

# Package ‘agrobox’

November 19, 2025

**Type** Package

**Title** Data Visualization and Statistical Tools for Agroindustrial Experiments

**Version** 0.1.0

**Maintainer** Joaquin Alejandro Salinas Angeles <joaquinsa03@gmail.com>

**Description**

Set of functions to create clear graphics and run common statistical analyses for agricultural experiments (ANOVA with post-hoc tests such as Tukey HSD and Duncan MRR, coefficient of variation, and simple power calculations), streamlining exploratory analysis and reporting. Functions build on 'ggplot2' and base 'stats' and follow methods widely used in agronomy (field trials, plant breeding). Key references include Tukey (1949) <[doi:10.2307/3001913](https://doi.org/10.2307/3001913)>, Duncan (1955) <[doi:10.2307/3001478](https://doi.org/10.2307/3001478)>, Cohen (1988, ISBN:9781138892899); see also 'agricolae' <<https://CRAN.R-project.org/package=agricolae>> and Wickham (2016, ISBN:9783319242750) for 'ggplot2'. Versión en español: Conjunto de funciones para generar gráficos claros y ejecutar análisis habituales en ensayos agrícolas (ANOVA con pruebas post-hoc como Tukey HSD y Duncan MRR, coeficiente de variación y cálculos simples de potencia), facilitando el análisis exploratorio y la elaboración de reportes. Los métodos implementados se basan en Tukey (1949) <[doi:10.2307/3001913](https://doi.org/10.2307/3001913)>, Duncan (1955) <[doi:10.2307/3001478](https://doi.org/10.2307/3001478)> y Cohen (1988, ISBN:9781138892899); ver también 'agricolae' <<https://CRAN.R-project.org/package=agricolae>> y Wickham (2016, ISBN:9783319242750) para 'ggplot2'.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Language** en, es

**Depends** R (>= 3.5.0)

**Imports** dplyr, ggplot2, tidyr, stringr, agricolae, stats, pwr, rlang

**Suggests** testthat (>= 3.0.0)

**Config/testthat.edition** 3

**RoxygenNote** 7.3.3

**URL** <https://github.com/Joa3aquin50/agrobox>

**BugReports** <https://github.com/Joa3aquin50/agrobox/issues>

**NeedsCompilation** no

**Author** Joaquin Alejandro Salinas Angeles [aut, cre] (ORCID:  
[<https://orcid.org/0009-0005-6174-9714>](https://orcid.org/0009-0005-6174-9714))

**Repository** CRAN

**Date/Publication** 2025-11-19 19:40:02 UTC

## Contents

agrobox	2
<b>Index</b>	5

---

agrobox	<i>Genera un grafico y analisis estadistico por grupo</i>
---------	---

---

### Description

Esta funcion realiza analisis estadisticos por grupos definidos (clusteres) usando ANOVA y pruebas post-hoc (Duncan o Tukey), y genera un grafico tipo boxplot con etiquetas de medias, letras de significancia, coeficiente de variacion (CV) y potencia estadistica (Power).

### Usage

```
agrobox(
  data,
  test = c("Duncan", "Tukey"),
  factor,
  factor2 = NULL,
  orden_factor = NULL,
  bloque = NULL,
  variable,
  niveles_factor = NULL,
  titulo = NULL,
  estructura = NULL,
  lim_sup = NULL,
  lim_inf = NULL,
  colores = NULL
)
```

### Arguments

data	Un data frame que contiene los datos.
test	Tipo de prueba post-hoc a usar: ““Duncan”“ (por defecto) o ““Tukey”“.
factor	Variable categorica principal (factor 1 o tratamiento).

factor2	(Opcional) Segundo factor para interaccion (usado en ANOVA).
orden_factor	(Opcional) Vector con el orden deseado para los niveles del ‘factor’.
bloque	(Opcional) Variable de bloque para disenos con bloques completos.
variable	Variable numerica dependiente.
niveles_factor	(Opcional) Etiquetas personalizadas para los niveles del ‘factor’.
titulo	Etiqueta para el eje Y del grafico (por ejemplo: "Altura (cm)").
estructura	(Opcional) Formula tipo ‘Grupo1 ~ Grupo2‘ para definir los clusteres.
lim_sup	(Opcional) Limite superior para el eje Y.
lim_inf	(Opcional) Limite inferior para el eje Y.
colores	Vector de colores para los niveles del ‘factor’.

### Value

Una lista con:

- plot** Objeto ‘ggplot’ con el grafico generado.
- levels** Niveles del factor utilizados.

### Examples

```
# Ejemplo de un experimento factorial con bloques y estructura bifactorial
library(dplyr)
library(tidyr)
library(ggplot2)

set.seed(123)

# Diseno factorial con bloques (ej. 4 bloques por Localidad-Variedad)
df_experimento <- expand.grid(
  Fertilizante = c("A", "B", "C"),
  Dosis = c("Baja", "Media", "Alta"),
  Localidad = c("Loc1", "Loc2"),
  Variedad = c("Var1", "Var2", "Var3"),
  Bloque = paste0("B", 1:4) # Bloques, no repeticiones
)

# Simular rendimiento (tn/ha) con efectos reales
df_experimento$tn_ha <- 20 +
  ifelse(df_experimento$Fertilizante == "B", 2,
         ifelse(df_experimento$Fertilizante == "C", 4, 0)) +
  ifelse(df_experimento$Dosis == "Media", 1,
         ifelse(df_experimento$Dosis == "Alta", 2, 0)) +
  ifelse(df_experimento$Localidad == "Loc2", 0.5, 0) +
  ifelse(df_experimento$Variedad == "Var2", 0.5,
         ifelse(df_experimento$Variedad == "Var3", -0.5, 0)) +
  rnorm(nrow(df_experimento), mean = 0, sd = 1.2)

# Ejecutar la funcion agrobox
agrobox(
```

```
data = df_experimento,      # La data que usaremos
test = "Duncan",          # Seleccionamos el test que deseamos utilizar,
#puede ser Tukey o Duncan
factor = "Fertilizante",   # Si tenemos un factorial aca colocamos el factor
#que nos interesa mostrar,
#suponiendo que previamente ya demostramos que no hay interaccion.
factor2 = "Dosis",         # Si tenemos un factorial este es el segundo factor,
#sino es un factorial solo se coloca NULL
orden_factor = c("A", "C", "B"), # Ordenamos los niveles del factor principal
#si fuera necesario, sino solo colocamos NULL
bloque = "Bloque",         # En campo siempre se deben tener bloques.
variable = "tn_ha",         # Es la variable respuesta
niveles_factor = c("A" = "Nitrato de amonio",
                   "B" = "Fosfato diamonico",
                   "C" = "Sulfato de amonio"), # Si deseamos cambiar de nombre a los niveles del
#factor principal aca podemos hacerlo, de lo contrario colocar NULL
titulo = "Rendimiento (tn/ha)", # El titulo del grafico.
estructura = "Localidad~Variedad", # Aca se colocan los grupo que deseamos evaluar,
#se realizara un ANOVA y un TEST POST-HOC por cada grupo.
lim_sup = NULL, lim_inf = NULL,    # podemos colocar los limites del eje Y si deseamos.
colores = c("A" = "#1b9e77",
           "B" = "#d95f02",
           "C" = "#7570b3")        # Se coloca los colores de cada nivel del factor principal.
) $plot                         # Recordemos que la funcion nos da una lista,
#de la cual si deseamos obtener el plot debemos seleccionarlo de esta forma.
```

# **Index**

agrobox, [2](#)