

Checking ANOVA Assumptions in R

NE Milla, Jr.

2025-09-11

Loading required packages

```
library(readxl)
library(tidyverse)
library(ExpDes)
library(agricolae)
library(afex)
library(performance)
library(nortest)
library(ggpubr)
```

Importing data

```
crop_yield <- read_excel("chap4demo.xlsx")
head(crop_yield)
```

```
## # A tibble: 6 x 3
##   trt   fertilizer yield
##   <chr>         <dbl> <dbl>
## 1 T1             0  4.89
## 2 T1             0  4.79
## 3 T1             0  4.65
## 4 T1             0  4.47
## 5 T2            50  5.08
## 6 T2            50  5.19
```

Generating the ANOVA

```
crop_yield <- crop_yield |>
  mutate(FERT = as.factor(fertilizer))
crop_aov <- aov(yield ~ FERT,
               data = crop_yield)
anova(crop_aov)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value    Pr(>F)
## FERT         5 1.3555  0.271107   19.567 1.04e-06 ***
## Residuals    18 0.2494  0.013856
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

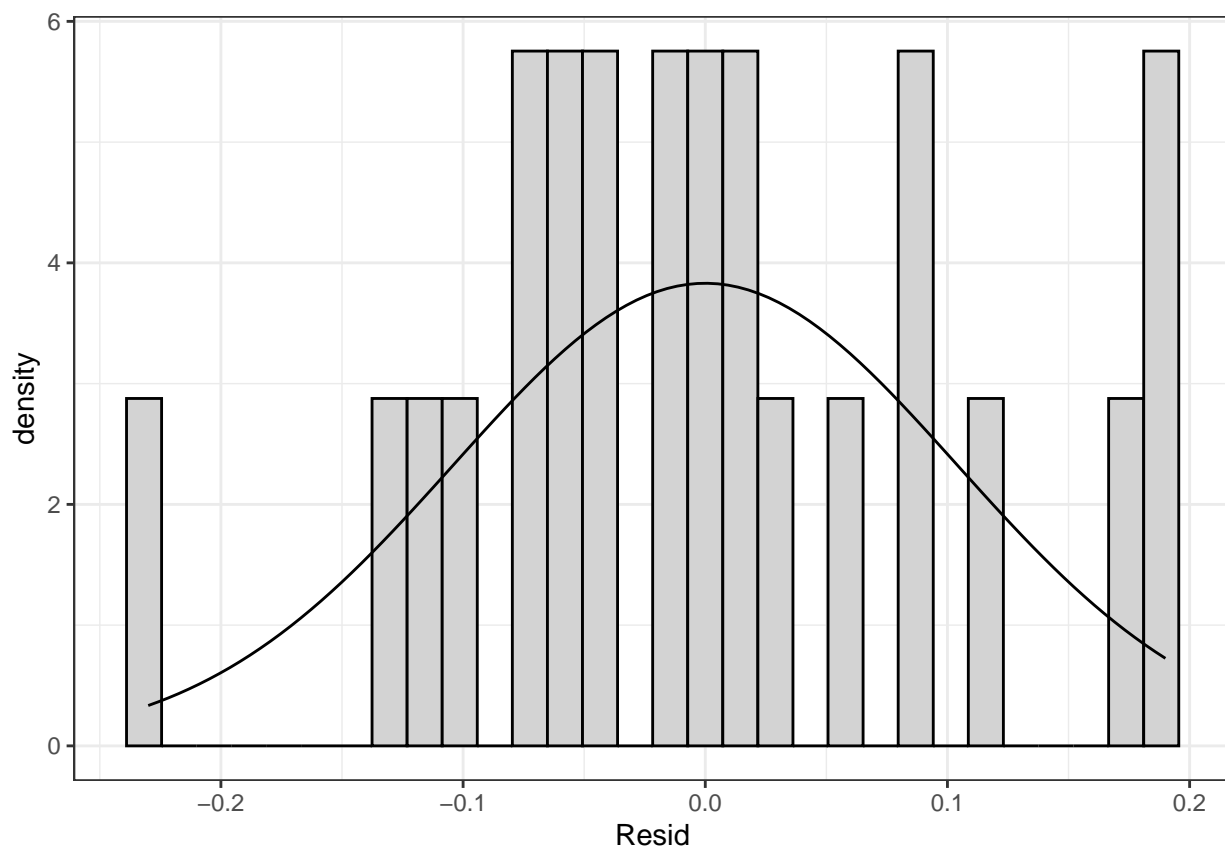
We generate the residuals and add to the data frame.

```
crop_yield <- crop_yield |>
  mutate(Resid = crop_aov$residuals)
```

Checking normality

First, we generate the relative frequency histogram of the residuals to visualize the distribution of the residuals. We add on top of this plot the theoretical normal density curve to determine if the empirical density looks similar to the theoretical density.

```
ggplot(crop_yield, aes(Resid)) +
  geom_histogram(aes(y = ..density..), fill='lightgray', col='black') +
  stat_function(fun = dnorm, args = list(mean=mean(crop_yield$Resid), sd=sd(crop_yield$Resid))) + theme_l
```

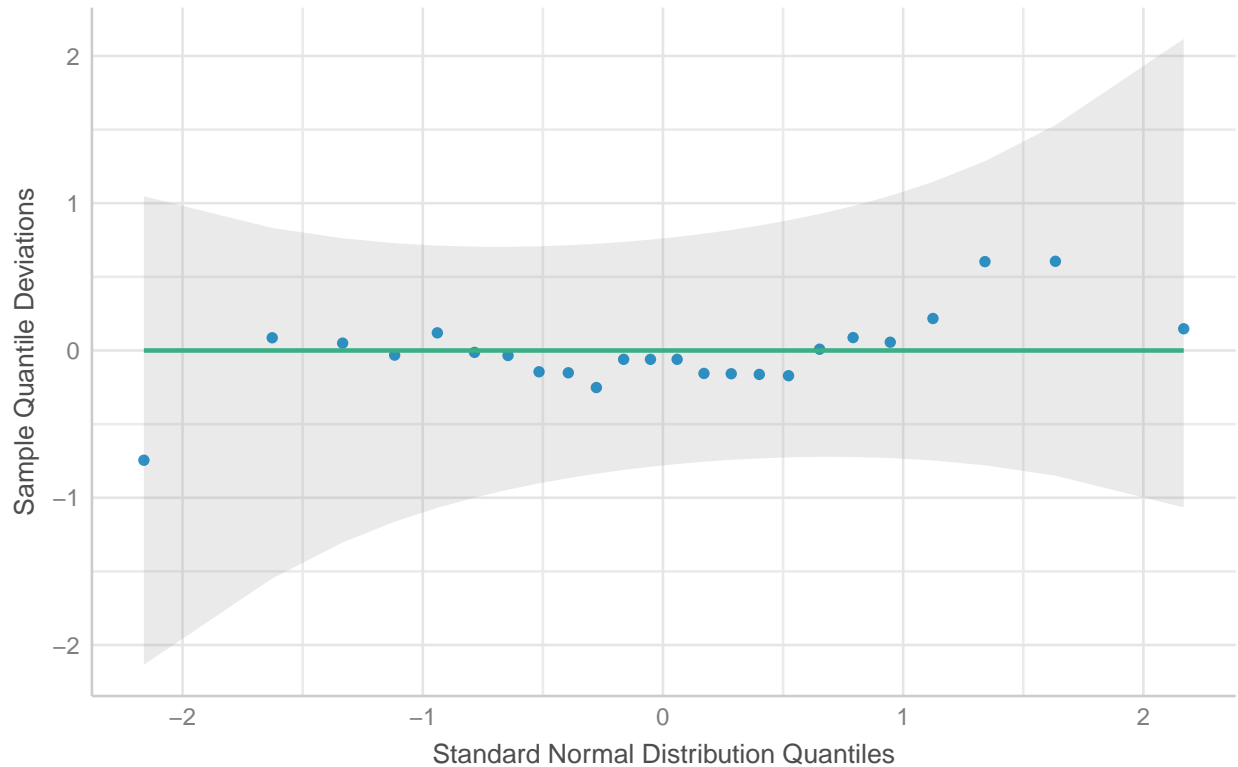


An alternative function for generating a QQ plot is given below.

```
plot(check_normality(crop_aov))
```

Normality of Residuals

Dots should fall along the line



We use the Shapiro-Wilk test to formally test if the residuals are normally distributed or not.

```
with(crop_yield, shapiro.test(Resid))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Resid
## W = 0.96856, p-value = 0.6316
```

Alternatively, we can use the `check_normality()` function in the **performance** package

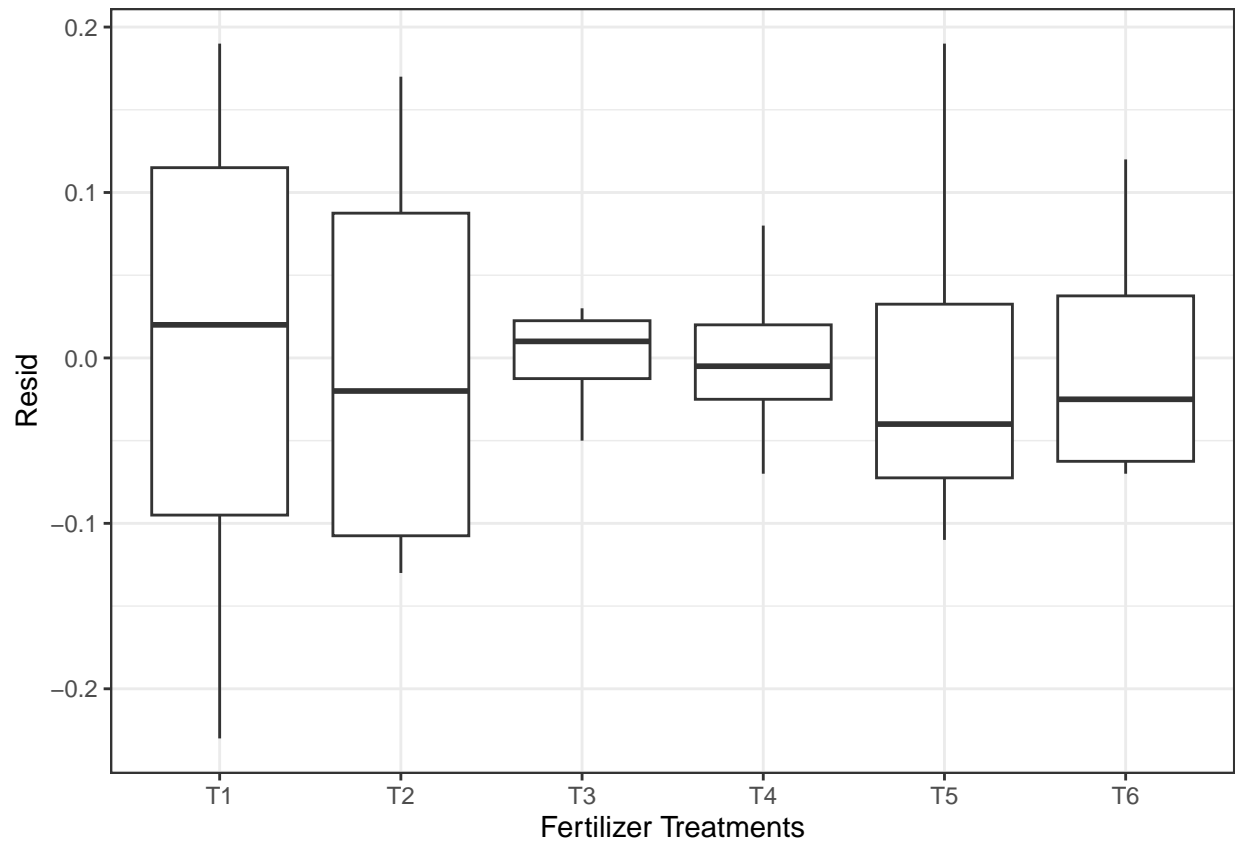
```
check_normality(crop_aov)
```

```
## OK: residuals appear as normally distributed (p = 0.632).
```

Checking homogeneity of variance assumption

```
ggplot(crop_yield, aes(x = trt, y = Resid)) +
  geom_boxplot() +
```

```
labs(x = "Fertilizer Treatments") +  
theme_bw()
```



```
# Bartlett's test  
bartlett.test(Resid ~ FERT,  
              data = crop_yield)
```

```
##  
## Bartlett test of homogeneity of variances  
##  
## data: Resid by FERT  
## Bartlett's K-squared = 7.4869, df = 5, p-value = 0.1869
```

```
#Levene's test  
car::leveneTest(Resid ~ FERT,  
               data = crop_yield,  
               center = mean)
```

```
## Levene's Test for Homogeneity of Variance (center = mean)  
##      Df F value Pr(>F)  
## group 5  2.5367 0.06613 .  
##      18  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Fligner's test
fligner.test(Resid ~ FERT,
             data = crop_yield)
```

```
##
## Fligner-Killeen test of homogeneity of variances
##
## data:  Resid by FERT
## Fligner-Killeen:med chi-squared = 9.6845, df = 5, p-value = 0.08468
```

- If the assumption of homogeneous variances is violated, one can apply data transformation on the response, run ANOVA, and test again the new residuals
- Alternatively, one can run the `oneway.test()` function in the **stats** package. This is only applicable for one way classification data such as that from CRD experiments. We call this as **Welch's ANOVA**.

```
oneway.test(yield ~ FERT,
            var.equal = FALSE,
            data = crop_yield)
```

```
##
## One-way analysis of means (not assuming equal variances)
##
## data:  yield and FERT
## F = 10.394, num df = 5.0000, denom df = 7.9358, p-value = 0.002484
```

Testing for the assumption of independence

```
crop_yield$resid_lag1 <- lag(crop_yield$Resid)
head(crop_yield)
```

```
## # A tibble: 6 x 6
##   trt   fertilizer yield FERT   Resid resid_lag1
##   <chr>         <dbl> <dbl> <fct>   <dbl>      <dbl>
## 1 T1             0  4.89  0      0.190        NA
## 2 T1             0  4.79  0      0.0900     0.190
## 3 T1             0  4.65  0     -0.0500    0.0900
## 4 T1             0  4.47  0     -0.23      -0.0500
## 5 T2            50  5.08  50     0.0600    -0.23
## 6 T2            50  5.19  50     0.170     0.0600
```

```
cor.test(crop_yield$Resid, crop_yield$resid_lag1)
```

```
##
## Pearson's product-moment correlation
##
## data:  crop_yield$Resid and crop_yield$resid_lag1
## t = -0.51752, df = 21, p-value = 0.6102
## alternative hypothesis: true correlation is not equal to 0
```

```
## 95 percent confidence interval:
## -0.501236 0.314532
## sample estimates:
##      cor
## -0.1122195
```

```
car::durbinWatsonTest(crop_aov)
```

```
## lag Autocorrelation D-W Statistic p-value
## 1 -0.1046512 2.044908 0.302
## Alternative hypothesis: rho != 0
```

- Therefore the assumption is valid!

Generate the table of means

```
crop_yield.ur <- read_excel("chap4demouequalreps.xlsx")
head(crop_yield.ur)
```

```
## # A tibble: 6 x 3
##   trt   fertilizer yield
##   <chr>      <dbl> <dbl>
## 1 T1         0  4.89
## 2 T1         0  4.79
## 3 T1         0  4.65
## 4 T2        50  5.08
## 5 T2        50  5.19
## 6 T2        50  4.89
```

```
crop_yield.ur |>
  dplyr::group_by(trt) |>
  dplyr::summarize(MEAN = round(mean(yield),2),
                  SD = round(sd(yield),4),
                  n = length(yield))
```

```
## # A tibble: 6 x 4
##   trt   MEAN   SD    n
##   <chr> <dbl> <dbl> <int>
## 1 T1    4.78 0.121    3
## 2 T2    5.02 0.141    4
## 3 T3    5.22 0.0361   3
## 4 T4    5.38 0.0616   4
## 5 T5    5.36 0.132    4
## 6 T6    5.28 0.107    3
```

```
crop.aov1 <- with(crop_yield.ur, aov(yield ~ trt))
anova(crop.aov1)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value    Pr(>F)
## trt         5  0.91256  0.182512   15.421 1.89e-05 ***
## Residuals   15  0.17753  0.011836
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```