practice13

Qiang Liu

2022-11-22

```
> data("ToothGrowth")
> head(ToothGrowth)
```

```
##     len supp dose
## 1  4.2   VC  0.5
## 2 11.5   VC  0.5
## 3  7.3   VC  0.5
## 4  5.8   VC  0.5
## 5  6.4   VC  0.5
## 6 10.0   VC  0.5
```

```
> data("PlantGrowth")
> head(PlantGrowth)
```

```
##   weight group
## 1   4.17  ctrl
## 2   5.58  ctrl
## 3   5.18  ctrl
## 4   6.11  ctrl
## 5   4.50  ctrl
## 6   4.61  ctrl
```

```
> my_data <- PlantGrowth
> library(dplyr)
```

```
## Warning:   'dplyr' R 4.2.1
```

```
##
##     'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
> group_by(my_data, group) %>%
+   summarise(
+     count = n(),
+     mean = mean(weight, na.rm = TRUE),
+     sd = sd(weight, na.rm = TRUE)
+       )
```

```
## # A tibble: 3 x 4
##    group count  mean    sd
##    <fct> <int> <dbl> <dbl>
## 1 ctrl     10  5.03 0.583
## 2 trt1     10  4.66 0.794
## 3 trt2     10  5.53 0.443
```

```
> # Box plot
> boxplot(weight ~ group, data = my_data,
+         xlab = "Treatment", ylab = "Weight",
+         frame = FALSE, col = c("#00AFBB", "#E7B800", "#FC4E07"))
```
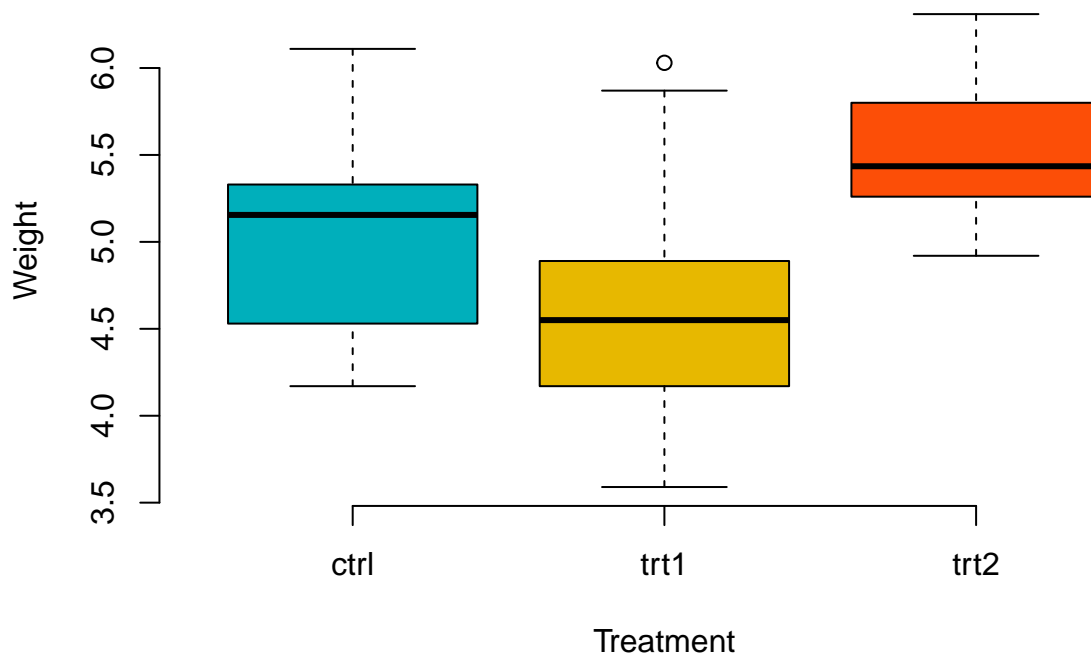


```
> # Compute the analysis of variance
> res.aov <- aov(weight ~ group, data = my_data)
> summary(res.aov)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
```

```
## group         2   3.766   1.8832    4.846 0.0159 *
## Residuals    27  10.492   0.3886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> TukeyHSD(res.aov)


##    Tukey multiple comparisons of means
##      95% family-wise confidence level
##
## Fit: aov(formula = weight ~ group, data = my_data)
##
## $group
##              diff         lwr        upr      p adj
## trt1-ctrl  -0.371  -1.0622161  0.3202161  0.3908711
## trt2-ctrl   0.494  -0.1972161  1.1852161  0.1979960
## trt2-trt1   0.865   0.1737839  1.5562161  0.0120064

> ## plot
> plot(TukeyHSD(res.aov))
> library(multcomp)


## Warning:   'multcomp' R 4.2.2


##        mvtnorm


##        survival


##        TH.data


## Warning:   'TH.data' R 4.2.2


##        MASS


## Warning:   'MASS' R 4.2.1


##
##      'MASS'


## The following object is masked from 'package:dplyr':
##
##      select


##
##      'TH.data'


## The following object is masked from 'package:MASS':
##
##      geyser
```
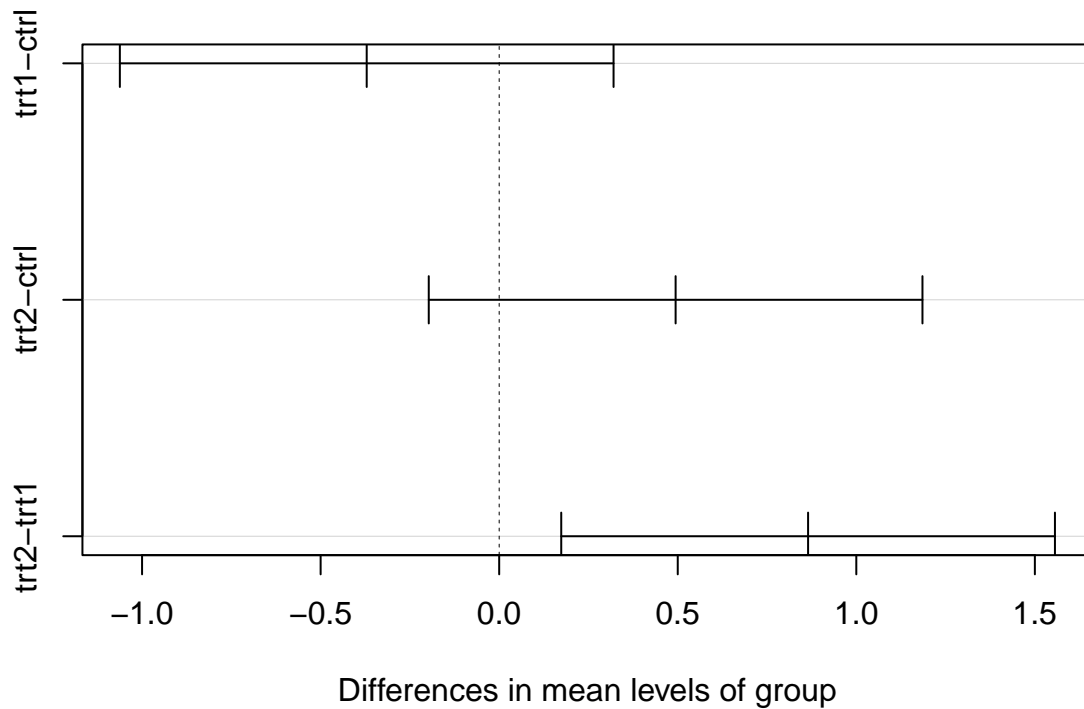
95% family–wise confidence level


Differences in mean levels of group

```
> tuk <- glht(res.aov, linfct = mcp(group = "Tukey"))
> summary(tuk)
```

```
##
##    Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = weight ~ group, data = my_data)
##
## Linear Hypotheses:
##                Estimate Std. Error t value Pr(>|t|)
## trt1 - ctrl == 0  -0.3710     0.2788  -1.331    0.391
## trt2 - ctrl == 0   0.4940     0.2788   1.772    0.198
## trt2 - trt1 == 0   0.8650     0.2788   3.103    0.012 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```
> plot(tuk)
```

95% family−wise confidence level
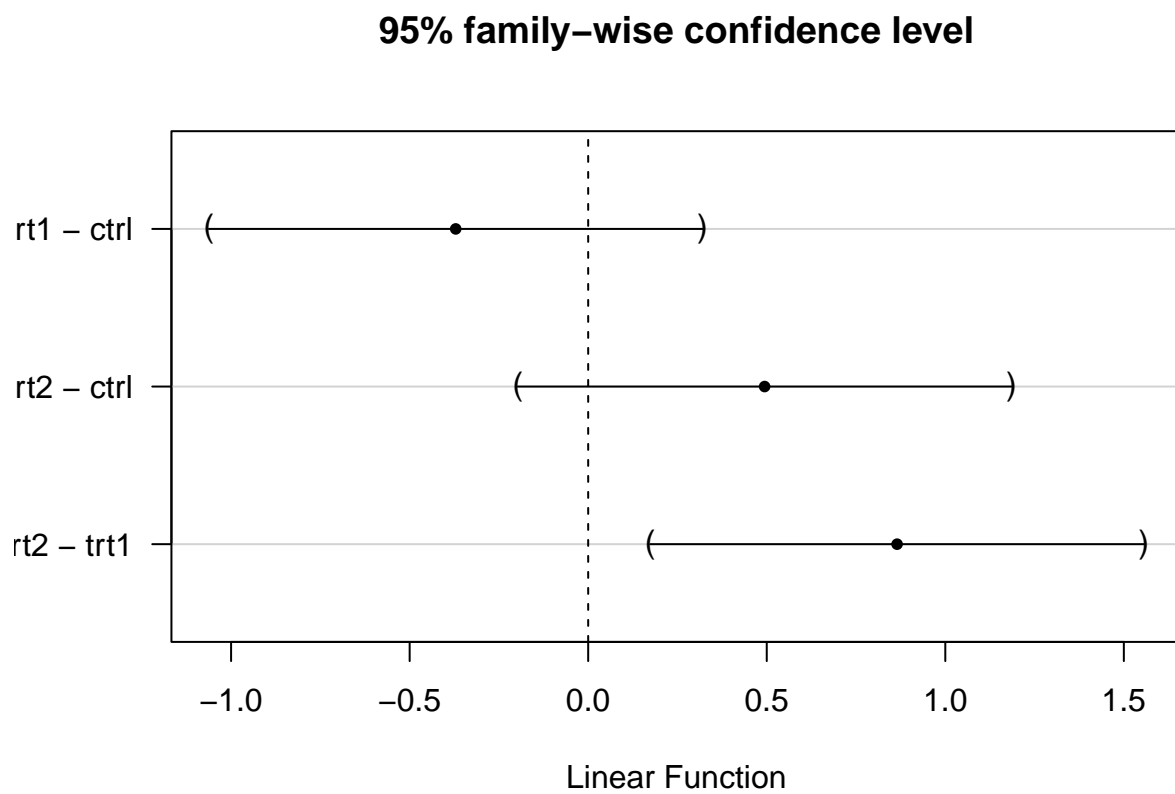

```
> plot(cld(tuk, level=0.05))
```
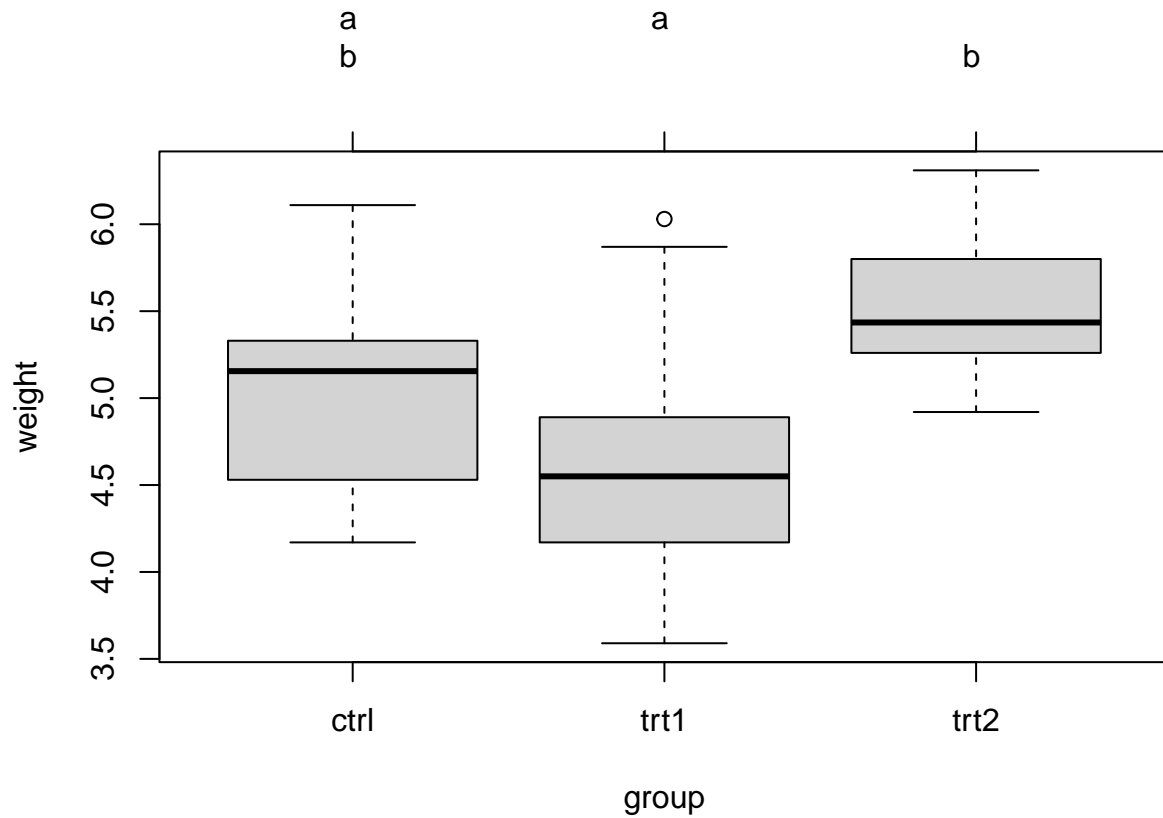
```
> library(rstatix)


##
##      'rstatix'


## The following object is masked from 'package:MASS':
##
##      select


## The following object is masked from 'package:stats':
##
##      filter

> my_data %>% dunn_test(weight ~ group)


## # A tibble: 3 x 9
##    .y.     group1 group2    n1    n2 statistic       p  p.adj p.adj.signif
## * <chr>   <chr>  <chr>  <int> <int>     <dbl>   <dbl>  <dbl> <chr>
## 1 weight ctrl    trt1      10    10     -1.12 0.264    0.264 ns
## 2 weight ctrl    trt2      10    10      1.69 0.0912   0.182 ns
## 3 weight trt1    trt2      10    10      2.81 0.00500 0.0150 *
```
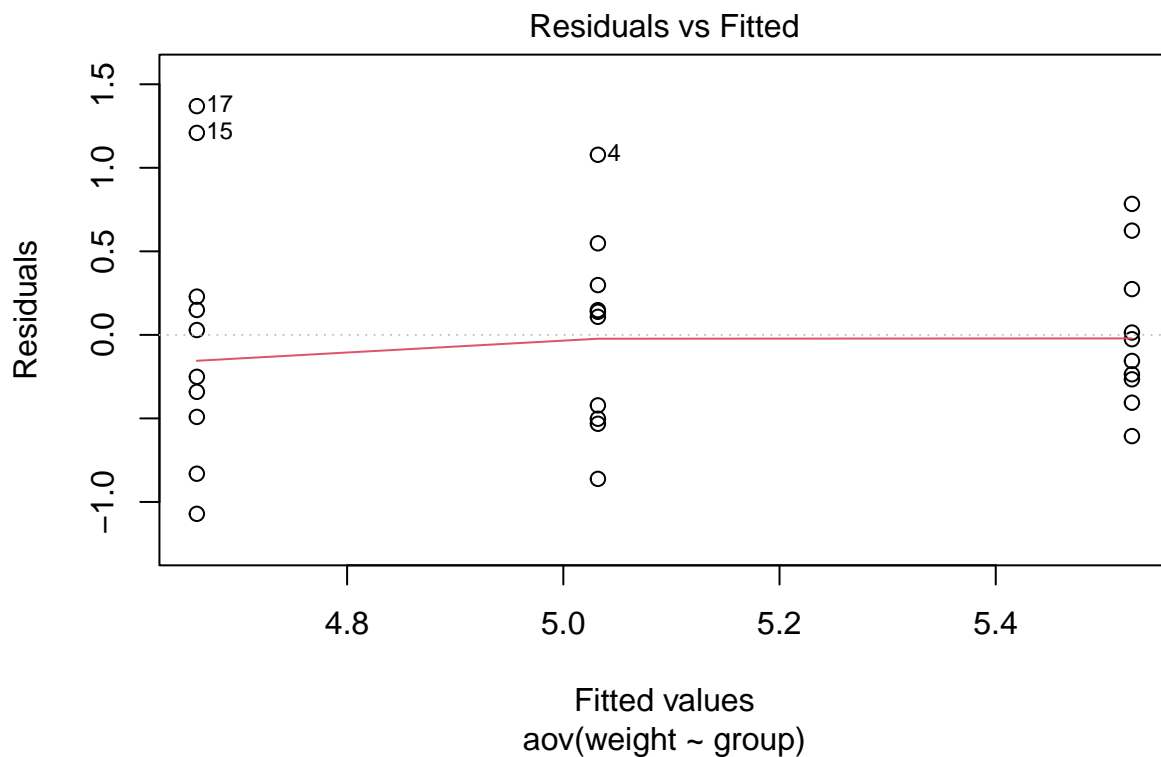
```
> pairwise.t.test(my_data$weight, my_data$group,
+                 p.adjust.method = "BH")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  my_data$weight and my_data$group
##
##      ctrl  trt1
## trt1 0.194 -
## trt2 0.132 0.013
##
## P value adjustment method: BH
```

```
> # 1. Homogeneity of variances
> plot(res.aov, 1)
```


Residuals vs Fitted

```
> bartlett.test(weight ~ group, data = my_data)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  weight by group
## Bartlett's K-squared = 2.8786, df = 2, p-value = 0.2371
```

```
> fligner.test(weight ~ group, data = my_data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  weight by group
## Fligner-Killeen:med chi-squared = 2.3499, df = 2, p-value = 0.3088
```

```
> library(car)
```

```
##      carData
```

```
##
##     'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
> leveneTest(weight ~ group, data = my_data)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  2  1.1192 0.3412
##       27
```

```
> library(rstatix)
> my_data %>%
+   group_by(group) %>%
+   identify_outliers(weight)
```

```
## # A tibble: 2 x 4
##   group weight is.outlier is.extreme
##   <fct>  <dbl> <lgl>      <lgl>
## 1 trt1    5.87 TRUE       FALSE
## 2 trt1    6.03 TRUE       FALSE
```

```
> outlierTest(res.aov)
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferroni p
## 17 2.537341           0.017509       0.52526
```

```
> ## ANOVA test with no assumption of equal variances
> oneway.test(weight ~ group, data = my_data)
```

```
## 
##   One-way analysis of means (not assuming equal variances)
## 
## data:  weight and group
## F = 5.181, num df = 2.000, denom df = 17.128, p-value = 0.01739
```

```
> ## Pairwise t-tests with no assumption of equal variances
> pairwise.t.test(my_data$weight, my_data$group,
+                 p.adjust.method = "BH", pool.sd = FALSE)
```

```
## 
##   Pairwise comparisons using t tests with non-pooled SD
## 
## data:  my_data$weight and my_data$group
## 
##       ctrl  trt1
## trt1 0.250 -
## trt2 0.072 0.028
## 
## P value adjustment method: BH
```

```
> # 2. Normality
> plot(res.aov, 2)
```


Normal Q–Q
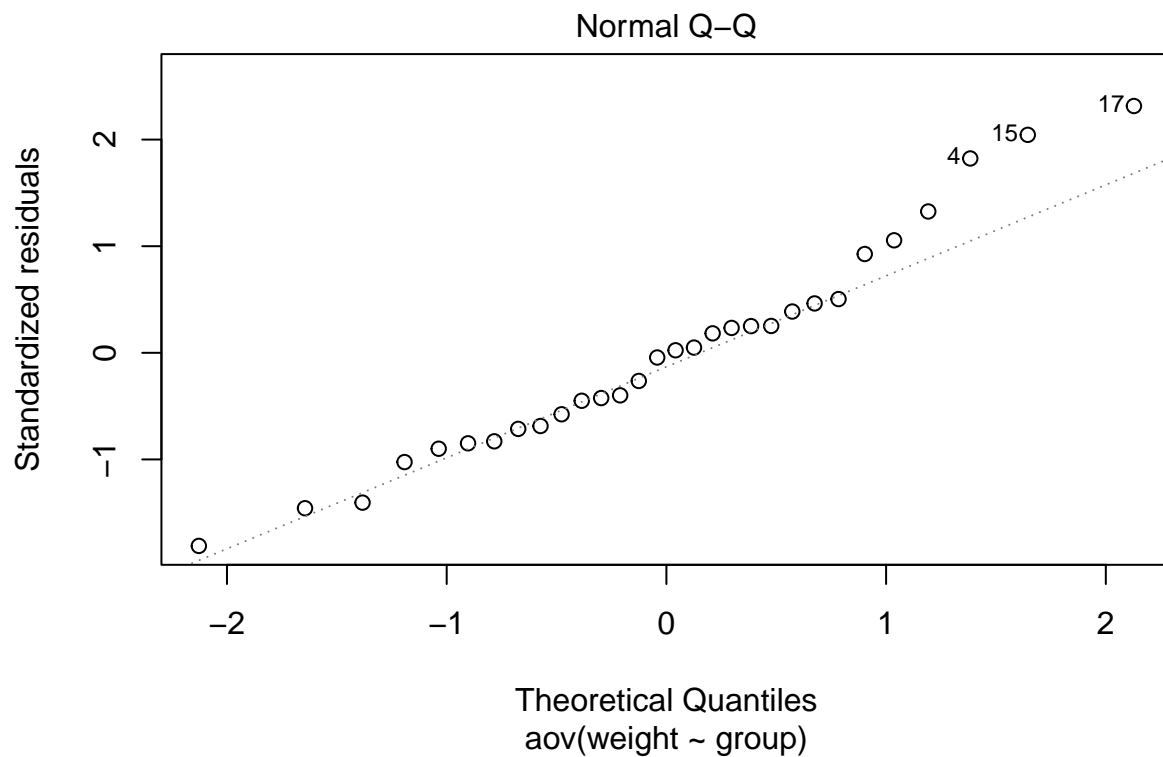
aov(weight ~ group)

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

```
##
##  Shapiro-Wilk normality test
##
## data:  aov_residuals
## W = 0.96607, p-value = 0.4379
```

```
> kruskal.test(weight ~ group, data = my_data)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  weight by group
## Kruskal-Wallis chi-squared = 7.9882, df = 2, p-value = 0.01842
```

```
> data("ToothGrowth")
> head(ToothGrowth)
```

```
##    len supp dose
## 1  4.2   VC  0.5
## 2 11.5   VC  0.5
## 3  7.3   VC  0.5
## 4  5.8   VC  0.5
## 5  6.4   VC  0.5
## 6 10.0   VC  0.5
```

```
> # Store the data in the variable my_data
> my_data <- ToothGrowth
> head(my_data)
```

```
##    len supp dose
## 1  4.2   VC  0.5
## 2 11.5   VC  0.5
## 3  7.3   VC  0.5
## 4  5.8   VC  0.5
## 5  6.4   VC  0.5
## 6 10.0   VC  0.5
```
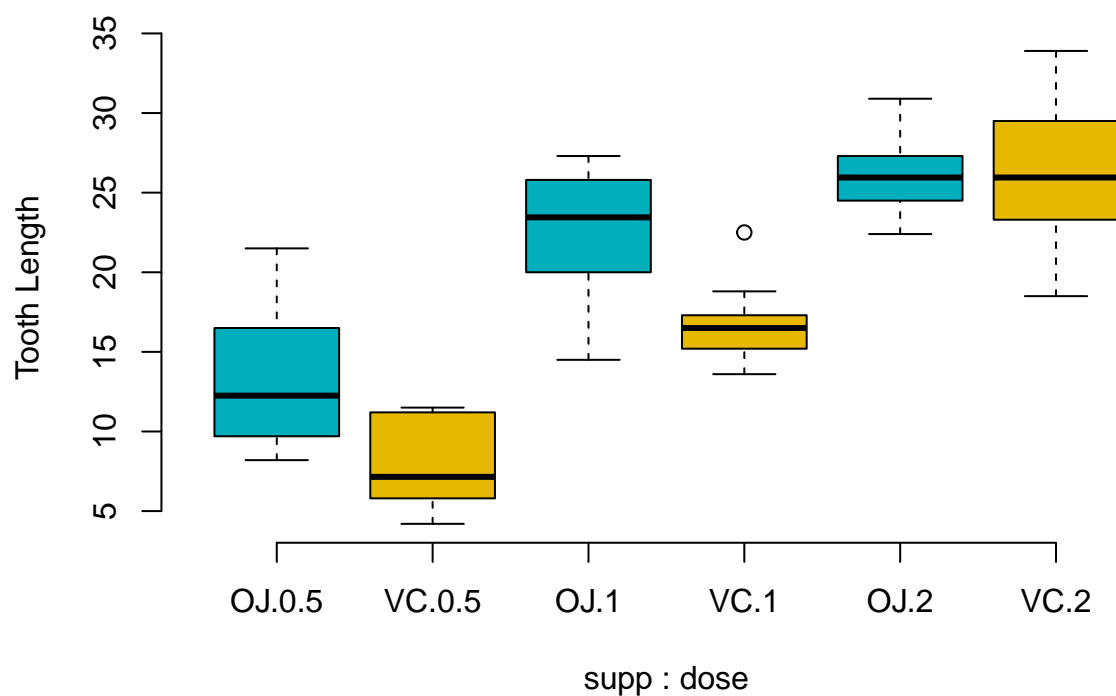
```
> str(my_data)
```

```
## 'data.frame':    60 obs. of  3 variables:
##  $ len : num  4.2 11.5 7.3 5.8 6.4 10 11.2 11.2 5.2 7 ...
##  $ supp: Factor w/ 2 levels "OJ","VC": 2 2 2 2 2 2 2 2 2 2 ...
##  $ dose: num  0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
```

```
> # Generate frequency tables:
> table(my_data$supp, my_data$dose)


##
##      0.5  1  2
##   OJ  10 10 10
##   VC  10 10 10

> # Box plot with two factor variables
> boxplot(len ~ supp * dose, data=my_data, frame = FALSE,
+         col = c("#00AFBB", "#E7B800"), ylab="Tooth Length")
```



```
> # Two-way interaction plot
> with(my_data,
+     interaction.plot(x.factor = dose, trace.factor = supp,
+                      response = len, fun = mean,
+                      type = "b", legend = TRUE,
+                      xlab = "Dose", ylab="Tooth Length",
+                      pch=c(1,19), col = c("#00AFBB", "#E7B800"))
+ )
```
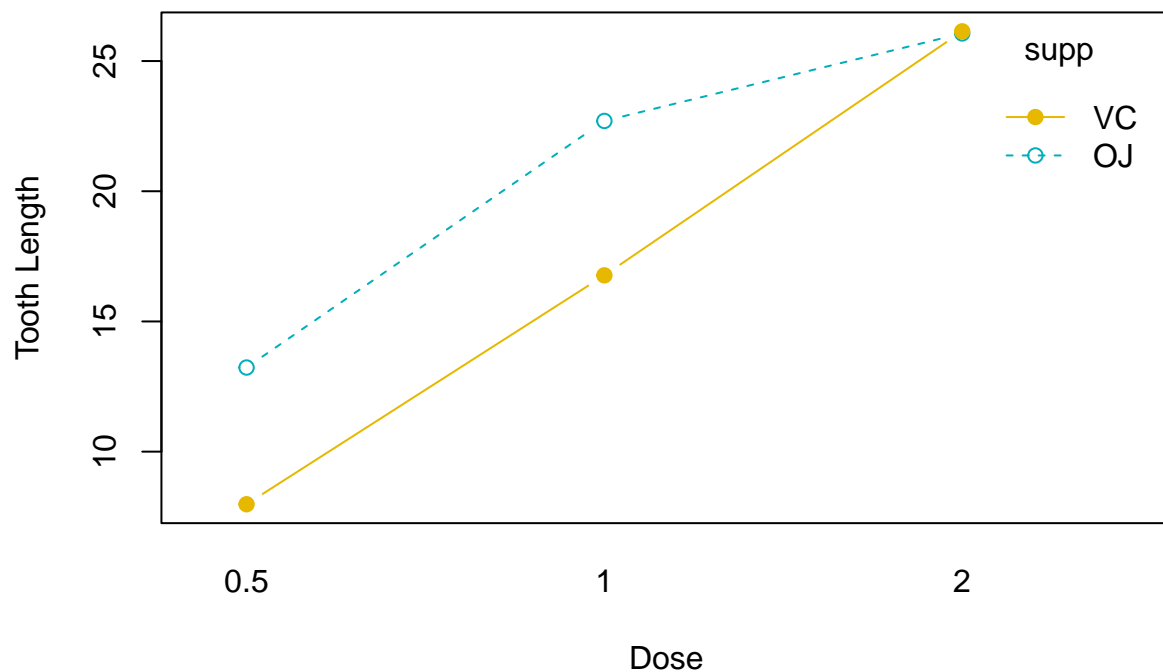
```
> library(gplots)


##
##      'gplots'

## The following object is masked from 'package:stats':
##
##      lowess

> plotmeans(len ~ interaction(supp, dose, sep = " "), data = my_data,
+           connect = list(c(1,3,5), c(2,4,6)),
+           main = "Interaction Plot with 95% CIs",
+           xlab = "Treatment and Dose Combination", ylab="Tooth Length",
+           barcol = c("#00AFBB", "#E7B800"), col = c("#00AFBB", "#E7B800"))
```
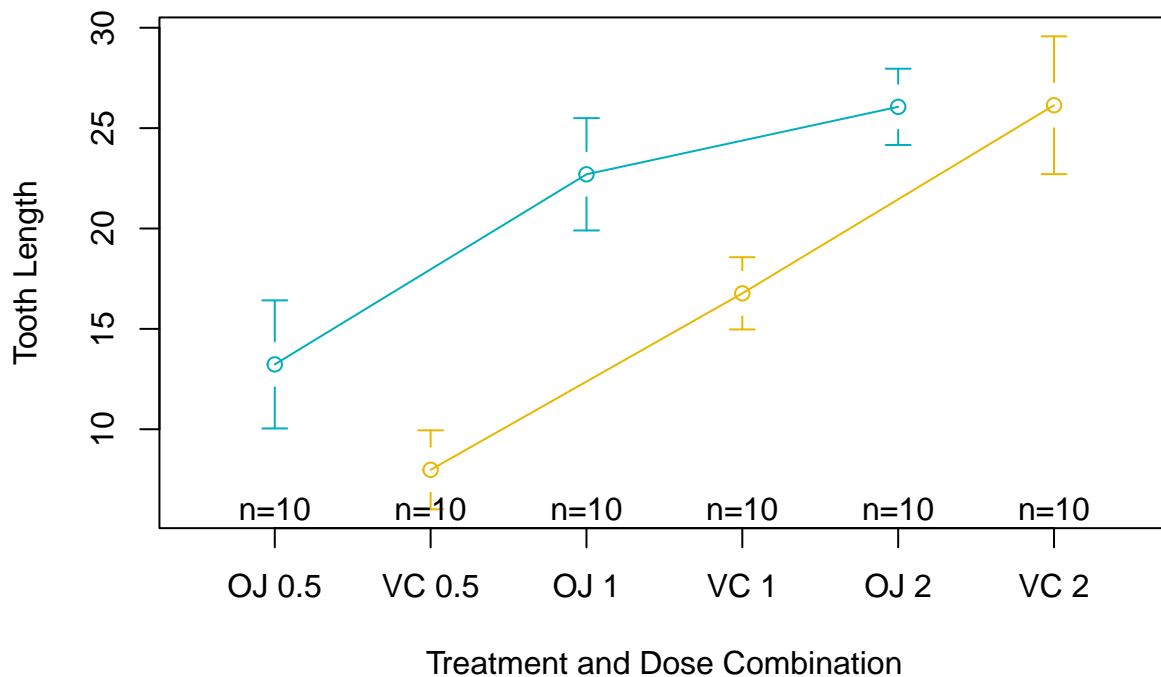
Interaction Plot with 95% CIs


```
> res.aov2 <- aov(len ~ supp + dose, data = my_data)
> summary(res.aov2)


##              Df Sum Sq Mean Sq F value   Pr(>F)
## supp          1  205.4   205.4   11.45   0.0013 **
## dose          1 2224.3  2224.3  123.99 6.31e-16 ***
## Residuals    57 1022.6    17.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> # Two-way ANOVA with interaction effect
> # These two calls are equivalent
> res.aov3 <- aov(len ~ supp * dose, data = my_data)
> res.aov3 <- aov(len ~ supp + dose + supp:dose, data = my_data)
> summary(res.aov3)


##              Df Sum Sq Mean Sq F value   Pr(>F)
## supp          1  205.4   205.4  12.317 0.000894 ***
## dose          1 2224.3  2224.3 133.415  < 2e-16 ***
## supp:dose     1   88.9    88.9   5.333 0.024631 *
## Residuals    56  933.6    16.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> require("dplyr")
> group_by(my_data, supp, dose) %>%
+   summarise(
+     count = n(),
+     mean = mean(len, na.rm = TRUE),
+     sd = sd(len, na.rm = TRUE)
+   )
```

```
## `summarise()` has grouped output by 'supp'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 6 x 5
## # Groups:   supp [2]
##   supp   dose count  mean    sd
##   <fct> <dbl> <int> <dbl> <dbl>
## 1 OJ      0.5    10 13.2   4.46
## 2 OJ      1      10 22.7   3.91
## 3 OJ      2      10 26.1   2.66
## 4 VC      0.5    10  7.98  2.75
## 5 VC      1      10 16.8   2.52
## 6 VC      2      10 26.1   4.80
```

```
> model.tables(res.aov3, type="means", se = TRUE)
```

```
## Warning in replications(paste("~", xx), data = mf):    : dose

## Warning in replications(paste("~", xx), data = mf):    : supp, dose

## Tables of means
## Grand mean
##
## 18.81333
##
##   supp
## supp
##     OJ     VC
## 20.663 16.963
##
##   dose
## dose
##    0.5      1      2
## 12.304 17.186 26.950
##
##   supp:dose
##      dose
## supp 0.5    1      2
##   OJ 15.456 19.361 27.173
##   VC  9.153 15.011 26.726
##
## Standard errors for differences of means
##          supp
##         1.054
## replic.    30
```

```
> library(HH)
```

Warning: 'HH' R 4.2.2

lattice

grid

latticeExtra

Warning: 'latticeExtra' R 4.2.1

gridExtra

##
'gridExtra'

The following object is masked from 'package:dplyr':
##
combine

##
'HH'

The following object is masked from 'package:gplots':
##
residplot

The following objects are masked from 'package:car':
##
logit, vif