Format phân tích ANOVA 2 yếu tố kiểu CRD

Duc Nguyen | tuhocr.com

3/27/23

Table of contents

Nội dung

Format phân tích ANOVA 2 yếu tố kiểu CRD trong R.

Dear bạn, mình là Duc Nguyen, chuyên đào tạo kỹ năng xử lý dữ liệu sử dụng R cho nhà nghiên cứu. Nếu bạn chưa nắm vững kỹ năng R thì bạn hãy tham gia khóa học ở www.tuhocr.com để trang bị kỹ năng này nhé.

Khi tham gia khóa học thì bạn được mình trực tiếp support qua video bài giảng thu sẵn cũng như kèm 1:1 qua teamviewer. Bên cạnh đó, bạn được cấp tài khoản truy cập database các file source code để có template xử lý các nhu cầu phân tích thường ngày, tiết kiệm thời gian tự xây dựng format code R.

Khi bạn có bất kỳ yêu cầu nào cần trợ giúp như là format vẽ đồ thị hay xử lý thống kê thì bạn đừng ngần ngại inbox mình nhé. Facebook mình là: https://www.facebook.com/tuhocr/ rất vui khi được trao đổi cùng bạn.

***R* is the solution for every researcher.**

Bước 1: Import dữ liệu

```
library(readxl)
data_anova <- read_excel("test.xlsx",
                         sheet = "anova-twoway", range = "D5:F53")
data_anova <- as.data.frame(data_anova)
data_anova$diet <- as.factor(data_anova$diet)
data_anova$time <- as.factor(data_anova$time)
data_anova -> my_data

my_data
```

```
   diet   time stability
1   NT1 10-min  89.17712
2   NT1 10-min  89.76035
3   NT1 10-min  89.56072
4   NT1  5-min  86.81088
5   NT1  5-min  87.28091
6   NT1  5-min  84.50821
7   NT2 10-min  88.51183
8   NT2 10-min  89.83705
9   NT2 10-min  89.02299
10  NT2  5-min  85.94260
11  NT2  5-min  86.53363
12  NT2  5-min  84.58238
13  NT3 10-min  90.92160
14  NT3 10-min  90.31614
15  NT3 10-min  86.30768
16  NT3  5-min  88.22052
17  NT3  5-min  87.87284
18  NT3  5-min  86.48140
19  NT4 10-min  91.13180
20  NT4 10-min  91.06331
21  NT4 10-min  90.97900
22  NT4  5-min  87.41313
```

```
23  NT4  5-min  87.79204
24  NT4  5-min  87.83304
25  NT5 10-min  90.33636
26  NT5 10-min  90.45067
27  NT5 10-min  90.97430
28  NT5  5-min  86.72079
29  NT5  5-min  87.59405
30  NT5  5-min  87.51170
31  NT6 10-min  90.73797
32  NT6 10-min  90.21814
33  NT6 10-min  90.40041
34  NT6  5-min  86.89038
35  NT6  5-min  86.45951
36  NT6  5-min  87.25631
37  NT7 10-min  90.87250
38  NT7 10-min  90.58467
39  NT7 10-min  90.56533
40  NT7  5-min  85.88389
41  NT7  5-min  86.83931
42  NT7  5-min  87.13404
43  NT8 10-min  90.83993
44  NT8 10-min  91.04827
45  NT8 10-min  90.10980
46  NT8  5-min  86.33665
47  NT8  5-min  86.20297
48  NT8  5-min  84.82913
```

```r
# Bố trí thí nghiệm
table(data_anova$diet, data_anova$time)
```

```
    10-min 5-min
NT1      3     3
NT2      3     3
NT3      3     3
NT4      3     3
NT5      3     3
NT6      3     3
NT7      3     3
NT8      3     3
```

Bước 2: Đánh giá mức độ phân bố chuẩn

Check the normality assumpttion

Cách 1

```
library(car)
leveneTest(stability ~ diet*time, data = my_data)
```

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group 15  0.7751 0.6936
      32
```

From the output above we can see that the p-value is (0.6936) not less than the significance level of 0.05. This means that there is no evidence to suggest that the variance across groups is statistically significantly different. Therefore, we can assume the homogeneity of variances in the different treatment groups.

Tạm dịch: Kết quả cho thấy p-value là 0.6936 lớn hơn 0.05 (giả thuyết cho là có sự phân bố không chuẩn - heterogeneity). Do đó bộ dataset này có sự phân bố chuẩn (homogeneity) trong sự khác biệt giữa các nghiệm thức.

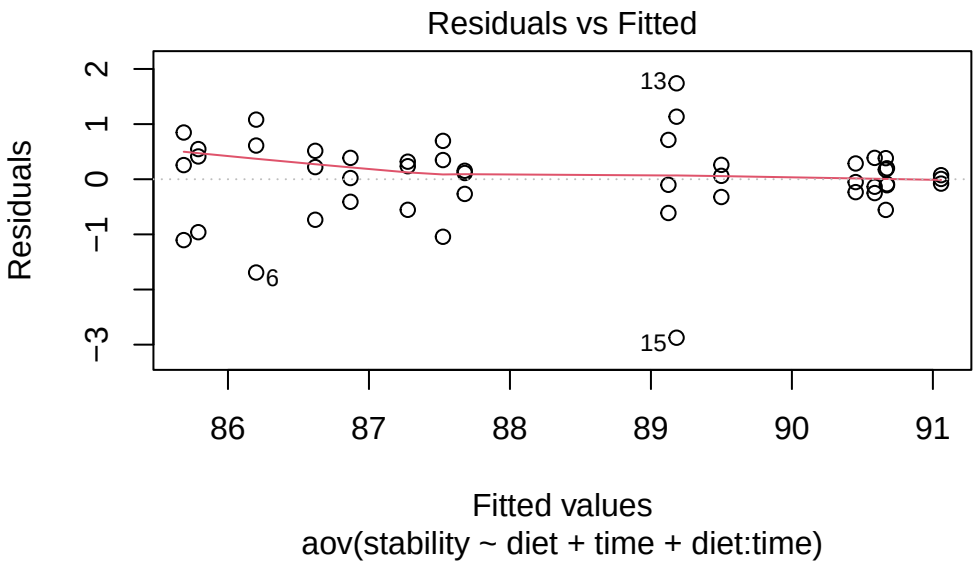
Cách 2

Cần tính anova trước để có data vẽ đồ thị qqplot

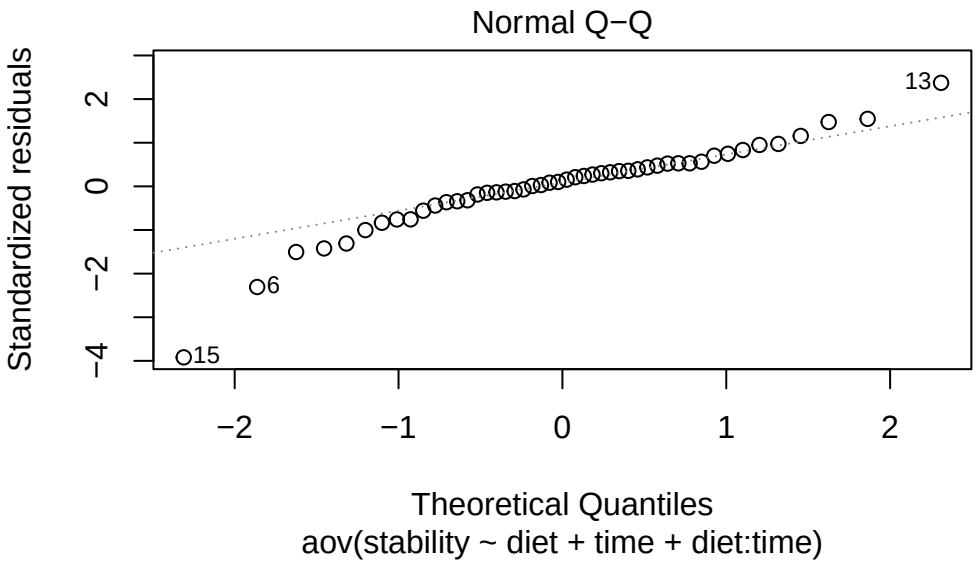
```
# Compute two-way ANOVA test
res.aov2 <- aov(stability ~ diet + time, data = my_data)
# summary(res.aov2)
# anova(res.aov2)

# Compute two-way ANOVA test with interaction effect
res.aov3 <- aov(stability ~ diet + time + diet:time, data = my_data)
# anova(res.aov3)
```

Vẽ đồ thị

```r
plot(res.aov3, 1) ## Homogeneity of variances
```




```r
plot(res.aov3, 2) ## Check the normality assumptiion
```



Phát hiện các data point 6, 13, 15 là outlier, *có thể* loại ra để làm dataset phân bố chuẩn hơn.

Cách 3

```
# Extract the residuals
aov_residuals <- residuals(object = res.aov3)
# Run Shapiro-Wilk test
shapiro.test(x = aov_residuals)
```

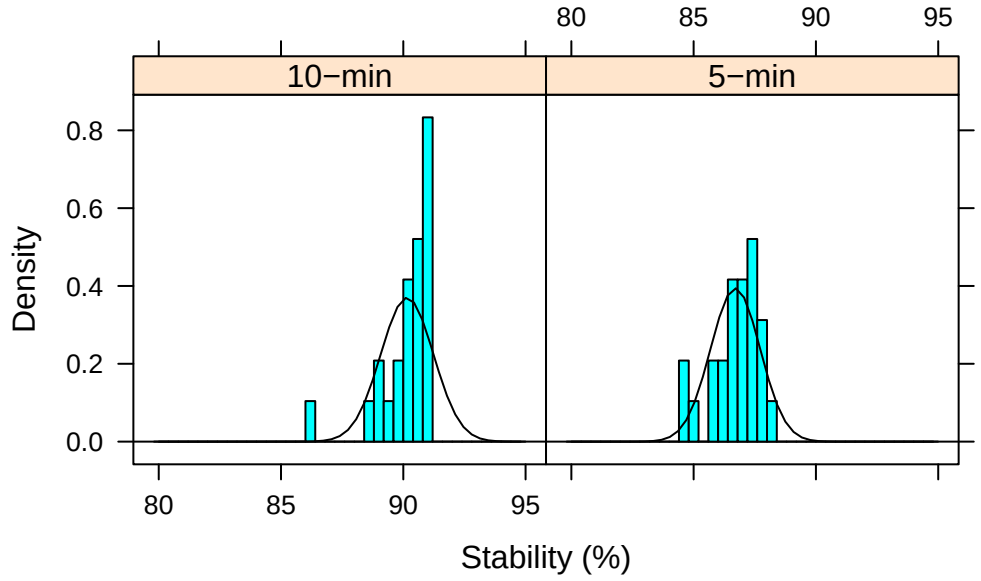
```
    Shapiro-Wilk normality test

data:  aov_residuals
W = 0.91161, p-value = 0.001523
```

p-value từ test Shapiro-Wilk normality cho thấy nhỏ hơn 0.05 (giả thuyết là phân bố chuẩn), do đó về mặt ý nghĩa thống kê thì bộ dataset này có phân bố chuẩn.

Bước 3: Khảo sát đặc điểm dữ liệu

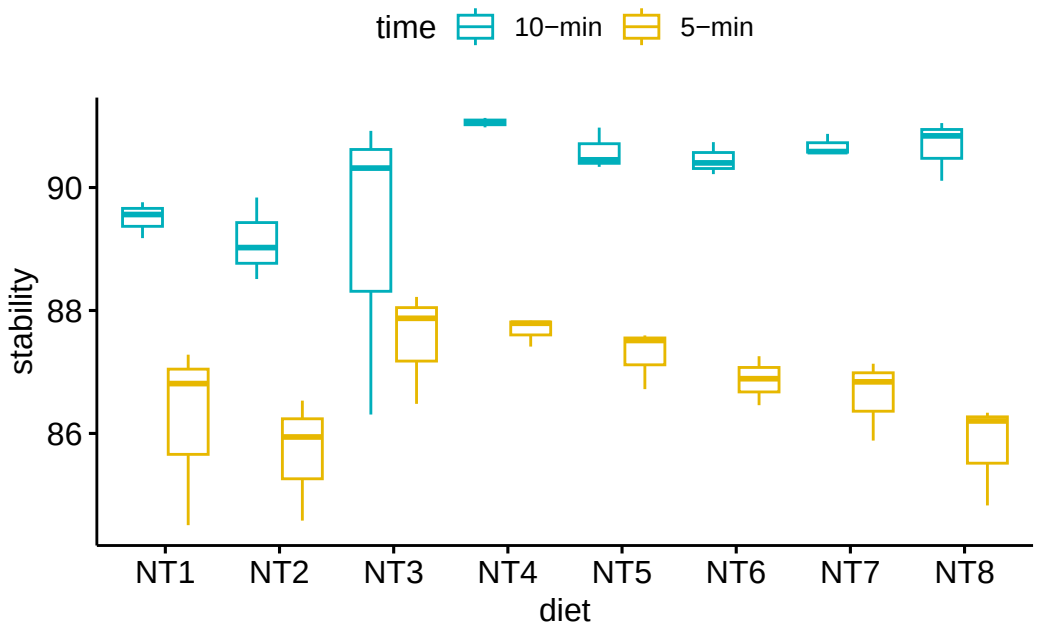
Histogram theo `time`

```
library(lattice)
histogram( ~ stability | time, data = my_data,
           xlab = "Stability (%)", type = "density",
           breaks = seq(from = 80, to = 95, by = 0.4),
           panel = function(x, ...) {
             panel.histogram(x, ...)
             panel.mathdensity(dmath = dnorm, col = "black",
                               args = list(mean=mean(x),sd=sd(x)))
           } )
```



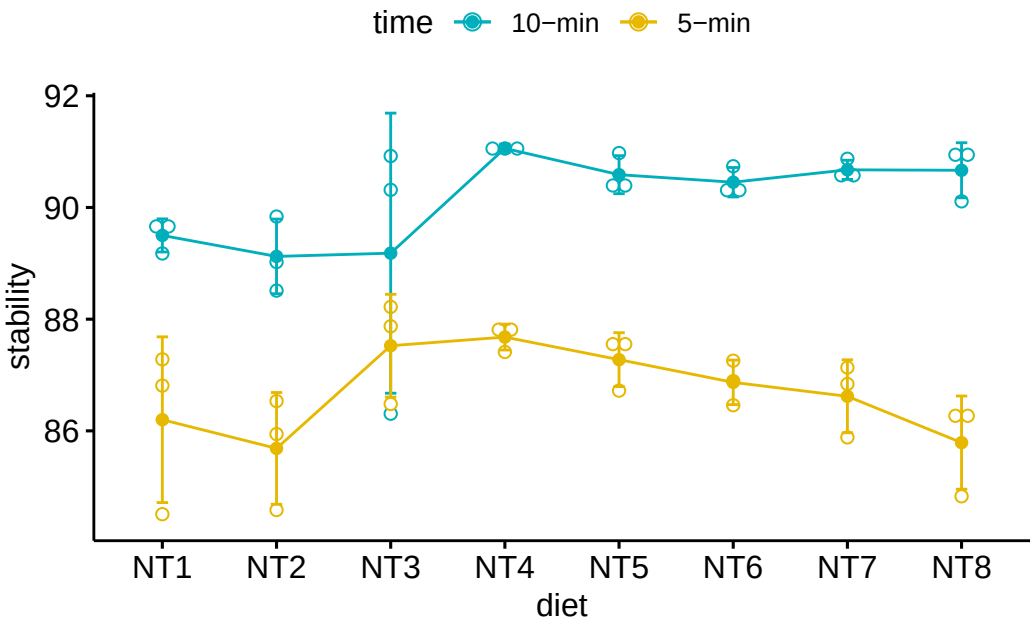
Box plot with multiple groups

```
library(ggpubr)
ggboxplot(my_data, x = "diet", y = "stability", color = "time",
          palette = c("#00AFBB", "#E7B800"))
```



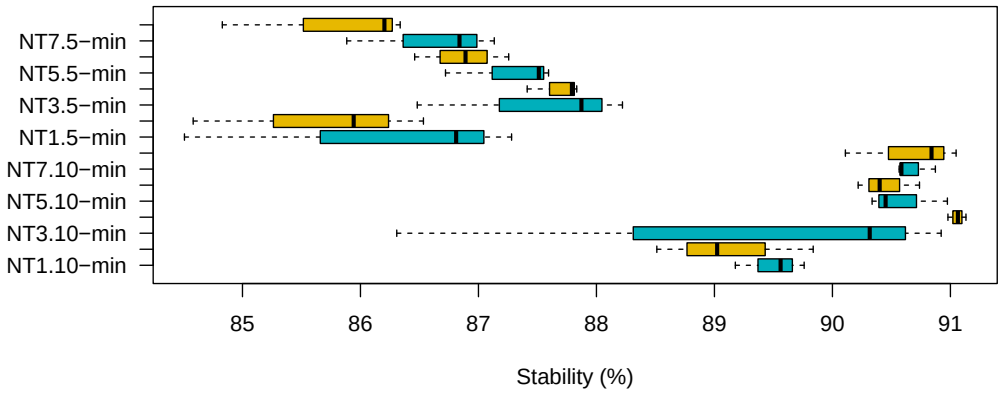
Line plots with multiple groups

```
# Add error bars: mean_se
# (other values include: mean_sd, mean_ci, median_iqr, ....)
library("ggpubr")
ggline(my_data, x = "diet", y = "stability", color = "time",
       add = c("mean_sd", "dotplot"),
       palette = c("#00AFBB", "#E7B800"))
```



Box plot with two factor variables

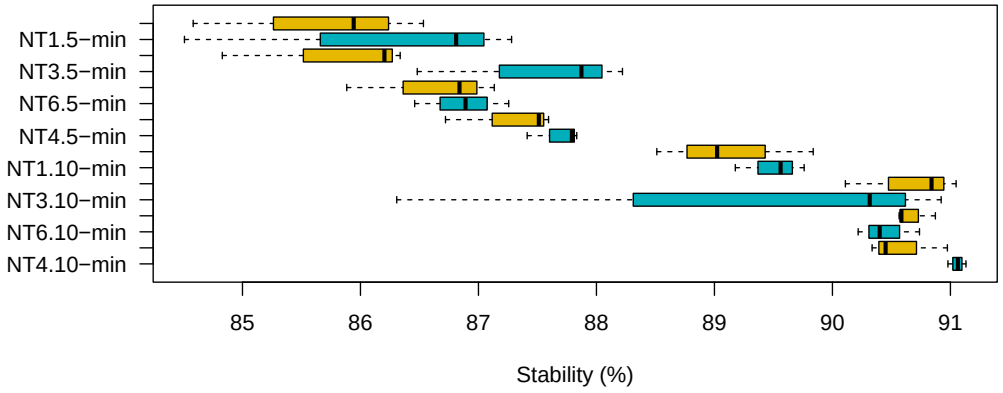
```
oldpar <- par(no.readonly = TRUE)
par(mar = c(6, 7, 1, 6))
boxplot(stability ~ diet * time, data = my_data, frame = TRUE,
        col = c("#00AFBB", "#E7B800"), horizontal = TRUE, las = 1,
        axisnames = TRUE, ylab = "", xlab = "Stability (%)")
```



```r
par(oldpar)
```

Nếu muốn vẽ `boxplot` theo thứ tự các cột thì cần `reorder` cột factor `diet` theo `stability`

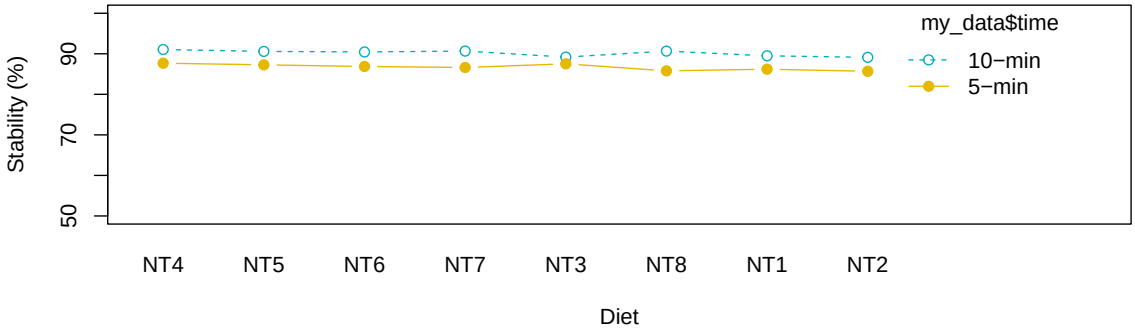
```r
my_data$diet <- reorder(my_data$diet, my_data$stability, decreasing = TRUE)
my_data$time <- reorder(my_data$time, my_data$stability, decreasing = TRUE)

oldpar <- par(no.readonly = TRUE)
par(mar = c(6, 7, 1, 6))
boxplot(stability ~ diet * time, data = my_data, frame = TRUE,
        col = c("#00AFBB", "#E7B800"), horizontal = TRUE, las = 1,
        axisnames = TRUE, ylab = "", xlab = "Stability (%)")
```




```r
par(oldpar)
```

Two-way interaction plot

```r
interaction.plot(x.factor = my_data$diet, trace.factor = my_data$time,
                 response = my_data$stability, fun = mean,
                 type = "b", legend = TRUE,
                 xlab = "Diet", ylab ="Stability (%)",
                 pch = c(1, 19), col = c("#00AFBB", "#E7B800"),
                 ylim = c(50, 100))
```


Stability (%)
50
70
90
my_data$time
10−min
5−min
NT4
NT5
NT6
NT7
NT3
NT8
NT1
NT2
Diet

Bước 4: Phân tích ANOVA 2 yếu tố CRD

Tính p-value

```r
# Compute two-way ANOVA test
res.aov2 <- aov(stability ~ diet + time, data = my_data)
# summary(res.aov2)
anova(res.aov2)
```

```
Analysis of Variance Table

Response: stability
          Df  Sum Sq Mean Sq  F value    Pr(>F)
diet       7  15.999   2.286   2.5966   0.02672 *
time       1 142.822 142.822 162.2555 1.781e-15 ***
Residuals 39  34.329   0.880
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Compute two-way ANOVA test with interaction effect
res.aov3 <- aov(stability ~ diet + time + diet:time, data = my_data)
anova(res.aov3)
```

```
Analysis of Variance Table

Response: stability
          Df  Sum Sq Mean Sq  F value   Pr(>F)
diet       7  15.999   2.286   2.8340  0.02041 *
time       1 142.822 142.822 177.0920 1.37e-14 ***
diet:time  7   8.521   1.217   1.5094  0.19953
Residuals 32  25.807   0.806
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Phân hạng

```r
library(agricolae)
LSD.test(res.aov2, c("diet", "time"), console = TRUE)
```

```
Study: res.aov2 ~ c("diet", "time")

LSD t Test for stability

Mean Square Error:  0.8802287

diet:time,  means and individual ( 95 %) CI

           stability        std r      LCL      UCL      Min      Max
NT1:10-min  89.49940 0.29641311 3 88.40376 90.59503 89.17712 89.76035
NT1:5-min   86.20000 1.48386449 3 85.10436 87.29564 84.50821 87.28091
NT2:10-min  89.12396 0.66835713 3 88.02832 90.21959 88.51183 89.83705
NT2:5-min   85.68620 1.00057733 3 84.59057 86.78184 84.58238 86.53363
NT3:10-min  89.18181 2.50740722 3 88.08617 90.27744 86.30768 90.92160
NT3:5-min   87.52492 0.92028383 3 86.42929 88.62056 86.48140 88.22052
NT4:10-min  91.05804 0.07653564 3 89.96240 92.15367 90.97900 91.13180
NT4:5-min   87.67940 0.23150732 3 86.58376 88.77504 87.41313 87.83304
NT5:10-min  90.58711 0.34015176 3 89.49147 91.68275 90.33636 90.97430
NT5:5-min   87.27551 0.48216583 3 86.17988 88.37115 86.72079 87.59405
NT6:10-min  90.45218 0.26375280 3 89.35654 91.54781 90.21814 90.73797
NT6:5-min   86.86873 0.39883737 3 85.77310 87.96437 86.45951 87.25631
NT7:10-min  90.67417 0.17203343 3 89.57853 91.76981 90.56533 90.87250
NT7:5-min   86.61908 0.65352540 3 85.52344 87.71472 85.88389 87.13404
NT8:10-min  90.66600 0.49282165 3 89.57036 91.76164 90.10980 91.04827
NT8:5-min   85.78958 0.83445867 3 84.69395 86.88522 84.82913 86.33665

Alpha: 0.05 ; DF Error: 39
Critical Value of t: 2.022691

least Significant Difference: 1.549465

Treatments with the same letter are not significantly different.

           stability groups
NT4:10-min  91.05804      a
NT7:10-min  90.67417     ab
NT8:10-min  90.66600    abc
NT5:10-min  90.58711    abc
NT6:10-min  90.45218    abc
```

```
NT1:10-min  89.49940     bc
NT3:10-min  89.18181    bcd
NT2:10-min  89.12396     cd
NT4:5-min   87.67940     de
NT3:5-min   87.52492      e
NT5:5-min   87.27551     ef
NT6:5-min   86.86873    efg
NT7:5-min   86.61908    efg
NT1:5-min   86.20000    efg
NT8:5-min   85.78958     fg
NT2:5-min   85.68620      g
```

```
duncan.test(res.aov2, c("diet", "time"), console = TRUE)
```

```
Study: res.aov2 ~ c("diet", "time")

Duncan's new multiple range test
for stability

Mean Square Error:  0.8802287

diet:time,  means

           stability        std r      Min      Max
NT1:10-min  89.49940 0.29641311 3 89.17712 89.76035
NT1:5-min   86.20000 1.48386449 3 84.50821 87.28091
NT2:10-min  89.12396 0.66835713 3 88.51183 89.83705
NT2:5-min   85.68620 1.00057733 3 84.58238 86.53363
NT3:10-min  89.18181 2.50740722 3 86.30768 90.92160
NT3:5-min   87.52492 0.92028383 3 86.48140 88.22052
NT4:10-min  91.05804 0.07653564 3 90.97900 91.13180
NT4:5-min   87.67940 0.23150732 3 87.41313 87.83304
NT5:10-min  90.58711 0.34015176 3 90.33636 90.97430
NT5:5-min   87.27551 0.48216583 3 86.72079 87.59405
NT6:10-min  90.45218 0.26375280 3 90.21814 90.73797
NT6:5-min   86.86873 0.39883737 3 86.45951 87.25631
NT7:10-min  90.67417 0.17203343 3 90.56533 90.87250
NT7:5-min   86.61908 0.65352540 3 85.88389 87.13404
NT8:10-min  90.66600 0.49282165 3 90.10980 91.04827
NT8:5-min   85.78958 0.83445867 3 84.82913 86.33665
```

```
Alpha: 0.05 ; DF Error: 39

Critical Range
       2        3        4        5        6        7        8        9
1.549465 1.629128 1.681203 1.718713 1.747311 1.769943 1.788331 1.803563
      10       11       12       13       14       15       16
1.816368 1.827257 1.836601 1.844677 1.851698 1.857828 1.863200

Means with the same letter are not significantly different.

           stability groups
NT4:10-min  91.05804      a
NT7:10-min  90.67417     ab
NT8:10-min  90.66600     ab
NT5:10-min  90.58711     ab
NT6:10-min  90.45218     ab
NT1:10-min  89.49940     ab
NT3:10-min  89.18181     bc
NT2:10-min  89.12396     bc
NT4:5-min   87.67940     cd
NT3:5-min   87.52492    cde
NT5:5-min   87.27551    def
NT6:5-min   86.86873    def
NT7:5-min   86.61908    def
NT1:5-min   86.20000    def
NT8:5-min   85.78958     ef
NT2:5-min   85.68620      f
```

```r
HSD.test(res.aov2, c("diet", "time"), console = TRUE)
```

```
Study: res.aov2 ~ c("diet", "time")

HSD Test for stability

Mean Square Error:  0.8802287

diet:time,  means

           stability        std r      Min      Max
NT1:10-min  89.49940 0.29641311 3 89.17712 89.76035
NT1:5-min   86.20000 1.48386449 3 84.50821 87.28091
NT2:10-min  89.12396 0.66835713 3 88.51183 89.83705
```

```
NT2:5-min   85.68620 1.00057733 3 84.58238 86.53363
NT3:10-min  89.18181 2.50740722 3 86.30768 90.92160
NT3:5-min   87.52492 0.92028383 3 86.48140 88.22052
NT4:10-min  91.05804 0.07653564 3 90.97900 91.13180
NT4:5-min   87.67940 0.23150732 3 87.41313 87.83304
NT5:10-min  90.58711 0.34015176 3 90.33636 90.97430
NT5:5-min   87.27551 0.48216583 3 86.72079 87.59405
NT6:10-min  90.45218 0.26375280 3 90.21814 90.73797
NT6:5-min   86.86873 0.39883737 3 86.45951 87.25631
NT7:10-min  90.67417 0.17203343 3 90.56533 90.87250
NT7:5-min   86.61908 0.65352540 3 85.88389 87.13404
NT8:10-min  90.66600 0.49282165 3 90.10980 91.04827
NT8:5-min   85.78958 0.83445867 3 84.82913 86.33665

Alpha: 0.05 ; DF Error: 39
Critical Value of Studentized Range: 5.171129

Minimun Significant Difference: 2.801061

Treatments with the same letter are not significantly different.

           stability groups
NT4:10-min  91.05804      a
NT7:10-min  90.67417      a
NT8:10-min  90.66600      a
NT5:10-min  90.58711      a
NT6:10-min  90.45218     ab
NT1:10-min  89.49940    abc
NT3:10-min  89.18181   abcd
NT2:10-min  89.12396   abcd
NT4:5-min   87.67940   bcde
NT3:5-min   87.52492    cde
NT5:5-min   87.27551    cde
NT6:5-min   86.86873    cde
NT7:5-min   86.61908     de
NT1:5-min   86.20000      e
NT8:5-min   85.78958      e
NT2:5-min   85.68620      e
```

Tham khảo

1. http://www.sthda.com/english/wiki/two-way-anova-test-in-r

2. https://stackoverflow.com/questions/43123462/how-to-obtain-rmse-out-of-lm-result
3. https://online.stat.psu.edu/stat501/lesson/2/2.6
4. https://rcompanion.org/handbook/G_14.html
5. RMSE (Root Mean Square Error) https://agronomy4future.org/?p=15930
6. https://stats.stackexchange.com/questions/445200/coefficient-of-variation-for-beween-gro

Sơ kết

Trên đây là format phân tích ANOVA 2 yếu tố trong R. Để học R bài bản từ A đến Z, thân mời Bạn tham gia **khóa học "HDSD R để xử lý dữ liệu" để có nền tảng vững chắc về R nhằm tự tay làm các câu chuyện dữ liệu của riêng mình!**

ĐĂNG KÝ NGAY: https://www.tuhocr.com/register

Hướng dẫn cài đặt package tuhocrhttps://tuhocr.github.io/