

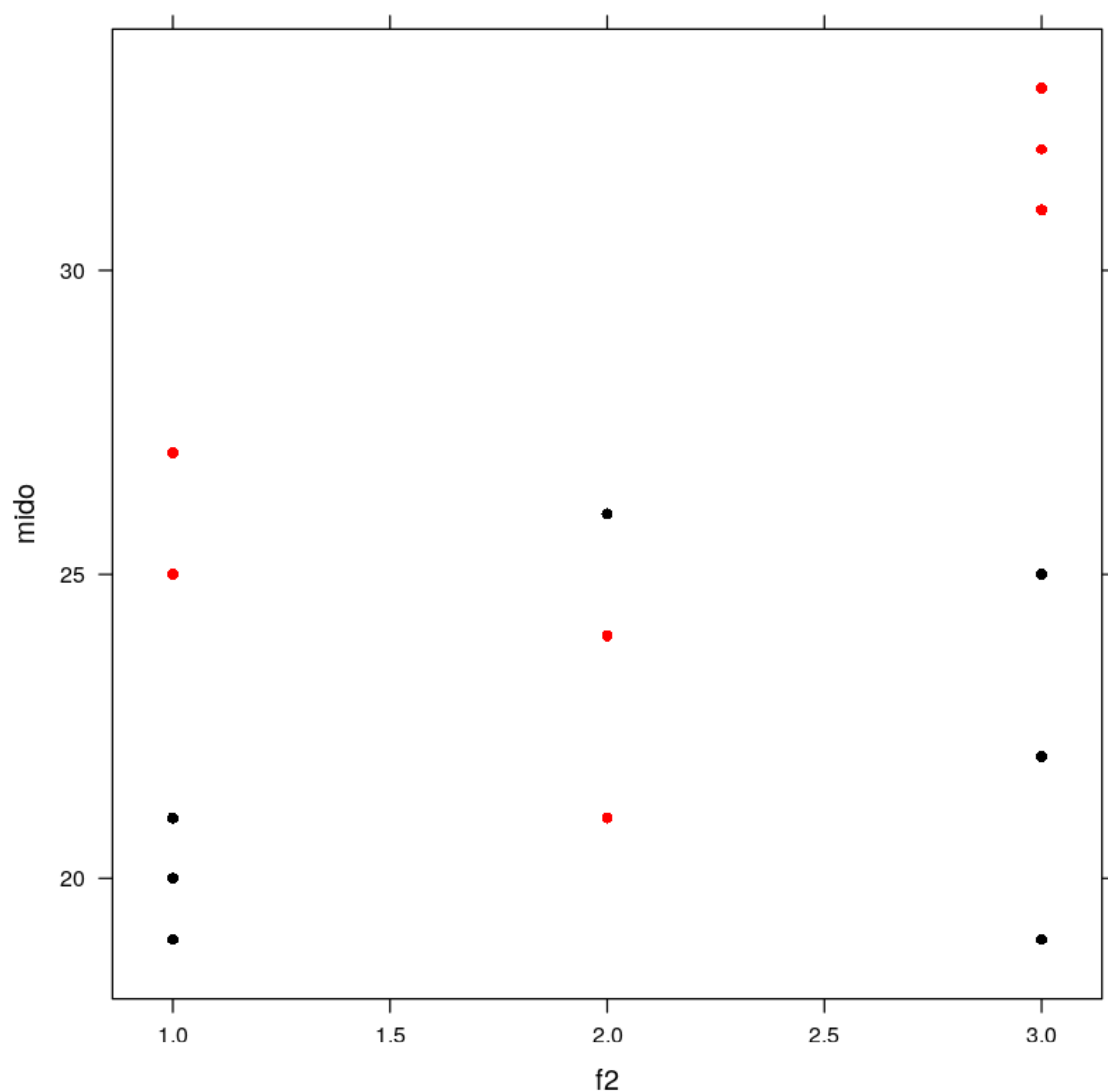
1 Clase 14 - 4/10/2018

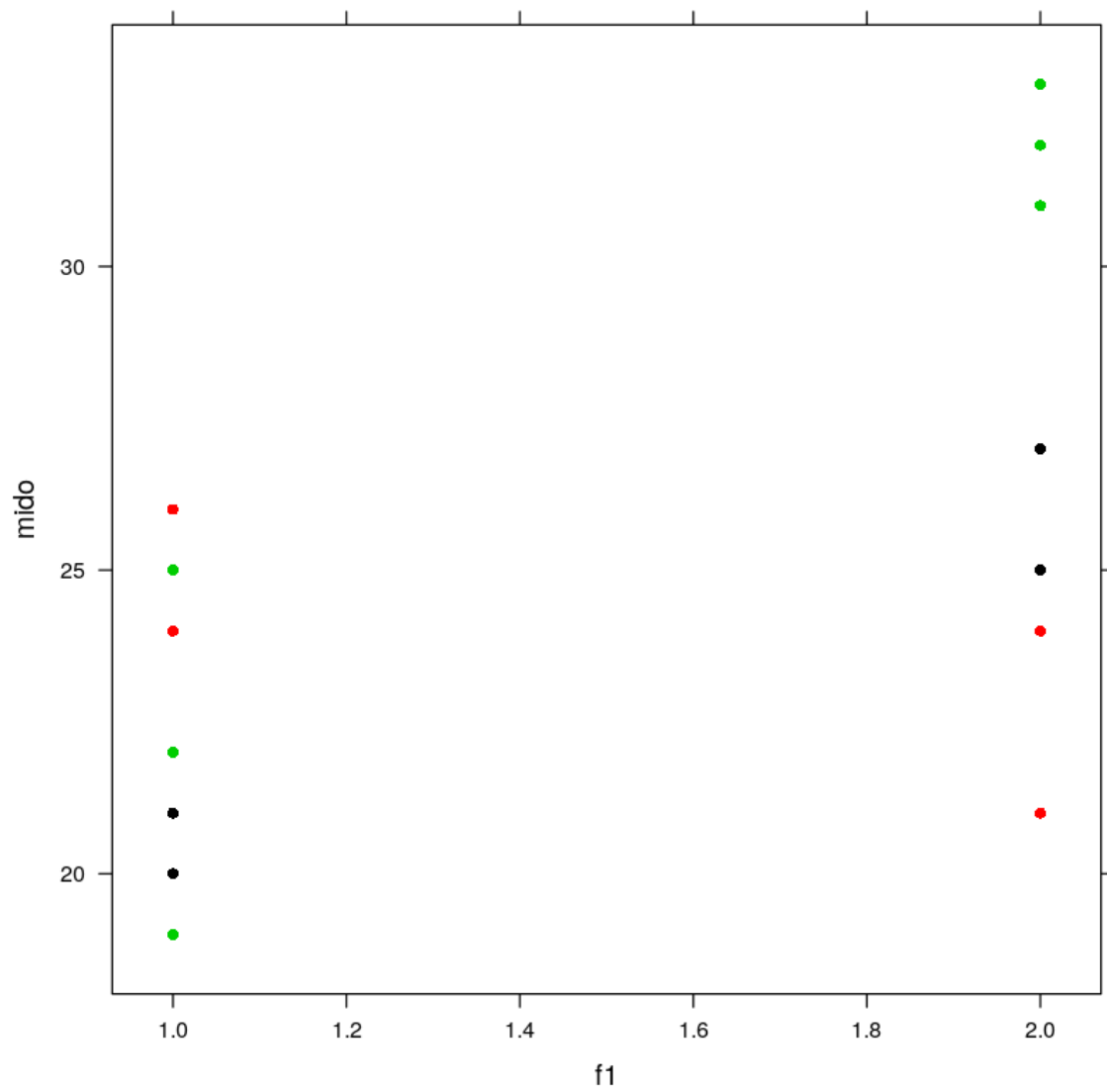
(x emilopez)

- Ejercicio de la página 42 del apunte 05.pdf
- En dropbox Estadistica.Aplicada.2018/02_Disenio_Factorial_y_bloques
- Tenemos 2 factores
- Usamos el archivo horti.csv

In [1]:

```
1 # Ejercicio de la pag. 42/90
2 datos = read.table("/home/emiliano/EstadisticaAplicada/Estadistica.Aplicada.201
3 attach(datos)
4 fa1 = as.factor(f1)
5 fa2 = as.factor(f2)
6
7 library(lattice)
8 fig1 = xyplot(mido ~f2, pch = 16, col = as.factor(f1))
9 print(fig1)
10
11 fig2 = xyplot(mido ~f1, pch = 16, col = as.factor(f2))
12 print(fig2)
```



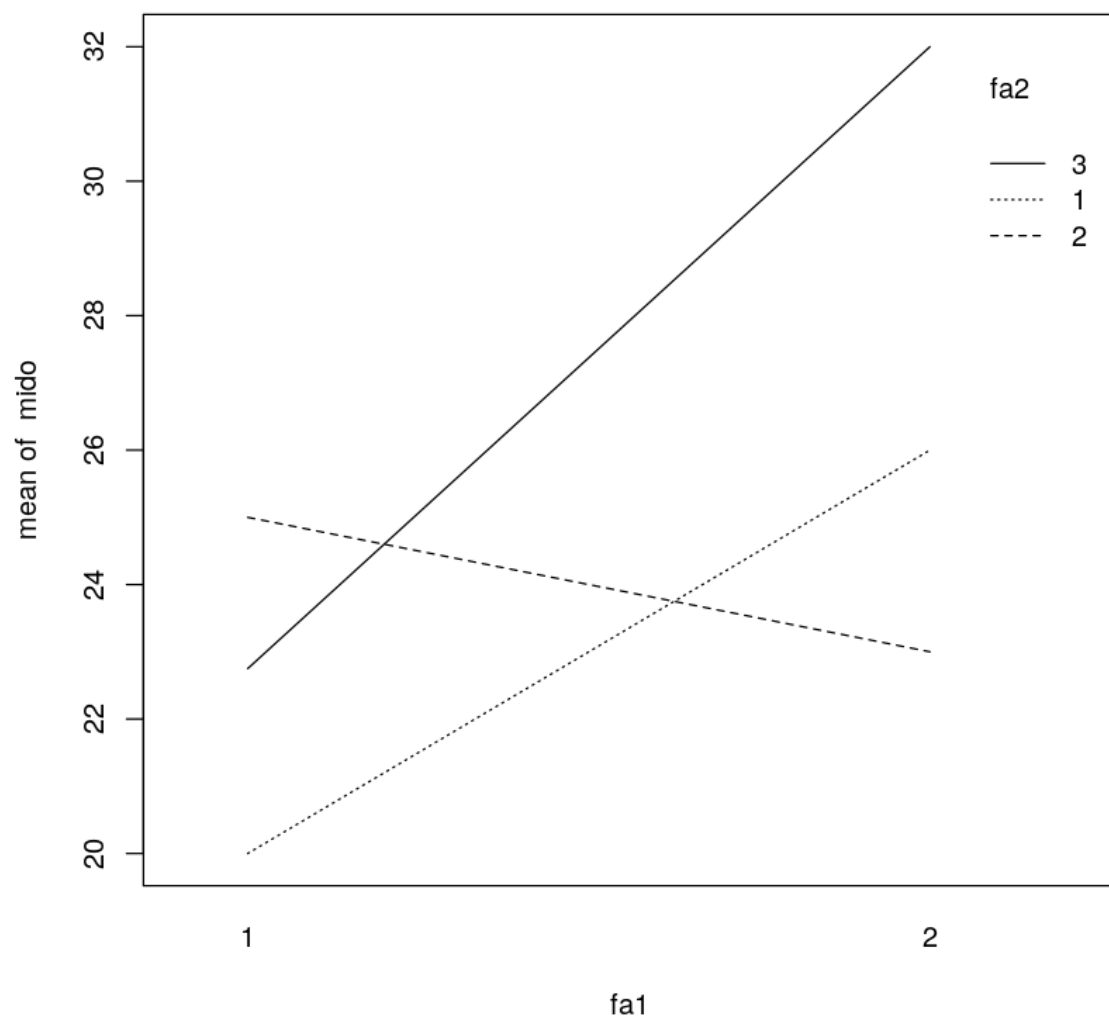


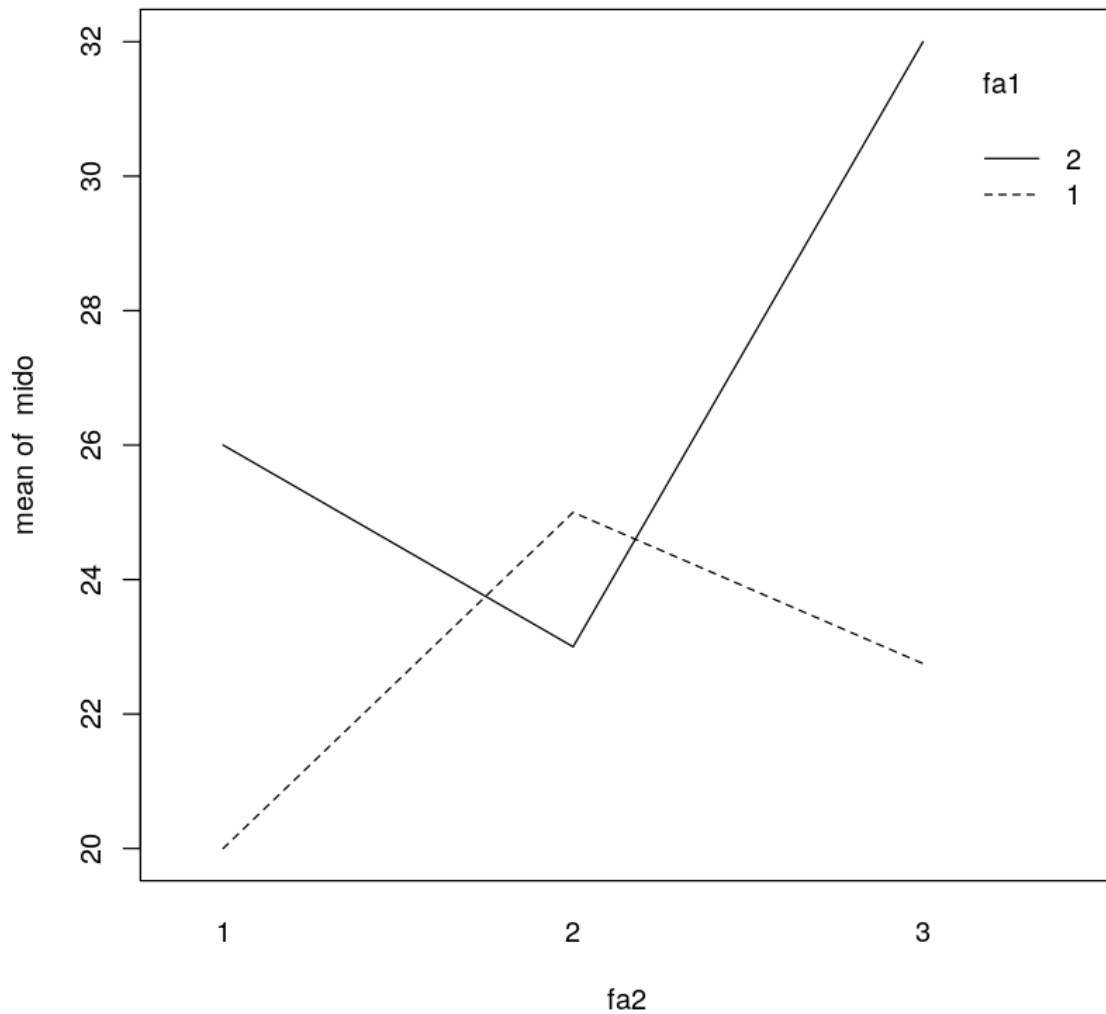
Pasos a seguir:

- Gráficos de interacción
- ANOVA
- Análisis de residuos
 - Si nos da satisfactorio, seguimos
 - Si no, deberíamos hacer alguna *transformación*

In [2]:

```
1 # graficos de interaccion
2 interaction.plot(fa1, fa2, mido)
3 interaction.plot(fa2, fa1, mido)
```





In [3]:

```
1 # modelo, vemos que la interaccion nos da significativa porque fa1:fa2 nos da u
2 model = aov(mido~fa1*fa2)
3 summary(model)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
fa1	1	128.00	128.00	39.64	3.97e-05	***
fa2	2	81.51	40.75	12.62	0.001119	**
fa1:fa2	2	95.74	47.87	14.82	0.000572	***
Residuals	12	38.75	3.23			

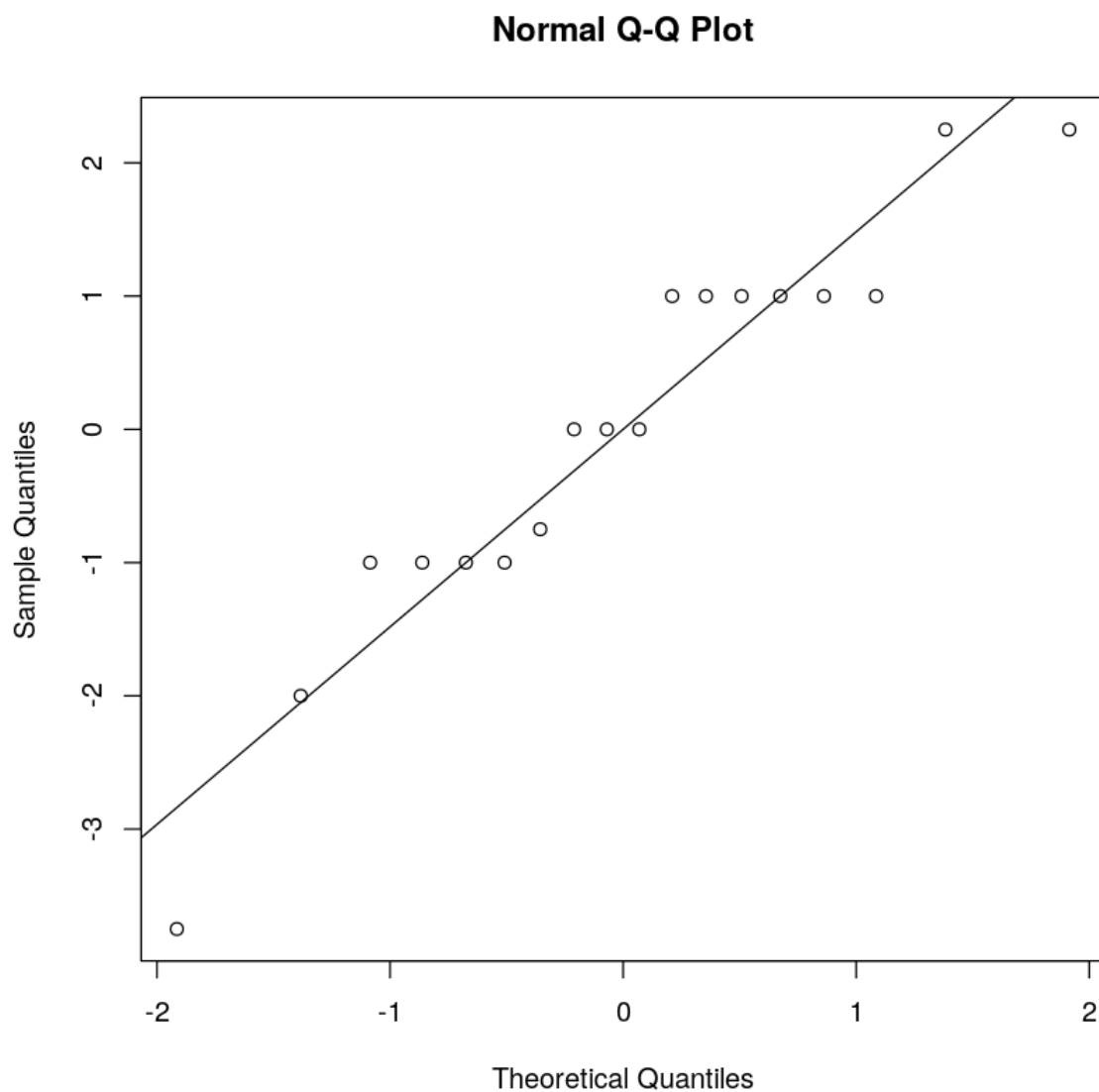
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

In [4]:

```
1 # fitea bien los datos?
2 # veamos los residuos
3 # Normalidad
4 qqnorm(model$residuals)
5 qqline(model$residuals)
6 # no tenemos evidencia fuerte en contra de la normalidad, pero ojo, pq tenemos
7 shapiro.test(model$residuals)
```

Shapiro-Wilk normality test

data: model\$residuals
W = 0.92408, p-value = 0.1527



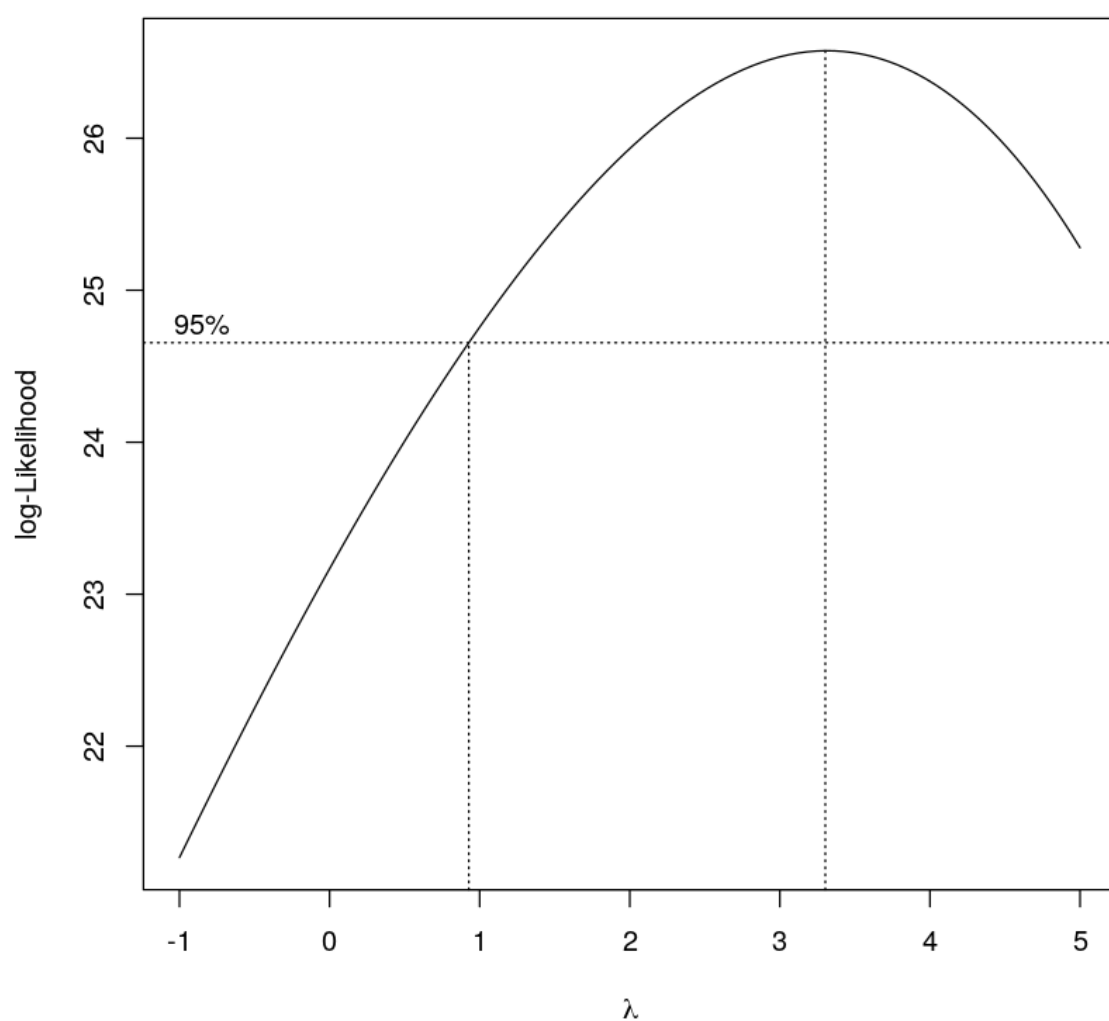
In [6]:

```
1 # vemos varianza constante
2 library(car)
3 leveneTest(model)
4 #outLierTest(model)
```

	Df	F value	Pr(>F)
group	5	1.225434	0.3556694
	12	NA	NA

In [7]:

```
1 # transformacion?? parece que conviene transformar, nos queda 1 dentro del inte
2 library(MASS)
3 boxcox(model, lambda = seq(-1, 5, length(20)))
4
5 # podriamos haber llegado a lo mismo con anova, para ver el mismo pvalor anova(
6 # model_red = aov(mido~fa1+fa2)
7 # anova(model_red, model)
```



In [9]:

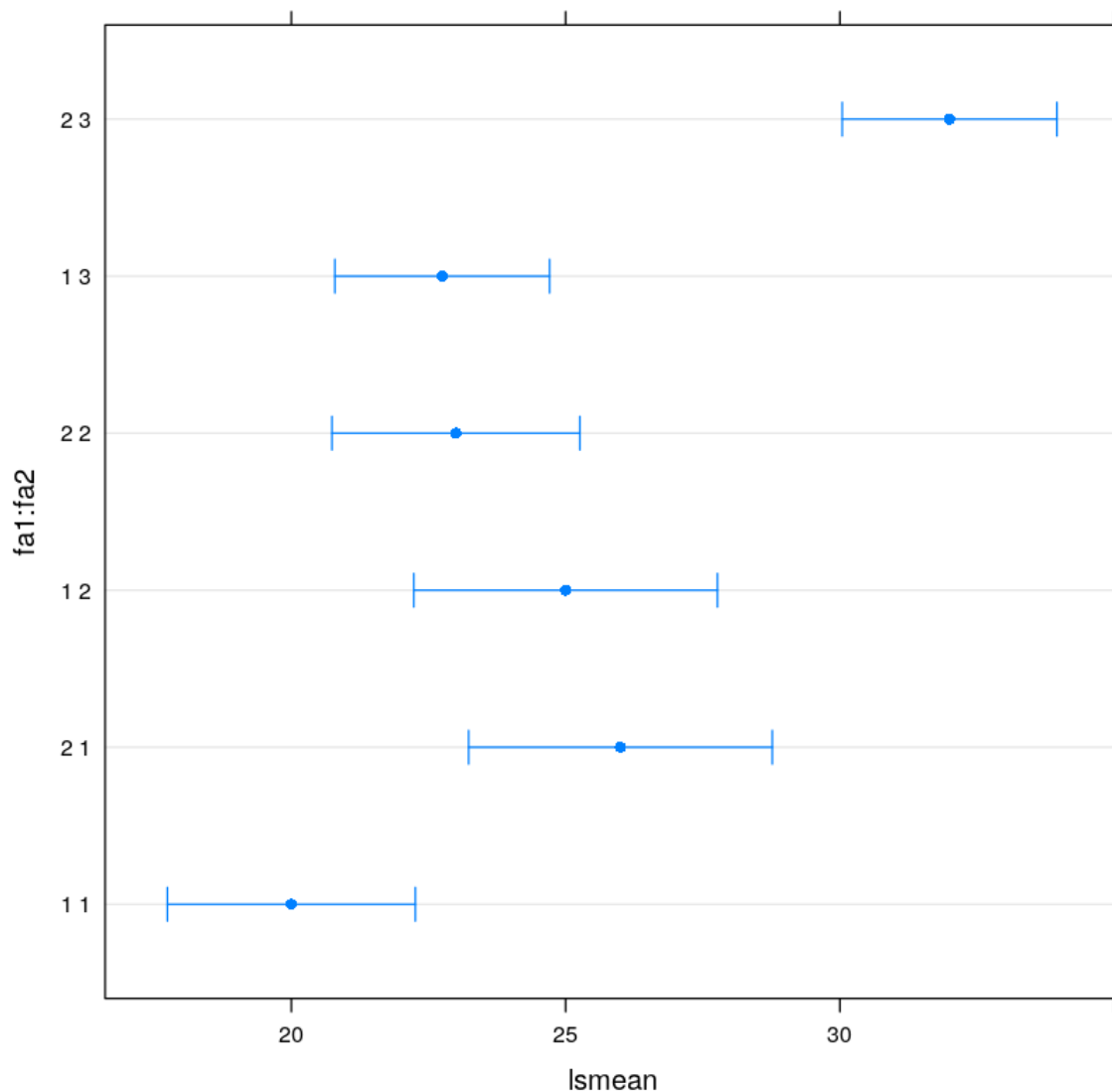
```
1 # cualquier analisis siguiente que tengamos que hacer lo vamos a hacer con la i
2 # si queremos las medias por grupo, x ej, debemos hacer un IC para esa media
3 # o sea, esto nos sirve para encontrar estimadores de las medias
4
5 library(lsmeans)
6 lsmeans(model, ~fa1*fa2)
7 plot(lsmeans(model, ~fa1*fa2))
8 print(lsmeans(model, ~fa1*fa2))
9 # el primer caso es el fungicida 2 con la dosis 3
```

fa1	fa2	lsmean	SE	df	lower.CL	upper.CL
1	1	20.00	1.0374916	12	17.73950	22.26050
2	1	26.00	1.2706626	12	23.23146	28.76854
1	2	25.00	1.2706626	12	22.23146	27.76854
2	2	23.00	1.0374916	12	20.73950	25.26050
1	3	22.75	0.8984941	12	20.79235	24.70765
2	3	32.00	0.8984941	12	30.04235	33.95765

Confidence level used: 0.95

fa1	fa2	lsmean	SE	df	lower.CL	upper.CL
1	1	20.00	1.0374916	12	17.73950	22.26050
2	1	26.00	1.2706626	12	23.23146	28.76854
1	2	25.00	1.2706626	12	22.23146	27.76854
2	2	23.00	1.0374916	12	20.73950	25.26050
1	3	22.75	0.8984941	12	20.79235	24.70765
2	3	32.00	0.8984941	12	30.04235	33.95765

Confidence level used: 0.95



In [10]:

```

1 # como la interaccion es importante deberia crear una nueva variable
2 # esto es para poder llamar a multcomp para comparar contra el mejor
3 datos$interaccion = factor(paste(f1, f2))
4 attach(datos)
5 m2 = aov(mido~interaccion, data=datos)
6 summary(m2)
7 # el estimador de la varianza es 3.23 con 12 grados de libertad

```

The following objects are masked from datos (pos = 8):

f1, f2, mido

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
interaccion	5	305.25	61.05	18.91	2.57e-05 ***
Residuals	12	38.75	3.23		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

In [11]:

```
1 # Ahora hacemos la comparacion con el MEJOR, pero OJO, NO PODEMOS USARLO PORQUE
2 # ENTONCES NO LO PODEMOS USAR AL FINAL PORQUE ES DESBALANCEADO, el profe se dio
3 source("/home/emiliano/EstadisticaAplicada/practica/mymultcomp.R")
4 maxHSU(mido, interaccion, alpha=0.05, 3.23,12)
```

```
[1] "WARNING: esta funcion considera que todos los ni son iguales"
```

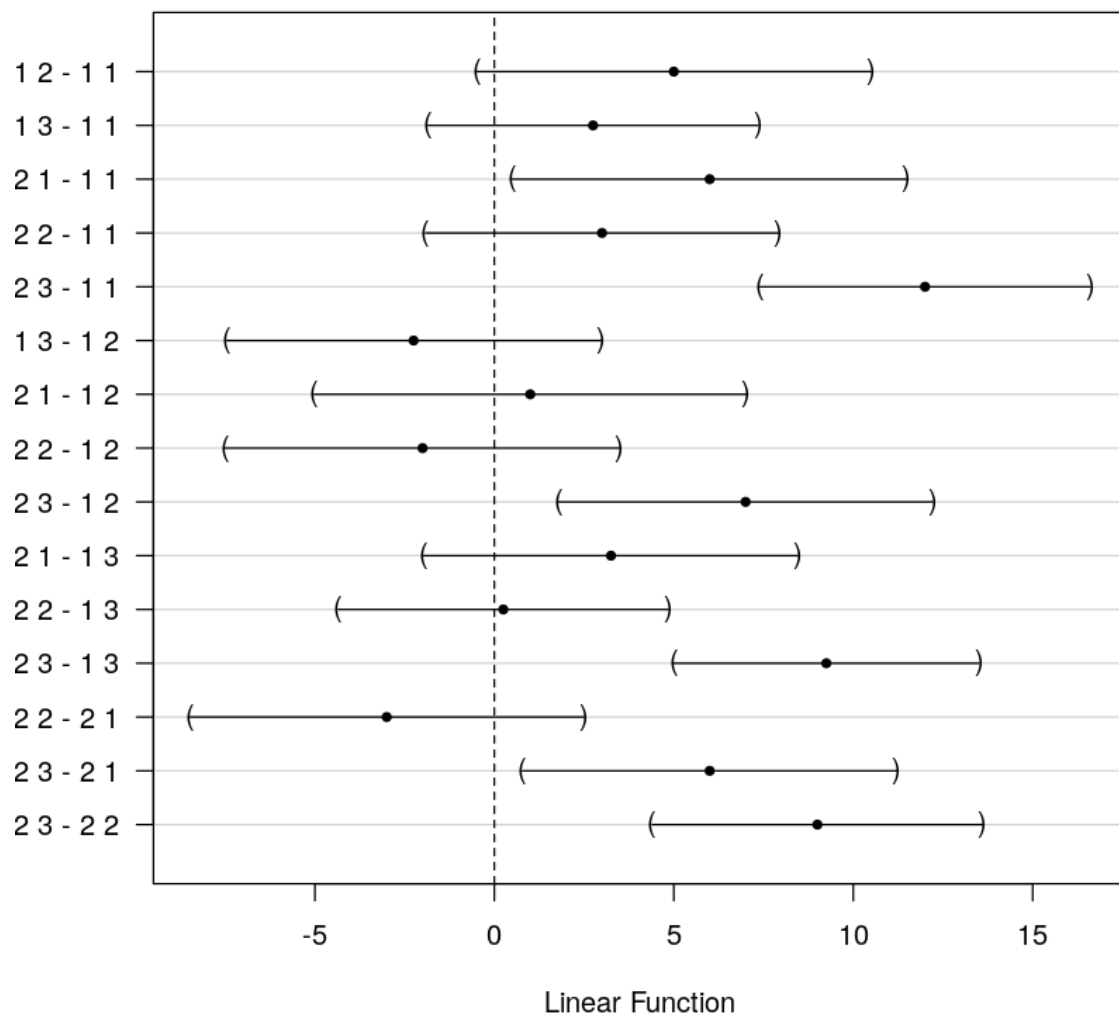
```
[1] "2 3"
```

```
NA '2 3'
```

In [14]:

```
1 # si podemos usar tukey para comparaciones multiples
2 library(multcomp)
3 mc = glht(m2, linfct=mcp(interaccion="Tukey"))
4 plot(mc)
5 # vemos los intervalos teniendo en cuenta si incluyen o no el 0
```

95% family-wise confidence level



In []:

```
1 # vemos que las combinaciones que no contienen el 0 son significativamente dist
2 # en el eje y, tenemos para el 1er caso x ej, (factor 1 es la dosis y el segund
3 # dosis 1 con fungicida 2, y dosis 1 con fungicida 1
4 # como la interaccion es significativa, no lo podes analizar al fungicida sin c
```