ResumR

```
db <- read.csv2('COL.csv')
```

Regression and lm

Estadística descriptiva

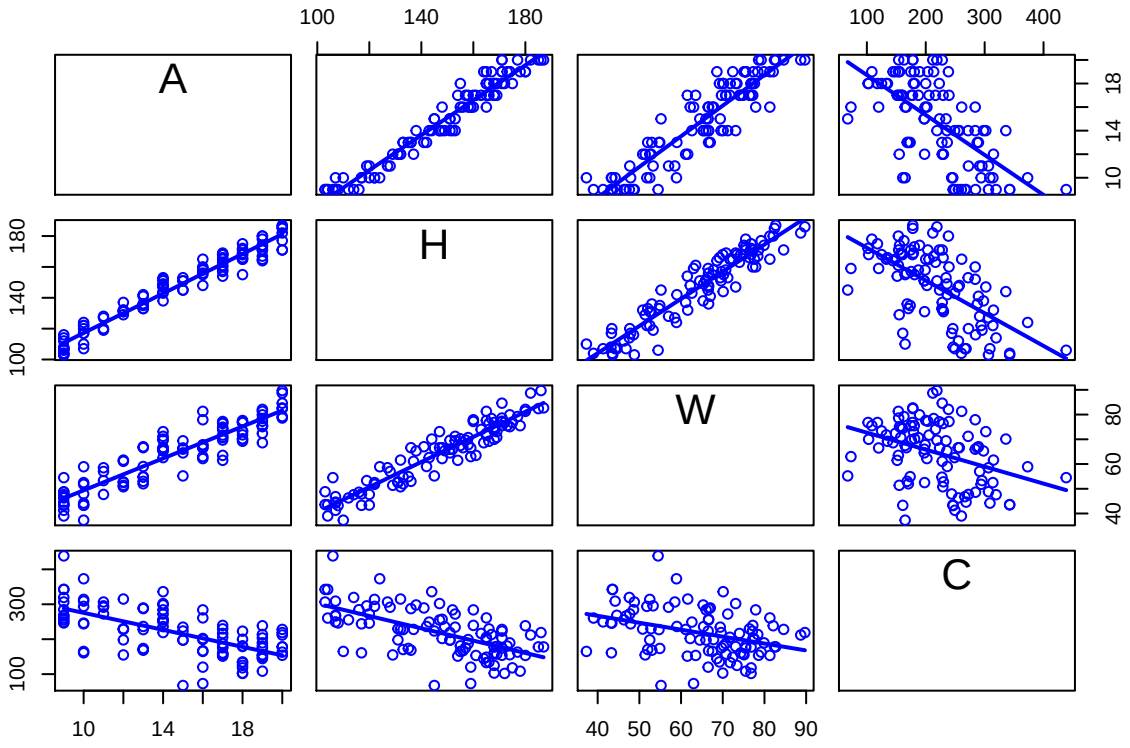
Per veure format dades

```
head(db)
```

```
##    A   H    W     C
## 1 19 174 79.9 189.5
## 2 15 151 64.5 197.5
## 3 13 133 52.0 170.5
## 4 19 173 75.5 180.5
## 5 17 163 74.0 216.5
## 6 13 135 54.9 173.5
```

Plot de totes les variables respecte totes i correlation matrix:

```
scatterplotMatrix(db, diagonal = F, smooth = F)
```



```r
cor(db)
```

```
##            A          H          W          C
## A  1.0000000  0.9755923  0.9159378 -0.6424197
## H  0.9755923  1.0000000  0.9453963 -0.6118937
## W  0.9159378  0.9453963  1.0000000 -0.3690117
## C -0.6424197 -0.6118937 -0.3690117  1.0000000
```

Sembla ser que les 3 variables explicatives estan bastant correlacionades entre sí.

Model lineal (regression line)

En primer lloc probem un model lineal senzill

```r
m1 <- lm(C~A+H+W, data = db)
summary(m1)
```

```
##
## Call:
## lm(formula = C ~ A + H + W, data = db)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -74.608 -22.137   1.888  21.156  65.410
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 490.9978    35.0517  14.008  < 2e-16 ***
## A           -13.0195     3.8530  -3.379  0.00105 **
## H            -5.0989     0.7227  -7.055 2.68e-10 ***
## W            10.3773     0.7365  14.090  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.11 on 96 degrees of freedom
## Multiple R-squared:  0.8101, Adjusted R-squared:  0.8041
## F-statistic: 136.5 on 3 and 96 DF,  p-value: < 2.2e-16
```

Per comprobar que no hi hagi cap problema de multicolinearitat:

```r
vif(m1)
```

```
##         A         H         W
## 20.904776 31.695499  9.489406
```

Sembla ser que el VIF de l'alçada i el pes és bastant alt, probem d'eliminar l'edat de les variables ja que és la menys significativa de les dos amb vif elevat:

```r
m2 <- lm(C~H+W, data = db)
summary(m2)
```

```
##
## Call:
## lm(formula = C ~ H + W, data = db)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -69.703 -26.437   1.281  21.041  83.838
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 586.8882    21.6514   27.11   <2e-16 ***
## H            -7.1466     0.4145  -17.24   <2e-16 ***
## W            10.5993     0.7719   13.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.69 on 97 degrees of freedom
## Multiple R-squared:  0.7875, Adjusted R-squared:  0.7831
## F-statistic: 179.7 on 2 and 97 DF,  p-value: < 2.2e-16
```

Tornem a comprobar el vif:

```
vif(m2)
```

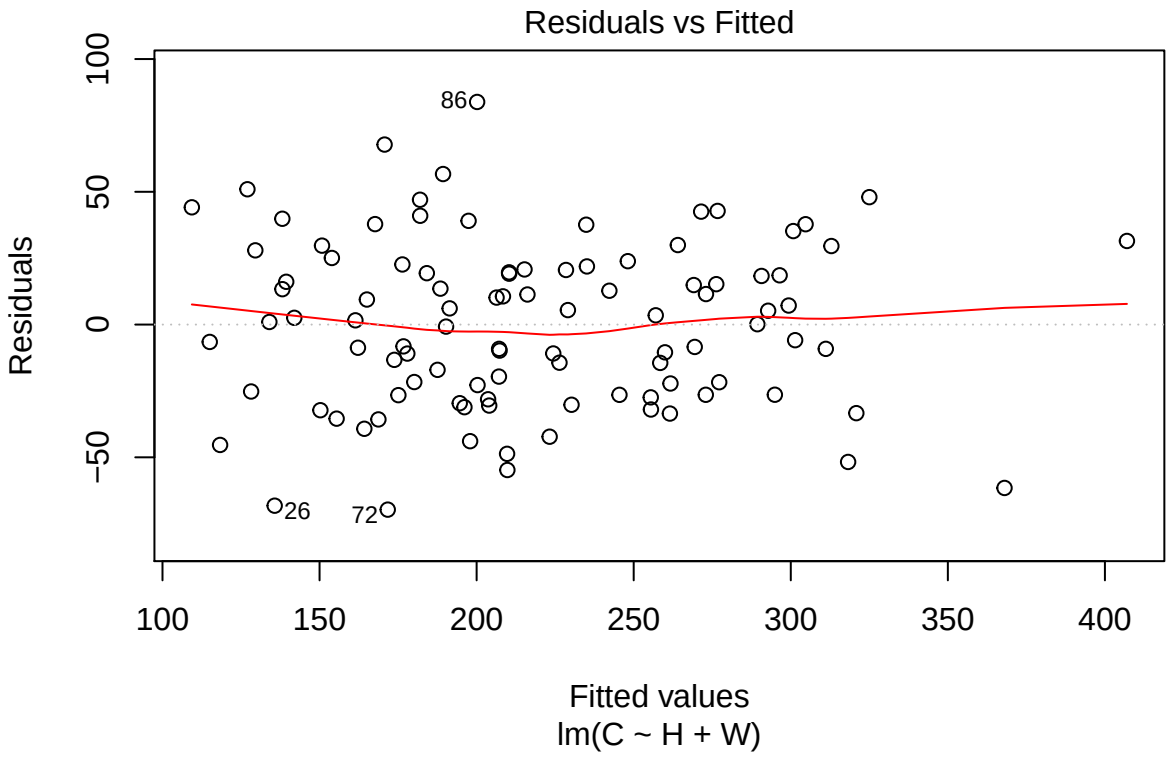
```
##        H        W
## 9.413912 9.413912
```

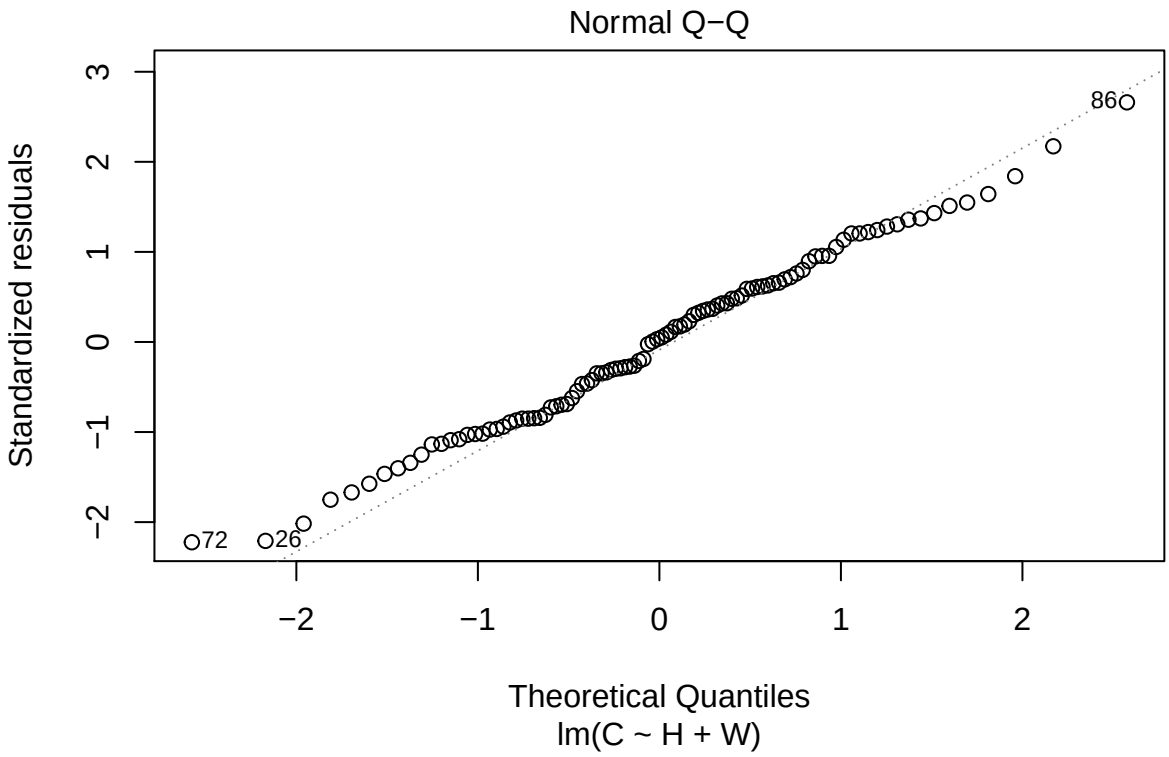
Sembla que ara el VIF és prou baix en les dues variables i el percentatge de variabilitat explicada ha disminuit molt poc. Analitzem ara el model:

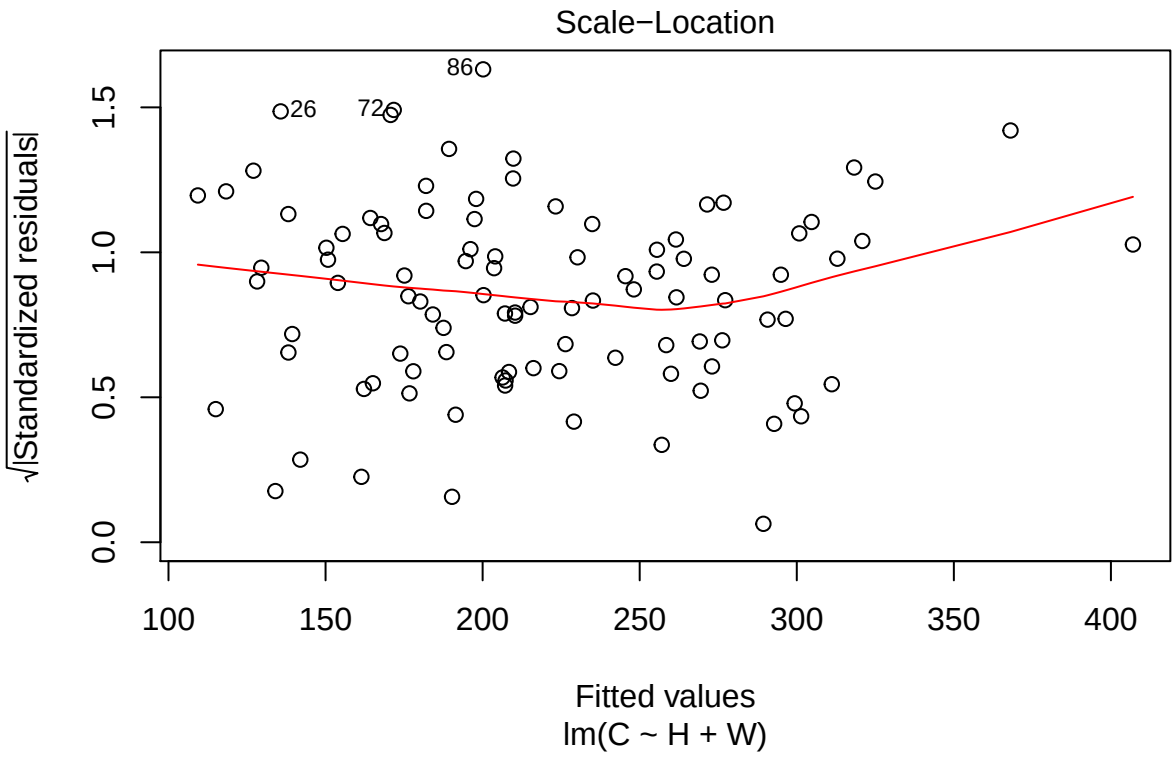
Residuals vs fitted

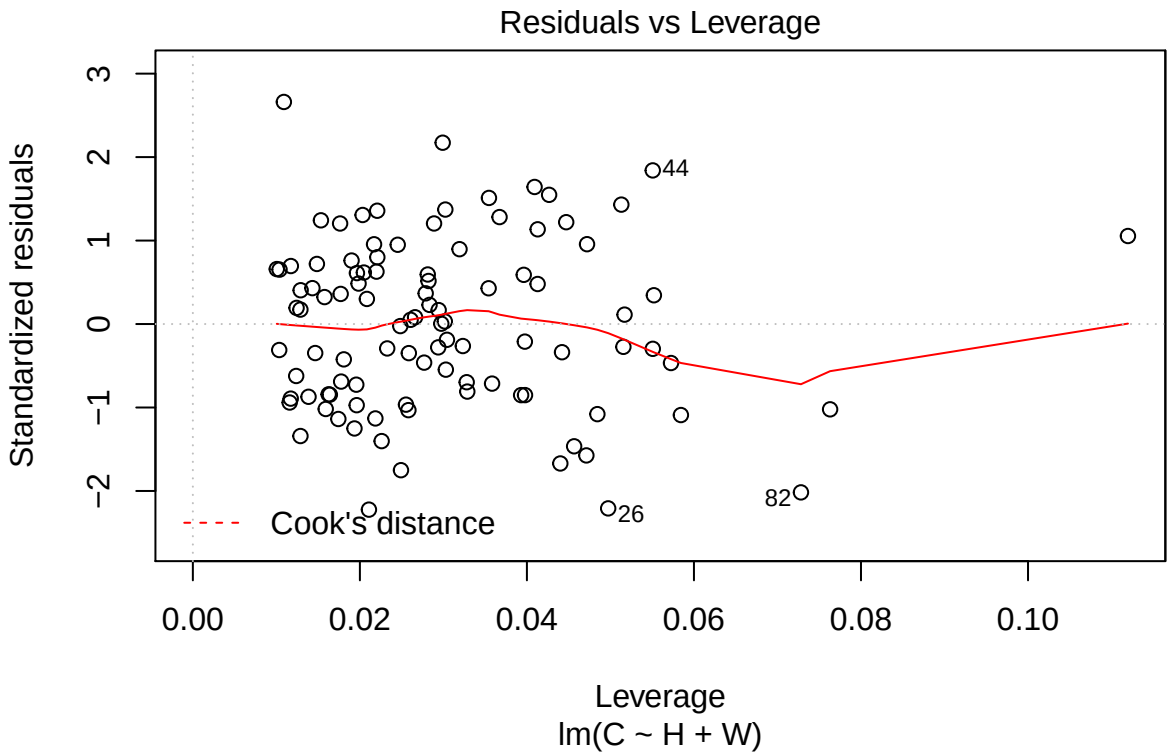
Per observar els residuals vs els fitted plotem el model

```
plot(m2)
```


Residuals vs Fitted
Residuals
100
50
0
−50
86
26
72
100
150
200
250
300
350
400
Fitted values
lm(C ~ H + W)


Normal Q−Q
Standardized residuals
3
2
1
0
−1
−2
72
26
86
−2
−1
0
1
2
Theoretical Quantiles
lm(C ~ H + W)


Scale−Location
86
26
72
√|Standardized residuals|
0.0
0.5
1.0
1.5
100
150
200
250
300
350
400
Fitted values
lm(C ~ H + W)



No s'observem patrons en els residus i sembla que hi ha homocedasticity, per assegurar-ho fem un Levene test:

```
leveneTest(m2$residuals ~ as.factor(H), data = db)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group 61  1.2206 0.2578
##       38
```

```
leveneTest(m2$residuals ~ as.factor(W), data = db)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group 87  0.8177 0.7193
##       12
```

Efectivament, respecte les dues variables del model la variança sembla constant. (No hi ha evidencies estadístiques que ens permetin rebutjar homogeneïtat en la variança)

Normalitat

En el Q-Q plot s'observa linealitat (sobretot pel centre), per assegurar que hi ha normalitat realitzem un shapiro test:

```
shapiro.test(m2$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  m2$residuals
## W = 0.9907, p-value = 0.7207
```

Sembla ser que hi ha normalitat.

També ho podem comprobar amb un chi-square test.

Significancia dels beta

```
anova(m2)
```

```
## Analysis of Variance Table
##
## Response: C
##           Df Sum Sq Mean Sq F value    Pr(>F)
## H          1 171564  171564  170.89 < 2.2e-16 ***
## W          1 189273  189273  188.53 < 2.2e-16 ***
## Residuals 97  97383    1004
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Efectivament son diferents de 0.

També podem veure-ho amb els intervals de confiança

```
confint(m2, lebel = 0.95)
```

```
##                   2.5 %     97.5 %
## (Intercept) 543.916164 629.860201
## H            -7.969231  -6.323902
## W             9.067182  12.131378
```

Cap dels dos inclou el 0.

Plot predicts

```
#ci.plot(m2) només amb una variable
```

Observacions influents

Busquem observacions amb leverage gran: (plot model) Sembla ser que les observacions 26, 44 i 82 podrien ser influents ja que tenen una mica de leverage i els residus son alts. Probablement no ho son ja que el leverage es mes petit que $3p/N$.

Lm ANCOVA

```
dbi <- read.csv2('Iogurt.csv')
```

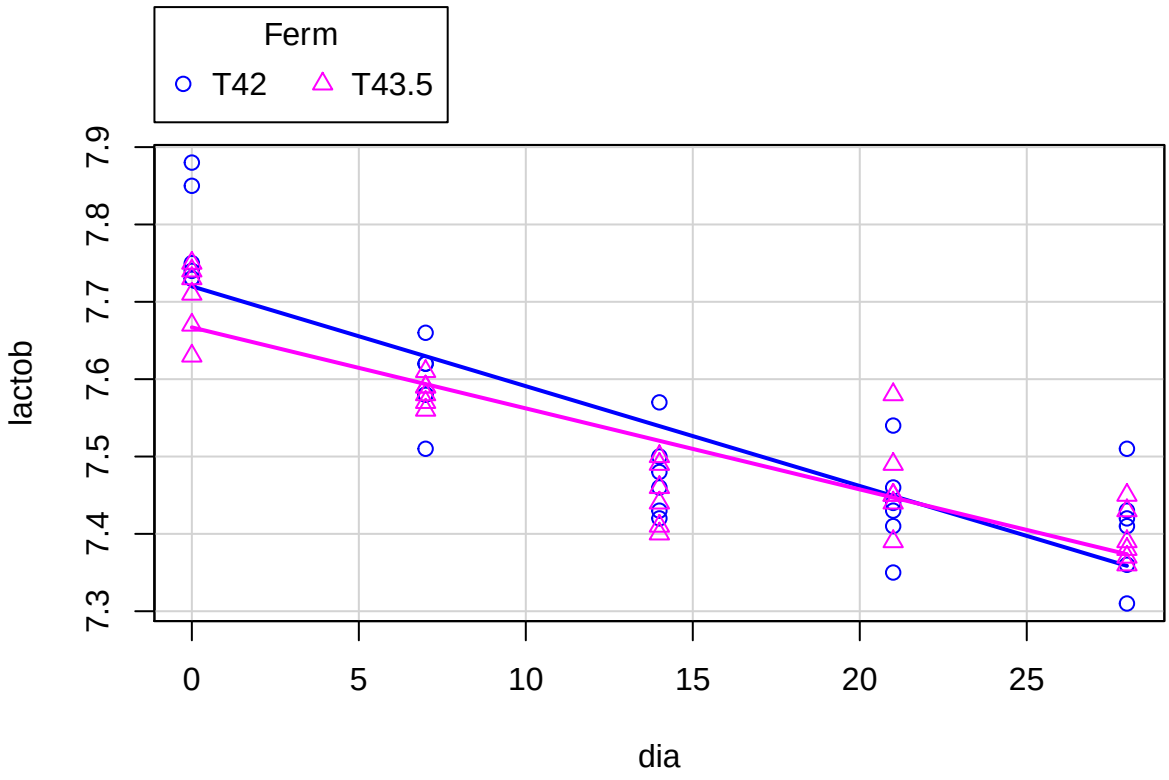
Estadistica descriptiva

```
head(dbi)
```

```
##   Ferm dia   pH strep lactob
## 1  T42  21 4.10  7.43   7.46
## 2  T42   0 4.44  7.65   7.75
## 3  T42  21 4.02  7.10   7.35
## 4  T42   7 4.24  7.54   7.62
## 5  T42   7 4.27  7.54   7.66
## 6  T42  28 4.01  7.25   7.41
```

Fem un plot dels lactobacilus en funcio del temps distingint segons la temperatura de fermentació:

```
sp(lactob ~ dia|Ferm, boxplot = F, smooth = F, data = dbi)
```



Podem observar diferencies en el pendent i l'intercept, per tant sembla bona opció afegir interacció al model.

Comrpobem la correlació entre ph i bacteris:

```
cor(dbi$lactob, dbi$pH)
```

```
## [1] 0.9417985
```

Per tant, fer un model pels lactobacilus o pel ph es gairebé el mateix.

Fem taules:

```
dbi$Fdia<-as.factor(dbi$dia)
tabular((pH+strep+lactob)*Ferm*((n=1)+mean+sd)~Fdia,dbi)
```

			Fdia				
	Ferm		0	7	14	21	28
pH	T42	n	6.00000	6.00000	6.00000	6.00000	6.00000
		mean	4.45000	4.23000	4.10167	4.05500	4.03333
		sd	0.02191	0.02683	0.04997	0.03886	0.04633
	T43.5	n	6.00000	6.00000	6.00000	6.00000	6.00000
		mean	4.35333	4.17167	4.10333	4.08667	4.02167
		sd	0.03141	0.02483	0.05785	0.03011	0.03312
strep	T42	n	6.00000	6.00000	6.00000	6.00000	6.00000
		mean	7.72167	7.46000	7.29500	7.24667	7.21167
		sd	0.06555	0.08854	0.11415	0.12644	0.13891
	T43.5	n	6.00000	6.00000	6.00000	6.00000	6.00000
		mean	7.63000	7.40167	7.26833	7.29667	7.20667
		sd	0.04382	0.08035	0.09042	0.04967	0.10013
lactob	T42	n	6.00000	6.00000	6.00000	6.00000	6.00000
		mean	7.78000	7.59500	7.47667	7.43833	7.40667
		sd	0.06693	0.05128	0.05465	0.06242	0.06772
	T43.5	n	6.00000	6.00000	6.00000	6.00000	6.00000
		mean	7.70500	7.58333	7.45000	7.46667	7.39667
		sd	0.04637	0.01751	0.04099	0.06408	0.03559

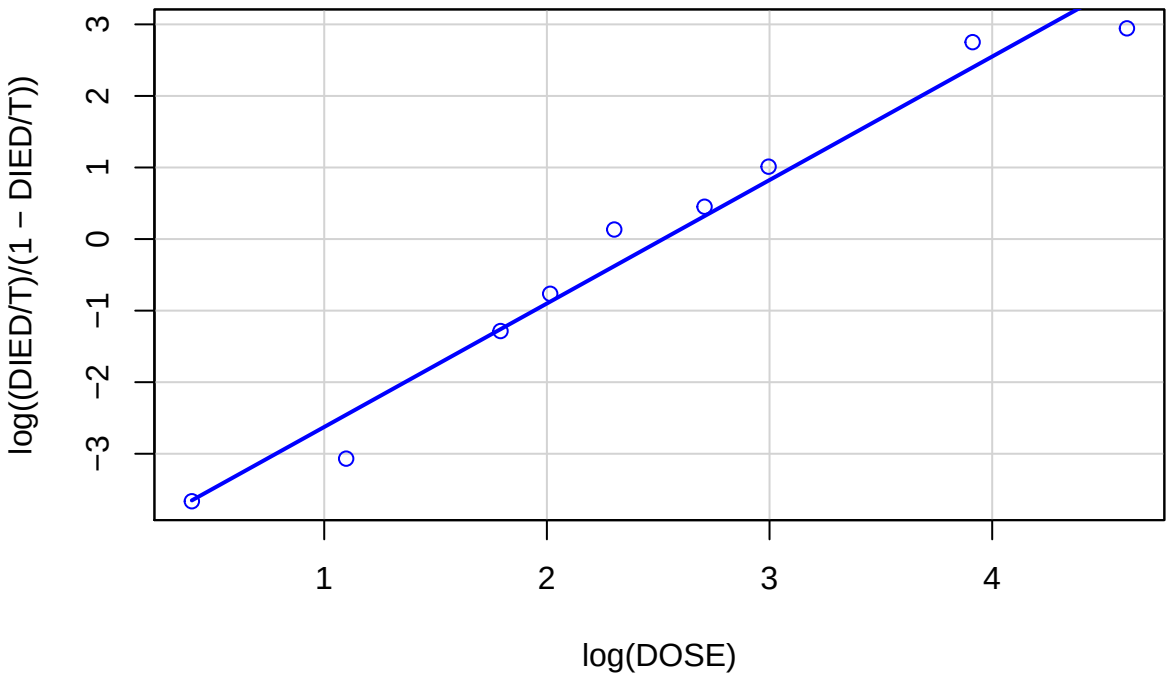
Altres comandes

```
#emm<-emmeans(model1,~DOSEFACTOR) # Mitjana separant per dosefactor
# pairs(emm) els compara un a un (Tukey)
# plot (emm, level = 9.99, adjust = "tukey")
# confint dona int de confiança
```

GLM

Binomial

```
insc <- read.csv2('insecticide.csv')
insc <- insc[-1,]
sp(log((DIED/T)/(1-DIED/T)) ~ log(DOSE), smooth = F, boxplot = F, data = insc) #canonic link en funcio
```



Apliquem glm resposta binomial en funcio de log(dosi)

```
glm1 <- glm(cbind(DIED, T) ~ log(DOSE), family = binomial, data = insc)
summary(glm1)
```

```
##
## Call:
## glm(formula = cbind(DIED, T) ~ log(DOSE), family = binomial,
##     data = insc)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9073  -2.3166  -0.3693   1.4167   1.9576
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.6439     0.2167 -12.202   <2e-16 ***
## log(DOSE)     0.6843     0.0733   9.336   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 130.060  on 8  degrees of freedom
## Residual deviance:  29.286  on 7  degrees of freedom
## AIC: 73.358
##
```

```
## Number of Fisher Scoring iterations: 4
```

Pearson residuals:

```
pres <- rstandard(glm1, type = "pearson")
X2 <- sum(pres^2)
phi = X2/glm1$df.residual
```

Per tant caldria canviar el dispersion parameter. En aquest cas es millor usar el probit.