Análisis de la varianza (ANOVA) para comparación de múltiples medias: teoría y práctica

Pablo Vinuesa, Centro de Ciencias Genómicas - UNAM.
http://www.ccg.unam.mx/~vinuesa/

v1, 1 de Agosto, 2018

Contents

1 Preparación del ambiente **2**
1.1 Carguemos los paquetes a usar en esta sesión . . . 2
1.2 Guardemos los parámetros gráficos originales en opar . . . 2

2 Análisis de la varianza (ANOVA): teoría y práctica **3**
2.1 Introducción: el concepto y ámbito de aplicación del análisis de la varianza . . . 3
2.1.1 ANOVA y Diseños experimentales: nomenclatura . . . 3
2.1.2 Sintaxis de fórmulas para ajuste de modelos ANOVA a diversos diseños experimentales 4
2.1.2.1 Símbolos y sintaxis de *formulas para ANOVA* . . . 4
2.1.2.2 Fórmulas para ANOVA de diseños experimentales comunes . . . 4
2.1.2.3 Sobre la importancia del órden de los términos en las fórmulas para ANOVA 4
2.1.3 Conceptos previos básicos - Varianza . . . 5
2.2 ANOVA de una vía - desarrollo gráfico y numérico del concepto . . . 6
2.2.1 Cálculo manual de las tablas de ANOVA de una vía y su interpretación gŕafica . . . 6
2.2.1.1 Particionado de la suma de cuadrados total SSY y la determinación de la significancia el análisis de la varianza . . . 8
2.2.2 Cómputo de ANOVA simple en R con aov() . . . 12
2.2.3 Validación de supuestos . . . 13
2.2.3.1 Análisis gráfico con *plot*(*aov*()) . . . 13
2.2.3.2 Validación de supuestos *homogeneidad de las varianzas* entre grupos con el **test de Levene** . . . 14
2.2.3.3 Validación de supuesto de *normalidad* de la variable de respuesta **test de Shapiro-Wilk** . . . 14
2.3 ANOVA de una vía - casos reales . . . 15
2.3.1 Datos que no violan los supuestos: *PlantGrowth* . . . 15
2.3.1.1 ¿Cómo reportar los resultados de una ANOVA de una vía? . . . 17
2.3.1.2 ANOVA de una vía como un caso particular de regresión lineal: tamaño de los efectos . . . 17
2.3.1.3 Graficado de resultados y pruebas post-hoc . . . 18
2.3.2 Alternativas al ANOVA cuando se violan los supuestos: . . . 22
2.3.2.1 El paquete *ggpubr* . . . 25
2.4 Experimentos factoriales - ANOVA de doble vía . . . 27
2.4.1 Estadísticas de resumen mediante *tapply*() y gráficos con *barplot*() (R base) . . . 27
2.4.1.1 *tapply*() . . . 27
2.4.1.2 *barplot* . . . 28
2.4.2 Análisis exploratorio usando ahora *dplyr*() y *ggplot*() . . . 29
2.4.2.1 Tabla de múltiples estadísticos de resumen con *dplyr* . . . 29
2.4.2.2 Gráficas de barras paralelas con 2 variables de agrupamiento con *ggplot*() . . 30
2.4.3 ANOVA de doble vía: experimento factorial completo . . . 30
2.4.3.1 Evaluación del tamaño de los efectos de cada nivel de cada factor . . . 31
2.4.3.2 Simplificación del modelo . . . 32

2.4.4 ANOVA de doble vía con interacciones . . . 35
2.4.4.1 Visualización de interacciones entre factores . . . 36

3 Funciones y paquetes de R usados para este documento 38
3.1 **Paquetes y software para investigación reproducible y generación de documentos en múltiples formatos** . . . 38
3.2 **Paquetes de uso general para procesamiento y graficado de datos** . . . 38
3.3 **Análisis de la varianza - ANOVA** . . . 38
3.3.1 **Funciones de paquetes base (R Core Team 2018)** . . . 38
3.3.2 **Datos del paquetes base** . . . 39
3.3.3 **Paquetes especializados** . . . 39

4 Recursos en línea 40
4.1 **The comprehensive R archive network (CRAN)** . . . 40
4.2 **Cursos** . . . 40
4.3 **Consulta** . . . 40
4.4 **Manipulación y graficado de datos con paquetes especializados** . . . 40

Referencias 40

1 Preparación del ambiente

1.1 Carguemos los paquetes a usar en esta sesión

```
# ipak function: install and load multiple R packages.
# check to see if packages are installed. Install them if they are not, then load them into the R sessi

ipak <- function(pkg){
    new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
    if (length(new.pkg))
        install.packages(new.pkg, dependencies = TRUE)
    sapply(pkg, require, character.only = TRUE)
}

# usage
packages <- c("ggplot2", "dplyr", "car", "gplots", "multcomp", "PMCMR", "ggpubr", "HH")
ipak(packages)
```

```
##  ggplot2    dplyr      car   gplots multcomp    PMCMR   ggpubr       HH
##     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE
```

1.2 Guardemos los parámetros gráficos originales en opar

```
# guardemos los parámetros gráficos originales en opar
opar <- par(no.readonly = TRUE)
```

2 Análisis de la varianza (ANOVA): teoría y práctica

Este tema es parte del **Taller 2 - Análisis exploratorio y estadístico de datos biológicos usando R**, de la Universidad Nacional Autónoma de México, impartido entre 30 de Julio y 3 de Agosto de 2018 en el Centro de Ciencias Genómicas. Para más información consultar la página del taller en: http://congresos.nnb.unam.mx/TIB2018/t2-analisis-exploratorio-y-estadistico-de-datos-biologicos-usando-r/.

La parte teórica está basada en (Crawley 2012), (Crawley 2015), (A. P. Field, Miles, and Field 2012) y (Kabacoff 2015).

2.1 Introducción: el concepto y ámbito de aplicación del análisis de la varianza

El **análisis de la varianza o ANOVA** engloba a un conjunto de métodos estadísticos que usamos cuando la **variable de respuesta** es **contínua** pero **la(s) variable(s) explicativa(s) o independiente(s)** es (son) **categórica(s)**. Las variables explicativas categóricas se conocen como **factores**, y cada factor tiene dos o más **niveles**. Si tenemos una sola variable explicativa hablamos de **análisis de la varianza de una vía (1-way ANOVA)**. Podemos considerar por tanto a la ANOVA de 1 vía como una genrealización de la *prueba t de Student*, cuando tenemos 3 o más niveles de la variable categórica y **queremos comparar 3 o más medias**. Al igual que en la *prueba t de Student*, estamos interesados en analizar cómo se comporta la media de la variable de respuesta (numérica, contínua) en función de los diversos niveles de la variable categórica de agrupamiento. Si tenemos más de una variable explicativa categórica, hablamos de **ANOVA de 2 o más vías**. Si el diseño exprimental comprende réplicas en cada nivel de una ANOVA multivía, el diseño corresponde a **diseño factorial**, en cuyo caso podremos estudiar **interacciones entre variables**, con el fin de establecer si la respuesta a un factor depende a su vez del nivel de algún otro factor o variable explicativa.

2.1.1 ANOVA y Diseños experimentales: nomenclatura

El diseño experimental en general, y el análisis de la varianza en particular, tienen un lenguaje propio interrelacionado. Es un tópico vasto libros y paquetes de R dedicados a ello. Un buen sitio para profundizar es este curso sobre ANOVA and experimental designs. Aquí sólo un breve resumen de conceptos y terminología claves adicionales a los arriba mencionados

- Diseños. pueden ser *balanceados* (iqual número de réplicas por tratamiento/nivel del factor) o *desbalanceados*
- Contrastes. Pueden ser *entre grupos* o *intra-grupo*, en este último caso generalmente asociado a *ANOVA de medidas repetidas* a lo largo del tiempo sobre un mismo individuo.
- Diseños fatctoriales. Cuando evaluamos los efectos sobre la variable de respuesta de dos o más factores, cada uno con dos o más niveles. Los *efectos principales* son los que podemos asignar a cada factor, mientas que los *efectos de interacción* se deben a las interacciones entre los niveles de cada factor analizado
- Modelos mixtos. Cuando en un diseño factorial evaluamos tanto contrastes *inter-grupo* como *intra-grupo*
- ANCOVA o análisis de la co-varianza. Cuando tenemos una segunda variable cuantitativa contínua (no un factor!) que pudiera afectar a las diferencias entre grupos de la variable de respuesta, se la considera una *variable o factor confusor*
- MANOVA o análisis multivariado de la varianza. Cuando tenemos más de una variable dependiente. Si además tenemos co-variables, tendríamos una *análisis multivariado de co-varianza* o *MANCOVA*.

Aquí veremos sólo algunos de estos casos o modelos.

2.1.2 Sintaxis de fórmulas para ajuste de modelos ANOVA a diversos diseños experimentales

Históricamente, si bien el *análisis de la varianza* se desarrolló independientemente de las metodologías de *regresión*, ambos representan casos especiales del **modelo lineal**.

Por tanto, como veremos, podemos usar tanto la función $aov()$ como $lm()$ para ajustar y analizar modelos de ANOVA. Nos enfocaremos primero en $aov()$, que reporta la salida en un formato "clásico", basado en la *tabla de ANOVA*

La sintaxis básica para llamar a $aov()$ es:

$$aov(formula,\ data = dataframe)$$

2.1.2.1 Símbolos y sintaxis de $formulas\ para\ ANOVA$

Símbolo	uso
~	Separa la variable de respuesta a la izquierda de las variables explicativas o independientea a la derecha. Ejemplo: $y \tilde{} A + B$. Predicción de respuesta de y en función de los factores A y B.
+	Separa los factores, como en $y \tilde{} A + B + C$
:	Denota interacción entre variables en un diseño factorial, como en $y \tilde{} A + B + A : B$
*	Denota el cruzamiento (interacciones) total entre variables. $y \tilde{} A * B * C$ expande a $y \tilde{} A + B + A : B + A : C + B : C + A : B : C$
^	Denota cruzamiento hasta un determinado nivel de interacciones. $y \tilde{} (A * B * C)^2$ expande a $y \tilde{} A + B + A : B + A : C + B : C$
.	Denota el resto de las variables o todas las variables. $y \tilde{}$. expande a $y \tilde{} A + B + C$

2.1.2.2 Fórmulas para ANOVA de diseños experimentales comunes

Diseño	Fórmula
ANOVA de una vía	$y \tilde{} A$
ANCOVA de una vía con una covariable x	$y \tilde{} x + A$
ANOVA factorial de dos vías	$y \tilde{} A * B$
ANCOVA de dos vías con dos covariables x_1, x_2	$y \tilde{} x_1 + x_2 + A * B$
ANOVA de una vía intra-grupo	$y \tilde{} A + Error(Sujeto/A)$

2.1.2.3 Sobre la importancia del órden de los términos en las fórmulas para ANOVA

El orden importa si: 1. hay más de un factor y el diseño es desbalanceado 2. hay covariables

Cuando se cumple cualquiera de estas condiciones, las variables en el lado derecho de la fórmula estarán correlacionadas, por lo que no hay manera unívoca de dividir su impacto sobre la variable dependiente. Es decir, en una ANOVA de doble vía con número desigual de observaciones en las combinaciones de tratamientos, el modelo $y \tilde{} A * B$ dará el mismo resultado que $y \tilde{} B * A$.

Por defecto, R usa la *aproximación secuencial* ("Type I") al cálculo de efectos en ANOVA, en el que son ajustados para aquellos que aparecen primero (secuencialmente) en la fórmula. Es decir, A está desajustado. B se ajusta por A. La interacción $A : B$ es ajustada por A y por B

A mayor desbalance, mayor impacto del órden de los términos en el resultado. En general, los efectos más importantes deben listarse primero. En particular, si hay co-variables, éstas deben ir primero, seguidas de los

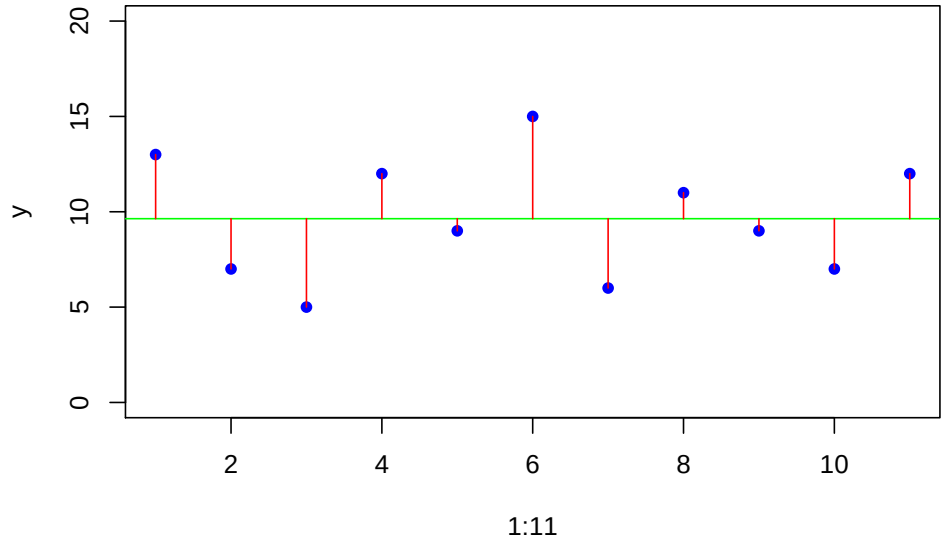
efectos principales, seguido de las interacciones entre pares, luego interacciónes de tres vías, etc.

En resumen, cuando el *diseño es ortogonal*, es decir, cuando los factores y/o covariables están correlacionados, hay que tener cuidado y considerar el órden el el que se especifican los efectos.

2.1.3 Conceptos previos básicos - Varianza

Es fundamental entender el concepto de **varianza** para poder hacer una ANOVA. Recordemos que la $varianza(s^2)$ es una medida de dispersión fundamental en estadística, que representa la **desviación cuadrática media** de los datos con respecto a la media.

Veamos un ejemplo muy sencillo: la *media* y los *residuos* asociados a 11 valores de una variable:



Formalmente, la varianza se calcula dividiendo la suma de cuadrados $sum(y_i - \bar{y})^2$ entre los **grados de libertad = g.l.**.

Estos corresponden al número de parámetros libres a estimar. Para la media $g.l. = (n-1)$, ya que la media es un parámetro estimado de los datos, por lo cual perdemos 1 g.l.

$$Varianza(s^2) = \frac{\sum(y_i - \bar{y})^2}{N-1} = \frac{\sum(y_i - \bar{y})(yi - \bar{y})}{N-1}$$

Para nuestro ejemplo, tenemos por tanto que:

```
y <- c(13,7,5,12,9,15,6,11,9,7,12)
media <- mean(y)
n <- length(y)
cat("media = ", media, "; n = ", n, "\n")
```

```
## media =  9.636364 ; n =  11
```

```
# suma de cuadrados:
SS <- sum((y-mean(y))^2)
cat("SS = ", SS, "\n")
```

```
## SS =  102.5455
```

```
# y la varianza
varianza <- SS/(n-1)
```

```r
cat("varianza = ", varianza, "\n")
```

```
## varianza =  10.25455
```

```r
# usando var() obtenemos el mismo valor ;)
var(y)
```

```
## [1] 10.25455
```

2.2 ANOVA de una vía - desarrollo gráfico y numérico del concepto

Al igual que en la prueba t de Student, el objetivo del ANOVA de 1 vía es determinar si existen diferencias significativas entre las medias de cada nivel de la variable explicativa. Ello se determina calculando las **sumas de cuadrados** SS_i asociadas a cada nivel. Si son más pequeñas que la suma de cuadrados del promedio global SSY, la diferencia entre medias es significativa.

2.2.1 Cálculo manual de las tablas de ANOVA de una vía y su interpretación gŕafica

Esto se puede ilustar gráficamente. Para simplificar al máximo, vamos a usar un factor con dos niveles nada más. Supongamos que tenemos las medidas de concentración de ozono (partes por 100 millones, ppcm) de dos invernaderos (A y B) productores de lechugas.

Antes de proceder, veamos la estructura del archivo de entrada: nótese que tenemos 1 variable contínua *ozone* y otra categórica *garden* para definir grupos.

```r
# leamos el archivo directamente desde una url
unavia <- read.csv("https://math.la.asu.edu/~coombs/ozone.txt", sep ="\t")

# veamos la estructura del dfr unavia
str(unavia)
```

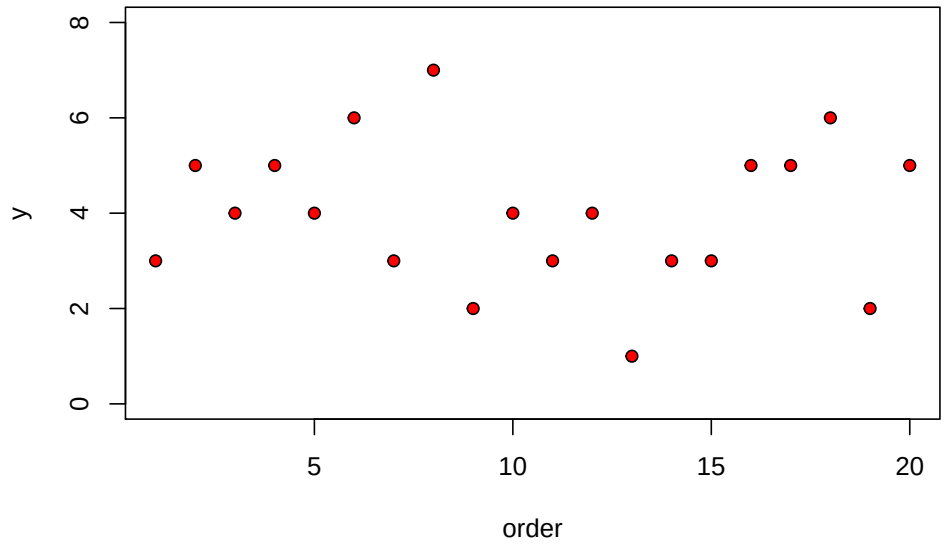
```
## 'data.frame':    20 obs. of  2 variables:
##  $ ozone : int  3 5 4 5 4 6 3 7 2 4 ...
##  $ garden: Factor w/ 2 levels "A","B": 1 2 1 2 1 2 1 2 1 2 ...
```

```r
# usando attach, añadimos los objetos creados por R directamente al "search path"
attach(unavia)

# veamos la cabecera de la tabla
head(unavia)
```

```
##   ozone garden
## 1     3      A
## 2     5      B
## 3     4      A
## 4     5      B
## 5     4      A
## 6     6      B
```

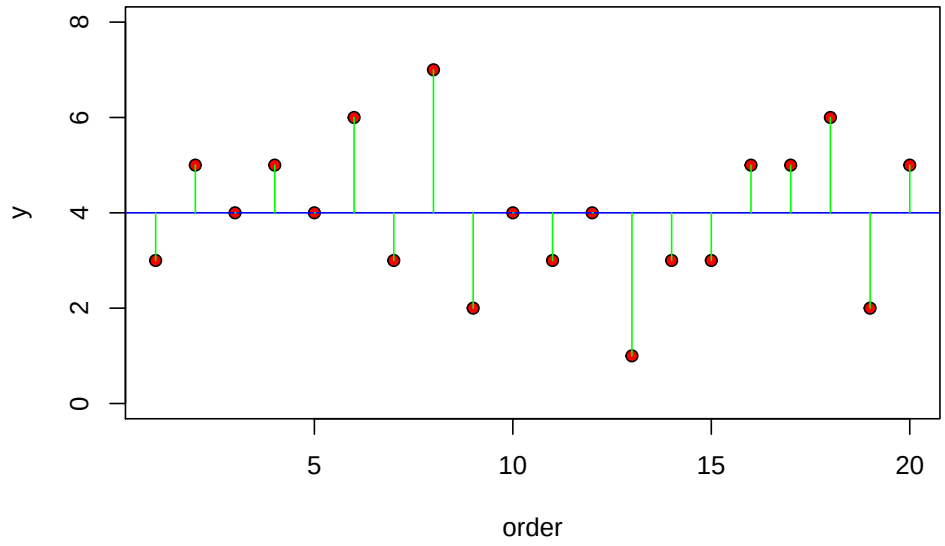
Ahora la gŕafica de dispersíon.

```r
# hagamos un gráfico de dispersión
plot(1:20,ozone,ylim=c(0,8),ylab="y",xlab="order",pch=21,bg="red")
```



Vemos una dispersión notable, indicano que la varianza total de la variable de respuesta y es grande. Visualizemos los resíduos para tener una mejor apreciación del nivel de varianza global en los datos.

```
plot(1:20,ozone,ylim=c(0,8),ylab="y",xlab="order",pch=21,bg="red")
abline(h=mean(ozone),col="blue")
for(i in 1:20) lines(c(i,i),c(mean(ozone),ozone[i]),col="green")
```



A esta dispersión global es a lo que llamamos **suma de cuadrados total** SSY

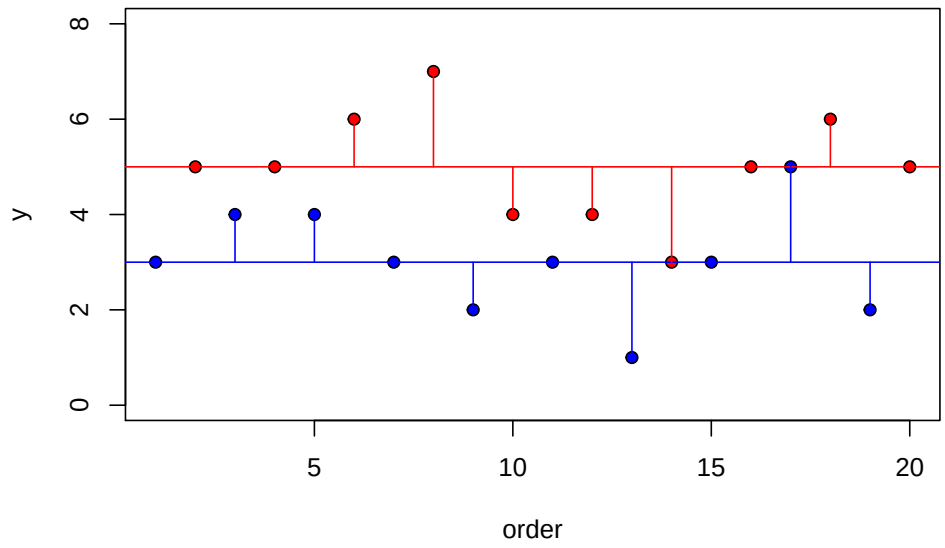
$$SSY = \sum (y_i - \bar{y})^2$$

.

Ahora vamos a ajustar la media correspondiente a cada nivel y visualizar las desviaciones de los datos con respecto a la media del tratamiento o nivel correspondiente.

```
plot(1:20,ozone,ylim=c(0,8),ylab="y",xlab="order",pch=21,bg=c("blue", "red"))
abline(h=mean(ozone[garden == "A"]), col = "blue")
abline(h=mean(ozone[garden == "B"]), col = "red")

# y ahora visualizemos los resíduos con respecto de las medias de los tratamientos
```

```r
# respectivos

index <- 1:length(ozone)
for (i in 1:length(index)){
if (garden[i] == "A" )
     lines(c(index[i],index[i]),c(mean(ozone[garden=="A"]),ozone[i]), col = "blue")
else
     lines(c(index[i],index[i]),c(mean(ozone[garden=="B"]),ozone[i]), col="red")
}
```



Nótese que si las medias de los tratamientos fueran iguales, entonces las líneas rojas y azules estarían en el mismo lugar, y por tanto la longitud de las líneas residuales sería la misma que en la figura anterior. En cambio, si las medias fueran diferentes, las líneas residuales de los tramientos individuales serían menores que cuando las calculamos a partir de la muestra global. Es decir **cuando las medias entre tratamientos son significativamente diferentes, las SSi será menor que la SSY calculada de la muestra global**.

La significancia de la diferencia entre estas dos SSs se juzga mediante el análisis de la varianza. Para ello necesitamos calcular la **suma de cuadrado de los errores o desviaciones** SSE, que corresponde a la la suma de los cuadrados de las longitudes de las barras rojas más la suma de los cuadrados de las longitudes de las barras azules.

$$SSE = \sum_{j=1}^{k} \sum (y - \bar{y_j})^2$$

Es decir, calculamos la media para el nivel j_i del factor, y sumamos seguidamente los cuadrados de las diferencias. La $SSE = varianza\ no\ explicada$.

¿Cuántos grados de libertad están asociados al SSE? En nuestro caso tenemos $n = 10$ réplicas para cada uno de los $k = 2$ tratamientos ($k * n$ números en total). Como estimamos k parámetros (medias en este caso), habremos perdido k grados de libertad en el proceso, y los $g.l. = kn - k = k(n - 1)$.

Visto de otra manera, si tenemos n réplicas en cada tratamiento, y por tanto $(n - 1)$ grados de libertad para el error en cada uno (ya que perdemos 1 g.l. al estimar la media de cada tratamiento: $\bar{y_i}$). Por lo tanto, para k tratamientos (k niveles del factor), tenemos $g.l. = k(n - 1)$ para el error en el experimento global.

2.2.1.1 Particionado de la suma de cuadrados total SSY y la determinación de la significancia el análisis de la varianza

La suma de cuadrados total SSY es particionada en sus componentes: la variación no explicada o error SSE y la **suma de cuadrados de los tratamientos** SSA. Es decir $SSY = SSA + SSE$. Dado que podemos calcular SSY y SSE con las fórmulas arriba presentadas, calculamos SSA (la variación explicada por las diferencias entre medias de los tratamientos) así:

$$SSA = SSY - SSE$$

- Cálculo de SSY

```
SSY <- sum((ozone - mean(ozone))^2)
SSY
```

```
## [1] 44
```

- Cálculo de SSE Es la suma de cuadrados de los residuos calculada por separado para cada tratamiento, usando la media correspondiente.

Para el invernadero A:

```
ssa <- sum((ozone[garden=="A"] - mean(ozone[garden=="A"]))^2)
cat("SS para A: ", ssa)
```

```
## SS para A:  12
```

Para el invernadero B:

```
ssb <- sum((ozone[garden=="B"] - mean(ozone[garden=="B"]))^2)
cat("SS para B: ", ssb)
```

```
## SS para B:  12
```

Por tanto: $SSE = 12 + 12 = 24$ y $SSA = 44 - 24 = 20$.

Ya tenemos los dos *componentes* (SSA y SSE) en los que se descompone la variación total de los datos (SSY). Hagamos un *análisis de la varianza*, la famosa **ANOVA**, para comparar estos dos componentes mediante un cociente F entre las varianzas asociadas a cada componente.

Estadísticos como F representan generalmente la cantidad de *varianza sistemática (del modelo o tratamienots) / varianza no sistemática (error de los datos)*, es decir F se basa en la razón de SSA/SSE. Pero como las sumas de cuadrados dependen del número de puntos (diferencias sumadas), F usa la *media de la suma de cuadrados*, $MS = promedio\ de\ suma\ de\ cuadrados$.

$$F = \frac{MS_M}{MS_E} = \frac{media\ de\ los\ cuadrados\ del\ modelo}{media\ cuadrados\ de\ los\ errores\ o\ resíduos}$$

donde

$$MS_M = \frac{SSR}{g.l.} = \frac{SSR}{no.\ parámetros\ estimados\ en\ el\ modelo:\ b\ (1\ g.l.\ en\ ml\ simple)}$$

y

$$MS_E = \frac{SSE}{g.l.} = \frac{SSE}{no.\ observaciones - no.\ parámetros\ libres:\ a,\ b\ (n-2)}$$

Construyamos la **tabla de ANOVA** correspondiente, paso a paso:

Ya tenemos calculados las sumas de cuadrados de cada fuente de variación, como se muestra abajo.

Fuente	Suma de cuadrados	Grados de libertad	Media de cuadrados (Varianza)	cociente F
Tratamiento	20 (SSA)			

Fuente	Suma de cuadrados	Grados de libertad	Media de cuadrados (Varianza)	cociente F
Error (resíduos)	24 (SSE)			
Total (datos)	44 (SSY)			

Llenemos ahora la columna de los *grados de libartad* ($g.l.$). Recordemos que, esencialmente, los **grados de libartad** representan el número de 'entidades' que están libres de variar cuando estimamos algún parámetro estadístico a partir de los datos. Los $g.l$ generalmente se calculan como $n - número\ de\ parámetros\ libres\ a\ estimar\ de\ los\ datos$. Por tanto:

- $g.l.(SSA) = k - 1 = 1$, ya que hay dos niveles del factor: A y B.
- $g.l.(SSE)$, dado que hay $n = 10$ replicas por nivel, tenemos que $g.l. = 10 - 1)$ por invernadero, por tanto $g.l.\ para\ SSE = 2\ x\ 9 = 18 = k(n-1)$.
- $g.l.(SSY) = n - 1$ del conjunto global de datos hemos estimado sólo un parámetro \$(n = 20), la media global $\bar{y}$ $g.l. = 20 - 1 = 19$

Noten que la suma de $g.l.(SSR) + g.l.(SSE) = g.l.(SSY)$, como se ve en la tabla.

Fuente	Suma de cuadrados	Grados de libertad	Media de cuadrados (Varianza)	cociente F
Tratamiento	20 (SSA)	1 ($k-1$)		
Error (resíduos)	24 (SSE)	18 ($k(n-1)$)		
Total (datos)	44 (SSY)	19 ($n-1$)		

Necesitamos llenar la 4a. columna, reservada para las varianzas. Recoredemos que:

$$var = s^2 = \frac{suma\ de\ cuadrados}{grados\ de\ libertad}$$

Esta columna (4) es muy fácil de llenar, ya que tenemos los valores que necesitamos pre-calculados en las columnas 2 y 3. La *var total* no nos hace falta, pero sí el $cociente - F = \frac{20}{1.333}$, que incluímos tambien en la **tabla de ANOVA**, completándola.

Fuente	Suma de cuadrados	Grados de libertad	Media de cuadrados (Varianza)	cociente F
Tratamiento	20 (SSA)	1 ($k-1$)	$s^2 = 20$ (20 / 1)	20/1.333 = 15.0
Error (resíduos)	24 (SSE)	18 ($k(n-1)$)	$s^2 = 1.333$ (24 /18)	
Total (datos)	44 (SSY)	19 ($n-1$)		

Calculemos ahora la significancia del $cociente - F$. Recordemos que:

- $H_0 : b = 0$, la pendiente de la recta de regresión es cero (el modelo no difiere de la media)
- $H_1 : b \neq 0$, usando una alternativa de doble cola asumimos que la pendiente es positiva o negativa

Podemos buscar el **valor crítico de** F en la tabla correspondiente, usando $g.l._{numerador} = 1$ y $g.l._{denominador} = 18$ y veríamos algo como ésto:

$g.l.(denom)$	p	**1** ... $(g.l.(num.))$
1	0.05	161.45
1	0.01	4052.18

$g.l.(denom)$	p	**1** ... $(g.l.(num.))$
...	...	...
18	0.05	**4.41**
18	0.01	**8.29**

Dado que nuestro estadístico $F = 15$ es mucho mayor que el valor crítico de $F_{(1,18)}$, rechazamos la hipótesis nula con $p < .01$.

Pero estamos *aprendiendo estadística usando R*, ¿verdad? ¿O cargas contigo las tablas de valores críticos para múltiples estadísticos?

Recordemos que los **intervalos de confianza** se definen como *quantiles* del 95% ó 99%.

Podemos calcular los quantiles para F usando la función $qf(.95, glNum, glDenom)$ o $qf(.99, glNum, glDenom)$. La p asociada al estadístico F la calculamos como $1-pf(F, glNum, glDenom)$, como se muestra seguidamente:

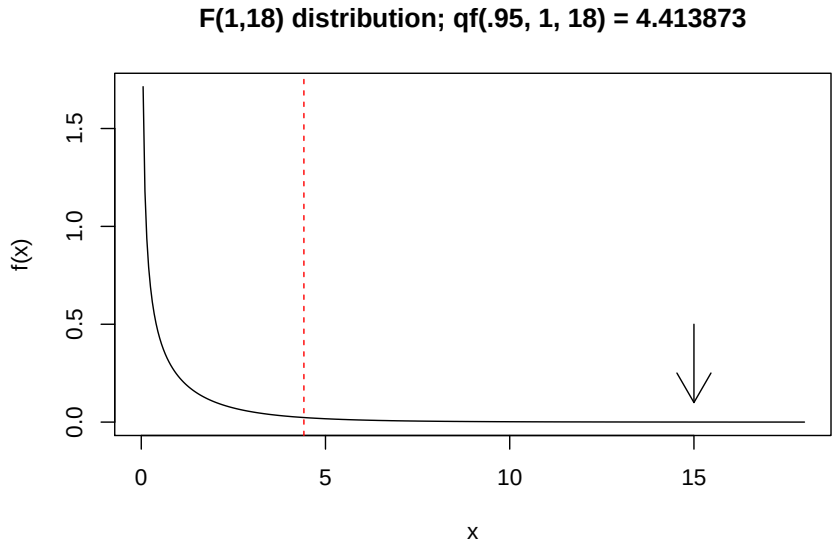
```
glNum <- 1
glDenom <- 18
Fratio <- 15

CI95_F <- qf(.95, glNum, glDenom)
CI99_F <- qf(.99, glNum, glDenom)

p <- 1-pf(Fratio,glNum,glDenom)

cat("CI95% = ", CI95_F, "| CI99% = ", CI99_F, " | p = ", p, "\n")
```

```
## CI95% =  4.413873 | CI99% =  8.28542  | p =  0.001114539
```

Por tanto, la probabilidad de observar un *cociente* $-$ F tan extremo (o más) como el observado ($F = 15$) si ambas medias fueran realmente iguales es de $\sim 0.1\%$. Es decir, observar esta F bajo la H_0 de no diferencia entre medias es muy poco probable, de hecho $p < 0.01$, por lo que rechazamos H_0 contundentemente.

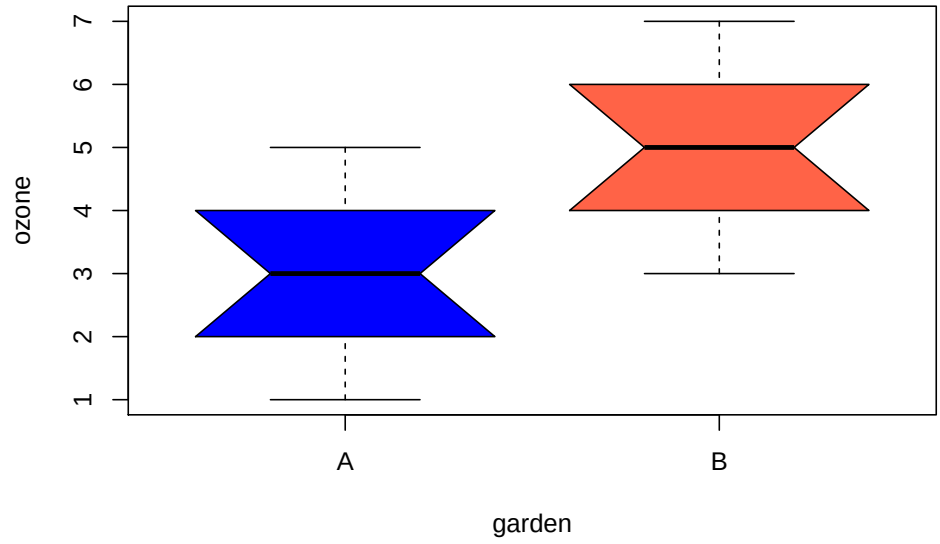
- Gráficamente, lo podemos representar con este código:

```
x <- seq(0, 18, .05)
plot(x, df(x,1,18), type="l", ylab="f(x)", xlab="x",
     main = "F(1,18) distribution; qf(.95, 1, 18) = 4.413873")
abline(v = qf(.95,1,18), col = "red", lty = 2)
arrows(15,0.5, 15, .1)
```



- Podemos además generar unos boxplots para visualizar las diferencias entre tratamientos.

```
plot(ozone~garden, notch = TRUE, col = c("blue", "tomato"))
```



Noten el uso de $notch = TRUE$.

Cuando los *notches* no solapan, las diferencias en las medianas son probablemente significativas

$$notch = \pm 1.58 \frac{IQR}{\sqrt{n}}$$

2.2.2 Cómputo de ANOVA simple en R con aov()

R hace muy fácil este cálculo usando la función stats::aov() del paquete base stats, ahorrándonos mucho trabajo. La sintaxis básica para usar la función es $aov(formula,\ data = dataframe)$ Veremos más adelante algunos detalles sobre $formulas$ en R. Para la ANOVA de 1 vía la fórmula es $y \sim A$, donde y es la variable dependiente y A el factor, como muestra la siguiente llamada:

```
# var_dep ~ factor
summary(aov(ozone ~ garden))
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## garden       1     20  20.000      15 0.00111 **
## Residuals   18     24   1.333
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

summary(*aov*()) imprime la tabla de ANOVA. Como ven, los resultado son iguales a los que calculamos a mano.

Recordemos que la *prueba t de Student* es equivalente al ANOVA de una vía cuando el factor tiene dos niveles. Podemos comprobarlo fácilmente con el siguiente código:

```
t.test(ozone[garden=="A"],ozone[garden=="B"])
```

```
##
##  Welch Two Sample t-test
##
## data:  ozone[garden == "A"] and ozone[garden == "B"]
## t = -3.873, df = 18, p-value = 0.001115
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.0849115 -0.9150885
## sample estimates:
## mean of x mean of y
##         3         5
```

2.2.3 Validación de supuestos

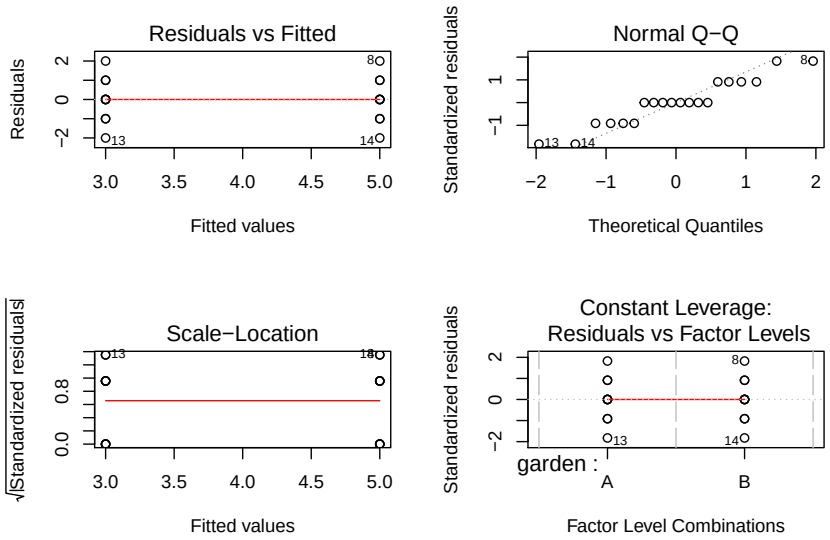
Es crítico recordar que todos los tests estadísticos que hacen uso de la varianza se basan en **dos supuestos**:

1. la **constancia u homogeneidad de varianzas entre tratamientos (niveles)**
2. **normalidad de los errores**

2.2.3.1 Análisis gráfico con *plot*(*aov*())

Podemos checarlos gráficamente con el siguiente código:

```
par(mfrow=c(2,2))
plot(aov(ozone~garden))
```




```
par(opar)
```

El primer gráfico muestra que las varianzas son iguales en los dos tratamientos, que es lo ideal. El segundo muestra una relación razonablemente lineal en plot de quantiles-quantiles normales, indicando que la no-normalidad de los errores no es un problema en este caso. El tercero muestra los residuos contra los valores ajustados a una escala diferente, indicando nuevamente varianza constante. El cuarto muestra las distancias de Cook, indicando que los puntos 8, 13 y 14 tienen residuos notoriamente grandes.

2.2.3.2 Validación de supuestos *homogeneidad de las varianzas* entre grupos con el test de Levene

La sintaxis básica del test, según se implemente en el paqute *car* es: $car :: leveneTest(value \tilde{} variable, dataset)$

```
#  vean las opciones del test ?car::leveneTest
car::leveneTest(ozone ~ garden, data = unavia)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1       0      1
##       18
```

En consonancia con lo visto en el primer gráfico de arriba, las varianzas son iguales. Esto se refleja en la $\Pr(>F) = 1$

2.2.3.3 Validación de supuesto de *normalidad* de la variable de respuesta test de Shapiro-Wilk

La sintaxis básica es: shapiro.test(x), donde x es un vector de datos numéricos

```
# vean las opciones del test con ?shapiro.test o help("shapiro.test")
shapiro.test(unavia$ozone)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  unavia$ozone
## W = 0.96508, p-value = 0.6495
```

```
# ya no vamos ausar mas el data frame 'unavia'. Conviene eliminarlo del search() path con detach()
detach(unavia)
```

Con una $p - value = 0.6495$ obviamente no podemos rechazar la H_0 de normalidad. Por tanto, no hay problemas obvios en nuestros los datos.

2.3 ANOVA de una vía - casos reales

2.3.1 Datos que no violan los supuestos: *PlantGrowth*

Vamos a usar ahora el data frame *PlantGrowth* que viene con la distribución base de R y contiene los resultados de un experimento diseñado para comparar los rendimientos (medidos como pesos secos) de plantas cultivadas bajo una condición control y dos tratamientos. Es un *diseño balanceado*, ya que cada tratamiento contiene 10 observaciones. Como sólo tenemos una variable contínua y una categórica, se trata de un *análisis de la varianza de una vía*. Como vamos a comparar entre grupos, se trata de un *análisis inter-grupal de la varianza de una vía, con diseño balanceado*.

Carguemos y exploremos la estructura de los datos:

```
pg_dfr <-  PlantGrowth
class(pg_dfr)
```

```
## [1] "data.frame"
```

```
dim(pg_dfr)
```

```
## [1] 30  2
```

```
names(pg_dfr)
```

```
## [1] "weight" "group"
```

```
head(pg_dfr)
```

```
##   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
```

```
str(pg_dfr)
```

```
## 'data.frame':    30 obs. of  2 variables:
##  $ weight: num  4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...
##  $ group : Factor w/ 3 levels "ctrl","trt1",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
levels(pg_dfr$group)
```

```
## [1] "ctrl" "trt1" "trt2"
```

Generemos unas estadísticas descriptivas de los datos para tener una primera impresión acerca de medidas de tendencia central y dispersión.

```
#pg_dfr$group <- ordered(pg_dfr$group,
#                        levels = c("ctrl", "trt1", "trt2"))
# paquete dplyr
group_by(pg_dfr, group) %>%
```

```
  summarise(
    count = n(),
    media = mean(weight, na.rm = TRUE),
    mediana = median(weight, na.rm = TRUE),
    varianza = var(weight, na.rm = TRUE)
  )
```

```
## # A tibble: 3 x 5
##   group count media mediana varianza
##   <fct> <int> <dbl>   <dbl>    <dbl>
## 1 ctrl     10  5.03    5.15    0.340
## 2 trt1     10  4.66    4.55    0.630
## 3 trt2     10  5.53    5.44    0.196
```

Vemos que las varianzas no son iguales y que las distribuciones están ligeramente sesgadas (medias y medianas no coinciden). Hagamos unas pruebas estadísticas de normalidad y homogeneidad de varianzas, para ver si son significativas o no estas desviaciones de las respectivas H_0.

- Prueba de normalidad de Shapiro-Wilk

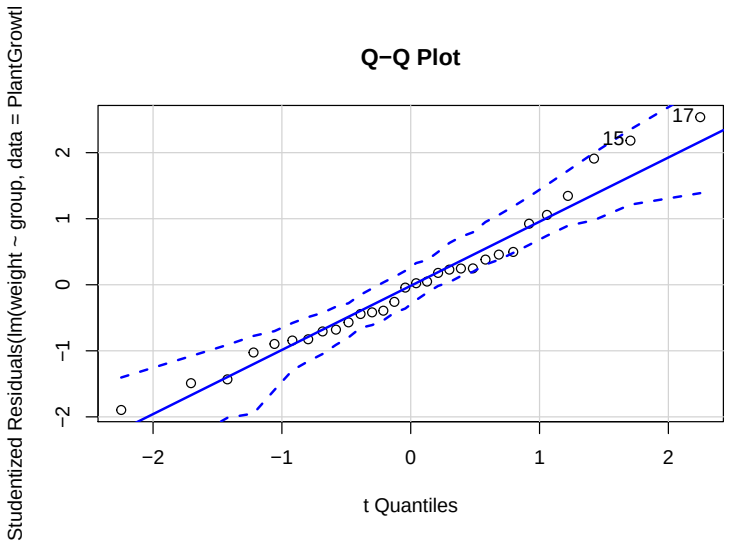
```
shapiro.test(PlantGrowth$weight)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  PlantGrowth$weight
## W = 0.98268, p-value = 0.8915
```

Prueba no significativa.

- Gráfica quantil-quantil del paquete *car* :: *qqPlot*

```
car::qqPlot(lm(weight ~ group, data = PlantGrowth), simulate = TRUE,
       main = "Q-Q Plot", labels = FALSE)
```




```
## [1] 15 17
```

Desvío mínimo de la normalidad.

- Prueba de homogeneidad de varianzas de Levene

```r
car::leveneTest(weight ~ group, data = PlantGrowth)
```

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

Ninguna de las dos pruebas es significativa, y el análisis de Q-Q muestra desviacíon mínima de la normalidad, por tanto podemos proceder al ANOVA.

- ANOVA de una vía del data frame PlantGrowth

```r
aov_pg <- aov(weight ~ group, data = PlantGrowth)
summary(aov_pg)
```

```
##             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
```

Recordemos que una ANOVA produce un *estadístico* F o *cociente* F, similar al usado en la *prueba t de Student*, que corresponde al cociente $F = \frac{modelo}{error}$ que compara la varianza sistemática en los datos a la no sistemática. La H_0 es la de no diferencia entre medias de los tratamientos. En nuestro caso la F es significativa.

2.3.1.1 ¿Cómo reportar los resultados de una ANOVA de una vía?

Obviamente debemos dar los detalles del *cociente* F y los grados de libertad a partir de los cuales fue calculado. En nuestro caso: $gl_M = 2$ y $gl_R = 27$

En conclusión: hubo un efecto significativo de los tratamientos sobre el peso seco medio de las plantas, $F(2, 27) = 4.846, p < .05$

2.3.1.2 ANOVA de una vía como un caso particular de regresión lineal: tamaño de los efectos

Generalmente es más informativo investigar los efectos de los diferentes niveles de un factor usando las funciones *summary.lm*() y *coef*() así:

```r
summary.lm(aov_pg)
```

```
##
## Call:
## aov(formula = weight ~ group, data = PlantGrowth)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0710 -0.4180 -0.0060  0.2627  1.3690
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.0320     0.1971  25.527   <2e-16 ***
## grouptrt1    -0.3710     0.2788  -1.331   0.1944
## grouptrt2     0.4940     0.2788   1.772   0.0877 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.6234 on 27 degrees of freedom
## Multiple R-squared:  0.2641, Adjusted R-squared:  0.2096
## F-statistic: 4.846 on 2 and 27 DF,  p-value: 0.01591
```

```
coef(aov_pg)
```

```
## (Intercept)   grouptrt1   grouptrt2
##       5.032      -0.371       0.494
```

El resultado de *summary.lm*() viene de evaluar el modelo $aov(y\ x)$, que es análogo al modelo lineal $y = a + bx_1 + cx_2$, donde a (Intercept) es la media, en este caso del tratamiento *control* (por convención, el primero de los niveles en orden alfabético), y $grouptrt1$ y $grouptrt2$ las diferencias de las correspondientes medias respecto al control.

El *error estandar* de *intercept* es por tanto el *error estandar* de la media:

$$SE_{\bar{y}} = \sqrt{\frac{s^2{}_a}{n_a}} = \sqrt{\frac{0.6234^2}{10}} = 0.1971$$

,

mientras que los *errors estandar* de las otras filas corresponde al de diferencia entre medias:

$$SE_{diff} = \sqrt{2\frac{s^2{}_a}{n_a}} = \sqrt{2\frac{0.6234^2}{10}} = 0.2788$$

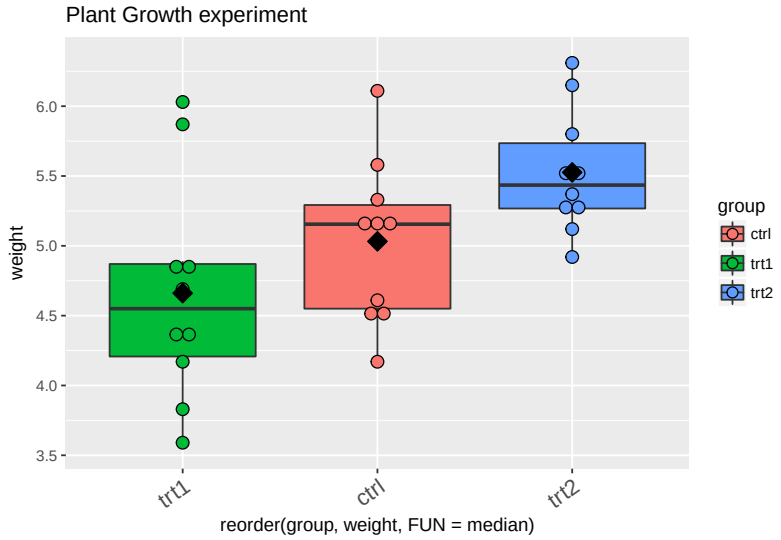
La salida de $coef()$ nos dice, en resumen, que el tratamiento control tiene asociado un rendimiento promedio de 5.032. El efecto del tratamiento 1 (trt1) es reducir el peso en -0.371, y el del tratamiento 2 en incrementarlo en 0.494 unidades por encima del control.

2.3.1.3 Graficado de resultados y pruebas post-hoc

El ANOVA es un test del efecto global de los tratamientos, y no da información sobre qué tratamientos particulares son los que tienen efectos significativos. Necesitamos graficar los resultados para descubrir cuáles tienen un efecto significativo.

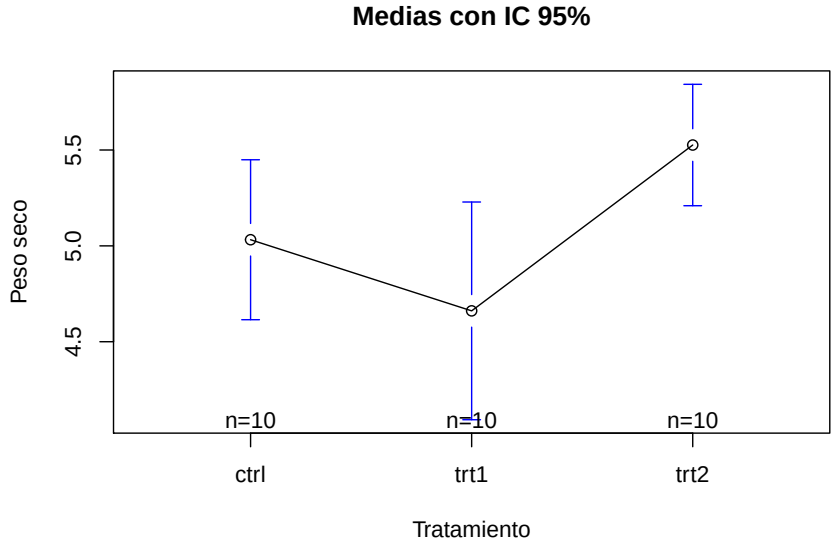
- Graficado de los datos mediante boxplots paralelos

```
# library(ggplot2)
bp <- ggplot(PlantGrowth, aes(x=reorder(group, weight, FUN = median), y=weight, fill=group)) + geom_boxp
  geom_dotplot(binaxis = "y", stackdir = "center") +
  stat_summary(fun.y = "mean", geom="point", shape=23, size=4, fill = "black")


bp + theme(axis.text.x = element_text(angle = 35, hjust = 1, vjust = 1, size = rel(1.5))) +

  ggtitle("Plant Growth experiment")
```



- Graficado de medias por tratamiento con intervalos de confianza del 95% El paquete *gplots* puede ser útil para ésto

```
# library(gplots)
gplots::plotmeans(weight ~ group, data = PlantGrowth, xlab = "Tratamiento",
          ylab = "Peso seco", main = "Medias con IC 95%")
```



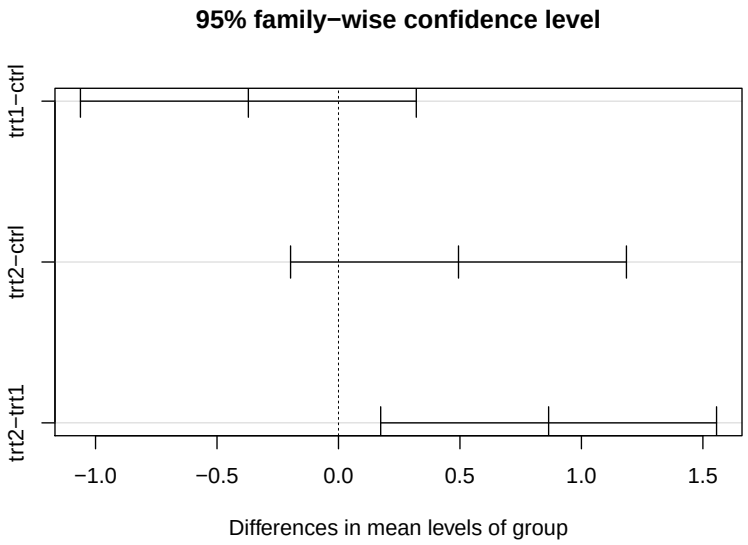
- Prueba post-hoc HSD de Tukey Esta es, sin duda, la más conveniente para determinar diferencias significativas entre tratamientos.

```
TukeyHSD(aov_pg)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = weight ~ group, data = PlantGrowth)
##
## $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(TukeyHSD(aov_pg))
```



En la gráfica, los intervalos de confianza que incluyen el 0 indican que los pares de tratamientos correspondientes no son significativamente diferentes ($p > 0.05$)

- Función *glht*() del paquete *multcomp* y evaluación con Tukey's Honestly Significant Differences HSD

El paquete *multcomp* incluye muchos métodos para el análisis de múltiples comparaciones, que pueden aplicarse tanto a modelos lineales como a modelos lineales generalizados. El siguiente ejemplo muestra cómo reproducir la anterior *prueba HSD de Tukey* pero usando un despliegue gráfico más claro. Se trata de *pruebas t* pareadas, con corrección para el múltiple testado.

Los grupos (representados por boxplots) que comparten letra no son significativamente diferentes entre ellos.

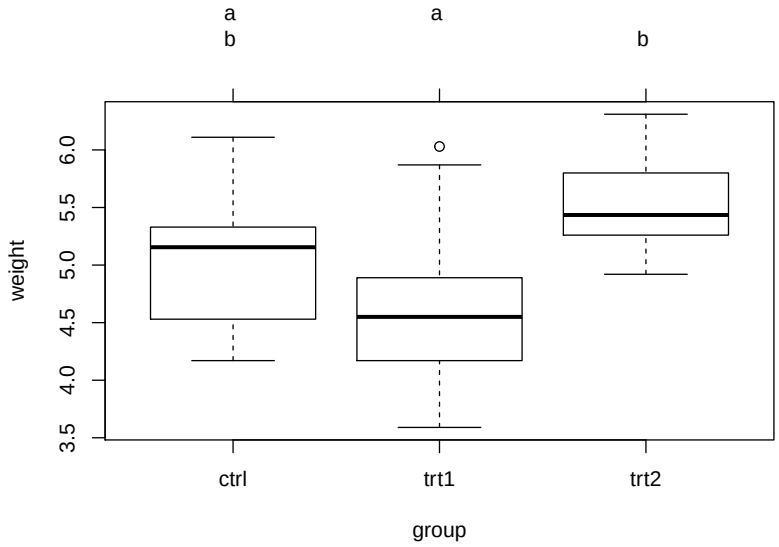
```
# test post-hoc de Tukey
tuk <- glht(aov_pg, linfct=mcp(group="Tukey"))

# resumen del test
summary(tuk)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = weight ~ group, data = PlantGrowth)
##
## Linear Hypotheses:
##                   Estimate Std. Error t value Pr(>|t|)
## trt1 - ctrl == 0  -0.3710     0.2788  -1.331   0.3909
## trt2 - ctrl == 0   0.4940     0.2788   1.772   0.1980
## trt2 - trt1 == 0   0.8650     0.2788   3.103   0.0121 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```r
# obtener los intervalos de confianza con confint()
confint(tuk)
```

```
##
##   Simultaneous Confidence Intervals
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = weight ~ group, data = PlantGrowth)
##
## Quantile = 2.4793
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##                   Estimate lwr     upr
## trt1 - ctrl == 0 -0.3710  -1.0622  0.3202
## trt2 - ctrl == 0  0.4940  -0.1972  1.1852
## trt2 - trt1 == 0  0.8650   0.1738  1.5562
```

```r
# graficado
plot(cld(tuk, level = .05))
```



Ambas representaciones indican que sólo las medias de los tratamientos 1 y 2 difieren significativamente entre ellas, pero ninguna con respecto al control.

Otra opción muy similar es usar la función *pairwise.t.test()* usando *corrección de Bonferroni*:

```r
pairwise.t.test(PlantGrowth$weight, PlantGrowth$group, p.adj="bonferroni")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  PlantGrowth$weight and PlantGrowth$group
##
##      ctrl  trt1
## trt1 0.583 -
```

```
## trt2 0.263 0.013
##
## P value adjustment method: bonferroni
```

2.3.2 Alternativas al ANOVA cuando se violan los supuestos:

Volvamos a cargar los datos *ozono* usados al inicio, y agregarle los datos correspondientes a un nuevo nivel C de la variable *garden*

```
# leamos el archivo directamente desde una url
ozono <- read.csv("https://math.la.asu.edu/~coombs/ozone.txt", sep ="\t")

# veamos la estructura del dfr ozono
str(ozono)
```

```
## 'data.frame':    20 obs. of  2 variables:
##  $ ozone : int  3 5 4 5 4 6 3 7 2 4 ...
##  $ garden: Factor w/ 2 levels "A","B": 1 2 1 2 1 2 1 2 1 2 ...
```

```
# veamos la cabecera de la tabla
head(ozono)
```

```
##   ozone garden
## 1     3      A
## 2     5      B
## 3     4      A
## 4     5      B
## 5     4      A
## 6     6      B
```

```
# vamos a agregarle los datos correspondientes a un nuevo nivel "C" de la variable garden
ozono_gC <- c(3,3,2,1,10,4,3,11,3,10)
gC <- rep("C", 10)

# agrupamos los dos vectores en un data frame, le agregamos colnames() y lo gegamos
# como nuevas filas al data frame ozono con rbind()
gC_dfr <- data.frame(ozono_gC, gC)
colnames(gC_dfr) <- c("ozone", "garden")
str(gC_dfr)
```

```
## 'data.frame':    10 obs. of  2 variables:
##  $ ozone : num  3 3 2 1 10 4 3 11 3 10
##  $ garden: Factor w/ 1 level "C": 1 1 1 1 1 1 1 1 1 1
```

```
ozono <- rbind(ozono, gC_dfr)


# veamos nuevamente la cabecera de la tabla
head(ozono)
```

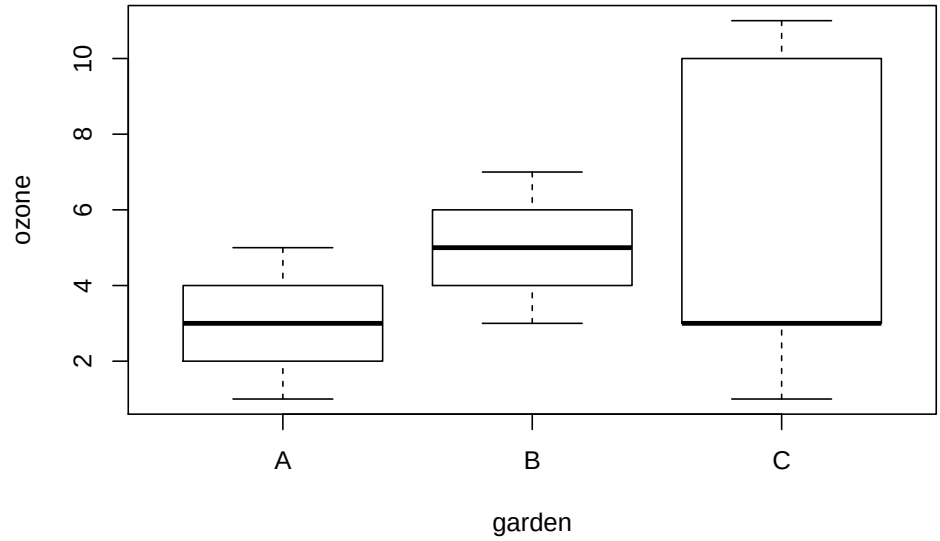
```
##   ozone garden
## 1     3      A
## 2     5      B
## 3     4      A
## 4     5      B
## 5     4      A
## 6     6      B
```

```r
tail(ozono)
```

```
##    ozone garden
## 25    10      C
## 26     4      C
## 27     3      C
## 28    11      C
## 29     3      C
## 30    10      C
```

Grafiquemos unos boxplots para resumir las distribuciones de los datos por variable de agrupación.

```r
# hagamos un gráfico de dispersión
plot(ozone ~ garden, data = ozono)
```



Generemos unas estadísticas descriptivas ...

```r
ozono %>% group_by(garden) %>% summarize(n = n(),
                                         media = mean(ozone, na.rm = TRUE),
                                         mediana = median(ozone, na.rm = TRUE),
                                         varianza = var(ozone, na.rm = TRUE)
                                         )
```

```
## # A tibble: 3 x 5
##   garden     n media mediana varianza
##   <fct>  <int> <dbl>   <dbl>    <dbl>
## 1 A         10     3       3     1.33
## 2 B         10     5       5     1.33
## 3 C         10     5       3    14.2
```

Vemos que la varinza es un órden de magnitud mayor para el tratamiento C que para los demás.

- Prueba de normalidad de Shapiro-Wilk

```r
shapiro.test(ozono$ozone)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  ozono$ozone
```

```
## W = 0.8571, p-value = 0.0008765
```

- Evaluemos la homogeneidad de varianzas entre el tratamiento B vs. C

```
gardenA <- ozono[ozono$garden == "A",]
gardenB <- ozono[ozono$garden == "B",]
gardenC <- ozono[ozono$garden == "C",]


var.test(gardenB$ozone, gardenC$ozone)
```

```
##
##  F test to compare two variances
##
## data:  gardenB$ozone and gardenC$ozone
## F = 0.09375, num df = 9, denom df = 9, p-value = 0.001624
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.02328617 0.37743695
## sample estimates:
## ratio of variances
##            0.09375
```

Vemos que las varianzas son significativamente diferentes $p < .01$. Los jardines A y B tienen diferente media, pero idéntica varianza. Pero al comparar B con C, vemos que aunque ambos tiene la misma media, tienen varianzas muy diferentes. ¿Son idénticas dos muestras con igual media? **NO**. El umbral de daño por ozono está en 8 pphm. Ambas medias son claramente menores a este umbral. No obstante, explorando los datos crudos es claro que las lechugas del invernadero C estuvieron estuvieron expuestas en 30% de los días a valores $>$ al umbral, como se muestra seguidamente:

```
ozono %>% filter(ozone > 8)
```

```
##   ozone garden
## 1    10      C
## 2    11      C
## 3    10      C
```

Por tanto, **cuando las varianzas son diferentes entre tratamientos, no deben compararse sus medias** De hacerlo, nos arriesgamos a llegar a una conclusión errónea.

Como en nuestro caso no podemos asumir tampoco normalidad, debemos usar **pruebas no paramétricas**. Dado que en nuestro caso los grupos/tratamientos son independientes, podemos usar la *prueba de* $Kruskal - Wallis$, la cual es un equivalente no paramétrico de la ANOVA de una vía.

```
# est no-parametrico de Kruskal-Walis
kruskal.test(ozone ~ garden, data = ozono)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  ozone by garden
## Kruskal-Wallis chi-squared = 7.378, df = 2, p-value = 0.025
```

Dado que el test de KW es significativo, es razonable correr ***pruebas post-hoc*** para ver qué tratamientos son significativos. Podemos hacerlo con el **Tukey y Kramer (Nemenyi) test**, o el **Dunn test**, implementados en el paquete $PMCMR$

```
# pruebas post-hoc de Nemeny y Dunn para el test no-parametrico de Kruskal-Walis
# Ver vignette con vignette("PMCMR")
```

```
PMCMR::posthoc.kruskal.nemenyi.test(x=ozono$ozone, g=ozono$garden, dist="Tukey" )
```

```
##
##  Pairwise comparisons using Tukey and Kramer (Nemenyi) test
##                    with Tukey-Dist approximation for independent samples
##
## data:  ozono$ozone and ozono$garden
##
##   A     B
## B 0.022 -
## C 0.503 0.274
##
## P value adjustment method: none
```

```
PMCMR::posthoc.kruskal.dunn.test(ozone ~ garden, data = ozono,
                                 p.adjust.method="bonferroni")
```

```
##
##  Pairwise comparisons using Dunn's-test for multiple
##                           comparisons of independent samples
##
## data:  ozone by garden
##
##   A    B
## B 0.02 -
## C 0.76 0.35
##
## P value adjustment method: bonferroni
```

Queda claro por ambos tests *post-hoc* que sólo es significativa la diferencia entre los tratamientos o niveles A-B

También podemos usar la **prueba de Wilcox pareada**, aplicando **correcciones para múltiples comparaciones**

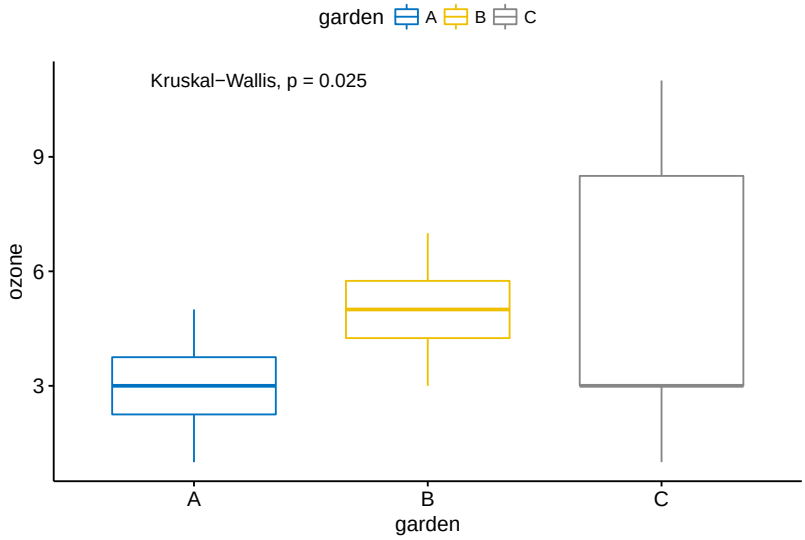
```
# rueba de Wilcox pareada, en este caso usando BH = fdr
# The false discovery rate is a less stringent condition than the family-wise error rate,
#   so these methods are more powerful than the others
# Usa ?p.adjust.method para ver las opciones
pairwise.wilcox.test(ozono$ozone, ozono$garden, p.adjust.method = "BH")
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test
##
## data:  ozono$ozone and ozono$garden
##
##   A     B
## B 0.009 -
## C 0.433 0.352
##
## P value adjustment method: BH
```

2.3.2.1 El paquete *ggpubr*

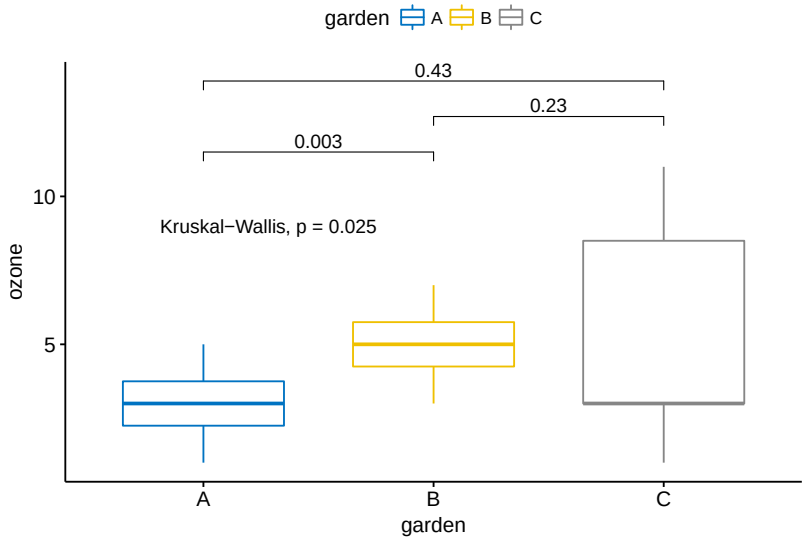
El paquete *ggpubr* tiene funciones muy convenientes para analizar y graficar resultados de pruebas de $ANOVA$, $Kruskal-Walis$ y tests post-hoc asociados.

```
ggboxplot(ozono, x = "garden", y = "ozone",
          color = "garden", palette = "jco") +
  stat_compare_means(method = "kruskal.test") # se puede cambiar por "anova"
```



Es muy conveniente que el paquete *ggpubr* puede señalar gráficamente los resultados de las pruebas post-hoc.

```
my_comparisons <- list( c("A", "B"), c("B", "C"), c("A", "C") )
ggboxplot(ozono, x = "garden", y = "ozone",
          color = "garden", palette = "jco")+
  stat_compare_means(comparisons = my_comparisons)+ # Add pairwise comparisons p-value
  stat_compare_means(label.y = 9)      # Add global p-value
```



Finalmente, si los datos:

1. no se desvían de la normalidad
2. pero las varianzas son heterogéneas,

podemos correr también el *oneway.test*(), que tiene más potencia que el de $Kruskal - Walis$.

```
oneway.test(ozone ~ garden, data = ozono)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  ozone and garden
## F = 7.5308, num df = 2.000, denom df = 16.458, p-value = 0.004761
```

2.4 Experimentos factoriales - ANOVA de doble vía

Un *experimento factorial* tiene **2 o más factores**, **cada uno con al menos dos niveles**, con **réplicas** en cada combinación de niveles de todos los factores. Bajo este diseño experimental podemos investigar **interacciones estadísticas**, las cuales se dan si la *respuesta a un factor depende de los niveles de otro factor*.

Vamos a usar un ejemplo clásico de diseño factorial balanceado, en el que se evalúan los efectos de las factores dieta (con 3 niveles) y suplemento (con 4 niveles), así como sus posibles interacciones, en la ganancia de peso de animales de granja a las 6 semanas de tratamiento.

Como siempre, empecemos leyendo los datos crudos y haciendo una exploración de los mismos

```
weights <- read.csv("data/growth.csv")
head(weights)
```

```
##   supplement  diet     gain
## 1  supergain wheat 17.37125
## 2  supergain wheat 16.81489
## 3  supergain wheat 18.08184
## 4  supergain wheat 15.78175
## 5    control wheat 17.70656
## 6    control wheat 18.22717
```

```
str(weights)
```

```
## 'data.frame':    48 obs. of  3 variables:
##  $ supplement: Factor w/ 4 levels "agrimore","control",..: 3 3 3 3 2 2 2 2 4 4 ...
##  $ diet      : Factor w/ 3 levels "barley","oats",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ gain      : num  17.4 16.8 18.1 15.8 17.7 ...
```

```
# hacemos attach para ahorrarnos algo de tecleo
attach(weights)
levels(diet)
```

```
## [1] "barley" "oats"   "wheat"
```

```
levels(supplement)
```

```
## [1] "agrimore"  "control"   "supergain" "supersupp"
```

2.4.1 Estadísticas de resumen mediante *tapply*() y gráficos con *barplot*() (R base)

2.4.1.1 *tapply*()

Apply a function to each cell of a ragged array, that is to each (non-empty) group of values given by a unique combination of the levels of certain factors

Usage, vean Vean ?tapply para los detalles

tapply(X, INDEX, FUN = NULL, ..., default = NA, simplify = TRUE)

- X an atomic object, typically a vector.

- INDEX a list of one or more factors, each of same length as X. The elements are coerced to factors by as.factor.
- FUN the function to be applied, or NULL. In the case of functions like +, %*%, etc., the function name must be backquoted or quoted. If FUN is NULL, tapply returns a vector which can be used to subscript the multi-way array tapply normally produces.

```
# Exploremos los datos usando R base, es decir, la via clasica, usando tapply()
tapply(gain, list(diet, supplement), length)
```

```
##        agrimore control supergain supersupp
## barley        4       4         4         4
## oats          4       4         4         4
## wheat         4       4         4         4
```

```
tapply(gain, list(diet, supplement), mean)
```

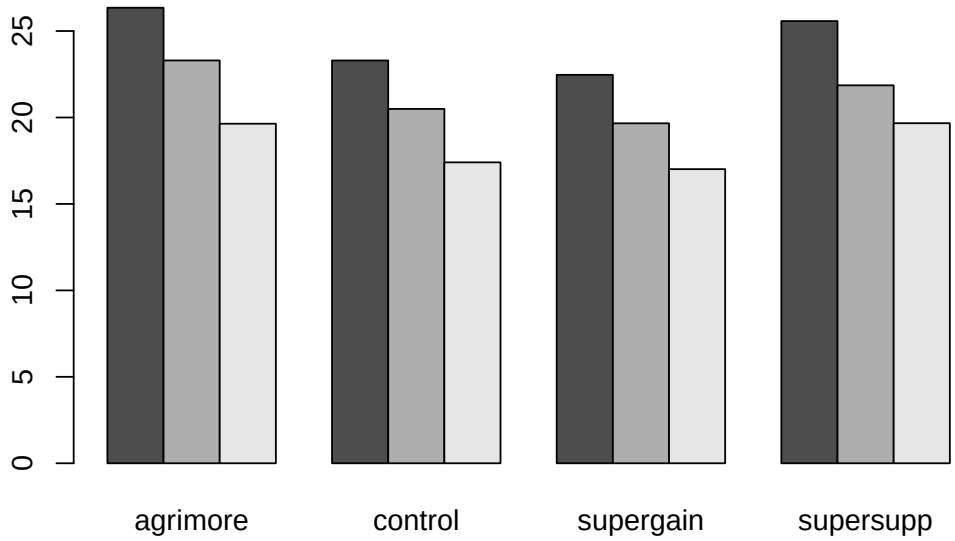
```
##        agrimore  control supergain supersupp
## barley 26.34848 23.29665  22.46612  25.57530
## oats   23.29838 20.49366  19.66300  21.86023
## wheat  19.63907 17.40552  17.01243  19.66834
```

```
tapply(gain, list(diet, supplement), var)
```

```
##        agrimore   control supergain supersupp
## barley 3.376391 1.9782372 2.3781610 4.4935647
## oats   1.503857 1.0226544 0.4870331 0.6830334
## wheat  2.015980 0.8480272 0.9419949 0.9011487
```

2.4.1.2 *barplot*

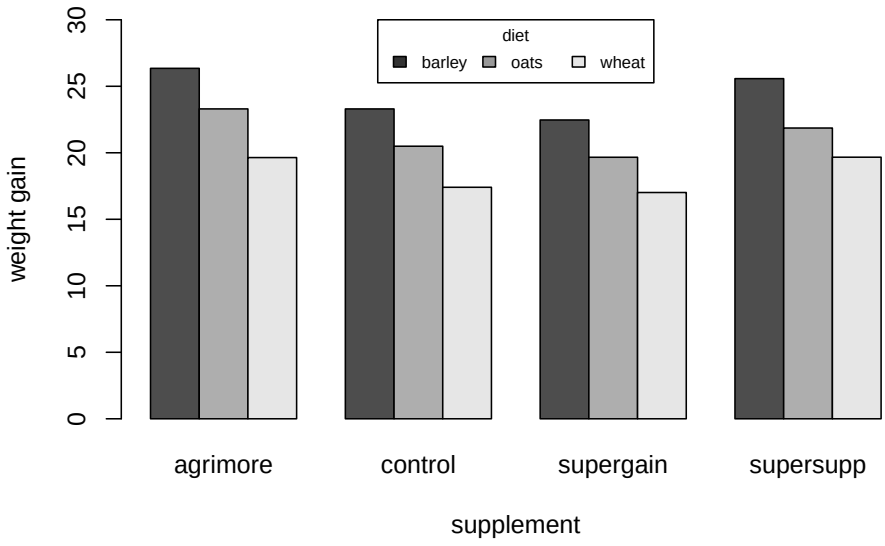
```
# grafiqumemos con barplot de R base
barplot(tapply(gain,list(diet,supplement),mean),beside=T)
```



- Podemos mejorar la gráfica, añadiendo una leyeda e incrementando la escala del eje y

```
# guardamos valores para la leyenda
labels <- levels(diet)
shade <- c(0.2,0.6,0.9)
```

```r
# graficamos, esta vez con nombres para los ejes x, y
barplot(tapply(gain,list(diet,supplement),mean),beside=T,
        ylab="weight gain",xlab="supplement",ylim=c(0,30))

# va la leyenda
legend("top", title = "diet", labels,
       fill =  gray(shade), horiz = TRUE, cex = 0.7)
```



¿Qué concluyes de estos resultados?

2.4.2 Análisis exploratorio usando ahora *dplyr*() y *ggplot*()

2.4.2.1 Tabla de múltiples estadísticos de resumen con *dplyr*

```r
# >>> Tabla de resumen de estadisticas descriptivas con dplyr
weights %>% group_by(diet, supplement) %>% summarise(n = n(),
                                                     mean = mean(gain, na.rm = TRUE),
                                                     median = median(gain, na.rm = TRUE),
                                                     variance = var(gain, na.rm = TRUE),
                                                     IQR = IQR(gain, na.rm = TRUE)
)
```

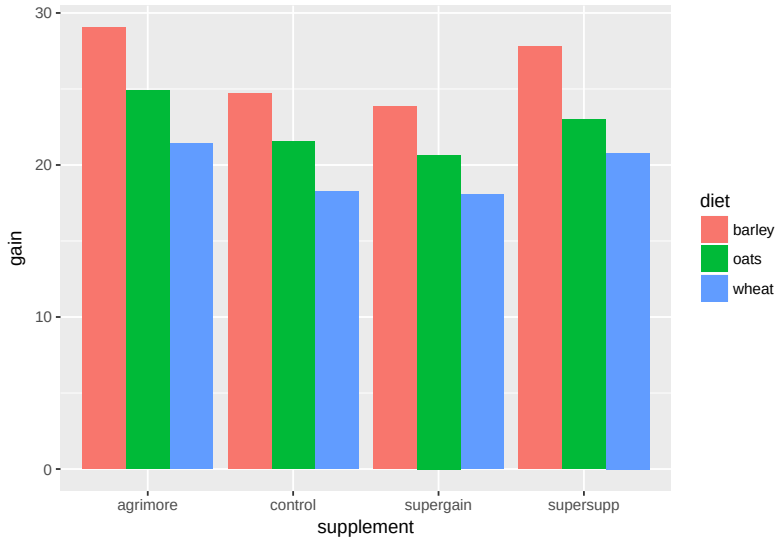
```
## # A tibble: 12 x 7
## # Groups:   diet [?]
##    diet   supplement     n  mean median variance   IQR
##    <fct>  <fct>      <int> <dbl>  <dbl>    <dbl> <dbl>
##  1 barley agrimore       4  26.3   25.7    3.38  1.57
##  2 barley control        4  23.3   23.2    1.98  2.30
##  3 barley supergain      4  22.5   22.6    2.38  2.53
##  4 barley supersupp      4  25.6   25.7    4.49  2.80
##  5 oats   agrimore       4  23.3   23.0    1.50  1.56
##  6 oats   control        4  20.5   20.6    1.02  1.10
##  7 oats   supergain      4  19.7   19.5    0.487 0.587
##  8 oats   supersupp      4  21.9   21.6    0.683 0.853
##  9 wheat  agrimore       4  19.6   19.6    2.02  0.914
## 10 wheat  control        4  17.4   17.7    0.848 0.614
## 11 wheat  supergain      4  17.0   17.1    0.942 0.992
```

```
## 12 wheat  supersupp      4  19.7   19.7    0.901 1.08
```

2.4.2.2 Gráficas de barras paralelas con 2 variables de agrupamiento con *ggplot*()

A medida que los graficos se hacen mas complejos, se hace mas facil configurarlos usando *ggplot*() que con *plot*()

```
# >>> gaficado con ggplot()
ggplot(weights, aes(x = supplement, y = gain, fill = diet)) +
  geom_bar(stat = "identity", position = "dodge")
```



2.4.3 ANOVA de doble vía: experimento factorial completo

Evaluemos el modelo *gain* ~*diet* ∗ *supplement*, que contempla todas las interacciones con supplement

Estimamos parámetros para los *efectos principales* de cada nivel de *diet* y de cada nivel de *supplement*, más términos para las interacciónes entre los niveles de ambos factores.

```
model <- aov(gain ~ diet * supplement)
summary(model)
```

```
##                Df Sum Sq Mean Sq F value   Pr(>F)
## diet            2 287.17  143.59   83.52 3.00e-14 ***
## supplement      3  91.88   30.63   17.82 2.95e-07 ***
## diet:supplement 6   3.41    0.57    0.33    0.917
## Residuals      36  61.89    1.72
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

De la tabla de ANOVA podemos concluir que no hay evidencia de interacciones entre los niveles de ambos factores (fila *diet* : *supplement*, $p = .916$). Además indica que los efectos de *diet* y *supplement* son aditivos y significativos.

Como ya discutimos anteriormente, la desventaja de *summary.aov*() es que no nos muestra los *tamaños de los efectos*. Para evaluar qué niveles de cada factor son significativamente diferentes, necesitamos recurrir a *summay.lm*().

2.4.3.1 Evaluación del tamaño de los efectos de cada nivel de cada factor

Resumen del ajuste de modelo ANOVA con *summary.lm*()

```
summary.lm(model)
```

```
##
## Call:
## aov(formula = gain ~ diet * supplement)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.48756 -1.00368 -0.07452  1.03496  2.68069
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    26.3485     0.6556  40.191  < 2e-16 ***
## dietoats                       -3.0501     0.9271  -3.290 0.002248 **
## dietwheat                      -6.7094     0.9271  -7.237 1.61e-08 ***
## supplementcontrol              -3.0518     0.9271  -3.292 0.002237 **
## supplementsupergain            -3.8824     0.9271  -4.187 0.000174 ***
## supplementsupersupp            -0.7732     0.9271  -0.834 0.409816
## dietoats:supplementcontrol      0.2471     1.3112   0.188 0.851571
## dietwheat:supplementcontrol     0.8183     1.3112   0.624 0.536512
## dietoats:supplementsupergain    0.2470     1.3112   0.188 0.851652
## dietwheat:supplementsupergain   1.2557     1.3112   0.958 0.344601
## dietoats:supplementsupersupp   -0.6650     1.3112  -0.507 0.615135
## dietwheat:supplementsupersupp   0.8024     1.3112   0.612 0.544381
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.311 on 36 degrees of freedom
## Multiple R-squared:  0.8607, Adjusted R-squared:  0.8182
## F-statistic: 20.22 on 11 and 36 DF,  p-value: 3.295e-12
```

El modelo evaluado es bastante complejo ya que se estimaron 12 parámetros (filas en la tabla): 6 $efectos\ principales$ y 6 *interacciones*. La última columna nos indica dos cosas:

1. enfatiza nuestra conclusión de que ninguna interacción es significativa
2. sugiere que el *modelo minimo adecuado* va a requerir a lo sumo 5 parámetros (filas con asteriscos):

- un punto de corte (Intercept)
- una diferencia debida a la avena (oats)
- una diferencia debida al trigo (wheat)
- una diferencia debida al control
- una diferencia debida al suplemento supergain

Una inspección detallada de la tabla muestra que los *tamaños de los efectos* debidos al *control* y *supergain* NO son significativamente diferents entre ellos. ¿Porqué?. Ello se infiere aplicando *pruebas t de Student* "mentalmente" a pares de parámetros (filas de la tabla).

Veamos cómo se calcula: Ignorando los signos (ambos negativos), vemos que la diferencia en es $dif.tam.efectos = 3.88 - 3.05 = .83$. Considerando los *errores/estándar* asociados a ambos $se = .927$, vemos que son ~ 1. Para que la diferencia entre ellos sea significativa, necesitamos ~2 *errores estandard* (Recordemos la regla aproximada de que $t \geq 2$, es significativa, donde $t = \frac{dif\ entre\ medias}{SE_{diff}}$

Podemos probarlo formalmente:

```r
ctrl <- weights %>% dplyr::filter(supplement == "control")
supg <- weights %>% dplyr::filter(supplement == "supergain")

length(ctrl$gain)
```

```
## [1] 12
```

```r
length(supg$gain)
```

```
## [1] 12
```

```r
# el valor critico
qt(.927, 22)
```

```
## [1] 1.507143
```

```r
# la prueba
t.test(ctrl$gain, supg$gain )
```

```
##
##  Welch Two Sample t-test
##
## data:  ctrl$gain and supg$gain
## t = 0.63826, df = 21.903, p-value = 0.5299
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.540776  2.910293
## sample estimates:
## mean of x mean of y
##  20.39861  19.71385
```

¿Porqué llevan estrellas *control* y *supergain* si no son significativas entre ellas?

Ello se debe a que los contrastes entre tratamientos comparan todos los *efectos principales* en las filas con la fila *Intercept*, donde, recordemos, cada factor es asignado a su primer nivel en orden alfabético (barley y agrimore, en nuestro caso). Cuando (como aquí) varios niveles de un factor son diferentes del *Intercept*, pero no entre ellos, todos obtienen estrellitas de significancia. Es por ello que **no podemos simplemente contar las filas con estrellas para determinar el número de niveles o tratamientos significativamente diferentes** para definir el *modelo minimo adecuado*.

2.4.3.2 Simplificación del modelo

Empezemos la simplificación del modelo defando fuera los *términos de interacciones*, ya que vimos que no son significativas. Podemos verificar con *anova*() que el modelo simplificado resultante no es significativamente peor que el que incluye todos los términos de las interacciones

```r
model1 <- lm(gain ~ diet+ supplement)
summary(model1)
```

```
##
## Call:
## lm(formula = gain ~ diet + supplement)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.30792 -0.85929 -0.07713  0.92052  2.90615
##
## Coefficients:
```

```
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          26.1230     0.4408  59.258  < 2e-16 ***
## dietoats             -3.0928     0.4408  -7.016 1.38e-08 ***
## dietwheat            -5.9903     0.4408 -13.589  < 2e-16 ***
## supplementcontrol    -2.6967     0.5090  -5.298 4.03e-06 ***
## supplementsupergain  -3.3815     0.5090  -6.643 4.72e-08 ***
## supplementsupersupp  -0.7274     0.5090  -1.429     0.16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.247 on 42 degrees of freedom
## Multiple R-squared:  0.8531, Adjusted R-squared:  0.8356
## F-statistic: 48.76 on 5 and 42 DF,  p-value: < 2.2e-16
```

```
anova(model1,model)
```

```
## Analysis of Variance Table
##
## Model 1: gain ~ diet + supplement
## Model 2: gain ~ diet * supplement
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     42 65.296
## 2     36 61.890  6    3.4058 0.3302 0.9166
```

Inspeccionando las estimas de los parámetros en la tabla de ANOVA, vemos claramente que necesitamos retener los tres niveles de *diet*, ya que $dietoats - dietwheat = -3.0928 - -5.9903 = 2.8975$, que es $>> 2$, que con un $std.err = .44$ implica que $(t >> 2)$, y por tanto significativa.

Formalmente, podemos demostrar que $p < 0.001$, como se muestra seguidamente:

```
levels(diet)
```

```
## [1] "barley" "oats"   "wheat"
```

```
barley_dfr <- weights %>% filter(diet == "wheat")
oats_dfr <- weights %>% filter(diet == "oats")

#s la prueba
t.test(barley_dfr$gain, oats_dfr$gain)
```

```
##
##  Welch Two Sample t-test
##
## data:  barley_dfr$gain and oats_dfr$gain
## t = -5.0197, df = 29.949, p-value = 2.213e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.076412 -1.718550
## sample estimates:
## mean of x mean of y
##  18.43134  21.32882
```

En cambio, *supersupp* no es claramente diferente de *agrimore* ($dif = -.727$, con $err.std = .509$). Tampoco *supergain* es claramente diferente del *control* sin suplemento ($dif = .68$ con $err.std = .51$), como habíamos probado anteriormente.

Por tanto parece que podemos simplificar el modelo, reemplazando los 4 factores originales de *supplement* por sólo dos niveles. Para ello recodificamos los niveles *agrimore* y *supersupp* como *best* y los tratamientos

control y *supergain* como *worst*, de la siguiente manera:

```
supp2 <- factor(supplement)
levels(supp2)
```

```
## [1] "agrimore"  "control"   "supergain" "supersupp"
```

```
levels(supp2)[c(1,4)] <- "best"   # agrimore, supersupp, dif no significativa entre ellos
levels(supp2)[c(2,3)] <- "worst"  # control, supergain, dif no significativa entre ellos
levels(supp2)
```

```
## [1] "best"  "worst"
```

Ahora procedemos a ajustar el modelos simplificado a los datos:

```
model2 <- lm(gain ~ diet + supp2)
```

Y evaluamos formalmente si el modelo simplificado representa o no un peor ajuste a los datos que model y model1

```
anova(model, model1, model2)
```

```
## Analysis of Variance Table
##
## Model 1: gain ~ diet * supplement
## Model 2: gain ~ diet + supplement
## Model 3: gain ~ diet + supp2
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     36 61.890
## 2     42 65.296 -6   -3.4058 0.3302 0.9166
## 3     44 71.284 -2   -5.9876 1.7414 0.1897
```

Voila: *model*2 es el *modelo minimo adecuado* que buscábamos, ya que su ajuste no es significativamente peor que los dos precedentes, claramente sobreparametrizados.

Exploremos los parámetros de *model*2

```
summary(model2)
```

```
##
## Call:
## lm(formula = gain ~ diet + supp2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6716 -0.9432 -0.1918  0.9293  3.2698
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  25.7593     0.3674  70.106  < 2e-16 ***
## dietoats     -3.0928     0.4500  -6.873 1.76e-08 ***
## dietwheat    -5.9903     0.4500 -13.311  < 2e-16 ***
## supp2worst   -2.6754     0.3674  -7.281 4.43e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.273 on 44 degrees of freedom
## Multiple R-squared:  0.8396, Adjusted R-squared:  0.8286
## F-statistic: 76.76 on 3 and 44 DF,  p-value: < 2.2e-16
```

Como ven, hemos logrado simplificar el modelo inicial de 12 parámetros a uno mucho más fácilmente interpretable de sólo 4. Por tanto, si nuestro objetivo es obtener la máxima ganancia en peso, entonces una dieta de avena (*barley*) suplementada con *agrimore* o *supersupp* es lo adecuado (escondidos en *Intercept*). Para visualizarlo, podemos usar la ya conocida función $TukeyHSD()$, pasándole el modelo en su formato de anova.

```
aov_model2 <- aov(gain ~ diet + supp2)
TukeyHSD(aov_model2)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = gain ~ diet + supp2)
##
## $diet
##                    diff       lwr       upr p adj
## oats-barley  -3.092817 -4.184312 -2.001322 1e-07
## wheat-barley -5.990298 -7.081792 -4.898803 0e+00
## wheat-oats   -2.897481 -3.988976 -1.805986 2e-07
##
## $supp2
##                  diff       lwr       upr p adj
## worst-best -2.675403 -3.415916 -1.934891     0
```

```
detach(weights)
```

2.4.4 ANOVA de doble vía con interacciones

Veamos un último ejemplo de ANOVA de doble vía, esta vez un caso en el que sí hay interacciones significativas.

Usaremos el data frame *ToothGrowth* que viene con R. Se evalúa el efecto del ácido ascórbico sobre el crecimiento de dientes. Son 60 conejos asignados al azar a recibir uno de 3 niveles de ascorbato, suministrado en una de dos formas posibles (jugo de naranja o vitamina C). Por tanto, cada tratamiento consta de 10 individuos ($10x2x3 = 60$).

Carguemos y exploremos los datos mediante estadísticas de resumen

```
attach(ToothGrowth)
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 ...
```

```
table(supp, dose)
```

```
##     dose
## supp 0.5  1  2
##   OJ  10 10 10
##   VC  10 10 10
```

```
ToothGrowth %>% group_by(supp, dose) %>% summarize(n = n(),
                                                   mean = mean(len, na.rm = TRUE),
                                                   sd   = sd(len, na.rm = TRUE),
                                                   var  = var(len, na.rm = TRUE)
                                                   )
```

```
## # A tibble: 6 x 6
## # Groups:   supp [?]
##   supp   dose     n  mean    sd   var
##   <fct> <dbl> <int> <dbl> <dbl> <dbl>
## 1 OJ      0.5    10 13.2   4.46 19.9
## 2 OJ      1      10 22.7   3.91 15.3
## 3 OJ      2      10 26.1   2.66  7.05
## 4 VC      0.5    10  7.98  2.75  7.54
## 5 VC      1      10 16.8   2.52  6.33
## 6 VC      2      10 26.1   4.80 23.0
```

Vemos en las tablas que se trata de un diseño balanceado (misma n para cada celda del diseño), por lo que no tenemos que preocuparnos por el orden en el que comparamos los efectos.

La salida de *str*() nos muestra que la variable *dose* está codificada como numérica. Necesitamos recodificarla como factor de agrupamiento, para que *aov*() no la trate como una covariable numérica (ANCOVA), tal y como se muestra seguidamente:

```
dose <- factor(dose)
fit <- aov(len ~ supp*dose)
summary(fit)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## supp         1  205.4   205.4  15.572 0.000231 ***
## dose         2 2426.4  1213.2  92.000  < 2e-16 ***
## supp:dose    2  108.3    54.2   4.107 0.021860 *
## Residuals   54  712.1    13.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

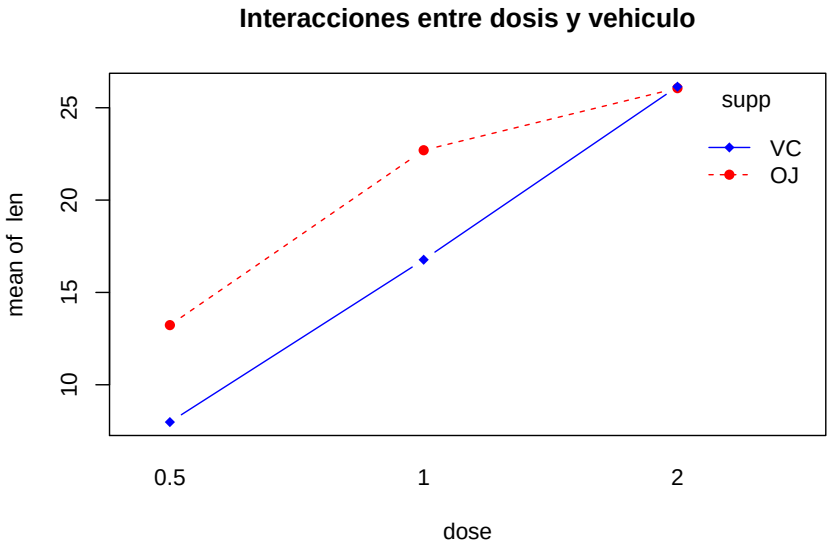
La salida de *summary*(fit) nos muestra que tanto los *efectos principales* (*sup* y *dose*) como las *interacciones* entre estos factores son significativas.

2.4.4.1 Visualización de interacciones entre factores

Existen varios paquetes para visualizar interacciones, así como la función *interaction.plot*() del paquete *stats* del sistema base.

- *interaction.plot*()

```
interaction.plot(dose, supp, len, type = "b",
                 col=c("red", "blue"), pch = c(16,18),
                 main = "Interacciones entre dosis y vehiculo")
```



La gráfica muestra la longitud promedio de los dientes para cada suplemento y dosis. Es claro que la longitud de los dientes incrementa con la dosis de ascorbato, tanto si se administra como jugo de naranja como en forma de vitamina C. A la dosis más alta (2 mg), ambos vehículos producen el mismo crecimiento. Por tanto existe una interacción entre el nivel más alto de dosis con la variable *type*. En el resto de los niveles para los dos factores no existen interacciones, como denota el correr paralelo de las líneas.

La salida de la función $summary.lm()$ nos confirma que existe una interacción significativa entre $suppVC$: $dose2$, pero no a la dosis1 o control.

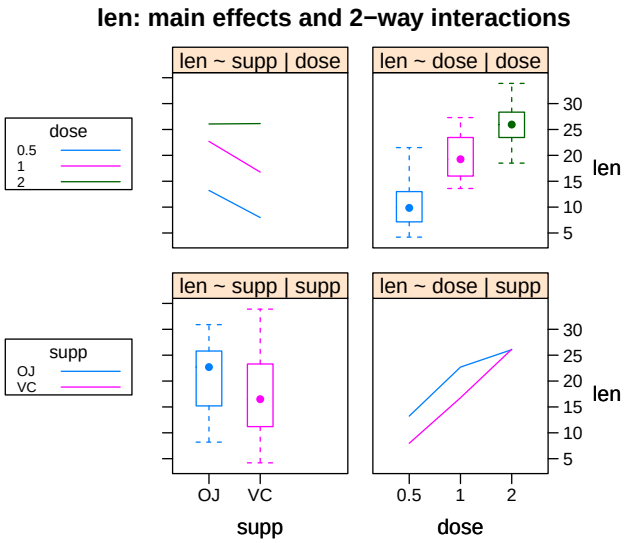
```
summary.lm(fit)
```

```
##
## Call:
## aov(formula = len ~ supp * dose)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8.20  -2.72  -0.27   2.65   8.27
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    13.230      1.148  11.521 3.60e-16 ***
## suppVC         -5.250      1.624  -3.233  0.00209 **
## dose1           9.470      1.624   5.831 3.18e-07 ***
## dose2          12.830      1.624   7.900 1.43e-10 ***
## suppVC:dose1   -0.680      2.297  -0.296  0.76831
## suppVC:dose2    5.330      2.297   2.321  0.02411 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.631 on 54 degrees of freedom
## Multiple R-squared:  0.7937, Adjusted R-squared:  0.7746
## F-statistic: 41.56 on 5 and 54 DF,  p-value: < 2.2e-16
```

- HH::interaction2wt()

El paquete HH provee funciones para un análisis gráfico más sofisticado. Despliega boxplots para los *efectos principales* y las interacciones de doble vía para diseños de complejidad arbitraria, siendo por tanto muy útil.

```
HH::interaction2wt(len ~ supp *  dose)
```



3 Funciones y paquetes de R usados para este documento

3.1 Paquetes y software para investigación reproducible y generación de documentos en múltiples formatos

- knitr (Xie 2018)
- pandoc (MacFerlane 2016)
- rmarkdown (Allaire et al. 2018)

3.2 Paquetes de uso general para procesamiento y graficado de datos

- dplyr (Wickham et al. 2018)
- ggplot2 (Wickham 2009)

3.3 Análisis de la varianza - ANOVA

3.3.1 Funciones de paquetes base (R Core Team 2018)

- abline()
- anova()
- aov()
- arrows()
- attach()
- c()
- cat()
- class()
- coef()
- confint()
- colnames()
- data()

- data.frame()
- df()
- dim()
- factor()
- head()
- help()
- interaction.plot()
- kruskal.test()
- levels()
- length()
- library()
- lines()
- lm()
- names()
- mean()
- median()
- oneway.test()
- par()
- pairwise.t.test()
- plot()
- qf()
- read.csv()
- rep()
- shapiro.test()
- str()
- subset()
- sum()
- summary()
- summary.aov()
- summary.lm()
- table()
- tail()
- tapply()
- t.test()
- tukeyHSD()
- var.test()

3.3.2 Datos del paquetes base

- datasets [R-datasets]

3.3.3 Paquetes especializados

- car::leveneTest() (Fox, Weisberg, and Price 2018)
- car::qqPlot() (Fox, Weisberg, and Price 2018)
- gplots::plotmeans() (Warnes et al. 2016)
- multcomp::glht() (Hothorn, Bretz, and Westfall 2008)
- PMCMR::posthoc.kruskal.nemenyi.test (Pohlert 2014)
- PMCMR::posthoc.kruskal.dunn.test (Pohlert 2014)
- ggpubr::ggboxplot() (Kassambara 2017)
- ggpubr::stat_compare_means() (Kassambara 2017)
- HH::interaction2wt() (Heiberger 2017)

4 Recursos en línea

4.1 The comprehensive R archive network (CRAN)

- CRAN

4.2 Cursos

- RStudio - online learning
- datacamp - learning R
- swirl - learn R, in R

4.3 Consulta

- R cookbook
- QuickR
- downloadable books o R and stats
- Use R!
- Official CRAN documentation
- r, stackoverflow

4.4 Manipulación y graficado de datos con paquetes especializados

- plotly and gglplot2 user guide
- Data wrangling with R and RStudio
- Data wrangling with R and RStudio - cheatsheet
- Data wrangling with R and RStudio - webinar
- Bradley C Boehmke - Data wrangling with R

Referencias

Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, and Winston Chang. 2018. *Rmarkdown: Dynamic Documents for R*. https://CRAN.R-project.org/package=rmarkdown.

Crawley, Michael J. 2012. *The R book*. 2nd ed. Wiley.

———. 2015. *Statistics : an introduction using R*. 2nd ed. Wiley.

Field, Andy P., Jeremy Miles, and Zoe. Field. 2012. *Discovering statistics using R*. 1st ed. London: Sage.

Fox, John, Sanford Weisberg, and Brad Price. 2018. *Car: Companion to Applied Regression*. https://CRAN.R-project.org/package=car.

Heiberger, Richard M. 2017. *HH: Statistical Analysis and Data Display: Heiberger and Holland.* https://CRAN.R-project.org/package=HH.

Hothorn, Torsten, Frank Bretz, and Peter Westfall. 2008. "Simultaneous Inference in General Parametric Models." *Biometrical Journal* 50 (3): 346–63.

Kabacoff, Robert. 2015. *R in action : data analysis and graphics with R*. 2nd ed. Manning.

https://github.com/kabacoff/RiA2 http://www.statmethods.net/.

Kassambara, Alboukadel. 2017. *Ggpubr: 'Ggplot2' Based Publication Ready Plots*. http://www.sthda.com/english/rpkgs/ggpubr.

MacFerlane, John. 2016. "Pandoc - a universal document converter." http://pandoc.org/.

Pohlert, Thorsten. 2014. *The Pairwise Multiple Comparison of Mean Ranks Package (Pmcmr)*. https://CRAN.R-project.org/package=PMCMR.

R Core Team. 2018. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Warnes, Gregory R., Ben Bolker, Lodewijk Bonebakker, Robert Gentleman, Wolfgang Huber Andy Liaw, Thomas Lumley, Martin Maechler, et al. 2016. *Gplots: Various R Programming Tools for Plotting Data*. https://CRAN.R-project.org/package=gplots.

Wickham, Hadley. 2009. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. http://ggplot2.org.

Wickham, Hadley, Romain Francois, Lionel Henry, and Kirill Mueller. 2018. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Xie, Yihui. 2018. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. https://CRAN.R-project.org/package=knitr.