

#One-Way ANOVA - Rice Example

One-way ANOVA is used to compare means when there are more than two groups. In this example, Tukey adjusted pairwise comparisons are also considered.

In this example, the effects of four acids on the growth of rice seedlings are compared in a completely randomized design. Seedling shoot dry weights are compared after 7 days in solution.

#Load the dplyr and emmeans packages.

```
#Remember packages need to be installed before first use!
library(dplyr)
library(emmeans)
```

#Two approaches to importing the data.

```
Rice <- read.csv("")
Rice <- read.csv(file.choose())
str(Rice)
boxplot(weight ~ trt, data = Rice, main = "Boxplots")
```

#Use dplyr package to calculate summary statistics by trt.

```
#First we use the group_by function.
#Resulting object is a "tibble".
RiceGrpd <- group_by(Rice, trt)
RiceGrpd
#Then pass the grouped data to the summarise function.
#Summary statistics are automatically computed "by group".
SumStats <- summarise(RiceGrpd,
                      n = n(),
                      mean = mean(weight),
                      sd = sd(weight),
                      SE = sd/sqrt(n))
```

```
SumStats
```

```
#In practice, we can combine the two steps.
```

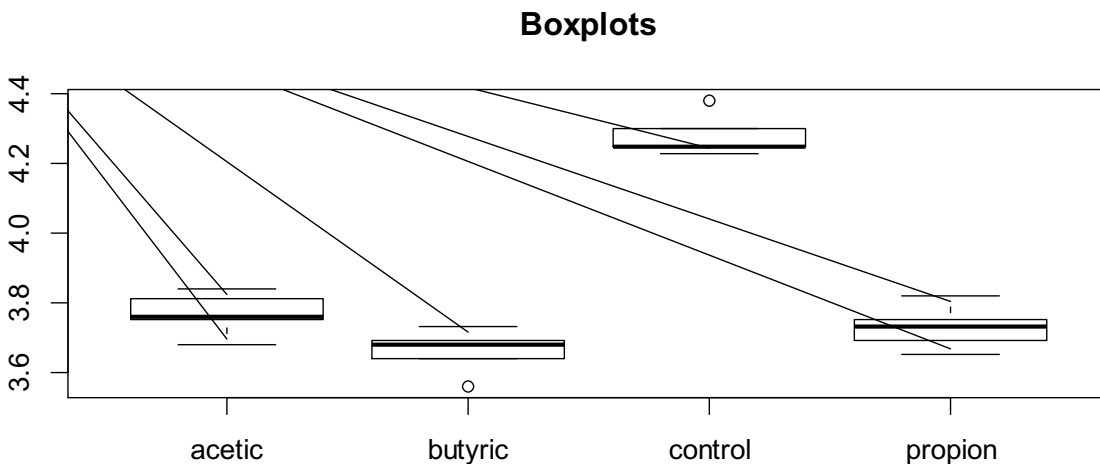
```
SumStats <- summarise(group_by(Rice, trt),
                      n = n(),
                      mean = mean(weight),
                      sd = sd(weight),
                      SE = sd/sqrt(n))
```

#One-way ANOVA

```
#Note: trt should be defined as.factor! See str() output above.
Fit <- lm(weight ~ trt, data = Rice)
#summary() output is not of direct interest here.
summary(Fit)
#anova() output is typically of more interest in ANOVA settings.
anova(Fit)
```

```
#Diagnostic plots
par(mfrow = c(2, 2))
plot(Fit)
#Use emmeans package to get pairwise comparisons with Tukey
adjustment.
emmeans(Fit, pairwise ~ trt)
```

```
> #Load the dplyr and emmeans packages
> #Remember packages need to be installed before first use!
> library(dplyr)
> library(emmeans)
>
> #Two approaches to importing the data
> Rice <- read.csv("/Rice.csv")
> Rice <- read.csv(file.choose())
> str(Rice)
'data.frame':      20 obs. of  2 variables:
 $ trt   : Factor w/ 4 levels "acetic","butyric",...: 3 3 3 3 3 1
1 1 1 1 ...
 $ weight: num  4.23 4.38 4.25 4.3 4.25 3.75 3.68 3.81 3.84 ...
> boxplot(weight ~ trt, data = Rice, main = "Boxplots")
```



```
> #Use dplyr package to calculate summary statistics by trt.
> #First we use the group_by function.
> #Resulting object is a "tibble".
> RiceGrpd <- group_by(Rice, trt)
```

```

> RiceGrpd
# A tibble: 20 x 2
# Groups:   trt [4]
      trt weight
  <fctr> <dbl>
1 control  4.23
2 control  4.38
3 control  4.25
4 control  4.30
5 control  4.25
6  acetic  3.75
7  acetic  3.68
8  acetic  3.81
9  acetic  3.84
10 acetic  3.76
11 propion 3.75
12 propion 3.65
13 propion 3.82
14 propion 3.69
15 propion 3.73
16 butyric 3.68
17 butyric 3.69
18 butyric 3.64
19 butyric 3.56
20 butyric 3.73
> #Then pass the grouped data to the summarise function.
> #Summary statistics are automatically computed "by group".
> SumStats <- summarise(RiceGrpd,
+                        n = n(),
+                        mean = mean(weight),
+                        sd = sd(weight),
+                        SE = sd/sqrt(n))
> SumStats
# A tibble: 4 x 5
      trt      n mean      sd      SE
  <fctr> <int> <dbl>   <dbl>   <dbl>
1  acetic     5 3.768 0.06140033 0.02745906
2 butyric     5 3.660 0.06442049 0.02880972
3 control     5 4.282 0.06058052 0.02709243
4 propion     5 3.728 0.06418723 0.02870540
> #In practice, we can combine the two steps
> SumStats <- summarise(group_by(Rice, trt),
+                        n = n(),
+                        mean = mean(weight),
+                        sd = sd(weight),
+                        SE = sd/sqrt(n))

```

```
> #One-way ANOVA
> #Note: trt should be defined as.factor! See str() output above
> Fit <- lm(weight ~ trt, data = Rice)
> #summary() output is not of direct interest here.
> summary(Fit)
```

```
Call:
lm(formula = weight ~ trt, data = Rice)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-0.1000 -0.0335 -0.0030  0.0330  0.0980
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.76800    0.02803  134.443 < 2e-16 ***
trtbutyric   -0.10800    0.03964   -2.725   0.015 *
trtcontrol    0.51400    0.03964   12.968 6.63e-10 ***
trtpropion   -0.04000    0.03964   -1.009   0.328
---

```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.06267 on 16 degrees of freedom
Multiple R-squared:  0.951,    Adjusted R-squared:  0.9418
F-statistic: 103.5 on 3 and 16 DF,  p-value: 1.083e-10
```

```
> #anova() output is typically of more interest in ANOVA
settings.
```

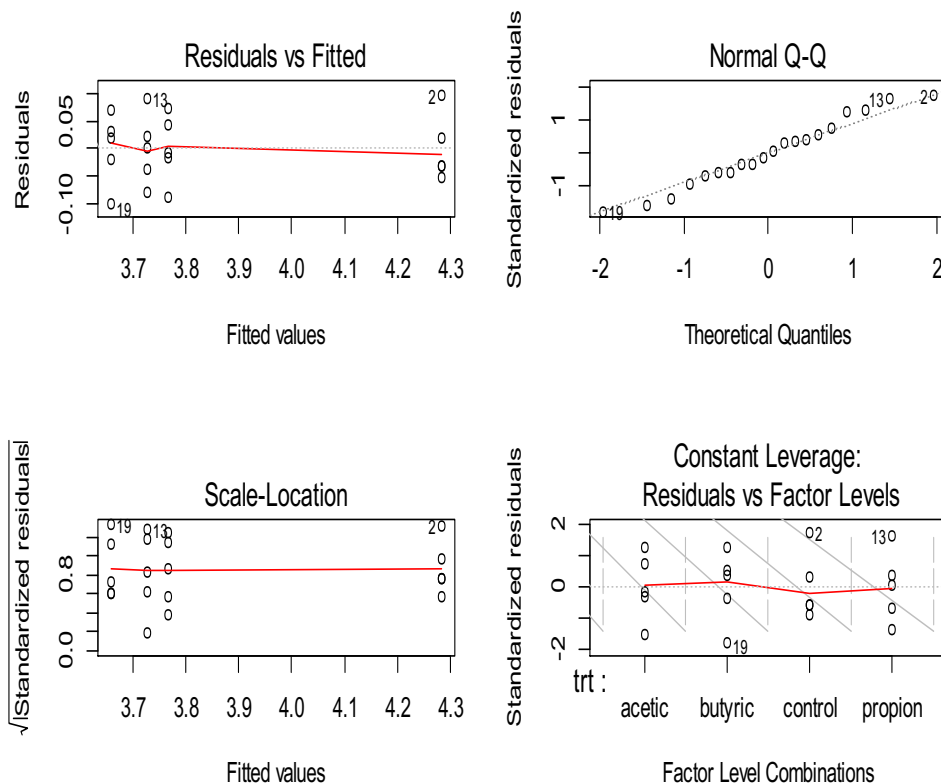
```
> anova(Fit)
Analysis of Variance Table
```

```
Response: weight
      Df Sum Sq Mean Sq F value    Pr(>F)
trt     3  1.21985  0.40662   103.53 1.083e-10 ***
Residuals 16  0.06284  0.00393
---

```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> #Diagnostic plots
> par(mfrow = c(2, 2))
> plot(Fit)
```



```
> #Use emmeans package to get pairwise comparisons with Tukey ad
justment.
```

```
> emmeans(Fit, pairwise ~ trt)
```

```
$emmeans
```

trt	emmean	SE	df	lower.CL	upper.CL
acetic	3.768	0.02802677	16	3.708586	3.827414
butyric	3.660	0.02802677	16	3.600586	3.719414
control	4.282	0.02802677	16	4.222586	4.341414
propion	3.728	0.02802677	16	3.668586	3.787414

Confidence level used: 0.95

```
$contrasts
```

contrast	estimate	SE	df	t.ratio	p.value
acetic - butyric	0.108	0.03963584	16	2.725	0.0648
acetic - control	-0.514	0.03963584	16	-12.968	<.0001
acetic - propion	0.040	0.03963584	16	1.009	0.7465
butyric - control	-0.622	0.03963584	16	-15.693	<.0001
butyric - propion	-0.068	0.03963584	16	-1.716	0.3481
control - propion	0.554	0.03963584	16	13.977	<.0001

P value adjustment: tukey method for comparing a family of 4 estimates