ResumR

```
## Loading required package: car
## Loading required package: carData
## Loading required package: sandwich
## Loading required package: grid
## Loading required package: latticeExtra
## Loading required package: RColorBrewer
## Loading required package: multcomp
## Loading required package: mvtnorm
## Loading required package: survival
## Loading required package: TH.data
## Loading required package: MASS
##
## Attaching package: 'TH.data'
## The following object is masked from 'package:MASS':
##
##     geyser
##
## Attaching package: 'multcomp'
## The following object is masked from 'package:emmeans':
##
##     cld
## Loading required package: gridExtra
##
## Attaching package: 'HH'
## The following object is masked from 'package:emmeans':
##
##     as.glht
## The following objects are masked from 'package:car':
##
##     logit, vif
## Loading required package: Formula
## Loading required package: ggplot2
##
## Attaching package: 'ggplot2'
## The following object is masked from 'package:latticeExtra':
##
##     layer
##
## Attaching package: 'Hmisc'
```

```
## The following object is masked from 'package:RcmdrMisc':
##
##     Dotplot

## The following objects are masked from 'package:base':
##
##     format.pval, units

##
## Attaching package: 'tables'

## The following object is masked from 'package:latexpdf':
##
##     as.tabular
```

```
setwd("~/Desktop/UNI/3rcurs/1rquatr/pie/dades/")
db <- read.csv2('COL.csv')
```

Regression and lm

Estadística descriptiva

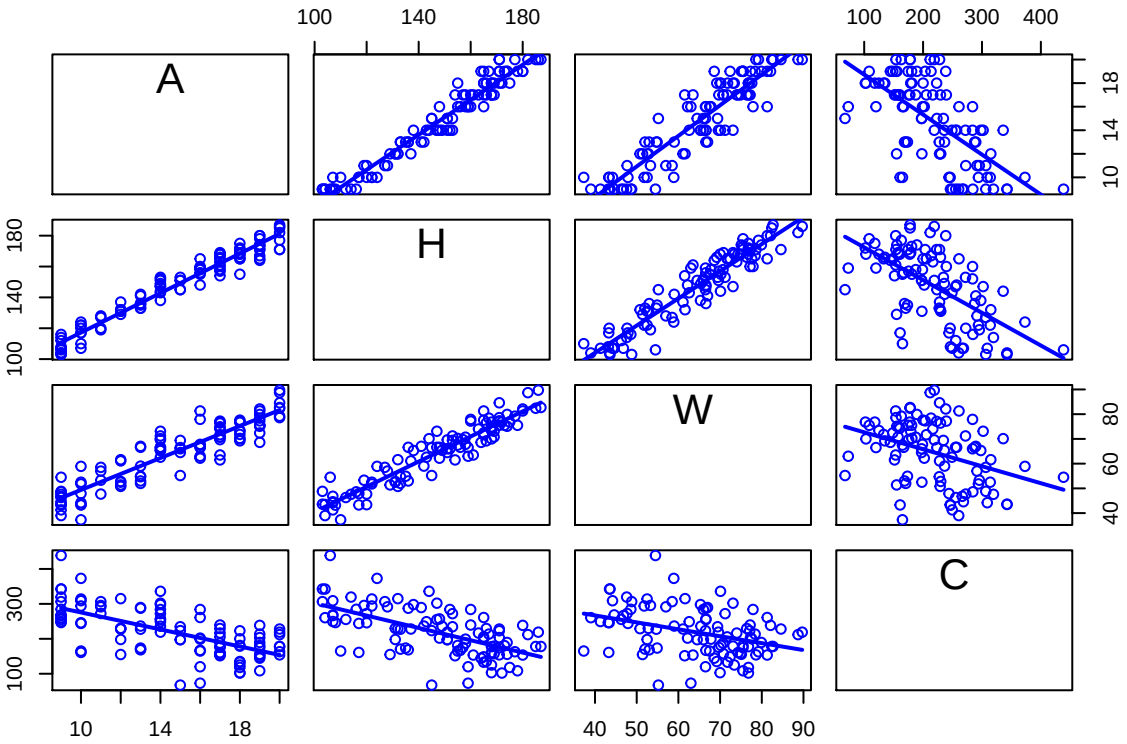
Per veure format dades

```
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)
```




```
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

```
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
```

Multicolinealitat (VIF)

Per comprobar que no hi hagi cap problema de multicolinearitat:

```
vif(m1)
```

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

Sembla ser que el VIF de l'alçada i l'edat és bastant alt, probem d'eliminar l'edat de les variables ja que és la menys significativa de les dos amb vif elevat. També podríem eliminar la que tingui VIF més elevat. Si les variables són significatives totes, per molt que estiguin molt correlacionades no n'eliminem cap.

```
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
```

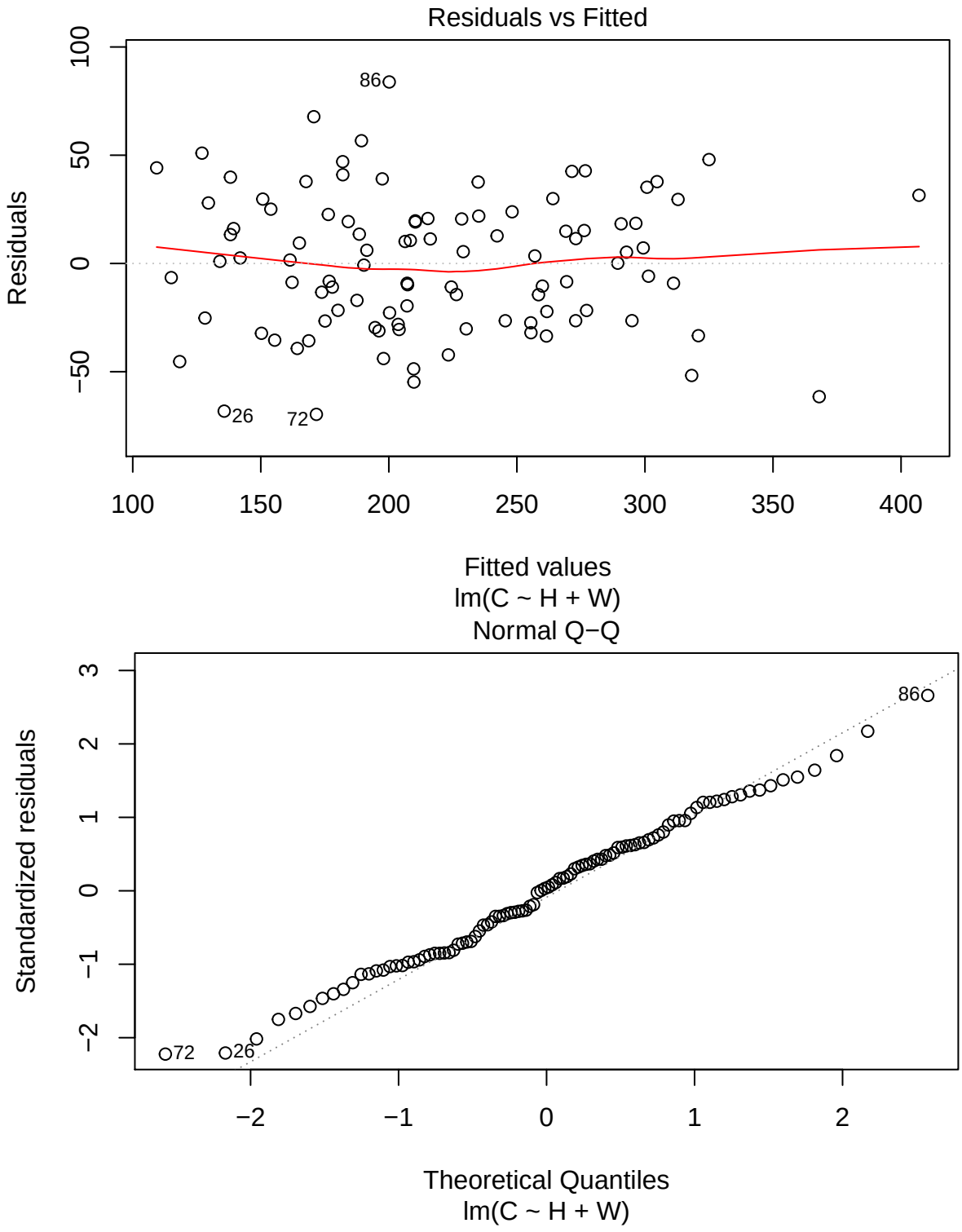
Sembla que ara el VIF és prou baix en les dues variables i el percentatge de variabilitat explicada ha disminuit molt poc.

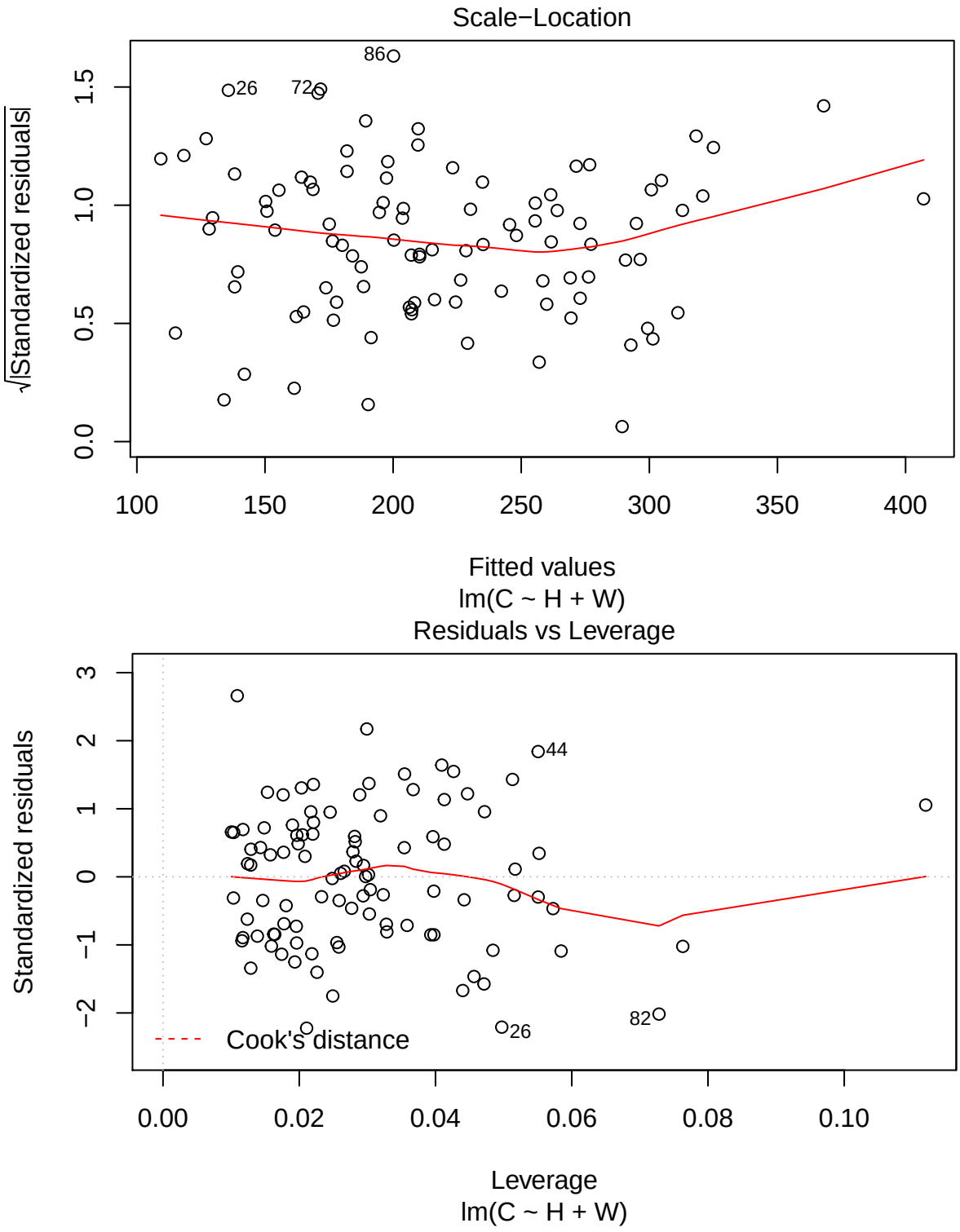
Analitzem ara el model:

Residuals vs fitted

Per observar els residuals vs els fitted plotem el model

```
plot(m2)
```



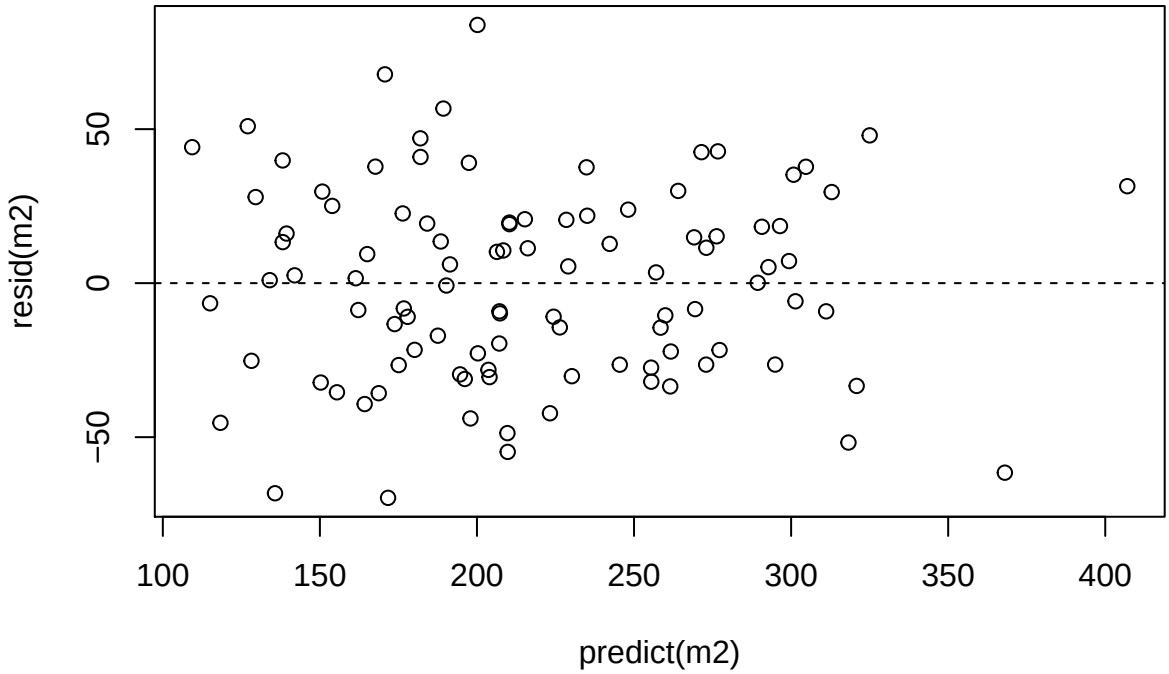


El que hem de mirar d'observar en el plot del model és el següent:

- Abans de res, en un *scatterplot de les y respecte les x* hauríem d'observar linealitat.
- En el scatterplot dels $\hat{e}_i$ vs. $\hat{y}_i$, és a dir, en el *Residuals vs Fittet* no hauríem de veure patrons. Si hi ha algun patró, el meu model s'està deixant de capturar alguna cosa. Els residus han de ser aleatoris i petits.

- En el *QQ-plot per a* $\hat{e}_i$ s'hauria d'observar linealitat, ja que estem comparant els quartils dels nostres residus amb els d'una normal $N(0,1)$.
- En el $\sqrt{\text{standarized residuals}}$* vs fittet* tampoc hauríem d'observar patrons.
- Pel que fa a la homocedasticitat, podem fer un *predicted vs residuals* plot i observar que la variabilitat és constant.
- En el plor dels *standarized residuals vs Leverage* busquem les observacions influents, que seran aquelles que tinguin alt residus i alt Leverage (26, 44, 82) en aquest cas. Tot i així, per ser considerades influents, han de tenir un leverage major que $3p/n$

```
plot(predict(m2), resid(m2))
abline(h=0, lty=2)
```



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, donat que el Pr(>F) és major que 0.05 en ambdós casos, podem afirmar que 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
```

```
Anova(m2)
```

```
## Anova Table (Type II tests)
##
## Response: C
##           Sum Sq Df F value    Pr(>F)
## H         298441  1  297.27 < 2.2e-16 ***
## W         189273  1  188.53 < 2.2e-16 ***
## Residuals  97383 97
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Efectivament son diferents de 0, ja que el Pr(>F) és major que 0.05 en tots els casos. En el test anova, les sumes dels quadrats depenen de l'ordre en què els factors s'han entrat al model. En canvi, en el test Anova, l'ordre no es té en compte.

Per veure si les covariants són estadísticament significatives, podem observar-ho amb els intervals de confiança de les β's, mirant si contenen el 0 o no. Si el contenen, no són estadísticament significatives.

```
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

Per veure els intervals de confiança dels valors predits:

```
C0<-data.frame(cbind(W=c(65,75,65),A=c(15,15,12),H=c(150,150,150)), row.names=1:3)
predict(m1, C0, interval="confidence", level=.95, se.fit=T)
```

```
## $fit
##        fit      lwr      upr
## 1 205.3908 199.1668 211.6148
## 2 309.1639 294.6188 323.7089
## 3 244.4492 219.8210 269.0774
##
## $se.fit
##         1         2         3
##  3.135539  7.327533 12.407261
##
## $df
## [1] 96
##
## $residual.scale
## [1] 30.1094
```

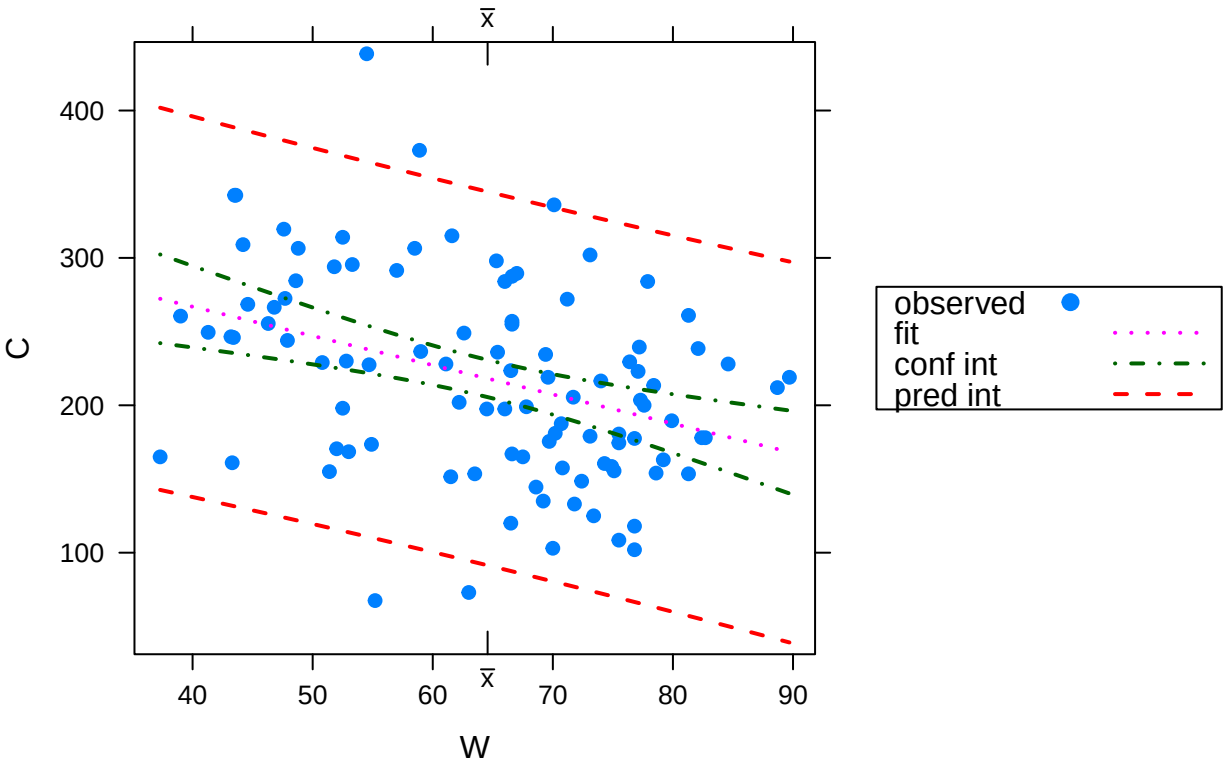
On podem veure, en primer lloc una matriu amb una fila amb el valor predit per les dades donades, i l'interval de confiança corresponent.

Per veure els intervals de predicció dels valors predits:

```
C0<-data.frame(cbind(W=c(65,75,65),A=c(15,15,12),H=c(150,150,150)), row.names=1:3)
predict(m1, C0, interval="prediction", level=.95, se.fit=T)
```

```
## $fit
##        fit      lwr      upr
## 1 205.3908 145.3009 265.4807
## 2 309.1639 247.6528 370.6749
## 3 244.4492 179.8071 309.0914
##
## $se.fit
##         1         2         3
##  3.135539  7.327533 12.407261
##
## $df
## [1] 96
##
## $residual.scale
## [1] 30.1094
```

```
m3 <- lm(C~W, db)
ci.plot(m3) #només amb una variable
```



Estimació de la variancia (S)

Si ens demanen a un model lineal estimar la S per a uns certs valors en concret, donat que per defició de model lineal es té homocedasticitat, es a dir, que $\sigma_i^2 = \sigma^2$, llavors la variància estimada serà:

```
summary(m2)$sigma
```

```
## [1] 31.68507
```

Estimació de esperança i variància (standard deviation) amb modificació de la variable resposta

Si a la nostra variable resposta H li hem aplicat una modificació a través d'una funció $g(H)$, llavors hem de tenir en compte que, si $f(H) = g^{-1}(H)$,

$m = \hat{E}(g(H)|Days = a)$ and $s = \sqrt{\hat{Var}(g(H)|Days = a)}$

$E(H|Days = a) = f(m)$ and $\sqrt{Var(H|Days = a)} = s$ů$|f'(m)|$

anova de dos models

Ens hem de mirar si el Pf(>F) és més gran que 0.05. En cas afirmatiu, es rebutja la H_0, és a dir, es rebutja el first_model.

```
#anova(first_model, second_model, test=F)
```

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

```
setwd("~/Desktop/UNI/3rcurs/1rquatr/pie/dades/")
dbi <- read.csv2('Iogurt.csv')
```

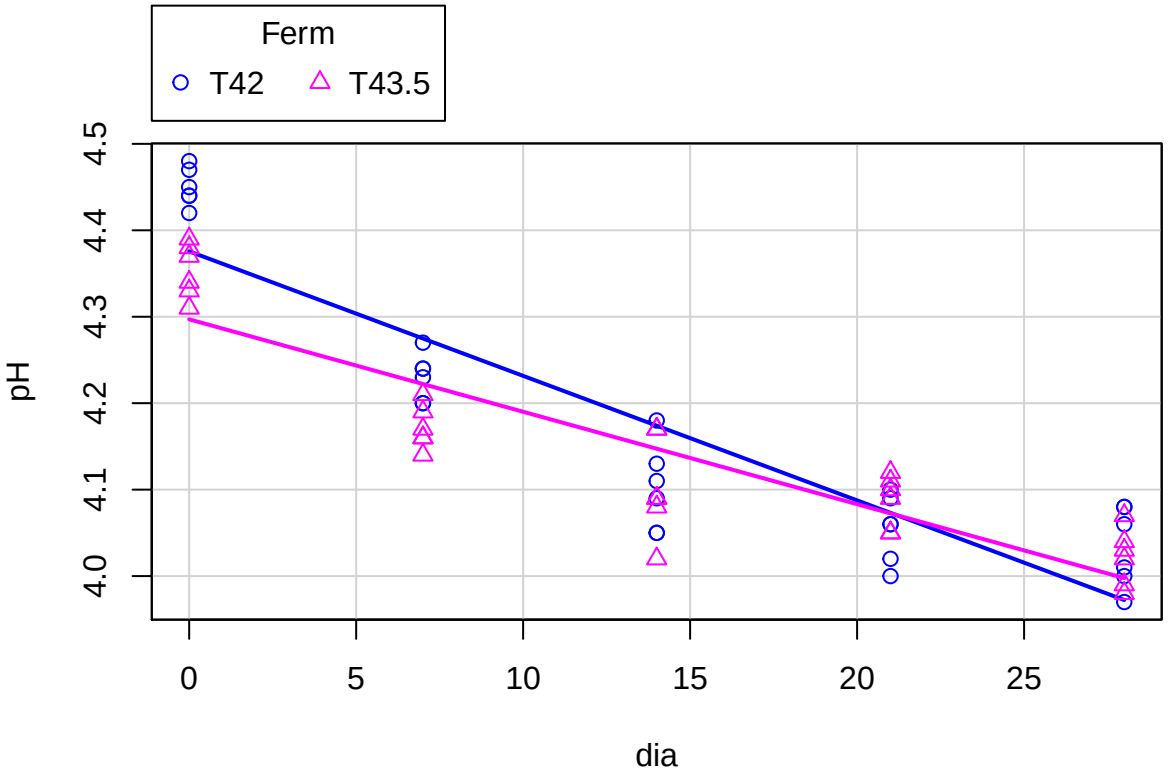
Estadistica descriptiva

```
head(dbi)
```

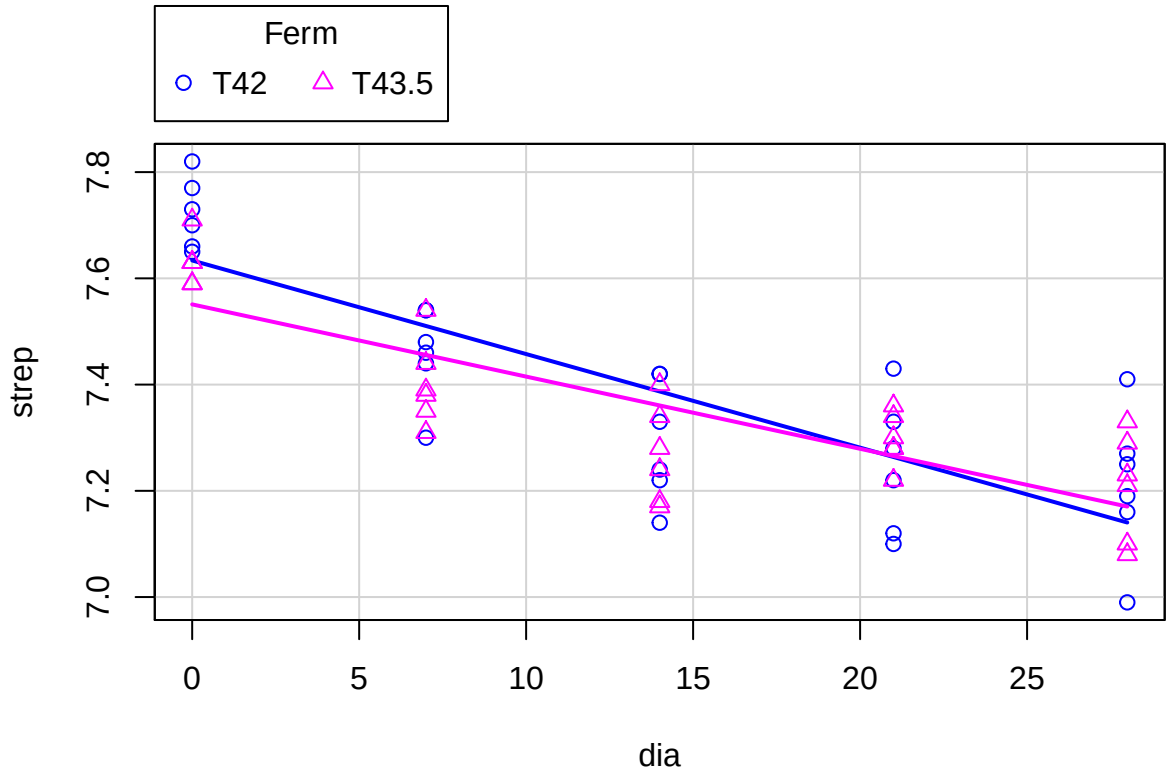
```
##   Ferm dia   pH strep lactob
## 1  T42  21 4.10  7.43   7.46
## 2  T42   0 4.44  7.65   7.75
## 3  T42  21 4.02  7.10   7.35
## 4  T42   7 4.24  7.54   7.62
## 5  T42   7 4.27  7.54   7.66
## 6  T42  28 4.01  7.25   7.41
```

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

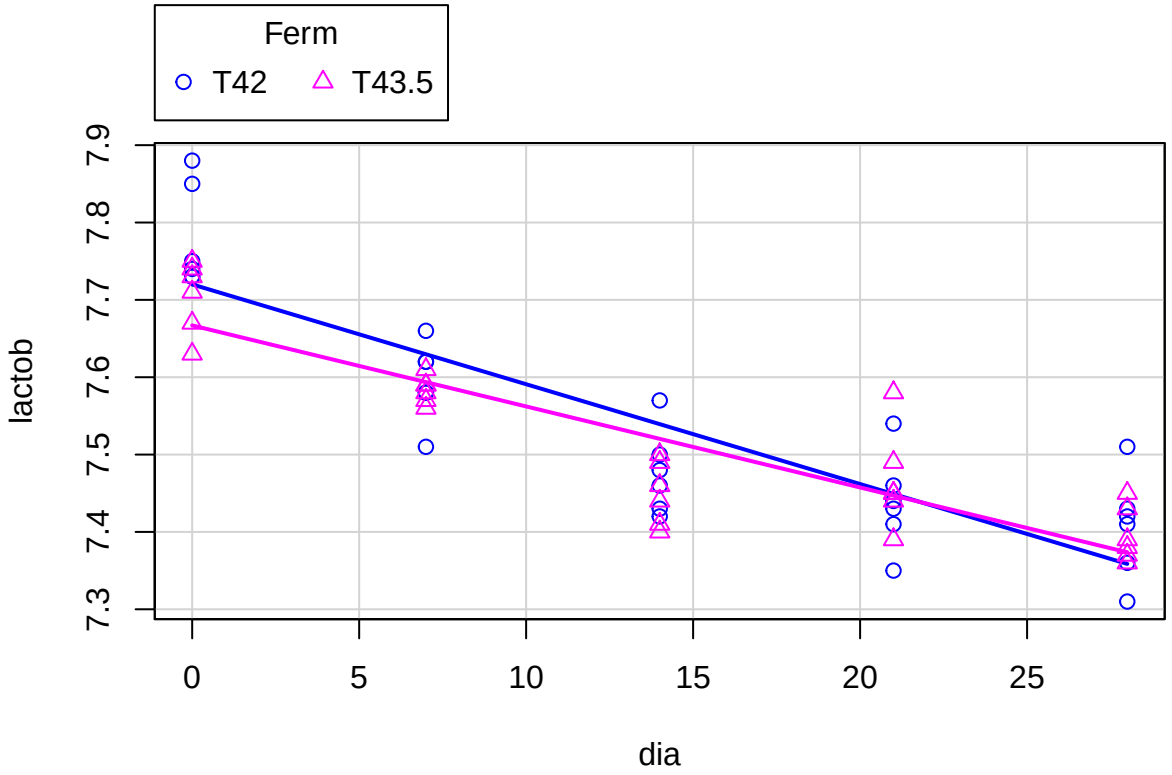
```
sp(pH ~ dia|Ferm, boxplot=F, smooth=F, data = dbi)
```



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




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



Per

que respecta al pH observem:

1) Els iogurts fermentats a T42 tenen els primers dies un pH superior al altres. Ara bé, a mida que passen els dies els valors de pH son pràcticament indistingibles entre les dues temperatures.
2) La manera en que el pH decreix a mida que passen els dies es mes dràstica pels iogurts fermentats a T42 que els fermentats a T43.5
3) Si considerem un *grup d'observacions* com les observacions que corresponen a un dia concret i a una temperatura de fermentació fixa, no es veuen moltes diferencies entre la variabilitat dels valors dels diferents grups d'observacions.

Pel que respecta a la variable Strep observem bàsicament el mateix que abans, però ara les diferencies entre les dues temperatures de Fermentació a l'inici no son tant acusades. Pels dies 21 i 28 i pel que respecte a la temperatura 42 de fermentació s'observen uns valors de pH que varien molt mes que per les altres temperatures i dies.

Pel que respecta a la variable Lactob observem: Hi ha dos iogurts fermentats a temperatura 42 que tenen una presencia de Lactobacilus molt mes superior que la resta de iogurts. Aquests dos continuen tenint valors mes alts que la resta a mida que passen els dies.

En general podem concloure que tant el pH com l'Strep com el Lactob van decreixent en el iogurt a mida que passen els dies des de la seva fermentació. Aquest decreixement sembla mes marcat pels iogurts fermentats a T42 que pels fermentats a T43.5. Podríem dir que quan han passat mes de 20 dies des de la fermentació, no sembla que hi hagi diferencies entre els dos grups de iogurts er a cap de les tres variables resposta.

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

```
setwd("~/Desktop/UNI/3rcurs/1rquatr/pie/entregable2/")
dd <- read.csv2("ah.csv")
```

Normal

Link canònic: Identitat Variace Function: 1

Quan especifiquem el model, hem d'indicar la distribució de les Y_i i la link function que fem servir. Si no indiquem res, s'utilitza la canonical link per defecte.

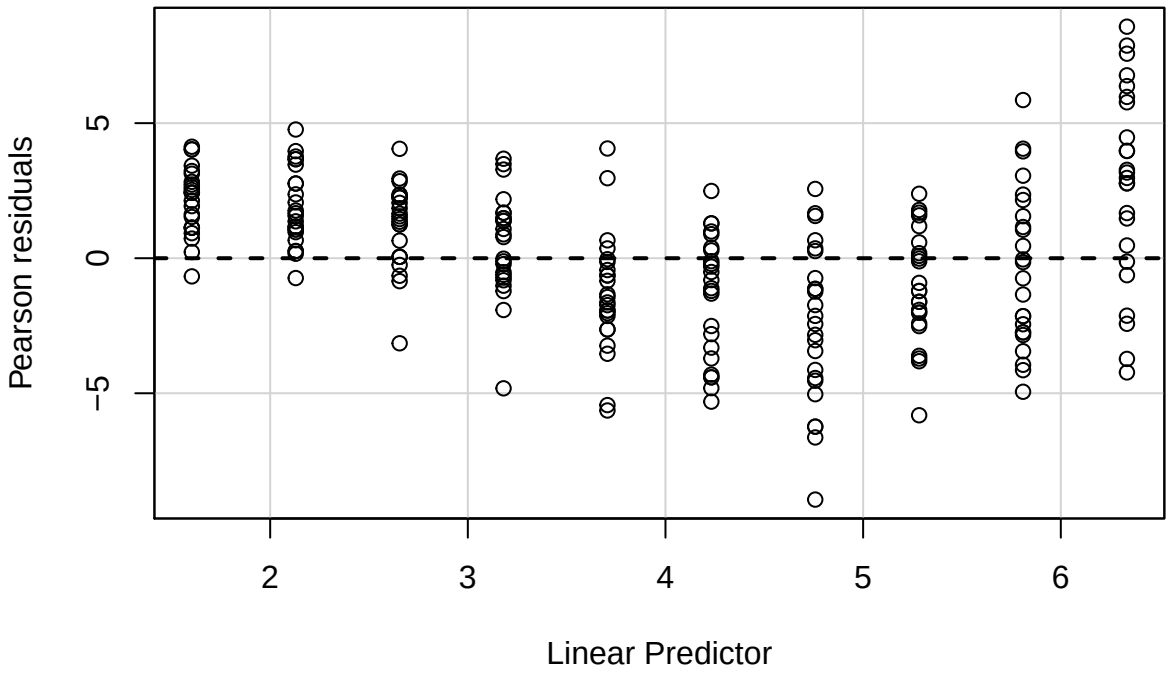
```
summary(gm1 <- glm(H~Days, family=gaussian(link="sqrt"), data=dd))
```

```
##
## Call:
## glm(formula = H ~ Days, family = gaussian(link = "sqrt"), data = dd)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.9356  -1.6618   0.3636   2.1348   8.5721
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.4535672  0.0660612    22.0   <2e-16 ***
## Days        0.0375469  0.0006658    56.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 8.144493)
##
##     Null deviance: 35624.7  on 239  degrees of freedom
## Residual deviance:  1938.4  on 238  degrees of freedom
## AIC: 1188.4
##
## Number of Fisher Scoring iterations: 6
```

Anàlisi dels residus

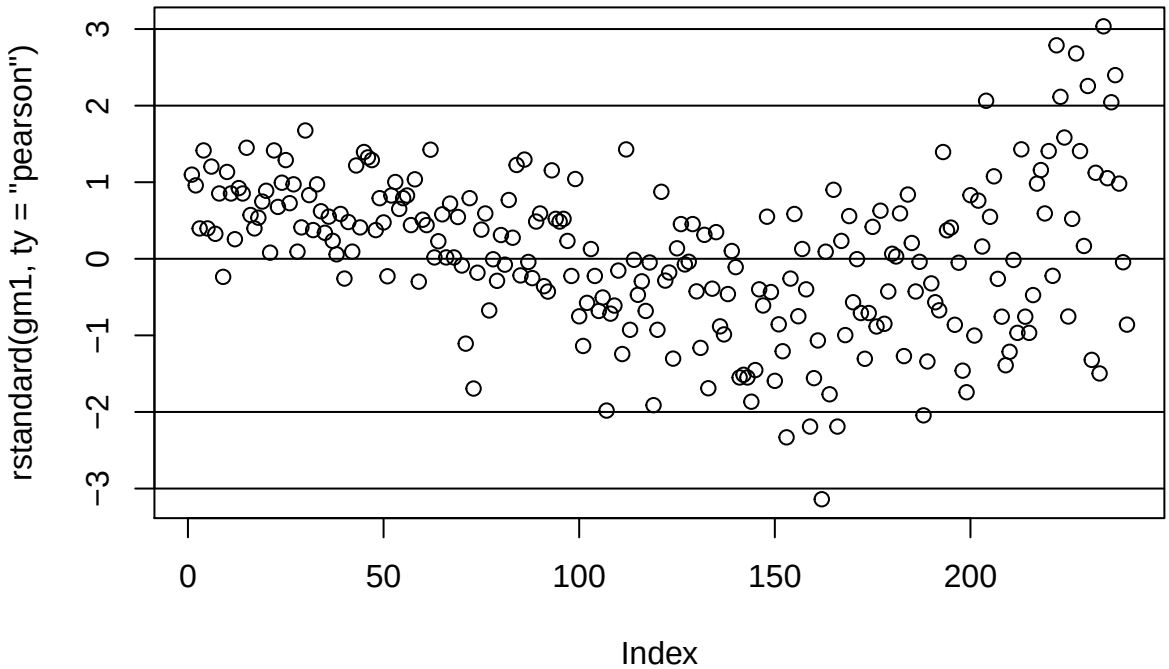
Pels residus de Pearson:

```
residualPlot(gm1, smooth = F)
```



Pels standarized Pearson Residuals:

```
plot(rstandard(gm1, ty="pearson"))
abline(h=c(-3,-2,0,2,3))
```



Per saber quins compleixen alguna condició en concret, podem utilitzar la següent comanda:

```
#which(abs(rstandard(m1)) > 2)
```

Sabem que l'estadístic de Pearson és $\sum_{i=1}^{n} r_i^2$, on els r_i són els residus de Pearson. Llavors, per calcular-lo:

```
X2 <- sum((rstandard(gm1, type = "pearson"))^2)
X2
```

```
## [1] 240.5083
```

```
phi = X2/gm1$df.residual
phi
```

```
## [1] 1.010539
```

Podem mirar si els Pearsons Stadistics tenen o no homocedasticitat amb el Levenne Test:

Intervals de confiança

```
#CLD(emmeans(mod2,~pH | Biomass, ty="response"))
```

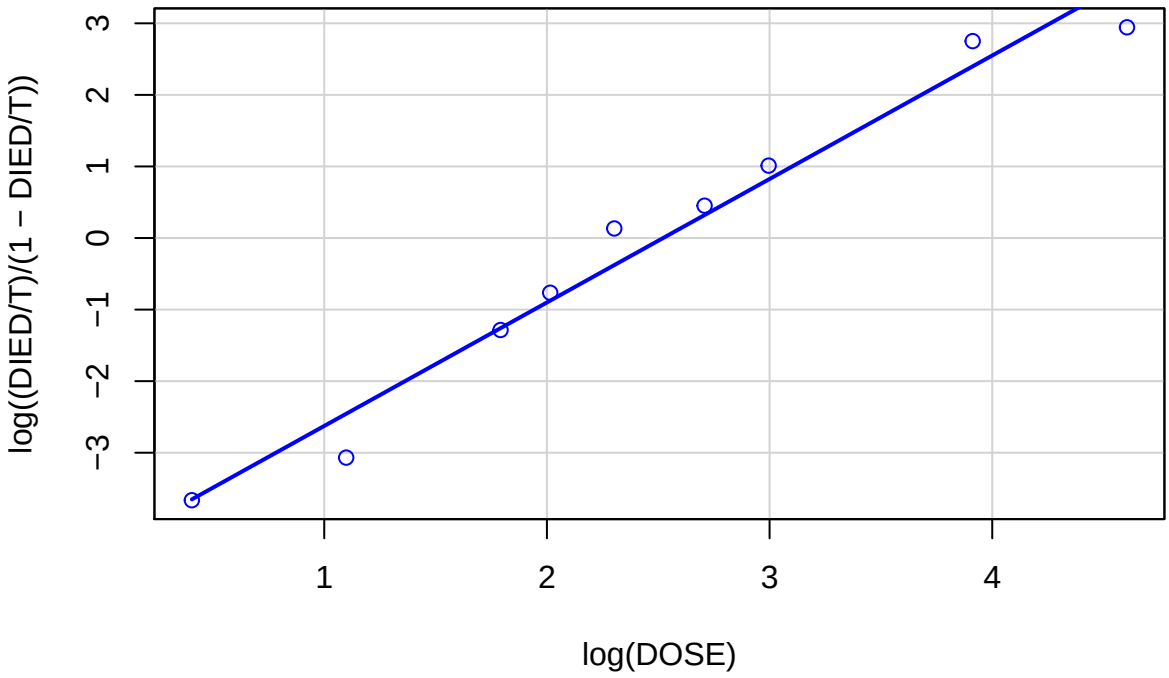
```
#CLD(emmeans(mod2,~pH | Biomass, at=list(Biomass=c(1,3,7)), ty="response"))
```

```
#pairs(emmeans(mod2,~pH | Biomass, at=list(Biomass=c(1,3,7))))
```

Binomial

Canonical link = logit Variance Function = $\mu(1-\mu)$

```
setwd("~/Desktop/UNI/3rcurs/1rquatr/pie/dades/")
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-DIED) ~ log(DOSE), family = binomial, data = insc)
summary(glm1)
```

```
##
## Call:
## glm(formula = cbind(DIED, T - DIED) ~ log(DOSE), family = binomial,
```

```
##     data = insc)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.27249  -0.27606  -0.07556   0.08958   1.49131
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -4.523      0.381  -11.87   <2e-16 ***
## log(DOSE)      1.855      0.158   11.74   <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: 307.2774  on 8  degrees of freedom
## Residual deviance:   5.5639  on 7  degrees of freedom
## AIC: 43.565
##
## Number of Fisher Scoring iterations: 4
```

Pearson residuals:

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

pres
```

```
##           2           3           4           5           6           7
##  0.16520744 -1.39030381 -0.30513837  0.10346074  1.64260589 -0.27026196
##           8           9          10
## -0.08835905  0.03475726 -1.63606997
```

```
X2
```

```
## [1] 7.520988
```

```
phi
```

```
## [1] 1.074427
```

Over / underdisperssion

Tant en la binomial com en la Poisson, aproximant el $\hat{\phi}$ per $X^2/(n-p)$, si és major que 1, tenim overdisperssion, si es menor que 1 tenim underdisperssion.

Poisson

Canonical link = log Variance function = μ

Predicted values

```
#customDays <- data.frame(Days=c(0,105,150))
#predC <- predict(mC, customDays, ty="response") # mu
#standevC <- sqrt(predC * mC$deviance/mC$df.residual) # sd
```

L'standard deviation es multiplica per la variance function de la distribució que estem agafant. En aquest cas, donat que la variance function és $/mu$, es multiplica pels valors predits

Gamma

Canonical link= $1/\mu$ Variance function = μ^2

Comparació entre Models

En primer lloc busquem la logLike més gran.

```
#logLik(mod)
```

En segon lloc, AIC més petit.

```
#AIC(mod)
```

Finalment podem comparar models niuats amb la deviancia.

Altres

A ~ B + C No hi ha interacció A ~ B * C Sí hi ha interacció

```
#Model quasi-likelihood

#summary(mC <- glm(H ~ Days, family = quasi(link=log, var="mu"), data = dd) ))

#Passar una variable contínua a factor

#db$FDays <- as.factor(db$Days)
#modB2 <- glm(H~Days+FDays, family = Gamma(link="log"), db)
```