BB503/BB602 - R Training - Week IX

Ege Ulgen

Assessing Normality

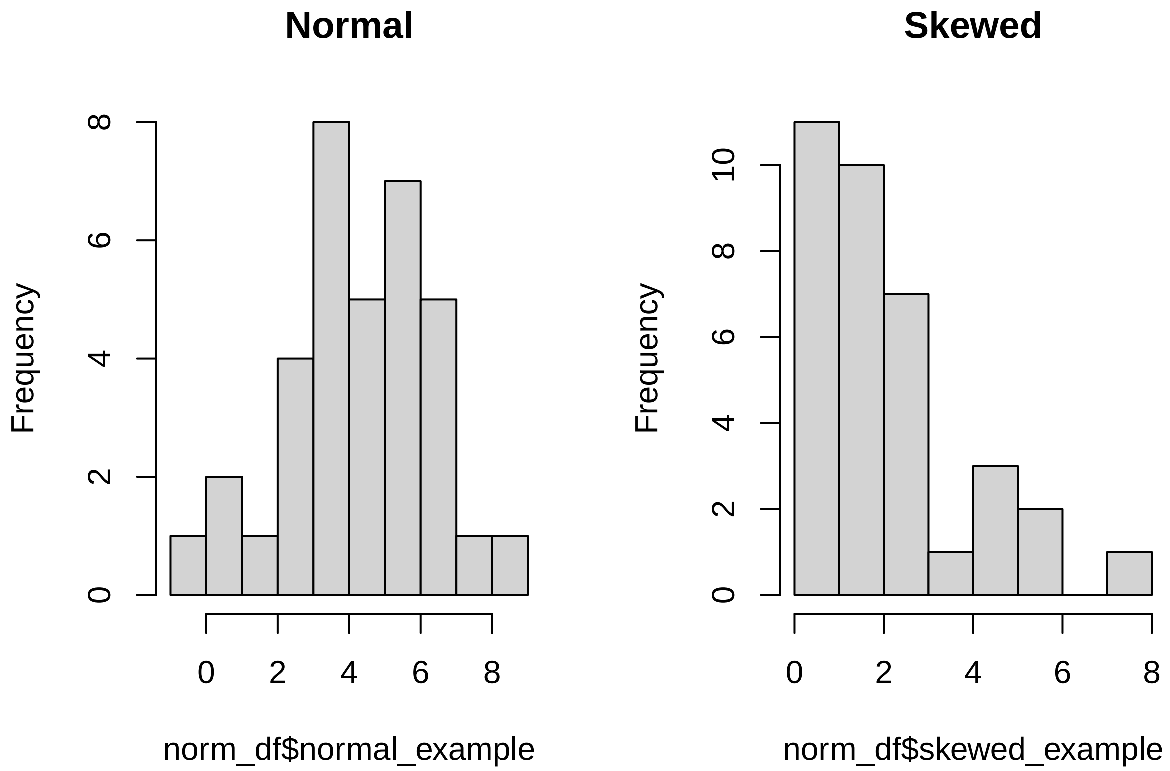
We'll be using "Normal" dataset for this exercise. This dataset contains 2 continuous variables where one is an example of normally distributed data and the other one is an example of skewed data.

```
norm_df <- read.csv("../data/Normal_R.csv")

head(norm_df, 3)
```

```
##   normal_example skewed_example
## 1          -0.50           0.21
## 2           2.47           1.04
## 3           2.54           1.11
```

Histogram

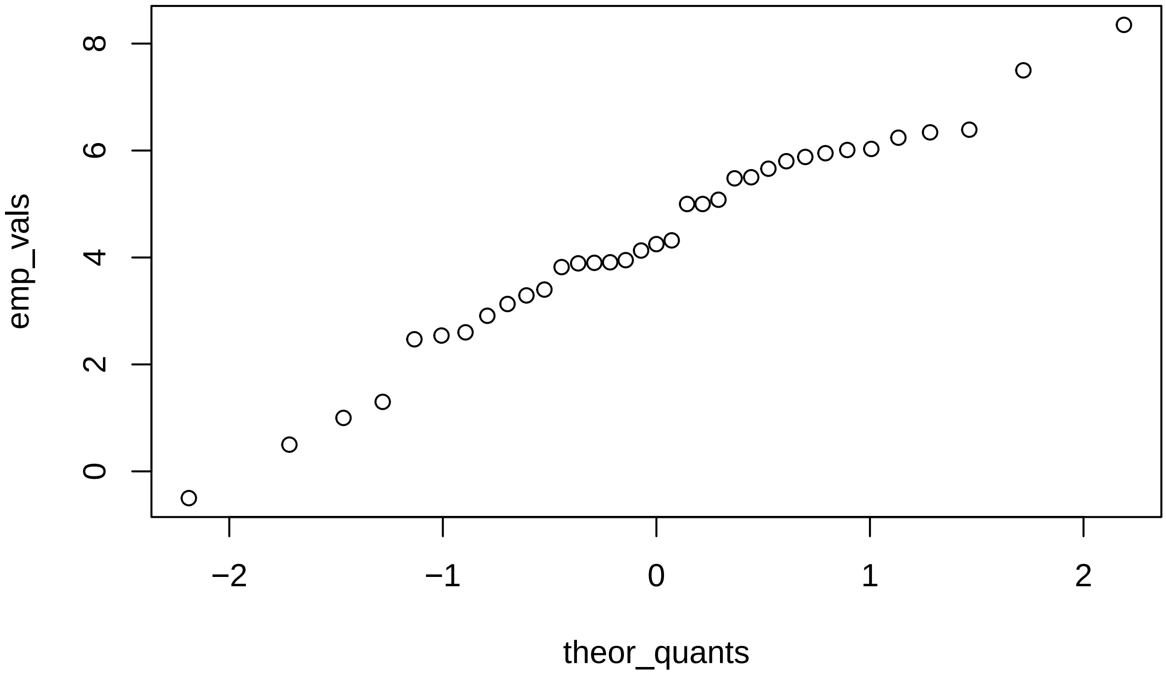
```
par(mfrow = c(1, 2))
hist(norm_df$normal_example, main = "Normal")
hist(norm_df$skewed_example, main = "Skewed")
```



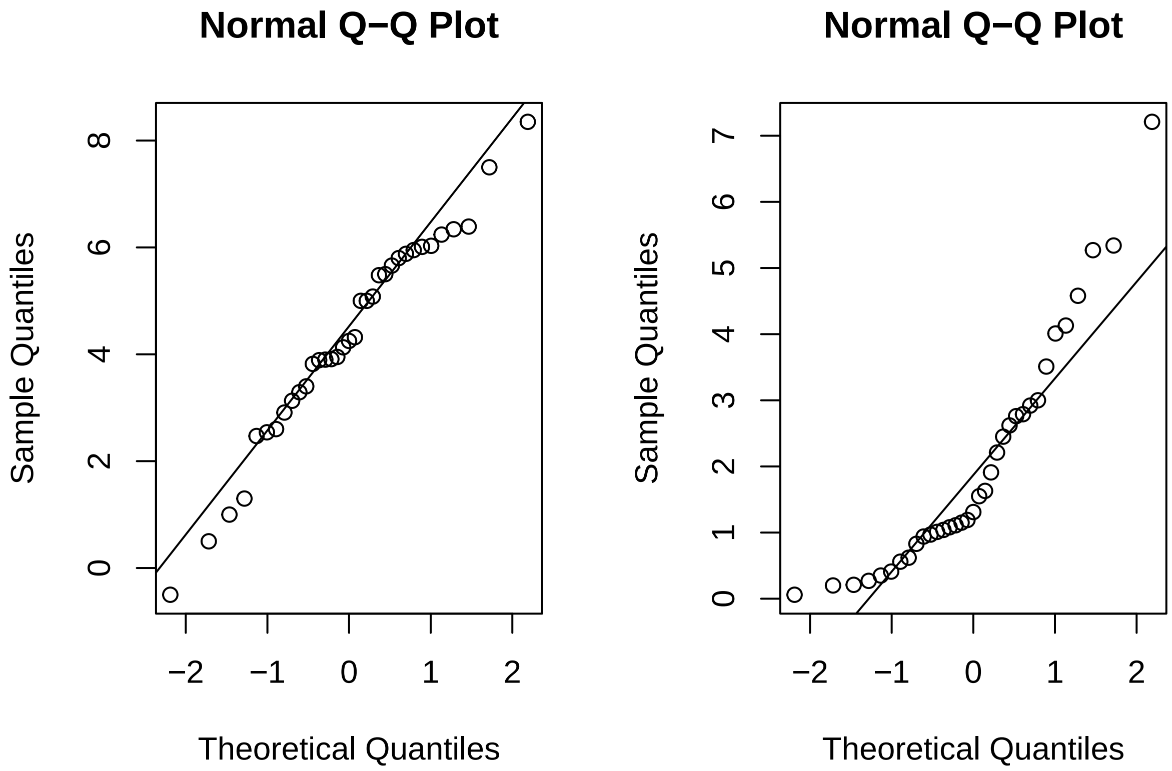
```r
par(mfrow = c(1, 1))
```

Quantile-Quantile Plot

```r
# manually
emp_vals <- sort(norm_df$normal_example)
theor_quants <- qnorm((seq_along(emp_vals) - .5) / length(emp_vals))
plot(theor_quants, emp_vals)
```



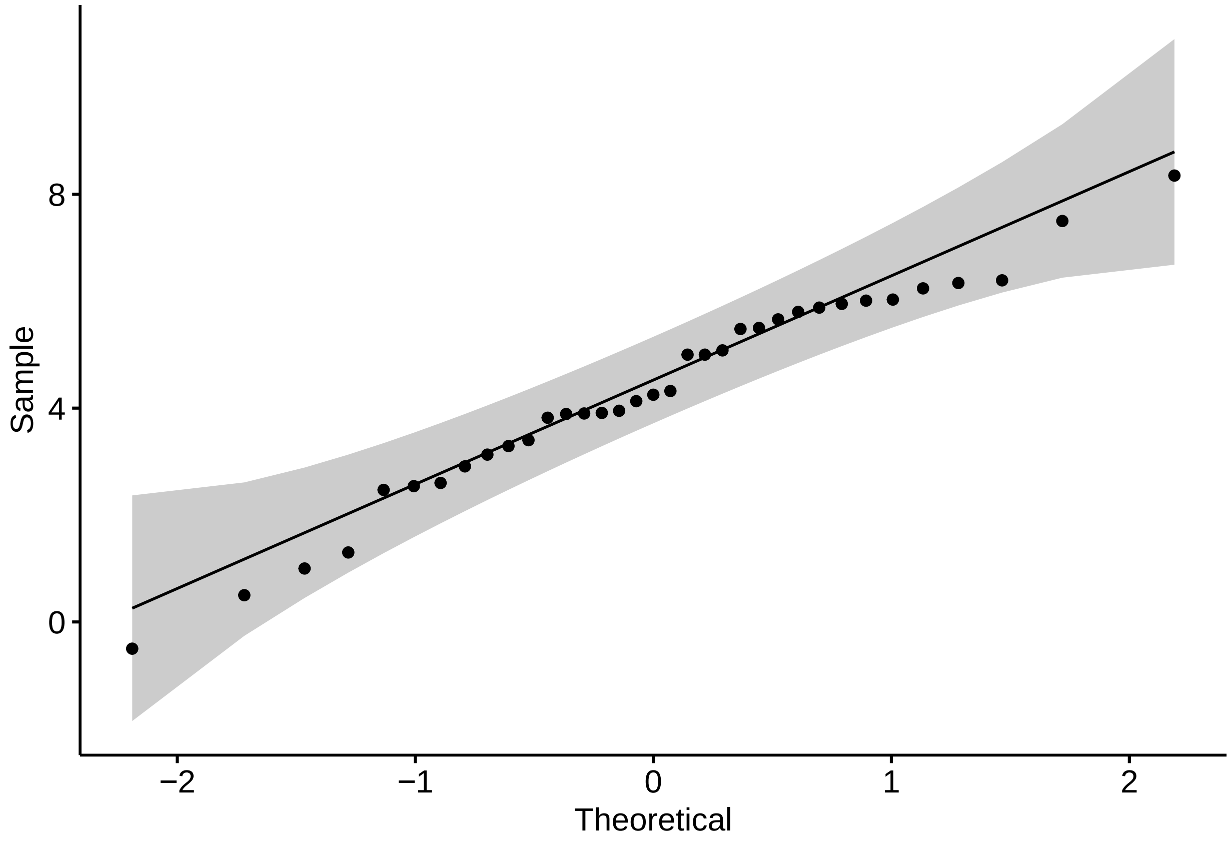
```
# using base R function 'qqnorm'
par(mfrow = c(1, 2))
qqnorm(norm_df$normal_example)
qqline(norm_df$normal_example)

qqnorm(norm_df$skewed_example)
qqline(norm_df$skewed_example)
```



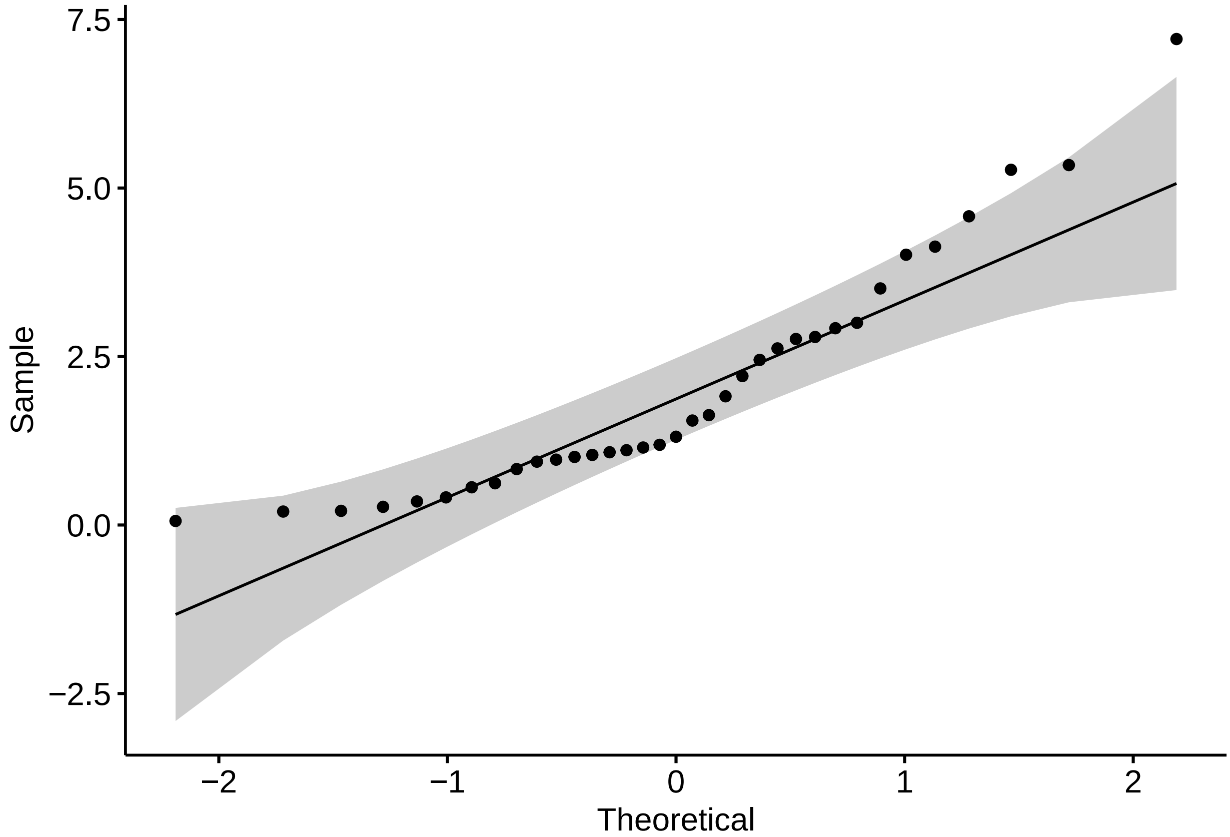
```r
par(mfrow = c(1, 1))

# using 'ggqqplot' from 'ggpubr'
# install.packages("ggpubr")
library(ggpubr)
```

```
## Loading required package: ggplot2
```

```r
ggqqplot(norm_df, "normal_example")
```



```
ggqqplot(norm_df, "skewed_example")
```



Shapiro–Wilk Test of Normality

$H_0:$ *the population is normally distributed*

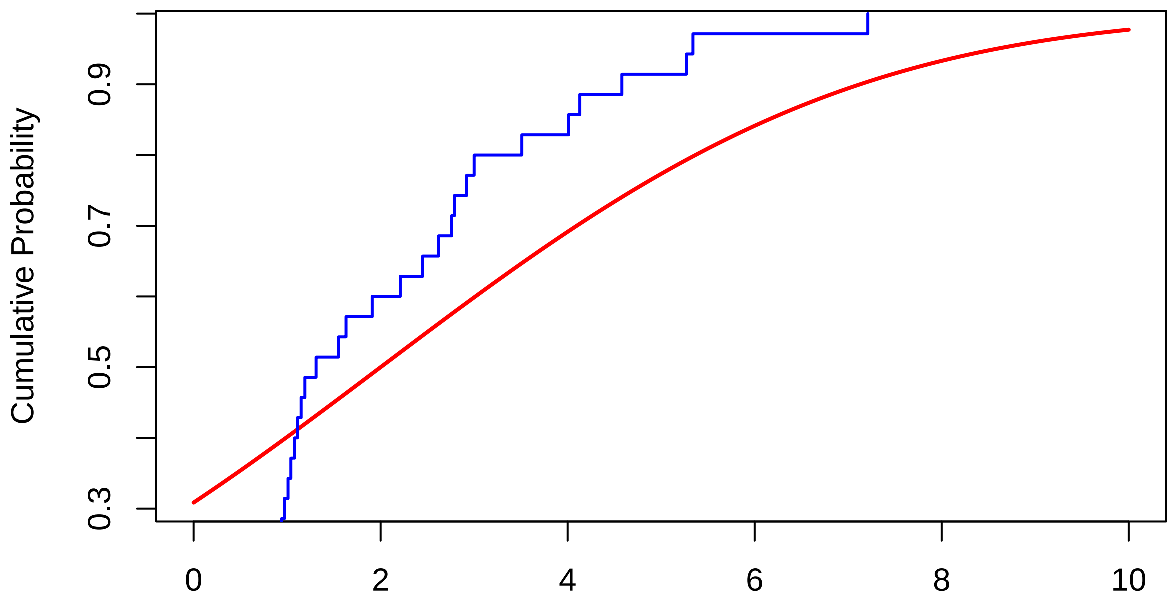
```
shapiro.test(norm_df$normal_example)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  norm_df$normal_example
## W = 0.975, p-value = 0.58
```

```
shapiro.test(norm_df$skewed_example)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  norm_df$skewed_example
## W = 0.885, p-value = 0.0016
```

Kolmogorov-Smirnov Tests



```
# does x come from the selected distribution with the specified parameters?
ks.test(norm_df$normal_example, pnorm, mean = mean(norm_df$normal_example), sd = sd(norm_df$normal_examp
```

```
## Warning in ks.test(norm_df$normal_example, pnorm, mean =
## mean(norm_df$normal_example), : ties should not be present for the Kolmogorov-
## Smirnov test

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  norm_df$normal_example
## D = 0.0948, p-value = 0.91
## alternative hypothesis: two-sided
```

```
# Do x and y come from the same distribution?
ks.test(norm_df$normal_example, norm_df$skewed_example)
```

```
## Warning in ks.test(norm_df$normal_example, norm_df$skewed_example): cannot
## compute exact p-value with ties

##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  norm_df$normal_example and norm_df$skewed_example
## D = 0.571, p-value = 2.2e-05
## alternative hypothesis: two-sided
```

Non-parametric Tests

Wilcoxon Rank Sum Test

We'll work on the `ToothGrowth` dataset for this exercise. The response is the length of odontoblasts (cells responsible for tooth growth) in 60 guinea pigs. Each animal received vitamin C by one of two delivery methods, orange juice (OJ) or ascorbic acid (VC). We'll compare the mean lengths between VC and OJ.

```
?ToothGrowth
head(ToothGrowth, 3)
```

```
##    len supp dose
## 1  4.2   VC  0.5
## 2 11.5   VC  0.5
## 3  7.3   VC  0.5
```

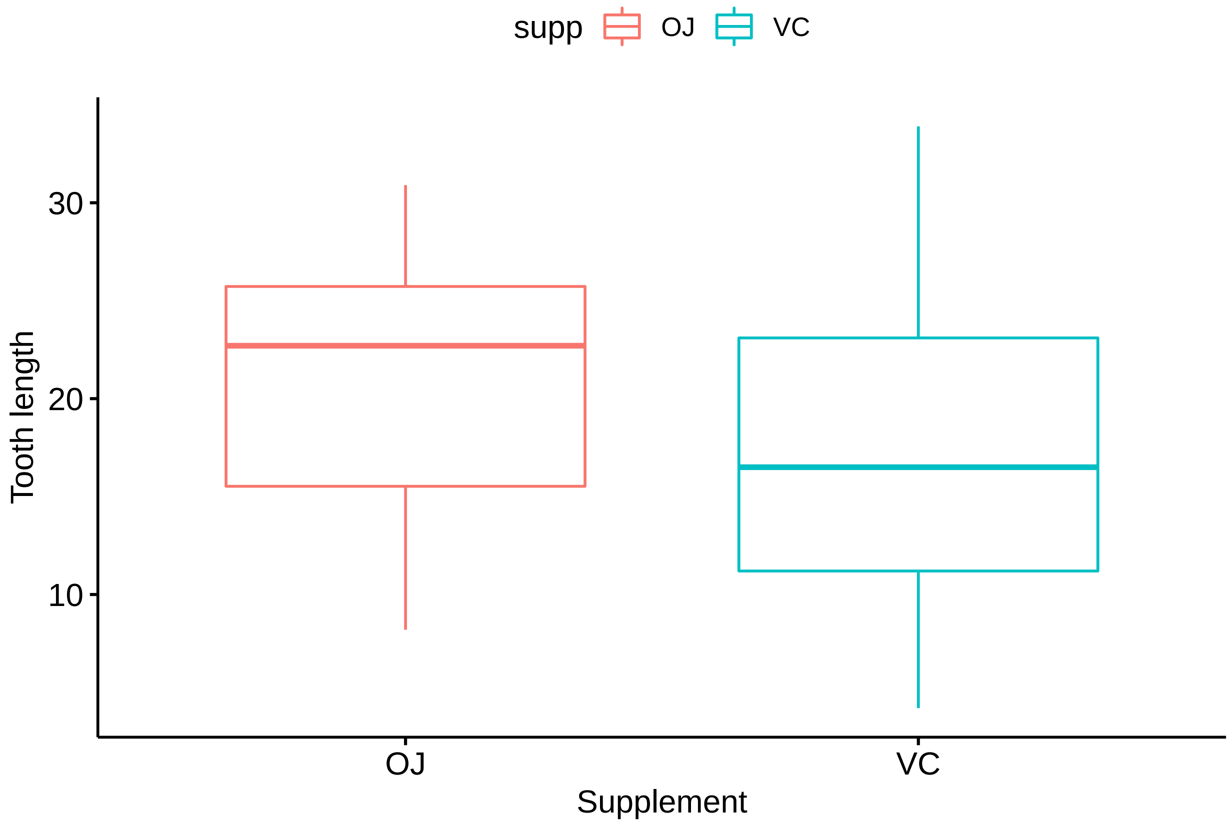
```
# Hypothesis: mean tooth lengths of VC and OJ are different
summary(ToothGrowth$len[ToothGrowth$supp == "VC"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     4.2    11.2    16.5    17.0    23.1    33.9
```

```
summary(ToothGrowth$len[ToothGrowth$supp == "OJ"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     8.2    15.5    22.7    20.7    25.7    30.9
```

```
g <- ggboxplot(data = ToothGrowth,
               x = "supp", y = "len",
               color = "supp",
               xlab = "Supplement", ylab = "Tooth length")
g
```



```
# Normal distribution?
shapiro.test(ToothGrowth$len[ToothGrowth$supp == "VC"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  ToothGrowth$len[ToothGrowth$supp == "VC"]
## W = 0.966, p-value = 0.43
```

```
shapiro.test(ToothGrowth$len[ToothGrowth$supp == "OJ"])
```

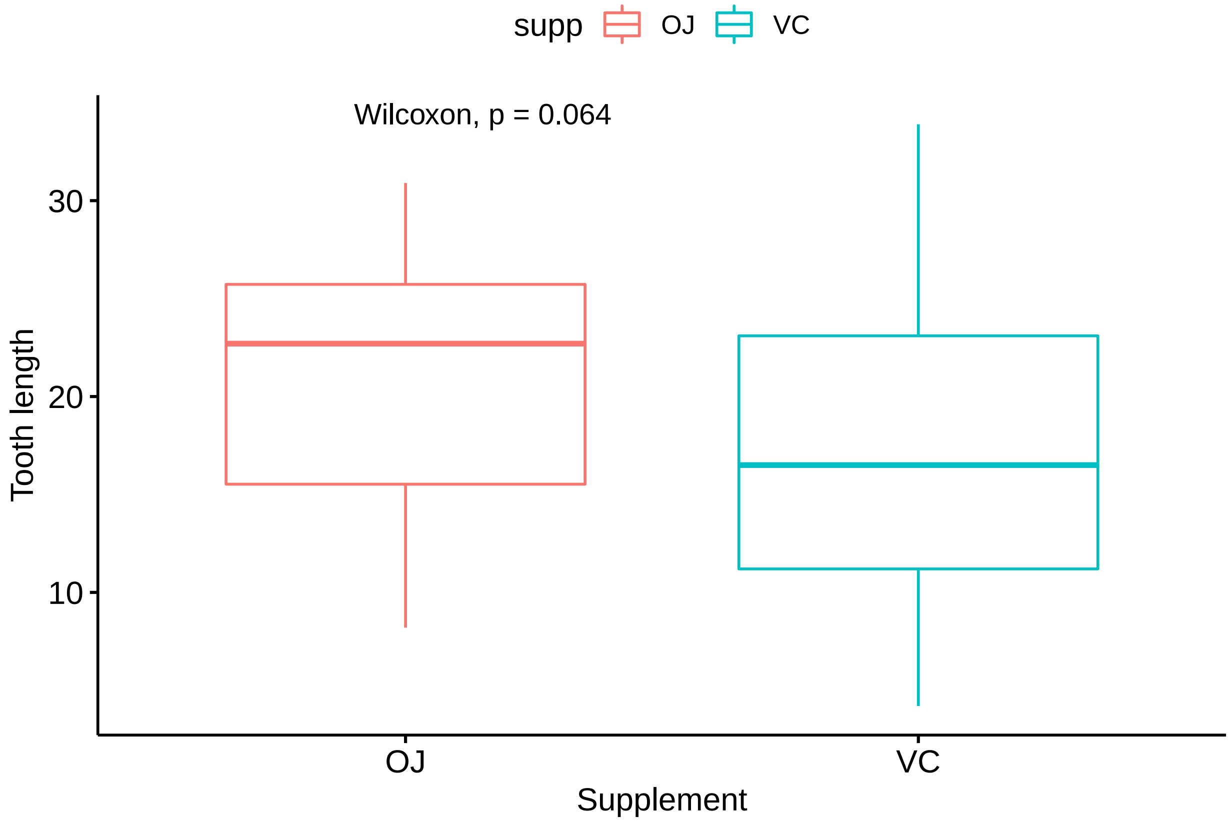
```
##
##  Shapiro-Wilk normality test
##
## data:  ToothGrowth$len[ToothGrowth$supp == "OJ"]
## W = 0.918, p-value = 0.024
```

```
# Wilcox test (Mann-Whitney U test)
wilcox.test(len~supp, data = ToothGrowth)
```

```
## Warning in wilcox.test.default(x = c(15.2, 21.5, 17.6, 9.7, 14.5, 10, 8.2, :
## cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  len by supp
## W = 576, p-value = 0.064
## alternative hypothesis: true location shift is not equal to 0
```

```
g + stat_compare_means(method = "wilcox")
```



Kruskal-Wallis Rank Sum Test

We'll use the diet dataset which contains information on 78 people using one of three diets.

```
diet_df <- read.csv("../data/Diet_R.csv")

head(diet_df)
```

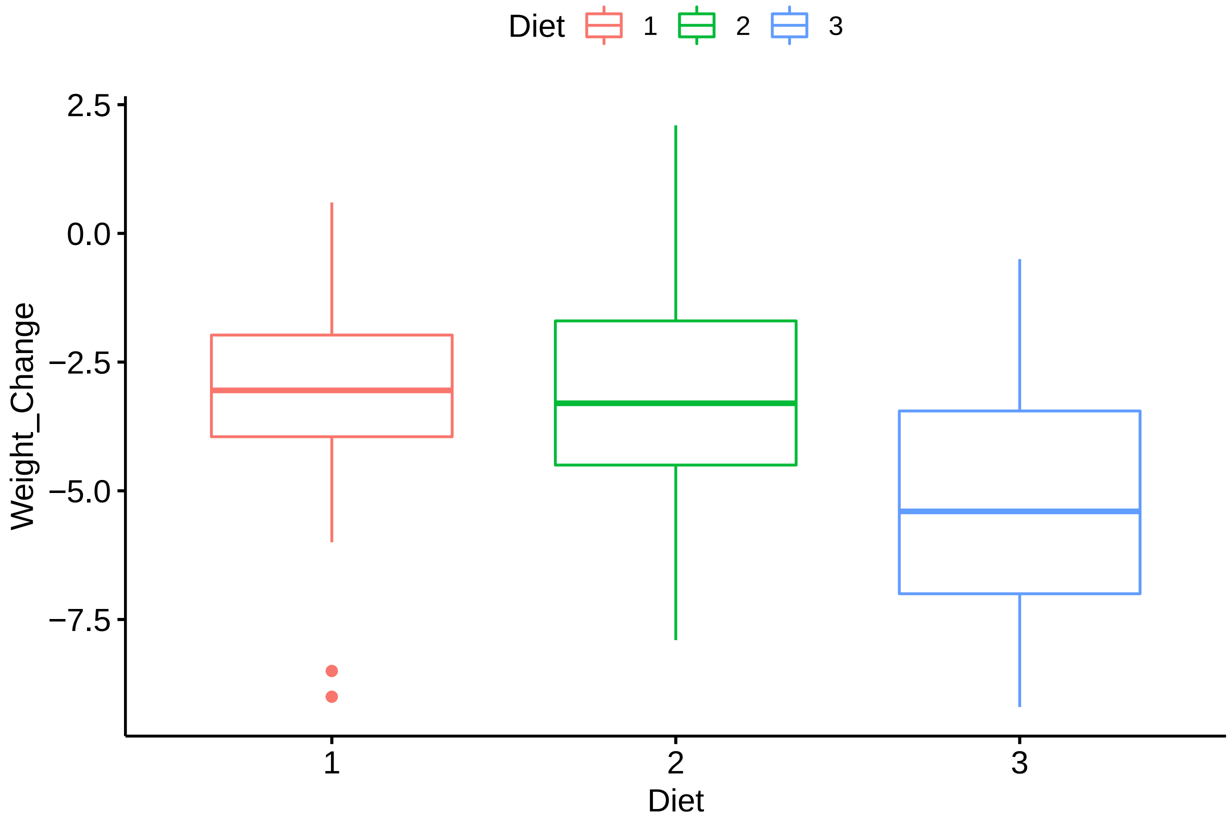
```
##   Person gender Age Height pre.weight Diet weight6weeks
## 1     25     NA  41    171         60    2         60.0
## 2     26     NA  32    174        103    2        103.0
## 3      1      0  22    159         58    1         54.2
## 4      2      0  46    192         60    1         54.0
## 5      3      0  55    170         64    1         63.3
## 6      4      0  33    171         64    1         61.1
```

```
# turn categorical variables into factor
diet_df$Diet <- as.factor(diet_df$Diet)
diet_df$gender <- as.factor(diet_df$gender)

# create new variable
diet_df$Weight_Change <- diet_df$weight6weeks - diet_df$pre.weight
```

```
summary(diet_df)
```

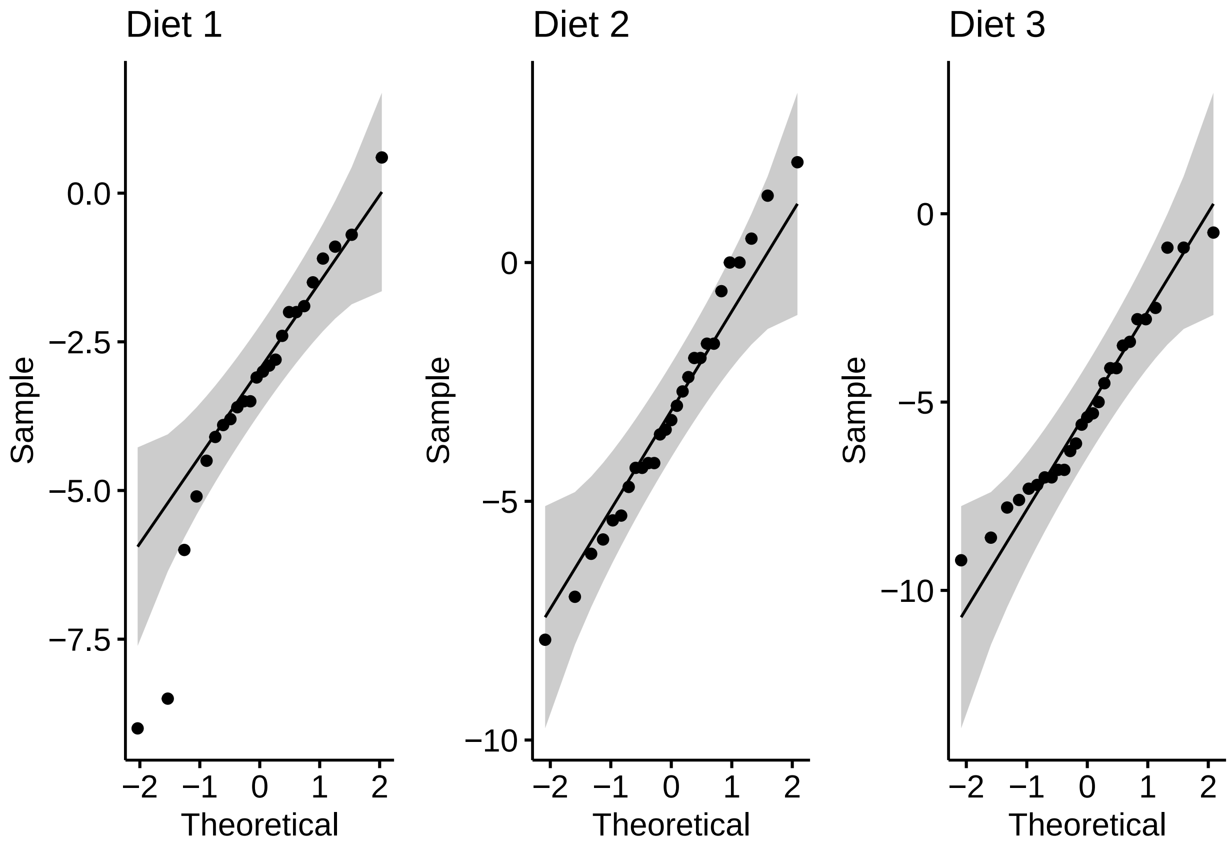
```
##      Person        gender        Age            Height       pre.weight       Diet
##  Min.   : 1.0   0    :43   Min.   :16.0   Min.   :141   Min.   : 58.0   1:24
##  1st Qu.:20.2   1    :33   1st Qu.:32.2   1st Qu.:164   1st Qu.: 66.0   2:27
##  Median :39.5   NA's: 2   Median :39.0   Median :170   Median : 72.0   3:27
##  Mean   :39.5              Mean   :39.2   Mean   :171   Mean   : 72.5
##  3rd Qu.:58.8              3rd Qu.:46.8   3rd Qu.:175   3rd Qu.: 78.0
##  Max.   :78.0              Max.   :60.0   Max.   :201   Max.   :103.0
##   weight6weeks    Weight_Change
##  Min.   : 53.0   Min.   :-9.20
##  1st Qu.: 61.9   1st Qu.:-5.55
##  Median : 69.0   Median :-3.60
##  Mean   : 68.7   Mean   :-3.84
##  3rd Qu.: 73.8   3rd Qu.:-2.00
##  Max.   :103.0   Max.   : 2.10
```

```
g <- ggboxplot(diet_df, x = "Diet", y = "Weight_Change", color = "Diet")
g
```



We'll compare weight changes of the three diet groups. Let's check the normality of weight changes of the three groups:

```
ggarrange(ggqqplot(diet_df[diet_df$Diet == 1, ], "Weight_Change", title = "Diet 1"),
          ggqqplot(diet_df[diet_df$Diet == 2, ], "Weight_Change", title = "Diet 2"),
          ggqqplot(diet_df[diet_df$Diet == 3, ], "Weight_Change", title = "Diet 3"), ncol = 3)
```

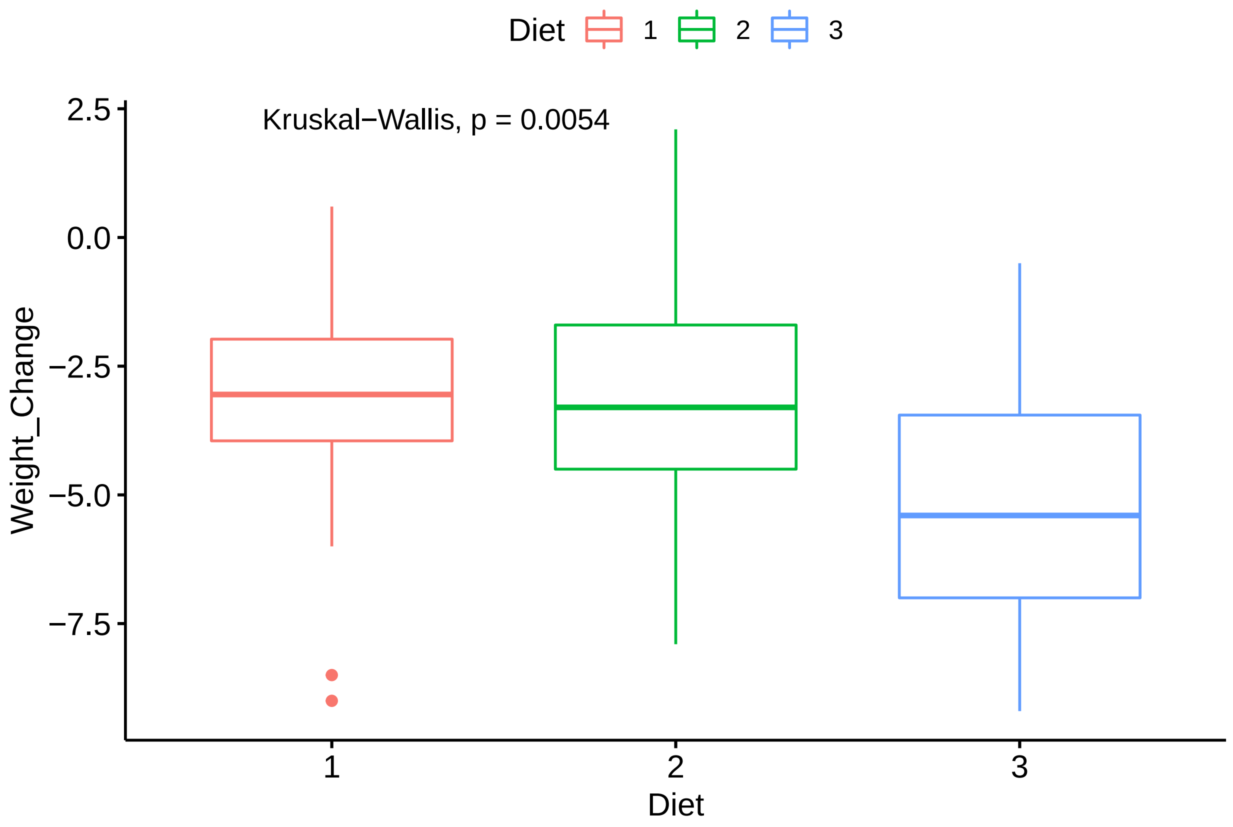


The normality assumption of ANOVA is not met, we'll use the Kruskal-Wallis test instead:

```
kruskal.test(Weight_Change~Diet, data = diet_df)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Weight_Change by Diet
## Kruskal-Wallis chi-squared = 10.4, df = 2, p-value = 0.0054
```

```
g + stat_compare_means()
```



```
pairwise.wilcox.test(diet_df$Weight_Change, diet_df$Diet)
```

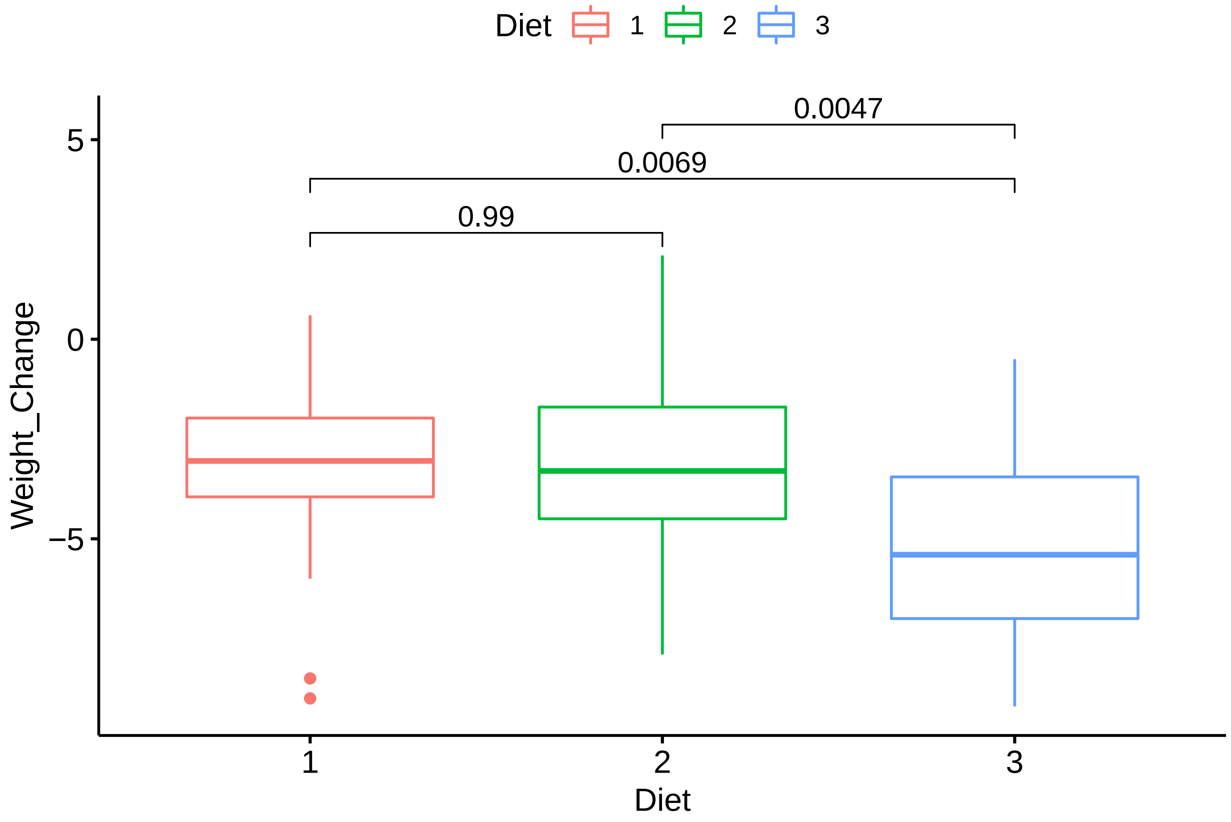
```
## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot compute
## exact p-value with ties

## 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 with continuity correction
##
## data:  diet_df$Weight_Change and diet_df$Diet
##
##   1    2
## 2 0.99 -
## 3 0.01 0.01
##
## P value adjustment method: holm
```

```
g + stat_compare_means(comparisons = list(c(1, 2), c(1, 3), c(2, 3)), method = "wilcox")
```

```
## Warning in wilcox.test.default(c(-3.8, -6, -0.700000000000003, -2.9, -2.8, :
## cannot compute exact p-value with ties

## Warning in wilcox.test.default(c(-3.8, -6, -0.700000000000003, -2.9, -2.8, :
```

```
## cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(c(0, 0, 2.1, -2, -1.7, -4.3, -7,
## -0.600000000000001, : cannot compute exact p-value with ties
```



Multiple Testing Correction

```
# install.packages("multtest")
data(golub, package="multtest")

dim(golub)
```

```
## [1] 3051   38
```

```
row.names(golub) <- paste("gene_", 1:nrow(golub), sep="")
golub[1:3,1:4]
```

```
##             [,1]     [,2]     [,3]     [,4]
## gene_1 -1.45769 -1.39420 -1.42779 -1.40715
## gene_2 -0.75161 -1.26278 -0.09052 -0.99596
## gene_3  0.45695 -0.09654  0.90325 -0.07194
```

```
# we know that groups are 1:27 vs. 28:38
p_plain <- apply(golub, 1, function(x) t.test(x[1:27], x[28:38])$p.value)

### implementations may be slightly different than in the lecture
?p.adjust
```

```r
p_bonf <- p.adjust(p_plain, method = "bonferroni")
p_holm <- p.adjust(p_plain, method = "holm")
p_fdr <- p.adjust(p_plain, method = "fdr")

head(sort(p_bonf), 10)
```

```
##   gene_2124    gene_896   gene_2600    gene_766    gene_829   gene_2851    gene_703 
## 8.4847e-09 4.6888e-06 2.5701e-05 4.8125e-05 6.9542e-05 1.2413e-04 1.7291e-04 
##   gene_2386   gene_2645   gene_2002 
## 1.8074e-04 2.0762e-04 2.2060e-04
```

```r
head(sort(p_holm), 10)
```

```
##   gene_2124    gene_896   gene_2600    gene_766    gene_829   gene_2851    gene_703 
## 8.4847e-09 4.6873e-06 2.5684e-05 4.8078e-05 6.9451e-05 1.2393e-04 1.7257e-04 
##   gene_2386   gene_2645   gene_2002 
## 1.8032e-04 2.0707e-04 2.1995e-04
```

```r
head(sort(p_fdr), 10)
```

```
##   gene_2124    gene_896   gene_2600    gene_766    gene_829   gene_2851    gene_703 
## 8.4847e-09 2.3444e-06 8.5669e-06 1.2031e-05 1.3908e-05 2.0689e-05 2.2060e-05 
##   gene_2002   gene_2386   gene_2645 
## 2.2060e-05 2.2060e-05 2.2060e-05
```

```r
sum(p_bonf <= 0.05)
```

```
## [1] 103
```

```r
sum(p_holm <= 0.05)
```

```
## [1] 103
```

```r
sum(p_fdr <= 0.05) # less conservative
```

```
## [1] 695
```