Introduction to the `TukeyC` Package

Multiple Comparisons Using the Tukey HSD Algorithm

Authors

J. C. Faria
E. G. Jelihovschi
I. B. Allaman

Institution

Universidade Estadual
de Santa Cruz — UESC
Ilhéus, Bahia, Brasil

May 24, 2026

Contents

Overview

The **TukeyC** package implements Tukey's Honestly Significant Difference (HSD) test as a multiple comparison method in the context of Analysis of Variance (ANOVA). The package follows the conventional approach widely used in experimental statistics: treatment means are compared using the Studentized range, then labelled with letters that may overlap when means are not significantly different.

The result is an easy-to-read table and plot showing mean estimates, minimum significant differences (MSD), and optional matrices of pairwise differences and adjusted p-values.

```
library(TukeyC)
```

1 Quick Start — Completely Randomized Design (CRD)

`CRD1` contains simulated data for a balanced CRD with **4 treatment levels** and **6 replicates** per treatment. The main function `TukeyC()` accepts a model formula, an `aov` object, or an `lm` object. The `which` argument names the factor to be compared.

```
data(CRD1)

tk1 <- with(CRD1,
            TukeyC(y ~ x,
               data = dfm,
               which = 'x'))
summary(tk1)
```

```
#> Groups of means at sig.level = 0.05
#>      Means G1 G2
#> tr-2 59.82  a
#> tr-3 54.11  a
#> tr-1 52.02  a
#> tr-4 40.19     b
#>
#> Matrix of the difference of means above diagonal and
#> respective p-values of the Tukey test below diagonal values
#>       tr-2  tr-3  tr-1   tr-4
#> tr-2 0.000 5.708 7.802 19.627
#> tr-3 0.386 0.000 2.093 13.918
#> tr-1 0.150 0.932 0.000 11.825
#> tr-4 0.000 0.004 0.015  0.000
```

The summary shows, for each level, the mean and the group letter assigned by the algorithm. Levels sharing the same letter do not differ significantly at the default 5 % level.

A single call to `plot()` produces the canonical dot plot with group letters displayed above each point:

```
plot(tk1,
     dispersion = 'mm',
     d.col = 'steelblue')
```

2 Accepted Input Classes

`TukeyC()` dispatches on the class of its first argument. The same grouping can be obtained from a `formula`, an `aov` object, or an `lm` object.

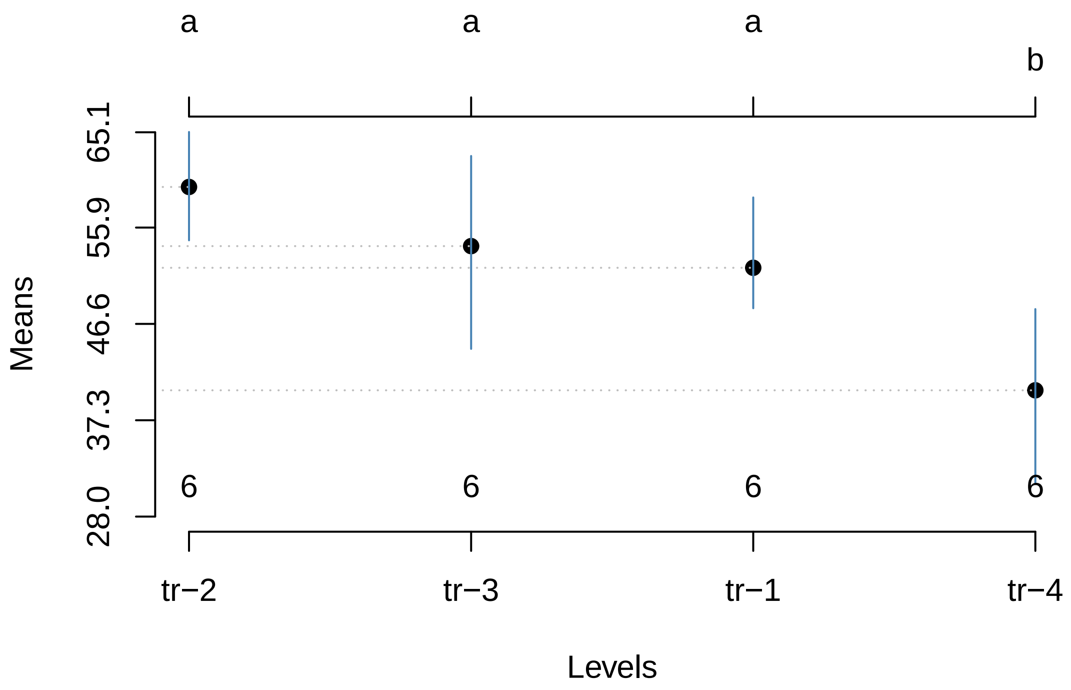

Figure 1: CRD1: treatment means with min-max dispersion bars and TukeyC groups.

```
## From: aov
av1 <- with(CRD1, aov(y ~ x, data = dfm))
tk2 <- TukeyC(av1, which = 'x')
summary(tk2)
```

```
#> Groups of means at sig.level = 0.05
#>      Means G1 G2
#> tr-2 59.82  a
#> tr-3 54.11  a
#> tr-1 52.02  a
#> tr-4 40.19     b
#>
#> Matrix of the difference of means above diagonal and
#> respective p-values of the Tukey test below diagonal values
#>       tr-2  tr-3  tr-1   tr-4
#> tr-2 0.000 5.708 7.802 19.627
#> tr-3 0.386 0.000 2.093 13.918
#> tr-1 0.150 0.932 0.000 11.825
#> tr-4 0.000 0.004 0.015  0.000
```

```
## From: lm
lm1 <- with(CRD1, lm(y ~ x, data = dfm))
tk3 <- TukeyC(lm1, which = 'x')
summary(tk3)
```

```
#> Groups of means at sig.level = 0.05
#>      Means G1 G2
```

```
#> tr-2 59.82  a
#> tr-3 54.11  a
#> tr-1 52.02  a
#> tr-4 40.19     b
#>
#> Matrix of the difference of means above diagonal and
#> respective p-values of the Tukey test below diagonal values
#>       tr-2  tr-3  tr-1   tr-4
#> tr-2 0.000 5.708 7.802 19.627
#> tr-3 0.386 0.000 2.093 13.918
#> tr-1 0.150 0.932 0.000 11.825
#> tr-4 0.000 0.004 0.015  0.000
```

3 Unbalanced Data

When observations are missing, `TukeyC()` automatically adjusts the means using the Least-Squares Means methodology (via the **emmeans** package). The analysis proceeds identically to the balanced case.

```
## Remove the first observation to create an unbalanced dataset
u_tk1 <- with(CRD1,
              TukeyC(y ~ x,
                 data = dfm[-1, ],
                 which = 'x'))
summary(u_tk1)
```

```
#> Groups of means at sig.level = 0.05
#>      Means G1 G2
#> tr-2 59.82  a
#> tr-3 54.11  a
#> tr-1 50.66  a
#> tr-4 40.19     b
#>
#> Matrix of the difference of means above diagonal and
#> respective p-values of the Tukey test below diagonal values
#>       tr-2  tr-3  tr-1   tr-4
#> tr-2 0.000 5.708 9.160 19.627
#> tr-3 0.376 0.000 3.452 13.918
#> tr-1 0.088 0.778 0.000 10.466
#> tr-4 0.000 0.004 0.043  0.000
```

The number of replicates shown at the bottom of the plot reflects the actual (unequal) sample sizes:

```
plot(u_tk1, dispersion = 'sd', d.col = 'tomato')
```

4 Randomized Complete Block Design (RCBD)

`RCBD` contains simulated data for a design with **5 treatment levels** and **4 blocks**. The blocking factor `blk` is included in the formula; `which` selects the factor of interest for the comparison.

```
data(RCBD)

tk4 <- with(RCBD,
```

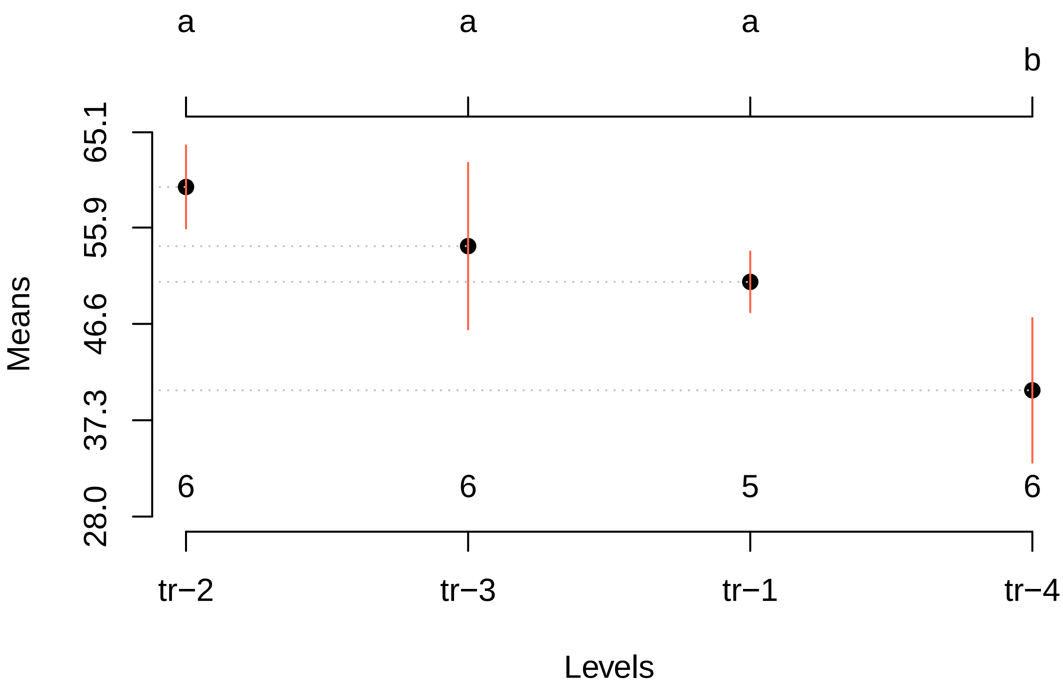

Figure 2: CRD1 (unbalanced): adjusted means with SD bars.

```
            TukeyC(y ~ blk + tra,
               data = dfm,
               which = 'tra'))
summary(tk4)
```

```
#> Groups of means at sig.level = 0.05
#>    Means G1 G2
#> E 155.37  a
#> A 142.93  a  b
#> D 140.39     b
#> B 138.57     b
#> C 138.56     b
#>
#> Matrix of the difference of means above diagonal and
#> respective p-values of the Tukey test below diagonal values
#>       E      A      D      B      C
#> E 0.000 12.438 14.975 16.795 16.805
#> A 0.101  0.000  2.537  4.357  4.367
#> D 0.039  0.978  0.000  1.820  1.830
#> B 0.020  0.864  0.994  0.000  0.010
#> C 0.020  0.863  0.993  1.000  0.000
```

```
plot(tk4,
     dispersion = 'ci',
     d.col = 'darkgreen',
     d.lty = 2)
```

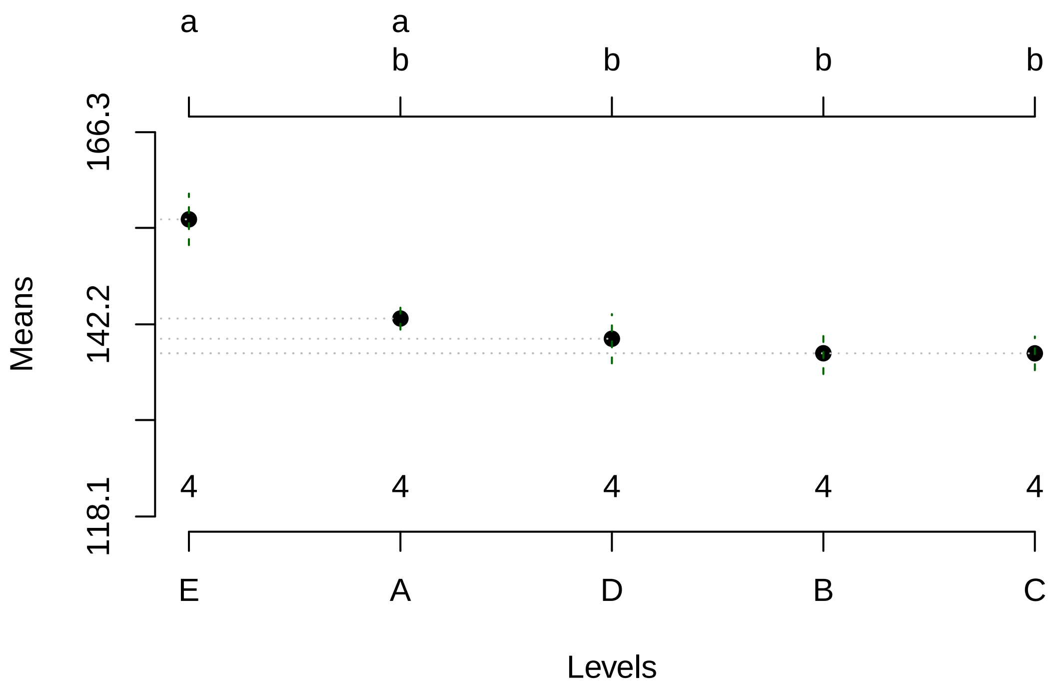

Figure 3: RCBD: treatment means with individual CI bars.

5 Significance Level

The default significance level is `sig.level = 0.05`. Stricter or looser levels lead to fewer or more groups, respectively.

```
## alpha = 0.01 (stricter)
tk_01 <- with(RCBD,
              TukeyC(y ~ blk + tra,
                 data = dfm,
                 which = 'tra',
                 sig.level = 0.01))

## alpha = 0.10 (looser)
tk_10 <- with(RCBD,
              TukeyC(y ~ blk + tra,
                 data = dfm,
                 which = 'tra',
                 sig.level = 0.10))

cat('--- sig.level = 0.01 ---\n')

#> --- sig.level = 0.01 ---

summary(tk_01)

#> Groups of means at sig.level = 0.01
#>    Means G1
#> E 155.37  a
#> A 142.93  a
```

```
#> D 140.39  a
#> B 138.57  a
#> C 138.56  a
#>
#> Matrix of the difference of means above diagonal and
#> respective p-values of the Tukey test below diagonal values
#>       E      A      D      B      C
#> E 0.000 12.438 14.975 16.795 16.805
#> A 0.101  0.000  2.537  4.357  4.367
#> D 0.039  0.978  0.000  1.820  1.830
#> B 0.020  0.864  0.994  0.000  0.010
#> C 0.020  0.863  0.993  1.000  0.000


cat('--- sig.level = 0.10 ---\n')
```

```
#> --- sig.level = 0.10 ---
```

```
summary(tk_10)
```

```
#> Groups of means at sig.level = 0.1
#>    Means G1 G2
#> E 155.37  a
#> A 142.93  a  b
#> D 140.39     b
#> B 138.57     b
#> C 138.56     b
#>
#> Matrix of the difference of means above diagonal and
#> respective p-values of the Tukey test below diagonal values
#>       E      A      D      B      C
#> E 0.000 12.438 14.975 16.795 16.805
#> A 0.101  0.000  2.537  4.357  4.367
#> D 0.039  0.978  0.000  1.820  1.830
#> B 0.020  0.864  0.994  0.000  0.010
#> C 0.020  0.863  0.993  1.000  0.000
```

6 Factorial Experiment (FE)

`FE` contains simulated data for a **3-factor factorial** design (N, P, K), each at 2 levels, in 4 blocks. `TukeyC()` supports both main-effect and nested comparisons using colon notation in `which` and the `fl1` / `fl2` arguments to select the level of the nesting factor.

```
data(FE)

## Main effect: factor N
tk5 <- with(FE,
            TukeyC(y ~ blk + N*P*K,
               data = dfm,
               which = 'N'))
summary(tk5)
```

```
#> Groups of means at sig.level = 0.05
#>    Means G1 G2
```

```
#> n1  2.75  a
#> n0  2.31     b
#>
#> Matrix of the difference of means above diagonal and
#> respective p-values of the Tukey test below diagonal values
#>       n1    n0
#> n1 0.000 0.443
#> n0 0.006 0.000
```

```r
## Nested: levels of N within level 1 of P
tk6 <- with(FE,
            TukeyC(y ~ blk + N*P*K,
               data = dfm,
               which = 'P:N',
               fl1 = 1))
summary(tk6)
```

```
#> Groups of means at sig.level = 0.05
#>       Means G1
#> p0/n1  2.60  a
#> p0/n0  2.41  a
#>
#> Matrix of the difference of means above diagonal and
#> respective p-values of the Tukey test below diagonal values
#>       p0/n1 p0/n0
#> p0/n1 0.000 0.193
#> p0/n0 0.356 0.000
```

```r
## Nested: levels of N within level 2 of P
tk7 <- with(FE,
            TukeyC(y ~ blk + N*P*K,
               data = dfm,
               which = 'P:N',
               fl1 = 2))
summary(tk7)
```

```
#> Groups of means at sig.level = 0.05
#>       Means G1 G2
#> p1/n1  2.90  a
#> p1/n0  2.20     b
#>
#> Matrix of the difference of means above diagonal and
#> respective p-values of the Tukey test below diagonal values
#>       p1/n1 p1/n0
#> p1/n1 0.000 0.694
#> p1/n0 0.003 0.000
```

7 Split-Plot Experiment (SPE)

`SPE` contains simulated data for a design with **3 whole plots** (factor P) and **4 sub-plot treatments** (factor SP). When testing the whole-plot factor, the appropriate error term must be specified via the `error` argument.

```
data(SPE)

## Sub-plot factor SP (residual error, default)
tk8 <- with(SPE,
            TukeyC(y ~ blk + P*SP + Error(blk/P),
               data = dfm,
               which = 'SP'))
summary(tk8)
```

```
#> Groups of means at sig.level = 0.05
#>     Means G1 G2
#> sp1 17.72  a
#> sp4 16.78  a
#> sp3 16.24  a
#> sp2 13.34     b
#>
#> Matrix of the difference of means above diagonal and
#> respective p-values of the Tukey test below diagonal values
#>       sp1   sp4   sp3   sp2
#> sp1 0.000 0.938 1.476 4.383
#> sp4 0.443 0.000 0.538 3.446
#> sp3 0.098 0.824 0.000 2.908
#> sp2 0.000 0.000 0.000 0.000
```

```
## Whole-plot factor P (must specify the blk:P error term)
tk9 <- with(SPE,
            TukeyC(y ~ blk + P*SP + Error(blk/P),
               data = dfm,
               which = 'P',
               error = 'blk:P'))
summary(tk9)
```

```
#> Groups of means at sig.level = 0.05
#>    Means G1
#> p1 16.70  a
#> p2 15.80  a
#> p3 15.56  a
#>
#> Matrix of the difference of means above diagonal and
#> respective p-values of the Tukey test below diagonal values
#>       p1    p2    p3
#> p1 0.000 0.893 1.140
#> p2 0.738 0.000 0.248
#> p3 0.614 0.976 0.000
```

8 Visualisation Options

8.1 Dispersion bars

Four dispersion options are available for `plot.TukeyC()`, as summarised in Table 1.

Table 1: Dispersion options available for `plot.TukeyC()`.

Option	Description
`'mm'`	Min-max range (default)
`'sd'`	±1 standard deviation
`'ci'`	Individual 95 % confidence interval
`'cip'`	Pooled 95 % confidence interval (uses MSE)

`CRD2` provides a more visually rich example with **45 treatment levels**:

```r
data(CRD2)

tk10 <- with(CRD2,
             TukeyC(y ~ x,
                data = dfm,
                which = 'x'))

col=c(rep(2, 6),
      rep(3, 36),
      rep(4, 1),
      rep(5, 2))

plot(tk10,
     dispersion='cip',
     yl=FALSE,
     id.las=2,
     col=col,
     d.col=col)
```

8.2 Comparing all four options

```r
op <- par(mfrow = c(2, 2), mar = c(4, 3, 4, 1))

plot(tk1, dispersion = 'mm',  d.col = 'steelblue')
mtext('(A)', side = 3, adj = 0, line = 2, font = 2)

plot(tk1, dispersion = 'sd',  d.col = 'tomato')
mtext('(B)', side = 3, adj = 0, line = 2, font = 2)

plot(tk1, dispersion = 'ci',  d.col = 'darkgreen')
mtext('(C)', side = 3, adj = 0, line = 2, font = 2)

plot(tk1, dispersion = 'cip', d.col = 'purple')
mtext('(D)', side = 3, adj = 0, line = 2, font = 2)
```

```r
par(op)
```

8.3 Boxplot

`boxplot.TukeyC()` extends the standard boxplot by overlaying the TukeyC group letters above the frame and drawing the treatment mean inside each box.

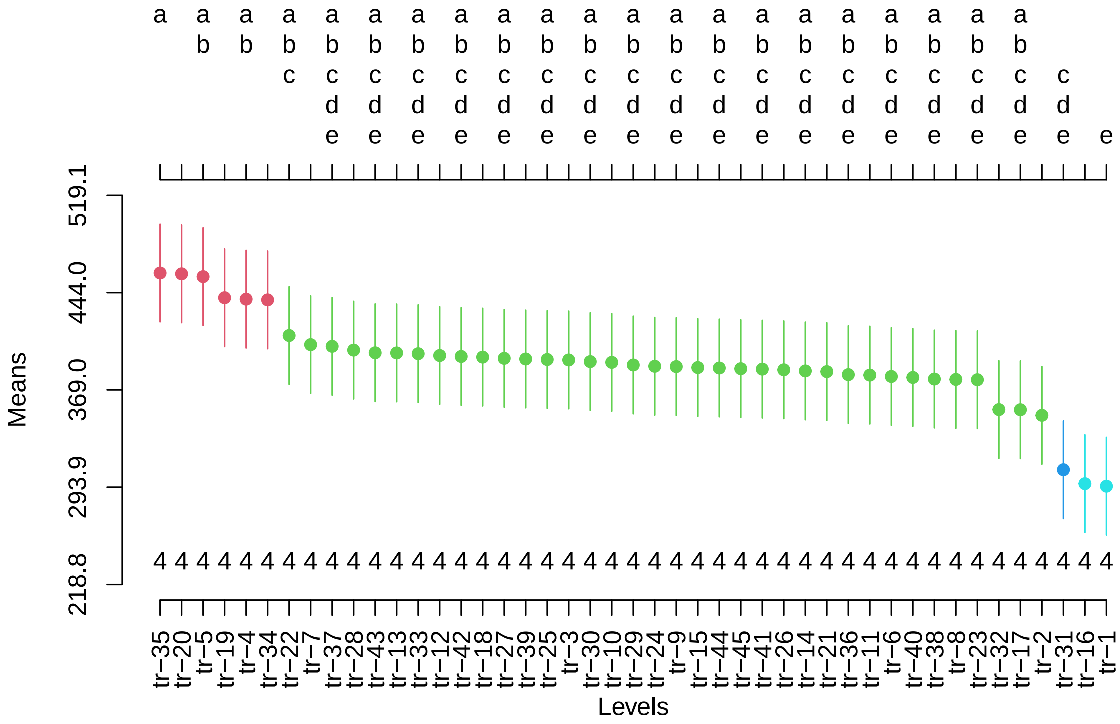

Figure 4: CRD2: 45 treatment means with pooled CI bars.

```r
## boxplot.TukeyC re-evaluates the data argument from the original call;
## pass CRD1$dfm directly so it is findable in any environment.
tk1_bp <- TukeyC(y ~ x,
                 data = CRD1$dfm,
                 which = 'x')

boxplot(tk1_bp,
        mean.col = 'red',
        mean.lwd = 2,
        args.legend = list(x = 'topright'))
```

9 Tabular Output

`xtable()` converts a `TukeyC` result to an `xtable` object for inclusion in LaTeX or HTML documents. Table 2 shows the Tukey HSD grouping for the RCBD example.

```r
library(xtable)

tb <- xtable(tk4,
             caption = 'RCBD: Tukey HSD grouping of treatment means.',
             label = 'tab:rcbd',
             digits = 3)
print(tb,
      type = 'latex',
      caption.placement = 'top',
      include.rownames = FALSE,
      booktabs = TRUE,
```

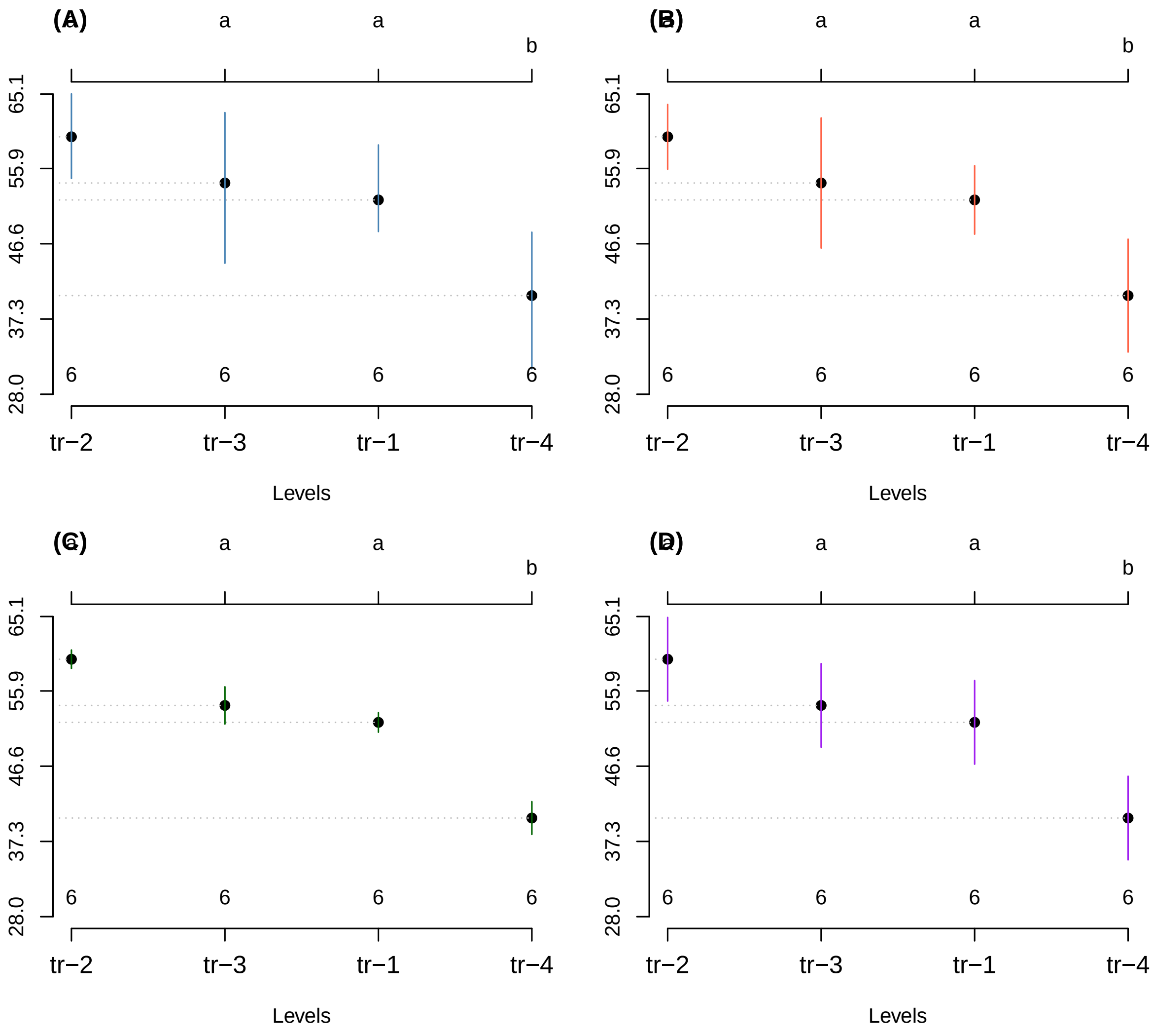

Figure 5: The four dispersion options applied to CRD1. (A) mm: min-max range; (B) sd: standard deviation; (C) ci: individual confidence interval; (D) cip: pooled confidence interval.

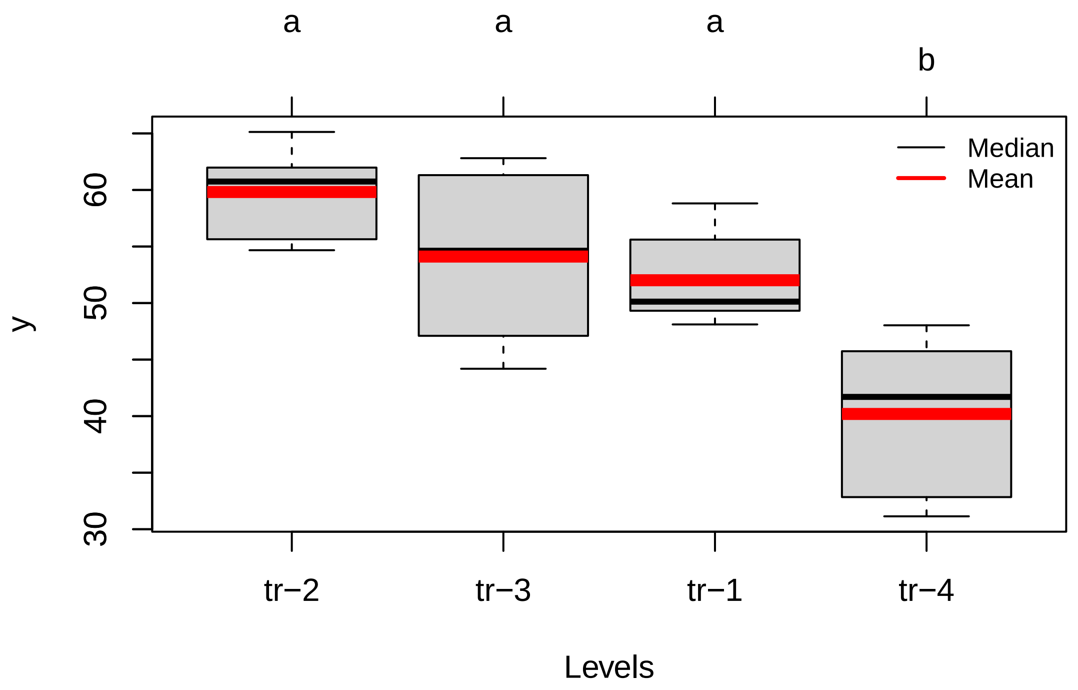

Figure 6: CRD1: boxplot with TukeyC group labels and means (red line).

```
      table.placement = 'H')
```

Table 2: RCBD: Tukey HSD grouping of treatment means.

Treatment	Means	G1	G2	Minimum Significant Difference	Sig.level
E	155.37	a		14.340	0.050
A	142.93	a	b		
D	140.39		b		
B	138.57		b		
C	138.56		b		

10 Mixed Models with lme4

`TukeyC()` also accepts `lmerMod` objects from the **lme4** package, useful when random effects need to be modelled explicitly.

```
library(lme4)

data(RCBD)

lmer1 <- with(RCBD,
              lmer(y ~ (1|blk) + tra,
                   data = dfm))

#> boundary (singular) fit:  see help(’isSingular’)
```

```
tk11 <- TukeyC(lmer1, which = 'tra')
summary(tk11)
```

```
#> Groups of means at sig.level = 0.05
#>    Means G1 G2
#> E 155.37  a
#> A 142.93  a  b
#> D 140.40     b
#> B 138.57     b
#> C 138.56     b
#>
#> Matrix of the difference of means above diagonal and
#> respective p-values of the Tukey test below diagonal values
#>       E      A      D      B      C
#> E 0.000 12.437 14.975 16.795 16.805
#> A 0.082  0.000  2.538  4.358  4.368
#> D 0.029  0.975  0.000  1.820  1.830
#> B 0.014  0.849  0.993  0.000  0.010
#> C 0.014  0.848  0.993  1.000  0.000
```

References

References

[1] Tukey, J. W. (1949). Comparing individual means in the analysis of variance. *Biometrics*, **5**(2), 99–114. doi:10.2307/3001913

[2] Miller, R. G. (1981). *Simultaneous Statistical Inference*. Springer.