Test

Steph Gervasi

5/13/2019

Contents

Load package for this demo

```
data(iris)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Run very short EDA

```
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
levels(iris$Species)
```

```
## [1] "setosa"     "versicolor" "virginica"
```

Summary Stats

```
# install.packages("tidyverse")
library(tidyverse)
```

```
## -- Attaching packages --------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.8
## v tidyr   0.8.1     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

```
## -- Conflicts ------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
group_by(iris, Species) %>%
  summarise(
    count = n(),
    mean = mean(Sepal.Length, na.rm = TRUE),
    sd = sd(Sepal.Length, na.rm = TRUE)
  )
```

```
## # A tibble: 3 x 4
##   Species    count  mean    sd
##   <fct>      <int> <dbl> <dbl>
## 1 setosa        50  5.01 0.352
## 2 versicolor    50  5.94 0.516
## 3 virginica     50  6.59 0.636
```

Visualizations

Boxplot with ggpubr

```
# install.packages("ggpubr")
library(ggpubr)
```

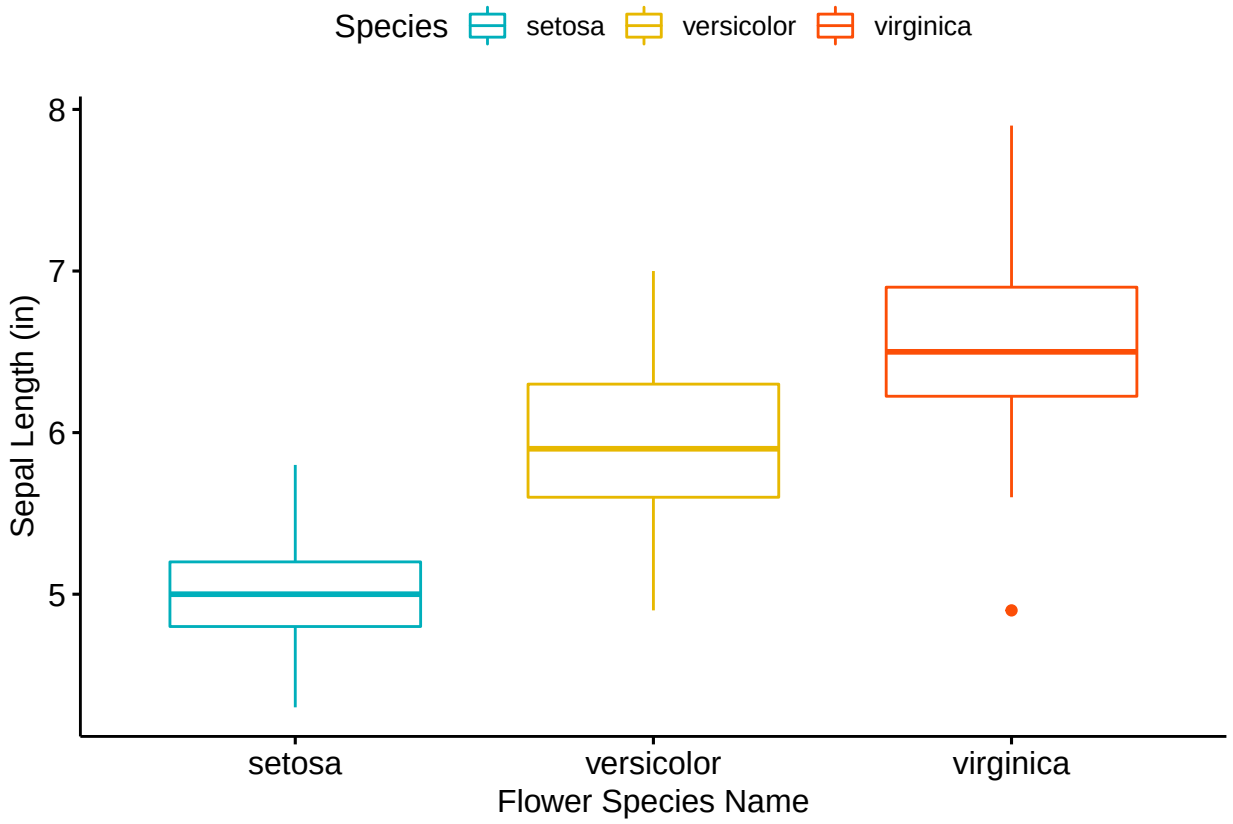
```
## Loading required package: magrittr
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
```

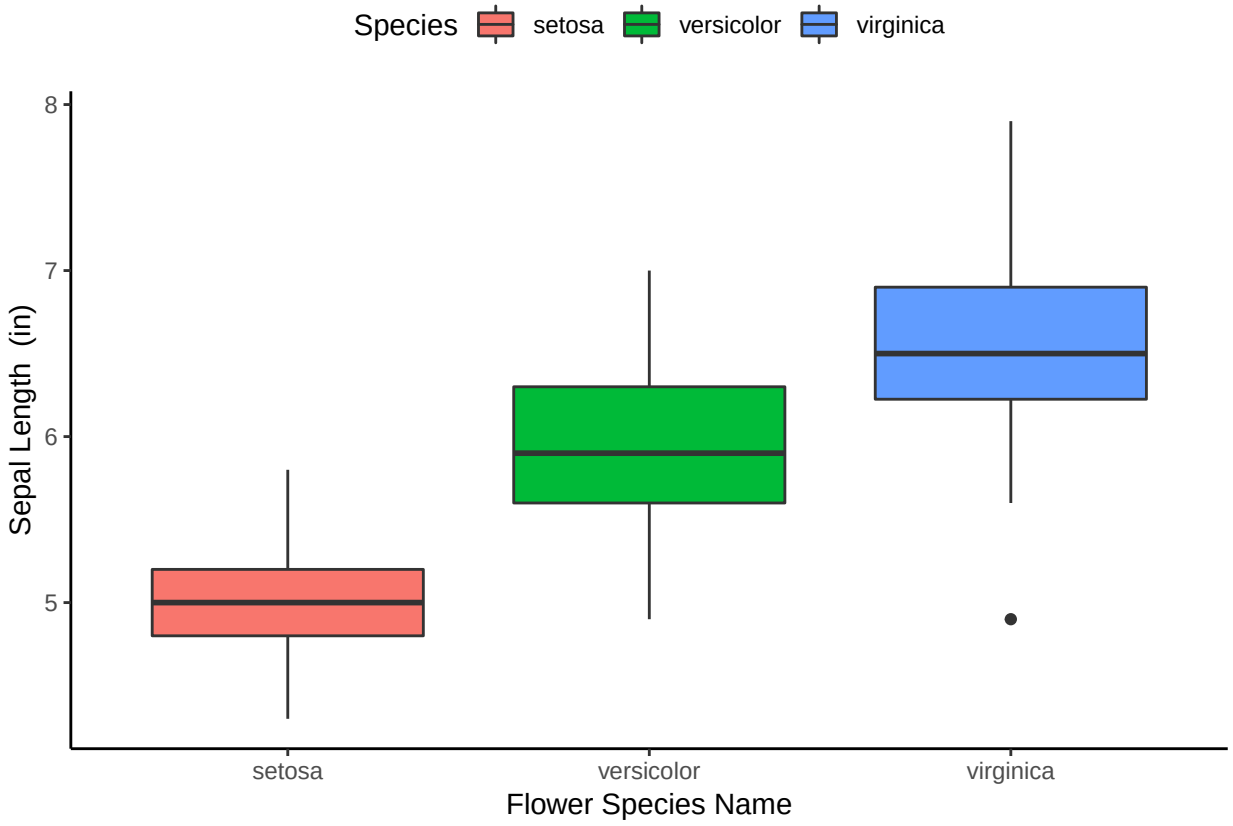
```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
ggboxplot(iris, x = "Species", y = "Sepal.Length",
          color = "Species", palette = c("#00AFBB", "#E7B800", "#FC4E07"),
          order = c("setosa", "versicolor", "virginica"),
          ylab = "Sepal Length (in)", xlab = "Flower Species Name")
```



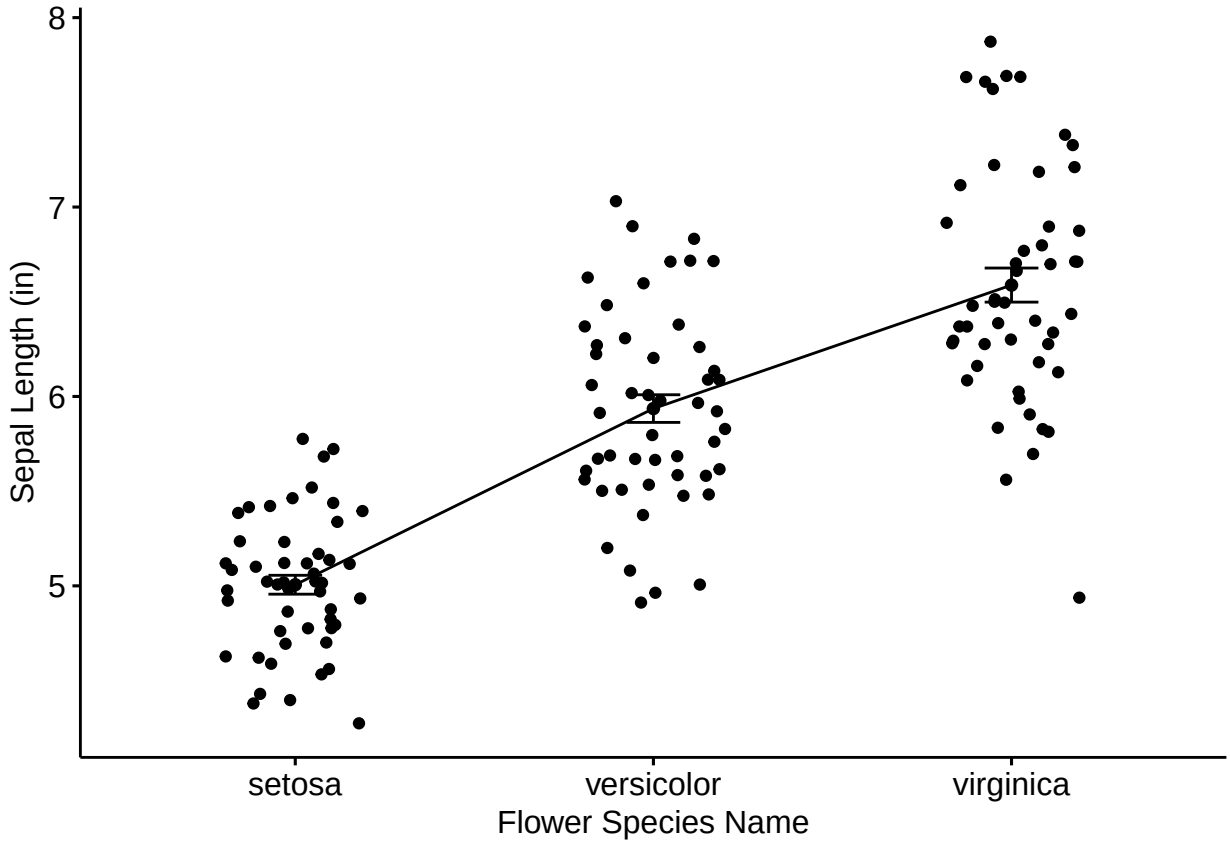
Boxplot with ggplot2

```
library(ggplot2)
ggplot(iris, aes(x=Species, y=Sepal.Length, fill=Species)) +
  geom_boxplot() + theme_classic() + theme(legend.position = "top") +
  labs(x="Flower Species Name", y = "Sepal Length  (in)")
```



Dot and line plot with ggpubr

```
ggline(iris, x = "Species", y = "Sepal.Length",
       add = c("mean_se", "jitter"),
       order = c("setosa", "versicolor", "virginica"),
       ylab = "Sepal Length (in)", xlab = "Flower Species Name")
```



Run group comparison analysis (ANOVA)

```
# Compute the analysis of variance
aov1 <- aov(Sepal.Length ~ Species, data = iris)

# Summary of the analysis
summary(aov1)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Species       2  63.21  31.606   119.3 <2e-16 ***
## Residuals   147  38.96   0.265
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Posthoc tests
TukeyHSD(aov1)
```

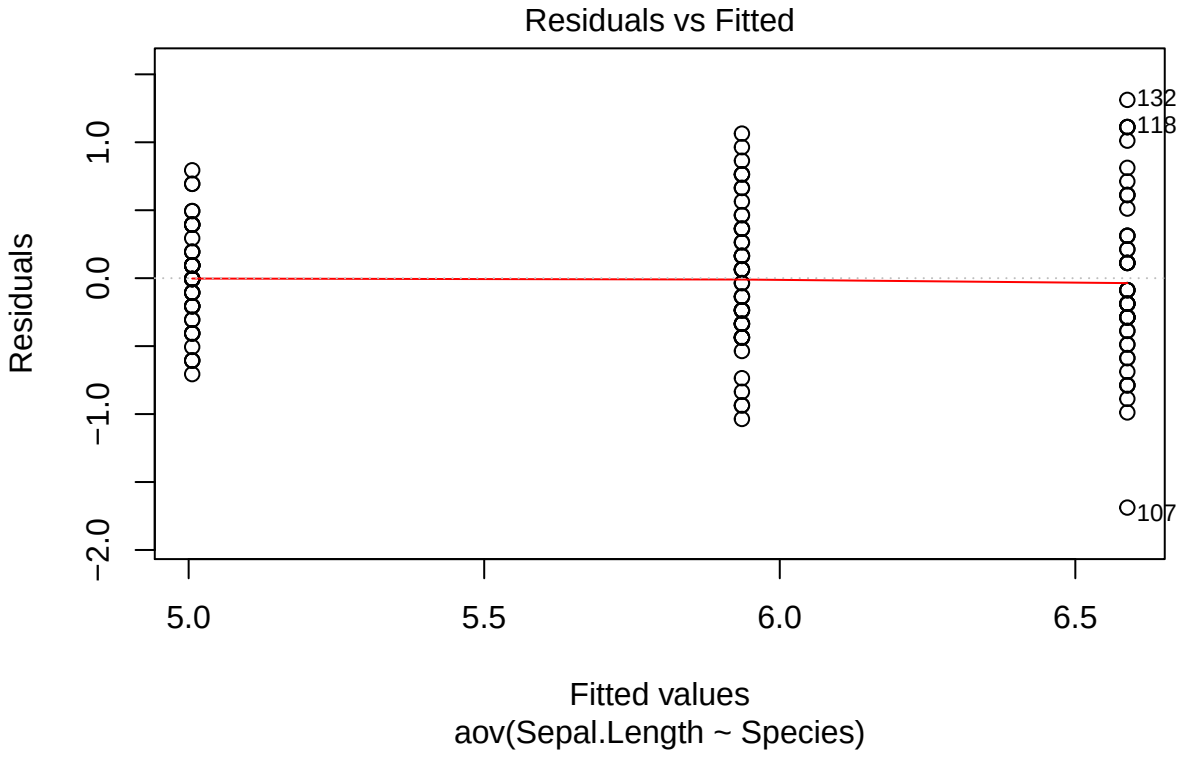
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Sepal.Length ~ Species, data = iris)
##
## $Species
##                       diff       lwr       upr p adj
## versicolor-setosa    0.930 0.6862273 1.1737727     0
## virginica-setosa     1.582 1.3382273 1.8257727     0
```

```
## virginica-versicolor 0.652 0.4082273 0.8957727     0
```

Check assumptions of analysis/model

Check homogeneity of variance assumption

```
plot(aov1, 1)
```




```
#plot(aov1)

library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

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

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
leveneTest(Sepal.Length ~ Species, data = iris)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value   Pr(>F)
## group   2  6.3527 0.002259 **
```

```
##       147
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# From the output we can see that the p-value is less than the significance level of 0.05. This means t

# It is possible to run a Welch's ANOVA where the assumption of homogeneity of variance is relaxed. It

oneway.test(Sepal.Length ~ Species, data = iris)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  Sepal.Length and Species
## F = 138.91, num df = 2.000, denom df = 92.211, p-value < 2.2e-16
```

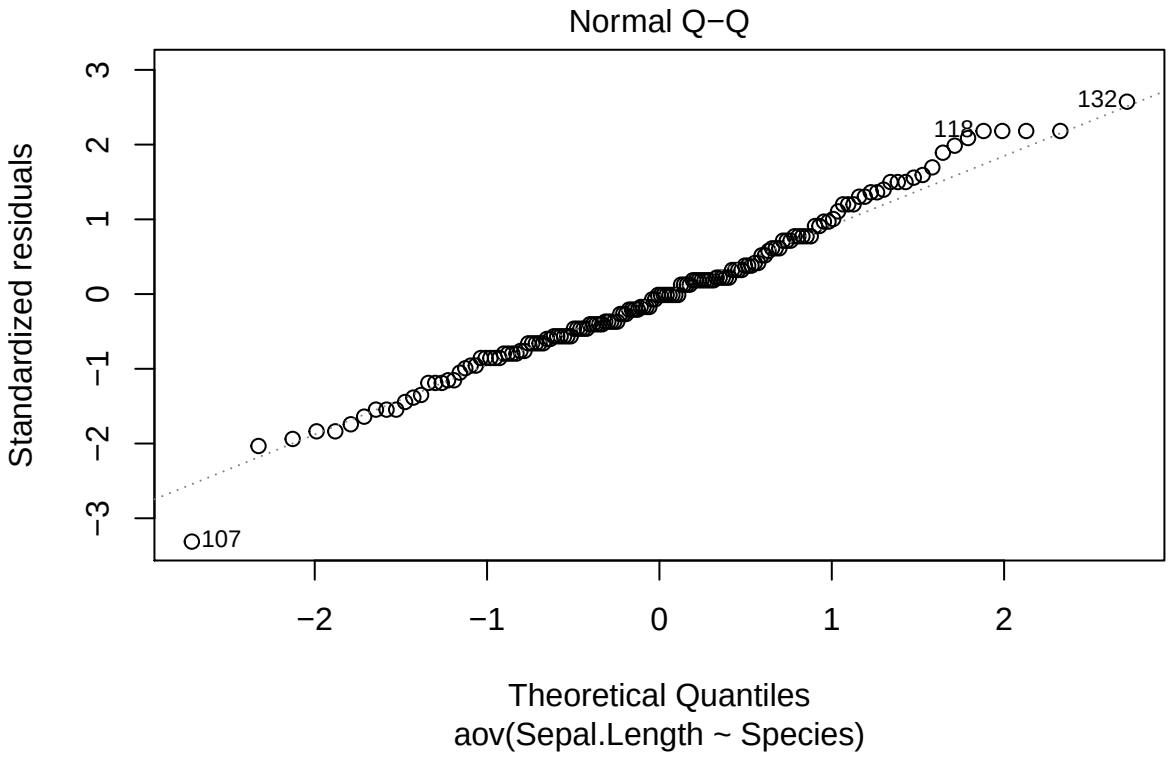
```
# Now pairwise t tests on this output:

pairwise.t.test(iris$Sepal.Length, iris$Species,
                p.adjust.method = "BH", pool.sd = FALSE)
```

```
##
##  Pairwise comparisons using t tests with non-pooled SD
##
## data:  iris$Sepal.Length and iris$Species
##
##            setosa  versicolor
## versicolor < 2e-16 -
## virginica  < 2e-16 1.9e-07
##
## P value adjustment method: BH
```

Check normality assumption

```
plot(aov1, 2)
```




```
# Additional check for normality besides QQ plot produced above.

# Extract the residuals
aov_residuals <- residuals(object = aov1)

# Run Shapiro-Wilk test
shapiro.test(x = aov_residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  aov_residuals
## W = 0.9879, p-value = 0.2189
```

```
# If this had been significant, it would indicate that we were in violation of the assumption of normal

kruskal.test(Sepal.Length ~ Species, data = iris)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Sepal.Length by Species
## Kruskal-Wallis chi-squared = 96.937, df = 2, p-value < 2.2e-16
```

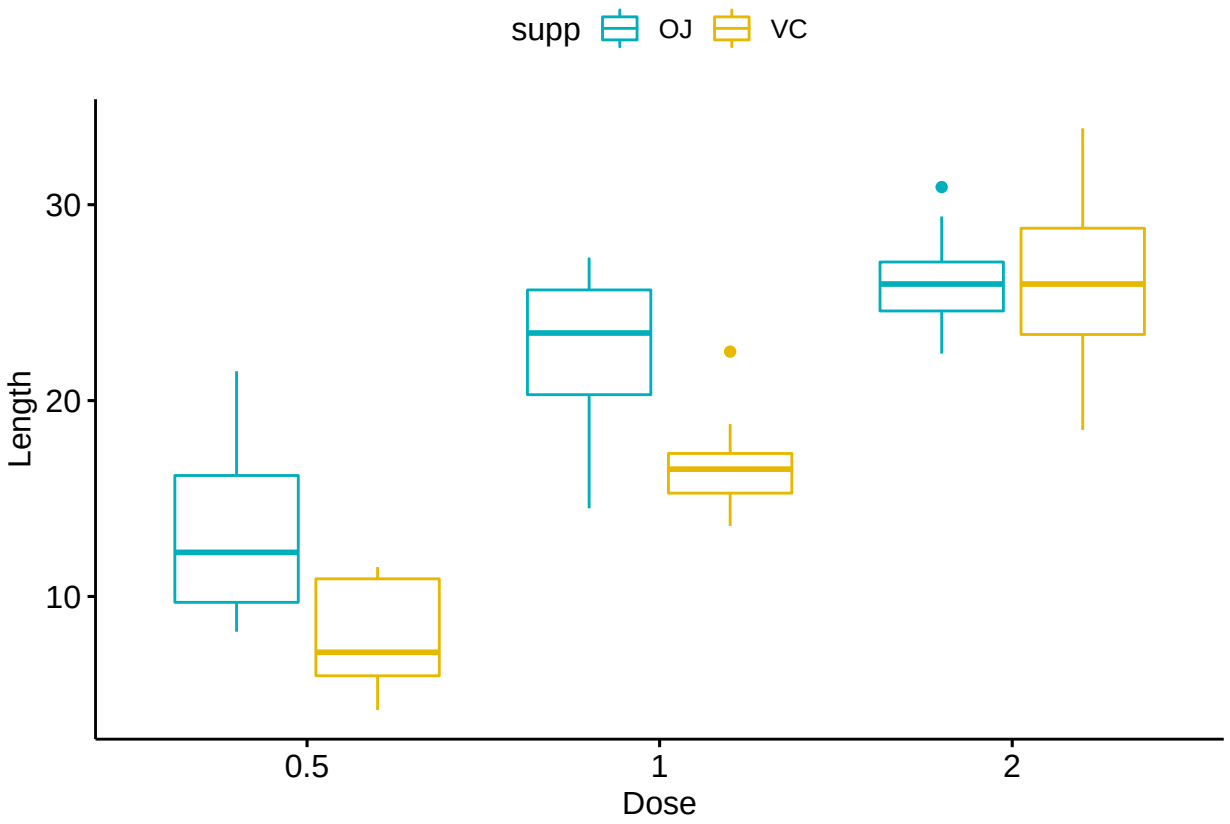
Some exploration with plotting in ggpubr

Multiple groups and Faceting

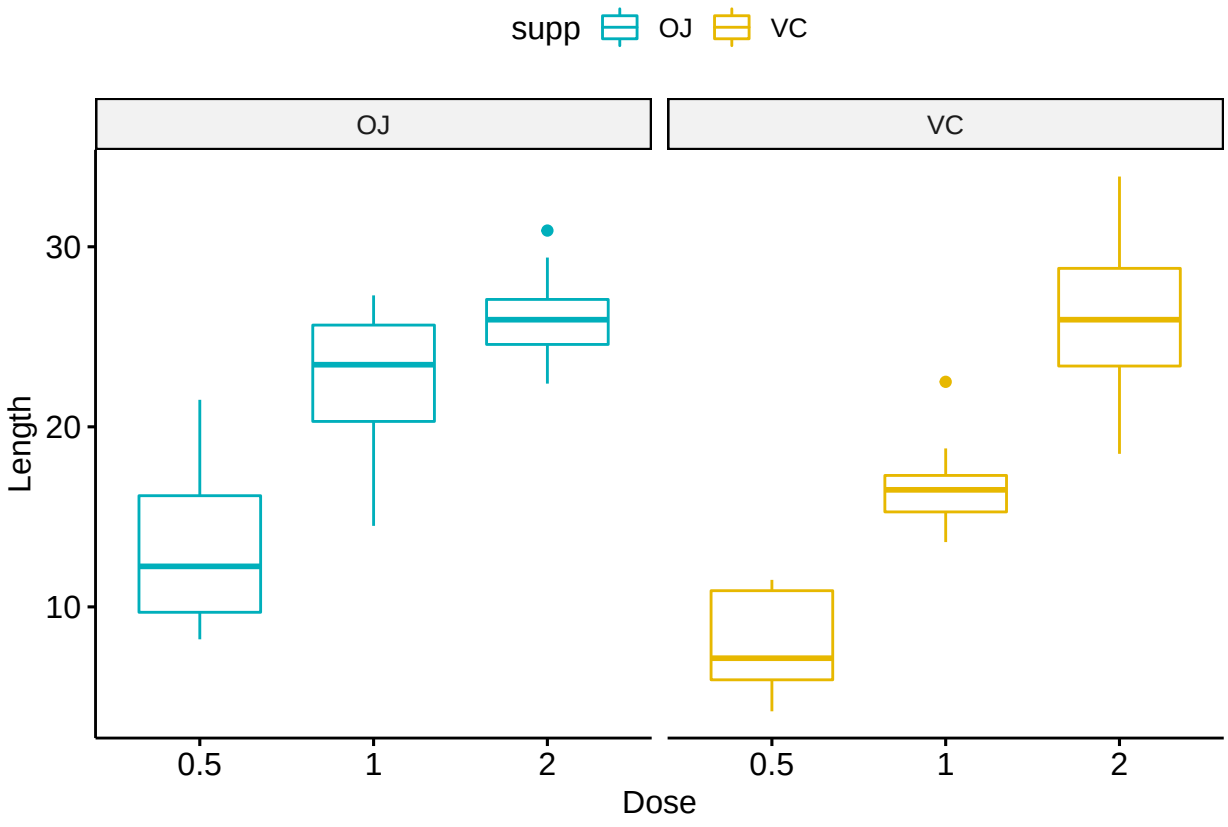
```
data(ToothGrowth)
head(ToothGrowth)
```

```
##    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
```

```
p <- ggboxplot(ToothGrowth, x = "dose", y = "len",
      color = "supp", palette = c("#00AFBB", "#E7B800"),
      ylab = "Length", xlab = "Dose")
print(p)
```

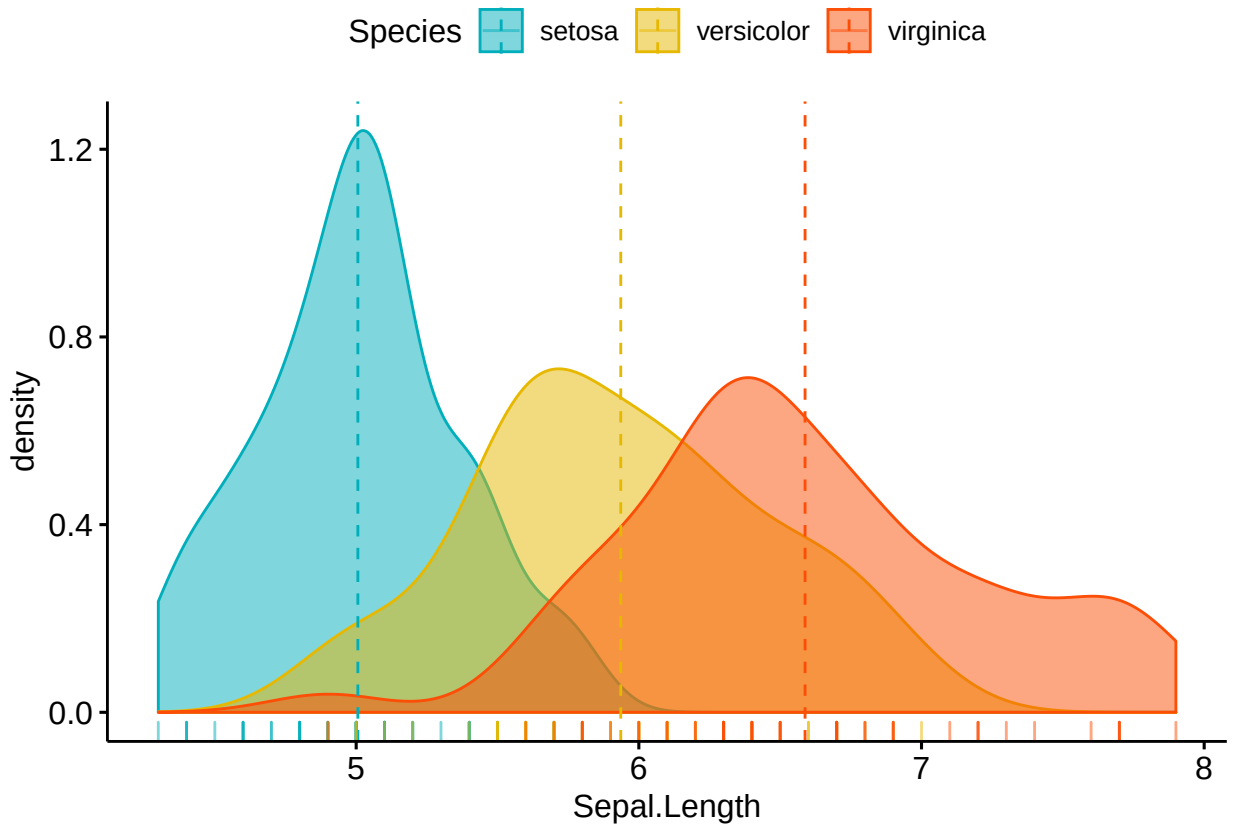



```
facet(p, facet.by = "supp")
```



Density plots

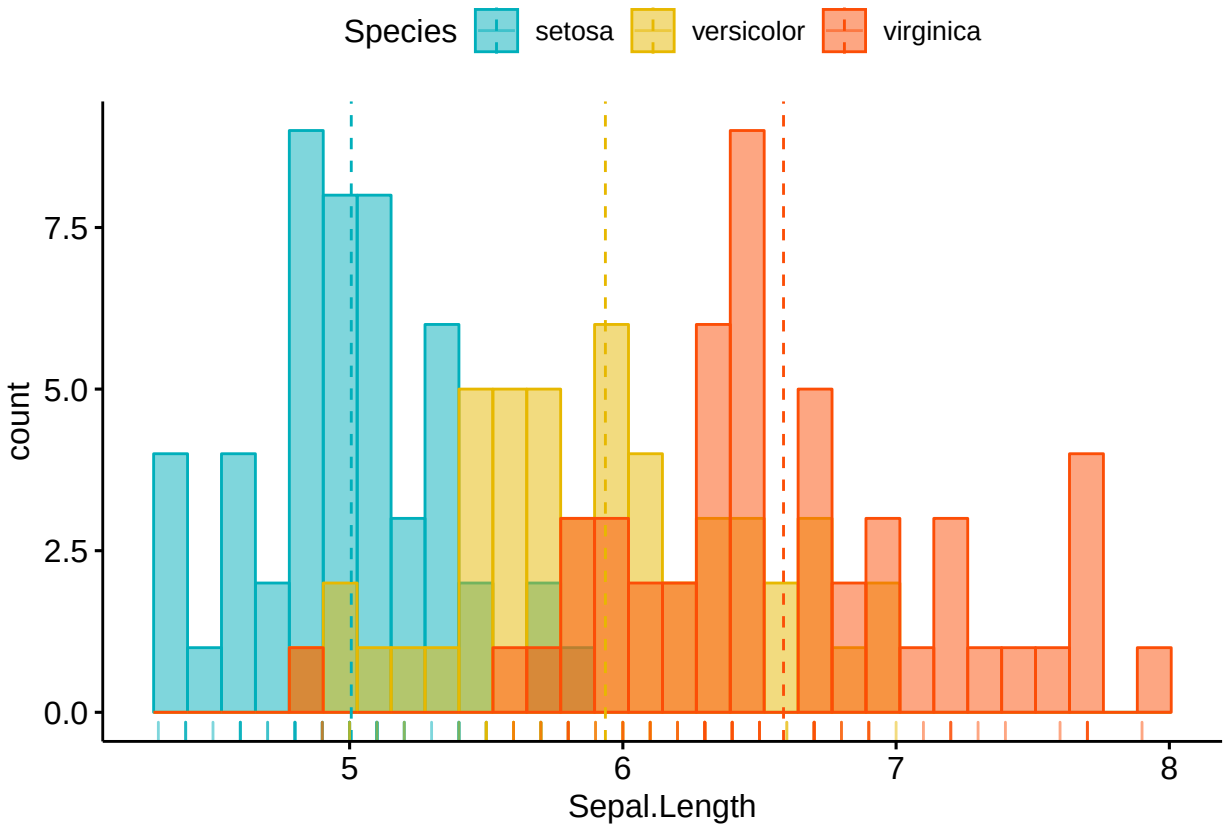
```
# Density plot with mean lines and marginal rug
# :::::::::::::::::::::::::::::::::::::::::::::::::
# Change outline and fill colors by groups
# Use custom palette
ggdensity(iris, x = "Sepal.Length",
   add = "mean", rug = TRUE,
   color = "Species", fill = "Species",
   palette = c("#00AFBB", "#E7B800", "#FC4E07"))
```



Histograms

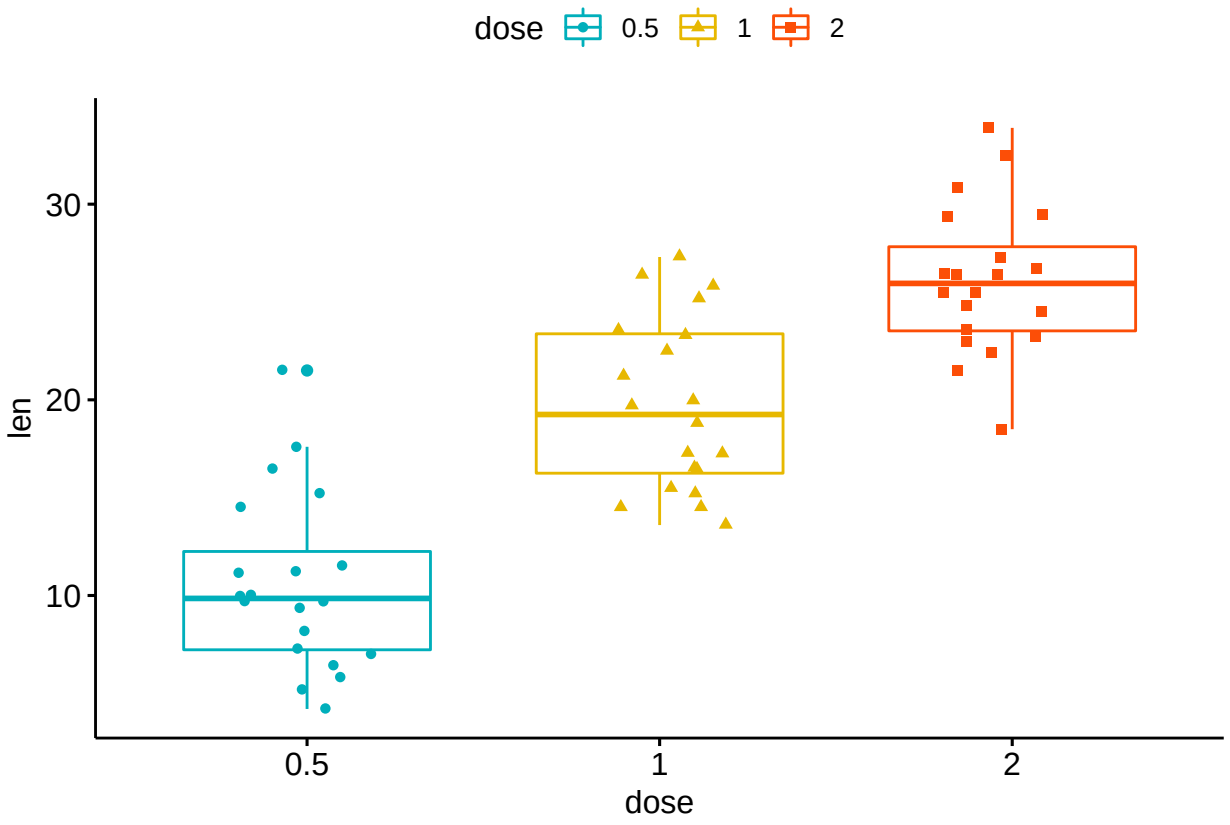
```
# Histogram plot with mean lines and marginal rug
# :::::::::::::::::::::::::::::::::::::::::::::::::
# Change outline and fill colors by groups ("sex")
# Use custom color palette
gghistogram(iris, x = "Sepal.Length",
   add = "mean", rug = TRUE,
   color = "Species", fill = "Species",
   palette = c("#00AFBB", "#E7B800", "#FC4E07"))
```

```
## Warning: Using `bins = 30` by default. Pick better value with the argument
## `bins`.
```



Boxplots with jittered points

```
# Box plots with jittered points
# :::::::::::::::::::::::::::::::::::::::::::::
# Change outline colors by groups: dose
# Use custom color palette
# Add jitter points and change the shape by groups
 p <- ggboxplot(ToothGrowth, x = "dose", y = "len",
                color = "dose", palette =c("#00AFBB", "#E7B800", "#FC4E07"),
                add = "jitter", shape = "dose")
 p
```



Kruskal Wallis

```
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
```

```
str(ToothGrowth)
```

```
## '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 ...
```

```
# Convert the dose variable to a factor
tg <- ToothGrowth
tg$dose <- as.factor(tg$dose)
head(tg)
```

```
##    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(tg)
```

```
## '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: Factor w/ 3 levels "0.5","1","2": 1 1 1 1 1 1 1 1 1 1 ...
```

```
# Perform the test
kruskal.test(len ~ dose, data = tg)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  len by dose
## Kruskal-Wallis chi-squared = 40.669, df = 2, p-value = 1.475e-09
```

```
# multiple comparisons with Dunn test
#install.packages("FSA")
library(FSA)
```

```
## Warning: package 'FSA' was built under R version 3.5.2
```

```
## ## FSA v0.8.23. See citation('FSA') if used in publication.
## ## Run fishR() for related website and fishR('IFAR') for related book.
```

```
##
## Attaching package: 'FSA'
```

```
## The following object is masked from 'package:car':
##
##     bootCase
```

```
dunnTest(len ~ dose,
              data=tg,
              method="bh")    # Can adjust p-values;
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison
```

```
##   p-values adjusted with the Benjamini-Hochberg method.
```

```
##   Comparison         Z      P.unadj        P.adj
## 1    0.5 - 1 -3.554911 3.781068e-04 5.671603e-04
## 2    0.5 - 2 -6.362612 1.983517e-10 5.950552e-10
## 3      1 - 2 -2.807701 4.989660e-03 4.989660e-03
```

```
                            # See ?p.adjust for options

# multiple comparisons using wilcoxon test :THIS APPEARS TO BE WHAT IS USED IN THE GRAPHS BELOW!!!
pairwise.wilcox.test(tg$len,
                          tg$dose,
                          p.adjust.method="none")
```

```
## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties
```

```
## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties
```

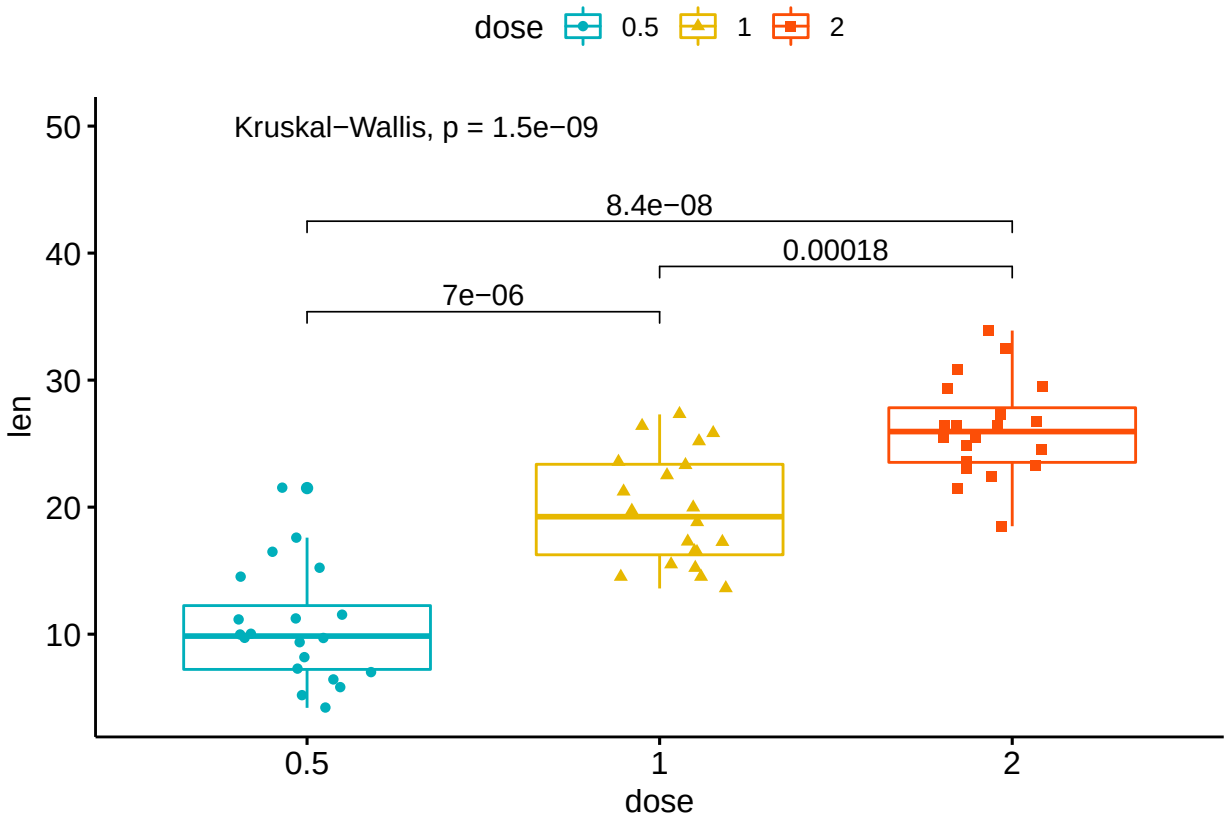
```
## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test
##
## data:  tg$len and tg$dose
##
##   0.5     1
## 1 7.0e-06 -
## 2 8.4e-08 0.00018
##
## P value adjustment method: none
```

```
                                # Can adjust p-values;
                                # See ?p.adjust for options
```

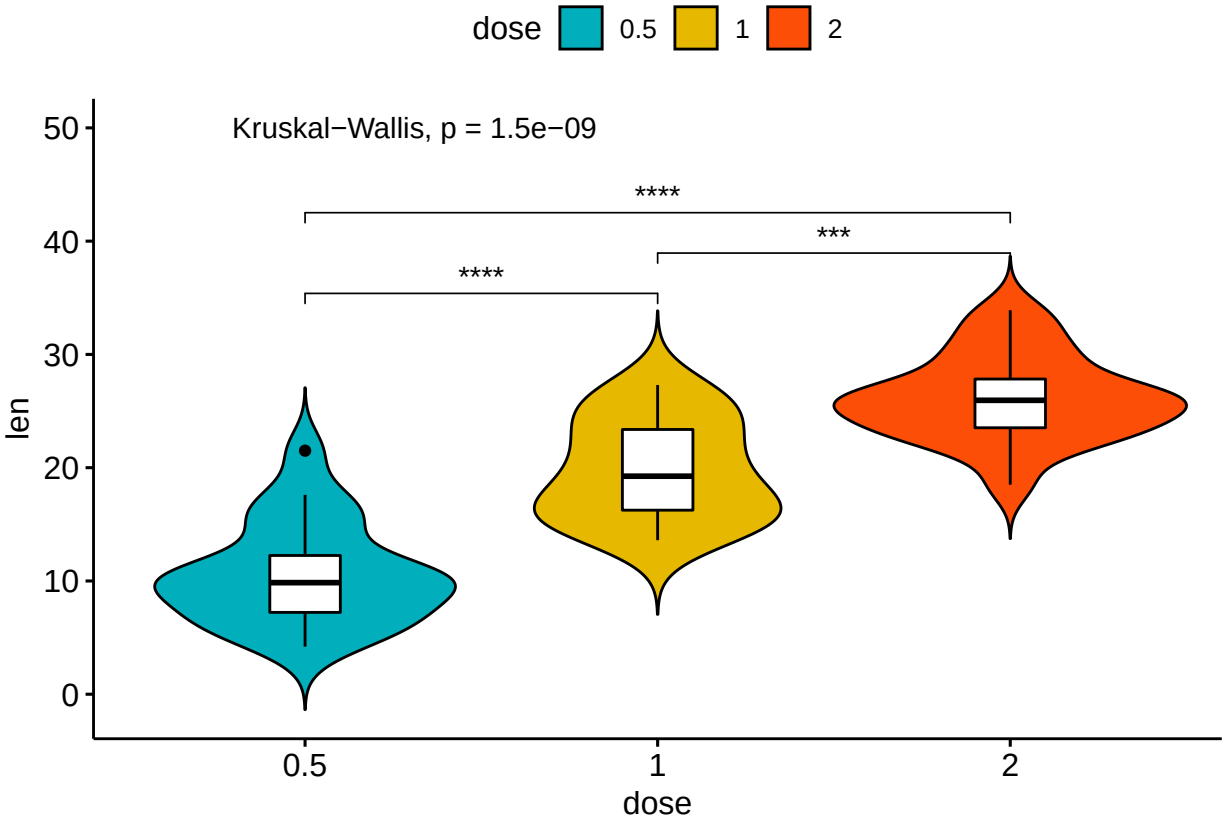
Boxplots with stats output!

```
# Add p-values comparing groups
 # Specify the comparisons you want
my_comparisons <- list( c("0.5", "1"), c("1", "2"), c("0.5", "2") )
p + stat_compare_means(comparisons = my_comparisons)+ # Add pairwise comparisons p-value
  stat_compare_means(label.y = 50)                    # Add global p-value
```



Violin with box plots including stats output

```
# Violin plots with box plots inside
# :::::::::::::::::::::::::::::::::::::::::::::::::
# Change fill color by groups: dose
# add boxplot with white fill color
ggviolin(ToothGrowth, x = "dose", y = "len", fill = "dose",
         palette = c("#00AFBB", "#E7B800", "#FC4E07"),
         add = "boxplot", add.params = list(fill = "white")) +
  stat_compare_means(comparisons = my_comparisons, label = "p.signif")+ # Add significance levels
  stat_compare_means(label.y = 50)                                      # Add global the p-value
```



Bar plots and references

```
# See these references:
# http://www.sthda.com/english/articles/24-ggpubr-publication-ready-plots/
# http://www.sthda.com/english/wiki/ggplot2-box-plot-quick-start-guide-r-software-and-data-visualizatio
# http://www.sthda.com/english/wiki/one-way-anova-test-in-r
# https://rcompanion.org/rcompanion/d_06.html
```