ANOVA course (Session 1)

S.Civit

21 de enero de 2020

1 Objetivos:

-**Objetivo 1: ANOVA 1-way**. Emplearemos en lo posible **tidyverse** y **dplyr** para realizar nuestros análisis de datos.

2 What is ANOVA?

La técnica de técnicas denominada Análisis de la varianza (ANOVA), del acrónimo Analysis of variance: ANalysis Of VAriance, tiene como objetivo básico la comparación de las medias de más de dos poblaciones.

En el ANOVA se comparan siempre las medias de varias poblaciones y se hace a través de un contraste de hipótesis donde se analiza la varianza, es cierto; pero no sólo eso, porque también se analizan las diferencias de medias que hay entre las muestras, y también, por supuesto, como siempre en Estadística, se analiza el tamaño de muestra.

3 Vocabulario básico:

3.1 Definir objetivo del experimento a analizar

3.2 Variable respuesta: Cuantitativa continua

3.3 Factor.

Un factor en ANOVA es una **variable cualitativa que genera o que contempla una serie de poblaciones a comparar**. Por ejemplo, se ensayan tres tipos de fertilizantes en unos campos de cultivo para evaluar la productividad, se ensayan cuatro medicamentos distintos para ver si aumentan los niveles de hemoglobina en pacientes con anemia. En estos casos tenemos, en primer lugar el factor tipo de fertilizante. En el segundo, el factor fármaco.

3.4 Niveles del factor

Los niveles de un factor son los **grupos o poblaciones que genera** un factor. En el primer ejemplo anterior tenemos tres niveles. En el segundo tenemos cuatro niveles.

3.5 Tipo de factor

Un factor es fijo si los niveles que tenemos de él en el estudio son realmente todos los que nos interesa comparar. Un factor es aleatorio si los niveles que tenemos en nuestro estudio es una muestra de niveles tomados de una población de niveles que son los que, en realidad, queremos comparar. Los dos ejemplos anteriores si los tres fertilizantes o los cuatro fármacos son nuestro objeto de comparación, estamos ante factores fijos. Pero, observemos lo siguiente: si en otro ejemplo, estoy comparando si hay diferencias en la calidad de un producto fabricado por 100 operarios trabajando en una industria y, para hacerlo, elijo al azar a 5 de esos 100 operarios y analizo 3 productos elaborados por cada uno de ellos, pero lo que me interesa es ver si hay diferencias entre los 100, no entre esos 5, estoy ante el factor operario con 5 niveles, pero ese factor es, ahora, no fijo, sino aleatorio.

3.6 Réplicas (balanceado no balanceado)

Número de experimentos realizados por condición experimental. Si todos los grupos (niveles del factor) tienen el mismo número de réplicas se denomina **diseño balanceado**, de gran trascendecia en el uso de funciones de R.

3.7 Comparaciones múltiples/Componentes de la varianza

Si tenemos un factor fijo y detectamos que hay diferencias entre esas poblaciones, nos interesará decir cuáles son esas diferencias concretas. Las comparaciones múltiples hacen esa labor, comparan, dos a dos, de una forma muy especial, todas las poblaciones para dibujar un mapa de las diferencias. Si tenemos un factor aleatorio, el planteamiento es ahora muy diferente: debemos pasar de la muestra de muestras de poblaciones que tenemos a una población de poblaciones y eso lo haremos estimando la varianza, la dispersión que debe haber dentro de esa población de poblaciones.

3.8 Factores cruzados/Factores anidados

Cuando hay más de un factor en un estudio, los factores, dos a dos, pueden estar cruzados o anidados. Tenemos factores cruzados cuando todos los niveles de un factor están combinados con todos los niveles del otro factor. Tenemos factores anidados cuando los niveles de un factor están jerarquizados entre los niveles del otro factor.

3.9 Interacción entre factores

Cuando los factores están cruzados podemos estudiar algo muy importante en ANOVA: la interacción entre esos factores. Hay interacción cuando la respuesta, el efecto conseguido con la presencia de un nivel de un factor, depende de con qué nivel del otro factor esté combinado.

3.10 Randomización

Este es un concepto clave “en el diseño de cualquier experimento des de la vertiente estadística y por tanto elemento clave en el Diseño experimental que no es otro que la”asignación aleatoria" de los unidades experimentales a los grupos (niveles del factor). Se denomina **completely randomized design (CRD)**.

Si queremos **randomizar** factor tratamiento a 4 niveles (treatment factor, 4 levels) A,B,C and D en un total de 20 unidades experimentales (a balanced design with 5 replicates), obtendriamos con el siguiente codigo como “realizar el experimento”

This means that the first experimental unit will get treatment xxx, the second xxx and so on.

```
treat.ord <- rep(c("A", "B", "C", "D"), each = 5) ## could also use LETTERS[1:4]
treat.ord
```

```
##  [1] "A" "A" "A" "A" "A" "B" "B" "B" "B" "B" "C" "C" "C" "C" "C" "D" "D" "D" "
D"
## [20] "D"
```

```
sample(treat.ord) ## random permutation
```

```
##  [1] "D" "D" "A" "D" "C" "B" "C" "B" "B" "D" "A" "D" "A" "C" "C" "A" "B" "A" "
B"
## [20] "C"
```

3.11 Ejemplo “conceptual”

Se toma una muestra de 30 alumnos durante toda la ESO.

Se dividen en tres clases distintas (definimos tres “líneas distintas de aprendizaje”). Cada una va a seguir, durante los cuatro años (duración ESO), un plan distinto de enseñanza del inglés.

Se sabe el nivel escrito y el nivel oral de esos alumnos al final de la primaria. Se han diferenciado dos niveles dentro de cada grupo, según el promedio de notas globales de esos alumnos ha sido alto o bajo, en el global de las materias.

Durante los cuatro cursos de la ESO se ha hecho un seguimiento, alumno por alumno, del nivel de inglés oral de esos alumnos.

```
-La variable estudiada es el nivel de inglés oral.
-Hay dos factores fijos: Grupo y Nivel.
-Grupo a tres niveles y Nivel a dos niveles.
-Los dos factores son fijos y están cruzados.
```

Fuera del scope del curso

```
-Hay tercer factor: el factor ESO, con cuatro niveles fijos.
-Grupo y Nivel son intersujetos.
-El factor ESO es intrasujetos.
Las variables InglésEscrito e InglésOral a finales de primaria podría tratarse com
o covariable.
```

4 Consideraciones sobre aov(), anova(), Anova(), lm(). Type I, II, III

anova function for analysis of variance uses Type I sum of squares by default and unfortunately for **unbalanced designs**, **order of terms will affect the p-values with Type I SS**.

Best option. Define your model with **lm** and then use the **Anova (look capital letter A from Anova) function in the car package**, with which you can specify Type II or Type III sum of squares.

But note that you should change the global options(contrasts = settings if you are going to use Type III

```
library(car)
data(mtcars)
mtcars$cyl = factor(mtcars$cyl)
mtcars$am = factor(mtcars$am)
model = lm(mpg ~ cyl*am, data=mtcars)

### Type I
anova(model)
### Type II
Anova(model)

### Type III

Anova(lm(mpg ~ cyl*am, data=mtcars, contrasts=list(topic=contr.sum, sys=contr.su
m)), type=3)
```

5 Fixed Effects Models: One-Way ANOVA

We also say that **Y is the response variable** and the grouping information is a **categorical predictor called factor**

We sometimes distinguish between unordered (or nominal) and ordered (or ordinal) factors. An example of an unordered factor would be eye color (e.g., with levels “brown”, “blue”, “green”) and an example of an ordered factor would be income class (e.g., with levels “low”, “middle”, “high”).

We start by formulating a parametric model for our data. Let Y_{ij} be the jth observation in treatment group i, where $i = 1, \ldots, g$ and $j = 1, \ldots, n_i$.

Model:

$$Y_{ij} = \mu + \alpha_i + e_{ij} \quad e_{ij} i.i.d \approx N(0, \sigma^2)$$

6 One-way ANOVA Fixed effects: Step by Step

InsectSprays dataset contains the counts of insects killed in agricultural experimental units treated with different insecticides (sprays).

```
library(tidyverse)
```

```
## -- Attaching packages ------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.4
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

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

```
data(InsectSprays)
summary(InsectSprays)
```

```
##      count       spray
##  Min.   : 0.00   A:12
##  1st Qu.: 3.00   B:12
##  Median : 7.00   C:12
##  Mean   : 9.50   D:12
##  3rd Qu.:14.25   E:12
##  Max.   :26.00   F:12
```

```
str(InsectSprays)
```

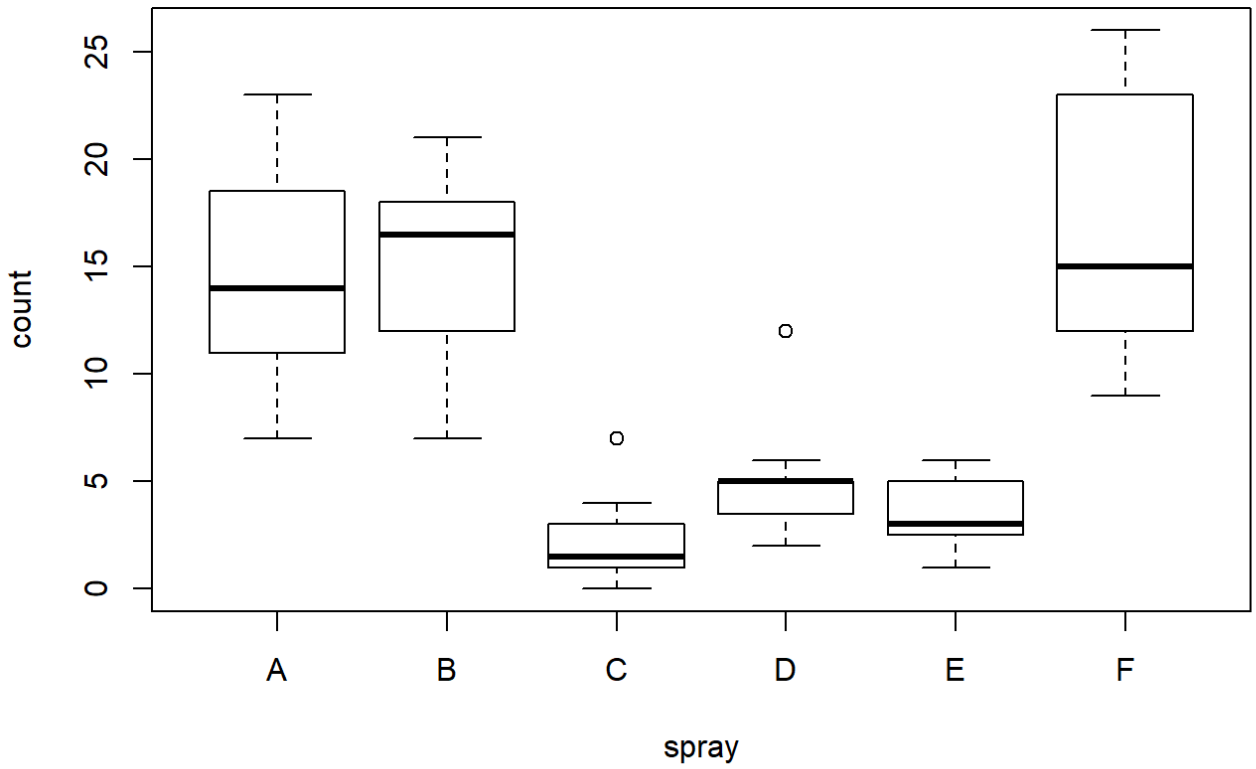
```
## 'data.frame':    72 obs. of  2 variables:
##  $ count: num  10 7 20 14 14 12 10 23 17 20 ...
##  $ spray: Factor w/ 6 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
# Descriptive
InsectSprays %>% group_by(spray) %>% summarise(media=mean(count), desvest=sd(coun
t), n=length(count))
```

spray <fctr>	media <dbl>	desvest <dbl>	n <int>
A	14.500000	4.719399	12
B	15.333333	4.271115	12
C	2.083333	1.975225	12
D	4.916667	2.503028	12
E	3.500000	1.732051	12
F	16.666667	6.213378	12

6 rows

```
boxplot(count ~ spray, data=InsectSprays)
```




```
aov.out = aov(count ~ spray, data=InsectSprays)
summary(aov.out)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## spray        5   2669   533.8    34.7 <2e-16 ***
## Residuals   66   1015    15.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lm.out = lm(count ~ spray, data=InsectSprays)
anova(lm.out)
```

	Df <int>	Sum Sq <dbl>	Mean Sq <dbl>	F value <dbl>	Pr(>F) <dbl>
spray	5	2668.833	533.76667	34.70228	3.182584e-17
Residuals	66	1015.167	15.38131	*NA*	*NA*

2 rows

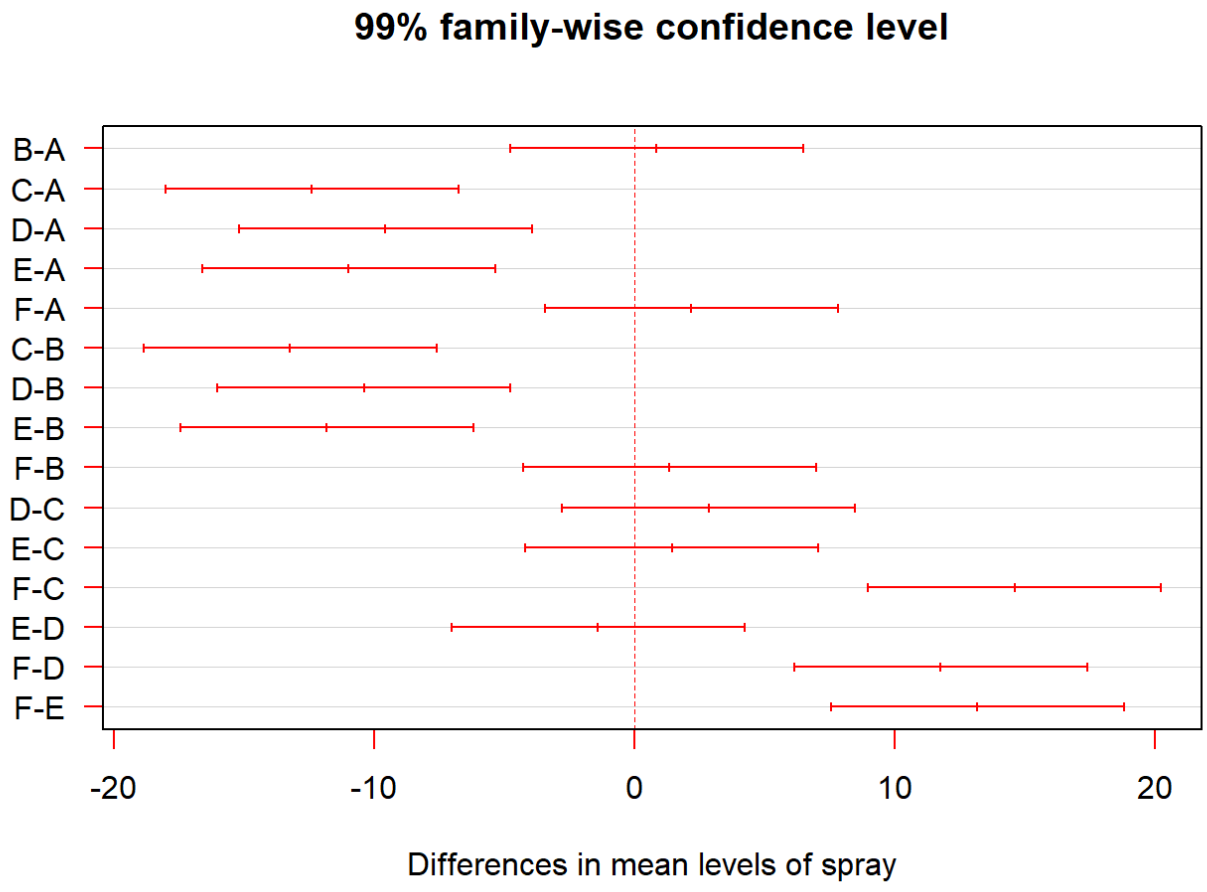
7 Post Hoc tests

-Tukey HSD(Honestly Significant Difference) is default in R

```
TukeyHSD(aov.out)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = count ~ spray, data = InsectSprays)
##
## $spray
##             diff        lwr       upr     p adj
## B-A   0.8333333  -3.866075  5.532742 0.9951810
## C-A -12.4166667 -17.116075 -7.717258 0.0000000
## D-A  -9.5833333 -14.282742 -4.883925 0.0000014
## E-A -11.0000000 -15.699409 -6.300591 0.0000000
## F-A   2.1666667  -2.532742  6.866075 0.7542147
## C-B -13.2500000 -17.949409 -8.550591 0.0000000
## D-B -10.4166667 -15.116075 -5.717258 0.0000002
## E-B -11.8333333 -16.532742 -7.133925 0.0000000
## F-B   1.3333333  -3.366075  6.032742 0.9603075
## D-C   2.8333333  -1.866075  7.532742 0.4920707
## E-C   1.4166667  -3.282742  6.116075 0.9488669
## F-C  14.5833333   9.883925 19.282742 0.0000000
## E-D  -1.4166667  -6.116075  3.282742 0.9488669
## F-D  11.7500000   7.050591 16.449409 0.0000000
## F-E  13.1666667   8.467258 17.866075 0.0000000
```

```
plot(TukeyHSD(aov.out, conf.level = 0.99),las=1, col = "red")
```



8 Regularity conditions: Test assumptions

-a.Homogeneity of variance

-b. Model checking plots

8.1 Homocedasticity

```
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(aov.out)
```

	Df <int>	F value <dbl>	Pr(>F) <dbl>
group	5	3.821356	0.004222791
	66	*NA*	*NA*

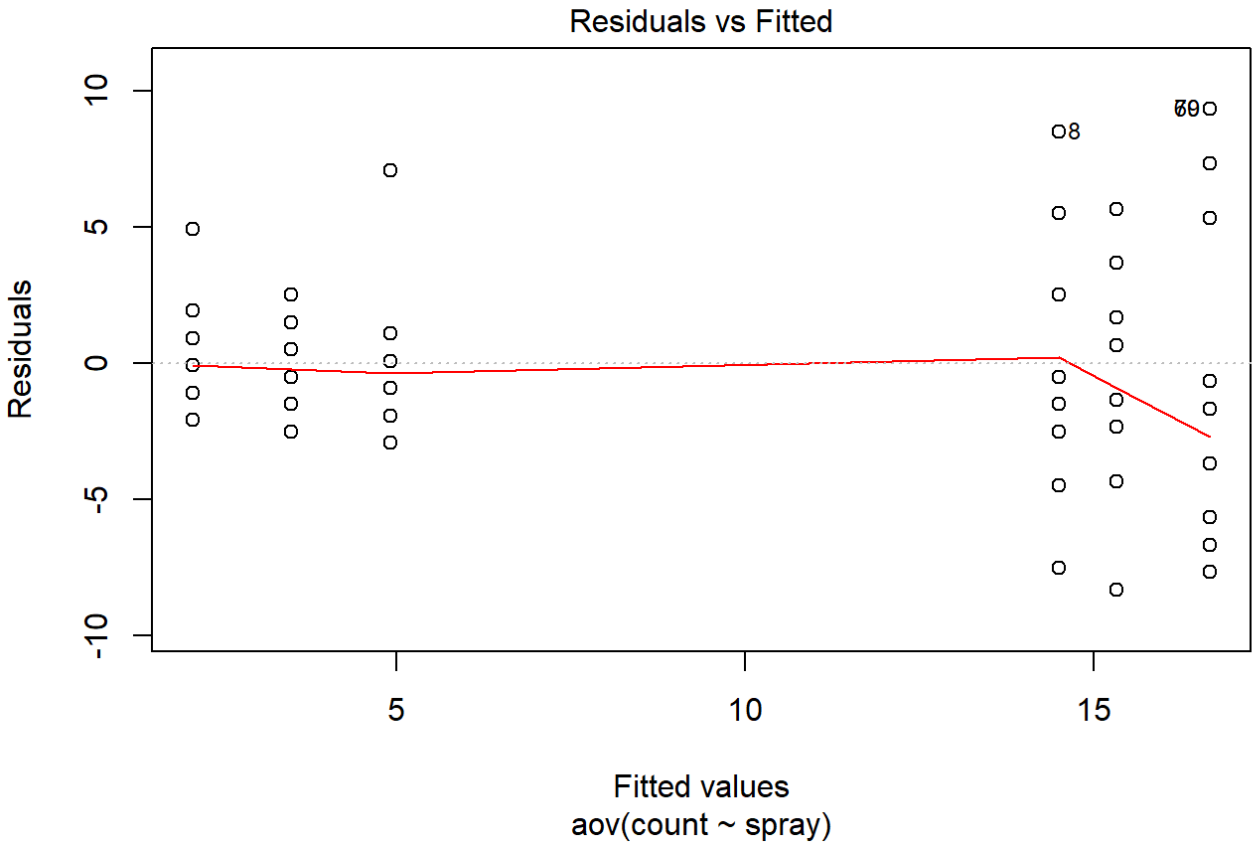
2 rows

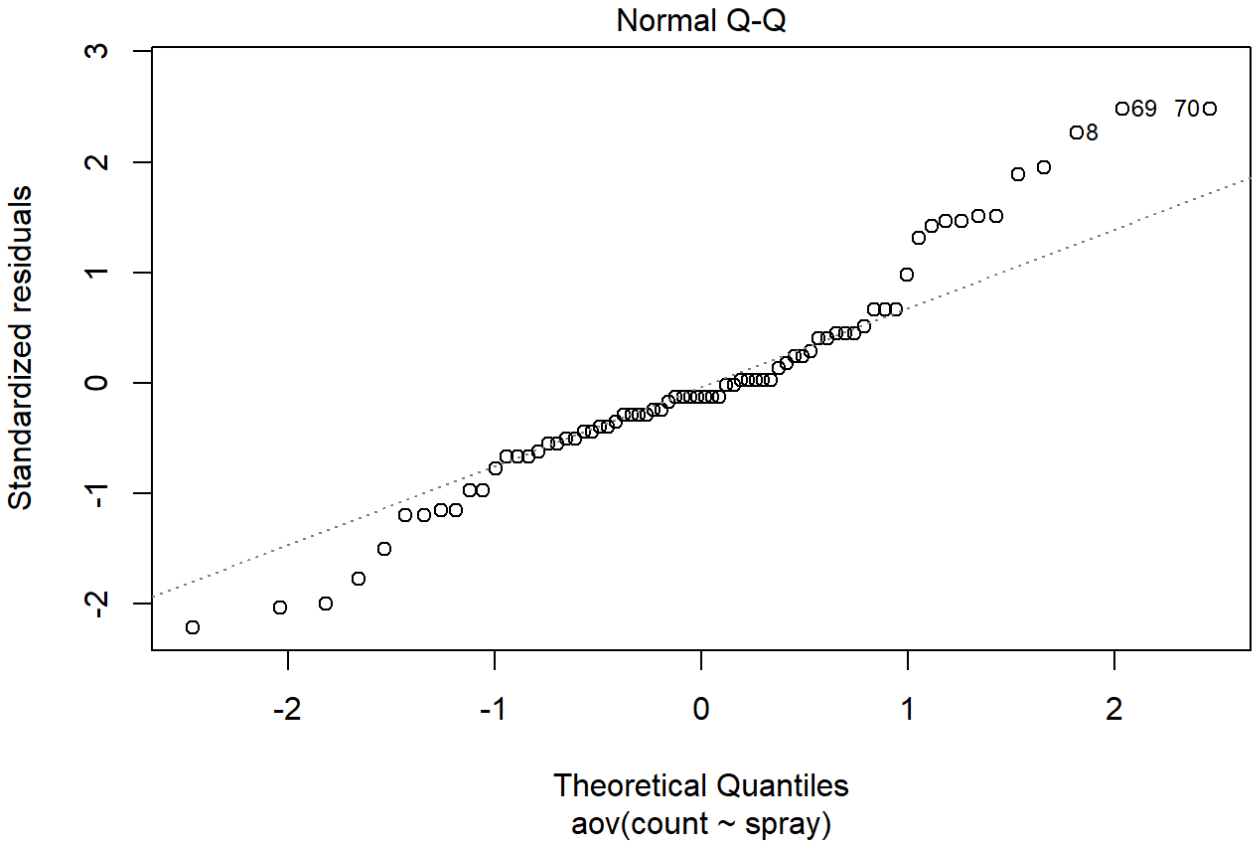
```
bartlett.test(count ~ spray, data=InsectSprays)
```

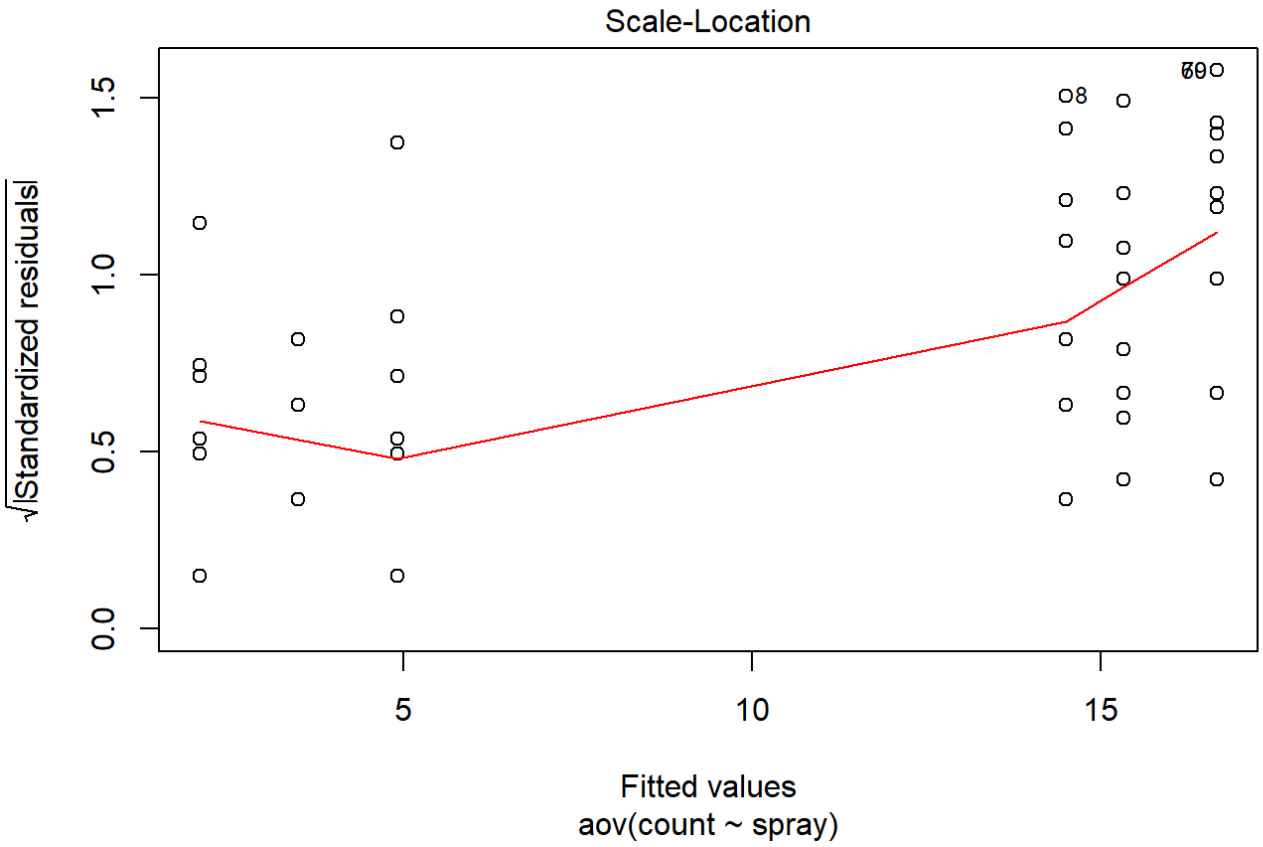
```
##
##  Bartlett test of homogeneity of variances
##
## data:  count by spray
## Bartlett's K-squared = 25.96, df = 5, p-value = 9.085e-05
```

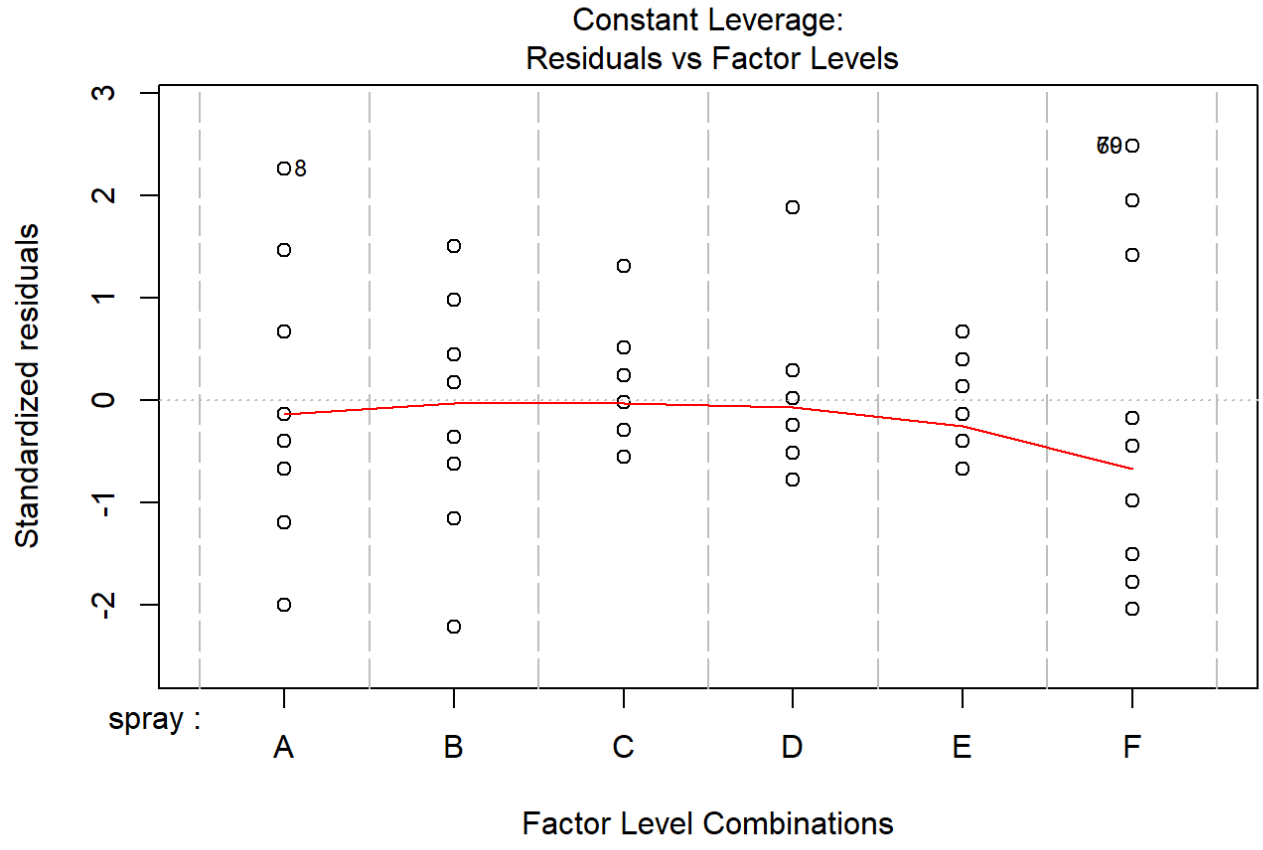
8.2 Normality

```
plot(aov.out)
```


Residuals vs Fitted
Residuals
10
5
0
-5
-10
8
69
70
5
10
15
Fitted values
aov(count ~ spray)


Normal Q-Q
Standardized residuals
3
2
1
0
-1
-2
69
70
8
-2
-1
0
1
2
Theoretical Quantiles
aov(count ~ spray)






```
shapiro.test(aov.out$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  aov.out$residuals
## W = 0.96006, p-value = 0.02226
```

First graph (Residuals vs fitted) Second graph (qqplot) Third graph (sqrt(standardized residuals) vs fitted) Fouth graph

9 Example 2: Using tidyverse

You will use the poison dataset to implement the one-way ANOVA test. The dataset contains 48 rows and 3 variables:

```
# Loading
library("readxl")
df <- read_excel("C:/Users/Usuario/Documents/CursJaume/df.xlsx")
df
```

X <chr>	time <dbl>	poison <chr>	treat <chr>
1	0.31	1	A
2	0.45	1	A
3	0.46	1	A
4	0.43	1	A
5	0.36	2	A
6	0.29	2	A
7	0.40	2	A
8	0.23	2	A
9	0.22	3	A
10	0.21	3	A

1-10 of 48 rows Previous **1** 2 3 4 5 Next

```
library(readxl)
library(tidyverse)

library(dplyr)

PATH <- "C:/Users/Usuario/Documents/CursJaume/df.xlsx"

df <- read_excel(PATH) %>%

select(-X) %>%
mutate(poison = factor(poison, ordered = TRUE))
glimpse(df)
```

```
## Observations: 48
## Variables: 3
## $ time   <dbl> 0.31, 0.45, 0.46, 0.43, 0.36, 0.29, 0.40, 0.23, 0.22, 0.21, ...
## $ poison <ord> 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 1, 1, 1, 1, 2, 2, 2, 2, ...
## $ treat  <chr> "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", ...
```

```
df
```

time <dbl>	poison <ord>	treat <chr>
0.31	1	A
0.45	1	A
0.46	1	A
0.43	1	A
0.36	2	A
0.29	2	A
0.40	2	A
0.23	2	A
0.22	3	A
0.21	3	A

1-10 of 48 rows Previous **1** 2 3 4 5 Next

```
-Time: Survival time of the animal
-poison: Type of poison used: factor level: 1,2 and 3
-treat: Type of treatment used: factor level: 1,2 and 3
```

Before you start to compute the ANOVA test, you need to prepare the data as follow:

```
-Step 1: Import the data
-Step 2: Remove unnecessary variable
-Step 3: Convert the variable poison as ordered level
```

9.1 Hipotesis test

Our objective is to test the following assumption:

```
H0: There is no difference in survival time average between group
H1: The survival time average is different for at least one group.
```

In other words, you want to know if there is a statistical difference between the mean of the survival time according to the type of poison given to the Guinea pig.

You will proceed as follow:

```
Step 1: Check the format of the variable poison
Step 2: Print the summary statistic: count, mean and standard deviation
Step 3: Plot a box plot
Step 4: Compute the one-way ANOVA test
Step 5: Run a pairwise t-test
```

9.1.1 Step 1: Some descriptive statistics

You can check the level of the poison with the following code. You should see three character values because you convert them in factor with the mutate verb.

```
# Asegurar que tenemos codificado como factor
library(tidyverse)
library(dplyr)
library(ggplot2)
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
```

```
levels(df$poison)
```

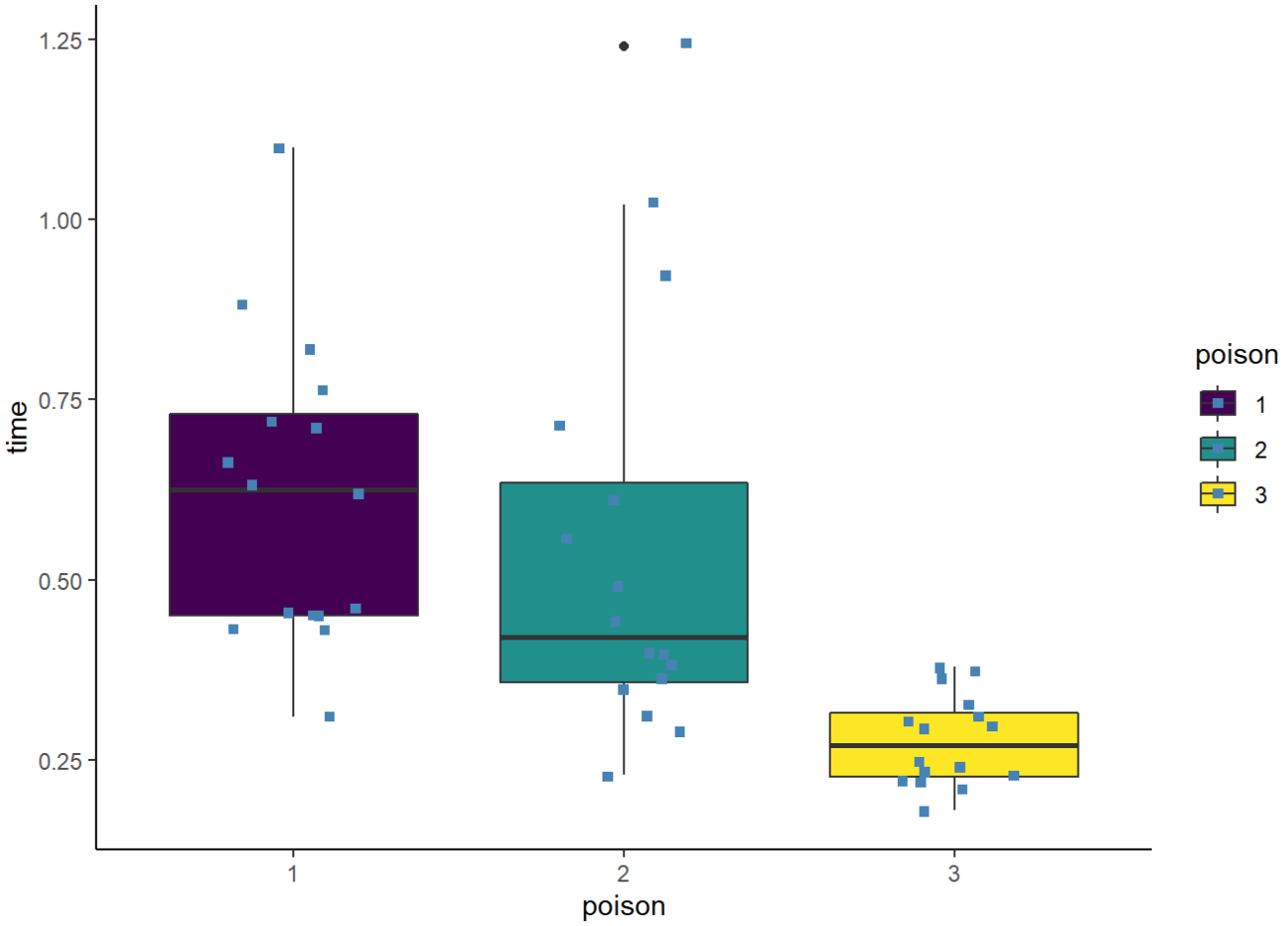
```
## [1] "1" "2" "3"
```

```
#Sumarize basico
df%>%
    group_by(poison) %>%
    summarise(
        count_poison = n(),
        mean_time = mean(time, na.rm = TRUE),
        sd_time = sd(time, na.rm = TRUE)
    )
```

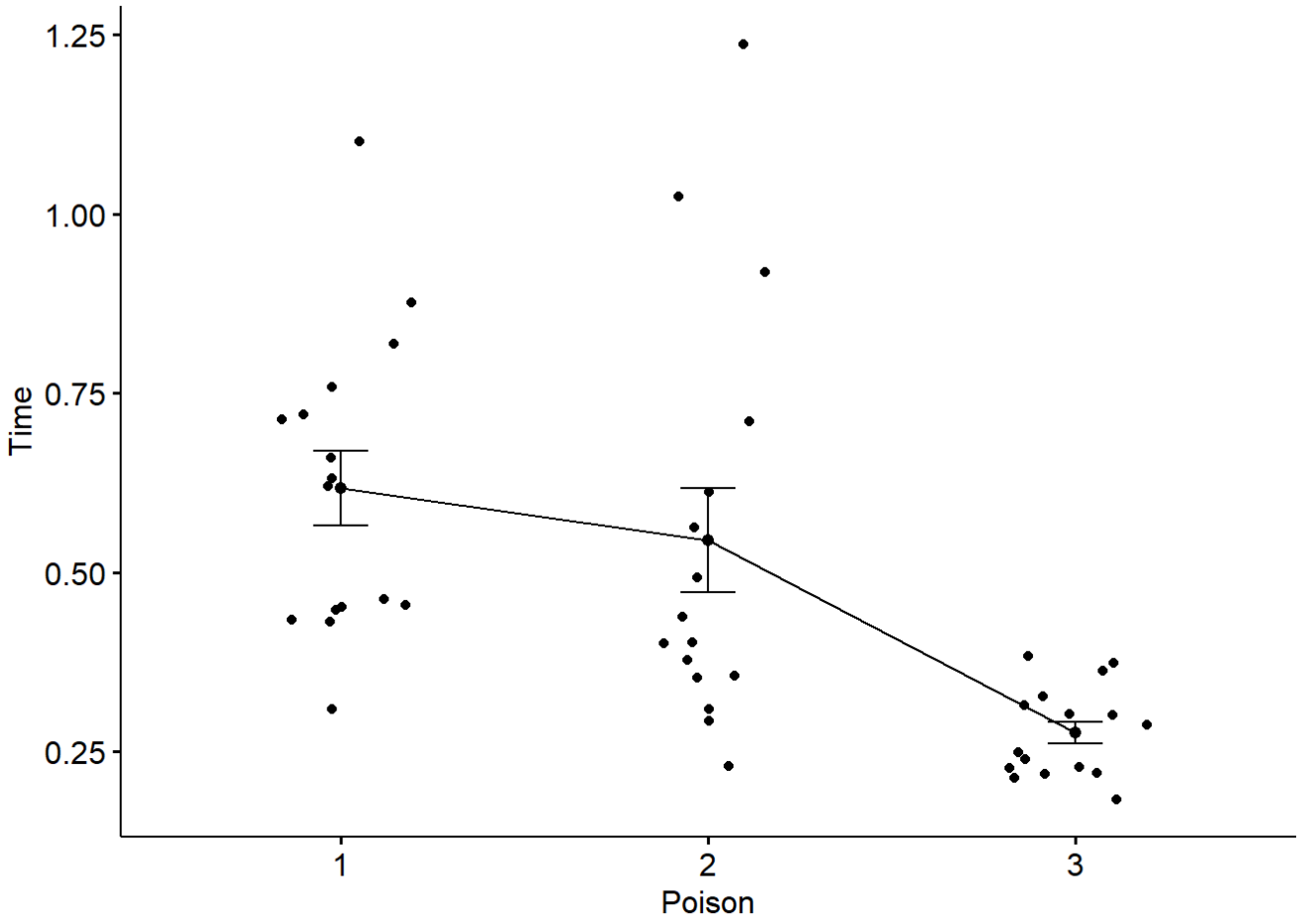
poison <ord>	count_poison <int>	mean_time <dbl>	sd_time <dbl>
1	16	0.617500	0.20942779
2	16	0.544375	0.28936641
3	16	0.276250	0.06227627

3 rows

```
#Boxplot
ggplot(df, aes(x = poison, y = time, fill = poison)) +
    geom_boxplot() +
    geom_jitter(shape = 15,
        color = "steelblue",
        position = position_jitter(0.21)) +
    theme_classic()
```



```
ggline(df, x = "poison", y = "time",
       add = c("mean_se", "jitter"),
       order = c("1", "2", "3"),
       ylab = "Time", xlab = "Poison")
```



9.1.2 Step 2: ANOVA Table

You can run the one-way ANOVA test with the command aov. The basic syntax for an ANOVA test is:

```
-aov(formula, data)
-Arguments:
-formula: The equation you want to estimate
-data: The dataset used
```

```
anova_one_way <- aov(time~poison, data = df)
summary(anova_one_way)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## poison       2  1.033  0.5165   11.79 7.66e-05 ***
## Residuals   45  1.972  0.0438
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lm.out = lm(time~poison, data = df)
anova(lm.out)
```

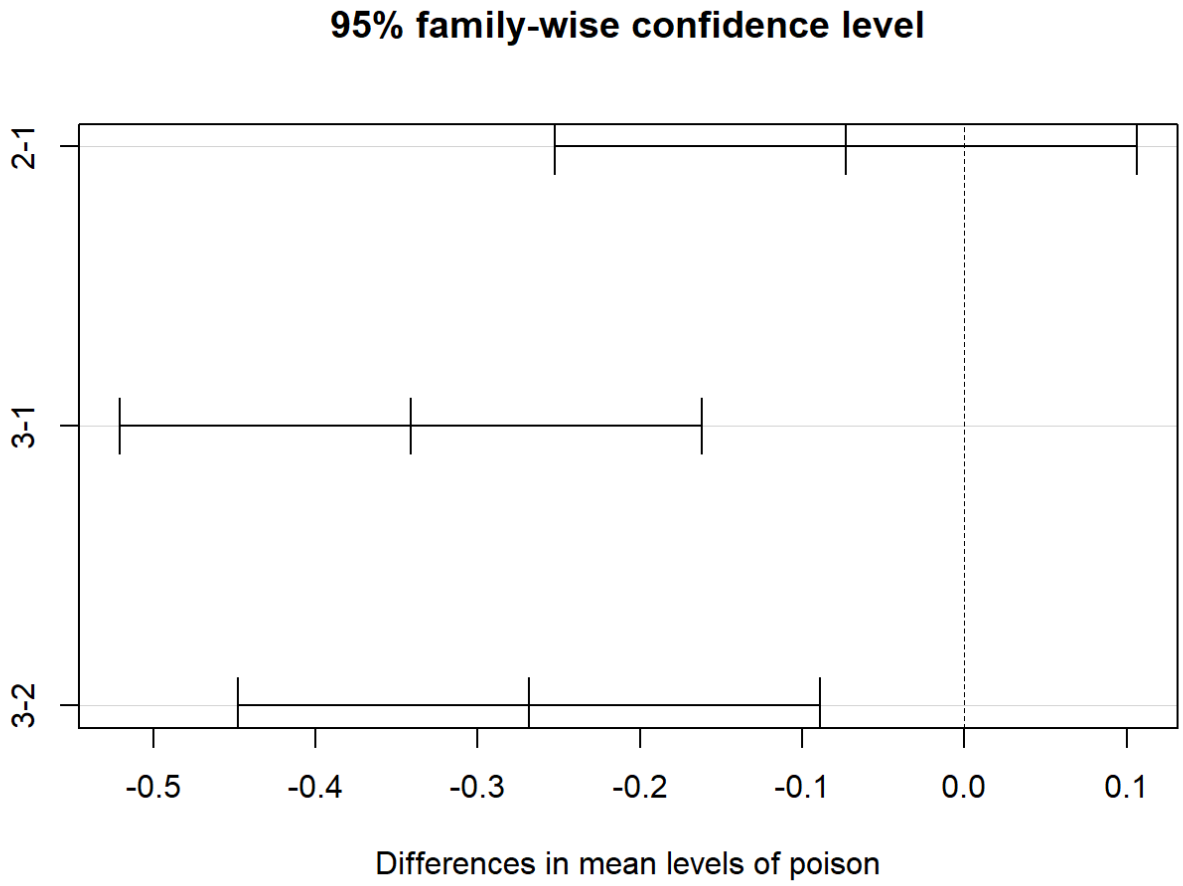
	Df <int>	Sum Sq <dbl>	Mean Sq <dbl>	F value <dbl>	Pr(>F) <dbl>
poison	2	1.033013	0.51650625	11.78599	7.655635e-05
Residuals	45	1.972069	0.04382375	*NA*	*NA*

2 rows

9.1.3 Pairwise comparison ()

The one-way ANOVA test does not inform which group has a different mean. Instead, you can perform a Tukey test with the function TukeyHSD().

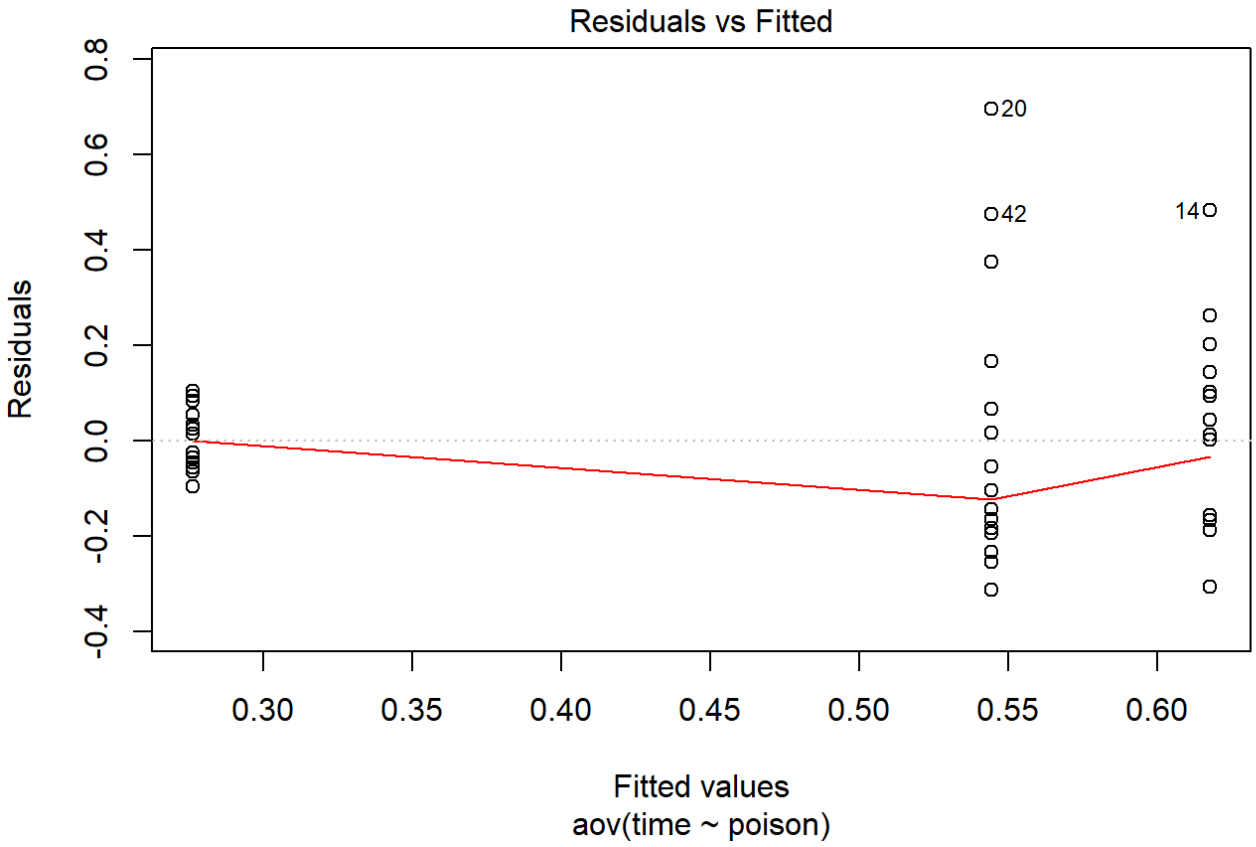
```
multcomp<-TukeyHSD(anova_one_way)
plot(multcomp)
```

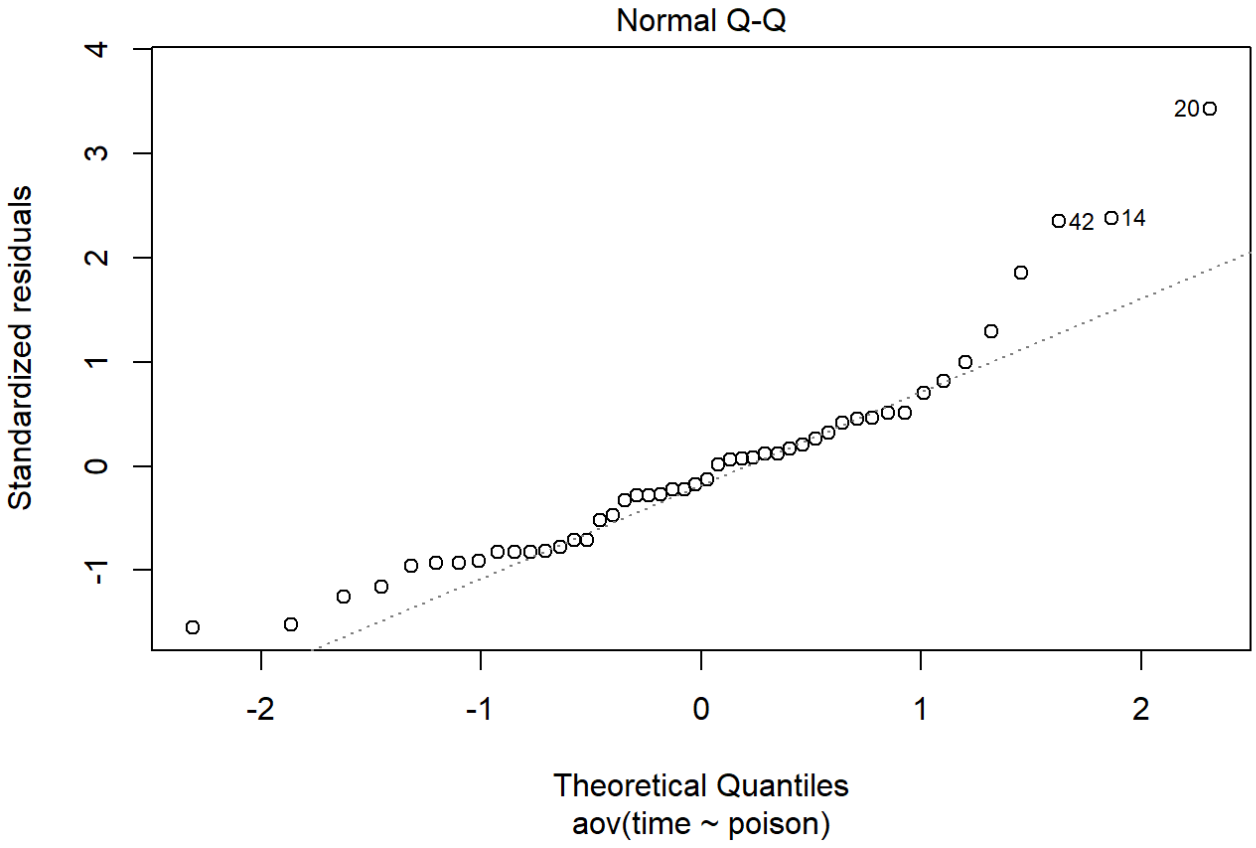


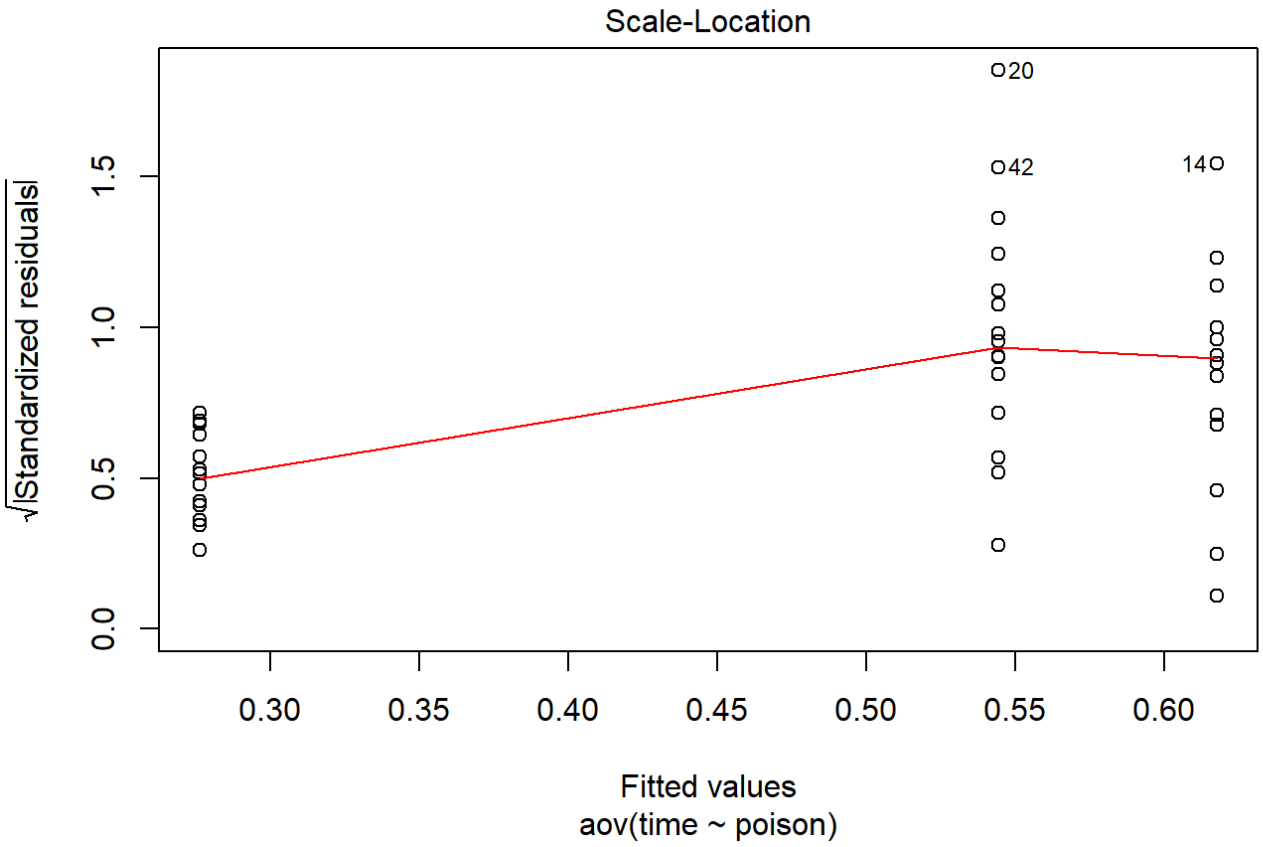
Regularity conditions

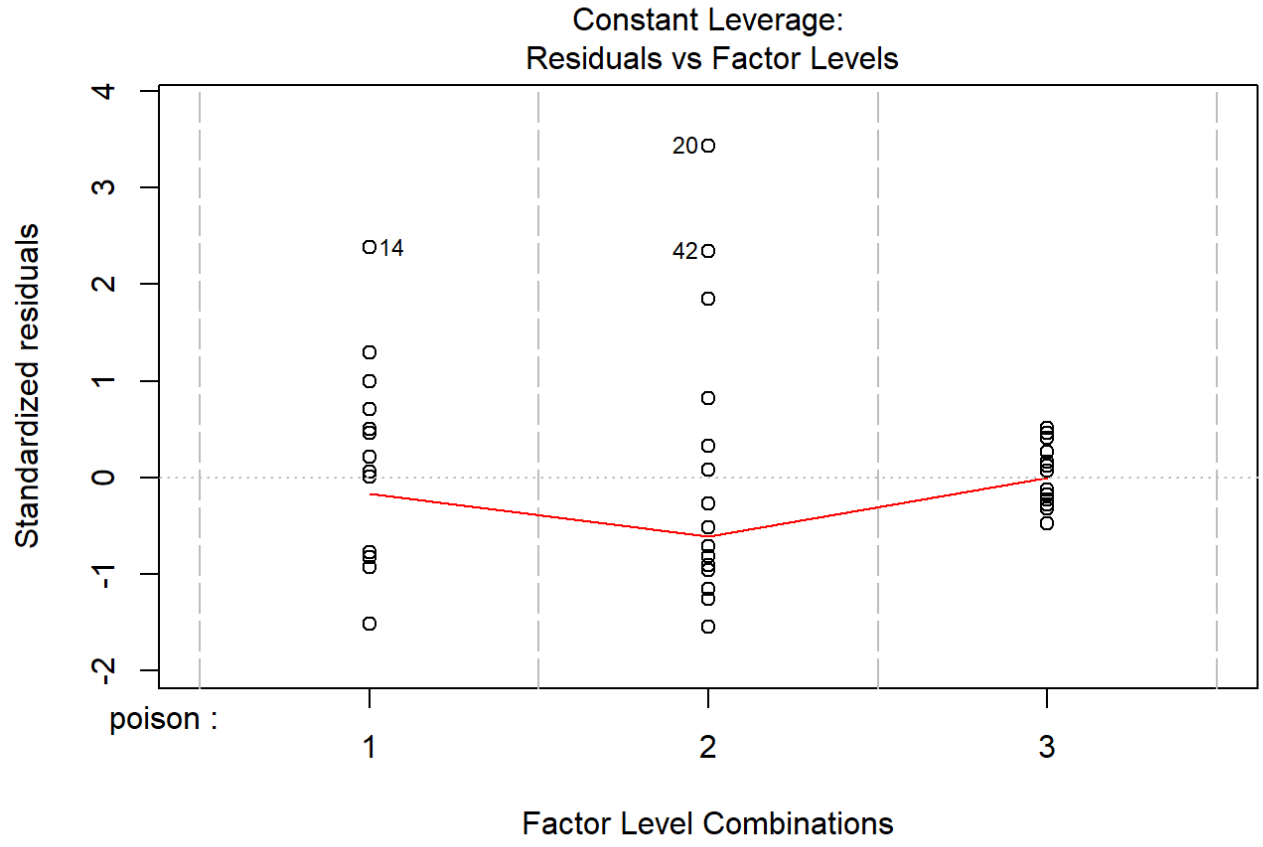
```
-Check for homogeneity of variances, (Graph 4)

-Check for Normality (residuals) (Graph 2)
```

```
plot(anova_one_way)
```


Residuals vs Fitted
Residuals
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
20
42
14
0.30
0.35
0.40
0.45
0.50
0.55
0.60
Fitted values
aov(time ~ poison)


Normal Q-Q
Standardized residuals
4
3
2
1
0
-1
20
42
14
-2
-1
0
1
2
Theoretical Quantiles
aov(time ~ poison)





10 Additional results: Effect sizes estimates

and the strength of our prediction

One relatively common question in statistics or data science is, how “big” is the difference or the effect?

Effect size is a measure we use in statistics to express how big the differences are. For this Oneway ANOVA the appropriate measure of effect size is eta squared (η^2). The more variance you explain the bigger the effect.

```
anovalm <- anova(lm(time ~ poison, data=df))

SSeffect<-anovalm["poison", "Sum Sq"]
SSresidual<-anovalm["Residuals", "Sum Sq"]

etaSq <- SSeffect / (SSeffect + SSresidual)
etaSq
```

```
## [1] 0.3437553
```

```
Input =("
Location   Aam
Tillamook  0.0571
Tillamook  0.0813
Tillamook  0.0831
Tillamook  0.0976
Tillamook  0.0817
Tillamook  0.0859
Tillamook  0.0735
Tillamook  0.0659
Tillamook  0.0923
Tillamook  0.0836
Newport    0.0873
Newport    0.0662
Newport    0.0672
Newport    0.0819
Newport    0.0749
Newport    0.0649
Newport    0.0835
Newport    0.0725
Petersburg 0.0974
Petersburg 0.1352
Petersburg 0.0817
Petersburg 0.1016
Petersburg 0.0968
Petersburg 0.1064
Petersburg 0.1050
Magadan    0.1033
Magadan    0.0915
Magadan    0.0781
Magadan    0.0685
Magadan    0.0677
Magadan    0.0697
Magadan    0.0764
Magadan    0.0689
Tvarminne  0.0703
Tvarminne  0.1026
Tvarminne  0.0956
Tvarminne  0.0973
Tvarminne  0.1039
Tvarminne  0.1045
")

Data = read.table(textConnection(Input),header=TRUE)
Data
```

Location <fctr>	**Aam** <dbl>
Tillamook	0.0571
Tillamook	0.0813
Tillamook	0.0831

Location <fctr>	Aam <dbl>
Tillamook	0.0976
Tillamook	0.0817
Tillamook	0.0859
Tillamook	0.0735
Tillamook	0.0659
Tillamook	0.0923
Tillamook	0.0836

1-10 of 39 rows Previous **1** 2 3 4 Next

```
#Fit the linear model and conduct ANOVA

model = lm(Aam ~ Location,
           data=Data)

a1<-anova(model)
a1
```

	Df <int>	Sum Sq <dbl>	Mean Sq <dbl>	F value <dbl>	Pr(>F) <dbl>
Location	4	0.004519674	0.0011299185	7.121019	0.0002812242
Residuals	34	0.005394906	0.0001586737	*NA*	*NA*

2 rows

```
library(car)

a2<-Anova(model, type="II")
a2
```

	Sum Sq <dbl>	Df <dbl>	F value <dbl>	Pr(>F) <dbl>
Location	0.004519674	4	7.121019	0.0002812242
Residuals	0.005394906	34	*NA*	*NA*

2 rows

```
#If you use type="III", you need the following line before the analysis
# options(contrasts = c("contr.sum", "contr.poly"))


options(contrasts = c("contr.sum", "contr.poly"))
a3<-Anova(model, type="III")
a3
```

	Sum Sq <dbl>	Df <dbl>	F value <dbl>	Pr(>F) <dbl>
(Intercept)	0.048687601	1	306.841023	1.368547e-18
Location	0.004519674	4	7.121019	2.812242e-04
Residuals	0.005394906	34	*NA*	*NA*

3 rows

```
drop1(model,.~.,test="F")
```

	Df <dbl>	Sum of Sq <dbl>	RSS <dbl>	AIC <dbl>	F value <dbl>	Pr(>F) <dbl>
<none>	*NA*	*NA*	0.005394906	-336.5486	*NA*	*NA*
Location	4	0.004519674	0.009914580	-320.8151	7.121019	0.0002812242

2 rows

11 Random Effects Models: One-Way ANOVA

Model:

$$Y_{ij} = \mu + A_i + e_{ij} \; being \;\; A_i \approx N(0, \sigma_A) \;\; e_{ij} i.i.d \approx N(0, \sigma^2)$$

Now we use another point of view: we consider situations where treatments are random samples from a large population of treatments. This might seem quite special at first sight, but it is actually very natural in many situations. Think for example of a random sample of school classes that were drawn from all school classes in a country. Another example could be machines that were randomly sampled from a large population of machines. Typically, we are interested in making a statement about some properties of the whole population and not of the observed individuals (here: school classes or machines).

This “small” change will have a large impact on the properties of the model. In addition, we have a new parameter σ_A^2 which is the variance of the random effect (here the variance between different machines). Sometimes, such models are also called **variance components models** because of the different variances σ_A^2, σ^2

Parameter estimation for the variance components σ2α and σ2 is typically being done with a technique called restricted maximum likelihood (REML). We could also use “classical” maximum-likelihood estimators here, but REML estimates are less biased. The parameter μ is estimated with maximum-likelihood assuming that the variances are known.

In R there are many packages that can fit such models. We will consider **lme4** (Bates et al. 2017) and later also **lmerTest** (Kuznetsova, Bruun Brockhoff, and Haubo Bojesen Christensen 2016).

Now we fit the random effects model with the lmer function in package lme4. A random effect can be specified with the notation (1 | factor) (facor is “ypur factor” in the model formula. This means that the “granularity” of the random effect is specified after the vertical bar “|”.

11.1 Example: Random efefct One way Anova

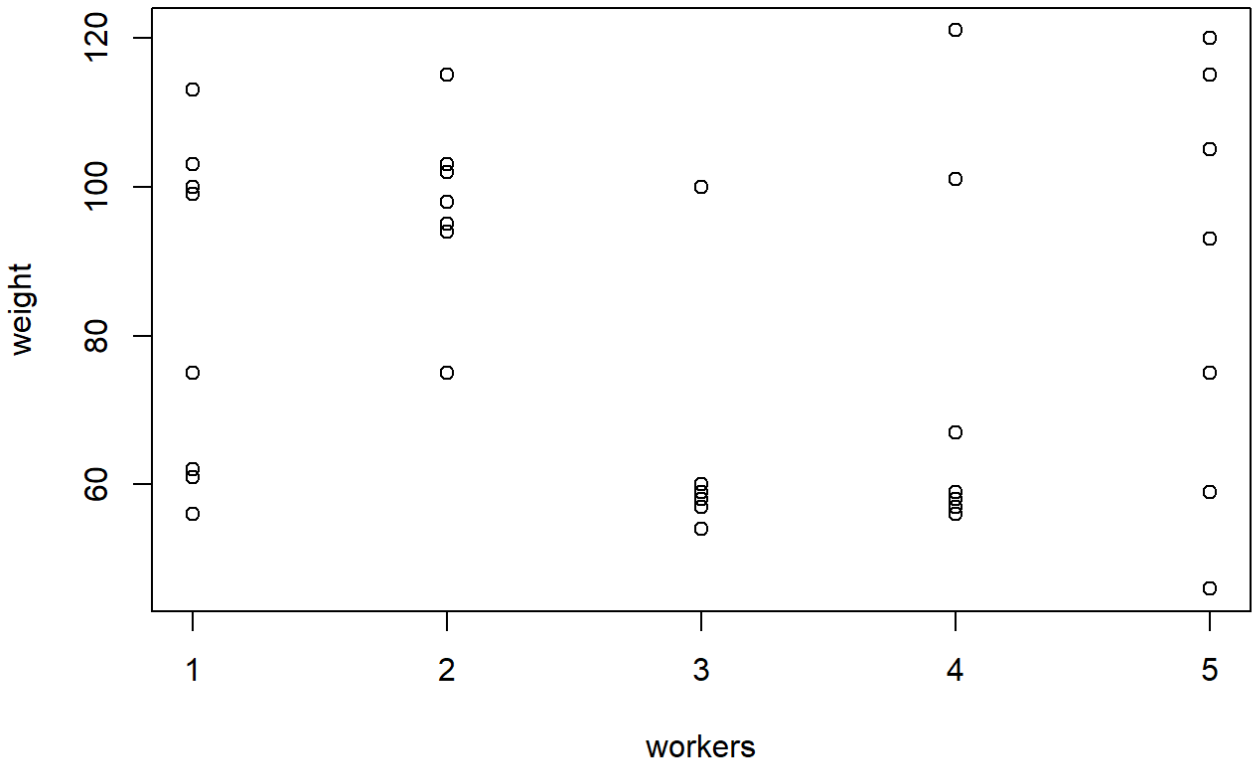
Supongamos que ahora lo que queremos es ver si una determinada empresa cumple los estandars de calidad exigidos por el organismo competente. Para ellos evaluaremosentre los diferentes operarios de una fábrica si hay diferencias en cuanto a la calidad del producto elaborado. En este caso estamos planteando como variable respuesta el peso en mg del proceso de pesaje de un determinado producto llevado a cabo por dichos operarios. Plantenamos pues seleccionar 5 operarios al azar y tomamos una muestra de 8 productos fabricados por cada uno de esos 5 operarios. Estamos ahora ante un caso de un factor aleatorio (el factor operario), de donde hemos tomado una muestra de 5 operarios al azar.

We first create the data set and visualize it.

```
## Create data set ####
weight <- c(61, 100,  56, 113,  99, 103,  75,  62,  ## work 1
            75, 102,  95, 103,  98, 115,  98,  94,  ## work 2
            58,  60,  60,  57,  57,  59,  54, 100,  ## work 3
            57,  56,  67,  59,  58, 121, 101, 101,  ## work 4
            59,  46, 120, 115, 115,  93, 105,  75)  ## work 5
workers    <- factor(rep(1:5, each = 8))
dataset <- data.frame(weight, workers)
str(dataset)
```

```
## 'data.frame':    40 obs. of  2 variables:
##  $ weight : num  61 100 56 113 99 103 75 62 75 102 ...
##  $ workers: Factor w/ 5 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 2 2 ...
```

```
## Visualize data ####
stripchart(weight ~ workers, vertical = TRUE, pch = 1, xlab = "workers", data = da
taset)
```



At first sight it looks like the variation between different sires is rather small.

Now we fit the random effects model with the lmer function in package lme4. We want to have a random effect per workers. This can be specified with the notation (1 | workers) in the model formula. This means that the random effect is specified after the vertical bar "|". All observations sharing the same level of sire will get the same random effect α_i. The 1 means that we want to have a random intercept per workers.

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car
```

```
fit.workers <- lmer(weight ~ (1 | workers), data = dataset)
summary(fit.workers)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: weight ~ (1 | workers)
##    Data: dataset
##
## REML criterion at convergence: 358.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9593 -0.7459 -0.1581  0.8143  1.9421
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  workers  (Intercept) 116.7    10.81
##  Residual             463.8    21.54
## Number of obs: 40, groups:  workers, 5
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   82.550      5.911   13.96
```

```
# Approximate confidence intervals can be obtained with the function confint
confint(fit.workers, oldNames = FALSE)
```

```
## Computing profile confidence intervals ...
```

```
##                            2.5 %   97.5 %
## sd_(Intercept)|workers  0.00000 24.61580
## sigma                  17.32943 27.76544
## (Intercept)            69.83802 95.26197
```

n the summary we can read off the table labelled **Random Effects** that $\hat{\sigma}_{\alpha}^2$=116.7 and $\hat{\sigma}^2$=463.8.

Note that the column Std.Dev is nothing else than the square root of the variance and not the standard error (accuracy) of the variance estimate.

The variance of Y_ij is therefore estimated as $116.7 + 463.8 = 580.5$. Hence, only about $116.7/580.5 = 20\%$ of the **total variance of the birth weight** is due to **workers** (this is the intraclass correlation).

Under Fixed effects we find the estimate $\hat{\mu}$= 82.55

It is an estimate for the expected weight of a worker of a randomly selected workers (randomly selected from the whole population of all workers). Hence, an approximate confidence interval for $\hat{\sigma}_{\alpha}^2$ is given by [0;24.62]. We see that the estimate $\hat{\sigma}_{\alpha}^2$ is therefore quite imprecise. The reason is that we only have five workers to estimate the variance.

Example 2: Los medios de cultivo bacteriológico en los laboratorios de los hospitales proceden de diversos fabricantes. Se sospecha que la calidad de estos medios de cultivo varía de un fabricante a otro. Para comprobar esta teoría, se hace una lista de fabricantes de un medio de cultivo concreto, se seleccionan

aleatoriamente los nombres de cinco de los que aparecen en la lista y se comparan las muestras de los instrumentos procedentes de éstos. La comprobación se realiza colocando sobre una placa dos dosis, en gotas, de una suspensión medida de un microorganismo clásico, Escherichia coli, dejando al cultivo crecer durante veinticuatro horas, y determinando después el número de colonias (en millares) del microorganismo que aparecen al final del período. Se quiere comprobar si la calidad del instrumental difiere entre fabricantes.

```
-Las cinco muestras representan muestras aleatorias independientes extraídas de po
blaciones seleccionadas aleatoriamente de un conjunto mayor de poblaciones.

-Todas las poblaciones del conjunto más amplio tienen distribución Normal, de modo
que cada una de las 5 poblaciones muestreadas se distribuyen según una Normal
```

```
library(lme4)
bacterias<-read.table("C:/Users/Usuario/Documents/CursJaume/Anova1factorRandom.tx
t", header = TRUE)
bacterias$Fabricante<- factor(bacterias$Fabricante)
mod.fabricante <- lmer(Calidad ~ (1 | Fabricante), data = bacterias)
summary(mod.fabricante)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Calidad ~ (1 | Fabricante)
##    Data: bacterias
##
## REML criterion at convergence: 494.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1990 -0.4149  0.1667  0.6673  1.7973
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  Fabricante (Intercept) 1193     34.54
##  Residual               3607     60.06
## Number of obs: 45, groups:  Fabricante, 5
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   299.87      17.85    16.8
```

```
# Approximate confidence intervals can be obtained with the function confint
confint(mod.fabricante, oldNames = FALSE)
```

```
## Computing profile confidence intervals ...
```

```
##                                2.5 %    97.5 %
## sd_(Intercept)|Fabricante   1.944988  75.18397
## sigma                      48.962654  76.07079
## (Intercept)               261.478002 338.25532
```

12 Non parametric approach

Of course if you really want to be cautious about all of your assumptions (normality and homoscedasticity) then the non-parametric Kruskal-Wallis rank sum test is the way to go. As the name implies it uses ranks for the dependent variable mileage rather than the number of miles itself. What the test essentially does is test the hypothesis that all the group medians are equal. That is the equivalent omnibus test to a traditional Oneway ANOVA. The Dunn test is the analog to the post hoc pairwise comparisons we ran earlier. I've shown both separately but conveniently R reports both if you just run the second command.

```
kruskal.test(count ~ spray, data=InsectSprays)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  count by spray
## Kruskal-Wallis chi-squared = 54.691, df = 5, p-value = 1.511e-10
```

```
library(dunn.test)
dunn.test(InsectSprays$count,InsectSprays$spray,method = "holm",alpha = 0.01)
```

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 54.6913, df = 5, p-value = 0
##
##
##                            Comparison of x by group
##                                     (Holm)
## Col Mean-|
## Row Mean |          A          B          C          D          E
## ---------+----------------------------------------------------------
##        B |  -0.312733
##          |     0.7545
##          |
##        C |   4.774077   5.086811
##          |    0.0000*    0.0000*
##          |
##        D |   3.117565   3.430299  -1.656512
##          |     0.0064    0.0024*     0.2929
##          |
##        E |   3.850535   4.163269  -0.923542   0.732969
##          |    0.0006*    0.0002*     0.8893     0.9272
##          |
##        F |  -0.405576  -0.092842  -5.179654  -3.523142  -4.256112
##          |     1.0000     0.4630    0.0000*    0.0019*    0.0001*
##
## alpha = 0.01
## Reject Ho if p <= alpha/2
```

13 Bayesian approach

Briefly Bayesian methods allow us to calculate the probability or the odds that the mean **count for spray** is different.

If the Bayes Factor is for example 5 we would say that the odds are 5:1 in favor of the hypothesis that **spray matter**.

Bayes factor Interpretation
1 - 3 Negligible evidence . 3 - 20 Positive evidence
20 - 150 Strong evidence
150 and above Very strong evidence

```
library(BayesFactor)
```

```
## Loading required package: coda
```

```
## ************
## Welcome to BayesFactor 0.9.12-4.2. If you have questions, please contact Richar
d Morey (richarddmorey@gmail.com).
##
## Type BFManual() to open the manual.
## ************
```

```
bayesfactor<- anovaBF(count ~ spray, data=InsectSprays)
summary(bayesfactor)
```

```
## Bayes factor analysis
## --------------
## [1] spray : 1.506706e+14 ±0%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

14 Box Cox transformation

-log(Y)Transformación válida para valores positivo. Empleada cunado cuando la distribución de los datos tiene una cola a la derecha o cuando el grupo que tiene la media mayor también tiene la desviación estándar mayor o cuando los datos comprenden varios ordenes de magnitud

-logit(Y) Usada fundamentalmente para proporciones

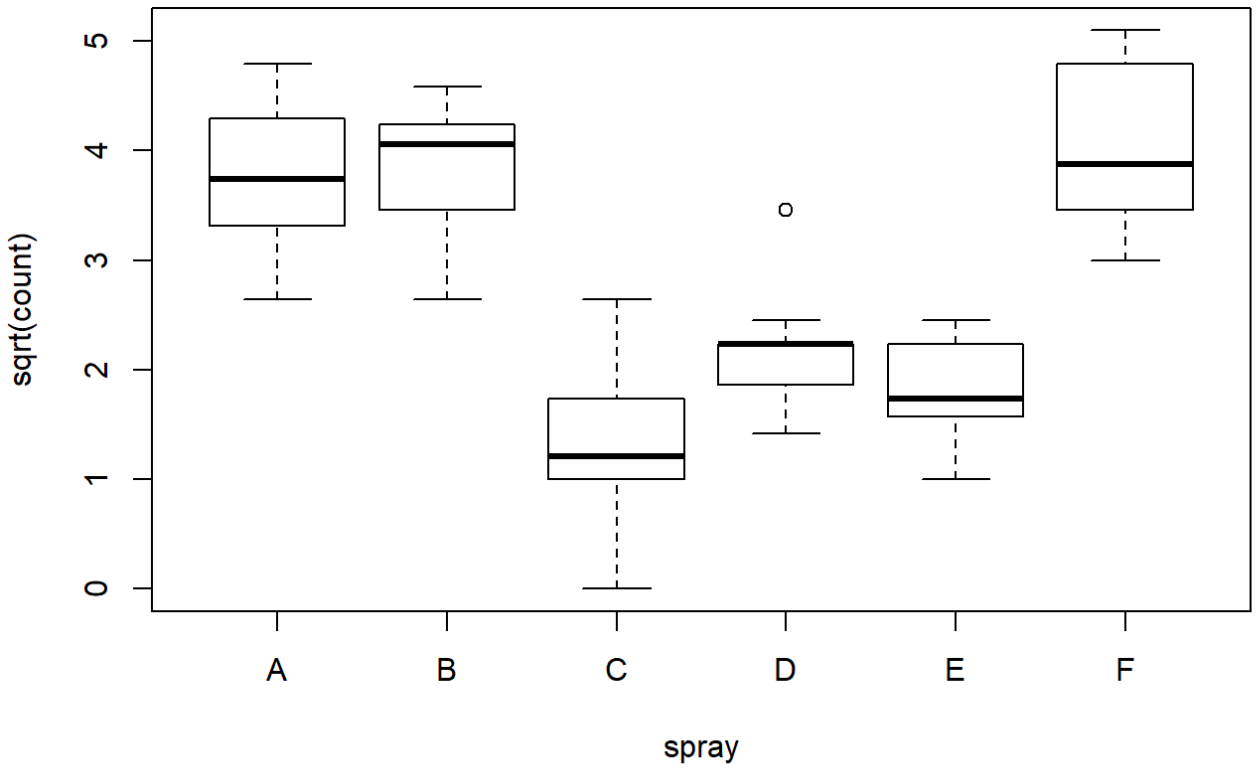
-sqrt(Y) Usada cuando los datos son conteos.

be careful, transformaciones porque podríamos incurrir en los mismos problemas que al hacer comparaciones múltiples (Incremento del error de tipo I)

```
bartlett.test(sqrt(count) ~ spray, data=InsectSprays)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  sqrt(count) by spray
## Bartlett's K-squared = 3.7525, df = 5, p-value = 0.5856
```

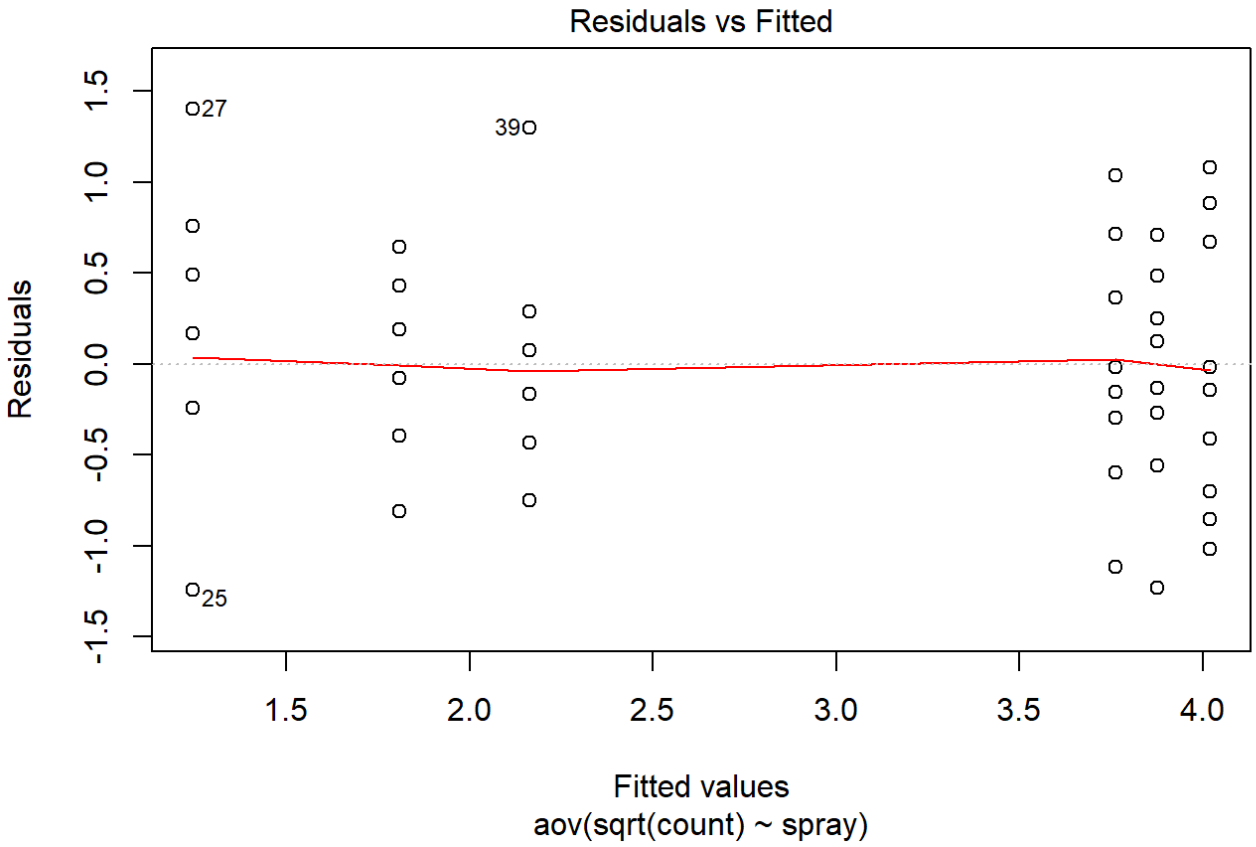
```
with(InsectSprays,boxplot(sqrt(count) ~ spray))
```



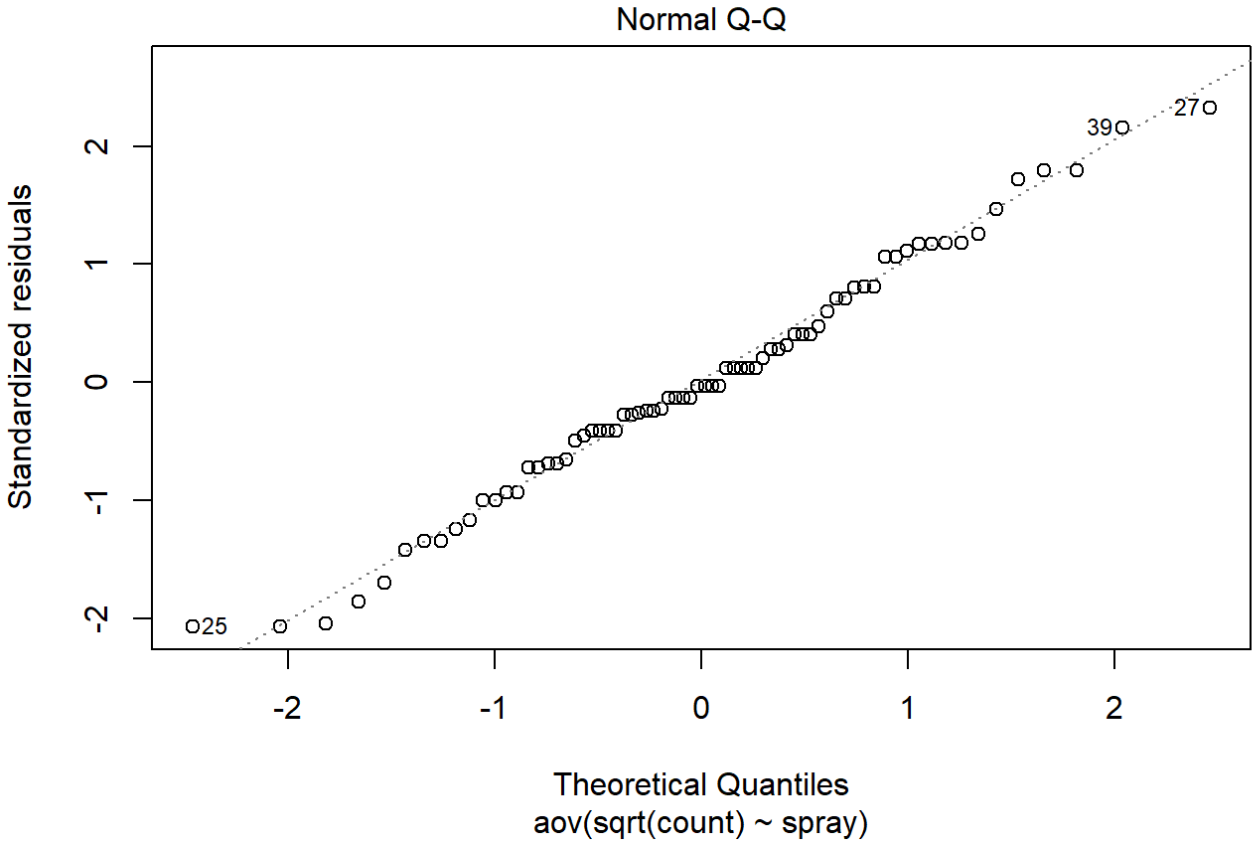

```
salida.aov2 <-aov(sqrt(count)~ spray, data = InsectSprays)

summary(salida.aov2)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## spray        5  88.44  17.688    44.8 <2e-16 ***
## Residuals   66  26.06   0.395
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(salida.aov2,1)
```




```
plot(salida.aov2,2)
```



15 Anex: The ANOVA SS types

Annoyingly, R uses Type I sum of squares by default. For unbalanced designs, order of terms will affect the p-values with Type I SS. Even more annoyingly, most guides to using R don't alert you to problems with using the native anova function for analysis of variance with unbalanced designs. The simplest answer is that you want to define your model with lm and then use the Anova function in the car package, with which you can specify Type II or Type III sum of squares. But note that you should change the global options(contrasts = settings if you are going to use Type III

```
library(car)
data(mtcars)
mtcars$cyl = factor(mtcars$cyl)
mtcars$am = factor(mtcars$am)
model = lm(mpg ~ cyl*am, data=mtcars)

### Type I
anova(model)
### Type II
Anova(model)

### Type III
options(contrasts=c("contr.sum","contr.poly")
model2 = lm(mpg ~ cyl*am, data=mtcars)

Anova(model, type=3)
options(contrasts = c("contr.treatment", "contr.poly"))
```

ANOVA is a statistical process for analysing the amount of variance that is contributed to a sample by different factors.

When data is **unbalanced**, there are different ways to calculate the sums of squares for ANOVA.

There are at least 3 approaches, commonly called **Type I, II and III sums of squares**.

15.1 Type I, II and III Sums of Squares

Consider a model that includes two factors A and B; there are therefore two main effects, and an interaction, AB. The full model is represented by SS(A, B, AB).

It is convenient to define incremental sums of squares to represent these differences.

```
SS(AB | A, B) = SS(A, B, AB) - SS(A, B)
SS(A | B, AB) = SS(A, B, AB) - SS(B, AB)
SS(B | A, AB) = SS(A, B, AB) - SS(A, AB)
SS(A | B)     = SS(A, B) - SS(B)
SS(B | A)     = SS(A, B) - SS(A)
```

-SS(AB | A, B) represents "the sum of squares for interaction after the main effects", and -SS(A | B) is "the sum of squares for the A main effect after the B main effect and ignoring interactions"

-When data is balanced, **the factors are orthogonal**, and **types I, II and III all give the same results**.

15.1.1 Type I, also called "sequential" sum of squares:

```
SS(A) for factor A.
SS(B | A) for factor B.
SS(AB | B, A) for interaction AB.
```

This tests the main effect of factor A, followed by the main effect of factor B after the main effect of A, followed by the interaction effect AB after the main effects.

Because of the sequential nature and the fact that the two main factors are tested in a particular order, **this type of sums of squares will give different results for unbalanced data depending on which main effect is considered first**.

For unbalanced data, this approach tests for a **difference in the weighted marginal means**. In practical terms, this means that the results are dependent on the realized sample sizes. In other words, it is **testing the first factor without controlling for the other factor**.

Note that this is often **not** the hypothesis that is of interest when dealing with unbalanced data.

15.1.2 Type II:

```
SS(A | B) for factor A.
SS(B | A) for factor B.
```

This type tests for each main effect after the other main effect.

Note that **no significant interaction** is assumed (in other words, you should test for interaction first (SS(AB | A, B)) and only if AB is not significant, continue with the analysis for main effects).

If there is indeed no interaction, then type II is statistically more powerful than type III (see Langsrud [3] for further details).

Computationally, this is equivalent to running a type I analysis with different orders of the factors, and taking the appropriate output (the second, where one main effect is run after the other, in the example above).

15.1.3 Type III:

```
SS(A | B, AB) for factor A.
SS(B | A, AB) for factor B.
```

This type tests for the presence of a main effect after the other main effect and interaction. This approach is therefore valid in the presence of significant interactions.

If the interactions are not significant, type II gives a more powerful test.

16 Summary

```
-Usually the hypothesis of interest is about the significance of one factor while
controlling for the level of the other factors. If the data is unbalanced, this eq
uates to using type II or III SS.
-In general, if there is no significant interaction effect, then type II is more p
owerful, and follows the principle of marginality.
-If interaction is present, then type II is inappropriate while type III can still
be used, but results need to be interpreted with caution.
```

17 Annex 2

17.1 The **anova and aov** Functions in R

The **anova and aov functions in R implement a sequential sum of squares (type I)**.

As indicated above, for unbalanced data, this rarely tests a hypothesis of interest, since essentially the effect of one factor is calculated based on the varying levels of the other factor.

In a practical sense, this means that the results are interpretable only in relation to the particular levels of observations that occur in the (unbalanced) data set.

Fortunately, based on the above discussion, it should be clear that it is relatively straightforward to obtain type II SS in R.

17.1.1 Type II SS in R

Since type II SS tests each main effect after the other main effects, and assumes no interactions, **the correct SS can be obtained using anova() and varying the order of the factors**.

For example, consider a data frame (search) for which the response variable is the time that it takes users to find a relevant answer with an information retreival system (time).

The user is assigned to one of two experimental search systems on which they run the test (sys). They are also assigned a number of different search queries (topic).

To obtain type I SS:

```
anova(lm(time ~ sys * topic, data=search))
```

If the data is unbalanced, you will obtain slightly different results if you instead use:

```
anova(lm(time ~  topic * sys, data=search))
```

The type II SS is obtained by using the second line of output from each of the above commands (since in type I SS, the second component will be the second factor, after the first factor). That is, you obtain the type II SS results for topic from the first command, and the results for sys from the second.

17.1.2 Type III SS in R

This is slightly more involved than the type II results.

First, it is necessary to set the contrasts option in R. Because the multi-way ANOVA model is over-parameterised, it is necessary to choose a contrasts setting that sums to zero, otherwise the ANOVA analysis will give incorrect results with respect to the expected hypothesis. (The default contrasts type does not satisfy this requirement.)

```
options(contrasts = c("contr.sum","contr.poly"))
```

Next, store the model:

```
model <- lm(time ~  topic * sys, data=search)
```

Finally, call the **drop1 function*+ on each model component:

```
drop1(model, .~., test="F")
```

The results give the type III SS, including the p-values from an F-test.

17.2 Type II and III SS Using the **car Package**

A somewhat easier way to obtain type II and III SS is through the car package. This defines a new function, Anova() (note the capital “A”), which can calculate type II and III SS directly.

17.2.1 Type II, using the same data set defined above:

```
Anova(lm(time ~  topic * sys, data=search), type=2)
```

17.2.2 Type III:

```
Anova(lm(time ~  topic * sys, data=search, contrasts=list(topic=contr.sum, sys=con
tr.sum)), type=3)
```

Due to the way in which the SS are calculated when incorporating the interaction effect, for type III you must specify the contrasts option to obtain sensible results (an explanation is given here).