1 Guía ANOVA factorial

(x emilopez)

Ejercicio 2. (Ejercicio 6.3, Kuehl). Se llevó a cabo un estudio del efecto de la temperatura sobre el porcentaje de encogimiento de telas teñidas, con dos réplicas para cada uno de cuatro tipos de tela en un diseño totalmente aleatorizado. Los datos son el porcentaje de encogimiento de dos réplicas de tela secadas a cuatro temperaturas.

- Guía `Practica_Anova_Factorial.pdf`
- Datos: `C6P3datos.txt`

A) Utilice un gráfico exploratorio adecuado para dar una respuesta preliminar a la pregunta de interés.

Para realizar este análisis primeramente vemos un gráfico de análisis de interacción con `interaction.plot()` . Se puede observar que como las rectas se cortan, entonces tenemos evidencia que los factores no son independientes como para analizarlos por separado.

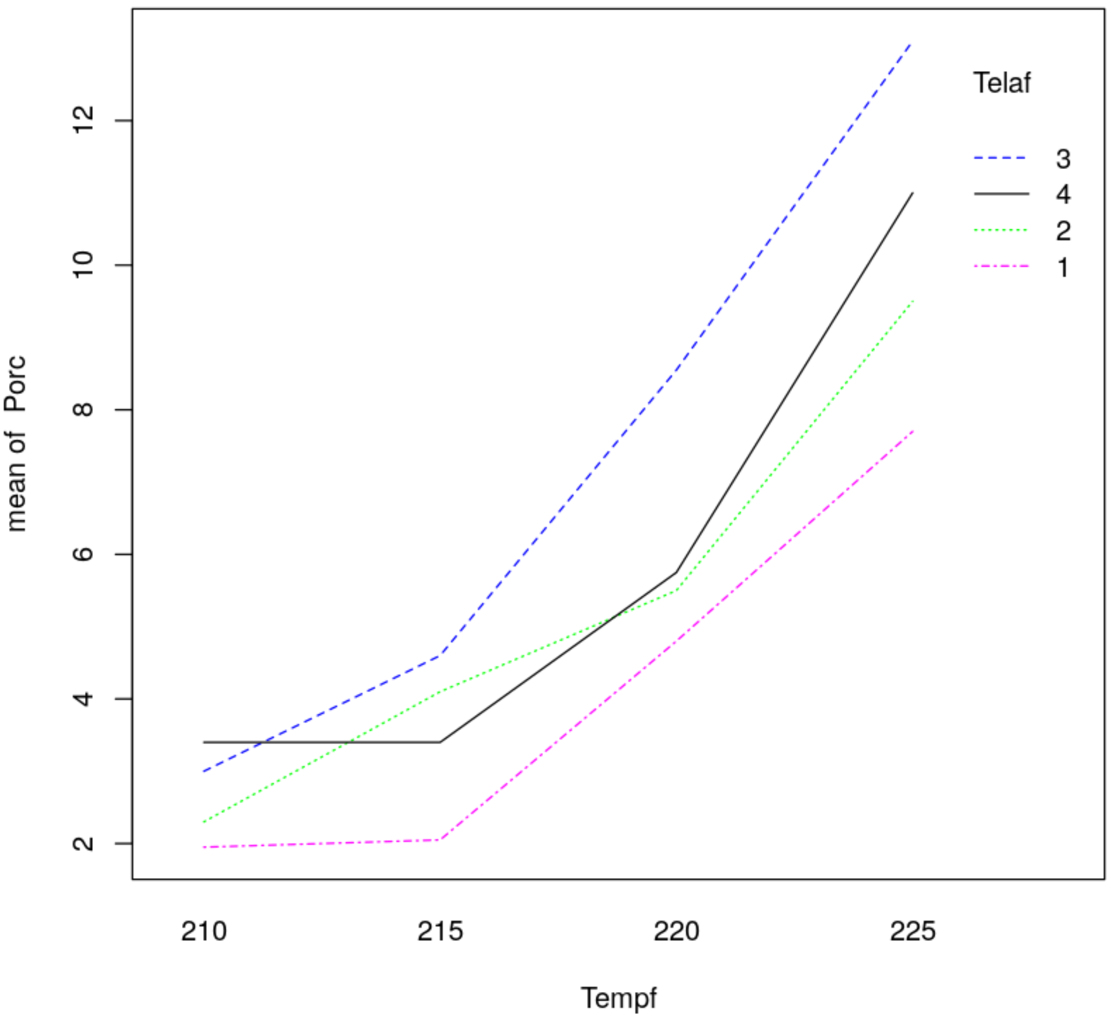
In [3]:

```
# lectura de datos
datos = read.table("/home/emiliano/EstadisticaAplicada/Estadistica.Aplicada.201
#head(datos)
attach(datos)
# nos aseguramos de hacerlos factor
Telaf = as.factor(Tela)
Tempf = as.factor(Temp)
```

```
The following objects are masked from datos (pos = 3):

    Porc, Tela, Temp

The following objects are masked from datos (pos = 4):

    Porc, Tela, Temp
```

In [3]:

```r
# Vemos un gráfico de interaccción
library(lattice)
interaction.plot(Tempf, Telaf, Porc, col=c('magenta', 'green', 'blue', 'black')
```



B) El modelo estadístico es:

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

donde μ es la media general, α_i es el efecto debido al i-ésimo nivel del fator A, β_j es el efecto j-ésimo nivel del factor B, $(\alpha\beta)_{ij}$ representa el efecto de la interacción en la combinación ij y ϵ_{ijk} es el error aleatorio

C) Las suposiciones que se siguieron sobre el error aleatorio (residuos) es que:

- sigue una distribución normal con media cero
- varianza constante σ^2
- independientes entre sí.

D) Analice la validez de los supuestos.

Hacemos el anális de ANOVA teniendo en cuenta la interacción de los factores, vemos el p-valor de Telaf:Tempf es pequeño (7.09e-09), por lo que podemos asumir que la interacción es significativa.

In [4]:

```
modelo = aov(Porc~Telaf*Tempf)
summary(modelo)
```

```
            Df Sum Sq Mean Sq F value   Pr(>F)    
Telaf        3  41.88   13.96  279.18 5.05e-14 ***
Tempf        3 283.94   94.65 1892.91  < 2e-16 ***
Telaf:Tempf  9  15.86    1.76   35.24 7.09e-09 ***
Residuals   16   0.80    0.05                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Ahora, evaluarmos los supuestos:

- *Test de Shapiro* para evaluar normalidad
- *Test de Levene* para evaluar varianzas constantes

Veremos que si no podemos asegurar su normalidad entonces debemos hacer una transformación.

In [5]:

```
shapiro.test(modelo$residuals) # no puedo arreglar mucho teniendo precaucion pq
library(car)
leveneTest(modelo) # no es confiable el test por los pocos datos
```

```
	Shapiro-Wilk normality test

data:  modelo$residuals
W = 0.93155, p-value = 0.04327

Loading required package: carData
Warning message in anova.lm(lm(resp ~ group)):
“ANOVA F-tests on an essentially perfect fit are unreliable”
```

	Df	F value	Pr(>F)
group	15	5.265436e+28	1.258744e-226
	16	NA	NA

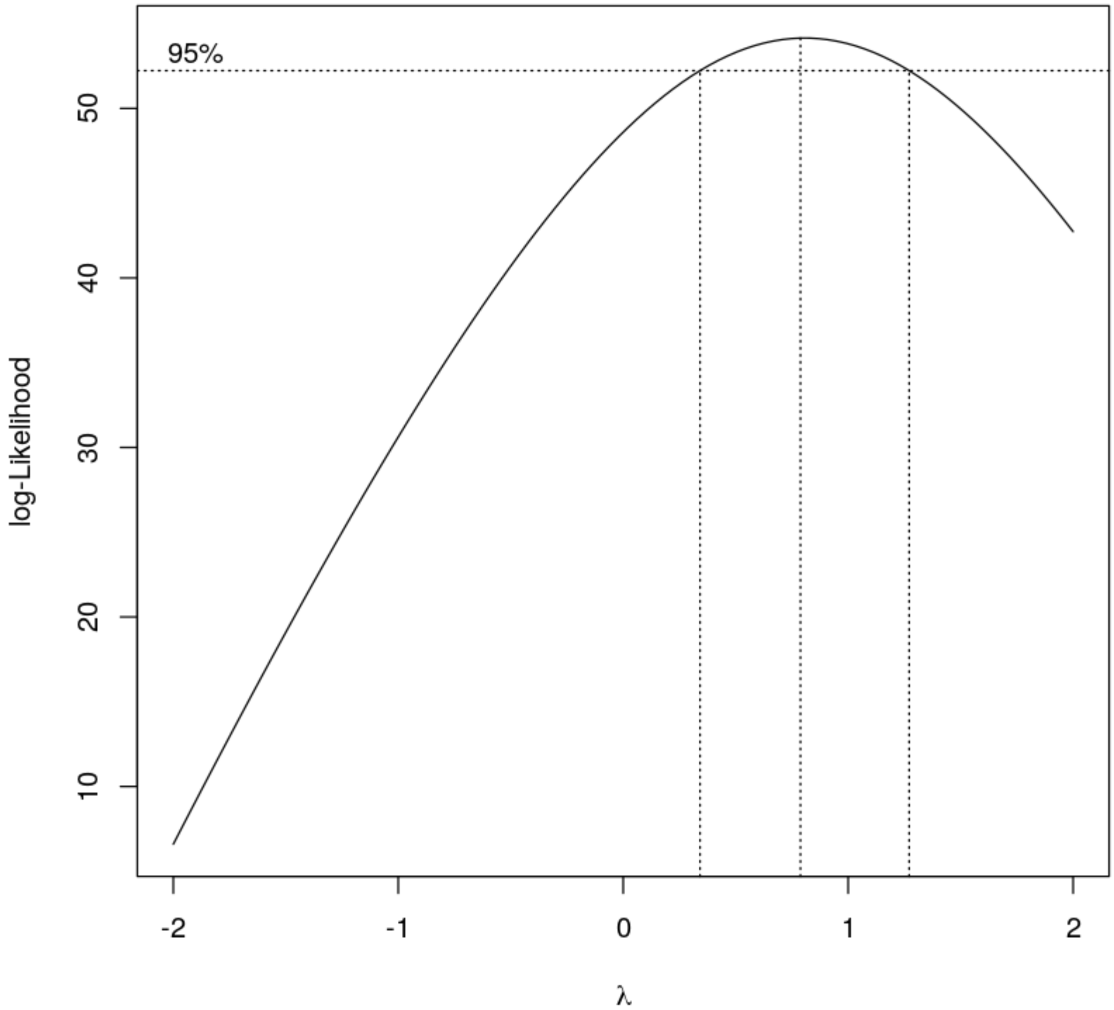
- Vemos que Shapiro nos da un pvalor apenas menor que la significancia, así que no podemos estar tan seguros de la normalidad, debemos tener esta precaución porque no la tenemos **tan** asegurada a la normalidad.
- Por otro lado, el test de levene no es confiable por la poca cantidad de datos.

La cuestión entonces es si nos conviene hacer alguna transformación para normalizarlo, así que vemos qué nos da **BOXCOX** y evaluamos si nos conviene transformar.

- NO TRANSFORMAMOS si en la banda de confianza se encuentra el 1.
- TRANSFORMAMOS si en la banda de confianza no está el 1.

In [6]:

```
library(MASS)
boxcox(modelo) # me dice que no puedo mejorar mucho, no puedo hacer transformac
# entonces, o bien aceptamos como esta o vamos por otro camino porque ninguna t
# aplicar a estos datos para que anova me quede mejor condicionado
```



Vemos que el **1 está en la banda de confianza** por lo tanto no vamos a transformar ya que no mejoraremos demasiado.

F) Realice un análisis de tendencias mediante contrastes ortogonales para el factor Temperatura.

Esto se hace cuando tenemos alguno de los factores cuantitativos, en este caso la temperatura. Lo que queremos ver es si hay una relación lineal, cuadrática, cúbica, etc. ¿Cómo estudiamos esta relación?

Con **contrastes ortogonales**. Esto lo que hace es descomponer los resultados con contrastes ortogonales que me indiquen si son lineales, cuadráticos o cúbicos.

In [7]:

```r
#f) aceptamos enotnces como esta, ahora vamos a hacer el analisis de tendencia
g = 4
# usamos c() con los valores de temp si no esta equispaciado, aca no hacia falt
# esto construye los contrastes ortogonales ya por definicion
contrasts(Tempf) = contr.poly(g, scores = c(210,215,220,225))
modelo = aov(Porc~Telaf*Tempf)
summary.lm(modelo)
```

```
Call:
aov(formula = Porc ~ Telaf * Tempf)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.3000 -0.1125  0.0000  0.1125  0.3000 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)     4.12500    0.07906  52.178  < 2e-16 ***
Telaf2          1.22500    0.11180  10.957 7.59e-09 ***
Telaf3          3.18750    0.11180  28.510 3.82e-15 ***
Telaf4          1.76250    0.11180  15.764 3.62e-11 ***
Tempf.L         4.47214    0.15811  28.284 4.33e-15 ***
Tempf.Q         1.40000    0.15811   8.854 1.45e-07 ***
Tempf.C        -0.55902    0.15811  -3.536  0.00275 ** 
Telaf2:Tempf.L  0.67082    0.22361   3.000  0.00848 ** 
Telaf3:Tempf.L  3.18640    0.22361  14.250 1.64e-10 ***
Telaf4:Tempf.L  1.15158    0.22361   5.150 9.68e-05 ***
Telaf2:Tempf.Q -0.30000    0.22361  -1.342  0.19845    
Telaf3:Tempf.Q  0.07500    0.22361   0.335  0.74167    
Telaf4:Tempf.Q  1.22500    0.22361   5.478 5.06e-05 ***
Telaf2:Tempf.C  1.22984    0.22361   5.500 4.85e-05 ***
Telaf3:Tempf.C  0.16771    0.22361   0.750  0.46414    
Telaf4:Tempf.C  0.68200    0.22361   3.050  0.00764 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2236 on 16 degrees of freedom
Multiple R-squared:  0.9977,	Adjusted R-squared:  0.9955 
F-statistic: 455.6 on 15 and 16 DF,  p-value: < 2.2e-16
```

Vemos que el cubico es significativo, estamos testeando 15 cosas al mismo tiempo. Fijemos un α para comparar, usando Bonferroni $\alpha_{PC} = 1 - (1 - \alpha_{0QueQuiero})^1/15$, donde $\alpha_{0QueQuiero} = 0.05$.

Veo que no me puedo sacar de encima el cubico, ya que los p-valores que son menores que el α_{PC} entonces son los que tengo que mirar en la tabla previa (en la tabla: los que terminan en `.C` son los cúbicos, `.Q` cuadráticos y `.L` , lineal).

In [9]:

```
alpha = 0.05
(alpha_PC = 1 - (1- alpha)^(1/15))
# 0.003413
```

0.00341371294659032

Entonces, resumiendo:

- Vimos que la interacción es significativa
- Sabemos que no podemos mejorar el modelo, esto nos dijo **BoxCox**
- Vimos la tendencia, ahora sabemos que la relación respecto a la temperatura es cúbica

Ahora, como tenemos interacción, tenemos que analizarla, para eso creamos una nueva variable que represente la interacción (`factor(paste0(F1, F2))`), y comparar todos contra el mejor usando `minHSU` o `maxHSU` (ojo que solo se puede usar si son balanceados). Un detalle es que al mejor no lo conocemos, lo encuentra la misma función `minHSU` / `maxHSU` .

In [10]:

```
# crea la variable interaccion que tiene la combicacion de Telaf y Tempf
datos$interaccion = factor(paste0(Telaf, Tempf))
attach(datos)
```

```
The following objects are masked from datos (pos = 7):

    Porc, Tela, Temp
```

In [11]:

```
# volvemos a ajustar, para luego ver los intervalos de confianza con lsmeans
m2 = aov(Porc~interaccion)
summary(m2)
```

```
            Df Sum Sq Mean Sq F value Pr(>F)    
interaccion 15  341.7   22.78   455.6 <2e-16 ***
Residuals   16    0.8    0.05                   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Con la variable `interaccion` invocamos a `minHSU` para que encuentre el mejor, que para el caso sería el que menos se encoje, y aquellos que son equivalentes al mejor. El resultado nos dice que la tela 1 a 210 °C es el mejor, pero que este es equivalente a la tela 1 a 215 °C y a la tela 2 a 210°C.

In [15]:

```
# OJO, ESTO DEMORA
source("/home/emiliano/EstadisticaAplicada/practica/mymultcomp.R")
minHSU(Porc, interaccion, alpha=0.05, 0.05, 16) # el mejor es el q encoje poco
# este nos dice que el mejor es 1210 y solamente son equivalentes a 1215 y 2210
# si uno usara tukey probablemente encontraría alguno mas porque tiene menos po
```

```
[1] "WARNING: esta funcion considera que todos los ni son iguales"
[1] "1210"
[1] "1215"
[1] "2210"
```

NA '1210' '1215' '2210'

Podemos ver los IC para corroborar este mismo resultado previo.

In [4]:

```
# usando el ajuste aov, ahora vemos los IC
library(lsmeans)
medias = lsmeans(m2, ~interaccion)
plot(medias)
```

```
The 'lsmeans' package is being deprecated.
Users are encouraged to switch to 'emmeans'.
See help('transition') for more information, including how
to convert 'lsmeans' objects and scripts to work with 'emmeans'.
```