

# Modelització de dades longitudinals amb efectes aleatoris de intercepció/pendent i presència de dades perdudes en el context dels estudis d'estabilitat – ANNEX 3

**Estudiant: Francesc Bernad Martin**

M0.178 TFM-Estadística i Bioinformàtica 2

Màster universitari en Bioinformàtica i bioestadística UOC-UB

Àrea: Bioestadística / Bioinformàtica

**Nom Consultor/a: Nuria Perez Alvarez**

**Nom Professor/a responsable de l'assignatura: Carles Ventura Royo**

Barcelona, 04 de Juny de 2019



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Exportació del codi de programació R  
Markdown corresponent a resolució del  
problema pràctic dins el context de l'aplicació  
dels models mixtos analitzat

```

---
title: "TFM_ANNEX-3_CAS_PRÀCTIC_LAKE"
author: "Francesc Bernad Martin"
date: "May 31, 2019"
output:
  html_document:
    df_print: paged
    toc: yes
    toc_depth: 4
  pdf_document:
    toc: yes
    toc_depth: 4
  word_document:
    toc: yes
    toc_depth: '4'
header-includes:
- \usepackage{float}
- \usepackage{booktabs}
- \usepackage{makecell}
- \usepackage{colortbl}
- \usepackage{xcolor}
---
<style>
body {
text-align: justify}
</style>
````{r, include=FALSE}
options(tinytex.verbose = TRUE)
options(kableExtra.latex.load_packages = FALSE)
````

````{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE, cache = T, warnings = F)
#Es posa EL CACHE EN TRUE PQ VAGI MÉS RÀPID A GENERAR INFORMES
````

````{r libraries, results='hide', message=FALSE, warning=FALSE}
usePackage <- function(p) {
  if (!is.element(p, installed.packages()[,1]))
    install.packages(p, dep = TRUE)
  require(p, character.only = TRUE)
}
usePackage("nlme")
usePackage("nlmeU")
usePackage("lme4")
usePackage("ggplot2")
usePackage("gridExtra")
usePackage("ggExtra")
usePackage("kableExtra")
usePackage("RLRsim")
usePackage("mixlm")
usePackage("magrittr")
usePackage("evaluate")
usePackage("haven")
usePackage("summarytools")
usePackage("psych")
usePackage("stringr")
usePackage("randomcolor")
usePackage("MASS")
usePackage("stats")
usePackage("lmtest")

```

```

usePackage("faraway")
usePackage("piecewiseSEM")

ctrl <- lmeControl(opt='optim') # Mesura de control per l'aplicació
dels paquets lme per evitar errors per incorrecte optimització
d'iteracions
` ``

```

Petita funció que ha servir d'utilitat ja creada en els anteriors codis per reorganitzar objectes del tipus llista:

```

` ``{r FunCLists}
#Es genera una funció amb la simple instrucció de quan es facin
l·listats dintre l·listats converteixi tot en una llista normal sense
subnivells per facilitar els resums
ConvList <- function(List){
  ListC <- list()
  for(l1 in 1:length(List)) {
    if(class(List[[l1]])=="list"){
      for (l2 in 1:length(List[[l1]])) {
        ListC[[length(ListC)+1]] <- List[[l1]][[l2]]
        names(ListC)[length(ListC)] <- names(List[[l1]])[l2]
      }
    } else {
      ListC[[length(ListC)+1]] <- List[[l1]]
      names(ListC)[length(ListC)] <- names(List)[l1]
    }
  }
}

return(ListC)
} ``
` ``

```

# ANÀLISI DESCRIPTIU DATASET

## Importació Dataset

S'importa el \*dataset\* a analitzar i es visualitza una part de la taula:

```

` ``{r Import DS}
workingDir <-getwd()
setwd(workingDir)
LAKE0 <- read_sav(paste(workingDir,"/BdD Clinical
Trial/LAKE/lake.sav",sep=""))
head(LAKE0[,1:10])
` ``

```

## Estructura y selecció

Es mostra el tipus de variables que formen el model:

```

` ``{r Estr DS}
dim(LAKE0)
table(sapply(LAKE0,class))

which(sapply(LAKE0,class)=="character")

which(sapply(LAKE0,class)=="Date")
` ``

```

La quantitat d'individus inicial és gran i suficient per analitzar de manera correcta semblaria (``r nrow(LAKE0)`` individus). HI ha també una quantitat enorme de variables (``r ncol(LAKE0)``). És d'interès pel model que es vol testar modelitzar amb poques variables d'inici ja que l'objectiu no és l'anàlisi multivariable en sí.

Respecte a la identificació del tipus de variables hi ha:

```
+ `r length(which(sapply(LAKE0,class)=="character"))` variables del
tipus caràcter que en principi serien identificadors dels individus i
que corresponen a *`r
names(which(sapply(LAKE0,class)=="character"))`. De moment no
interessa eliminar-les (almenys les dos primeres).
+ `r length(which(sapply(LAKE0,class)=="Date"))` variables del tipus
data. Serien útils en el cas que es volgués estudiar algun tipus
d'autocorrelació com per exemple l'estacionalitat, però d'inici es
transformaran per poder tenir-les en format numèric de temps i així
assimilar el tipus d'estudi treballat anteriorment.
+ `r length(which(sapply(LAKE0,class)=="haven_labelled"))` variables
del tipus *haven_labelled* per importar dades d'altres softwares
estadístics. De moment no es tindran en compte excepte la variable
`Grupo` que marca els dos tractaments de l'estudi i `sexo` que podria
aportar informació interessant al tipus de model a aplicar.
+ `r length(which(sapply(LAKE0,class)=="numeric"))` variables del
tipus numèric. Algunes seran també identificadores dels individus,
però moltes d'elles poden ser predictors o observacions objectiu a
analitzar. En el present estudi s'està assimilant els models que s'han
simulat en el context dels estudis d'estabilitat pel que en la
selecció de variables d'aquest estudi no es tindrà en compte el
context de l'objectiu de l'estudi i simplement s'analitzaran algunes
de les variables a l'atzar per veure com reaccionen al model i
d'aquesta manera també s'impedeix el biaix de seguir l'estudi original
que ja té les seves pròpies conclusions amb l'anàlisi multivariant. A
excepció de la variable de càrrega viral que semblaria evidentment la
variable a observar més important, la resta es prenen de manera
semialeatòria.
```

Si s'observa quines són les variables numèriques:

```
```{r LAKE0 t}
which(sapply(LAKE0,class)=="numeric")
```
```

Es mostra que l'estructura de la matriu és amb els paràmetres col·locats per cada individu. S'ha de modificar aquest format perquè encaixi amb el format necessari per aplicar amb el codi R on s'afegeix la variable temps i cada variable individual com a columna.

L'objectiu d'aquesta pràctica no és l'anàlisi multivariant i per tant interessa una matriu amb poques variables d'inici, amb variables identificadores i la variable temps. Per fer això es genera un nou \*dataset\* a partir de l'original fent les eliminacions comentades, càlculs de temps i en el cas de les variables numèriques s'extreuen semialeatòriament 10 variables:

```
```{r Subset LAKE1}
LAKE0a<- LAKE0[,which(sapply(LAKE0,class)!="haven_labelled" &
sapply(LAKE0,class)!="Date")]
LAKE0a$Grupo <- LAKE0$Grupo
LAKE0a$sexo <- LAKE0$sexo
```

```

#De les variables numèriques es deixa només amb edad i grup com a
idenfiticadors i tractament, i amb les que depenen del temps

#Var no depenents del temps
LAKE1a0
LAKE0a[,c(which(sapply(LAKE0a,class)==="character"),which(colnames(LAKE
0a) == c("edad","Grupo","sexo")))]
LAKE1a0 <- LAKE1a0[,-3] #S'elimina la var classificació que sembla no
tenir sentit aquí

#Es converteix a factors els dos termes identificatius per facilitar
la interpretació de les dades i es genera una nova columna
identificadora que combina el nusuario i ncap
LAKE1a0[,1] <- as.factor(LAKE1a0[,1][[1]])
LAKE1a0[,2] <- as.factor(LAKE1a0[,2][[1]])
LAKE1a0$Ident <- as.factor(paste(LAKE1a0$nusuario,LAKE1a0$npac))

LAKE1a0$Grupo <- as.factor(LAKE1a0$Grupo)# Es converteix el tractament
en una variable del tipus factor
levels(LAKE1a0$Grupo) <- list("Tratm.1"=-1, "Tratm.2"=0)
LAKE1a0$sexo <- as.factor(LAKE1a0$sexo) # Es converteix la variable
sexo en una variable del tipus factor
levels(LAKE1a0$sexo) <- list("Hombre"=1, "Mujer"=2)

#Var depenents del temps
LAKE1b0 <- LAKE0a[,which(sapply(LAKE0a,class)==="numeric")]
LAKE1b0 <- LAKE1b0[,-grep("diff_",colnames(LAKE1b0))]#S'elimina les
variables diff_ que poden fer confondre en aquest punt de
reorganització de dades

nvar_t_string <- length(vart_string <- c("_0","_12","_24","_36","_48"))

vart_pos <- as.vector(sapply(vart_string,function(s){
  grep(paste(s),colnames(LAKE1b0))
}))
LAKE1b0 <- LAKE1b0[,vart_pos]#Es conserva només les variables que
acaben en _0, _12, _24, _36 i _48 corresponents a les depenents del
temps
nvar <- ncol(LAKE1b0)/length(vart_string)
var_names <- str_remove(colnames(LAKE1b0)[1:nvar],"_0")
var_names <- var_names[-
c(which(var_names=="week"),which(var_names=="CargaViral"))]#Elimine la
variable week i la CargaViral (aquesta última es recupera a
posteriori)
#Es tria de manera semialeatoria algunes variables de les ja
seleccionades
set.seed(7982)
nset <- 3
(var_names_sel
c("CargaViral",var_names[sample(1:length(var_names),nset,replace=F)]))
nvar_sel <- length(var_names_sel)

LAKE1b0_sel
LAKE1b0[,c(paste(var_names_sel,rep(vart_string,each=length(var_names_s
el)),sep=""))]
LAKE1ab0_prov <- cbind(LAKE1a0,LAKE1b0_sel)
LAKE1ab0_prov_noNA <- na.omit(LAKE1ab0_prov,
cols=((ncol(LAKE1a0)+1):ncol(LAKE1ab0_prov_noNA))) #només a les
variables numèriques es busca els NA

LAKE1a0_noNA <- LAKE1ab0_prov_noNA[,1:ncol(LAKE1a0)]

```

```

LAKE1b0_noNA <- LAKE1ab0_prov_noNA[,
(ncol(LAKE1a0)+1):ncol(LAKE1ab0_prov_noNA)]

#Es reorganitzen les variables (es crea una funció per poder
utilitzar en altres casos)
reorg <- function(df,nvar,nTA){
  reorg1 <- lapply(1:nvar,function(v){
    var_v <- df[,v]
    for(r in 1:(nTA-1)){
      var_v <- rbind(var_v,df[, (v+(nvar*r))])
    }
    return(var_v)
  })
  reorg2 <- sapply(1:nvar,function(v){
    as.vector(reorg1[[v]])
  })
  return(reorg2)
}

reorgLAKE1b0_noNA <- reorg(LAKE1b0_noNA,nvar_sel,nvart_string)

LAKE1b_noNA <- reorgLAKE1b0_noNA
colnames(LAKE1b_noNA) <- var_names_sel

LAKE1a_noNA <-
LAKE1a0_noNA[rep(1:nrow(LAKE1a0_noNA),each=nvart_string),]
LAKE1a_noNA <- droplevels(LAKE1a_noNA) #Es netegen les variables
factor de nivells no utilitzats
LAKE1a_noNA$Tiempo <-
rep(seq(from=0,to=48,by=12),nlevels(LAKE1a_noNA$Ident))#Es genera la
variable temps corresponent a les setmanes de la mesura

head(LAKE1_noNA <- cbind(LAKE1a_noNA, LAKE1b_noNA))

str(LAKE1_noNA)

LAKE1 <- LAKE1_noNA #Es deixa un nom més fàcil d'utilitzar
#S'elimina de moment npac i nusuari ja que ja s'ha creat la Ident
LAKE1 <- LAKE1[,3:ncol(LAKE1)]
```



Al depurar el dataset per acomodar-lo als propòsits d'aquesta part pràctica s'ha quedat en un conjunt de dades amb nlevels(LAKE1$Ident) subjectes i ncol(LAKE1) variables.



## Anàlisi descriptiu Dataset



Es realitzen els anàlisis descriptius bàsics al dataset resultant per donar una visió més clara de les variables que intervenen:



```

```{r Descr DS LAKE1}
summary(LAKE1)
by(LAKE1, LAKE1$Grupo, summary)
by(LAKE1, LAKE1$sexo, summary)

table(subset(LAKE1, LAKE1$Grupo=="Treatment.1")$sexo)
table(subset(LAKE1, LAKE1$Grupo=="Treatment.2")$sexo)
```

```


```



Excepte la variable de càrrega viral que sembla moure's d'una manera diferent per cada individu, la resta de variables numèriques no mostra en aquest punt una diferència que es percebi amb claredat en el resum de dades. Sí que es veu en aquest cas que la proporció de sexes és `r paste(round(prop.table(table(subset(LAKE1, LAKE1\$Grupo=="Treatm.1")\$sexo)) \* 100, digits = 1), "%", sep=" ")` (Homes/Dones) pel tractament 1 i `r paste(round(prop.table(table(subset(LAKE1, LAKE1\$Grupo=="Treatm.2")\$sexo)) \* 100, digits = 1), "%", sep=" ")` (Homes/Dones) pel cas del tractament 2 (en cas que no hi hagi repartiment equitatiu pot ser un problema per analitzar el significat real del factor sexe).

Per analitzar possibles relacions entre les diferents variables es pot fer una primera aproximació de càlcul de correlacions i la visualització de l'anomenat *\*scatterplot matrix\**:

```
```{r Scatterplot Matrix LAKE 1}
pairs.panels(LAKE1[,sapply(LAKE1,class)==="numeric"],
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE, # show density plots
             ellipses = TRUE # show correlation ellipses
             )

LAKE1cor <- as.data.frame(cor(LAKE1[,sapply(LAKE1,class)==="numeric"]))
LAKE1cor2 <- LAKE1cor[which(LAKE1cor$CargaViral!=1),]
LAKE1maxpred <- rownames(LAKE1cor2[which.max(abs(LAKE1cor2$CargaViral)),])
```
```

D'aquest gràfic i càlculs interessa saber quines són les variables que tenen una correlació més alta amb la resposta de la càrrega viral que en aquest cas en resulta una correlació amb les variables *\*`r rownames(LAKE1cor2)`\** de *`r round(LAKE1cor2\$CargaViral,2)`* respectivament, prenent com a predictor principal la variable de correlació més alta en valor absolut: *\*`r LAKE1maxpred`\**.

Com a prova addicional i veient l'histograma de dades de la variable càrrega viral, es repeteix la matriu de correlació eliminant el temps 0 de tots els subjectes per veure com es modifiquen els resultats:

```
```{r Scatterplot Matrix LAKE 1 no T0}
LAKE1r <- subset(LAKE1, LAKE1$Tiempo!=0)
pairs.panels(LAKE1r[,sapply(LAKE1r,class)==="numeric"],
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE, # show density plots
             ellipses = TRUE # show correlation ellipses
             )
```
```

La proporció és similar pel que de moment no es modificarà res. Tot i així és possible que s'hagi d'aplicar alguna transformació per poder explicar la variabilitat de la càrrega viral amb models més simples.

```
```{r LAKE1 Plots0}
p01<-ggplot(LAKE1, aes(x=CargaViral)) +
  geom_vline(aes(xintercept=mean(LAKE1$CargaViral)),
            color=randomColor(), linetype="dashed", size=1) +
  geom_histogram()
```
```

```

p02<-ggplot(LAKE1, aes(x="", y=CargaViral, fill=randomColor())) +
  geom_boxplot() + geom_dotplot(binaxis='y', stackdir='center',
dotsize=0.5) +
  theme(legend.position="none")

p03<-ggplot(LAKE1, aes(sample = CargaViral)) + stat_qq() +
stat_qq_line()

grid.arrange(p01,p02,p03)
``,``

```

La distribució d'aquests valor s'allunya clarament de la normalitat. Es mostra si amb la transformació logarítmica canvia una mica aquests primers anàlisis descriptius:

```

```{r LAKE 1 logViral}
LAKE1$LogCargaViral <- log(LAKE1$CargaViral)
pairs.panels(LAKE1[,sapply(LAKE1,class)==="numeric"],
  method = "pearson", # correlation method
  hist.col = "#00AFBB",
  density = TRUE, # show density plots
  ellipses = TRUE # show correlation ellipses
)
p11<-ggplot(LAKE1, aes(x=LogCargaViral)) +
  geom_vline(aes(xintercept=mean(LAKE1$LogCargaViral)),
  color=randomColor(), linetype="dashed", size=1) +
  geom_histogram()

p12<-ggplot(LAKE1, aes(x="", y=LogCargaViral, fill=randomColor())) +
  geom_boxplot() + geom_dotplot(binaxis='y', stackdir='center',
dotsize=0.5) +
  theme(legend.position="none")

p13<-ggplot(LAKE1, aes(sample = LogCargaViral)) + stat_qq() +
stat_qq_line()

grid.arrange(p11,p12,p13)
``,``

```

Sembla millorar lleugerament. Es prova addicionalment també la transformació de Box-Cox que utilitza 2 funcions una logarítmica i una potencial per determinar la millor transformació en un hipotètic model simple depenent del temps:

```

```{r LAKE 1 BoxCox}
bct<-boxcox(CargaViral~Tiempo,lambda = seq(-2, 2, length =
10),data=LAKE1,plotit=T)
lambda<-bct$x[which.max(bct$y)]
lambda
y <- LAKE1$CargaViral
gm<-geometric.mean(y)

LAKE1$CargaViral_t<-(y^lambda-1)/(lambda*gm^(lambda-1))
LAKE1$LogCargaViral <- log(LAKE1$CargaViral)

pairs.panels(LAKE1[,sapply(LAKE1,class)==="numeric"],
  method = "pearson", # correlation method
  hist.col = "#00AFBB",
  density = TRUE, # show density plots
  ellipses = TRUE # show correlation ellipses
)

```

```

p21<-ggplot(LAKE1, aes(x=CargaViral_t)) +
  geom_vline(aes(xintercept=mean(LAKE1$CargaViral_t)),
            color=randomColor(), linetype="dashed", size=1) +
  geom_histogram()

p22<-ggplot(LAKE1, aes(x="", y=CargaViral_t, fill=randomColor())) +
  geom_boxplot() + geom_dotplot(binaxis='y', stackdir='center',
dotsize=0.5) +
  theme(legend.position="none")

p23<-ggplot(LAKE1, aes(sample = CargaViral_t)) + stat_qq() +
stat_qq_line()

grid.arrange(p21,p22,p23)
` ``

```

Millora lleugerament respecte a la transformació logarítmica, tot i que els inconvenients potser no justifiquen mantenir aquesta transformació. En principi es mantindrà únicament la logarítmica i en algun cas es pot fer alguna prova per si aquesta transformació aporta una millora significativa.

```

````{r LAKE 1 Plots0}
Lake1SCPlot.a <- ggplot(LAKE1, aes(x=Tiempo, y=CargaViral,
color=Grupo, shape=Grupo)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title="Càrrega viral s/ tract.",
        x="Tiempo", y = "CargaViral")+
  scale_color_brewer(palette="Accent")

Lake1SCPlot.b <- ggplot(LAKE1, aes(x=Tiempo, y=LogCargaViral,
color=Grupo, shape=Grupo)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title="Càrrega viral s/ tract.",
        x="Tiempo", y = "Log CargaViral")+
  scale_color_brewer(palette="Accent")

Lake1SCPlot.c <- ggplot(LAKE1, aes(x=Tiempo, y=CargaViral_t,
color=Grupo, shape=Grupo)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title="Càrrega viral s/ tract.",
        x="Tiempo", y = "CargaViral_t")+
  scale_color_brewer(palette="Accent")

grid.arrange(Lake1SCPlot.a,Lake1SCPlot.b,Lake1SCPlot.c)

Lake1BPlot.a <- ggplot(LAKE1, aes(x=Grupo, y=CargaViral, fill=Grupo))
+
  geom_boxplot()+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1)+
  labs(title="Càrrega viral s/ tract.",
        x="Tractament", y = "CargaViral")+
  scale_fill_brewer(palette="Accent")
Lake1BPlot.b <- ggplot(LAKE1, aes(x=Grupo, y=LogCargaViral,
fill=Grupo)) +
  geom_boxplot()+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1)+
  labs(title="Càrrega viral s/ tract.",

```

```

        x="Tractament", y = "LogCargaViral")+
        scale_fill_brewer(palette="Accent")
Lake1BPlot.c <- ggplot(LAKE1, aes(x=Grupo, y=CargaViral_t,
fill=Grupo)) +
  geom_boxplot()+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1)+
  labs(title="Càrrega viral s/ tract.",
        x="Tractament", y = "CargaViral_t")+
  scale_fill_brewer(palette="Accent")

grid.arrange(Lake1BPlot.a,Lake1BPlot.b,Lake1BPlot.c)

```

```
# AJUST MODEL SIMPLE
```

```
## Model mínims quadrats *Stepwise*
```

S'ha deixat un model amb pocs individus i poques variables per poder il·lustrar el context habitual dels estudis d'estabilitat on es solen tenir pocs lots de referència pels primers models de predicció i poques variables predictores a part del temps.

Abans de començar a entrar amb models més complexos es comença ajustant el model més simple per mínims quadrats, però en aquest cas s'aplica per ser més pràctics una aproximació mitjançant algoritmes que segueixen el model *\*stepwise\** el qual és una combinació de les comparacions ANOVA entre afegir predictors al model base o eliminar-ne al model complet per arribar al model més òptim (des del punt de vista d'aquest sistema). Habitualment no s'arriba al model òptim, però servirà com a punt de partida per elaborar els models més complexos:

```
```{r Lake1 Model Simple Stepwise}
```

```

# Fit the full model
Lake1.fm0 <- lm(CargaViral ~ . - LogCargaViral - CargaViral_t - Ident,
data = LAKE1)
# Stepwise regression model
Lake1.step.model0 <- stepAIC(Lake1.fm0, direction = "both",
                             trace = F)
(Lake1.step.model0s <- summary(Lake1.step.model0))

```

```
anova(Lake1.step.model0)
```

```

# Fit the full model
Lake1.fm1 <- lm(LogCargaViral ~ . - CargaViral - CargaViral_t - Ident,
data = LAKE1)
# Stepwise regression model
Lake1.step.model1 <- stepAIC(Lake1.fm1, direction = "both",
                             trace = F)
(Lake1.step.model1s <- summary(Lake1.step.model1))

```

```

# Fit the full model
Lake1.fm2 <- lm(CargaViral_t ~ . - CargaViral - LogCargaViral - Ident,
data = LAKE1)
# Stepwise regression model
Lake1.step.model2 <- stepAIC(Lake1.fm2, direction = "both",
                             trace = F)
(Lake1.step.model2s <- summary(Lake1.step.model2))

```

```

```{r Taules1}
Lake1.st1 <- (cbind(round(Lake1.step.model0s$coefficients[-1, -3], 4),
                    rbind(round(Lake1.step.model1s$coefficients[-1, -3], 4), rep("--", 3), rep("--", 3))),
              rbind(round(Lake1.step.model2s$coefficients[-1, -3], 4), rep("--", 3), rep("--", 3))))
kable(Lake1.st1) %>%
  kable_styling("striped") %>%
  add_header_above(c(" " = 1, "M.Simple sense transf. 1" = 3,
                    "M.Simple log transf" = 3, "M.Simple Box-Cox t" = 3))

Lake1.st2 <- data.frame(Model=c("M.Simple sense transf", "M.Simple log transf", "M.Simple Box-Cox t"))
Lake1.st2$"Mediana"      residual"      <-
c(median(Lake1.step.model0s$residuals),
  median(Lake1.step.model1s$residuals),
  median(Lake1.step.model2s$residuals))

kable(Lake1.st2) %>%
  kable_styling("striped")
```

```

Com es pot observar en els 3 modelatges de model simple:

```

+ El model amb variables originals s'ha considerat òptima la utilització de 3 predictors (*`r attr(Lake1.step.model0$terms,"term.labels")`*) per explicar la resposta, els 3 amb una prova de significació no significativa (p-valors de `r round(Lake1.step.model0s$coefficients[2:nrow(Lake1.step.model0s$coefficients),4],4)` respectivament). En el cas de l'ANOVA es calcula la significació si es treuen del model 1 a 1 veient l'ordre de importància dels termes *`r rownames(anova(Lake1.step.model0))[1:length(attr(Lake1.step.model0$terms,"term.labels"))]*` de més a menys significatiu.
+ En els 2 models transformats, el sistema decideix pels 2 un model més simple que depèn només del temps essent aquest molt significatiu per explicar la resposta (p-valors de `r round(Lake1.step.model1s$coefficients[2:nrow(Lake1.step.model1s$coefficients), 4], 4)` i `r round(Lake1.step.model2s$coefficients[2:nrow(Lake1.step.model2s$coefficients), 4], 4)` respectivament per transformació Log i *Box-Cox*).

```

El que també s'observa és que de moment no hi ha cap prova que diferenciï el factor de tractament amb els paràmetres que s'estan utilitzant pel model, pel que no es pren de moment la consideració que hi hagi diferències entre tractaments (en l'estudi original hauria estat l'objectiu, però en aquest es pren l'objectiu d'aconseguir un model de predicció el màxim de exacte i precís per poder explicar correctament i predir les respostes futures en base al conjunt de dades seleccionat).

```
## Diagnòstic model simple
```

```
### Normalitat / Gràfics Residuals
```

```

```{r Diag N Model simple Lake1}
Lake1.SM0.sdres <- rstandard(Lake1.step.model0)
Lake1.SM0.studres <- studres(Lake1.step.model0)
par(mfrow=c(1,1))
```

```

```

qqnorm(Lake1.step.model0$residuals,
       ylab="Raw Residuals",
       xlab="Normal Scores",
       main="Model Original - NPP of Raw Residuals")
qqline(Lake1.step.model0$residuals)

qqnorm(Lake1.SM0.sdres,
       ylab="Standardized Residuals",
       xlab="Normal Scores",
       main="Model Original - NPP of Standardized Residuals")
qqline(Lake1.SM0.sdres)

qq01<-qqnorm(Lake1.SM0.studres,
             ylab="Studentized Residuals",
             xlab="Normal Scores",
             main="Model Original - NPP of Studentized Residuals")
qqline(Lake1.SM0.studres)

S0 <- shapiro.test(Lake1.step.model0$residuals)
S0

Lake1.SM1.sdres <- rstandard(Lake1.step.model1)
Lake1.SM1.studres <- studres(Lake1.step.model1)
par(mfrow=c(1,1))

qqnorm(Lake1.step.model1$residuals,
       ylab="Raw Residuals",
       xlab="Normal Scores",
       main="Model Log - NPP of Raw Residuals")
qqline(Lake1.step.model1$residuals)

qqnorm(Lake1.SM1.sdres,
       ylab="Standardized Residuals",
       xlab="Normal Scores",
       main="Model Log - NPP of Standardized Residuals")
qqline(Lake1.SM1.sdres)

qqnorm(Lake1.SM1.studres,
       ylab="Studentized Residuals",
       xlab="Normal Scores",
       main="Model Log - NPP of Studentized Residuals")
qqline(Lake1.SM1.studres)

S1 <- shapiro.test(Lake1.step.model1$residuals)
S1

Lake1.SM2.sdres <- rstandard(Lake1.step.model2)
Lake1.SM2.studres <- studres(Lake1.step.model2)
par(mfrow=c(1,1))

qqnorm(Lake1.step.model2$residuals,
       ylab="Raw Residuals",
       xlab="Normal Scores",
       main="Model Box-Cox - NPP of Raw Residuals")
qqline(Lake1.step.model2$residuals)

qqnorm(Lake1.SM2.sdres,
       ylab="Standardized Residuals",
       xlab="Normal Scores",
       main="Model Box-Cox - NPP of Standardized Residuals")
qqline(Lake1.SM2.sdres)

```

```

qqnorm(Lake1.SM2.studres,
       ylab="Studentized Residuals",
       xlab="Normal Scores",
       main="Model Box-Cox - NPP of Studentized Residuals")
qqline(Lake1.SM2.studres)

S2 <- shapiro.test(Lake1.step.model2$residuals)
S2

par(mfrow=c(1,3))

qqnorm(Lake1.SM0.sdres,
       ylab="Standardized Residuals",
       xlab="Normal Scores",
       main="Model Original - Standardized Res.",cex.main=1)
qqline(Lake1.SM0.sdres)

qqnorm(Lake1.SM1.sdres,
       ylab="Standardized Residuals",
       xlab="Normal Scores",
       main="Model Log - Standardized Res.",cex.main=1)
qqline(Lake1.SM1.sdres)

qqnorm(Lake1.SM2.sdres,
       ylab="Standardized Residuals",
       xlab="Normal Scores",
       main="Model Box-Cox - Standardized Res.",cex.main=1)
qqline(Lake1.SM2.sdres)

```

Sorprenentment els 3 models entren en el que es podria classificar com una distribuci3 normal dels residus del model tot i que no queda suportat per la prova de \*shapiro\* amb p-valors del model original, logtransformat i \*box-cox\* de `r round(S0\$p.value,4)`, `r round(S1\$p.value,4)` i `r round(S2\$p.value,4)` respectivament.

### Homoscedasticitat

```

```{r Diag H Model simple Lake1}
pH01 <- ggplot(LAKE1, aes(x=fitted(Lake1.step.model0),
y=residuals(Lake1.step.model0),
shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Homocedasticity test)",
x="Fitted", y = "Residuals")

pH02 <-ggplot(LAKE1, aes(x=fitted(Lake1.step.model0),
y=sqrt(abs(residuals(Lake1.step.model0))),
shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Homocedasticity test)",
x="Fitted", y = expression(sqrt("|residuals|")))

grid.arrange(pH01, pH02,ncol=2)
plot(Lake1.step.model0,which=1)

BP01<-bptest(Lake1.step.model0)
BP01

```

```

pH11 <- ggplot(LAKE1, aes(x=fitted(Lake1.step.model1),
y=residuals(Lake1.step.model1),
shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Homocedasticity test)",
x="Fitted", y = "Residuals")

pH12 <-ggplot(LAKE1, aes(x=fitted(Lake1.step.model1),
y=sqrt(abs(residuals(Lake1.step.model1))),
shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Homocedasticity test)",
x="Fitted", y = expression(sqrt("|residuals|")))

grid.arrange(pH11, pH12,ncol=2)
plot(Lake1.step.model1,which=1)

BP11<-bptest(Lake1.step.model1)
BP11

pH21 <- ggplot(LAKE1, aes(x=fitted(Lake1.step.model2),
y=residuals(Lake1.step.model2),
shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Homocedasticity test)",
x="Fitted", y = "Residuals")

pH22 <-ggplot(LAKE1, aes(x=fitted(Lake1.step.model2),
y=sqrt(abs(residuals(Lake1.step.model2))),
shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Homocedasticity test)",
x="Fitted", y = expression(sqrt("|residuals|")))

grid.arrange(pH21, pH22,ncol=2)
plot(Lake1.step.model2,which=1)

BP21<-bptest(Lake1.step.model2)
BP21

par(mfrow=c(1,3))
plot(Lake1.step.model0,which=1)
plot(Lake1.step.model1,which=1)
plot(Lake1.step.model2,which=1)
` ``

```

Els gràfics es veu que:

+ Semblaria clarament que al model original es té un problema de heteroscedasticitat amb els residus que semblen seguir clarament una tendència negativa excepte per uns quants punts. També ho suporta el test de \*Breusch-Pagan\* amb un p-valor de `r paste(round(BP01\$p.value,4))`.

+ Al transformar logarítmicament sembla que es compensa tot i donar també aparentment un patró de comportament negatiu i amb un p-valor de BP de `r paste(round(BP11\$p.value,4))`.



+ En la transformació Cox-Box seria potser l'únic model que donaria una homogeneïtat aparent residual tot i que també sembla seguir una certa forma no lineal, i amb un p-valor de BP de `round(BP21$p.value,4)`.

En aquest punt es podria plantejar algun altre tipus de transformació, plantejar un model no lineal o intentar aplicar funcions dins el model que modulin la variància del model. D'acord amb el que es planteja en el treball interessa en aquest punt posar en pràctica les funcions de modulació per intentar millorar el model, però això es veurà en la part d'aplicació de models mixtes.

S'observa addicionalment que la variància no es diferencia de manera significativa aparentment entre els dos grups de tractament en els 3 models ajustats simples:

```
```{r Residus Lake1}
by(residuals(Lake1.step.model0), LAKE1$Grupo, var)
by(residuals(Lake1.step.model1), LAKE1$Grupo, var)
by(residuals(Lake1.step.model2), LAKE1$Grupo, var)
```
```

### Diagnòstic possibles outliers

En el cas concret del model original hi ha un valor que a nivell de residus estandarditzats o estudentitzats apareix amb una distància fora del comú tant per els gràfics de normalitat com els de homogeneïtat de residus:

```
```{r Outlier LAKE1 Original}
LAKE1[which.max(qq01$y),]
LAKE1[which.max(sqrt(abs(residuals(Lake1.step.model0))))],]
```
```

Habitualment s'investigaria si és un possible *outlier* i els motius com es va obtenir el valor per veure si hi ha algun error que s'hagi de corregir. En aquest cas s'opta per fer una eliminació del valor d'aquesta observació i veure les diferències (es grafica directe els QQ *plot* de residus estandarditzats i estudentitzats per veure més clarament els valors aberrants):

```
```{r Outlier LAKE1 Original LM}
Lake1.step.model0s

LAKE1c <- LAKE1[-which.max(qq01$y),]
# Fit the full model
Lake1c.fm0 <- lm(CargaViral ~ . - LogCargaViral - CargaViral_t -
Ident, data = LAKE1c)
# Stepwise regression model
Lake1c.step.model0 <- stepAIC(Lake1c.fm0, direction = "both",
trace = F)
(Lake1c.step.model0s <- summary(Lake1c.step.model0))

Lake1c.SM0.sdres <- rstandard(Lake1c.step.model0)
Lake1c.SM0.studres <- studres(Lake1c.step.model0)

par(mfrow=c(1,2))

qqnorm(Lake1c.SM0.sdres,
ylab="Standardized Residuals",
xlab="Normal Scores",
main="Model Original - NPP of Standardized Residuals")
```

```

qqline(Lake1c.SM0.sdres)

qq02<-qqnorm(Lake1c.SM0.studres, #Es converteix en objecte per
investigar el possible outlier a continuació
            ylab="Studentized Residuals",
            xlab="Normal Scores",
            main="Model Original - NPP of Studentized Residuals")
qqline(Lake1c.SM0.studres)

S0 <- shapiro.test(Lake1c.step.model0$residuals)
S0

LAKE1c[which.max(qq02$y),]
````

```

Es mostra un gran canvi ja que el sistema deixa de tenir en compte els 2 predictors que tenia en compte abans \*`r attr(Lake1.step.model0\$terms,"term.labels")`\* i té en compte ara `r attr(Lake1c.step.model0\$terms,"term.labels")`, però es continua veient que al gràfic QQ estudentitzat i estandarditzat punts que estan exageradament per fora de la mostra. Aquí cal plantejar-se si aquests punts són tots \*outliers\*, si formen part d'una població diferent o si el model li falta un ajust addicional per contemplar aquesta diferenciació.

Si s'eliminés el segon i tercer punts del model s'obtindrien els següents resultats:

```

````{r Outlier LAKE1 Original LM 2}
Lake1.step.model0s
Lake1c.step.model0s

LAKE1c2 <- LAKE1c[-which.max(qq02$y),]
# Fit the full model
Lake1c2.fm0 <- lm(CargaViral ~ . - LogCargaViral - CargaViral_t -
Ident, data = LAKE1c2)
# Stepwise regression model
Lake1c2.step.model0 <- stepAIC(Lake1c2.fm0, direction = "both",
                             trace = F)
(Lake1c2.step.model0s <- summary(Lake1c2.step.model0))

Lake1c2.SM0.sdres <- rstandard(Lake1c2.step.model0)
Lake1c2.SM0.studres <- studres(Lake1c2.step.model0)

```

```

par(mfrow=c(1,2))

qqnorm(Lake1c2.SM0.sdres,
        ylab="Standardized Residuals",
        xlab="Normal Scores",
        main="Model Original - NPP of Standardized Residuals")
qqline(Lake1c2.SM0.sdres)

qq03<-qqnorm(Lake1c2.SM0.studres, #Es converteix en objecte per
investigar el possible outlier a continuació
            ylab="Studentized Residuals",
            xlab="Normal Scores",
            main="Model Original - NPP of Studentized Residuals")
qqline(Lake1c2.SM0.studres)

S0 <- shapiro.test(Lake1c2.step.model0$residuals)
S0

```

```

LAKE1c2[which.max(qq03$y),]

LAKE1c3 <- LAKE1c2[-which.max(qq02$y),]
# Fit the full model
Lake1c3.fm0 <- lm(CargaViral ~ . - LogCargaViral - CargaViral_t -
Ident, data = LAKE1c3)
# Stepwise regression model
Lake1c3.step.model0 <- stepAIC(Lake1c3.fm0, direction = "both",
                             trace = F)
(Lake1c3.step.model0s <- summary(Lake1c3.step.model0))

Lake1c3.SM0.sdres <- rstandard(Lake1c3.step.model0)
Lake1c3.SM0.studres <- studres(Lake1c3.step.model0)

par(mfrow=c(1,2))

qqnorm(Lake1c3.SM0.sdres,
       ylab="Standardized Residuals",
       xlab="Normal Scores",
       main="Model Original - NPP of Standardized Residuals")
qqline(Lake1c3.SM0.sdres)

qq04<-qqnorm(Lake1c3.SM0.studres, #Es converteix en objecte per
investigar el possible outlier a continuació
             ylab="Studentized Residuals",
             xlab="Normal Scores",
             main="Model Original - NPP of Studentized Residuals")
qqline(Lake1c3.SM0.studres)

S0 <- shapiro.test(Lake1c3.step.model0$residuals)
S0

LAKE1c3[which.max(qq04$y),]
````

```

En el tercer dels modelatges curiosament el sistema afegeix la variable tractament de Grup que fins ara s'havien descartat com a significativa i també és interessant veure que tots els possibles \*outliers\* provenen de la mateixa categoria de la variable `nusuari` del \*dataset\* original. Per comprovar possibles diferències es fan alguns gràfics descriptius de les 3 transformacions:

```

````{r Test grup. nusuari LAKE1}
LAKE1_test <- LAKE1
LAKE1_test$nusuari <- LAKE1_noNA$nusuari
table(LAKE1_test$nusuari)

Lake1_testSCPlot.a <- ggplot(LAKE1_test, aes(x=Tiempo, y=CargaViral,
color=nusuari, shape=nusuari)) +
  geom_point()+
  geom_smooth(method=lm,se=F)+
  labs(title="Càrrega viral s/ nusuari",
       x="Tiempo", y = "CargaViral")+
  scale_color_brewer(palette="Accent")

Lake1_testSCPlot.b <- ggplot(LAKE1_test, aes(x=Tiempo, y=CargaViral,
color=nusuari, shape=Grupo)) +
  geom_point()+
  geom_smooth(method=lm,se=F)+
  labs(title="Càrrega viral s/ tract. y nusuari",
       x="Tiempo", y = "CargaViral")+

```

```

scale_color_brewer(palette="Accent")

grid.arrange(Lake1_testSCPlot.a,Lake1_testSCPlot.b)

Lake1_testBPlot.a <- ggplot(LAKE1_test, aes(x=nusuariu, y=CargaViral,
fill=nusuariu)) +
  geom_boxplot()+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1)+
  labs(title="Càrrega viral s/ nusuariu",
        x="Tractament", y = "CargaViral")+
  scale_fill_brewer(palette="Accent")

Lake1_testBPlot.b <- ggplot(LAKE1_test, aes(x=nusuariu, y=CargaViral,
fill=Grup)) +
  geom_boxplot()+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1)+
  labs(title="Càrrega viral s/ tract. y nusuariu",
        x="Tractament", y = "CargaViral")+
  scale_fill_brewer(palette="Accent")

grid.arrange(Lake1_testBPlot.a,Lake1_testBPlot.b)

```

Log:

```

`r Test grup. nusuari LAKE1 log}
Lake1_testSCPlot.a <- ggplot(LAKE1_test, aes(x=Tiempo,
y=LogCargaViral, color=nusuariu, shape=nusuariu)) +
  geom_point()+
  geom_smooth(method=lm,se=F)+
  labs(title="Càrrega viral s/ nusuariu",
        x="Tiempo", y = "LogCargaViral")+
  scale_color_brewer(palette="Accent")

Lake1_testSCPlot.b <- ggplot(LAKE1_test, aes(x=Tiempo,
y=LogCargaViral, color=nusuariu, shape=Grup)) +
  geom_point()+
  geom_smooth(method=lm,se=F)+
  labs(title="Càrrega viral s/ tract. y nusuariu",
        x="Tiempo", y = "LogCargaViral")+
  scale_color_brewer(palette="Accent")

grid.arrange(Lake1_testSCPlot.a,Lake1_testSCPlot.b)

Lake1_testBPlot.a <- ggplot(LAKE1_test, aes(x=nusuariu,
y=LogCargaViral, fill=nusuariu)) +
  geom_boxplot()+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1)+
  labs(title="Càrrega viral s/ nusuariu",
        x="Tractament", y = "LogCargaViral")+
  scale_fill_brewer(palette="Accent")

Lake1_testBPlot.b <- ggplot(LAKE1_test, aes(x=nusuariu,
y=LogCargaViral, fill=Grup)) +
  geom_boxplot()+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1)+
  labs(title="Càrrega viral s/ tract. y nusuariu",
        x="Tractament", y = "LogCargaViral")+
  scale_fill_brewer(palette="Accent")

grid.arrange(Lake1_testBPlot.a,Lake1_testBPlot.b)

```

```

**\*Box-Cox\***

```
```{r Test grup. nusuari LAKE1 BC}
Lake1_testSCPlot.a <- ggplot(LAKE1_test, aes(x=Tiempo, y=CargaViral_t,
color=nusuario, shape=nusuario)) +
  geom_point()+
  geom_smooth(method=lm,se=F)+
  labs(title="Càrrega viral s/ nusuario",
        x="Tiempo", y = "CargaViral_t")+
  scale_color_brewer(palette="Accent")

Lake1_testSCPlot.b <- ggplot(LAKE1_test, aes(x=Tiempo, y=CargaViral_t,
color=nusuario, shape=Grupo)) +
  geom_point()+
  geom_smooth(method=lm,se=F)+
  labs(title="Càrrega viral s/ tract. y nusuario",
        x="Tiempo", y = "CargaViral_t")+
  scale_color_brewer(palette="Accent")

grid.arrange(Lake1_testSCPlot.a,Lake1_testSCPlot.b)

Lake1_testBPlot.a <- ggplot(LAKE1_test, aes(x=nusuario,
y=CargaViral_t, fill=nusuario)) +
  geom_boxplot()+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1)+
  labs(title="Càrrega viral s/ nusuario",
        x="Tractament", y = "CargaViral_t")+
  scale_fill_brewer(palette="Accent")

Lake1_testBPlot.b <- ggplot(LAKE1_test, aes(x=nusuario,
y=CargaViral_t, fill=Grupo)) +
  geom_boxplot()+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1)+
  labs(title="Càrrega viral s/ tract. y nusuario",
        x="Tractament", y = "CargaViral_t")+
  scale_fill_brewer(palette="Accent")

grid.arrange(Lake1_testBPlot.a,Lake1_testBPlot.b)

LAKE1c2_test <- LAKE1_test[-c(which.max(qq01$y),which.max(qq02$y)),]

Lake1c2_testBPlot.a <- ggplot(LAKE1c2_test, aes(x=nusuario,
y=CargaViral_t, fill=nusuario)) +
  geom_boxplot()+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1)+
  labs(title="Càrrega viral s/ nusuario (- 2 outl.)",
        x="Tractament", y = "CargaViral_t")+
  scale_fill_brewer(palette="Accent")

Lake1c2_testBPlot.b <- ggplot(LAKE1c2_test, aes(x=nusuario,
y=CargaViral_t, fill=Grupo)) +
  geom_boxplot()+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1)+
  labs(title="Càrrega viral s/ tract. y nusuario (- 2 outl.)",
        x="Tractament", y = "CargaViral_t")+
  scale_fill_brewer(palette="Accent")

grid.arrange(Lake1c2_testBPlot.a,Lake1c2_testBPlot.b)
```
```

No queda clara una diferenciació significativa entre aquest 2 grups de `nusuarió`, però sí que s'observa algunes punts interessants que podrien explicar el que s'ha vist en aquest apartat:

+ Pels resultats sembla que `r LAKE1\_test\$nusuarió[which.max(LAKE1\_test\$CargaViral)]` és el grup de `nusuarió` que té els valors més extrems i això explicaria possiblement perquè els 3 primers valors aberrants que s'eliminarien són d'aquest grup. Possiblement si es seguis eliminant valors d'aquesta categoria en algun moment es trobarien els primers aberrants de l'altre grup de `nusuarió`.

+ Quan s'avança en l'eliminació de punts del grup més extrems `r LAKE1\_test\$nusuarió[which.max(LAKE1\_test\$CargaViral)]` l'altre grup `aocampo` pren més pes i també canvia la proporció entre tractaments pel primer grup. Tot i així es queda en un model no balancejat de temps i `nusuarió` pel que també es perdria certesa.

Sembla que en els dos grups hi ha un conjunt de valors relativament gran que es mostren com aberrants i influents ja que al eliminar-los fan variar significativament el model en cada eliminació. Eliminar-los no seria un procediment correcte sense una justificació com ara una variable oculta que el convertís en una nova població. Com que no es té aquesta evidència, s'opta per no eliminar cap d'ells i amb avançar en paral·lel amb els conjunts transformats que mitigarien aquest efecte.

### ### Colinealitat

En els models on apareix més d'un predictor es pot donar el cas que algun predictor sigui una combinació lineal dels altres donant lloc a una matriu  $X^T X$  singular o encara que no hi hagi una combinació lineal perfecte una aproximació a la matriu singular. En aquests casos anomenats de multicolinealitat hi pot haver un problema al tenir més d'un càlcul de mínims quadrats possibles pel coeficient del predictor i això sovint provoca imprecisions en l'estimació de  $\beta$ .

A continuació es realitzen algunes proves en el model on apareix més d'un predictor per detectar si hi ha algun problema en aquest sentit:

+ Prova de correlació. Es torna a mostrar la matriu de correlacions per aquestes variables concretes per veure quines correlacions mostren:

```
```{r Colin LAKE1 corr}
LAKE1col <- LAKE1[,attr(Lake1.step.model0$terms,"term.labels")]

pairs.panels(LAKE1col,
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE, # show density plots
             ellipses = TRUE # show correlation ellipses
             )
LAKE1colcor <- as.data.frame(cor(LAKE1col))
LAKE1colcor2 <- LAKE1colcor[which(LAKE1colcor$Tiempo!=1),] #Es mostra
el més gran relacionat amb Temps
LAKE1colmaxpred <- LAKE1colcor2[which.max(abs(LAKE1colcor2$Tiempo)),]
rownames(LAKE1colcor2[which.max(abs(LAKE1colcor2$Tiempo)),])
```
```

Semblaria en aquest primer diagnòstic que la variable amb correlació més gran amb la principal covariant `Tiempo` seria `r LAKE1maxpred`.

+ Prova de regressió del predictor principal en funció dels altres:

```
```{r Colin LAKE1 reg}
(LAKE1col.lms<-summary(LAKE1col.lm <- lm(Tiempo ~ .,data=LAKE1col))
````
```

També ho s'observa en aquest resultat al comprovar els valors de significació dels diferents predictors secundaris \*`r attr(LAKE1col.lm\$terms,"term.labels")`\* amb p-valors de `r paste(round(LAKE1col.lms\$coefficients[2:(1+length(attr(LAKE1col.lm\$terms,"term.labels"))],4),4))` respectivament.

+ Càlcul dels valors propis de  $X^T X$ : La presència de alguns valors propis petits indica habitualment la multicolinealitat. Es pot calcular mitjançant un coeficient  $k$  amb el ratio entre el valor propi principal i els altres  $k = \sqrt{\frac{\lambda_1}{\lambda_p}}$  per veure els mides relatius (habitualment es considera que per sobre de 30 ja és suficientment gran per no ser un problema):

```
```{r Colin LAKE1 eigenval}
x <- model.matrix(Lake1.step.model0)[-1]
e <- eigen(t(x) %*% x)
e$val
(k<-sqrt(e$val[1]/e$val)[-1])
````
```

Ratios bastant petits i essent per sota de 30 els ratios amb \*`r colnames(x)[which(k<30)+1]`\*, sobretot el ratio de \*`r colnames(x)[which.min(k)+1]`\* amb un ratio de `r round(k[which.min(k)],2)`\*, indicant possible multicolinealitat.

+ Prova de VIF: Segons la fórmula següent es pot veure una mesura de colinealitat en el càlcul de la variància del coeficient del predictor  $\text{var}(\hat{\beta}_j) = \sigma^2 \frac{1}{1-R^2_j} \frac{1}{\sum_i (x_{ij} - \bar{x}_j)^2}$  El factor de inflació de variància corresponent a cada predictor  $\frac{1}{1-R^2_j}$  que provoca un augment de la variància del coeficient en cas que sigui gran, és el que indica la possible multicolinealitat del predictor:

```
```{r Colin LAKE1 vif}
vif(x)
````
```

En aquest cas no es mostra una gran inflació de variància en cap dels predictors, pel que no s'aconsegueix una prova molt definitiva de presència de colinealitat.

Es mostra les diferències en l'ajust del model simple:

```
```{r Colin LAKE1 Anova}
Lake1.step.model0s
(Lake1.step.model0col.s<-summary(Lake1.step.model0col<-
update(Lake1.step.model0,paste(".", ~ . "-"),colnames(x)[which.min(k)+1],sep=" ")))
(Lake1.step.model0col.an<-
anova(Lake1.step.model0col,Lake1.step.model0))
````
```

Per un marge molt estret no es consideren significativament diferents els dos models contrastats amb anova amb un p-valor de `round(Lake1.step.model0col.an$Pr(>F)"[2],4)`, però sí que es pot veure que en el segon model la variable temps pren més significació passant de un p-valor de `round(Lake1.step.model0s$coefficients["Tiempo",4],4)` a `round(Lake1.step.model0col.s$coefficients["Tiempo",4],4)`, encara que també es redueix el coeficient de determinació  $R^2$  absolut i ajustat de `round(Lake1.step.model0s$r.squared,4)` a `round(Lake1.step.model0col.s$r.squared,4)` percent % d'explicabilitat de la variància de la resposta.

S'està doncs en una situació de compromís on no hi ha una prova definitiva entre les fetes que doni una direcció molt clara. Com que l'objectiu és crear un model de predicció es sacrificarà tenir incertesa en els coeficients dels predictors en front de tenir un % d'explicabilitat més gran de la resposta agafant el model amb més variables, encara que hi hagi certa colinealitat present entre `attr(Lake1.step.model0s$terms,"term.labels")`. En tot cas en les properes passes s'ha de tenir en compte com una opció la d'eliminar aquest predictor amb colinealitat en cas que es compensi amb altres estructures més eficients.

# COMPROVACIÓ INTERACCIONS MODEL

## Paralelisme o Interacció Individu-Temps

És d'interès comprovar l'efecte que té el factor individu amb la covariant temps per plantejar-nos si en propers passos es considera l'efecte aleatori d'aquest. Per fer això un dels sistemes és afegir el factor individu dins el model i construir el terme creuat amb la covariant temps per veure si aquest factor és significatiu, és a dir si la variació de les observacions amb la covariant temps depenen del factor individu o no (seria l'equivalent a la prova de paral·lelisme dels models de regressió):

```
```{r Model LAKE1 Factor individu x temps}
#Model original
Lake1.step.model0Is <- summary(Lake1.step.model0I<-
update(Lake1.step.model0,. ~ . + Ident*Tiempo)) #S'afegeix el factor
creuat al model original
(Lake1.step.model0I.an<-Anova(Lake1.step.model0I))

summary(aov(formula(paste(Lake1.step.model0I$call[2])),data=LAKE1))

#Model amb correcció per colinealitat
Lake1.step.model0colIs <- summary(Lake1.step.model0colI<-
update(Lake1.step.model0col,. ~ . + Ident*Tiempo)) #S'afegeix el
factor creuat al model original
(Lake1.step.model0colI.an<-Anova(Lake1.step.model0colI))

summary(aov(formula(paste(Lake1.step.model0colI$call[2])),data=LAKE1))

#Model Log
Lake1.step.model1Is <- summary(Lake1.step.model1I<-
update(Lake1.step.model1,. ~ . + Ident*Tiempo)) #S'afegeix el factor
creuat al model original
(Lake1.step.model1I.an<-Anova(Lake1.step.model1I))
```



```
summary(aov(formula(paste(Lake1.step.model1I$call[2])),data=LAKE1))

#Model Box-Cox
Lake1.step.model2Is <- summary(Lake1.step.model2I<-
update(Lake1.step.model2,. ~ . + Ident*Tiempo)) #S'afegeix el factor
creuat al model original
(Lake1.step.model2I.an<-Anova(Lake1.step.model2I))

summary(aov(formula(paste(Lake1.step.model2I$call[2])),data=LAKE1))
````
```

Es considera els resultats de p-valor del factor interacció que inferiors al 5% de significància com a significativa la variació de pendent per individu. En aquest cas els resultats dels p-valors dels diferents models Original, Original corregit per multicolinealitat, Log transformat i amb transformació Box-Cox són de `round(Lake1.step.model0I.an$`Pr(>F)`[length(Lake1.step.model0I.an$`Pr(>F)`)-1],4)``, `round(Lake1.step.model0colI.an$`Pr(>F)`[length(Lake1.step.model0colI.an$`Pr(>F)`)-1],4)``, `round(Lake1.step.model1I.an$`Pr(>F)`[length(Lake1.step.model1I.an$`Pr(>F)`)-1],4)`` i `round(Lake1.step.model2I.an$`Pr(>F)`[length(Lake1.step.model2I.an$`Pr(>F)`)-1],4)`` respectivament.

Segons els resultats en principi no s'aplicarien mesures per ajustar el model a les variacions de pendents dependent del factor individu.

#### ## Altres Interaccions

En el model original s'ha considerat més variables predictores a part de la variable temps. Seria interessant veure si la interacció entre aquestes variables té alguna significació que millori l'explicabilitat del model i per això es fan els ajustos corresponents per veure les proves ANOVA de significància. Es fa també el procés de stepwise amb el model complet d'interaccions:

```
````{r Model Original LAKE1 Interaccions}
#Original
Inter.0 <- attr(Lake1.step.model0s$terms,"term.labels")
Lake1.step.model0int.s <- summary(Lake1.step.model0int <-
update(Lake1.step.model0,paste(". ~ . +",paste(Inter.0,collapse="*"),sep=" "))
(Lake1.step.model0int2.s2 <- summary(Lake1.step.model0int2 <-
stepAIC(Lake1.step.model0int, direction = "both", trace = F))
(Lake1.step.model0int2.an2<-Anova(Lake1.step.model0int2))
(Lake1.step.model0int2.an1_2<-
anova(Lake1.step.model0int2,Lake1.step.model0))

if (Lake1.step.model0int2.an1_2$`Pr(>F)`[2]>0.05){
  intM0 <- Lake1.step.model0
  intM0s <- Lake1.step.model0s
} else {
  intM0 <- Lake1.step.model0int2
  intM0s <- Lake1.step.model0int2.s2
}

intM0col <- Lake1.step.model0col
intM0cols <- Lake1.step.model0col.s
```

...

En el cas dels models transformats no seria necessari en principi ja que només ha quedat un predictor com a model òptim i per tant no té altres predictors per creuar-se. Tot i així es fa la prova on apareixen totes les variables amb interaccions per comprobar la diferència entre models:

```
```{r Model Transf LAKE1 Interaccions}
#Log
#Step1
Lake1.step.model1int.s      <-      summary(Lake1.step.model1int      <-
update(Lake1.step.model1, paste(" ~
+", paste(Inter.0, collapse="*"), sep=" "))
(Lake1.step.model1int2.s2    <-      summary(Lake1.step.model1int2    <-
stepAIC(Lake1.step.model1int, direction = "both", trace = F))
(Lake1.step.model1int2.an2<-Anova(Lake1.step.model1int2))
(Lake1.step.model1int2.an1_2                                     <-
anova(Lake1.step.model1int2, Lake1.step.model1))

if (Lake1.step.model1int2.an1_2$`Pr(>F)`[2]>0.05){
  intM1 <- Lake1.step.model1
  intM1s <- Lake1.step.model1s
} else {
  intM1 <- Lake1.step.model1int2
  intM1s <- Lake1.step.model1int2.s2
}

#Box-Cox
#Step1
Lake1.step.model2int.s      <-      summary(Lake1.step.model2int      <-
update(Lake1.step.model2, paste(" ~
+", paste(Inter.0, collapse="*"), sep=" "))
(Lake1.step.model2int2.s2    <-      summary(Lake1.step.model2int2    <-
stepAIC(Lake1.step.model2int, direction = "both", trace = F))
(Lake1.step.model2int2.an2<-Anova(Lake1.step.model2int2))
(Lake1.step.model2int2.an1_2                                     <-
anova(Lake1.step.model2int2, Lake1.step.model2))

if (Lake1.step.model2int2.an1_2$`Pr(>F)`[2]>0.05){
  intM2 <- Lake1.step.model2
  intM2s <- Lake1.step.model2s
} else {
  intM2 <- Lake1.step.model2int2
  intM2s <- Lake1.step.model2int2.s2
}
```

```{r Taules2}
intM1s_mod <- as.data.frame(round(intM1s$coefficients[-1, -3], 4))
intM1s_mod[4, ]<-intM1s_mod[3, ]
intM1s_mod[c(3, 5:7), ]<-NA

intM2s_mod <- as.data.frame(round(intM2s$coefficients[-1, -3], 4))
intM2s_mod[4, ]<-intM2s_mod[3, ]
intM2s_mod[c(3, 5:7), ]<-NA

Lake1.it1 <- (cbind(round(intM0s$coefficients[-1, -3], 4),
                    intM1s_mod,
                    intM2s_mod))
```

```

kable(Lake1.it1) %>%
  kable_styling(bootstrap_options = "striped", full_width = F) %>%
  add_header_above(c(" " = 1, "M.Int sense transf. 1" = 3, "M.Int log
transf" = 3, "M.Int Box-Cox t" = 3))

Lake1.it2 <- data.frame(Model=c("M.Int sense transf","M.Int log
transf","M.Int Box-Cox t"))
Lake1.it2$"Mediana residual" <- c(median(intM0$residuals),
                                median(intM1$residuals),
                                median(intM2$residuals))

kable(Lake1.it2) %>%
  kable_styling("striped")
````

```

S'ha tingut en compte per un costat les interaccions entre els termes existents dels models simples optimitzats i, excepte en el cas de reducció per colinealitat, s'ha comprovat l'efecte de l'aparició dels factors d'interacció per paràmetres que no estaven en el model òptim simple en els de variable de resposta transformada. A continuació es resumeix el model trobat com a òptim al fer els càlculs i la p-valor del model òptim d'interacció amb el model simple òptim del qual parteixen:

```

+ Pel model original el model òptim seria *`r paste(intM0$call[2])`*
i s'ha obtingut un p-valor entre models de `r
paste(round(Lake1.step.model0int2.an1_2$"Pr(>F)"[2],4))`.
+ Pel model original el model òptim seria *`r paste(intM1$call[2])`*
i s'ha obtingut un p-valor entre models de `r
paste(round(Lake1.step.model1int2.an1_2$"Pr(>F)"[2],4))`.
+ Pel model original el model òptim seria *`r paste(intM2$call[2])`*
i s'ha obtingut un p-valor entre models de `r
paste(round(Lake1.step.model2int2.an1_2$"Pr(>F)"[2],4))`.

```

### Diagnòstics models interaccions

S'accepta per tant de moment els models òptims amb la presència d'interaccions i es torna a generar alguns dels diagnòstics per observar les diferències:

```

````{r Diag Model simple Lake1 INT}
Lake1Int.SM0.sdres <- rstandard(intM0)
Lake1Int.SM0.studres <- studres(intM0)
par(mfrow=c(2,2))

qqnorm(Lake1.SM0.sdres,
       ylab="Standardized Residuals",
       xlab="Normal Scores",
       main="M. Original - Standardized Residuals")
qqline(Lake1.SM0.sdres)

qqnorm(Lake1.SM0.studres,
       ylab="Studentized Residuals",
       xlab="Normal Scores",
       main="M. Original - Studentized Residuals")
qqline(Lake1.SM0.studres)

qqnorm(Lake1Int.SM0.sdres,
       ylab="Standardized Residuals",
       xlab="Normal Scores",
       main="M. Original Int. - Standardized Residuals")
qqline(Lake1Int.SM0.sdres)

```

```

qqnorm(Lake1Int.SM0.studres,
       ylab="Studentized Residuals",
       xlab="Normal Scores",
       main="M. Original Int. - Studentized Residuals")
qqline(Lake1Int.SM0.studres)

S0
(S0Int <- shapiro.test(intM0$residuals))

Lake1.SM0col.sdres <- rstandard(Lake1.step.model0col)
Lake1.SM0col.studres <- studres(Lake1.step.model0col)
Lake1Int.SM0col.sdres <- rstandard(intM0col)
Lake1Int.SM0col.studres <- studres(intM0col)
par(mfrow=c(2,2))

qqnorm(Lake1.SM0col.sdres,
       ylab="Standardized Residuals",
       xlab="Normal Scores",
       main="M. Original Correcció Col. - Standardized Residuals")
qqline(Lake1.SM0col.sdres)

qqnorm(Lake1.SM0col.studres,
       ylab="Studentized Residuals",
       xlab="Normal Scores",
       main="M. Original Correcció Col. - Studentized Residuals")
qqline(Lake1.SM0col.studres)

qqnorm(Lake1Int.SM0col.sdres,
       ylab="Standardized Residuals",
       xlab="Normal Scores",
       main="M. Original Int. Correcció Col. - Standardized
Residuals")
qqline(Lake1Int.SM0col.sdres)

qqnorm(Lake1Int.SM0col.studres,
       ylab="Studentized Residuals",
       xlab="Normal Scores",
       main="M. Original Int. Correcció Col. - Studentized Residuals")
qqline(Lake1Int.SM0col.studres)

(S0col <- shapiro.test(Lake1.step.model0col$residuals))
(S0colInt <- shapiro.test(intM0col$residuals))

Lake1Int.SM1.sdres <- rstandard(intM1)
Lake1Int.SM1.studres <- studres(intM1)
par(mfrow=c(2,2))

qqnorm(Lake1.SM1.sdres,
       ylab="Standardized Residuals",
       xlab="Normal Scores",
       main="M. Log - Standardized Residuals")
qqline(Lake1.SM1.sdres)

qqnorm(Lake1.SM1.studres,
       ylab="Studentized Residuals",
       xlab="Normal Scores",
       main="M. Log - Studentized Residuals")
qqline(Lake1.SM1.studres)

qqnorm(Lake1Int.SM1.sdres,

```

```

      ylab="Standardized Residuals",
      xlab="Normal Scores",
      main="M. Log Int.- Standardized Residuals")
qqline(Lake1Int.SM1.sdres)

qqnorm(Lake1Int.SM1.studres,
      ylab="Studentized Residuals",
      xlab="Normal Scores",
      main="M. Log Int.- Studentized Residuals")
qqline(Lake1Int.SM1.studres)

S1
(S1Int <- shapiro.test(intM1$residuals))

Lake1Int.SM2.sdres <- rstandard(intM2)
Lake1Int.SM2.studres <- studres(intM2)
par(mfrow=c(2,2))

qqnorm(Lake1.SM2.sdres,
      ylab="Standardized Residuals",
      xlab="Normal Scores",
      main="M. Box-Cox - Standardized Residuals")
qqline(Lake1.SM2.sdres)

qqnorm(Lake1.SM2.studres,
      ylab="Studentized Residuals",
      xlab="Normal Scores",
      main="M. Box-Cox - Studentized Residuals")
qqline(Lake1.SM2.studres)

qqnorm(Lake1Int.SM2.sdres,
      ylab="Standardized Residuals",
      xlab="Normal Scores",
      main="M. Box-Cox Int.- Standardized Residuals")
qqline(Lake1Int.SM2.sdres)

qqnorm(Lake1Int.SM2.studres,
      ylab="Studentized Residuals",
      xlab="Normal Scores",
      main="M. Box-Cox Int.- Studentized Residuals")
qqline(Lake1Int.SM2.studres)

S2
(S2Int <- shapiro.test(intM2$residuals))

```

...

Tot i que en general sembla millorar la normalitat pel test de Shapiro en els gràfics queda clar pel model original (tant complet com corregit per la colinealitat), però no tant aparent en els models transformats.

```

```{r Diag H Model simple Lake1 INT}
pHInt01 <- ggplot(LAKE1, aes(x=fitted(intM0), y=residuals(intM0),
      shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Int.Homocedasticity test)",
      x="Fitted", y = "Residuals")

```

```

pHInt02 <- ggplot(LAKE1, aes(x=fitted(intM0),
y=sqrt(abs(residuals(intM0))),
shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Int.Homocedasticity test)",
x="Fitted", y = expression(sqrt("|residuals|")))

grid.arrange(pH01, pH02, pHInt01, pHInt02,ncol=2)
par(mfrow=c(1,2))
plot(Lake1.step.model0,which=1)
plot(intM0,which=1)

BP01
(BPInt01<-bptest(intM0))

pHcol01 <- ggplot(LAKE1, aes(x=fitted(Lake1.step.model0col),
y=residuals(Lake1.step.model0col),
shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Col.Homocedasticity test)",
x="Fitted", y = "Residuals")

pHcol02 <- ggplot(LAKE1, aes(x=fitted(Lake1.step.model0col),
y=sqrt(abs(residuals(Lake1.step.model0col))),
shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Col.Homocedasticity test)",
x="Fitted", y = expression(sqrt("|residuals|")))

pHcolInt01 <- ggplot(LAKE1, aes(x=fitted(intM0col),
y=residuals(intM0col),
shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (colInt.Homocedasticity test)",
x="Fitted", y = "Residuals")

pHcolInt02 <- ggplot(LAKE1, aes(x=fitted(intM0col),
y=sqrt(abs(residuals(intM0col))),
shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (colInt.Homocedasticity test)",
x="Fitted", y = expression(sqrt("|residuals|")))

grid.arrange(pHcol01, pHcol02, pHcolInt01, pHcolInt02,ncol=2)
par(mfrow=c(1,2))
plot(Lake1.step.model0col,which=1)
plot(intM0col,which=1)

(BPcol01<-bptest(Lake1.step.model0col))
(BPcolInt01<-bptest(intM0col))

pHInt11 <- ggplot(LAKE1, aes(x=fitted(intM1), y=residuals(intM1),
shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Log.Int.Hom. test)",

```

```

x="Fitted", y = "Residuals")

pHInt12          <-ggplot(LAKE1,          aes(x=fitted(intM1),
y=sqrt(abs(residuals(intM1))),
                shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Log.Int.Hom. test)",
        x="Fitted", y = expression(sqrt("|residuals|")))

grid.arrange(pH11, pH12,pHInt11,pHInt12,ncol=2)
plot(Lake1.step.model1,which=1)
plot(intM1,which=1)

BP11
(BPInt11<-bptest(intM1))

pHInt21 <- ggplot(LAKE1, aes(x=fitted(intM2), y=residuals(intM2),
                             shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Int.BC.Hom test)",
        x="Fitted", y = "Residuals")

pHInt22          <-ggplot(LAKE1,          aes(x=fitted(intM2),
y=sqrt(abs(residuals(intM2))),
                shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Int.BC.Hom test)",
        x="Fitted", y = expression(sqrt("|residuals|")))

grid.arrange(pH21, pH22,pHInt21,pHInt22,ncol=2)
plot(Lake1.step.model2,which=1)
plot(intM2,which=1)

BP21
(BPint21<-bptest(intM2))
````

```

En els dos models sense transformació, independentment del test d'homogeneïtat els gràfics segueixen mostrant un patró que no correspondria a la homocedasticitat de residus i que podria ser preocupant per assumir aquesta condició. En els altres dos models que s'havien transformat, el cas logarítmic millora lleugerament tot o que encara no es pot assegurar veient els gràfics que hi hagi la condició, i el cas del transformat de box cox potser és l'únic cas on no es veuria tan clarament l'heteroscedasticitat i inclús milloraria en els models amb els termes d'interacció.

En resum els models generats per les proves de diagnòstic encara poden patir les conseqüències de no complir les 2 assumpcions principals en els resultats d'estimació i altres contrastos d'hipòtesis, pel que s'ha d'anar en compte al fer-los servir.

```

````{r ML names}
#Es simplifica els noms dels models inicials per poder utilitzar-los
més comodament en el codi
M0<-Lake1.step.model0
M0col<-Lake1.step.model0col
M1<-Lake1.step.model1

```

```
M2<-Lake1.step.model2
```

```
M0s<-Lake1.step.model0s  
M0cols<-Lake1.step.model0col.s  
M1s<-Lake1.step.model1s  
M2s<-Lake1.step.model2s  
````
```

```
# AJUST MODEL EFECTES ALEATORIS
```

Pels resultats que s'han anat obtinguent semblaria que es tracta en una possible variant de la simulació ja vista identificada com a He2 amb paral·lelisme i diferències entre individus, tot i que en aquest cas és possible que sigui un model més complex al afegir la heteroscedasticitat que s'ha anat comprovant i els efectes addicionals en el cas del model original.

```
## Intercepció Aleatòria
```

Si s'analitza amb més detall el context en el que es troba el model s'ha de considerar que el conjunt utilitzat és un subgrup dels individus del model original, que a la vegada ja es una subpoblació de la població en general que pugui tenir el VIH. Cal tenir e compte que l'objectiu d'aquest model de predicció és fer inferència en la població general d'afectats de VIH i no només les de la mostra estudiada, i per tal de fer això s'ha de tenir en compte que si es realitzessin més experiments amb altres conjunts d'individus la resposta podria variar. Per poder representar això és necessari afegir el component aleatori que aporta l'efecte individu amb la seva pròpia variància dins el model.

Per fer aquesta actualització del model s'optarà com a les simulacions per tenir en compte 3 models amb diferents plantejaments del càlcul òptim: El model clàssic de mínims quadrats amb l'error aleatori, el model de màxima versemblança ML i el model de la màxima versemblança restringida REML. Així en els següents models es veurà si canvia molt la bondat d'ajust respecte als models amb interaccions afegint els termes de efectes aleatoris i tinguent en compte els 2 sistemes addicionals d'estimació de resultats òptims.

```
### Ajust models
```

Seguint un model semblant a com s'ha procedit en el codi de programació de la simulació es genera una funció per facilitar la obtenció de resultats de diferents conjunts en cas necessari:

```
````{r funció RI}  
Update_LM_RI <- function(LM,RI_var,df) { #Objecte del tipus LM  
  
  fmOLS <- update(LM$call$formula,paste(" ~ . + Error(",RI_var,""))  
  ML_I_RI_sOLS <- summary(ML_I_RI_OLS <- aov(fmOLS, data = df))  
  
  fmMLf <- LM$call$formula  
  fmMLr <- formula(paste("~1|",RI_var))  
  
  ML_I_RI_sMLE <- summary(ML_I_RI_MLE <- lme(fmMLf, random = fmMLr,  
data = df, method="ML"))  
  ML_I_RI_sREML <- summary(ML_I_RI_REML <- lme(fmMLf, random = fmMLr,  
data = df, method="REML"))
```



```

                                1                                <-
list(ML_I_RI_OLS$within,ML_I_RI_sMLE,ML_I_RI_sREML,ML_I_RI_OLS,ML_I_RI
_MLE,ML_I_RI_REML)
  names(l) <- c("OLS I RI (Within)","MLE I RI","REML I RI",
              "OLS I RI base","MLE I RI base","REML I RI base")

  return(l)
}
...

```{r Update RI}
M0_RI<-Update_LM_RI(M0,"Ident",LAKE1)
eMLE_M0_RI<-exactRLRT(M0_RI_MLE<-lme(M0$call$formula, random = ~1|
Ident, data = LAKE1, method="ML"))
eREML_M0_RI<-exactRLRT(M0_RI_REML<-lme(M0$call$formula, random = ~1|
Ident, data = LAKE1, method="REML"))

intM0_RI<-Update_LM_RI(intM0,"Ident",LAKE1)
eMLE_intM0_RI<-exactRLRT(intM0_RI_MLE<-lme(intM0$call$formula, random
= ~1|Ident, data = LAKE1, method="ML"))
eREML_intM0_RI<-exactRLRT(intM0_RI_REML<-lme(intM0$call$formula,
random = ~1|Ident, data = LAKE1, method="REML"))

M0col_RI<-Update_LM_RI(M0col,"Ident",LAKE1)
eMLE_M0col_RI<-exactRLRT(M0col_RI_MLE<-lme(M0col$call$formula, random
= ~1|Ident, data = LAKE1, method="ML"))
eREML_M0col_RI<-exactRLRT(M0col_RI_REML<-lme(M0col$call$formula,
random = ~1|Ident, data = LAKE1, method="REML"))

intM0col_RI<-Update_LM_RI(intM0col,"Ident",LAKE1)
eMLE_intM0col_RI<-exactRLRT(intM0col_RI_MLE<-
lme(intM0col$call$formula, random = ~1|Ident, data = LAKE1,
method="ML"))
eREML_intM0col_RI<-exactRLRT(intM0col_RI_REML<-
lme(intM0col$call$formula, random = ~1|Ident, data = LAKE1,
method="REML"))

M1_RI<-Update_LM_RI(M1,"Ident",LAKE1)
eMLE_M1_RI<-exactRLRT(M1_RI_MLE<-lme(M1$call$formula, random = ~1|
Ident, data = LAKE1, method="ML"))
eREML_M1_RI<-exactRLRT(M1_RI_REML<-lme(M1$call$formula, random = ~1|
Ident, data = LAKE1, method="REML"))

intM1_RI<-Update_LM_RI(intM1,"Ident",LAKE1)
eMLE_intM1_RI<-exactRLRT(intM1_RI_MLE<-lme(intM1$call$formula, random
= ~1|Ident, data = LAKE1, method="ML"))
eREML_intM1_RI<-exactRLRT(intM1_RI_REML<-lme(intM1$call$formula,
random = ~1|Ident, data = LAKE1, method="REML"))

M2_RI<-Update_LM_RI(M2,"Ident",LAKE1)
eMLE_M2_RI<-exactRLRT(M2_RI_MLE<-lme(M2$call$formula, random = ~1|
Ident, data = LAKE1, method="ML"))
eREML_M2_RI<-exactRLRT(M2_RI_REML<-lme(M2$call$formula, random = ~1|
Ident, data = LAKE1, method="REML"))

intM2_RI<-Update_LM_RI(intM2,"Ident",LAKE1)
eMLE_intM2_RI<-exactRLRT(intM2_RI_MLE<-lme(intM2$call$formula, random
= ~1|Ident, data = LAKE1, method="ML"))
eREML_intM2_RI<-exactRLRT(intM2_RI_REML<-lme(intM2$call$formula,
random = ~1|Ident, data = LAKE1, method="REML"))

```

```

NomsModels <- c("M.Base", "M.Base Int.",
               "M.Corr.Col", "M.Corr.Col. Int.",
               "M.LogTrans.", "M.LogTrans.Int.",
               "M.Box-Cox", "M.Box-Cox Int.")

RLRT_resum_RI <- data.frame(Model_Lineal =
paste(rep(NomsModels, each=2), c("MLE", "REML")))

RLRT_list_RI <-
list(eMLE_M0_RI, eREML_M0_RI, eMLE_intM0_RI, eREML_intM0_RI,
      eMLE_M0col_RI, eREML_M0col_RI, eMLE_intM0col_RI, eRE
ML_intM0col_RI,
      eMLE_M1_RI, eREML_M1_RI, eMLE_intM1_RI, eREML_intM1_
RI,
      eMLE_M2_RI, eREML_M2_RI, eMLE_intM2_RI, eREML_intM2_
RI)
RLRT_resum_RI$p-valor <- RLRT"
sapply(1:length(RLRT_list_RI), function(l1){
  if (class(RLRT_list_RI[[l1]])!="hstest"){
    return("-")
  } else {
    round(RLRT_list_RI[[l1]]$p.value, 4)
  }
})

kable(RLRT_resum_RI, booktabs = T) %>% kable_styling(latex_options =
c("striped", "condensed"))

```

Tot i que la significació de l'efecte aleatori RI surt no significatiu a nivell més concret pot ser que signifiquin algun canvi.

### Diagnòstic models

Tot i tenir un resultat de no significació en tots els models amb l'efecte aleatori de l'intercepció dependent de l'individu, es recupera la funció ja utilitzada per evaluar alguns indicadors del model:

```

```{r Resum ML Funció}
ML_Resum <- function(R,Key) {
  ML_table <- data.frame(Sigma = sapply(1:length(R),function(s){
    if (class(R[[s]])[1]=="aov" | class(R[[s]])[1]=="lmm"){
      round(sigma(summary.lm(R[[s]])),4)
    } else {
      round(sigma(R[[s]]),4)
    }
  })
)
ML_table$AIC <- sapply(1:length(R),function(s){
  round(AIC(R[[s]]),4)
})
ML_table$BIC <- sapply(1:length(R),function(s){
  round(BIC(R[[s]]),4)
})

rownames(ML_table) <- paste(Key,names(R),sep="--")

R1 <- ML_table

```

```

        ML_table_p      <-      data.frame(Model      =
rep(rownames(ML_table),ncol(ML_table)))
        ML_table_p$Model <-      factor(ML_table_p$Model,      levels      =
rownames(ML_table))
        ML_table_p$ParQ <- unlist(ML_table)
        ML_table_p$Par <- rep(colnames(ML_table),each=nrow(ML_table))

        fact <- if(mean(abs(ML_table$AIC)) >= mean(abs(ML_table$Sigma)))
{ #Es fa un càlcul del factor més adient per graficar les 2 escales en
el gràfic
        10^(round(log10(abs(mean(ML_table$AIC))/abs(mean(ML_table$Sigma)))
))
} else {
        1/(10^(round(log10(abs(mean(ML_table$Sigma))/abs(mean(ML_table$AIC
))))))
}

        if (fact==0){
                fact <-0.0001
        }

        ML_table_p$ParQ[ML_table_p$Par=="Sigma"]      <-
ML_table_p$ParQ[ML_table_p$Par=="Sigma"]*fact

        R2 <- ggplot(data=ML_table_p, aes(x=Model, y=ParQ, fill=Par)) +
geom_bar(stat="identity", position=position_dodge()+
scale_fill_brewer(palette="Pastel1") + theme_minimal() +
labs(title=Key,y = "AIC and BIC par.") +
scale_y_continuous(expand = c(0, 0),sec.axis = sec_axis(~./fact,
name = "Error.std. residual (Sigma)")) +
theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))

        return(list(R1,R2))
}
...

```{r RI Diag}
M_RI_listA0 <- list(M0,intM0,M0col,intM0col,
                M0_RI[1:3],intM0_RI[1:3],M0col_RI[1:3],intM0col_RI[
1:3])
M_RI_listA <- ConvList(M_RI_listA0)

M_RI_listB0 <- list(M1,intM1,M2,intM2,
                M1_RI[1:3],intM1_RI[1:3],M2_RI[1:3],intM2_RI[1:3])
M_RI_listB <- ConvList(M_RI_listB0)

names(M_RI_listA)[1:4] <- NomsModels[1:4]
names(M_RI_listA)[(4+1):length(M_RI_listA)]<-
paste(rep(NomsModels[1:4],each=3),rep(paste("RI",c("OLS", "MLE", "REML")
),4))
names(M_RI_listB)[1:4] <- NomsModels[5:8]
names(M_RI_listB)[(4+1):length(M_RI_listB)]<-
paste(rep(NomsModels[5:8],each=3),rep(paste("RI",c("OLS", "MLE", "REML")
),4))

(DiagA_RI <- ML_Resum(M_RI_listA,"Lake1"))

BestA <- unique(c(order(DiagA_RI[[1]]$AIC,decreasing=F)[1:4],
                order(DiagA_RI[[1]]$Sigma,decreasing=F)[1:4]))
ML_Resum(M_RI_listA[sort(BestA)],"Lake1") #Comparativa més selectiva

```

```

(DiagB_RI<-ML_Resum(M_RI_listB, "Lake1"))

BestB <- unique(c(order(DiagB_RI[[1]]$AIC,decreasing=F)[1:4],
                      order(DiagB_RI[[1]]$Sigma,decreasing=F)[1:4]))
ML_Resum(M_RI_listB[sort(BestB)],"Lake1") #Comparativa més selectiva

#Deixem només de base els d'interacció i fora el de correcció de
colinealitat també:
(DiagAi_RI <- ML_Resum(M_RI_listA[-c(1,3:7,11:16)],"Lake1"))

BestAi <- unique(c(order(DiagAi_RI[[1]]$AIC,decreasing=F)[1:4],
                      order(DiagAi_RI[[1]]$Sigma,decreasing=F)[1:4]))
ML_Resum((M_RI_listA[-c(1,3:7,11:16)])[sort(BestAi)],"Lake1")
#Comparativa més selectiva

(DiagBi_RI<-ML_Resum(M_RI_listB[-c(1,3,5:7,11:13)],"Lake1"))

BestBi <- unique(c(order(DiagBi_RI[[1]]$AIC,decreasing=F)[1:4],
                      order(DiagBi_RI[[1]]$Sigma,decreasing=F)[1:4]))
ML_Resum((M_RI_listB[-c(1,3,5:7,11:13)])[sort(BestBi)],"Lake1")
#Comparativa més selectiva
```

```

En els primers gràfics s'ha comparat tots els models, per una banda sense transformació i per l'altra amb les transformacions. Posteriorment s'ha fet una selecció dels models amb els indicadors AIC/BIC i la variància residual més baixos per veure quins eren els millors models a tenir en compte:

- + Pels models no transformats no sembla que els models amb l'efecte RI aportin una millora significativa en variància residual o AIC/BIC. Possiblement es pot veure una lleugera millora si s'observa el model d'interacció base amb el model d'interacció RI REML o MLE.
- + El cas del smodels transformats a nivell de selecció només s'han seleccionat models amb la logtransformació i en aquest cas segueix sense apreciar-se una millora significativa respecte al model base amb interaccions.

Es calcula el coeficient de determinació  $R^2$ . En el cas dels models amb efectes aleatoris es calcula a partir del coeficient de determinació condicionat que suma el que aporta els efectes fixos i els efectes aleatoris (en el cas d'aquest models només s'ha trobat fàcilment com calcular-ho en R amb els models calculats per ML i dins d'aquí en els casos amb termes d'interacció es fa una aproximació mitjançant la correlació lineal entre la resposta i els valors ajustats):

```

```{r RI Diag R2}
M_RI_listA_R2 <- list(M0,intM0,M0col,intM0col,M0_RI_MLE,M0_RI_REML,
                    intM0_RI_MLE,intM0_RI_REML,M0col_RI_MLE,M0col_RI_REML,
                    intM0col_RI_MLE,intM0col_RI_REML)

names(M_RI_listA_R2)[1:4] <- NomsModels[1:4]
names(M_RI_listA_R2)[(4+1):length(M_RI_listA_R2)]<-
paste(rep(NomsModels[1:4],each=2),rep(paste("RI",c("MLE","REML")),4))

M_RI_dfa_R2 <- data.frame(Model = names(M_RI_listA_R2))
M_RI_dfa_R2$R2 <- sapply(1:length(M_RI_listA_R2),function(M){

```

```

if (class(M_RI_listA_R2[[M]])[1]=="lmm"){
  summary(M_RI_listA_R2[[M]])$r.squared
} else if (class(M_RI_listA_R2[[M]])=="lme"){
  if(all(paste(attr(M_RI_listA_R2[[M]]$terms, "predvars")[-c(1:2)])
==
  paste(attr(M_RI_listA_R2[[M]]$terms, "term.labels"))){
    rsquared(M_RI_listA_R2[[M]])$Conditional
  } else {
    summary(lm(formula(paste(paste("M_RI_listA_R2[[" ,M, "]]$data$",
                                attr(M_RI_listA_R2[[M]]
$terms, "variables")[[2]], sep=""),
                                "~", "M_RI_listA_R2[[" ,M, "]]$fitted"), sep="")))
    $r.squared
  }
}
})

R2_A <- ggplot(data=M_RI_dfA_R2, aes(x=Model, y=R2, fill=Model)) +
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Set3") + theme_minimal() +
  labs(y = "R2 Coeff.") +
  theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))

M_RI_listB_R2 <- list(M1,intM1,M2,intM2,M1_RI_MLE,
                    M1_RI_REML,intM1_RI_MLE,intM1_RI_REML,
                    M2_RI_MLE,M2_RI_REML,intM2_RI_MLE,intM2_RI_REML)
names(M_RI_listB_R2)[1:4] <- NomsModels[5:8]
names(M_RI_listB_R2)[(4+1):length(M_RI_listB_R2)]<-
paste(rep(NomsModels[5:8], each=2), rep(paste("RI",c("MLE", "REML")), 4))

M_RI_dfB_R2 <- data.frame(Model = names(M_RI_listB_R2))
M_RI_dfB_R2 <- data.frame(Model = names(M_RI_listB_R2))
M_RI_dfB_R2$R2 <- sapply(1:length(M_RI_listB_R2), function(M){
  if (class(M_RI_listB_R2[[M]])[1]=="lmm"){
    summary(M_RI_listB_R2[[M]])$r.squared
  } else if (class(M_RI_listB_R2[[M]])=="lme"){
    if(all(paste(attr(M_RI_listB_R2[[M]]$terms, "predvars")[-c(1:2)])
==
    paste(attr(M_RI_listB_R2[[M]]$terms, "term.labels"))){
      rsquared(M_RI_listB_R2[[M]])$Conditional
    } else {
      summary(lm(formula(paste(paste("M_RI_listB_R2[[" ,M, "]]$data$",
                                attr(M_RI_listB_R2[[M]]
$terms, "variables")[[2]], sep=""),
                                "~", "M_RI_listB_R2[[" ,M, "]]$fitted"), sep="")))
      $r.squared
    }
  }
})

R2_B <- ggplot(data=M_RI_dfB_R2, aes(x=Model, y=R2, fill=Model)) +
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Set3") + theme_minimal() +
  labs(y = "R2 Coeff.") +
  theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))

R2_A
R2_B

BestA_R2 <- order(M_RI_dfA_R2$R2,decreasing=T)[1:4]
BestB_R2 <- order(M_RI_dfB_R2$R2,decreasing=T)[1:4]

```

```

M_RI_df_sel_R2 <- rbind(M_RI_dfA_R2[BestA_R2,],M_RI_dfB_R2[BestB_R2,])

R2_sel <- ggplot(data=M_RI_df_sel_R2, aes(x=Model, y=R2, fill=Model))
+
  geom_bar(stat="identity", position=position_dodge())+
  scale_fill_brewer(palette="Set3") + theme_minimal() +
  labs(y = "R2 Coeff.") +
  theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))

R2_sel
...

```

Com en l'anterior cas s'ha fet dos gràfics amb tots els models i un tercer gràfic on s'ha seleccionat les  $R^2$  més elevades. En aquest cas es pot veure que s'han seleccionat el model base amb interaccions i el Box-Cox transformat amb interaccions que també dona informació de quins models tenen un % d'explicabilitat més elevat de la resposta, però tampoc sembla haver diferències amb els models on s'afegeix l'efecte aleatori RI.

### ## Pendent Aleatòria

Tot i que en el test realitzat d'interacció entre factor i covariant ha donat el resultat del paral·lelisme de pendent entre individus, seria interessant veure com es comporta si s'afegeix en el model una component aleatòria d'aquest paràmetre ja que en el supòsit que el test de interacció fet no fos correcte la interpretació, igual que la resposta dels individus, la degradació del paràmetre observat en front al temps podria no ser igual per tots els individus. Al fer la inferència a partir de la mostra podria ser que es trobés que hi ha una degradació equivalent per tots els individus o al contrari que cada un es comporta diferent. Això donaria peu també a suposar un possible efecte aleatori en la pendent respecte a la covariant temps de cada individu i per tant també amb un altre efecte de variància dins el model, encara que pels resultats obtinguts no s'espera que sigui així.

### ### Ajust models

```

```{r funció RIS}
Update_LM_RIS <- function(LM,RI_var,RIS_var,df) { #Objecte del tipus
LM
  fmOLS <- update(LM$call$formula,paste(" ~ . +
Error(",RI_var,"/",RIS_var,""))
  ML_I_RIS_sOLS <- summary(ML_I_RIS_OLS <- aov(fmOLS, data = df))

  fmMLf <- LM$call$formula
  fmMLr <- formula(paste("~1+",RIS_var,"|",RI_var))

  ML_I_RIS_sMLE <- summary(ML_I_RIS_MLE <- lme(fmMLf, random = fmMLr,
control=ctrl, data = df, method="ML"))
  ML_I_RIS_sREML <- summary(ML_I_RIS_REML <- lme(fmMLf, random =
fmMLr, control=ctrl, data = df, method="REML"))

  l <-
list(ML_I_RIS_OLS$within,ML_I_RIS_sMLE,ML_I_RIS_sREML,ML_I_RIS_OLS,ML_
I_RIS_MLE,ML_I_RIS_REML)
  names(l) <- c("OLS I RIS (Within)","MLE I RIS","REML I RIS",
"OLS I RIS base","MLE I RIS base","REML I RIS base")
}

```

```

    return(l)
}

```

A diferència del cas anterior no és tan fàcil aplicar la prova RLRT al haver dos efectes aleatoris. Es fa la prova ANOVA dels models corresponents ML amb 1 efecte aleatori (intercepció) i 2 (pendent i intercepció):

```

```{r Update RIS, warning=F}
#NO ES PODEN FER ANOVAS DE RESULTATS DE FUNCIONS JA QUE NECESSITA LES
FORMULES ORIGINALS, ES FA INDIVIDUALMENT AMB ELS MODELS AJUSTATS PER
REML

M0_RIS<-Update_LM_RIS(M0,"Ident","Tiempo",LAKE1)

M0_RIS_REML<-lme(M0$call$formula, random = ~1+Tiempo|Ident,
control=ctrl, data = LAKE1, method="REML")
(M0_RI_RIS_anREML <- anova(M0_RI_REML,M0_RIS_REML))

intM0_RIS<-Update_LM_RIS(intM0,"Ident","Tiempo",LAKE1)

intM0_RIS_REML<-lme(intM0$call$formula, random = ~1+Tiempo|Ident,
control=ctrl, data = LAKE1, method="REML")
(intM0_RI_RIS_anREML <- anova(intM0_RI_REML,intM0_RIS_REML))

M0col_RIS<-Update_LM_RIS(M0col,"Ident","Tiempo",LAKE1)

M0col_RIS_REML<-lme(M0col$call$formula, random = ~1+Tiempo|Ident,
control=ctrl, data = LAKE1, method="REML")
(M0col_RI_RIS_anREML <- anova(M0col_RI_REML,M0col_RIS_REML))

intM0col_RIS<-Update_LM_RIS(intM0col,"Ident","Tiempo",LAKE1)

intM0col_RIS_REML<-lme(intM0col$call$formula, random = ~1+Tiempo|
Ident, control=ctrl, data = LAKE1, method="REML")
(intM0col_RI_RIS_anREML <- anova(intM0col_RI_REML,intM0col_RIS_REML))

M1_RIS<-Update_LM_RIS(M1,"Ident","Tiempo",LAKE1)

M1_RIS_REML<-lme(M1$call$formula, random = ~1+Tiempo|Ident,
control=ctrl, data = LAKE1, method="REML")
(M1_RI_RIS_anREML <- anova(M1_RI_REML,M1_RIS_REML))

intM1_RIS<-Update_LM_RIS(intM1,"Ident","Tiempo",LAKE1)

intM1_RIS_REML<-lme(intM1$call$formula, random = ~1+Tiempo|Ident,
control=ctrl, data = LAKE1, method="REML")
(intM1_RI_RIS_anREML <- anova(intM1_RI_REML,intM1_RIS_REML))

M2_RIS<-Update_LM_RIS(M2,"Ident","Tiempo",LAKE1)

M2_RIS_REML<-lme(M2$call$formula, random = ~1+Tiempo|Ident,
control=ctrl, data = LAKE1, method="REML")
(M2_RI_RIS_anREML <- anova(M2_RI_REML,M2_RIS_REML))

intM2_RIS<-Update_LM_RIS(intM2,"Ident","Tiempo",LAKE1)

intM2_RIS_REML<-lme(intM2$call$formula, random = ~1+Tiempo|Ident,
control=ctrl, data = LAKE1, method="REML")

```

```

(intM2_RI_RIS_anREML <- anova(intM2_RI_REML,intM2_RIS_REML))

#Generació dels objectes MLE i comparativa
M0_RIS_MLE<-lme(M0$call$formula, random = ~1+Tiempo|Ident,
control=ctrl, data = LAKE1, method="ML")
(M0_RI_RIS_anMLE <- anova(M0_RI_MLE,M0_RIS_MLE))

intM0_RIS<-Update_LM_RIS(intM0,"Ident","Tiempo",LAKE1)

intM0_RIS_MLE<-lme(intM0$call$formula, random = ~1+Tiempo|Ident,
control=ctrl, data = LAKE1, method="ML")
(intM0_RI_RIS_anMLE <- anova(intM0_RI_MLE,intM0_RIS_MLE))

M0col_RIS<-Update_LM_RIS(M0col,"Ident","Tiempo",LAKE1)

M0col_RIS_MLE<-lme(M0col$call$formula, random = ~1+Tiempo|Ident,
control=ctrl, data = LAKE1, method="ML")
(M0col_RI_RIS_anMLE <- anova(M0col_RI_MLE,M0col_RIS_MLE))

intM0col_RIS<-Update_LM_RIS(intM0col,"Ident","Tiempo",LAKE1)

intM0col_RIS_MLE<-lme(intM0col$call$formula, random = ~1+Tiempo|Ident,
control=ctrl, data = LAKE1, method="ML")
(intM0col_RI_RIS_anMLE <- anova(intM0col_RI_MLE,intM0col_RIS_MLE))

M1_RIS<-Update_LM_RIS(M1,"Ident","Tiempo",LAKE1)

M1_RIS_MLE<-lme(M1$call$formula, random = ~1+Tiempo|Ident,
control=ctrl, data = LAKE1, method="ML")
(M1_RI_RIS_anMLE <- anova(M1_RI_MLE,M1_RIS_MLE))

intM1_RIS<-Update_LM_RIS(intM1,"Ident","Tiempo",LAKE1)

intM1_RIS_MLE<-lme(intM1$call$formula, random = ~1+Tiempo|Ident,
control=ctrl, data = LAKE1, method="ML")
(intM1_RI_RIS_anMLE <- anova(intM1_RI_MLE,intM1_RIS_MLE))

M2_RIS<-Update_LM_RIS(M2,"Ident","Tiempo",LAKE1)

M2_RIS_MLE<-lme(M2$call$formula, random = ~1+Tiempo|Ident,
control=ctrl, data = LAKE1, method="ML")
(M2_RI_RIS_anMLE <- anova(M2_RI_MLE,M2_RIS_MLE))

intM2_RIS<-Update_LM_RIS(intM2,"Ident","Tiempo",LAKE1)

intM2_RIS_MLE<-lme(intM2$call$formula, random = ~1+Tiempo|Ident,
control=ctrl, data = LAKE1, method="ML")
(intM2_RI_RIS_anMLE <- anova(intM2_RI_MLE,intM2_RIS_MLE))
````

```{r RI_RIS an resum}
RI_RIS_an <- data.frame(Model =
paste(rep(NomsModels,2),rep(c("RI_RIS_MLE","RI_RIS_REML"),each=length(
NomsModels))))
RI_RIS_an$Model.base <- rep(NomsModels,2)
RI_RIS_an$ML.method <-
rep(c("RI_RIS_MLE","RI_RIS_REML"),each=length(NomsModels))
RI_RIS_an$`p.valor` <-
c(M0_RI_RIS_anMLE$`p-value`[2],
intM0_RI_RIS_anMLE$`p-value`[2], M0col_RI_RIS_anMLE$`p-value`[2],
intM0col_RI_RIS_anMLE$`p-value`[2],
M1_RI_RIS_anMLE$`p-value`[2], intM1_RI_RIS_anMLE$`p-value`[2],

```



```

intM2_RI_RIS_anMLE$`p-value`[2],
M2_RI_RIS_anMLE$`p-value`[2],
intM0_RI_RIS_anREML$`p-value`[2],
M0_RI_RIS_anREML$`p-value`[2],
M0col_RI_RIS_anREML$`p-value`[2],
intM0col_RI_RIS_anREML$`p-value`[2],
M1_RI_RIS_anREML$`p-value`[2],
intM1_RI_RIS_anREML$`p-value`[2],
M2_RI_RIS_anREML$`p-value`[2],
intM2_RI_RIS_anREML$`p-value`[2])

RI_RIS_an_plot <- ggplot(data=RI_RIS_an, aes(x=Model.base, y=p.valor,
fill=ML.method)) +
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Blues") + theme_minimal() +
  labs(y = "RI/RIS anova p-valor") +
  theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))

RI_RIS_an_plot
````

```

En la majoria de models segueix essent la significació baixa (p-valor alt), però es veu curiosament que hi ha alguns models on es torna o queda proper a la significància, concretament si es pren els que tenen p-valors per sota de 0.25 serien `paste(RI_RIS_an$Model[which(RI_RIS_an$p.valor<0.25)])` amb p-valors de paste(round(RI_RIS_an$p.valor[which(RI_RIS_an$p.valor<0.25)],4))` respectivament.`

### Diagnòstic models

Es tornen a generar els diagnòstics per aquests models amb efecte RIS:

```

```{r RIS Diag}
M_RIS_listA0 <- list(M0,intM0,M0col,intM0col,
M0_RIS[1:3],intM0_RIS[1:3],M0col_RIS[1:3],intM0col_
RIS[1:3])
M_RIS_listA <- ConvList(M_RIS_listA0)

M_RIS_listB0 <- list(M1,intM1,M2,intM2,
M1_RIS[1:3],intM1_RIS[1:3],M2_RIS[1:3],intM2_RIS[1:
3])
M_RIS_listB <- ConvList(M_RIS_listB0)

names(M_RIS_listA)[1:4] <- NomsModels[1:4]
names(M_RIS_listA)[(4+1):length(M_RIS_listA)]<-
paste(rep(NomsModels[1:4],each=3),rep(paste("RIS",c("OLS", "MLE", "REML"
)),4))
names(M_RIS_listB)[1:4] <- NomsModels[5:8]
names(M_RIS_listB)[(4+1):length(M_RIS_listB)]<-
paste(rep(NomsModels[5:8],each=3),rep(paste("RIS",c("OLS", "MLE", "REML"
)),4))

(DiagA_RIS <- ML_Resum(M_RIS_listA, "Lake1"))

BestA2 <- unique(c(order(DiagA_RIS[[1]]$AIC,decreasing=F)[1:4],
order(DiagA_RIS[[1]]$Sigma,decreasing=F)[1:4]))
ML_Resum(M_RIS_listA[sort(BestA2)],"Lake1") #Comparativa més selectiva

(DiagB_RIS<-ML_Resum(M_RIS_listB, "Lake1"))

BestB2 <- unique(c(order(DiagB_RIS[[1]]$AIC,decreasing=F)[1:4],
order(DiagB_RIS[[1]]$Sigma,decreasing=F)[1:4]))

```

```

ML_Resum(M_RIS_listB[sort(BestB2)],"Lake1") #Comparativa més selectiva

#Deixem només de base els d'interacció i fora el de correcció de
colinealitat també:
(DiagAi_RIS <- ML_Resum(M_RIS_listA[-c(1,3:7,11:16)],"Lake1"))

BestAi <- unique(c(order(DiagAi_RIS[[1]]$AIC,decreasing=F)[1:4],
  order(DiagAi_RIS[[1]]$Sigma,decreasing=F)[1:4]))
ML_Resum((M_RIS_listA[-c(1,3:7,11:16)])[sort(BestAi)],"Lake1")
#Comparativa més selectiva

(DiagBi_RIS<-ML_Resum(M_RIS_listB[-c(1,3,5:7,11:13)],"Lake1"))

BestBi <- unique(c(order(DiagBi_RIS[[1]]$AIC,decreasing=F)[1:4],
  order(DiagBi_RIS[[1]]$Sigma,decreasing=F)[1:4]))
ML_Resum((M_RIS_listB[-c(1,3,5:7,11:13)])[sort(BestBi)],"Lake1")
#Comparativa més selectiva
` ``

```

Pels models seleccionats amb una variància residual més baixa així com els indicadors AIC/BIC més baixos curiosament es pot veure que pels 2 models que havien donat una significació alta el fet d'afegir l'efecte aleatori RIS, aparentment es nota d'una manera més efectiva la millora en aquests paràmetres al comparar-los amb la referència base. No és una millora espectacular, però podria ser suficient per fer prediccions més precises.

Es calcula com abans el coeficient de determinació  $R^2$ . En el cas dels models amb efectes aleatoris es calcula a partir del coeficient de determinació condicionat que suma el que aporta els efectes fixos i els efectes aleatoris (en el cas d'aquest models només s'ha trobat fàcilment com calcular-ho en R amb els models calculats per ML i dins d'aquí en els casos amb termes d'interacció es fa una aproximació mitjançant la correlació lineal entre la resposta i els valors ajustats):

```

` ``{r RIS Diag R2}
M_RIS_listA_R2 <- list(M0,intM0,M0col,intM0col,M0_RIS_MLE,M0_RIS_REML,
  intM0_RIS_MLE,intM0_RIS_REML,M0col_RIS_MLE,M0col
  l_RIS_REML,
  intM0col_RIS_MLE,intM0col_RIS_REML)

names(M_RIS_listA_R2)[1:4] <- NomsModels[1:4]
names(M_RIS_listA_R2)[(4+1):length(M_RIS_listA_R2)]<-
paste(rep(NomsModels[1:4],each=2),rep(paste("RIS",c("MLE","REML")),4))

M_RIS_dfA_R2 <- data.frame(Model = names(M_RIS_listA_R2))
M_RIS_dfA_R2$R2 <- sapply(1:length(M_RIS_listA_R2),function(M){
  if (class(M_RIS_listA_R2[[M]])[1]=="lmm"){
    summary(M_RIS_listA_R2[[M]])$r.squared
  } else if (class(M_RIS_listA_R2[[M]])=="lme"){
    if(all(paste(attr(M_RIS_listA_R2[[M]]$terms,"predvars")[-c(1:2)])
    ==
    paste(attr(M_RIS_listA_R2[[M]]$terms,"term.labels"))){
      rsquared(M_RIS_listA_R2[[M]])$Conditional
    } else {
      summary(lm(formula(paste(paste("M_RIS_listA_R2[[" ,M, "]]$data$",
        attr(M_RIS_listA_R2[[M]]
$terms,"variables")[[2]],sep=""),
        "~", "M_RIS_listA_R2[[" ,M, "]]$fitted"),sep="")))
    $r.squared

```

```

    }
  })
})

R2_A2 <- ggplot(data=M_RIS_dfA_R2, aes(x=Model, y=R2, fill=Model)) +
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Set3") + theme_minimal() +
  labs(y = "R2 Coeff.") +
  theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))

M_RIS_listB_R2 <- list(M1,intM1,M2,intM2,M1_RIS_MLE,
  M1_RIS_REML,intM1_RIS_MLE,intM1_RIS_REML,
  M2_RIS_MLE,M2_RIS_REML,intM2_RIS_MLE,intM2_RIS_REM
L)
names(M_RIS_listB_R2)[1:4] <- NomsModels[5:8]
names(M_RIS_listB_R2)[(4+1):length(M_RIS_listB_R2)]<-
paste(rep(NomsModels[5:8],each=2),rep(paste("RIS",c("MLE","REML")),4))

M_RIS_dfB_R2 <- data.frame(Model = names(M_RIS_listB_R2))
M_RIS_dfB_R2 <- data.frame(Model = names(M_RIS_listB_R2))
M_RIS_dfB_R2$R2 <- sapply(1:length(M_RIS_listB_R2),function(M){
  if (class(M_RIS_listB_R2[[M]])[1]=="lmm"){
    summary(M_RIS_listB_R2[[M]])$r.squared
  } else if (class(M_RIS_listB_R2[[M]])=="lme"){
    if(all(paste(attr(M_RIS_listB_R2[[M]]$terms,"predvars")[-c(1:2)])
==
      paste(attr(M_RIS_listB_R2[[M]]$terms,"term.labels"))){
      rsquared(M_RIS_listB_R2[[M]])$Conditional
    } else {
      summary(lm(formula(paste(paste("M_RIS_listB_R2[[" ,M, "]]$data$",
      attr(M_RIS_listB_R2[[M]]
$terms,"variables")[[2]],sep=""),
      "~", "M_RIS_listB_R2[[" ,M, "]]$fitted"),sep="")))
    }
  }
})

R2_B2 <- ggplot(data=M_RIS_dfB_R2, aes(x=Model, y=R2, fill=Model)) +
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Set3") + theme_minimal() +
  labs(y = "R2 Coeff.") +
  theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))

R2_A2
R2_B2

BestA2_R2 <- order(M_RIS_dfA_R2$R2,decreasing=T)[1:4]
BestB2_R2 <- order(M_RIS_dfB_R2$R2,decreasing=T)[1:4]
M_RIS_df_sel_R2
rbind(M_RIS_dfA_R2[BestA2_R2,],M_RIS_dfB_R2[BestB2_R2,])

R2_sel2 <- ggplot(data=M_RIS_df_sel_R2, aes(x=Model, y=R2,
fill=Model)) +
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Set3") + theme_minimal() +
  labs(y = "R2 Coeff.") +
  theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))

R2_sel2

```

```

#Deixem només els models amb interacció
M_RIS_dfA_R2i <- M_RIS_dfA_R2[-c(1,3:6,9:12),]
R2_A2i <- ggplot(data=M_RIS_dfA_R2i, aes(x=Model, y=R2, fill=Model)) +
  geom_bar(stat="identity", position=position_dodge())+
  scale_fill_brewer(palette="Set3") + theme_minimal() +
  labs(y = "R2 Coeff.") +
  theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))

M_RIS_dfB_R2i <- M_RIS_dfB_R2[-c(1,3,5:6,9:10),]
R2_B2i <- ggplot(data=M_RIS_dfB_R2i, aes(x=Model, y=R2, fill=Model)) +
  geom_bar(stat="identity", position=position_dodge())+
  scale_fill_brewer(palette="Set3") + theme_minimal() +
  labs(y = "R2 Coeff.") +
  theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))

R2_A2i
R2_B2i

M_RIS_df_sel_R2i <- rbind(M_RIS_dfA_R2i,M_RIS_dfB_R2i)

R2_sel2i <- ggplot(data=M_RIS_df_sel_R2i, aes(x=Model, y=R2,
fill=Model)) +
  geom_bar(stat="identity", position=position_dodge())+
  scale_fill_brewer(palette="Set3") + theme_minimal() +
  labs(y = "R2 Coeff.") +
  theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))

R2_sel2i
...

```

Sí que s'observa aquesta vegada una millora més aparent en el coeficient  $R^2$ , concretament si es pren els 4 models amb un coeficient més alt serien `paste(M_RIS_df_sel_R2$Model[order(M_RIS_df_sel_R2$R2,decreasing =T)[1:4])` amb coeficients de paste(round(M_RIS_df_sel_R2$R2[order(M_RIS_df_sel_R2$R2,decreasing =T)[1:4],4))` respectivament. Curiosament segueixen essent els que denotaven millora amb l'efecte aleatori.`

Per els següents apartats de codi es fa una selecció dels models en base a les proves que s'han anat fent en aquest últim pas pel que es tindrà en compte els models següents:

```

+ Model base amb interaccions i efectes RIS ajustat per MLE i REML.
+ Model LogTransformat amb interaccions i efectes RIS ajustat per MLE i REML.
+ Model Box-Cox amb interaccions i efectes RIS ajustat per MLE i REML.

```{r Filtrat models}
#Els models que s'utilitzaran per continuar amb l'anàlisi son
Models.sel <- list(intM0_RIS_MLE,intM0_RIS_REML,
                  intM1_RIS_MLE,intM1_RIS_REML,
                  intM2_RIS_MLE,intM2_RIS_REML)

names(Models.sel) <- c("M.Int.RIS.MLE","M.Int.RIS.REML",
                      "M.Logt.Int.RIS.MLE","M.Logt.Int.RIS.REML",
                      "M.BCt.Int.RIS.MLE","M.BCt.Int.RIS.REML")
...

```

En aquest apartat on es denota més efecte per l'efecte aleatori RIS també és interessant veure si ha canviat gaire el diagnòstic clàssic del model lineal. Es fa en els models seleccionats únicament:

```
```{r Diag Model simple Lake1 RIS Sel}
Lake1IntRIS_MLE.SM0.sdres <- residuals(intM0_RIS_MLE,type = "pearson")
Lake1IntRIS_REML.SM0.sdres <- residuals(intM0_RIS_REML,type =
"pearson")
par(mfrow=c(1,3))

qqnorm(Lake1Int.SM0.sdres,
      ylab="Standardized Res.",
      xlab="Normal Scores",
      main="M. Original Int. - Standardized Res.")
qqline(Lake1Int.SM0.sdres)

qqnorm(Lake1IntRIS_MLE.SM0.sdres,
      ylab="Standardized Res.",
      xlab="Normal Scores",
      main="M. Original Int RIS MLE - Standardized Res.")
qqline(Lake1Int.SM0.sdres)

qqnorm(Lake1IntRIS_REML.SM0.sdres,
      ylab="Standardized Res.",
      xlab="Normal Scores",
      main="M. Original Int RIS REML - Standardized Res.")
qqline(Lake1Int.SM0.sdres)

S0Int
(S0IntRIS_MLE <- shapiro.test(intM0_RIS_MLE$residuals))
(S0IntRIS_REML <- shapiro.test(intM0_RIS_REML$residuals))

Lake1IntRIS_MLE.SM1.sdres <- residuals(intM1_RIS_MLE,type = "pearson")
Lake1IntRIS_REML.SM1.sdres <- residuals(intM1_RIS_REML,type =
"pearson")
par(mfrow=c(1,3))

qqnorm(Lake1Int.SM1.sdres,
      ylab="Standardized Res.",
      xlab="Normal Scores",
      main="M. Logt Int. - Standardized Res.")
qqline(Lake1Int.SM1.sdres)

qqnorm(Lake1IntRIS_MLE.SM1.sdres,
      ylab="Standardized Res.",
      xlab="Normal Scores",
      main="M. Logt Int RIS MLE - Standardized Res.")
qqline(Lake1Int.SM1.sdres)

qqnorm(Lake1IntRIS_REML.SM1.sdres,
      ylab="Standardized Res.",
      xlab="Normal Scores",
      main="M. Logt Int RIS REML - Standardized Res.")
qqline(Lake1Int.SM1.sdres)

S1Int
(S1IntRIS_MLE <- shapiro.test(intM1_RIS_MLE$residuals))
(S1IntRIS_REML <- shapiro.test(intM1_RIS_REML$residuals))

Lake1IntRIS_MLE.SM2.sdres <- residuals(intM2_RIS_MLE,type = "pearson")
```

```

Lake1IntRIS_REML.SM2.sdres <- residuals(intM2_RIS_REML,type =
"pearson")
par(mfrow=c(1,3))

qqnorm(Lake1Int.SM2.sdres,
       ylab="Standardized Res.",
       xlab="Normal Scores",
       main="M. Bct Int. - Standardized Res.")
qqline(Lake1Int.SM2.sdres)

qqnorm(Lake1IntRIS_MLE.SM2.sdres,
       ylab="Standardized Res.",
       xlab="Normal Scores",
       main="M. Bct Int RIS MLE - Standardized Res.")
qqline(Lake1Int.SM2.sdres)

qqnorm(Lake1IntRIS_REML.SM2.sdres,
       ylab="Standardized Res.",
       xlab="Normal Scores",
       main="M. Bct Int RIS REML - Standardized Res.")
qqline(Lake1Int.SM2.sdres)

S2Int
(S2IntRIS_MLE <- shapiro.test(intM2_RIS_MLE$residuals))
(S2IntRIS_REML <- shapiro.test(intM2_RIS_REML$residuals))
````

```

El que es veu en el test Shapiro no seria bo ja que sembla que els models s'allunyen de la normalitat al afegir els efectes aleatoris RIS, tot i així visualment es pot veure que hi ha poques diferències en el QQplot, pel que potser seria bo no prendre's aquest test de Shapiro com definitiu (en cas que es volgués aprofundir més es podria intentar fer més proves de normalitat).

```

````{r Diag H Model simple Lake1 RIS INT}
pHIntrIS_MLE01 <- ggplot(LAKE1, aes(x=fitted(intM0_RIS_MLE),
y=residuals(intM0_RIS_MLE),
                                shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (IntrIS_MLE.Homocedasticity test)",
       x="Fitted", y = "Residuals")

pHIntrIS_MLE02 <-ggplot(LAKE1, aes(x=fitted(intM0_RIS_MLE),
y=sqrt(abs(residuals(intM0_RIS_MLE))),
                                shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (IntrIS_MLE.Homocedasticity test)",
       x="Fitted", y = expression(sqrt("|residuals|")))

pHIntrIS_REML01 <- ggplot(LAKE1, aes(x=fitted(intM0_RIS_REML),
y=residuals(intM0_RIS_REML),
                                shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (IntrIS_REML.Homocedasticity test)",
       x="Fitted", y = "Residuals")

pHIntrIS_REML02 <-ggplot(LAKE1, aes(x=fitted(intM0_RIS_REML),
y=sqrt(abs(residuals(intM0_RIS_REML))),
                                shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (IntrIS_REML.Homocedasticity test)",
       x="Fitted", y = expression(sqrt("|residuals|")))

```

```

                                shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (IntrIS_REML.Homocedasticity test)",
        x="Fitted", y = expression(sqrt("|residuals|")))

grid.arrange(pHInt01, pHIntrIS_MLE01,pHIntrIS_REML01,nrow=3)
grid.arrange(pHInt02, pHIntrIS_MLE02,pHIntrIS_REML02,nrow=3)

plot(intM0,which=1)
plot(intM0_RIS_MLE,which=1)
plot(intM0_RIS_REML,which=1)

pHIntrIS_MLE11 <- ggplot(LAKE1, aes(x=fitted(intM1_RIS_MLE),
y=residuals(intM1_RIS_MLE),
                                shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Logt.IntrIS_MLE.Homocedasticity
test)",
        x="Fitted", y = "Residuals")

pHIntrIS_MLE12 <-ggplot(LAKE1, aes(x=fitted(intM1_RIS_MLE),
y=sqrt(abs(residuals(intM1_RIS_MLE))),
                                shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Logt.IntrIS_MLE.Homocedasticity
test)",
        x="Fitted", y = expression(sqrt("|residuals|")))

pHIntrIS_REML11 <- ggplot(LAKE1, aes(x=fitted(intM1_RIS_REML),
y=residuals(intM1_RIS_REML),
                                shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Logt.IntrIS_REML.Homocedasticity
test)",
        x="Fitted", y = "Residuals")

pHIntrIS_REML12 <-ggplot(LAKE1, aes(x=fitted(intM1_RIS_REML),
y=sqrt(abs(residuals(intM1_RIS_REML))),
                                shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Logt.IntrIS_REML.Homocedasticity
test)",
        x="Fitted", y = expression(sqrt("|residuals|")))

grid.arrange(pHInt11, pHIntrIS_MLE11,pHIntrIS_REML11,nrow=3)
grid.arrange(pHInt12, pHIntrIS_MLE12,pHIntrIS_REML12,nrow=3)

plot(intM1,which=1)
plot(intM1_RIS_MLE,which=1)
plot(intM1_RIS_REML,which=1)

pHIntrIS_MLE21 <- ggplot(LAKE1, aes(x=fitted(intM2_RIS_MLE),
y=residuals(intM2_RIS_MLE),
                                shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +

```

```

    labs(title="Fitted vs Residuals (Bct.IntRIS_MLE.Homocedasticity
test)",
    x="Fitted", y = "Residuals")

pHIntRIS_MLE22 <-ggplot(LAKE1, aes(x=fitted(intM2_RIS_MLE),
y=sqrt(abs(residuals(intM2_RIS_MLE))),
shape=Grupo, color=Grupo)) +
geom_jitter(position=position_jitter(1), cex=1.5) +
geom_hline(yintercept=0, linetype="dashed", color = "black") +
labs(title="Fitted vs Residuals (Bct.IntRIS_MLE.Homocedasticity
test)",
x="Fitted", y = expression(sqrt("|residuals|")))

pHIntRIS_REML21 <- ggplot(LAKE1, aes(x=fitted(intM2_RIS_REML),
y=residuals(intM2_RIS_REML),
shape=Grupo, color=Grupo)) +
geom_jitter(position=position_jitter(1), cex=1.5) +
geom_hline(yintercept=0, linetype="dashed", color = "black") +
labs(title="Fitted vs Residuals (Bct.IntRIS_REML.Homocedasticity
test)",
x="Fitted", y = "Residuals")

pHIntRIS_REML22 <-ggplot(LAKE1, aes(x=fitted(intM2_RIS_REML),
y=sqrt(abs(residuals(intM2_RIS_REML))),
shape=Grupo, color=Grupo)) +
geom_jitter(position=position_jitter(1), cex=1.5) +
geom_hline(yintercept=0, linetype="dashed", color = "black") +
labs(title="Fitted vs Residuals (Bct.IntRIS_REML.Homocedasticity
test)",
x="Fitted", y = expression(sqrt("|residuals|")))

grid.arrange(pHInt21, pHIntRIS_MLE21,pHIntRIS_REML21,nrow=3)
grid.arrange(pHInt22, pHIntRIS_MLE22,pHIntRIS_REML22,nrow=3)

plot(intM2,which=1)
plot(intM2_RIS_MLE,which=1)
plot(intM2_RIS_REML,which=1)
````

```

Tampoc es mostra una diferència aparentment significativa en l'homocedasticitat que ja tenien els models de referència. Sembla que segueix fent falta algun tipus de modulació de la variància excepte potser en algun dels models transformats.

```
# AJUST FUNCIONS DE COVARIÀNCIA
```

```
#### Ajust models
```

Es recupera una de les funcions generades en altres simulacions per obtenir els models amb funcions de variància amb algunes modificacions per adaptar-la a l'objectiu d'aquest codi:

```

````{r ML weights funció,error=TRUE}
Comp_W_LM <- function(fLME,WVI,WVE,WVP) # WVI;WVE;WVP Per decidir
quines funcions s'apliquen essent-> 0:No aplica, 1:Aplicar MLE,
2:Aplicar REML, 3:Aplicar les 2
{

baseML <- update(fLME, control=ctrl, method="ML")
baseREML <- update(fLME, control=ctrl, method="REML")
l <- list()

```



```

fMLk <- "IS"
fREk <- "RIS"

  if(WVI==1 | WVI==3){
    ML_I_RI_WVI_sMLE <- summary(ML_I_RI_WVI_MLE <- update(baseML,
weights = varIdent(form = ~Tiempo)))
    le <- length(l)+1
    l[[le]] <- ML_I_RI_WVI_sMLE
    names(l)[le] <- paste(fMLk,"MLE ",fREk,"VarIdent")
    le <- length(l)+1
    l[[le]] <- ML_I_RI_WVI_MLE
    names(l)[le] <- paste(fMLk,"MLE ",fREk,"VarIdent Base")
  } else {
    le <- length(l)+1
    l[[le]] <- "Pendent ML_I_RI_WVI_sMLE"
    names(l)[le] <- paste("NE Model",fMLk,"MLE",fREk,"VarIdent")
    le <- length(l)+1
    l[[le]] <- "Pendent ML_I_RI_WVI_MLE"
    names(l)[le] <- paste("NE Model",fMLk,"MLE",fREk,"VarIdent
Base")
  }
  if(WVI==2 | WVI==3){
    ML_I_RI_WVI_sREML <- summary(ML_I_RI_WVI_REML <-
update(baseREML, weights = varIdent(form = ~Tiempo)))
    le <- length(l)+1
    l[[le]] <- ML_I_RI_WVI_sREML
    names(l)[le] <- paste(fMLk,"REML ",fREk,"VarIdent")
    le <- length(l)+1
    l[[le]] <- ML_I_RI_WVI_REML
    names(l)[le] <- paste(fMLk,"REML ",fREk,"VarIdent Base")
  } else {
    le <- length(l)+1
    l[[le]] <- "Pendent ML_I_RI_WVI_sREML"
    names(l)[le] <- paste("NE Model",fMLk,"REML",fREk,"VarIdent")
    le <- length(l)+1
    l[[le]] <- "Pendent ML_I_RI_WVI_REML"
    names(l)[le] <- paste("NE Model",fMLk,"REML",fREk,"VarIdent
Base")
  }
}

  if(WVE==1 | WVE==3){
    ML_I_RI_WVE_sMLE <- summary(ML_I_RI_WVE_MLE <- update(baseML,
weights = varExp(form = ~Tiempo)))
    le <- length(l)+1
    l[[le]] <- ML_I_RI_WVE_sMLE
    names(l)[le] <- paste(fMLk,"MLE ",fREk,"VarExp")
    le <- length(l)+1
    l[[le]] <- ML_I_RI_WVE_MLE
    names(l)[le] <- paste(fMLk,"MLE ",fREk,"VarExp Base")
  } else{
    le <- length(l)+1
    l[[le]] <- "Pendent ML_I_RI_WVE_sMLE"
    names(l)[le] <- paste("NE Model",fMLk,"MLE",fREk,"VarExp")
    le <- length(l)+1
    l[[le]] <- "Pendent ML_I_RI_WVE_MLE"
    names(l)[le] <- paste("NE Model",fMLk,"MLE",fREk,"VarExp Base")
  }
  if(WVE==2 | WVE==3){
    ML_I_RI_WVE_sREML <- summary(ML_I_RI_WVE_REML <-
update(baseREML, weights = varExp(form = ~Tiempo)))

```

```

le <- length(l)+1
l[[le]] <- ML_I_RI_WVE_sREML
names(l)[le] <- paste(fMLk, "REML ", fREk, "VarExp")
le <- length(l)+1
l[[le]] <- ML_I_RI_WVE_REML
names(l)[le] <- paste(fMLk, "REML ", fREk, "VarExp Base")
} else{
le <- length(l)+1
l[[le]] <- "Pendent ML_I_RI_WVE_sREML"
names(l)[le] <- paste("NE Model", fMLk, "REML", fREk, "VarExp")
le <- length(l)+1
l[[le]] <- "Pendent ML_I_RI_WVE_REML"
names(l)[le] <- paste("NE Model", fMLk, "REML", fREk, "VarExp Base")
}
}

```

#La funció varPower s'ha hagut de deixar sense formules ja que al fer-la dependre de la covariable temps no convergia a un resultat real en els algoritmes de nlme pels conjunts donats

```

if(WVP==1 | WVP==3){
  ML_I_RI_WVP_sMLE <- summary(ML_I_RI_WVP_MLE <- update(baseML,
weights = varPower()))
  le <- length(l)+1
  l[[le]] <- ML_I_RI_WVP_sMLE
  names(l)[le] <- paste(fMLk, "MLE ", fREk, "VarPower")
  le <- length(l)+1
  l[[le]] <- ML_I_RI_WVP_MLE
  names(l)[le] <- paste(fMLk, "MLE ", fREk, "VarPower Base")
} else{
le <- length(l)+1
l[[le]] <- "Pendent ML_I_RI_WVP_sMLE"
names(l)[le] <- paste("NE Model", fMLk, "MLE", fREk, "VarPower")
le <- length(l)+1
l[[le]] <- "Pendent ML_I_RI_WVP_MLE"
names(l)[le] <- paste("NE Model", fMLk, "MLE", fREk, "VarPower
Base")
}
if(WVP==2 | WVP==3){
  ML_I_RI_WVP_sREML <- summary(ML_I_RI_WVP_REML <-
update(baseREML, weights = varPower()))
  le <- length(l)+1
  l[[le]] <- ML_I_RI_WVP_sREML
  names(l)[le] <- paste(fMLk, "REML ", fREk, "VarPower")
  le <- length(l)+1
  l[[le]] <- ML_I_RI_WVP_REML
  names(l)[le] <- paste(fMLk, "REML ", fREk, "VarPower Base")
} else{
le <- length(l)+1
l[[le]] <- "Pendent ML_I_RI_WVP_sREML"
names(l)[le] <- paste("NE Model", fMLk, "REML", fREk, "VarPower")
le <- length(l)+1
l[[le]] <- "Pendent ML_I_RI_WVP_REML"
names(l)[le] <- paste("NE Model", fMLk, "REML", fREk, "VarPower
Base")
}
return(l)
}
...

```

Com en les simulacions es va trobar que no sempre convergien en una resposta es fa algunes proves per veure en quins models podem ajustar aquestes funcions:

```

```{r ML I RIS W,error=TRUE,message=F}
#Models.sel <- list(intM0_RIS_MLE,intM0_RIS_REML,
intM1_RIS_MLE,intM1_RIS_REML, intM2_RIS_MLE,intM2_RIS_REML)

intM0_RIS_W <- Comp_W_LM(intM0_RIS_MLE,3,3,0)
intM1_RIS_W <- Comp_W_LM(intM1_RIS_MLE,3,3,3)
intM2_RIS_W <- Comp_W_LM(intM2_RIS_MLE,3,3,2)
```

#### Diagnòstic models

```{r Resum ML}
intM0_RIS_Ws0 <- list(Models.sel[c(1:2)],intM0_RIS_W[c(1,3,5,7)])
intM0_RIS_Ws <- ConvList(intM0_RIS_Ws0)
(intM0_QI<-ML_Resum(intM0_RIS_Ws,"CargaV.)) #Es guarda la taula per
us posterior
ML_Resum(intM0_RIS_Ws[1:4], "CargaV.")[[2]]

intM1_RIS_Ws0 <- list(Models.sel[c(3:4)],intM1_RIS_W[c(1,3,5,7,9,11)])
intM1_RIS_Ws <- ConvList(intM1_RIS_Ws0)
(intM1_QI <- ML_Resum(intM1_RIS_Ws,"LogCargaV.))
ML_Resum(intM1_RIS_Ws[-c(5:6)], "LogCargaV.")[[2]]

intM2_RIS_Ws0 <- list(Models.sel[c(1:2)],intM2_RIS_W[c(1,3,5,7,11)])
intM2_RIS_Ws <- ConvList(intM2_RIS_Ws0)
(intM2_QI <- ML_Resum(intM2_RIS_Ws,"BCTCargaV.))
ML_Resum(intM2_RIS_Ws[-c(1:2)], "BCTCargaV.")[[2]]
```

```

En aquests primers diagnòstics semblaria que la funció varPower és la que té més efectivitat al modular la variància d'una manera molt significativa. S'ha pogut comprobar en els dos models transformats, però no en el model base on no ha convergit en una solució real i on les altres funcions de modulació de variància no semblen aportar cap millora al model.

Es calcula com abans el coeficient de determinació  $R^2$  amb els mateixos criteris aproximats de càlcul:

```

```{r RIS W Diag R2}
M0_RIS_W_list_R20 <- list(Models.sel[c(1:2)],intM0_RIS_W[c(2,4,6,8)])
M0_RIS_W_list_R2 <- ConvList(M0_RIS_W_list_R20)

M0_RIS_W_dfA_R2 <- data.frame(Model =
paste("M.Int",str_remove(str_remove(str_remove(names(M0_RIS_W_list_R2)
, "Model"),"Base"),"RE: "))

M0_RIS_W_dfA_R2$R2 <- sapply(1:length(M0_RIS_W_list_R2),function(M){
  if (class(M0_RIS_W_list_R2[[M]])[1]=="lmm"){
    summary(M0_RIS_W_list_R2[[M]])$r.squared
  } else if (class(M0_RIS_W_list_R2[[M]])=="lme"){
    if(all(paste(attr(M0_RIS_W_list_R2[[M]]$terms,"predvars")[-
c(1:2)]) ==
paste(attr(M0_RIS_W_list_R2[[M]]$terms,"term.labels"))){
      rsquared(M0_RIS_W_list_R2[[M]])$Conditional

```

```

    } else {
      summary(lm(formula(paste(paste("M0_RIS_W_list_R2[[" , M, "]"
$terms, "variables")[[2]], sep=""),
      attr(M0_RIS_W_list_R2[[M]]
      "~", "M0_RIS_W_list_R2[[" , M, "]"
      $fitted"), sep="")))$r.squared
    }
  }
})

R2_M0W <- ggplot(data=M0_RIS_W_dfA_R2, aes(x=Model, y=R2, fill=Model))
+
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Set3") + theme_minimal() +
  labs(y = "R2 Coeff.") +
  theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))
R2_M0W

M1_RIS_W_list_R20 <-
list(Models.sel[c(1:2)],intM1_RIS_W[c(2,4,6,8,10,12)])
M1_RIS_W_list_R2 <- ConvList(M1_RIS_W_list_R20)

M1_RIS_W_dfA_R2 <- data.frame(Model =
paste("Log.M.Int",str_remove(str_remove(str_remove(names(M1_RIS_W_list
_R2), "Model"),"Base"),"RE: ")))

M1_RIS_W_dfA_R2$R2 <- sapply(1:length(M1_RIS_W_list_R2),function(M){
  if (class(M1_RIS_W_list_R2[[M]])[1]=="lmm"){
    summary(M1_RIS_W_list_R2[[M]])$r.squared
  } else if (class(M1_RIS_W_list_R2[[M]])=="lme"){
    if(all(paste(attr(M1_RIS_W_list_R2[[M]]$terms,"predvars")[-
c(1:2)]) ==
    paste(attr(M1_RIS_W_list_R2[[M]]$terms,"term.labels"))){
      rsquared(M1_RIS_W_list_R2[[M]])$Conditional
    } else {
      summary(lm(formula(paste(paste("M1_RIS_W_list_R2[[" , M, "]"
$terms, "variables")[[2]], sep=""),
      attr(M1_RIS_W_list_R2[[M]]
      "~", "M1_RIS_W_list_R2[[" , M, "]"
      $fitted"), sep="")))$r.squared
    }
  }
})

R2_M1W <- ggplot(data=M1_RIS_W_dfA_R2, aes(x=Model, y=R2, fill=Model))
+
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Set3") + theme_minimal() +
  labs(y = "R2 Coeff.") +
  theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))
R2_M1W

M2_RIS_W_list_R20 <-
list(Models.sel[c(1:2)],intM2_RIS_W[c(2,4,6,8,12)])
M2_RIS_W_list_R2 <- ConvList(M2_RIS_W_list_R20)

M2_RIS_W_dfA_R2 <- data.frame(Model =
paste("Bct.M.Int",str_remove(str_remove(str_remove(names(M2_RIS_W_list
_R2), "Model"),"Base"),"RE: ")))

```

```

M2_RIS_W_dfA_R2$R2 <- sapply(1:length(M2_RIS_W_list_R2),function(M){
  if (class(M2_RIS_W_list_R2[[M]])[1]=="lmm"){
    summary(M2_RIS_W_list_R2[[M]])$r.squared
  } else if (class(M2_RIS_W_list_R2[[M]])=="lme"){
    if(all(paste(attr(M2_RIS_W_list_R2[[M]]$terms,"predvars")[-
c(1:2)]) ==
    paste(attr(M2_RIS_W_list_R2[[M]]$terms,"term.labels"))){
      rsquared(M2_RIS_W_list_R2[[M]])$Conditional
    } else {
      summary(lm(formula(paste(paste("M2_RIS_W_list_R2[["M,"]]
$data$",
                                attr(M2_RIS_W_list_R2[[M]]
$terms,"variables")[[2]],sep=""),
                                "~", "M2_RIS_W_list_R2[["M,"]]
$fitted"),sep=""))$r.squared
    }
  }
})

R2_M2W <- ggplot(data=M2_RIS_W_dfA_R2, aes(x=Model, y=R2, fill=Model))
+
  geom_bar(stat="identity", position=position_dodge()+
scale_fill_brewer(palette="Set3") + theme_minimal() +
labs(y = "R2 Coeff.") +
theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))
R2_M2W
```

```

En aquesta ocasió no sembla millorar el model des del punt de vista del coeficient de determinació.

```

```{r Diag Model simple Lake1 RIS W Sel}
M0_RIS_W_list_base <- M0_RIS_W_list_R2

names(M0_RIS_W_list_base)
str_remove(str_remove(str_remove(names(M0_RIS_W_list_base),
"Model"),"Base"),"RE: ")
#M0_RIS_W_list_R2
par(mfrow=c(1,2))
QQ_M0_list <- lapply(1:length(M0_RIS_W_list_base),function(QQ){
  IntrIS_MLE_W.SM0.sdres <- residuals(M0_RIS_W_list_base[[QQ]],type =
"pearson")
  qqnorm(IntrIS_MLE_W.SM0.sdres,
        ylab="Standardized Res.",
        xlab="Normal Scores",
        main=paste("M0",names(M0_RIS_W_list_base[QQ]),"Standard.
Res.))
  qqline(IntrIS_MLE_W.SM0.sdres)
})

Shap_M0_df <- data.frame(Model =
paste("M2",names(M0_RIS_W_list_base)))
Shap_M0_df$Shapiro_p.valor <- Shap_M0_v <-
sapply(1:length(M0_RIS_W_list_base),function(Sh){
  shapiro.test(M0_RIS_W_list_base[[Sh]]$residuals)$p
})

Shap_M0_df

M1_RIS_W_list_base <- M1_RIS_W_list_R2

```

```

names(M1_RIS_W_list_base) <-
str_remove(str_remove(str_remove(names(M1_RIS_W_list_base),
"Model"), "Base"), "RE: ")
#M1_RIS_W_list_R2
par(mfrow=c(1,2))
QQ_M1_list <- lapply(1:length(M1_RIS_W_list_base),function(QQ){
  IntrIS_MLE_W.SM1.sdres <- residuals(M1_RIS_W_list_base[[QQ]],type =
"pearson")
  qqnorm(IntrIS_MLE_W.SM1.sdres,
  ylab="Standardized Res.",
  xlab="Normal Scores",
  main=paste("M1",names(M1_RIS_W_list_base[QQ]),"Standard.
Res.))
  qqline(IntrIS_MLE_W.SM1.sdres)
})

Shap_M1_df <- data.frame(Model =
paste("M2",names(M1_RIS_W_list_base)))
Shap_M1_df$Shapiro_p.valor <- Shap_M1_v <-
sapply(1:length(M1_RIS_W_list_base),function(Sh){
  shapiro.test(M1_RIS_W_list_base[[Sh]]$residuals)$p
})

Shap_M1_df

M2_RIS_W_list_base <- M2_RIS_W_list_R2

names(M2_RIS_W_list_base) <-
str_remove(str_remove(str_remove(names(M2_RIS_W_list_base),
"Model"), "Base"), "RE: ")
#M2_RIS_W_list_R2
par(mfrow=c(1,2))
QQ_M2_list <- lapply(1:length(M2_RIS_W_list_base),function(QQ){
  IntrIS_MLE_W.SM2.sdres <- residuals(M2_RIS_W_list_base[[QQ]],type =
"pearson")
  qqnorm(IntrIS_MLE_W.SM2.sdres,
  ylab="Standardized Res.",
  xlab="Normal Scores",
  main=paste("M2",names(M2_RIS_W_list_base[QQ]),"Standard.
Res.))
  qqline(IntrIS_MLE_W.SM2.sdres)
})

Shap_M2_df <- data.frame(Model =
paste("M2",names(M2_RIS_W_list_base)))
Shap_M2_df$Shapiro_p.valor <- Shap_M2_v <-
sapply(1:length(M2_RIS_W_list_base),function(Sh){
  shapiro.test(M2_RIS_W_list_base[[Sh]]$residuals)$p
})

Shap_M2_df
` ``

```

El que es veu en el test Shapiro no seria bo ja que sembla que els models s'allunyen de la normalitat al afegir els efectes aleatoris RIS, tot i així visualment es pot veure que hi ha poques diferències en el QQplot, pel que potser seria bo no prendre's aquest test de Shapiro com definitiu (en cas que es volgués aprofundir més es podria intentar fer més proves de normalitat).

```

```{r funció test homoscedasticity}
#Variation of "Levene's Test"
lev.test <- function(lm,id,df){ #Posar la funció, el nom dels
subjectes i el dataset
  df$Model.F.Res<- residuals(lm) #extracts the residuals and places
them in a new column in our original data table
  df$Abs.Model.F.Res <-abs(df$Model.F.Res) #creates a new column with
the absolute value of the residuals
  df$Model.F.Res2 <- df$Abs.Model.F.Res^2 #squares the absolute values
of the residuals to provide the more robust estimate
  Levene.Model.F <- lm(paste("Model.F.Res2 ~ ",paste(id),sep=""),
data=df) #ANOVA of the squared residuals
  return(anova(Levene.Model.F))
}
```

```

```

```{r Diag H Model simple Lake1 RIS INT W}

RP_M0_list <- lapply(1:length(M0_RIS_W_list_base),function(RP){
  plot(M0_RIS_W_list_base[[RP]],which=1,main=paste("M0",names(M0_RIS_W
_list_base)[RP]))
})

marrangeGrob(RP_M0_list, ncol=2, nrow=1)

LT_M0_list <- lapply(1:length(M0_RIS_W_list_base),function(LT){
  lev.test(M0_RIS_W_list_base[[LT]],"Ident",LAKE1)
})

names(LT_M0_list)<-names(M0_RIS_W_list_base)
LT_M0_list

RP_M1_list <- lapply(1:length(M1_RIS_W_list_base),function(RP){
  plot(M1_RIS_W_list_base[[RP]],which=1,main=paste("M1",names(M1_RIS_W
_list_base)[RP]))
})

marrangeGrob(RP_M1_list, ncol=2, nrow=1)

LT_M1_list <- lapply(1:length(M1_RIS_W_list_base),function(LT){
  lev.test(M1_RIS_W_list_base[[LT]],"Ident",LAKE1)
})

names(LT_M1_list)<-paste("Logt",names(M1_RIS_W_list_base))
LT_M1_list

RP_M2_list <- lapply(1:length(M2_RIS_W_list_base),function(RP){
  plot(M2_RIS_W_list_base[[RP]],which=1,main=paste("M2",names(M2_RIS_W
_list_base)[RP]))
})

marrangeGrob(RP_M2_list, ncol=1, nrow=1)

LT_M2_list <- lapply(1:length(M2_RIS_W_list_base),function(LT){
  lev.test(M2_RIS_W_list_base[[LT]],"Ident",LAKE1)
})

names(LT_M2_list)<-paste("Bct",names(M2_RIS_W_list_base))
LT_M2_list
```

```

En aquest resultat és on es pot veure realment el que s'ha aconseguit amb les funcions de modulació de variància ja que podem recuperar la homocedasticitat, una de les propietats essencials perquè un model no tingui un biaix important en la desviació.

### Optimització

S'hauria de prendre com a òptim el model que reuneixi les propietats més bones, pel que recuperant els resultats últims es pot fer un petit resum. Donat que a priori no hi ha un factor de pes per cada paràmetre de bondat d'ajust estudiada, l'estratègia a seguir serà tenint en compte els paràmetres Sigma, AIC/BIC,  $R^2$ , p-valor Shapiro i p-valor Levene-Test(modificat), seleccionar per cada paràmetre 3 models, combinar la selecció i representar els models resultants:

```

```{r Resum models RIS W}
NomsModelsRISW <- paste(rep(paste(rep(paste("M.int
RIS",c("MLE","REML")),4),
                        rep(c("No.W","VarI","VarE","VarP"),e
ach=2)),3),
                        rep(c("No.t","Logt","Bct"),each=8))

NomsModelsRISW<-NomsModelsRISW[-which(NomsModelsRISW=="M.int RIS MLE
VarP No.t","M.int RIS REML VarP No.t")]
NomsModelsRISW<-NomsModelsRISW[-which(NomsModelsRISW=="M.int RIS MLE
VarP Bct")]

ResumRISW <- rbind(intM0_QI[[1]],intM1_QI[[1]],intM2_QI[[1]])
rownames(ResumRISW) <- NomsModelsRISW
ResumRISW$R2 <-
c(M0_RIS_W_dfa_R2$R2,M1_RIS_W_dfa_R2$R2,M2_RIS_W_dfa_R2$R2)
ResumRISW$Shapiro_p.valor <-
c(Shap_M0_df$Shapiro_p.valor,Shap_M1_df$Shapiro_p.valor,Shap_M2_df$Sha
piro_p.valor)

LT_list <- list(LT_M0_list,LT_M1_list,LT_M2_list)
ResumRISW$Levene_p.valor <-
unlist(sapply(1:length(LT_list),function(n){
      sapply(1:length(LT_list[[n]]),function(i){LT_list[[n]][[i]]
$`Pr(>F)`[1]})
}))

#Es genera una funció per convertir i graficar la taula
Convdf_plot <- function(df){
  p1<-c(1)
  p2<-c(2:3)
  p3<-c(4)
  p4<-c(5)
  p5<-c(6)

  p <- list(p1,p2,p3,p4,p5)

  pnames<-c("Sigma","AIC i BIC","R2","Shapiro p.valor","Levene p-
valor")

  ML_table_p <- lapply(1:length(p),function(pvar){
    Table <- data.frame(Model = if(length(p[[pvar]])>1){
      rep(rownames(df),ncol(df[,p[[pvar]]])
    } else{
      rownames(df)
    }
  })
}

```



```

    })
    Table$Model <- factor(Table$Model, levels = rownames(df))
    Table$ParQ <- unlist(df[,p[[pvar]])]
    Table$Par <- rep(colnames(df)[p[[pvar]]],each=nrow(df))
    return(Table)
  })

  ML_plot <- lapply(1:length(ML_table_p),function(plot){
    ggplot(data=ML_table_p[[plot]], aes(x=Model, y=ParQ, fill=Par)) +
    geom_bar(stat="identity", position=position_dodge()+
    scale_fill_brewer(palette="Set3") + theme_minimal() +
    labs(title="Optimització Model",y = paste(pnames[plot])) +
    theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))
  })
  return(ML_plot)
}

#Selecció #unique
optorder <- c("F","F","F","T","T","T")

BestFi <- sort(unique(as.vector(sapply(1:ncol(ResumRISW),function(v){
  order(ResumRISW[,v],decreasing=optorder[v])[1:2]
}))))

Convdf_plot(ResumRISW[BestFi,])

#No es considera el shapiro ni els models amb Sigmes massa grans
ResumRISW2 <- ResumRISW[-which(ResumRISW$Sigma>100),]

BestFi2 <- sort(unique(as.vector(sapply(1:ncol(ResumRISW2[, -
4]),function(v){
  order(ResumRISW2[,v],decreasing=optorder[v])[1:2]
}))))

Convdf_plot(ResumRISW2[BestFi2,])

```

De manera resumida:

- + Sigma: Els models transformats amb funció VarPower semblen donar més bons resultats.
- + AIC/BIC: Models transformats donen més bons resultats. Els models transformats logarítmics semblen més eficients i en especial els que tenen funció de variància varExp i varPower.
- + R2: Els models transformats semblen donar més bons resultats en especial el logarítmic sense modulació de variància, però també els models transformats per varPower de log i Box-Cox.
- + Homocedasticitat: Semblen que els models amb més bons resultats serien els transformats amb Log i BC i modulació de variància.

En conjunt sembla que seria ideal utilitzar els dos models transformats amb efectes aleatoris i funció de modulació de variància.

Dels models elegits és interessant veure'n les propietats en quant a matriu de variància-covariància i de correlacions:

```

```{r Var-Cov RIS W Opt}
Opt <- list(M1_RIS_W_list_base[[8]],M2_RIS_W_list_base[[7]])
names(Opt) <- c("M.Logt.RIS.W-VarP.", "M.BCt.RIS.W-V")

lapply(1:length(Opt),function(B){

```

```

    VarCorr(Opt[[B]])
  })

lapply(1:length(Opt), function(B){
  getVarCov(Opt[[B]], type="conditional")
})

lapply(1:length(Opt), function(B){
  a<-getVarCov(Opt[[B]], type="marginal")
  b<-cov2cor(a[[1]])
  return(list(a,b))
})
...

```

Tot i obtenir resultats diferents per cada conjunt He les conclusions són semblants:

+ En els primers resultats s'observen a les matrius D donant poca informació, simplement es veu la baixa escala obtinguda per la variància residual.

+ La segona sèria de matrius són les matrius R que reflexen la matriu de variàncies-covariàncies residuals dels diferents temps que tenen tots els lots. S'està en el cas on es considera la variància residual heterogènia i per tant es veu com la funció addicional recalcula de manera que a mesura que s'avança en els punts de temps augmenta la variància pròpia del temps (i si el conjunt és més dispers també més variància es calcula). En els 2 casos sembla que el sistema calcula que a mesura que avança el temps es redueix la variància.

+ La tercera sèria de matrius mostra:

+ La matriu primera de cada punt de la sèria és de tipus marginal on es suma la variància residual i la variància degut a l'efecte aleatori en la diagonal, mentre que la variància degut a l'efecte aleatori forma part de la covariància en aquest cas. Això és degut a que en aquesta matriu es reflexa l'efecte de la variància total dins un dels individus (equivalent per tots els altres) que té la variància de l'efecte aleatori sumat a la variància residual al centrar-nos en un punt, i es relaciona amb la resta de punts corresponents als altres temps amb el factor calculat aleatori de la variància dins el mateix lot. En els dos casos hi ha un recàlcul a partir de la funció de la variància que determina com es relacionen les diferents repeticions. NO sembla haver-hi un patró senzill en funció de la distància ja que el sistema ho calcula amb les dades de què disposa.

+ Es reflecteix també a la segona matriu on es plasma la correlació de variàncies i on es veu que s'ha calculat una correlació variant segons quin temps es relaciona amb quin, però baixes en general (és a dir que per aquest cas no s'ha aconseguit trobar un model que expliqui bé aquesta correlació, o simplement al haver una dispersió en funció del temps baixa la correlació).

Tot i haver afegit la funció per modular la covariància no sembla tenir un efecte molt gran tal com es veu en la correlació entre temps, però possiblement pot corregir el model encara que sigui lleugerament. En etapes anteriors també s'ha vist com realment l'efecte aleatori RIS pot haver suposat un canvi positiu pel model.

Els models finals són:

```

```{r FI mod}
lapply(Opt, summary)
```

```