muestran ejemplos de uso tanto del paquete *geneticae* como de Geneticae Shiny Web APP.

4.1. Paquete de R *geneticae*

Para instalar la versión del paquete publicada en CRAN: `install.packages("geneticae")`, mientras que la versión en desarrollo se debe instalar desde el repositorio de GitHub: `devtools::install_github("jangelini/geneticae")`. Una vez instalado el paquete, se debe cargar en la sesión de R mediante el comando: `library(geneticae)`.

Información detallada sobre las funciones del paquete *geneticae* se puede obtener mediante `help(package = "geneticae")`. La ayuda para una función, por ejemplo `imputation()`, en una sesión R se puede obtener usando `?imputation` o `help(imputation)`. La función `browseVignettes("geneticae")` permite obtener la viñeta del paquete, es decir una descripción del problema que está diseñado para resolver así como ejemplos de aplicación del mismo.

Además, se encuentra disponible una página web (<https://jangelini.github.io/geneticae/>) que contiene una breve descripción de la utilidad del paquete, las funciones que se incluyen en él, un tutorial de uso, un enlace de acceso a la shiny app, entre otra información.

4.1.1. Conjuntos de datos en *geneticae*

El paquete *geneticae* proporciona dos conjuntos de datos que pueden utilizarse para ilustrar la metodología incluida para analizar los datos provenientes de EMA.

-
- *yan.winterwheat dataset* (Wright, 2020): se cuenta con información sobre el rendimiento de 18 variedades de trigo de invierno cultivadas en nueve ambientes en Ontario en 1993. A pesar de que el experimento contaba con cuatro bloques o réplicas en cada ambiente, sólo el rendimiento medio para cada combinación de variedad y ambiente se encuentra disponible.

```
data(yan.winterwheat)
head(yanwinterwheat)[1:3,]

##   gen  env yield
## 1 Ann BH93 4.460
## 2 Ari BH93 4.417
## 3 Aug BH93 4.669
```

- *plrv dataset* (de Mendiburu, 2020) se registró información sobre el rendimiento, el peso de planta y de la parcela de 28 genotipos en 6 localidades de Perú con el fin de estudiar la resistencia a PLRV (*Patato Leaf Roll Virus*) causante del enrollamiento de la hoja. Cada clon fue evaluado tres veces en cada ambiente.

```
data(plrv)
head(plrv)[1:3,]

##   Genotype Locality Rep WeightPlant WeightPlot   Yield
## 1   102.18    Ayac   1   0.5100000      5.10 18.88889
## 2   104.22    Ayac   1   0.3450000      2.76 12.77778
## 3   121.31    Ayac   1   0.5425000      4.34 20.09259
```

En las siguiente subsecciones se muestran las herramientas de análisis incluidas en el paquete utilizando el conjunto de datos *yan.winterwheat*.

4.1.2. Modelo AMMI

Para visualizar el efecto de IGA se utiliza el biplot GE obtenido del modelo AMMI. Este gráfico es posible obtenerlo utilizando la función `rAMMI()`. Esta función requiere datos en formato largo, es decir, cada fila corresponde a una observación y cada columna a una variable (genotipo, ambiente, repetición (si existe) y fenotipo observado). Si cada genotipo ha sido evaluado más de una vez en cada ambiente, la media fenotípica para cada combinación de genotipo y ambiente se calcula internamente y luego se estima el modelo. Las variables adicionales que no se utilizarán en el análisis pueden estar presentes en el conjunto de datos. No se permiten valores perdidos pero se pueden imputar como

se indica en la subsección 4.1.4.

El biplot clásico para el conjunto de datos *yan.winterwheat* se muestra en la figura 4.1 junto con la sentencia utilizada para obtener el mismo. El primer argumento es el conjunto de datos de entrada, luego se indican los nombres de las columnas en las cuales se encuentra la información necesaria para aplicar la técnica y por último el biplot que se desea obtener que por defecto es el derivado del modelo AMMI clásico. Opcionalmente, el porcentaje de IGA explicado por el biplot se puede agregar como una nota al pie con el argumento *footnote = T* así como un título con *titles = T*.

En este ejemplo, BH93, KE93 y OA93 son los ambientes que más contribuyen a la interacción ya que sus vectores son los de mayor magnitud. Los cultivares m12 y Kat presentan patrones de interacción similares (sus marcadores están próximos entre sí) y son muy diferentes de Ann y Aug, por ejemplo. La cercanía entre el cultivar Dia y el ambiente BH93 indica una fuerte asociación positiva entre ellos, lo que significa que BH93 es un ambiente extremadamente favorable para ese genotipo. Como los marcadores OA93 y Luc son opuestos, este ambiente es considerablemente desfavorable para ese genotipo. Por último, Cas y Reb están cerca del origen, lo que significa que se adaptan en igual medida a todos los ambientes.

```
rAMMI(yan.winterwheat, genotype = "gen", environment = "env",
       response = "yield", type = "AMMI", footnote = F, titles = F)
```

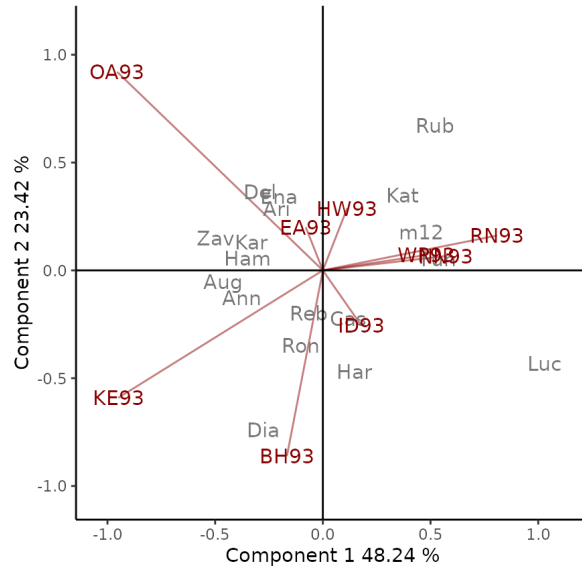


Figura 4.1: Biplot GE obtenido del modelo AMMI clásico basado en los datos de rendimiento de trigo de invierno obtenidos en Ontario en 1993. El 71,66 % de la variabilidad de la IGA se explica por los dos primeros términos multiplicativos. Los cultivares se muestran en letras minúsculas y los ambientes en mayúsculas.

Como se mencionó anteriormente, el modelo AMMI, en su forma estándar, asume que no hay valores atípicos presentes en los datos. Por lo tanto, en presencia de *outliers* se debe utilizar alguna de las alternativas robustas propuestas por Rodrigues et al. (2016), las cuales no se encuentran disponible en R hasta el momento. Sin embargo, dada la importancia práctica de este reciente avance metodológico, se incluyeron en la función `rAMMI()`. Para obtener los biplots GE derivados de los modelos robustos se debe indicar en el argumento *type* cuál de ellos se desea ajustar: “rAMMI”, “hAMMI”, “gAMMI”, “lAMMI”, “ppAMMI”.

Dado que el conjunto de datos de muestra *yan.winterwheat* no presenta valores atípicos, las conclusiones obtenidas con biplots robustos no diferirán de las obtenidas con el biplot clásico (Rodrigues et al., 2016). Por lo tanto, no se presenta ninguna interpretación de los biplots robustos.

4.1.3. Modelo de Regresión por Sitio

Para visualizar conjuntamente el efecto de G e IGA Yan et al. (2000) propuso el biplot GGE mediante el cual se pueden abordar diversos aspectos relacionados con la evaluación de genotipos y ambientes. Para obtener dicho biplot en primer lugar se debe ajustar el modelo SREG mediante la función `GGEmodel()`. Ésta es un *wrapper* de `GGEModel()` del paquete `GGEbiplots` (Dumble, 2017). Como en el caso de `rAMMI()`, para poder utilizarla los datos deben presentarse en un formato largo y se permiten repeticiones o variables adicionales en el conjunto de datos. El rasgo fenotípico para cada combinación de genotipo y ambiente debe estar registrado, sino se debe recurrir previamente a alguna técnica de imputación para completar los datos (subsección 4.1.4).

La sentencia utilizada para ajustar el modelo GGE en el conjunto de datos *yan.winterwheat* se muestra a continuación. El primer argumento de la misma consiste en el nombre del conjunto de datos y en los siguientes indican los nombres que reciben las columnas que contienen la información de los genotipos, ambientes y del rasgo fenotípico de interés. Por defecto, la función considera que no hay réplicas en el conjunto de datos, sin embargo, si existieran en el parámetro *rep* se debe indicar el nombre de la columna con dicha información. Otros argumentos de dicha función son el método de centrado, de partición de los valores singulares (SVP de sus siglas en inglés *Singular Value Partition*) y escalado. Por defecto los datos se centran utilizando la opción *centering* = “*tester*” lo cual resulta en el modelo SREG, otro valor dará lugar a un modelo diferente. La elección del método de SVP no altera las relaciones o interacciones relativas entre los genotipos y los ambientes, aunque la apariencia del biplot será diferente (Yan 2002). El método de partición de los valores singulares centrado en los genotipos (*SVP* = “*row*”) muestra la interrelación entre genotipos con mayor precisión, el enfocado a los ambientes (*SVP* = “*column*”) es el más informativo de las interrelaciones entre los ambientes, mientras que el simétrico (*SVP* = “*symmetrical*”) permite visualizar la magnitud relativa tanto de la variación de los genotipos como de los ambientes, por lo que se utiliza por defecto. Por último, se indica que los datos no se deben escalar con el parámetro *scaling* = “*none*”.

```
GGE1 <- GGEmodel(yan.winterwheat, genotype = "gen", environment = "env", response = "yield", rep = NULL, centering = "tester", scaling = "none", SVP = "symmetrical")
```

La salida de `GGEModel()` es una lista con los siguientes objetos:

- `coordgenotype`: coordenadas para los genotipos en cada componente.
- `coordenviroment`: coordenadas para los ambientes en cada componente.

-
- `eigenvalues`: vector de autovalores para cada componente.
 - `vartotal`: varianza general.
 - `varexpl`: porcentaje de varianza explicado por cada componente.
 - `labelgen`: nombres de los genotipos.
 - `labelenv`: nombres de los ambientes.
 - `axes`: etiquetas de los ejes.
 - `Data`: datos escalados y centrados.
 - `centering`: método de centrado.
 - `scaling`: método de escala.
 - `SVP`: método de partición.

Utilizando la salida de `GGEmodel()`, la función `GGEPlot()` crea numerosas vistas del biplot GGE que permiten dar respuesta a distintos objetivos de los fitomejoradores. En estos gráficos los cultivares se muestran en minúsculas y los ambientes en mayúsculas. El método de centrado, escalado y SVP se muestran en una nota al pie junto con el porcentaje de $G + IGA$ explicado por los dos ejes al agregar el argumento `footnote = T` y un título con `titles = T`.

Comparaciones simples utilizando GGE biplot

El biplot básico se obtiene con el parámetro `type = "Biplot"` (Figura 4.2). En este ejemplo, el 78 % de la variabilidad de G e IGA se explica por los dos primeros términos multiplicativos. Los ángulos entre los marcadores de genotipos y entre los vectores ambientales son utilizados para interpretar el gráfico. Así, por ejemplo, Kat tiene un rendimiento por debajo de la media en todos los ambientes debido a su ángulo superior a 90° con todos ellos. Por otro lado, Fun presenta un rendimiento superior a la media en todas las localidades excepto OA93 y KE93, como lo indican los ángulos agudos. La longitud de los vectores ambientales es una medida de la capacidad del ambiente para discriminar entre cultivos.


```
GGEPlot(GGE1, type = "Biplot", footnote = F, titles = F)
```

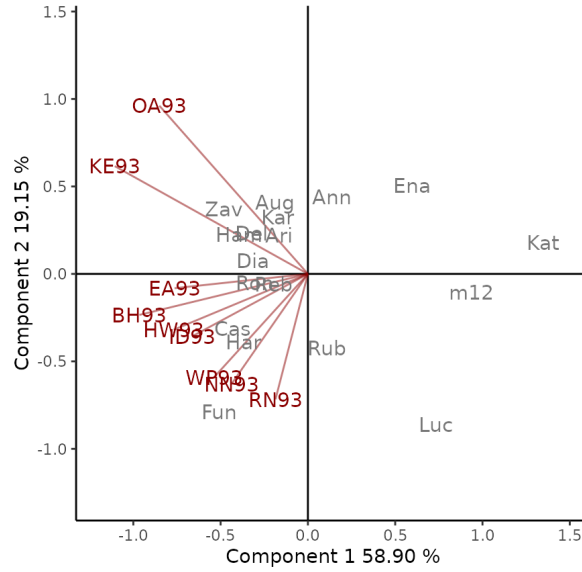


Figura 4.2: Biplot GGE basado en datos de rendimiento de trigo de invierno obtenido de Ontario en 1993. El método de partición de valores singulares utilizado es el simétrico (opción por defecto). El 78 % de la variabilidad de $G + GE$ se explica por los dos primeros términos multiplicativos. Los cultivares se muestran en minúsculas y los entornos en mayúsculas.

Los mejoradores quieren identificar los cultivares más adaptados a su área, es decir a un ambiente particular, por ejemplo OA93. Para esto, Cornelius et al. (2002) sugieren constituir un eje del ambiente de interés (OA93), trazando una recta que una el identificador del ambiente y el origen de coordenadas, y lo denominan eje OA93. Los genotipos se clasifican en función del rendimiento en dicho ambiente de acuerdo con sus proyecciones, en la dirección indicada por el eje OA93 (Figura 4.3 A). Para obtener esta vista del biplot GGE, se indica la opción *Selected Environment* en el argumento *type* de la función y el ambiente a evaluar en el argumento *selectedE*. En este ejemplo, el cultivar de mayor rendimiento fue es Zav seguido por Aug, Ham hasta llegar al genotipo Luc, que es el de menor rendimiento en ese ambiente. El eje perpendicular al del ambiente de interés, separa los genotipos con rendimiento mayor al promedio, de Zav a Cas, de aquellos con valores inferior a la media, de Ema a Luc, en OA93.

En forma similar, el ambiente más adecuado para un cultivar es posible determinarlo graficando una línea que conecte el origen de coordenadas y el marcador del genotipo de interés, por ejemplo Kat, como se muestra en la figura 4.3 B (Cornelius et al., 2002). Los ambientes se clasifican a lo largo del eje del genotipo en la dirección indicada por la fle-

cha. Para obtener este gráfico la opción *Selected Genotype* debe indicarse en el argumento *type*, y el genotipo de interés en *selectedG*. El eje perpendicular al del genotipo separa los ambientes en los que el cultivar presentó un rendimiento por debajo y por encima del promedio. En este ejemplo, Kat presentó un desempeño por debajo de la media en todos los ambientes estudiados.

```
# Ranking de cultivares en el ambiente OA93
```

```
GGEPlot(GGE1, type = "Selected Environment", selectedE = "OA93", footnote = F, titles  
= F)
```

```
# Ranking de ambientes para cultivar Kat
```

```
GGEPlot(GGE1, type = "Selected Genotype", selectedG = "Kat", footnote = F, titles = F)
```

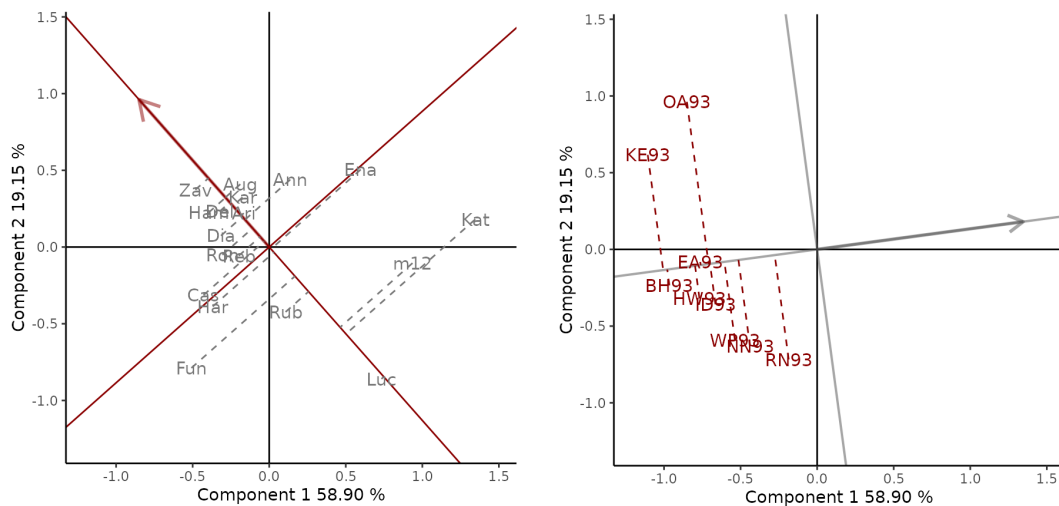


Figura 4.3: A: Ranking de cultivares en el ambiente OA93. B: Ranking de ambientes para cultivar Kat, basado en datos de rendimiento de trigo de invierno obtenido de Ontario en 1993. El método de partición de valores singulares utilizado es el simétrico (opción por defecto). El 78 % de la variabilidad de $G + GE$ se explica por los dos primeros términos multiplicativos. Los cultivares se muestran en minúsculas y los entornos en mayúsculas.

También es posible comparar dos cultivares, por ejemplo Kat y Cas, vinculándolos con una línea y una perpendicular a la anterior (figura 4.4). Este biplot se obtiene con *Comparison of Genotype* en el argumento *type* y los genotipos a comparar en *selectedG1* y *selectedG2*. Cas fue más rendidor que Kat en todos los ambientes, ya que todos se ubican en el mismo lado de la línea perpendicular que Cas.

```
GGEPlot(GGE1, type = "Comparison of Genotype", selectedG1 = "Kat", selectedG2 = "Cas", footnote = F, titles = F)
```

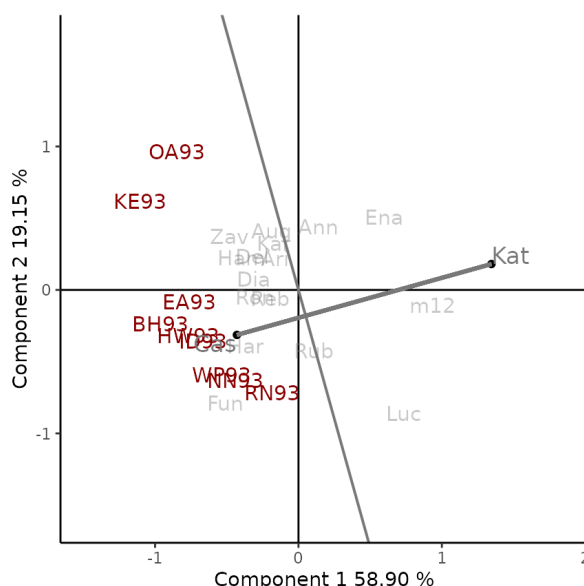


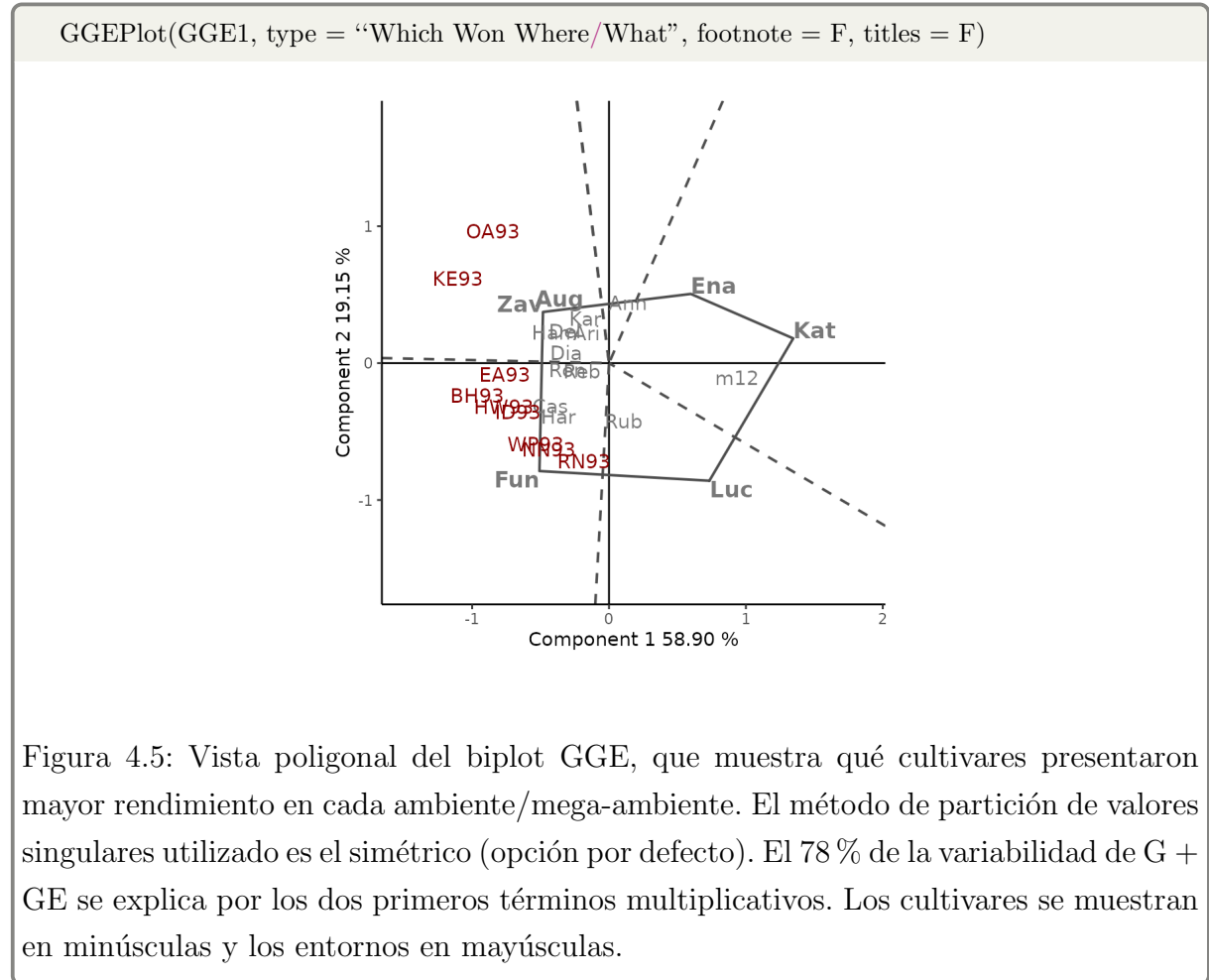
Figura 4.4: comparación de los cultivares Kat y Cas. El método de partición de valores singulares utilizado es el simétrico (opción por defecto). El 78 % de la variabilidad de G + GE se explica por los dos primeros términos multiplicativos. Los cultivares se muestran en minúsculas y los entornos en mayúsculas.

Identificación de mega-ambientes con GGE biplot

La vista poligonal del biplot GGE, obtenida al indicar *Which Won Where/What* en el argumento *type*, proporciona un medio eficaz de visualización del patrón “quién ganó dónde” de un conjunto de datos provenientes de EMA (Figura 4.5). El polígono se obtiene uniendo los cultivares (fun, zav, ena, kat y luc) que se encuentran más alejados del origen de coordenadas, de modo que todos los restantes se encuentren contenidos en el polígono. La distancia de los cultivares respecto del origen de coordenadas, en sus respectivas direcciones, es una medida de la capacidad de respuesta a los ambientes. Los ubicados en los vértices son los más alejados, por lo tanto son los cultivares que más responden, mientras que los que se encuentran en el origen de coordenadas no responden en absoluto a los ambientes estudiados.

Las perpendiculares a los lados del polígono dividen al biplot en mega-ambientes, siendo el cultivar de mayor rendimiento en todos los ambientes que se encuentran en él aquel que se encuentra en el vértice de dicho sector. Por un lado, se observa que OA93 y KE93 conforman un mega-ambiente y que Zav es el mejor cultivar. Otro está formado por el resto de los ambientes, al cual llamaremos ME1 en futuros análisis, siendo Fun el que se

encuentra en el vértice. En el sector con ena, kat y luc en los vértices del polígono no se observó ningún ambiente, lo cual indica que estos cultivares fueron los menos rendidores en algunos o todos los ambientes considerados.



Evaluación de los cultivos dentro de un mega-ambientes con GGE biplot

Una vez identificado los mega-ambientes, el siguiente paso es seleccionar cultivares dentro de cada uno de ellos. De acuerdo con la figura 4.5, zav es el mejor cultivar para los ambientes en uno de los mega-ambiente y fun para el otro. Sin embargo, los fitomejoradores no seleccionarán un único cultivar en cada mega-ambiente, sino que es necesario evaluar todos los cultivares con el fin de conocer su desempeño (rendimiento y estabilidad).

El biplot GGE, particularmente utilizando el factor de partición de la descomposición en valores singulares enfocando en los genotipos, es decir utilizando el argumento $SVP = "row"$ en la función `GGEmodel()`, proporciona un medio superior para visualizar tanto el rendimiento medio como la estabilidad de los genotipos (Figura 4.6). Esto se debe a que la unidad de ambos ejes para los genotipos es la unidad original de los datos. Además, dado que el interés radica en los genotipos y no en los ambientes, se indica con el

argumento `sizeEnv = 0` de la función `GGEPlot()` para que no se los muestre en el gráfico.

La visualización del rendimiento medio y la estabilidad de los genotipos se logra dibujando una coordenada ambiental promedio (AEC, por sus siglas en inglés *Average environment coordination*). Por ejemplo, la Figura 4.6 muestra el AEC para el megaambiente ME1 compuesto por los entornos BH93, EA93, HW93, ID93, NN93, RN93, WP93. Mientras que la abscisa representa el efecto de G la ordenada el de la IGA, que es una medida de la variabilidad o inestabilidad, asociada con cada genotipo. Una mayor proyección sobre la ordenada AEC, independientemente de la dirección, significa mayor inestabilidad. Fun fue claramente el cultivar de mayor rendimiento, en promedio, en este megaambiente, seguido por Cas y Har, y Kat fue el más pobre. Mientras que Rub y Dia son más variables y menos estables que otros cultivares, por el contrario, Cas, Zav, Reb, Del, Ari y Kar, fueron más estables.

La Figura 4.6 compara los cultivares con el “ideal” que es el más rendidor y con estabilidad absoluta. Este cultivar ideal se usa como referencia, ya que rara vez existe. La distancia entre los cultivares y el ideal se puede utilizar como medida de conveniencia. Los círculos concéntricos ayudan a visualizar estas distancias. En el ejemplo, para el ME1, Fun es el más cercano al cultivo ideal, y por tanto el más deseable, seguido de Cas y Har, y Kat fue el más lejano.

FALTA EL SIGNO \$ EN LA PRIMER LINEA DEL CODIGO QUE SIGUE...PERO ME TIRA ERROR O PONERLO CON FILTER???

```
ME1 <- yan.winterwheat[yan.winterwheat env %in% c("BH93", "EA93", "HW93", "ID93",
"NN93", "RN93", "WP93"), ]

# Modelo SREG enfocando SVD en los genotipos
GGE.Gpartition <- GGEmodel(ME1, genotype = "gen", environment = "env", response =
"yield", SVP = "row")

# Visualizacion del rendimiento medio y la estabilidad
GGEPlot(GGE.Gpartition, type = "Mean vs. Stability", footnote = F, titles = F, sizeEnv =
0)

# Ranking de los genotipos respecto a uno ideal
GGEPlot(GGE.Gpartition, type = "Ranking Genotypes", footnote = F, titles = F, sizeEnv
= 0)
```

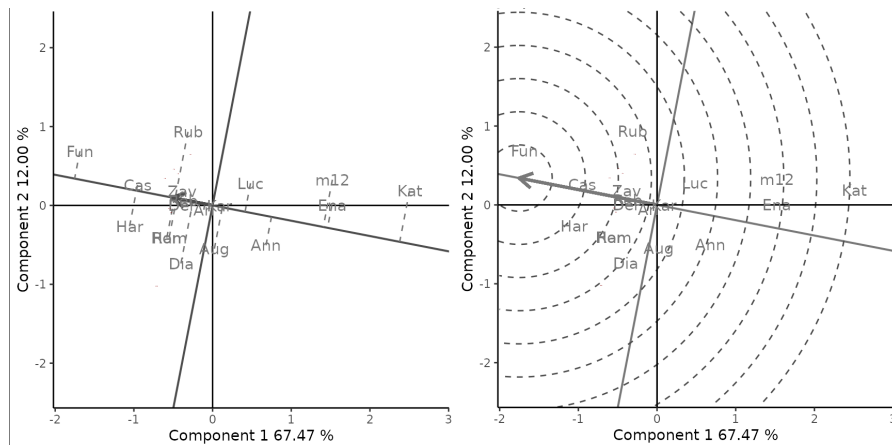


Figura 4.6: A: Evaluación de los cultivares con base en el rendimiento promedio y la estabilidad y B: Clasificación de genotipos con respecto al genotipo ideal, basado en el método de partición de la descomposición en valores singulares enfocado en los genotipos.

Evaluación de los ambientes con GGE biplot

A pesar de que el objetivo principal de los EMA es seleccionar cultivares también es posible evaluar los ambientes. Esto incluye varios aspectos: (i) evaluar si la región objetivo pertenece a uno o más megaambientes; (ii) identificar mejores entornos de prueba; (iii) detectar ambientes redundantes que no brindan información adicional sobre cultivares; y (iv) determinar los ambientes que se pueden utilizar para la selección indirecta. Para ello, se enfoca la partición de los valores singulares en los ambientes al ajustar el modelo SREG ($SVP = "column"$ en la función `GGEmodel()`).

En la figura 4.7 los ambientes están conectados con el origen de coordenadas a través de vectores, permitiendo comprender las interrelaciones entre ellos. Esta visualización del biplot GGE se obtiene indicando *Relationship Among Environments* (Figura 4.7) en el parámetro *type*. El coeficiente de correlación entre dos ambientes es aproximadamente el coseno del ángulo entre sus vectores. En este ejemplo se considera la relación entre los ambientes de ME1. El ángulo entre los vectores para los entornos NN93 y WP93 es de aproximadamente 10° entre sus vectores; por lo tanto, están estrechamente relacionados; mientras que RN93 y OA93 presentan una correlación negativa débil ya que el ángulo es levemente mayor a 90° . El coseno de los ángulos no se traduce precisamente en coeficientes de correlación, ya que el biplot no explica toda la variabilidad en el conjunto de datos. Sin embargo, son lo suficientemente informativos como para comprender la interrelación entre los entornos de prueba.

Si algunos de los ambientes tienen ángulos pequeños y, por lo tanto, están altamente correlacionados, la información sobre los genotipos obtenidos de estos ambientes debe ser similar. Si esta similitud es repetible a través de los años, estos ambientes son redundantes y por lo tanto, uno solo debería ser suficiente. Obtener la misma o mejor información utilizando menos ambientes reducirá el costo y aumentará la eficiencia de producción.

La capacidad de discriminación así como la representatividad respecto del ambiente objetivo, son medidas fundamentales para un ambiente. Si no tiene capacidad de discriminación, no proporciona información sobre los cultivares y, por lo tanto, carece de utilidad. A su vez, si no es representativo no sólo que carece de utilidad sino que también puede proporcionar información sesgada sobre los cultivares evaluados. Para visualizar estas medidas, se define una coordenada ambiental promedio (AEC mencionado anteriormente) y el ambiente ideal como el centro de un conjunto de círculos concéntricos (Figura 4.7). Para obtener este biplot se debe indicar *Ranking Environments* en el argumento *type* de **GGEPlot()** (Figura 4.7). El ángulo entre el vector de un ambiente y el eje proporciona una medida de la representatividad. Por lo tanto, EA93 e ID93 son los más representativos, mientras que RN93 y BH93 son los menos representativos del ambiente promedio, cuando se analiza ME1. Por otro lado, para ser discriminativo debe estar cercano al ambiente ideal. HW93 es el ambiente más cercano al ideal y, por lo tanto, es el más deseable del ME1, seguido por EA93 e ID93. Por el contrario, RN93 y BH93 fueron los ambientes de prueba menos deseables de ME1.

```
# Modelo SREG enfocando SVD en los ambientes
GGE_Epartition <- GGEmodel(ME1, genotype="gen", environment="env", response="yield", SVP="column")

# Relacion entre ambientes
GGEPlot(GGE_Epartition, type = "Relationship Among Environments", footnote = F, titles = F)

# Clasificacion de ambientes con respecto al ambiente ideal
GGEPlot(GGE_Epartition, type = "Ranking Environments", footnote = F, titles = F)
```

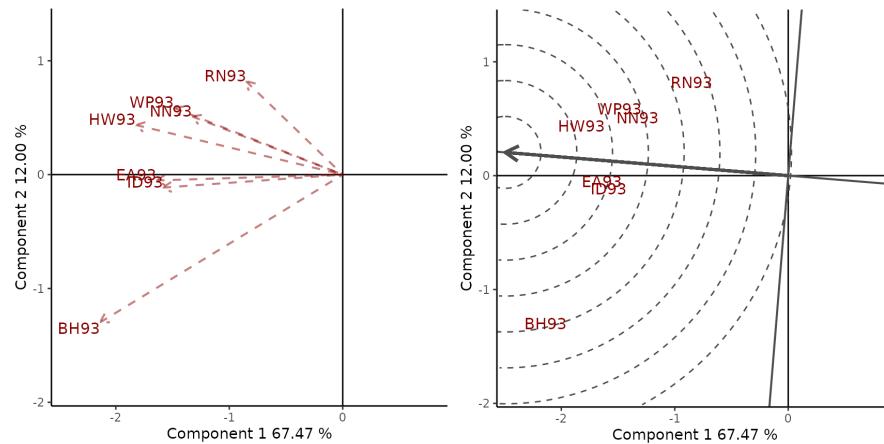


Figura 4.7: A: Relación entre ambientes y B: Clasificación de ambientes con respecto al ambiente ideal, basado en el escalado centrado en los genotipos.

4.1.4. Métodos de imputación

Una limitación importante de los modelos presentados anteriormente es que requieren que el conjunto de datos este completo, es decir que todos los genotipos sean evaluados en todos los ambientes. Por lo tanto, en el paquete se incluyen una serie de metodologías de imputación desarrolladas específicamente para datos genotipo-ambiente recientemente publicadas, algunas de las cuales no se encuentran disponible en R, para superar el problema de las observaciones perdidas. Entre los métodos incluidos se encuentran: “EM-AMMI”, “EM-SVD”, “Gabriel”, “WGabielz”, “EM-PCA”, los cuales se indican en la opción *type* de la función `imputation()`. El formato requerido para el conjunto de datos de entrada es análogo al indicado en las otras funciones incluidas en el paquete.

Para presentar un ejemplo, se eliminan algunas observaciones del conjunto de datos *yan.winterwheat* ya que contaba con todos los registros completos:


```
# Generando datos faltantes
yan.winterwheat [1,3] <- NA
yan.winterwheat [3,3] <- NA
yan.winterwheat [2,3] <- NA
```

La imputación de valores perdidos con el método “EM-AMMI” se puede realizar de la siguiente manera:

```
imputation(yanwinterwheat, PC.nb = 2, genotype = “gen”, environment = “env”, response
  = “yield”, type = “EM-AMMI”)
```

El resultado es la matriz con datos imputados en aquellas celdas vacías.

4.2. Geneticae Shiny Web App

El objetivo de Geneticae Shiny Web APP es proporcionar una interfaz gráfica de usuario para el paquete *geneticae* de R descrito anteriormente, de modo que pueda ser utilizado por fitomejoradores y analistas sin experiencia previa en programación R.

Es un software interactivo, no comercial y de código abierto, que ofrece una alternativa gratuita al software comercial disponible para analizar datos provenientes de ensayos multiambientales. Se encuentra disponible en un servidor gratuito <https://geneticae.shinyapps.io/geneticae-shiny-web-app/> el cual tiene límite en el tiempo de uso, sin embargo, la APP será ubicada en el servidor de CONICET para su libre navegación. Además, se puede acceder a la misma desde la página web del instituto CEFOTI de CONICET <https://www.cefobi-conicet.gov.ar/bases-de-datos-y-programas/>.

En las subsecciones siguientes se presentará un ejemplo de cómo cargar y analizar datos con la aplicación.

4.2.1. Preparación de un archivo de datos

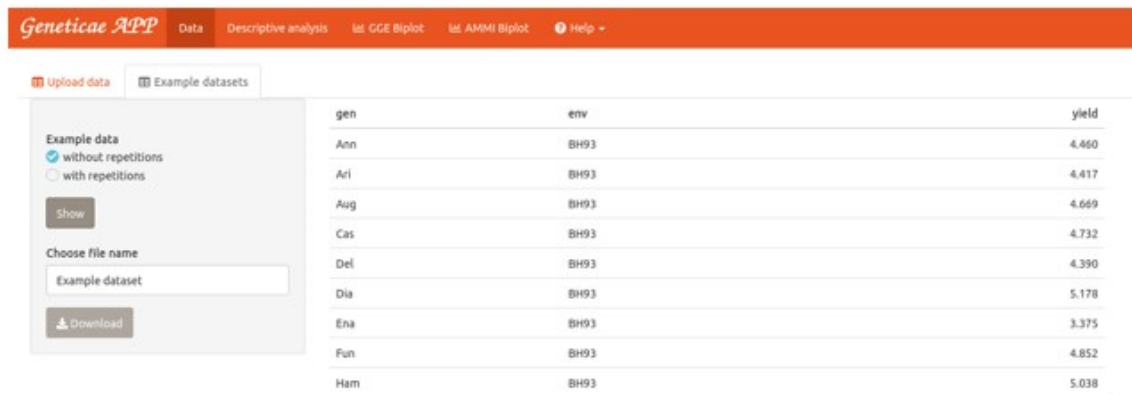
La APP requiere que los datos de entrada se encuentren en formato .csv, delimitados por comas, punto y coma o tabulaciones. Los nombres de las columnas pueden ubicarse en la primera fila del archivo (*heading*). Los datos deben estar en formato largo, es decir, cada fila corresponde a una observación y cada columna a una variable (genotipo, ambiente, repetición (si existe) y fenotipo observado). Si cada genotipo ha sido evaluado más de una vez en cada ambiente, la media fenotípica requerida por el modelo SREG y AMMI para cada combinación de genotipo y ambiente se calcula internamente antes de ajustar dichos modelos. Las variables adicionales que no se utilizarán en el análisis pueden estar

presentes en el conjunto de datos. No se permiten valores perdidos.

Los dos conjuntos de datos *plrv* y *yanwinterwheat* descriptos en la subsección 4.1.1 están disponibles en la pestaña *Data* – > *Example datasets* y se pueden descargar en formato .csv (Figura 4.8) para poder seguir el tutorial de uso de la APP. El conjunto de datos *yanwinterwheat* no tiene repeticiones, mientras que *plrv* sí.

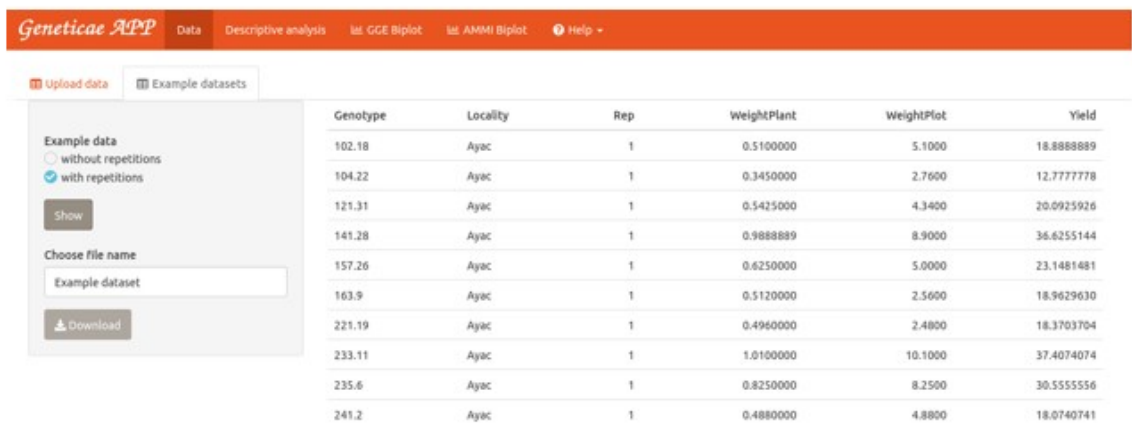
SE VE HORRIBLE, CAMBIAR CAPTURA

A



gen	env	yield
Ann	BH93	4.460
Ari	BH93	4.417
Aug	BH93	4.669
Cas	BH93	4.732
Del	BH93	4.390
Dia	BH93	5.178
Ena	BH93	3.375
Fun	BH93	4.852
Ham	BH93	5.038

B



Genotype	Locality	Rep	WeightPlant	WeightPlot	Yield
102.18	Ayac	1	0.5100000	5.1000	18.8888889
104.22	Ayac	1	0.3450000	2.7600	12.7777778
121.31	Ayac	1	0.5425000	4.3400	20.0925926
141.28	Ayac	1	0.9888889	8.9000	36.6255144
157.26	Ayac	1	0.6250000	5.0000	23.1481481
163.9	Ayac	1	0.5120000	2.5600	18.9629630
221.19	Ayac	1	0.4960000	2.4800	18.3703704
233.11	Ayac	1	1.0100000	10.1000	37.4074074
235.6	Ayac	1	0.8250000	8.2500	30.5555556
241.2	Ayac	1	0.4880000	4.8800	18.0740741

Figura 4.8: Conjunto de datos (A) *plrv* (B) *yanwinterwheat* disponible en Geneticae Shiny Web APP

En los ejemplos que continúan se utilizará el conjunto de datos *yanwinterwheat*.

Cargando un conjunto de datos en la APP

El conjunto de datos que se analizará debe cargarse en la pestaña *Data* – > *Upload data*. Por ejemplo, para importar el conjunto de datos *yanwinterwheat*, se debe cargar el archivo .csv. Una vez cargado, se debe indicar que está delimitado por comas, que la primera fila contiene los nombres de cada variable (*heading*) y los nombres de las

columnas que contienen la información del genotipo, ambiente y rasgo fenotípico (gen, env y rendimiento en este ejemplo). Si hay repeticiones disponibles, se debe especificar el nombre de la columna con dicha información; de lo contrario, el campo queda vacío.

The screenshot shows the Geneticae APP interface. The sidebar on the left has two tabs: 'Upload data' and 'Example datasets'. Under 'Upload data (.csv format)', there are buttons for 'Browse...' and 'Example without repetition', and an 'Upload complete' button. Below these are checkboxes for 'Heading in first row' (checked) and 'Values separated by' (set to 'Comma'). There are also sections for selecting column names for 'Genotypes', 'Environments', 'Repetitions', and 'Phenotype', each with checkboxes for 'gen', 'env', and 'yield'. The main area shows a table with 10 entries (showing 1 to 10 of 162 entries). The table has columns for 'gen', 'env', and 'yield'. The data is as follows:

gen	env	yield
Ann	BH93	4.46
Ari	BH93	4.417
Aug	BH93	4.669
Cas	BH93	4.732
Del	BH93	4.39
Dia	BH93	5.178
Ena	BH93	3.375
Fun	BH93	4.852
Ham	BH93	5.038
Har	BH93	5.195

Figura 4.9: Importar el conjunto de datos *yanwinterwheat* en Geneticae Shiny Web APP

4.2.2. Análisis descriptivo

Cualquier estudio debe comenzar con un análisis descriptivo del conjunto de datos. La pestaña *Descriptive Analysis* proporciona algunas herramientas para esto, como *boxplot*, diagrama y matriz de correlación así como también gráficos de interacción.

Un *boxplot* que compara el rasgo cuantitativo entre ambientes o genotipos puede ser de interés (Figura 4.10). Las medidas de resumen utilizadas para su construcción se muestran de forma interactiva moviendo el mouse dentro del panel de la figura. Además, se puede descargar como un archivo .png o en forma interactiva (.html), haciendo clic en la cámara que aparece en la figura o en el botón Descargar, respectivamente. El usuario puede personalizar algunos aspectos del gráfico, como el color de las cajas y los nombres de los ejes.

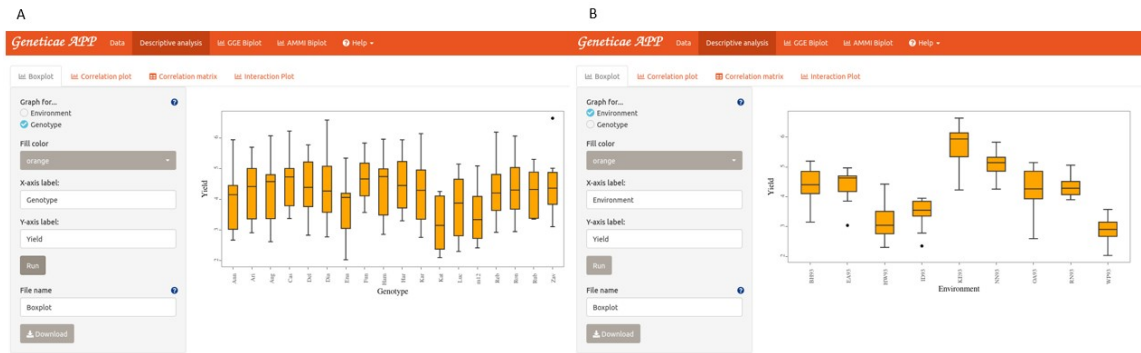


Figura 4.10: Diagrama de caja de (A) genotipos y (B) ambientes para el conjunto de datos *yanwinterwheat*.

Los coeficientes de correlación de Pearson o Spearman entre genotipos se pueden mostrar como un gráfico o una matriz (Figura 4.11). Gráficamente, las correlaciones positivas se muestran en azul y las negativas en rojo, mientras que la intensidad del color y el tamaño del círculo son proporcionales a la magnitud de los coeficientes de correlación. La gráfica de correlación se puede descargar en formato .png. Se observan altas correlaciones entre el rendimiento de los genotipos estudiados.

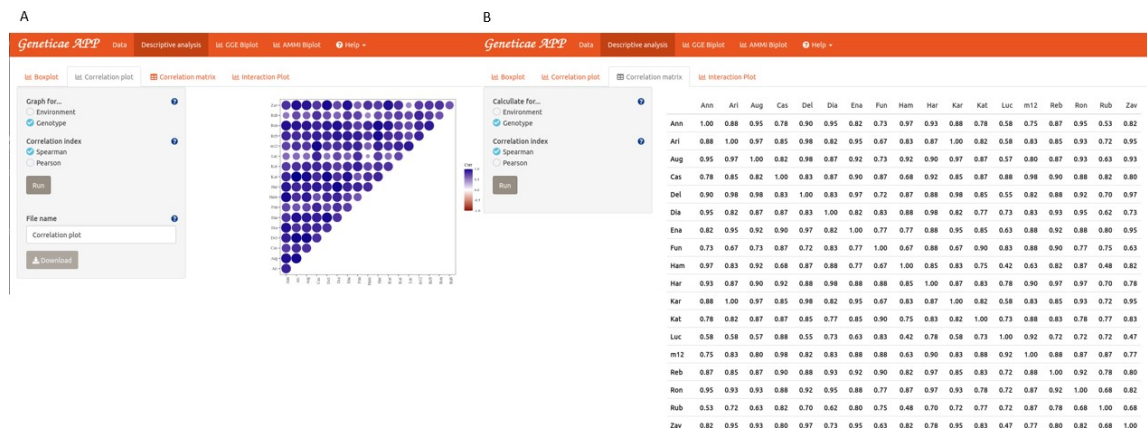


Figura 4.11: Gráfico de correlación (A) y matriz (B) entre genotipos para conjunto de datos *yanwinterwheat*

Dado que IGA genera respuestas genotípicas diferenciales en diferentes ambientes, lo que complica selección de los mejores cultivares, una gráfico de interacción puede ser de interés (Figura 4.12). El cambio en el efecto genotípico a través de los ambientes se muestra en la figura 4.12 A, mientras que el cambio en el efecto ambiental a través de los genotipos en la figura 4.12 B. Del mismo modo que el *boxplot* es un gráfico interactivo,

y además los gráficos pueden ser descargados.

El biplot básico, la vista del biplot GGE que muestra los cultivares más adecuados para un ambiente particular (OA93), los ambientes más adecuados para un genotipo (Kat), la comparación de dos genotipos (Cas y Kat) y la vista del polígono se pueden obtener como se indica en la figura 4.13, donde el escalado es el simétrico (*SVP type* → *symmetrical*) y las opciones de *plot type* son *Biplot*, *Selected Environment*, *Selected Genotype*, *Comparison of Genotype* y *Which Won Where/What*, respectivamente. Al indicar *Selected Environment* el ambiente de interés se debe especificar, de igual modo cuando se utiliza *Selected Genotype* y *Comparison of Genotype* se debe señalar cuál es el genotipo a analizar.



Figura 4.13: Vistas del biplot GGE usando la partición en valores singulares simétrica.

La selección de cultivares dentro de cada megaambiente se realiza con la partición de valores singulares enfocada en los genotipos (*SVP type* → *genotype-focused*), y los tipos de gráficos que se pueden realizar son: *Mean vs. Stability* que permite la visualización de la media y estabilidad de genotipos y *Ranking Genotypes* que compara las cultivares con el “ideal” (Figura 4.14). Dado que estos análisis son propios de cada megaambiente, al indicar

alguno de estas vistas del biplot GGE se tendrá que señalar cuales son los ambientes que forman el megaambiente de interés.

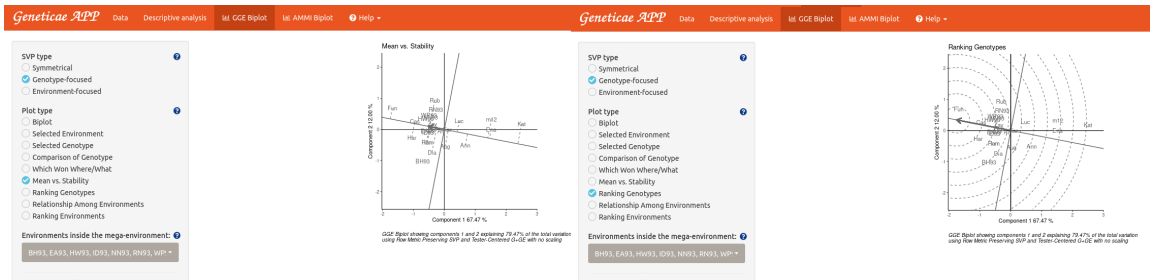


Figura 4.14: Vistas del biplot GGE usando la partición de valores singulares enfocada en los genotipos.

Por último, para el análisis de los ambientes de cada megaambiente se utiliza el método de partición de valores singulares centrado en los ambientes (*SVP type* → *environment-focused*). Para comprender las interrelaciones entre ellos el tipo de gráfico *Relationship Among Environments* se debe seleccionar y para visualizar la capacidad de discriminación y representatividad *Ranking Environments* (Figura 4.15). Dado que estos análisis son propios de cada megaambiente, al indicar alguno de estas vistas del biplot GGE se tendrá que señalar cuales son los ambientes que forman el megaambiente de interés.

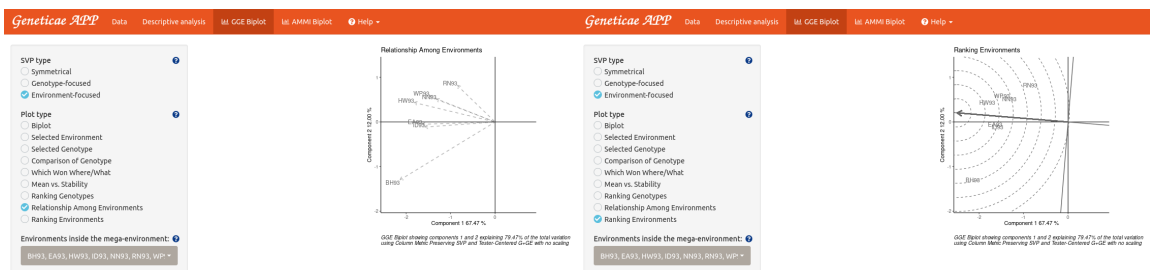


Figura 4.15: Vistas del biplot GGE usando la partición de valores singulares enfocada en los ambientes.

4.2.4. modelo AMMI

La pestaña *AMMI Biplot* crea el biplot GE, en el que los cultivares se muestran en minúsculas y los entornos en mayúsculas. Dado que las alternativas clásica y robustas requieren una única observación para cada combinación de genotipo y ambiente, si hay repeticiones, el valor promedio fenotípico se calcula automáticamente antes de ajustar el modelo. No se permiten valores perdidos. Al igual que en el biplot de GGE, una nota

a pie de página que indica que el porcentaje de variación de IGA explicado por los dos ejes, el gráfico de título, los ejes y los nombres de los mismos se pueden configurar para que aparezcan o no. Además, el color y tamaño del marcador de genotipos y ambientes pueden ser personalizados por el usuario. Los biplots pueden ser descargados.

Por ejemplo, para obtener el biplot GE derivado del modelo AMMI clásico se debe indicar AMMI en *plot type* (Figura ...). En caso de contar con *outliers* alguna de las alternativas robustas (rAMMI, hAMMI, gAMMI, lAMMI o ppAMMI) se debe especificar.

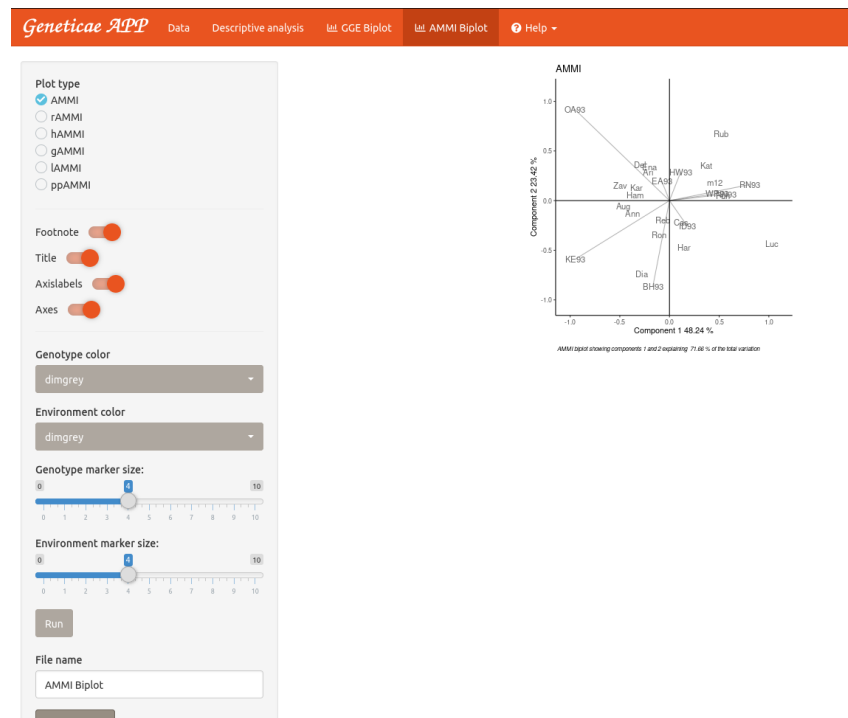


Figura 4.16: Biplot GE obtenido del modelo AMMI clásico basado en datos de rendimiento de trigo de invierno obtenido de Ontario en 1993.

4.2.5. Ayuda

En la pestaña *Help* se presenta información general, un tutorial y un video sobre cómo utilizar la APP.

Capítulo 5

Conclusión

En etapas avanzadas de los programas de mejoramiento vegetal, comúnmente se llevan a cabo ensayos multiambientales (EMA) donde un conjunto de variedades se evalúan en múltiples ambientes. Un análisis adecuado de la información de los EMA es indispensable para el éxito del programa de mejoramiento genético de los cultivos. Dado que, la metodología utilizada se encuentra en constante desarrollo, muchas de ellas no se encuentran disponibles en programas comerciales.

En este trabajo se muestra un flujo de trabajo reproducible para la construcción de paquetes de R que puede utilizarse de ejemplo para el desarrollo de nuevos paquetes. En particular se creó el paquete de R *geneticae* que es de gran utilidad para el análisis de datos provenientes de EMA por incluir metodología recientemente publicada, además de la reunir las funciones más útiles, y que a pesar del poco tiempo transcurrido desde la publicación ya tiene más de 400 descargas.

Por otro lado, dado que el uso del software puede resultar dificultoso para aquellos analistas no familiarizados con la programación, se crea una aplicación web Shiny de libre acceso mediante conexión a internet que permite realizar los principales análisis implementados en el paquete sin necesidad de escribir líneas de código.

Se plantea para un futuro, continuar con la inclusión de metodología frecuentemente utilizada así como aquellas que se vayan desarrollando en el contexto de datos provenientes de EMA tanto en el paquete como en la aplicación web Shiny.

Bibliografía

- F. Aguater, J. Crossa, y M. Balzarini. Effect of missing values on variance component estimates in multi-environment trials. *Crop Science*, 59, 2019.
- S. Arciniegas-Alarcón, M. García-Peña, C.T.S. Dias, y W.J. Krzanowski. An alternative methodology for imputing missing data in trials with genotype-by-environment interaction. *Biometrical Letters*, 47:1–47, 2010.
- S. Arciniegas-Alarcón, M. García-Peña, W.J. Krzanowski, y C.T.S. Dias. An alternative methodology for imputing missing data in trials with genotype-by-environment interaction: some new aspects. *Biometrical Letters*, 51:75–88, 2014.
- S. Arciniegas-Alarcón, M. García-Peña, y P. C. Rodrigues. New multiple imputation methods for genotype-by-environment data that combine singular value decomposition and jackknife resampling or weighting schemes. *Computers and Electronics in Agriculture*, 176, 2020.
- R.E. Cooper, M. and Stucker, I.H. DeLacy, y B.D. Harch. Wheat breeding nurseries, target environments, and indirect selection for grain yield. *Crop Science*, 37:1168–1176, 1997.
- P.L. Cornelius, J. Crossa, y M.S. Seyedsadr. *Genotype by Environment Interaction*, cap. Statistical test and estimators of multiplicative models for genotype-by-environment interaction., págs. 199–234. CRC Press, Boca Raton, 1996.
- P.L. Cornelius, J. Crossa, y M.S. Seyedsadr. *Quantitative Genetics, Genomics and Plant Breeding.*, cap. Biplot analysis of Multi-environment trial data., págs. 199–234. CABI Publishing, Wallingford, 2002.
- J. Crossa y P. L. Cornelius. Sites regression and shifted multiplicative model clustering of cultivar trial sites under heterogeneity of error variances. *Crop Science*, 37:406–415, 1997.
- J. Crossa, H.G. Gauch, y R.W. Zobel. Additive main effects and multiplicative interaction analysis of two international maize cultivar trials. *Crop Science*, 30:493–500, 1990.

-
- R. Cruz Medina. Some exact conditional tests for the multiplicative models to explain genotype-environment interaction. *Heredity*, 69:128—132, 1992.
- F. de Mendiburu. *agricolae: Statistical Procedures for Agricultural Research*, 2020. URL <https://CRAN.R-project.org/package=agricolae>. R package version 1.3-3.
- L. A. de Oliveira, C. P. da Silva, J. J. Nuvunga, A. Q. da Silva, y M. Balestre. Bayesian gge biplot models applied to maize multi-environments trials. *Genetics and Molecular Research*, 15, 2016.
- S. Dumble. *GGEbiplots: GGE Biplots with 'ggplot2'*, 2017. URL <https://CRAN.R-project.org/package=GGEbiplots>. R package version 0.1.1.
- C. Ganz, G. Csárdi, J. Hester, M. Lewis, y R. Tatman. *available: Check if the Title of a Package is Available, Appropriate and Interesting*, 2019. URL <https://CRAN.R-project.org/package=available>. R package version 1.0.4.
- H. G. Gauch. Model selection and validation for yield trials. *Theoretical and Applied Genetics*, 80:153–160, 1988.
- H.G. Gauch y R.W. Zobel. Imputing missing yield trial data. *Theoretical and Applied Genetics*, 79:753–761, 1990.
- H.G. Gauch y R.W. Zobel. Identifying mega-environments and targeting genotypes. *Crop Science*, 37:311—326, 1997.
- S. Hadasch, J. Forkman, W.A. Malik, y H.P. Piepho. Weighted estimation of ammi and gge models. *Journal of Agricultural, Biological and Environmental Statistics*, 23:255–27, 2018.
- J. Hester. *covr: Test Coverage for Packages*, 2020. URL <https://CRAN.R-project.org/package=covr>. R package version 3.5.0.
- Jr. R.R. Hill y J.L. Rosenberg. Models for combining data from germplasm evaluation trials. *Crop Science*, 25:467–470, 1985.
- P. J. Huber. *Robust Statistics*. John Wiley and Sons, 1981.
- D. Jarquín, S. Pérez-Elizalde, J. Burgueño, y J. Crossa. A hierarchical bayesian estimation model for multi-environment plant breeding trials in successive years. *Crop Science*, 56:2260–2276, 2016.
- J. Josse y F. Husson. missMDA: A package for handling missing values in multivariate data analysis. *Journal of Statistical Software*, 70(1):1–31, 2016. doi:10.18637/jss.v070.i01.

-
- M.S. Kang y R. Magari. *Genotype by Environment Interaction*, cap. New Developments in Selecting for Phenotypic Stability in Crop Breeding., págs. 201–213. Elsevier, New York, 1996.
- R. A. Kempton. The use of biplot in interpreting variety by environment interactions. *Journal of Agricultural Science*, 122:335–342, 1984.
- P.O. Perry. *bcv: Cross-Validation for the SVD (Bi-Cross-Validation)*, 2015. R package version 1.0.1.
- R. c. Peto. *Treatment of cancer*, cap. Statistical aspects of cancer trials, págs. 867–871. Chapman and hall, London, 1982.
- P.C. Rodrigues, A. Monteiro, , y V.M. Lourenço. A robust ammi model for the analysis of genotype-by-environment data. *Bioinformatics*, 32:58–66, 2016.
- H. Wickham. testthat: Get started with testing. *The R Journal*, 3:5–10, 2011. URL https://journal.r-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf.
- H. Wickham y J. Bryan. *usethis: Automate Package and Project Setup*, 2021. URL <https://CRAN.R-project.org/package=usethis>. R package version 2.0.1.
- H. Wickham, P. Danenberg, G. Csárdi, y M. Eugster. *roxygen2: In-Line Documentation for R*, 2020. URL <https://CRAN.R-project.org/package=roxygen2>. R package version 7.1.1.
- H. Wickham, J. Hester, y W. Chang. *devtools: Tools to Make Developing R Packages Easier*, 2021. URL <https://CRAN.R-project.org/package=devtools>. R package version 2.4.2.
- L. G. Woyann, G. Benin, L. Storck, D. M. Trevizan, C. Meneguzzi, V. S. Marchioro, M. Tonnatto, y A. Madureira. Estimation of missing values affects important aspects of gge biplot analysis. *Crop Science*, 57:40–52, 2017.
- K. Wright. *agridat: Agricultural Datasets*, 2020. URL <https://CRAN.R-project.org/package=agridat>. R package version 1.17.
- W. Yan, P.L. Cornelius, J. Crossa, y L.A. Hunt. Two types of gge biplots for analyzing multi-environment trial data. *Crop Science*, 41:656–663, 2001.
- W. Yan y L. A. Hunt. Genetic and environment causes of genotype by environment interaction for winter wheat yield in ontario. *Crop Science*, 41:19–25, 2001.

-
- W. Yan, L. A. Hunt, Q. Sheng, y Z. Szlavnic. Cultivar evaluation and mega-environment investigation based on the GGE biplot. *Crop Science*, 40:597—605, 2000.
- W. Yan y M. Kang. *GGE Biplot Analysis: A Graphical Tool for Breeders, Geneticists*. CRC Press, 2003.
- W. Yan y I. Rajcan. Biplot evaluation of test sites and traitrelations of soybean in ontario. *Crop Science*, 42:11—20, 2002.