

Modelització de dades longitudinals amb efectes aleatoris de intercepció/pendent i presència de dades perdudes en el context dels estudis d'estabilitat – ANNEX 1

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 la simulació de
models mixtos en el context dels estudis
d'estabilitat

```

---
title: "TFM_ANNEX-1_SIMULACIÓ_MODELS_MIXTOS"
author: "Francesc Bernad Martin"
date: "May 30, 2019"
output:
  word_document:
    toc: yes
    toc_depth: '4'
  pdf_document:
    toc: yes
    toc_depth: 4
  html_document:
    df_print: paged
    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)
#Es posa EL CACHE EN TRUE PQ VAGI MÉS RÀPID A GENERAR INFORMES
```

```{r libraries, message=F}
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")

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

# MODEL MIXT - DATASET HOMOGENI HO

```

GENERACIÓ DATASET

Per la generació dels datasets s'utilitza el generador de valors de distribució uniforme mitjançant el paquet ****Uniform****.

Es genera un dataset simulat amb rangs mitjos en quant a individus i temps (posteriorment es valorarà la opció de fer un escombrat en tot el rang d'aquestes 2 variables):

```
```{r Average DS}
set.seed(7982)
i3 <- 3
i10 <- 10

j4 <- 4
j10 <- 10

(av_i <- round(mean(c(3,10))))
(av_j <- round(mean(c(4,10))))
TA <- c(0,3,6,9,12,18,24,30,36,42) # Es genera un vector amb els temps
més habituals d'estabilitat del qual es prenen els valors segons la
longitud del dataset a utilitzar (es tindrà en compte en el mateix
ordre que està posada encara que es podria contemplar agafant valors
de manera equilibrada entre el principi i el final)
TA_av <- TA[1:av_j]
Av_DS <- data.frame(Lot = rep(paste("Lot",seq(1:av_i)),each=av_j))
Av_DS$NTemps <- as.factor(rep(paste("Temps",seq(1:av_j)),av_i))
Av_DS$Temps <- rep(TA[1:av_j],av_i)
```
```

Per la generació de datasets s'utilitza sempre la mateixa llavor essent aquesta `7982`.

GENERACIÓ DATASET Ho1 (NÚVOL DE PUNTS HOMOGENI)

Es generen punts a l'atzar dins un rang reduït (a efectes d'escalat al voltant de 0 entre -1 i 1) mitjançant la funció ***runif*** per obtenir un nuvol de punts amb variància homogènia:

```
```{r Ho1 Dataset}
set.seed(7982)
Av_DS$Ho1 <- runif(n=av_i*av_j, min=-1, max=1) #Observacions constant
de rang petit entre -1 i 1 per simular que no hi ha tendències ni
diferències

kable(head(Av_DS[,c(colnames(Av_DS)[1:3],"Ho1")]), booktabs = T) %>%
kable_styling(latex_options = c("striped","condensed"))
```
```

GENERACIÓ DATASET Ho2 (HOMOGENI PERÒ AMB VARIACIÓ DEPENDENT DEL LOT)

En aquesta ocasió es genera un dataset amb variància homogènia, però que depengui del lot. Per fer això es considera un rang petit de variància entre medicions de cada lot i una variació de les mitges de cada lot aleatòria entre un ordre superior de magnitud. Per fer es fa servir la funció ***runif*** combinada amb alguna de les funcions de la família ***apply*** per modificar el rang de cada lot de manera seqüencial:

```

```{r Ho2 Dataset}
set.seed(7982)
Av_DS$Ho2 <- as.vector(sapply(1:av_i, function(i){
 m_lot <- runif(1, min=-1, max=1)
 runif(n=av_j, min=m_lot-0.1, max=m_lot+0.1)
}))

kable(head(Av_DS[,c(colnames(Av_DS)[1:3],"Ho2")]), booktabs = T) %>%
kable_styling(latex_options = c("striped","condensed"))
```

```

```
## VISUALITZACIÓ DATASETS GENERATS HO
```

Es visualitza gràficament les diferències entre els datasets generats. Per veure amb més claredat les diferències es visualitza primer mitjançant un gràfic tipus *scatterplot* on es veu l'error estàndard global i la diferenciació per lot, i posteriorment es visualitza mitjançant els diagrames de caixes per apreciar més fàcilment els errors estàndards en el temps:

```
### *SCATTERPLOTS* Ho
```

es veu els datasets amb variància homogènia.

```

```{r Ho Scatterplots}
labsHo_1 <- c("Dataset Ho1 - Núvol de punts","Dataset Ho2 -
Homogeneïtat en lot")
labsHo_2 <- c("Dataset Ho1 s/ lot","Dataset Ho2 s/ lot")

HoVars <- c("Ho1","Ho2")

Av_DS_plotHo_sc1 <- lapply(1:length(HoVars),function(graph){
 ggplot(Av_DS, aes(x=Temps, y=Av_DS[,HoVars[graph]])) + geom_point()
+ geom_smooth(linetype="dashed") +
 labs(title=labsHo_1[graph], y = "Observacions")
})

Av_DS_plotHo_sc2 <- lapply(1:length(HoVars),function(graph){
 ggplot(Av_DS, aes(x=Temps, y=Av_DS[,HoVars[graph]], color=Lot,
group=Lot)) + geom_point() + geom_line() +
 labs(title=labsHo_2[graph], y = "Observacions")
})

ydens_DS_plotHo <- lapply(1:length(HoVars),function(graph){
 ggplot(Av_DS, aes(x=Av_DS[,HoVars[graph]])) +
 geom_density(color="darkblue", fill="lightblue") +
 geom_vline(aes(xintercept=mean(Av_DS[,HoVars[graph]])),
color="blue", linetype="dashed", size=1)+
 coord_flip() + labs(x = "Observacions")
})

grid.arrange(Av_DS_plotHo_sc1[[1]], ydens_DS_plotHo[[1]],
Av_DS_plotHo_sc1[[2]], ydens_DS_plotHo[[2]],
ncol=2, widths=c(4, 1))

grid.arrange(Av_DS_plotHo_sc2[[1]], ydens_DS_plotHo[[1]],
Av_DS_plotHo_sc2[[2]], ydens_DS_plotHo[[2]],
ncol=2, widths=c(4, 1))
```

```

Els gràfics de densitat ens mostren que en el cas del núvol de punts les dades tenen un repartiment bastant homogeni en tot el rang de observacions, mentre que en el cas del repartiment per lot s'observa que es centren on aleatòriament els lots hagin centrat la seva mitja. Tot i que en el gràfic sense distinció de lot es veu poques diferències entre els 2 conjunts, en el gràfic posterior es veu que realment sí que es diferencien en el grup que pertany cada dada.

```
#### *BOXPLOTS* Ho
```

es veu els datasets amb variància homogènia.

```
```{r Ho Boxplots}
Av_DS_plotHo_bp1 <- lapply(1:length(HoVars),function(graph){
 ggplot(Av_DS, aes(x=N Temps, y=Av_DS[,HoVars[graph]])) +
 geom_boxplot()+ geom_dotplot(binaxis='y', stackdir='center',
dotsize=1)+
 labs(title=labsHo_1[graph], y = "Observacions")
})

grid.arrange(Av_DS_plotHo_bp1[[1]], ydens_DS_plotHo[[1]],
Av_DS_plotHo_bp1[[2]], ydens_DS_plotHo[[2]],
..., ncol=2, widths=c(4, 1))
```
```

A part de la diferència d'escala del rang en aquests *boxplot* tot i haver petites diferències no es pot veure clarament que no hi hagi homogeneïtat, tant en el cas del núvol de punts com el de diferenciació per lots.

```
## MODELS LINEALS Ho
```

Per l'ajust dels models lineals dels datasets en aquest cas es consideren diversos models diferents per comparar entre ells. Es pretén generar un model lineal simple i un model lineal d'efectes mixtos amb la intercepció aleatòria. És important tenir en compte que els datasets generats s'han de considerar independents entre ells i per tant no són directament comparables al ajustar els models, pel que l'anàlisi que es segueix és veure les diferències entre l'ajust dels diferents models a un mateix dataset i posteriorment es compara els diferents datasets, però en termes de la bondat d'ajust dels models aplicats:

```
#### ML I/IS Ho
```

Es genera el model lineal simple i el model amb la variable temps per comparar-los i definir el model de base simple. (s'aprofita per convertir-ho en una funció R per repetir aquest anàlisi en els altres datasets):

```
```{r ML I/IS funció}
Comp_I_IS_LM <- function(var,ds){
 (ML_I_s <- summary(ML_I <- lm(paste(var,"~",1,sep=" "), data = ds)))
 (ML_IS_s <- summary(ML_IS <- lm(paste(var,"~"," Temps",sep=" "), data
= ds)))
 (ML_I_IS_an <- anova(ML_IS,ML_I))

 l <- list(ML_I_s,ML_IS_s,ML_I_IS_an,ML_I,ML_IS)
 names(l) <- c("OLS I","OLS IS","Anova comp. efecte S","OLS I
base","OLS IS base")
 return(l)
}
```
```

```
````
```

```
````{r ML I/IS Ho1/Ho2}
(Comp_I_IS_Ho1 <- Comp_I_IS_LM("Ho1",Av_DS))[1:3]

(Comp_I_IS_Ho2 <- Comp_I_IS_LM("Ho2",Av_DS))[1:3]
````
```

Per començar s'observa que, com és d'esperar, afegir la variable temps al model no té sentit i no dona una significació adicional al model per explicar la resposta ja que el resultat de l'Anova corresponent entre models dona un p-valor de ``r round(Comp_I_IS_Ho1[[3]]$"Pr(>F)"[2],4)`` (cas Ho1) i ``r round(Comp_I_IS_Ho2[[3]]$"Pr(>F)"[2],4)`` (cas Ho2), pel que per regla general sempre és preferible utilitzar el model més simple possible que no afegeix tants factors redundants que poden fer augmentar la variabilitat.

En el model només amb intercepció la prova de significació de la intercepció no té molt de sentit i sol ser més important veure l'error estàndar de l'estimació del paràmetre i l'error estàndar residual. En aquest sentit si es comparen els dos datsets s'observa a primera vista que és més baix en el cas de Ho2:

```
+ Ho1: `r round(Comp_I_IS_Ho1[[1]]$coefficients[,2],4)` (Error
Estimació); `r round(Comp_I_IS_Ho1[[1]]$sigma,4)` (Error residual)
+ Ho2: `r round(Comp_I_IS_Ho2[[1]]$coefficients[,2],4)` (Error
Estimació); `r round(Comp_I_IS_Ho2[[1]]$sigma,4)` (Error residual)
```

Al tenir un patró de punts més definit a Ho2 possiblement es produeix per aquest motiu la diferència en l'ajust més simple.

```
ML RI Ho
```

Avançant a partir del model simple amb intercepció ML I es fa l'ajust de models amb efecte aleatori (es genera adicionalment la funció):

```
````{r ML I RI funció}
Comp_I_RI_LM <- function(var,ds) {
  fm1 <- formula(paste(var,"~",1,"+", "Error(Lot)"))
  ML_I_RI_sOLS <- summary(ML_I_RI_OLS <- aov(fm1, data = ds))
  ML_I_RI_OLS
  ML_I_RI_OLS_err <- sigma(summary.lm(ML_I_RI_OLS$Within))

  fm2 <- formula(paste(var,"~",1))

  ML_I_RI_sMLE <- summary(ML_I_RI_MLE <- lme(fm2, random = ~1|Lot,
data = ds, method="ML"))

  ML_I_RI_sREML <- summary(ML_I_RI_REML <- lme(fm2,random = ~1|Lot,
data = ds, method="REML"))

  l <-
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","MLE I RI","REML I RI","OLS I RI base","MLE
I RI base","REML I RI base")

  return(l)
}
````
```



```

```{r ML I RI Ho1/Ho2}

(Comp_I_RI_Ho1 <- Comp_I_RI_LM("Ho1",Av_DS))[1:3]

fmHo1 <- formula(paste("Ho1","~",1))
(eMLE_Ho1<-exactRLRT(lme(fmHo1, random = ~1|Lot, data = Av_DS,
method="ML")))
(eREML_Ho1<-exactRLRT(lme(fmHo1, random = ~1|Lot, data = Av_DS,
method="REML")))

(Comp_I_RI_Ho2 <- Comp_I_RI_LM("Ho2",Av_DS))[1:3]

fmHo2 <- formula(paste("Ho2","~",1))
(eMLE_Ho2<-exactRLRT(lme(fmHo2, random = ~1|Lot, data = Av_DS,
method="ML")))
(eREML_Ho2<-exactRLRT(lme(fmHo2, random = ~1|Lot, data = Av_DS,
method="REML")))
```

```

Es podria considerar una mida mostral correcte la d'aquest model base amb `r nrow(Av\_DS)` observacions, a part de no tenir més efectes fixos que la pròpia mitja. En aquest càlcul es comencen a veure les diferències dels datasets simulats:

+ Ho1: Com s'espera no hi ha diferències molt grans entre el càlcul del model amb els 3 sistemes veient una desviació residual estàndard de `r round(sigma(summary.lm(Comp\_I\_RI\_Ho1[[1]]),4)` (OLS Mètode ANOVA), `r round(Comp\_I\_RI\_Ho1[[2]]\$sigma,4)` (MLE) i `r round(Comp\_I\_RI\_Ho1[[3]]\$sigma,4)` (REML) ni tampoc comparat amb els errors estàndards dels models més simples sense efectes aleatoris.

+ Ho2: Com en Ho1 no hi ha diferències significatives entre els 3 sistemes veient una desviació residual estàndard de `r round(sigma(summary.lm(Comp\_I\_RI\_Ho2[[1]]),4)` (OLS Mètode ANOVA), `r round(Comp\_I\_RI\_Ho2[[2]]\$sigma,4)` (MLE) i `r round(Comp\_I\_RI\_Ho2[[3]]\$sigma,4)` (REML). A diferències de Ho1 sí que s'observa una disminució possiblement significativa de la desviació residual essent `r round(Comp\_I\_IS\_Ho2[[1]]\$sigma,4)` la del model sense efecte aleatori RI i `r round(mean(sigma(summary.lm(Comp\_I\_RI\_Ho2[[1]]),Comp\_I\_RI\_Ho2[[2]]\$sigma,Comp\_I\_RI\_Ho2[[3]]\$sigma),4)` la mitja entre els errors estàndards dels 3 mètodes en els models amb l'efecte aleatori RI, a part que en aquest cas les proves de significació de l'efecte aleatori donen significatives pels dos mètodes de ML (p-valors de `r eMLE\_Ho2\$p.value` i `r eREML\_Ho2\$p.value` respectivament).

Al existir un patró que modula la localització dels punts per cada lot en Ho2, encara que els lots tinguin una ubicació aleatòria, l'efecte aleatori sembla que permet ajustar aquesta característica per augmentar la certesa del model.

## RESUM I DIAGNÒSTIC MODELS HO

```

```{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 = "Residual Std. Error (Sigma)")) +
      theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))

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

  ```{r Resum ML Ho}
 Ho1_R <-
list(Comp_I_IS_Ho1[[4]],Comp_I_RI_Ho1[[1]],Comp_I_RI_Ho1[[2]],Comp_I_R
I_Ho1[[3]])
names(Ho1_R) <- c(names(Comp_I_IS_Ho1)[4],names(Comp_I_RI_Ho1)[1:3])
ML_Resum(Ho1_R, "Ho1")

 Ho2_R <-
list(Comp_I_IS_Ho2[[4]],Comp_I_RI_Ho2[[1]],Comp_I_RI_Ho2[[2]],Comp_I_R
I_Ho2[[3]])
names(Ho2_R) <- c(names(Comp_I_IS_Ho2)[4],names(Comp_I_RI_Ho2)[1:3])
ML_Resum(Ho2_R, "Ho2")

```

...

Com s'ha comentat en l'anterior càlcul, es denoten aquí encara més les diferents característiques dels datasets:

+ Ho1: S'observen resultats dels indicadors molt similars i que simplement a nivell de comparació podria semblar que el model ajustat per ANOVA clàssic seria el millor, els errors estàndards residuals queden molt igualades. Al ser un model homogeni no es preveu necessari utilitzar models complexos com els de màxima versemblança essent correctes els models calculats de la manera clàssica.

+ Ho2: Tant l'error estàndard residual com els indicadors d'ajust del model AIC i BIC donen a entendre les diferències entre el model bàsic i els models amb efectes aleatoris que semblen ser clau per millorar l'ajust del model. En aquest cas concret no hi ha grans diferències entr els 3 models amb efecte aleatori RI, tot i que a la vista dels gràfics podria semblar que el millor model seria l'ajust pels mètodes clàssics amb efecte aleatori.

S'extreu a partir dels models les estimacions de la matriu D per veure l'error estàndard dels efectes aleatoris i els residus, i la matriu R on es veu com es distribueix l'error estàndard residual en el model:

```
```{r D Matrix Ho1/Ho2}
Ho1_B <- list(Comp_I_RI_Ho1[[5]])
names(Ho1_B) <- names(Comp_I_RI_Ho1)[5]

Ho2_B <- list(Comp_I_RI_Ho2[[5]])
names(Ho2_B) <- names(Comp_I_RI_Ho2)[5]

lapply(1:length(Ho1_B),function(B){
  VarCorr(Ho1_B[[B]])
})

lapply(1:length(Ho2_B),function(B){
  VarCorr(Ho2_B[[B]])
})

lapply(1:length(Ho1_B),function(B){
  getVarCov(Ho1_B[[B]],type="conditional")
})

lapply(1:length(Ho2_B),function(B){
  getVarCov(Ho2_B[[B]],type="conditional")
})

lapply(1:length(Ho1_B),function(B){
  a<-getVarCov(Ho1_B[[B]],type="marginal")
  b<-cov2cor(a[[1]])
  return(list(a,b))
})

lapply(1:length(Ho2_B),function(B){
  a<-getVarCov(Ho2_B[[B]],type="marginal")
  b<-cov2cor(a[[1]])
  return(list(a,b))
})
```

...

Les matrius s'extreuen del model REML i es consideren que no variarien gaire en els altres models provats de efectes aleatoris:

+ En els primers resultat s'observen a les matrius D altra vegada les diferències entre l'variància residual i l'efecte aleatori com canvia entre un conjunt i l'altra.

+ Els resultats 3 i 4 on s'extreuen 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 homogènia i igual en tots els punts de temps.

+ Els resultats 5 i 6 reflexen:

+ La matriu tipus marginal on es suma la variància i la variància degut a l'efecte aleatori en la diagonal, mentre que la que és 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 de l'efecte aleatori sumat a l'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 l'variància dins el mateix lot. És interessant notar també en aquesta matriu que en la diagonal on es sumen els errors estàndards resulta una variància global menor en el model que en teoria està més ben ajustat de Ho2.

+ Es reflecteix també a la segona matriu on es plasma la correlació de variàncies i on es veu que pel primer cas no hi ha correlació de variàncies entre els diferents temps i en canvi en el segon conjunt amb Ho2 sí que es pot veure un nivell clarament alt de correlació.

Es pot graficar el gràfic de distribució dels efectes aleatoris trobats, però com es veu a continuació té poc sentit al tenir un nombre de lots habitualment tant petit:

```
```{r Ho Diag plots}
qqnorm(Comp_I_RI_Ho1[[5]], ~ranef(.))
qqnorm(Comp_I_RI_Ho2[[5]], ~ranef(.))
```
```

Finalment si es genera els gràfics bàsics de diagnòstic del model simple:

```
```{r Covar Ho}
par(mfrow=c(1,2))
plot(Comp_I_IS_Ho1[[5]],which=c(1:2),main="OLS IS Ho1")
plot(Comp_I_IS_Ho2[[5]],which=c(1:2),main="OLS IS Ho2")
par(mfrow=c(1,1))
```

```
par(mfrow=c(1,2))
plot(Comp_I_IS_Ho1[[4]],which=c(1:2),main="OLS I Ho1")
plot(Comp_I_IS_Ho2[[4]],which=c(1:2),main="OLS I Ho2")
par(mfrow=c(1,1))
```

...

S'observa que la distribució dels residus és relativament homogènia pels dos casos quadrant amb el que es pretenia generar. En l'anàlisi d'aquest dataset no es considera per tant la inclusió de una funció per modular les variabilitat entre repeticions (o covariables).

```
PROVES DE VARIACIÓ DE LOTS/TEMPS HO
```

Es desenvolupen unes fòrmules personalitzades dins el codi que ens permeti graficar \*heatmaps\* dels indicadors sigma, AIC i BIC passant pel rang definit de lots i temps possibles de l'estudi tant pel mètode clàssic ANOVA com pels mètodes de màxima versemblança (ML):

```
```{r formula HeatMapOLS}
HeatMapRange_OLS <- function(Key,df,fm){
```

```

ML <- lapply(i,function(i_){
  lapply(j,function(j_){
    df <- data.frame(p = as.vector(unlist(df[c(1:j_),c(1:i_)])))
    colnames(df)[1] <- Key
    df$Lot <- rep(paste("Lot",c(1:i_),sep="-"),each=j_)
    aov(fm, data = df)$Within
  })
})

sigma <- sapply(1:length(i),function(i_1){
  sapply(1:length(j),function(j_1){
    sigma(summary.lm(ML[[i_1]][[j_1]]))
  })
})

sigma <- data.frame(sigma = as.vector(unlist(sigma)))
sigma$NLots <- rep(i,each=length(j))
sigma$NTemps <- rep(j,length(i))

AIC <- sapply(1:length(i),function(i_1){
  sapply(1:length(j),function(j_1){
    AIC(ML[[i_1]][[j_1]])
  })
})
AIC <- data.frame(AIC = as.vector(unlist(AIC)))
AIC$NLots <- rep(i,each=length(j))
AIC$NTemps <- rep(j,length(i))

BIC <- sapply(1:length(i),function(i_1){
  sapply(1:length(j),function(j_1){
    BIC(ML[[i_1]][[j_1]])
  })
})
BIC <- data.frame(BIC = as.vector(unlist(BIC)))
BIC$NLots <- rep(i,each=length(j))
BIC$NTemps <- rep(j,length(i))

p1<-ggplot(sigma, aes(x = NLots, y = NTemp, fill = sigma)) +
geom_tile() +
  labs(title="Sigma Random ef. OLS HeatM.")
p2<-ggplot(AIC, aes(x = NLots, y = NTemp, fill = AIC)) + geom_tile()
+
  labs(title="AIC Random ef. OLS HeatM.")
p3<-ggplot(BIC, aes(x = NLots, y = NTemp, fill = BIC)) + geom_tile()
+
  labs(title="BIC Random ef. OLS HeatM.")

grid.arrange(p1,p2,p3,ncol=2)
}
...

```{r formula HeatMapML}
HeatMapRange_ML <- function(Key,df,fm,re,met,weight,weightF){ #Weight
0 per no tenir weights, 1 varIdent, 2 varExp, 3 varPower.

if (weight=="0"){
 ML <- lapply(i[1:(ncol(df)-i[1]+1)],function(i_){
 lapply(j[1:(nrow(df)-j[1]+1)],function(j_){
 df <- data.frame(p = as.vector(unlist(df[c(1:j_),c(1:i_)])))
 colnames(df)[1] <- Key

```

```

df$Lot <- rep(paste("Lot",c(1:i_),sep="-"),each=j_)
df$Temps <- rep(TA[1:j_],i_)
summary(lme(fm, random = re, data = df, method=met))
})
})
} else {
ML <- lapply(i[1:(ncol(df)-i[1]+1)],function(i_){
lapply(j[1:(nrow(df)-j[1]+1)],function(j_){
df <- data.frame(p = as.vector(unlist(df[c(1:j_),c(1:i_)])))
colnames(df)[1] <- Key
df$Lot <- rep(paste("Lot",c(1:i_),sep="-"),each=j_)
df$Temps <- rep(TA[1:j_],i_)
summary(lme(fm, random = re, data = df, method=met, weights =
if (weight=="1"){varIdent(form=weightF)}
else if (weight=="2"){varExp(form=weightF)}
else if (weight=="3"){
if(weightF=="0"){varPower()} else
{varPower(form=weightF)}
}
))
})
})
})
sigmaM <- sapply(1:length(i[1:(ncol(df)-i[1]+1)]),function(i_l){
sapply(1:length(j[1:(nrow(df)-j[1]+1)]),function(j_l){
ML[[i_l]][[j_l]]$sigma
})
})
sigmaM <- data.frame(sigma = as.vector(unlist(sigmaM)))
sigmaM$NLots <- rep(i[1:(ncol(df)-i[1]+1)],each=length(j[1:(nrow(df)-
j[1]+1)]))
sigmaM$NTemps <- rep(j[1:(nrow(df)-j[1]+1)],length(i[1:(ncol(df)-
i[1]+1)]))
AICM <- sapply(1:length(i[1:(ncol(df)-i[1]+1)]),function(i_l){
sapply(1:length(j[1:(nrow(df)-j[1]+1)]),function(j_l){
ML[[i_l]][[j_l]]$AIC
})
})
AICM <- data.frame(AIC = as.vector(unlist(AICM)))
AICM$NLots <- rep(i[1:(ncol(df)-i[1]+1)],each=length(j[1:(nrow(df)-
j[1]+1)]))
AICM$NTemps <- rep(j[1:(nrow(df)-j[1]+1)],length(i[1:(ncol(df)-
i[1]+1)]))
BICM <- sapply(1:length(i[1:(ncol(df)-i[1]+1)]),function(i_l){
sapply(1:length(j[1:(nrow(df)-j[1]+1)]),function(j_l){
ML[[i_l]][[j_l]]$BIC
})
})
BICM <- data.frame(BIC = as.vector(unlist(BICM)))
BICM$NLots <- rep(i[1:(ncol(df)-i[1]+1)],each=length(j[1:(nrow(df)-
j[1]+1)]))
BICM$NTemps <- rep(j[1:(nrow(df)-j[1]+1)],length(i[1:(ncol(df)-
i[1]+1)]))
p1<-ggplot(sigmaM, aes(x = NLots, y = NTemps, fill = sigma)) +
geom_tile() +
labs(title="Sigma Random ef. ML HeatM.")

```

```

p2<-ggplot(AICM, aes(x = NLots, y = NTemps, fill = AIC)) + geom_tile()
+
 labs(title="AIC Random ef. ML HeatM.")
p3<-ggplot(BICM, aes(x = NLots, y = NTemps, fill = BIC)) + geom_tile()
+
 labs(title="BIC Random ef. ML HeatM.")

grid.arrange(p1,p2,p3,ncol=2)
}
...

```{r Heatmaps Ho1}
set.seed(7982)
i <- seq(i3,i10)
j <- seq(j4,j10)

DS_Ho1_df <- sapply(1:i[length(i)],function(a){
  runif(n=j[length(j)], min=-1, max=1)
}) #Es considera un Dataset amb el màxim de dades i després s'extreuen
les que interessin a cada punt del rang a testar

fm_rOLS_Ho1 <- formula(Ho1 ~ 1 + Error(Lot))

HeatMapRange_OLS("Ho1",DS_Ho1_df, fm_rOLS_Ho1)

fm_ML_Ho1 <- formula(Ho1 ~ 1)
re_RI <- formula(~1|Lot)
meMLE <- "ML"

HeatMapRange_ML("Ho1",DS_Ho1_df, fm_ML_Ho1, re_RI, meMLE, 0)

meREML <- "REML"
HeatMapRange_ML("Ho1",DS_Ho1_df, fm_ML_Ho1, re_RI, meREML, 0)
```

```{r HeatMaps Ho2}
set.seed(7982)

DS_Ho2_df <- sapply(1:i[length(i)],function(a){
  m_lot <- runif(1, min=-1, max=1)
  runif(n=j[length(j)], min=m_lot-0.1, max=m_lot+0.1)
})

fm_rOLS_Ho2 <- formula(Ho2 ~ 1 + Error(Lot))

HeatMapRange_OLS("Ho2",DS_Ho2_df, fm_rOLS_Ho2)

fm_ML_Ho2 <- formula(Ho2 ~ 1)

HeatMapRange_ML("Ho2",DS_Ho2_df, fm_ML_Ho2, re_RI, meMLE, 0)

HeatMapRange_ML("Ho2",DS_Ho2_df, fm_ML_Ho2, re_RI, meREML, 0)
```

```

S'observa que per l'error estàndard residual sembla no haver un patró concret per tenir una variància residual més baixa o més alta possiblement pel fet de treballar amb una simulació de dades tipus núvol de punts en el cas Ho1 o els lots en posicions aleatòries en

Ho2. Sí que s'observa clarament com s'inverteix el comportament pel cas dels AIC i BIC segons el tipus de datset que S'està tractant:

+ Ho1: Al contrari del que habitualment s'esperaria, quan el model disminueix la mida tant en lots com en temps es calcula uns factors més baixos i per tant suposaria un millor ajust per les dades. Probablement es deu també al tipus de simulació de datset utilitzat on un major nombre de punts simplement afegeix més variància al tractar-se d'un núvol de punts.

+ Ho2: AIC i BIC disminueixen significativament quan s'augmenta el nombre de lots o de punts d'anàlisi possiblement degut a que un major nombre de mostres permet ajustar amb més precisió l'efecte aleatori provocat per la variació dels lots.

```
MODEL MIXT - DATASET HOMOGENI He
```

```
GENERACIÓ DATASET
```

S'aprofita la matriu base generada per generar les noves variables utilitzant sempre la mateixa llavor `7982`.

```
GENERACIÓ DATASETS He1/He2/He3 (HETEROGENI DEPENDENT DEL LOT)
```

En els dos datsets Ho s'assumeix la homogeneïtat residual (una de les assumpcions habituals per poder ajustar els models lineals), pel que és interessant generar datsets pel cas que no es compleixi aquesta condició. Es simula un dels casos més habituals que és l'augment de l'error estàndard en funció del temps. Per fer això es considera un rang de variància variable entre medicions de cada lot. Es té en compte 3 graus de variació:

- + Dataset He1 més simple on es depengui del temps multiplicat per 2.
- + Dataset He2 on depengui del temps al quadrat.
- + Dataset He3 més complex on intervingui una funció exponencial.

Per tots els casos es continua considerant una variació de les mitges de cada lot aleatòria entre un ordre superior de magnitud (-10 i 10). Per fer això es fa servir la funció `*runif*` combinada amb alguna de les funcions de la família `*apply*` per modificar el rang de cada lot i temps de manera seqüencial:

```
```{r He1 Dataset}
set.seed(7982)
Av_DS$He1 <- as.vector(sapply(1:av_i, function(i){
  m_lot <- runif(1, min=-10, max=10)
  sapply(1:av_j, function(j){
    runif(n=1, min=m_lot-(j*2), max=m_lot+(j*2))
  })
}))
kable(head(Av_DS[,c(colnames(Av_DS)[1:3],"He1")]), booktabs = T) %>%
kable_styling(latex_options = c("striped","condensed"))
```
```

```
```{r He2 Dataset}
set.seed(7982)
Av_DS$He2 <- as.vector(sapply(1:av_i, function(i){
  m_lot <- runif(1, min=-10, max=10)
  sapply(1:av_j, function(j){
    runif(n=1, min=m_lot-(j^2), max=m_lot+(j^2))
  })
}))
```



```
kable(head(Av_DS[,c(colnames(Av_DS)[1:3],"He2")]), booktabs = T) %>%
kable_styling(latex_options = c("striped","condensed"))
```

```

```
```{r He3 Dataset}
set.seed(7982)
Av_DS$He3 <- as.vector(sapply(1:av_i, function(i){
  m_lot <- runif(1, min=-10, max=10)
  sapply(1:av_j, function(j){
    runif(n=1, min=m_lot-(exp(j)), max=m_lot+(exp(j)))
  })
}))
kable(head(Av_DS[,c(colnames(Av_DS)[1:3],"He3")]), booktabs = T) %>%
kable_styling(latex_options = c("striped","condensed"))
```

```

## ## VISUALITZACIÓ DATASETS GENERATS He

Es visualitza gràficament les diferències entre els datasets generats. Per veure amb més claredat les diferències es visualitza primer mitjançant un gràfic tipus *\*scatterplot\** on es veu l'error estàndard global i la diferenciació per lot, i posteriorment es visualitza mitjançant els diagrames de caixes per apreciar més fàcilment els errors estàndards en el temps:

### ### \*SCATTERPLOTS\* He

es veu els datasets amb variació de variància en el temps:

```
```{r He Scatterplots}
labsHe_1 <- c("Dataset He1 - Factorx2","Dataset He2 - Factor
quadràtic","Dataset He3 - Factor exponencial")
labsHe_2 <- c("Dataset He1 s/ lot","Dataset He2 s/ lot","Dataset He3
s/ lot")

```

```
HeVars <- c("He1","He2","He3")
```

```
Av_DS_plotHe_sc1 <- lapply(1:length(HeVars),function(graph){
  ggplot(Av_DS, aes(x=Temps, y=Av_DS[,HeVars[graph]])) + geom_point()
+ geom_smooth(linetype="dashed") +
  labs(title=labsHe_1[graph], y = "Observacions")
})
```

```
Av_DS_plotHe_sc2 <- lapply(1:length(HeVars),function(graph){
  ggplot(Av_DS, aes(x=Temps, y=Av_DS[,HeVars[graph]], color=Lot,
group=Lot)) + geom_point() + geom_line() +
  labs(title=labsHe_2[graph], y = "Observacions")
})
```

```
ydens_DS_plotHe <- lapply(1:length(HeVars),function(graph){
  ggplot(Av_DS, aes(x=Av_DS[,HeVars[graph]])) +
  geom_density(color="darkblue", fill="lightblue") +
  geom_vline(aes(xintercept=mean(Av_DS[,HeVars[graph]])),
color="blue", linetype="dashed", size=1)+
  coord_flip() + labs(x = "Observacions")
})
```

```
grid.arrange(Av_DS_plotHe_sc1[[1]], ydens_DS_plotHe[[1]],
Av_DS_plotHe_sc1[[2]], ydens_DS_plotHe[[2]],
Av_DS_plotHe_sc1[[3]], ydens_DS_plotHe[[3]],ncol=2,
widths=c(4, 1))
```

```

grid.arrange(Av_DS_plotHe_sc2[[1]],          ydens_DS_plotHe[[1]],
Av_DS_plotHe_sc2[[2]], ydens_DS_plotHe[[2]],
              Av_DS_plotHe_sc2[[3]], ydens_DS_plotHe[[3]], ncol=2,
widths=c(4, 1))
` ``

```

Els gràfics de densitat ens mostren que al tenir més dispersió de dades hi ha una major concentració de punts en la mitja i que aquest efecte s'intensifica quan hi ha una progressió de variància més alta. Per la diferència d'escala i el tipus de gràfic no s'aprecia tant clar les diferències de variància, però sí que s'observa amb relativa claredat que la evolució en el temps és de diferent magnitud.

```

### *BOXPLOTS* He

```

es veu els datasets amb variació de variància en el temps:

```

` ``{r He Boxplots}
Av_DS_plotHe_bp1 <- lapply(1:length(HeVars),function(graph){
  ggplot(Av_DS, aes(x=N Temps, y=Av_DS[,HeVars[graph]])) +
    geom_boxplot()+ geom_dotplot(binaxis='y', stackdir='center',
dotsize=0.5)+
  labs(title=labsHe_1[graph], y = "Observacions")
})

```

```

lapply(1:3, function(ga){
  grid.arrange(Av_DS_plotHe_bp1[[ga]], ydens_DS_plotHe[[ga]],
              ncol=2, widths=c(4, 1))
})

```

```

` ``

```

S'observa clarament que en aquests 3 datasets hi ha un canvi de variància al augmentar el temps que és el que es buscava generar en la simulació.

```

## MODELS LINEALS He

```

Com en el cas de l'anàlisi de Ho es pretén generar un model lineal simple i un model lineal d'efectes mixtos amb la intercepció aleatòria, afegint per aquest cas models amb variables per explicar el canvi de variància en el temps.

```

### ML I/IS He

```

Es genera el model lineal simple i el model amb la variable temps per comparar-los i definir el model de base simple (s'aprofita la funció ja creada):

```

` ``{r ML I/IS He}
(Comp_I_IS_He1 <- Comp_I_IS_LM("He1",Av_DS))[1:3]

(Comp_I_IS_He2 <- Comp_I_IS_LM("He2",Av_DS))[1:3]

(Comp_I_IS_He3 <- Comp_I_IS_LM("He3",Av_DS))[1:3]
` ``

```

Al haver simulat una variància que varia amb el temps al voltant de la mateixa mitja s'esperaria que el predictor Temps no fos significatiu per explicar la mitja de la resposta. Tot i així, al tenir un nombre de lots limitat, provoca que l'augment de variabilitat emmascari la

mitja real sobre la qual es reparteixen i dona un fals efecte de tendència. És per aquest motiu probablement que s'observa que afegir la variable temps al model pren més significació al tenir un model amb un factor d'augment de variància més gran:

```
+ Model He1 amb factor multiplicador dona un contrast no significatiu pel Temps, però ja més significatiu que el model anterior homogeni (p-valors: `r round(Comp_I_IS_He1[[3]]$"Pr(>F)"[2],4)` (cas He1), `r round(Comp_I_IS_Ho1[[3]]$"Pr(>F)"[2],4)` (cas Ho1) i `r round(Comp_I_IS_Ho2[[3]]$"Pr(>F)"[2],4)` (cas Ho2)).
+ Model He1 no significatiu, però al límit de la significació si es pren una significació límit de 0.05: p-valor de `r round(Comp_I_IS_He2[[3]]$"Pr(>F)"[2],4)`.
+ Model He2 significatiu: p-valor de `r round(Comp_I_IS_He3[[3]]$"Pr(>F)"[2],4)`.
```

En aquest cas no seria tan senzill com descartar el model amb l'efecte fix de covariància del predictor Temps, però és interessant veure si aquest fals efecte de tendència en els propers models és necessari mantenir-lo o si es podrà explicar el model amb les variables d'efectes aleatoris i/o funcions de variància-covariància entre repeticions.

S'observa també com ha resultat l'error estimatiu i l'error estàndard residual del model segons el dataset :

Models I:

```
```{r ML I/IS He I resum}
I_HE_res <- data.frame(Dataset = HeVars)
I_HE_res$"Error estimació" <- round(c(Comp_I_IS_He1[[1]]
$coefficients[,2],
 Comp_I_IS_He2[[1]]
$coefficients[,2],Comp_I_IS_He3[[1]]$coefficients[,2]),4)
I_HE_res$"Error residual" <- round(c(Comp_I_IS_He1[[1]]$sigma,
 Comp_I_IS_He2[[1]]
$sigma,Comp_I_IS_He3[[1]]$sigma),4)

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

Models IS:

```
```{r ML I/IS He IS resum}
IS_HE_res <- data.frame(Dataset = HeVars)
IS_HE_res$"Error estimació intercepció" <- round(c(Comp_I_IS_He1[[2]]
$coefficients[1,2],
 Comp_I_IS_He2[[2]]
$coefficients[1,2],Comp_I_IS_He3[[2]]$coefficients[1,2]),4)
IS_HE_res$"Error estimació predictor" <- round(c(Comp_I_IS_He1[[2]]
$coefficients[2,2],
 Comp_I_IS_He2[[2]]
$coefficients[2,2],Comp_I_IS_He3[[2]]$coefficients[2,2]),4)
IS_HE_res$"Error residual" <- round(c(Comp_I_IS_He1[[2]]$sigma,
 Comp_I_IS_He2[[2]]
$sigma,Comp_I_IS_He3[[2]]$sigma),4)

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

L'estimació i errors residuals augmenten a mesura que s'estudien models més dispersos ja que van lligats a la incertesa de predicció de les dades, però el més interessant en aquestes últimes taules és veure que tot i augmentar la significació del predictor, aquest no té la força suficient per provocar un canvi molt significatiu si es comparen els errors estàndards residuals dels mateixos datasets en models amb i sense la pendent S, cosa que ens donaria pistes en cas que fos un dataset desconegut de que hi ha algun factor adicional amagat en el model.

```
### ML RI He
```

Avançant a partir del model simple amb intercepció ML I es fa l'ajust de models amb efecte aleatori (s'utilitza la funció generada). Es genera una funció adicional per fer el mateix a partir de la funció de model simple ML IS per veure també aquest ajust.

```
```{r ML IS RI funció}
Comp_IS_RI_LM <- function(var,ds) {
 fm1 <- formula(paste(var,"~","Temps","+","Error(Lot)"))
 ML_IS_RI_sOLS <- summary(ML_IS_RI_OLS <- aov(fm1, data = ds))
 ML_IS_RI_OLS
 ML_IS_RI_OLS_err <- sigma(summary.lm(ML_IS_RI_OLS$Within))

 fm2 <- formula(paste(var,"~","Temps"))

 ML_IS_RI_sMLE <- summary(ML_IS_RI_MLE <- lme(fm2, random = ~1|Lot,
data = ds, method="ML"))

 ML_IS_RI_sREML <- summary(ML_IS_RI_REML <- lme(fm2,random = ~1|Lot,
data = ds, method="REML"))

 list(ML_IS_RI_OLS$Within,ML_IS_RI_sMLE,ML_IS_RI_sREML,ML_IS_RI_OLS,ML_
IS_RI_MLE,ML_IS_RI_REML)
 names(l) <- c("OLS IS RI","MLE IS RI","REML IS RI","OLS IS RI
base","MLE IS RI base","REML IS RI base")

 return(l)
}...

```{r ML I RI He}

(Comp_I_RI_He1 <- Comp_I_RI_LM("He1",Av_DS))[1:3]

fmHe1 <- formula(paste("He1","~",1))
(eMLE_I_He1<-exactRLRT(lme(fmHe1, random = ~1|Lot, data = Av_DS,
method="ML")))
(eREML_I_He1<-exactRLRT(lme(fmHe1, random = ~1|Lot, data = Av_DS,
method="REML")))

(Comp_I_RI_He2 <- Comp_I_RI_LM("He2",Av_DS))[1:3]

fmHe2 <- formula(paste("He2","~",1))
(eMLE_I_He2<-exactRLRT(lme(fmHe2, random = ~1|Lot, data = Av_DS,
method="ML")))
(eREML_I_He2<-exactRLRT(lme(fmHe2, random = ~1|Lot, data = Av_DS,
method="REML")))
```

```

(Comp_I_RI_He3 <- Comp_I_RI_LM("He3",Av_DS))[1:3]

fmHe3 <- formula(paste("He3","~",1))
(eMLE_I_He3<-exactRLRT(lme(fmHe3, random = ~1|Lot, data = Av_DS,
method="ML")))
(eREML_I_He3<-exactRLRT(lme(fmHe3, random = ~1|Lot, data = Av_DS,
method="REML")))
```



```

```{r ML IS RI He}

(Comp_IS_RI_He1 <- Comp_IS_RI_LM("He1",Av_DS))[1:3]

fmHe1_S <- formula(paste("He1","~","Temps"))
(eMLE_IS_He1<-exactRLRT(lme(fmHe1_S, random = ~1|Lot, data = Av_DS,
method="ML")))
(eREML_IS_He1<-exactRLRT(lme(fmHe1_S, random = ~1|Lot, data = Av_DS,
method="REML")))

(Comp_IS_RI_He2 <- Comp_IS_RI_LM("He2",Av_DS))[1:3]

fmHe2_S <- formula(paste("He2","~","Temps"))
(eMLE_IS_He2<-exactRLRT(lme(fmHe2_S, random = ~1|Lot, data = Av_DS,
method="ML")))
(eREML_IS_He2<-exactRLRT(lme(fmHe2_S, random = ~1|Lot, data = Av_DS,
method="REML")))

(Comp_IS_RI_He3 <- Comp_IS_RI_LM("He3",Av_DS))[1:3]

fmHe3_S <- formula(paste("He3","~","Temps"))
(eMLE_IS_He3<-exactRLRT(lme(fmHe3_S, random = ~1|Lot, data = Av_DS,
method="ML")))
(eREML_IS_He3<-exactRLRT(lme(fmHe3_S, random = ~1|Lot, data = Av_DS,
method="REML")))
```

```


```

Es genera una taula resum dels resultats obtinguts del test RLRT per començar:

```

```{r ML I/IS RI He I resum RLRT}
I_IS_RI_HE_RLRT <- data.frame(Dataset =
paste(rep(paste(rep(HeVars,each=2),c("I RI","IS RI")),each=3),c("Clàssic OLS","MLE","REML")))

Comp_RI_He_2 <-
list("-",eMLE_I_He1,eREML_I_He1,"-",eMLE_IS_He1,eREML_IS_He1,
" ",eMLE_I_He2,eREML_I_He2,"-",eMLE_IS_He2,eREML_
IS_He2,
" ",eMLE_I_He3,eREML_I_He3,"-",eMLE_IS_He3,eREML_
IS_He3)
I_IS_RI_HE_RLRT$"p-valor" <- RLRT"
sapply(1:length(Comp_RI_He_2),function(l1){
  if (class(Comp_RI_He_2[[l1]])!="htest"){
    return("-")
  } else {
    round(Comp_RI_He_2[[l1]]$p.value,4)
  }
})

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

```

...

Es podria considerar una mida mostral correcte la d'aquest model base amb `r nrow(Av_DS)` observacions, i entre 1 i 2 efectes fixos (si es considera la mitja com un efecte fix). En aquests resultats com a comentari general es pot apreciar que:

- + No sembla haver-hi diferències significatives entre mètodes (MLE; REML).
- + En el cas de la utilització de la covariable temps semblaria que al afegir-la disminueix el p-valor calculat de significació de l'efecte aleatori (és a dir, es tornaria més significatiu).
- + Entre els tres conjunts de dades donaria la impressió que com més disperses es troben les dades menys significatiu es torna l'efecte aleatori.

ML RI Weights He

```
```{r FuncLists}
#Es genera una funció amb la simple instrucció de quan es facin
l·listats dintre l·listats converteixi tot en una l·lista 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)
}
```
```

S'observa els gràfics de diagnòstic de residus dels models amb la covariable temps generats pels conjunts He:

```
```{r Diag Homog Res He}
Diag_Hom_He1 <- list(Comp_I_IS_He1[5],Comp_IS_RI_He1[5:6])

Diag_Hom_lHe1 <- ConvList(Diag_Hom_He1)
lapply(1:length(Diag_Hom_lHe1),function(l){
 plot(Diag_Hom_lHe1[[l]],which=1,main=paste("He1",names(Diag_Hom_lHe1)
)[l],sep="--"))
})

Diag_Hom_He2 <- list(Comp_I_IS_He2[5],Comp_IS_RI_He2[5:6])

Diag_Hom_lHe2 <- ConvList(Diag_Hom_He2)
lapply(1:length(Diag_Hom_lHe2),function(l){
 plot(Diag_Hom_lHe2[[l]],which=1,main=paste("He2",names(Diag_Hom_lHe2)
)[l],sep="--"))
})
```
```

```

Diag_Hom_He3 <- list(Comp_I_IS_He3[5],Comp_IS_RI_He3[5:6])

Diag_Hom_lHe3 <- ConvList(Diag_Hom_He3)
lapply(1:length(Diag_Hom_lHe3),function(l){
  plot(Diag_Hom_lHe3[[l]],which=1,main=paste("He3",names(Diag_Hom_lHe3
)[l],sep="--"))
})
````

```

Com estava previst en la simulació s'observa heterogeneïtat de residus (amb més claredat com més dispers és el conjunt de dades).

En aquest apartat s'avança amb el conjunts \*He\* al intentar aplicar i veure com es comporta el model amb les funcions per modular la variància de l'error en els casos on ja no es considera homogeneïtat residual. A nivell de programació és complicat afegir aquests paràmetres de pes als objectes `aov` pel que tot i que el model ajustat per ANOVA dona més bons resultats que els ML, es veurà les funcions per Heteroscedasticitat aplicat als models ajustats per màxima versemblança més preparats per l'aplicació d'aquestes funcions a nivell d'algoritmes computacionals:

```

````{r ML Weights funció,error=TRUE}
Comp_W_LM <- function(fML,fRE,ds,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 <- lme(fML, random = fRE, control=ctrl, data =
ds,method="ML")
  baseREML <- lme(fML, random = fRE, control=ctrl, data =
ds,method="REML")
  l <- list()

  if (paste(fML[3])=="1"){
    fMLk <- "I"
  } else if (paste(fML[3])=="Temps"){
    fMLk <- "IS"
  } else {
    fMLk <- "desconegut"
  }

  if (paste(fRE[2])=="1 | Lot"){
    fREk <- "RI"
  } else if (paste(fRE[2])=="1 + Temps | Lot"){
    fREk <- "RIS"
  } else if (class(fRE)=="list"){
    fREk <- "DRIS"
  } else {
    fREk <- "desconegut"
  }

  if(WVI==1 | WVI==3){
    ML_I_RI_WVI_sMLE <- summary(ML_I_RI_WVI_MLE <- update(baseML,
weights = varIdent(form = ~Temps)))
    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 = ~Temps)))
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 = ~Temps)))
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 = ~Temps)))
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)
}...

```

Pel moment només s'aplica la funció sense la covariant en la fórmula base del model per avançar d'una manera lògica en l'anàlisi de la simulació.

```

```{r ML_I_RI_W_He,error=TRUE}
fML_I_He1 <- formula(He1 ~ 1)
fRE_RI <- formula(~1|Lot)
(Comp_I_RI_W_He1 <- Comp_W_LM(fML_I_He1, fRE_RI, Av_DS, 3, 3, 3))[1:3]
fML_I_He2 <- formula(He2 ~ 1)
(Comp_I_RI_W_He2 <- Comp_W_LM(fML_I_He2, fRE_RI, Av_DS, 3, 3, 2))[1:3]
fML_I_He3 <- formula(He3 ~ 1)

```

```

(Comp_I_RI_W_He3 <- Comp_W_LM(fML_I_He3, fRE_RI, Av_DS, 3, 3, 1))[1:3]
```

```{r ML IS RI W He, error=TRUE}
fML_IS_He1 <- formula(He1 ~ Temps)
fRE_RI <- formula(~1|Lot)
(Comp_IS_RI_W_He1 <- Comp_W_LM(fML_IS_He1, fRE_RI, Av_DS, 3, 3, 1))[1:3]
fML_IS_He2 <- formula(He2 ~ Temps)
(Comp_IS_RI_W_He2 <- Comp_W_LM(fML_IS_He2, fRE_RI, Av_DS, 3, 3, 3))[1:3]
fML_IS_He3 <- formula(He3 ~ Temps)
(Comp_IS_RI_W_He3 <- Comp_W_LM(fML_IS_He3, fRE_RI, Av_DS, 3, 3, 1))[1:3]
```

```{r ML IS RIS W He, error=TRUE}
fML_IS_He1 <- formula(He1 ~ Temps)
fRE_RIS <- formula(~1 + Temps|Lot)
(Comp_IS_RIS_W_He1 <- Comp_W_LM(fML_IS_He1, fRE_RIS, Av_DS, 3, 3, 0))[1:3]
fML_IS_He2 <- formula(He2 ~ Temps)
(Comp_IS_RIS_W_He2 <- Comp_W_LM(fML_IS_He2, fRE_RIS, Av_DS, 3, 3, 0))[1:3]
fML_IS_He3 <- formula(He3 ~ Temps)
(Comp_IS_RIS_W_He3 <- Comp_W_LM(fML_IS_He3, fRE_RIS, Av_DS, 3, 3, 3))[1:3]
```

RESUM I DIAGNÒSTIC MODELS He

En el codi resum que es mostra a continuació s'han realitzat varies generacions de llistats i de gràfics segons els resultats que s'han anat obtenint tenint en compte els tres indicadors de qualitat. Es comenten al final ja que les conclusions són similars pels 3 conjunts.

```{r Resum ML He1}
#He1 sense funció de variància
He1_R0 <- list(Comp_I_IS_He1[4:5], Comp_I_RI_He1[1:3], Comp_IS_RI_He1[1:3])
He1_R <- ConvList(He1_R0)
ML_Resum(He1_R, "He1")

#He1 funcions var1
He1_RWa0 <- list(Comp_I_RI_W_He1[c(1, 3, 5, 7, 9, 11)])
He1_RWa <- ConvList(He1_RWa0)
ML_Resum(He1_RWa, "He1")

#He1 funcions var2
He1_RWb0 <- list(Comp_IS_RI_W_He1[c(1, 3, 5, 7, 9)])
He1_RWb <- ConvList(He1_RWb0)
ML_Resum(He1_RWb, "He1")

#He1 funcions var3
He1_RWc0 <- list(Comp_IS_RIS_W_He1[c(1, 3, 5, 7)])
He1_RWc <- ConvList(He1_RWc0)
ML_Resum(He1_RWc, "He1")

He1_RGW0 <- list(He1_R[c(3, 6)], He1_RWa[c(4, 6)], He1_RWb[c(4, 5)], He1_RWc[c(3, 4)])
He1_RGW <- ConvList(He1_RGW0)
ML_Resum(He1_RGW, "He1")
```

```{r Resum ML He2}
#He2 sense funció de variància

```

```

He2_R0
list(Comp_I_IS_He2[4:5], Comp_I_RI_He2[1:3], Comp_IS_RI_He2[1:3])
He2_R <- ConvList(He2_R0)
ML_Resum(He2_R, "He2")

#He2 funcions var
He2_RWa0 <- list(Comp_I_RI_W_He2[c(1,3,5,7,11)])
He2_RWa <- ConvList(He2_RWa0)
ML_Resum(He2_RWa, "He2")

#He2 funcions var
He2_RWb0 <- list(Comp_IS_RI_W_He2[c(1,3,5,7,9,11)])
He2_RWb <- ConvList(He2_RWb0)
ML_Resum(He2_RWb, "He2")
He2_RWbb0 <- list(Comp_IS_RI_W_He2[c(9,11)])
He2_RWbb <- ConvList(He2_RWbb0)
ML_Resum(He2_RWbb, "He2")

#He2 funcions var3
He2_RWc0 <- list(Comp_IS_RIS_W_He2[c(1,3,5,7)])
He2_RWc <- ConvList(He2_RWc0)
ML_Resum(He2_RWc, "He2")

He2_RGW0
list(He2_R[c(3,6)], He2_RWa[c(4,5)], He2_RWb[c(4,6)], He2_RWc[c(3,4)])
He2_RGW <- ConvList(He2_RGW0)
ML_Resum(He2_RGW, "He2")
```



```

```{r Resum ML He3}
#He3 sense funció de variància
He3_R0
list(Comp_I_IS_He3[4:5], Comp_I_RI_He3[1:3], Comp_IS_RI_He3[1:3])
He3_R <- ConvList(He3_R0)
ML_Resum(He3_R, "He3")

#He3 funcions var
He3_RWa0 <- list(Comp_I_RI_W_He3[c(1,3,5,7,9)])
He3_RWa <- ConvList(He3_RWa0)
ML_Resum(He3_RWa, "He3")
He3_RWaa0 <- list(Comp_I_RI_W_He3[c(5,7,9)])
He3_RWaa <- ConvList(He3_RWaa0)
ML_Resum(He3_RWaa, "He3")

#He3 funcions var b
He3_RWb0 <- list(Comp_IS_RI_W_He3[c(1,3,5,7,9)])
He3_RWb <- ConvList(He3_RWb0)
ML_Resum(He3_RWb, "He3")
He3_RWbb0 <- list(Comp_IS_RI_W_He3[c(5,7,9)])
He3_RWbb <- ConvList(He3_RWbb0)
ML_Resum(He3_RWbb, "He3")

#He3 funcions var3
He3_RWc0 <- list(Comp_IS_RIS_W_He3[c(1,3,5,7,9,11)])
He3_RWc <- ConvList(He3_RWc0)
ML_Resum(He3_RWc, "He3")
He3_RWcc0 <- list(Comp_IS_RIS_W_He3[c(5,7,9,11)])
He3_RWcc <- ConvList(He3_RWcc0)
ML_Resum(He3_RWcc, "He3")

```


```

```

He3_RGW0
list(He3_R[c(3,6)],He3_RWa[c(4,5)],He3_Rwb[c(4,5)],He3_Rwc[c(4,6)])
He3_RGW <- ConvList(He3_RGW0)
ML_Resum(He3_RGW,"He3")
He3_RGWb0 <- list(He3_RWa[c(4,5)],He3_Rwb[c(4,5)],He3_Rwc[c(4,6)])
He3_RGWb <- ConvList(He3_RGWb0)
ML_Resum(He3_RGWb,"He3")
```

```

Els models sense la funció de l'error estàndard i amb efectes aleatoris de l'intercepció són similars entre ells. Al afegir al model la funció de modificació de l'error estàndard s'observa que es redueix significativament l'error estàndard residual, sobretot quan s'apliquen la funció `varPower` juntament amb el model fixe que conté la covariable temps que sembla el model on es redueix més l'error estàndard residual (concretament als models amb la funció varPower es nota la diferència al afegir la variable temps a la part d'efectes fixos en relació al model on només compté la intercepció com efecte fix).

Pel que fa als indicadors AIC/BIC no hi ha realment una millora substancial al anar augmentant la complexitat del model encara que sí que denota una lleugera reducció (i per tant millora en la bondat d'ajust) en general al afegir les funcions de modulació de variància.

Segons el conjunt no s'ha pogut aplicar la `varPower` a totes les variants del model ja que en algun cas s'arribava a resultats inconsistents (cal recordar que al aplicar algorismes d'aquesta complexitat matemàtica segons com es distribueixen les nostres dades es pot no arribar a una solució matemàticament lògica), tot i així no es denota una millora significativa al modificar els efectes aleatoris de la intercepció per efectes aleatoris de pendent i intercepció en aquests casos, que ja quadraria amb els conjunts simulats. S'entén que la importància d'afegir la pendent en els efectes fixos ve com en el conjunt Ho2 pel fet que la pròpia dispersió al haver un nombre reduït de lots es falseja en forma de pendent segons com s'hagi produït la dispersió.

En aquest tipus de conjunts de dades es pot extreure que es considera important aplicar la funció de modular l'error estàndard i en cas necessari per compensar la mida dels lots aplicar el factor de la pendent, tot i que en aquest especte s'ha d'anar amb compte si es vol utilitzar com un model de predicció per futurs lots.

Prenent el model més efectiu com el Model IS MLE/REML RI VarPower s'extreuen les matrius internes d'interès per analitzar-lo amb més detall:

```

```{r D Matrix He}
He_B
list(Comp_IS_RI_W_He1[[10]],Comp_IS_RI_W_He2[[10]],Comp_IS_RI_W_He3[[10]])
names(He_B) <- c(names(Comp_IS_RI_W_He1)[10],names(Comp_IS_RI_W_He2)[10],names(Comp_IS_RI_W_He3)[10])

lapply(1:length(He_B),function(B){
  VarCorr(He_B[[B]])
})

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

```

```

})

lapply(1:length(He_B),function(B){
 a<-getVarCov(He_B[[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 l'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 l'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 l'variància pròpia del temps (i si el conjunt és més dispers també més variància es calcula).

+ La tercera sèria de matrius mostra:

+ La matriu primera de cada punt de la sèria és de tipus marginal on es suma l'variància residual i l'variància degut a l'efecte aleatori en la diagonal, mentre que l'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 l'variància total dins un dels individus (equivalent per tots els altres) que té l'variància de l'efecte aleatori sumat a l'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 l'variància dins el mateix lot.

+ 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).

Es visualitzen els gràfics de diagnòstic del model:

```

```{r Diag Models He}
He_Diag0 <- list(Comp_I_IS_He1[5],Comp_I_IS_He2[5],Comp_I_IS_He3[5],
               Comp_IS_RI_W_He1[10],Comp_IS_RI_W_He2[10],Comp_IS_RI_W_
               _He3[10])

```

```

He_Diag <- ConvList(He_Diag0)

```

```

names(He_Diag) <- paste(HeVars,names(He_Diag))

```

```

lapply(1:length(He_Diag),function(QQ){
  if (class(He_Diag[[QQ]])=="lme") {
    list(qqnorm(He_Diag[[QQ]],abline = c(0,1),main=names(He_Diag)
[QQ]),plot(He_Diag[[QQ]])
  } else {
    list(plot(He_Diag[[QQ]],which=1,main=names(He_Diag)
[QQ]),plot(He_Diag[[QQ]],which=2,main=names(He_Diag)[QQ]))
  }
})

```

...

Es denota com a mesura que augmenta la dispersió també hi ha una desviació més pronunciada de la normalitat que comença semblant només petit efecte de cua llarga (long tails) i que es va acusant a mesura que es pren conjunts més dispersos de dades. Probablement és un dels motius de l'anterior indicador de pitjor ajust del model MLE en comparació al clàssic OLS.

També s'observa com al aplicar el model amb la funció moduladora de l'error estàndard sembla que es corregeix bastant l'efecte d'heterogeneïtat que es veia en els residus així com la desviació de la normalitat.

```
## PROVES DE VARIACIÓ DE LOTS/TEMPS He
```

S'utilitzen les fórmules ja generades per graficar el heatmap de variació de variants sobre el model trobat com a òptim dins les proves realitzades:

```
```{r Heatmaps He1}
set.seed(7982)
i <- seq(i3,i10)
j <- seq(j4,j10)
```

```
DS_He1_df <- sapply(1:i[length(i)],function(a){
 m_lot <- runif(1, min=-10, max=10)
 sapply(1:j[length(j)], function(j){
 runif(n=1, min=m_lot-(j*2), max=m_lot+(j*2))
 })
})
```

#Es considera un Dataset amb el màxim de dades i després s'extreuen les que interessin a cada punt del rang a testar

```
fm_ML_He1 <- formula(He1 ~ 1 + Temps)
re_RI <- formula(~1|Lot)
fm_W <- formula(~Temps)
```

```
meREML <- "REML"
HeatMapRange_ML("He1",DS_He1_df, fm_ML_He1, re_RI, meREML, 2, fm_W)
```
```

```
```{r Heatmaps He2}
set.seed(7982)
DS_He2_df <- sapply(1:i[length(i)],function(a){
 m_lot <- runif(1, min=-10, max=10)
 sapply(1:j[length(j)], function(j){
 runif(n=1, min=m_lot-(j^2), max=m_lot+(j^2))
 })
})
```

#Es considera un Dataset amb el màxim de dades i després s'extreuen les que interessin a cada punt del rang a testar

```
fm_ML_He2 <- formula(He2 ~ 1 + Temps)
fm_W <- formula(~Temps)
```

```
meREML <- "REML"
HeatMapRange_ML("He2",DS_He2_df, fm_ML_He2, re_RI, meREML, 2, fm_W)
```
```

```

```{r Heatmaps He3}
set.seed(7982)
DS_He3_df <- sapply(1:i[length(i)],function(a){
 m_lot <- runif(1, min=-10, max=10)
 sapply(1:j[length(j)], function(j){
 runif(n=1, min=m_lot-(exp(j)), max=m_lot+(exp(j)))
 })
})
```

```

#Es considera un Dataset amb el màxim de dades i després s'extreuen les que interessin a cada punt del rang a testar

```

fm_ML_He3 <- formula(He3 ~ 1 + Temps)
fm_W <- formula(~Temps)

```

```

meREML <- "REML"
HeatMapRange_ML("He3", DS_He3_df, fm_ML_He3, re_RI, meREML, 2, fm_W)
```

```

Per generar els *heatmaps* de comparació entre la variació de lots i temps d'anàlisi s'ha utilitzat el segon model més eficient que s'ha trobat en els anteriors càlculs degut a que en el cas de la funció *varPower*, al calcular un nombre tan gran de datasets, en alguns d'ells no troba convergència en les iteracions per generar un resultat de la funció de variància. La funció utilitzada ha estat la de *varExp* que a efectes de comprovar els efectes de variar aquests 2 paràmetres ens pot servir de símil correctament.

A part de les diferències ja vistes entre els 3 datasets que provoquen l'augment de dispersió de dades en aquest cas s'observa que:

+ Pels paràmetres indicatius de la qualitat del model AIC i BIC sembla que al tenir més lots/Temps augmenta aquest paràmetre disminuint l'eficàcia del model. Possiblement l'efecte de dispersió augmenta al afegir més lots i temps i per tant fa més difícil l'ajust d'un model eficient amb els recursos que s'han utilitzat fins ara.

+ El paràmetre de l'error estàndard residual semblaria que segueix una patró similar a AIC i BIC, però és curiós com sembla que pren més importància el paràmetre del nombre de temps. És a dir al variar el nombre de lots sembla no haver un rang molt gran de canvi mentre que en el nombre de temps sembla haver-hi dos zones molt diferenciades amb sigmes alts i baixos.

Tot i que en aquest apartat es podria treure la conclusió que és millor tenir un nombre més limitat de lots i temps, en la realitat sempre és millor disposar del màxim de dades possibles ja que habitualment les dades d'un mateix lot no es comportaran amb una dispersió tan exagerada com s'ha simulat en aquests models. Si es donés aquest cas amb aquest diagnòstic, potser s'hauria de començar a sospitar que hi ha algun factor que no s'ha tingut en compte o algun error en les dades.

# MODEL MIXT - DATASET AMB TENDÈNCIA

## GENERACIÓ DATASET

Per la generació de datasets s'utilitza sempre la mateixa llavor essent aquesta `7982` i el dataset base `Av\_DS` com a referència sobre el qual es van generant les noves variables.

### ### GENERACIÓ DATASET So1 (TENDÈNCIA AMB RECTES PARALELES CENTRADES)

Es generen punts seguint una tendència concreta amb els punts centrats en una recta concreta amb poca dispersió (a efectes d'escalat al voltant de 1 d'una recta amb pendent -1) mitjançant la funció \*runif\*:

```
```{r So1 Dataset}
set.seed(7982)
Av_DS$So1 <- as.vector(sapply(1:av_i,function(i){
  sapply(1:av_j, function(j){
    runif(n=1, min=(TA[length(TA)]-TA[j])-1, max=(TA[length(TA)]-
TA[j])+1) #Observacions constant de rang petit entre -1 i 1 per
simular que no hi Ha tendències ni diferències
  })
})
)

kable(head(Av_DS[,c(colnames(Av_DS)[1:3],"So1")]), booktabs = T) %>%
kable_styling(latex_options = c("striped","condensed"))
```
```

### ### GENERACIÓ DATASET So2 (TENDÈNCIA AMB RECTES PARALELES DISPERSSES)

El mateix cas que l'anterior, però suposant que hi ha dispersió més gran entre els lots (aquesta simulació seria un dels casos més habituals en situació ideal que es dona en els estudis d'estabilitat si el disseny és correcte):

```
```{r So2 Dataset}
set.seed(7982)
Av_DS$So2 <- as.vector(sapply(1:av_i, function(i){
  m_lot <- runif(1, min=TA[length(TA)]-30, max=TA[length(TA)]+30)
  sapply(1:av_j,function(j){
    runif(n=1, min=(m_lot-TA[j])-1, max=(m_lot-TA[j])+1)
  })
})
)

kable(head(Av_DS[,c(colnames(Av_DS)[1:3],"So2")]), booktabs = T) %>%
kable_styling(latex_options = c("striped","condensed"))
```
```

### ### GENERACIÓ DATASET So3 (TENDÈNCIA AMB RECTES NO PARALELES DISPERSSES)

En aquest cas es considera certa dispersió en la intercepció dels lots i una variabilitat en la pendent segons el lot dins un rang definit (aquesta simulació seria un dels altres casos habituals en els estudis d'estabilitat en les fases de testatge de possible combinació de lots):

```
```{r So3 Dataset}
set.seed(7982)
Av_DS$So3 <- as.vector(sapply(1:av_i, function(i){
  m_lot <- runif(1, min=TA[length(TA)]-10, max=TA[length(TA)]+10)
  s_lot <- runif(1, min=0.1, max=2)
  sapply(1:av_j,function(j){
    runif(n=1, min=(m_lot-(s_lot*TA[j]))-1, max=(m_lot-(s_lot*TA[j]))
+1)
  })
})
)
```
```



```

)

kable(head(Av_DS[,c(colnames(Av_DS)[1:3],"So3")]), booktabs = T) %>%
kable_styling(latex_options = c("striped","condensed"))
```

## VISUALITZACIÓ DATASETS GENERATS So

Es visualitza gràficament les diferències entre els datasets generats. Per veure amb més claredat les diferències es visualitza primer mitjançant un gràfic tipus *scatterplot* on es veu l'error estàndard global i la diferenciació per lot, i posteriorment es visualitza mitjançant els diagrames de caixes per apreciar més fàcilment per on es mouen les dades:

### *SCATTERPLOTS* So

es veu els datasets amb variància homogènia.

```{r So Scatterplots}
labsSo_1 <- c("Dataset So1 - Rectes paraleles centrades","Dataset So2
- Rectes paraleles disperses","Dataset Soe - Rectes no paraleles
disperses")
labsSo_2 <- c("Dataset So1 s/ lot","Dataset So2 s/ lot","Dataset So3
s/ lot")

SoVars <- c("So1","So2","So3")

Av_DS_plotSo_sc1 <- lapply(1:length(SoVars),function(graph){
 ggplot(Av_DS, aes(x=Temps, y=Av_DS[,SoVars[graph]])) + geom_point()
+ geom_smooth(linetype="dashed") +
 labs(title=labsSo_1[graph], y = "Observacions")
})

Av_DS_plotSo_sc2 <- lapply(1:length(SoVars),function(graph){
 ggplot(Av_DS, aes(x=Temps, y=Av_DS[,SoVars[graph]], color=Lot,
group=Lot)) + geom_point() + geom_line() +
 labs(title=labsSo_2[graph], y = "Observacions")
})

ydens_DS_plotSo <- lapply(1:length(SoVars),function(graph){
 ggplot(Av_DS, aes(x=Av_DS[,SoVars[graph]])) +
 geom_density(color="darkblue", fill="lightblue") +
 geom_vline(aes(xintercept=mean(Av_DS[,SoVars[graph]])),
color="blue", linetype="dashed", size=1)+
 coord_flip() + labs(x = "Observacions")
})

grid.arrange(Av_DS_plotSo_sc1[[1]], ydens_DS_plotSo[[1]],
Av_DS_plotSo_sc1[[2]], ydens_DS_plotSo[[2]],
Av_DS_plotSo_sc1[[3]], ydens_DS_plotSo[[3]],ncol=2,
widths=c(4, 1))

grid.arrange(Av_DS_plotSo_sc2[[1]], ydens_DS_plotSo[[1]],
Av_DS_plotSo_sc2[[2]], ydens_DS_plotSo[[2]],
Av_DS_plotSo_sc2[[3]], ydens_DS_plotSo[[3]],ncol=2,
widths=c(4, 1))
```

```

S'ha intentat diferenciar entre els lots centrats amb certa dispersió (mínima en comparació a la magnitud de la tendència), el conjunt on

realment hi ha diferències en la intercepció, però amb mateixa pendent i l'últim conjunt on hi ha diferències de pendents.

```
### *BOXPLOTS* So
```

es veu *boxplots* dels datasets.

```
```{r So Boxplots}
Av_DS_plotSo_bp1 <- lapply(1:length(SoVars),function(graph){
 ggplot(Av_DS, aes(x=NTemps, y=Av_DS[,SoVars[graph]])) +
 geom_boxplot()+ geom_dotplot(binaxis='y', stackdir='center',
dotsize=1)+
 labs(title=labsSo_1[graph], y = "Observacions")
})

grid.arrange(Av_DS_plotSo_bp1[[1]], ydens_DS_plotSo[[1]],
Av_DS_plotSo_bp1[[2]], ydens_DS_plotSo[[2]],
Av_DS_plotSo_bp1[[3]], ydens_DS_plotSo[[3]],
...,
ncol=2, widths=c(4, 1))
```

Es reflexa amb claredat en quin cas els lots estan centrats i en quin no.

```
MODELS LINEALS So
```

Per l'ajust dels models lineals dels datasets en aquest cas es consideren diversos models diferents per comparar entre ells. Es pretén generar un model lineal simple, un model lineal d'efectes mixtos amb la intercepció aleatòria i també el d'efectes aleatoris de pendent i intercepció. És important tenir en compte que els datasets generats s'han de considerar independents entre ells i per tant no són directament comparables al ajustar els models, pel que l'anàlisi que es segueix és veure les diferències entre l'ajust dels diferents models a un mateix dataset i posteriorment es compara els diferents datasets, però en termes de la bondat d'ajust dels models aplicats:

```
ML I/IS So
```

Es genera el model lineal simple i el model amb la variable temps per comparar-los i definir el model de base simple amb la funció creada per aquest fi:

```
```{r ML I/IS So}
(Comp_I_IS_So1 <- Comp_I_IS_LM("So1",Av_DS))[1:3]

(Comp_I_IS_So2 <- Comp_I_IS_LM("So2",Av_DS))[1:3]

(Comp_I_IS_So3 <- Comp_I_IS_LM("So3",Av_DS))[1:3]
```
```

Per començar s'observa que, com és d'esperar, afegir la variable temps al model és una diferència significativa i una suposada millora al model per explicar la resposta ja que el resultat de l'Anova corresponent entre models dona un p-valor de `round(Comp_I_IS_So1[[3]]$"Pr(>F)"[2],4)` (cas So1), `round(Comp_I_IS_So2[[3]]$"Pr(>F)"[2],4)` (cas So2) i `round(Comp_I_IS_So3[[3]]$"Pr(>F)"[2],4)` (cas So3), pel que per regla general sempre és preferible utilitzar el model més simple possible que no afegeix tants factors redundants que poden fer augmentar la variabilitat.

En quant a l'error estàndard residual s'observa a primera vista que és més baix en el cas de So1 ja que és el conjunt amb menys variabilitat i més fàcil de definir en general:

```
+ So1: `r round(Comp_I_IS_So1[[2]]$sigma,4)` (Error residual)
+ So2: `r round(Comp_I_IS_So2[[2]]$sigma,4)` (Error residual)
+ So3: `r round(Comp_I_IS_So2[[2]]$sigma,4)` (Error residual)
```

### ML LotxTemps So

En el cas concret de tenir gràfiques amb pendent i possibles combinacions de lot, una de les proves habituals també és la comprovació de si en un model simple l'efecte pendent amb lot és significatiu. És a dir si hi ha equivalència de lots en la pendent o no:

```
```{r ML LotxTemps}
(anova(Creuat_IS_So1 <- lm(So1 ~ 1 + Temps*Lot,data=Av_DS))
(anova(Creuat_IS_So2 <- lm(So2 ~ 1 + Temps*Lot,data=Av_DS))
(anova(Creuat_IS_So3 <- lm(So3 ~ 1 + Temps*Lot,data=Av_DS))
```
```

Com s'observa en els 2 primers casos de línies paraleles aquest terme no té significació. Pel contrari sí que en té al model So3 on sí que hi ha diferències de pendents. Aquest diagnòstic ens serviria per veure s'ha d'aplicar algun terme que ajusti correctament el model com podria ser els efectes aleatoris de pendents.

### ML RI So

Avançant a partir del model simple amb intercepció ML I es fa l'ajust de models amb efecte aleatori generant la funció generada:

```
```{r ML IS RI So}
(Comp_IS_RI_So1 <- Comp_IS_RI_LM("So1",Av_DS))[1:3]
fmSo1_S <- formula(paste("So1","~","Temps"))
(eMLE_IS_So1<-exactRLRT(lme(fmSo1_S, random = ~1|Lot, data = Av_DS,
method="ML")))
(eREML_IS_So1<-exactRLRT(lme(fmSo1_S, random = ~1|Lot, data = Av_DS,
method="REML")))
(Comp_IS_RI_So2 <- Comp_IS_RI_LM("So2",Av_DS))[1:3]
fmSo2_S <- formula(paste("So2","~","Temps"))
(eMLE_IS_So2<-exactRLRT(lme(fmSo2_S, random = ~1|Lot, data = Av_DS,
method="ML")))
(eREML_IS_So2<-exactRLRT(lme(fmSo2_S, random = ~1|Lot, data = Av_DS,
method="REML")))
(Comp_IS_RI_So3 <- Comp_IS_RI_LM("So3",Av_DS))[1:3]
fmSo3_S <- formula(paste("So3","~","Temps"))
(eMLE_IS_So3<-exactRLRT(lme(fmSo3_S, random = ~1|Lot, data = Av_DS,
method="ML")))
(eREML_IS_So3<-exactRLRT(lme(fmSo3_S, random = ~1|Lot, data = Av_DS,
method="REML")))
```
```

Es genera una taula resum dels resultats obtinguts del test RLRT per començar:

```

```{r ML IS RI So resum RLRT}
I_IS_RI_So_RLRT <- data.frame(Dataset = paste(rep(paste(SoVars,c("I
RI", "IS RI")),each=3),c("Clàssic OLS", "MLE", "REML")))

Comp_RI_So_2 <- list("-", eMLE_IS_So1, eREML_IS_So1,
                    "-", eMLE_IS_So2, eREML_IS_So2,
                    "-", eMLE_IS_So3, eREML_IS_So3)
I_IS_RI_So_RLRT$p-valor          RLRT"          <-
sapply(1:length(Comp_RI_So_2), function(l1){
  if (class(Comp_RI_So_2[[l1]])!="htest"){
    return("-")
  } else {
    round(Comp_RI_So_2[[l1]]$p.value, 4)
  }
})

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

```

En aquests resultats com a comentari general es pot apreciar que:

- + No sembla haver-hi diferències significatives entre mètodes (MLE; REML).
- + Entre els tres conjunts de dades sembla que la dispersió de dades és la que activa la significació de l'efecte aleatori de la intercepció com és de lògica.

ML RIS So

Es modifica l'efecte aleatori afegint la variable temps:

```

```{r ML IS RIS funció}
Comp_IS_RIS_LM <- function(var, ds) {
 fm1 <- formula(paste(var, "~", 1, " + Temps", "+ Error(Lot/Temps)"))
 ML_IS_RIS_sOLS <- summary(ML_IS_RIS_OLS <- aov(fm1, data = ds))
 ML_IS_RIS_OLS
 ML_IS_RIS_OLS_err <- sigma(summary.lm(ML_IS_RIS_OLS$within))

 fm2 <- formula(paste(var, "~", "Temps"))

 ML_IS_RIS_sMLE <- summary(ML_IS_RIS_MLE <- lme(fm2, random =
~1+Temps|Lot, data = ds, control = ctrl1, method="ML"))

 ML_IS_RIS_sREML <- summary(ML_IS_RIS_REML <- lme(fm2, random =
~1+Temps|Lot, data = ds, control = ctrl1, method="REML"))

 1
list(ML_IS_RIS_OLS$within, ML_IS_RIS_sMLE, ML_IS_RIS_sREML, ML_IS_RIS_OLS
, ML_IS_RIS_MLE, ML_IS_RIS_REML)
 names(l) <- c("OLS IS RIS", "MLE IS RIS", "REML IS RIS", "OLS IS RIS
base", "MLE IS RIS base", "REML IS RIS base")

 return(l)
}
...

```

```
```{r ML IS RIS So}
#No es pot aplicar la funció exactRLRT quan hi ha més d'un efecte
aleatori, pel que es farà la comparació amb el model anterior on només
es tenia l'efecte RI mitjançant un ANOVA comparatiu.
```

```
(Comp_IS_RIS_So1 <- Comp_IS_RIS_LM("So1",Av_DS))[1:3]
```

```
(Comp_IS_RIS_So2 <- Comp_IS_RIS_LM("So2",Av_DS))[1:3]
```

```
(Comp_IS_RIS_So3 <- Comp_IS_RIS_LM("So3",Av_DS))[1:3]
```\n
```

```
```{r ML IS RIS So Anovas}
```

```
#NO ES PODEN FER ANOVAS DE RESULTATS DE FUNCIONS JA QUE NECESSITA LES
FORMULES ORIGINALS, ES FA INDIVIDUALMENT AMB ELS MODELS AJUSTATS PER
REML
```

```
So1_RI <- lme(So1 ~ 1 + Temps, random = ~1|Lot,control=ctrl,
method="REML",data=Av_DS)
So1_RIS <- lme(So1 ~ 1 + Temps, random = ~1+Temps|Lot,control=ctrl,
method="REML",data=Av_DS)
(So1_RI_RIS_an <- anova(So1_RI,So1_RIS))
```

```
So2_RI <- lme(So2 ~ 1 + Temps, random = ~1|Lot,control=ctrl,
method="REML",data=Av_DS)
So2_RIS <- lme(So2 ~ 1 + Temps, random = ~1+Temps|Lot,control=ctrl,
method="REML",data=Av_DS)
(So2_RI_RIS_an <- anova(So2_RI,So2_RIS))
```

```
So3_RI <- lme(So3 ~ 1 + Temps, random = ~1|Lot,control=ctrl,
method="REML",data=Av_DS)
So3_RIS <- lme(So3 ~ 1 + Temps, random = ~1+Temps|Lot,control=ctrl,
method="REML",data=Av_DS)
(So3_RI_RIS_an <- anova(So3_RI,So3_RIS))
```\n
```

Aquest possiblement seria un dels punts claus per entendre la utilitat de la utilització dels efectes aleatoris amb la covariant inclosa:

+ So1: El model centrat no hi ha significació entre els models RI i RIS ja que amb l'efecte aleatori dels lots o inclús sense aquest es podria generar el model fàcilment al voltant de les dades.

+ So2: Tot i ser no significatiu ja comença a baixar el p-valor donat que comença a haver més dispersió, però al ser encara rectes paraleles pot ser modelat correctament per l'efecte aleatori de la intercepció del lot.

+ So3: Al tenir diferents pendents en els diferents lots pren un efecte molt significatiu el fet d'afegir la variable de l'efecte aleatori relacionat amb la pendent o el que és el mateix amb la relació amb la covariant temps.

A continuació es fa la modificació afegint als models RIS les funcions de modulació de variància per veure si serien fets diferencials per millorar els models:

```
```{r ML IS RIS W So Anovas}
```

#NO ES PODEN FER ANOVAS DE RESULTATS DE FUNCIONS JA QUE NECESSITA LES FORMULES ORIGINALS, ES FA INDIVIDUALMENT AMB ELS MODELS AJUSTATS PER REML

```
So1_RI <- lme(So1 ~ 1 + Temps, random = ~1|Lot,control=ctrl,
method="REML",data=Av_DS)
So1_RIS <- lme(So1 ~ 1 + Temps, random = ~1+Temps|Lot,control=ctrl,
method="REML",data=Av_DS)
So1_RIS_w1 <- update(So1_RIS,control=ctrl,weights= varExp(form =
~Temps))
```

```
(So1_RI_RIS_an <- anova(So1_RIS,So1_RIS_w1))
```

```
So2_RI <- lme(So2 ~ 1 + Temps, random = ~1|Lot,control=ctrl,
method="REML",data=Av_DS)
So2_RIS <- lme(So2 ~ 1 + Temps, random = ~1+Temps|Lot,control=ctrl,
method="REML",data=Av_DS)
So2_RIS_w1 <- update(So2_RIS,control=ctrl,weights= varExp(form =
~Temps))
```

```
(So2_RI_RIS_an <- anova(So2_RIS,So2_RIS_w1))
```

```
So3_RI <- lme(So3 ~ 1 + Temps, random = ~1|Lot,control=ctrl,
method="REML",data=Av_DS)
So3_RIS <- lme(So3 ~ 1 + Temps, random = ~1+Temps|Lot,control=ctrl,
method="REML",data=Av_DS)
So3_RIS_w1 <- update(So3_RIS,control=ctrl,weights=varPower())
```

```
(So3_RI_RIS_an <- anova(So3_RIS,So3_RIS_w1))
```

```
So3_RIS_w2 <- update(So3_RIS,control=ctrl,weights=varExp(form =
~Temps))
```

```
(So3_RI_RIS_an2 <- anova(So3_RIS,So3_RIS_w2))
```

En els casos on ha estat possible s'ha aplicat la funció *varPower* i els casos on no s'ha arribat a cap convergència en les iteracions s'ha optat per l'aplicació de *varExp* en funció del temps.

Les conclusions no són molt clares en aquest punt, però semblaria que en l'únic cas on s'ha arribat a un model amb *varPower* dona un resultat no significatiu, però amb un p-valor bastant baix (`round(So3_RI_RIS_an[extract_itex]"p-value",4)`) al afegir aquest terme al model. La resta provat amb *varExp* resulten no significatius.

De fet, de lògica, no seria necessari en aquest tipus de conjunt simulats la funció de modular l'error estàndard ja que s'ha dissenyat dins de cada lot una dispersió relativament petita, pel que habitualment no es veuria necessari d'aplicar.

RESUM I DIAGNÒSTIC MODELS So

En el codi resum que es mostra a continuació es comparen els diferents models generats per veure els indicadors AIC i BIC, així com l'error estàndard residual:

```
```{r Resum ML So}
#So1
```

```

So1_R0 <-
list(Comp_I_IS_So1[5], Comp_IS_RI_So1[c(1,5:6)], Comp_IS_RIS_So1[c(1,5:6
)])
So1_R <- ConvList(So1_R0)
So1_R[[length(So1_R)+1]] <- So1_RIS_w1
names(So1_R)[length(So1_R)] <- "So1 Model IS REML RIS W"
ML_Resum(So1_R, "So1")

#So2
So2_R0 <-
list(Comp_I_IS_So2[5], Comp_IS_RI_So2[c(1,5:6)], Comp_IS_RIS_So2[c(1,5:6
)])
So2_R <- ConvList(So2_R0)
So2_R[[length(So2_R)+1]] <- So2_RIS_w1
names(So2_R)[length(So2_R)] <- "So2 Model IS REML RIS W"
ML_Resum(So2_R, "So2")

#So3
So3_R0 <-
list(Comp_I_IS_So3[5], Comp_IS_RI_So3[c(1,5:6)], Comp_IS_RIS_So3[c(1,5:6
)])
So3_R <- ConvList(So3_R0)
So3_R[[length(So3_R)+1]] <- So3_RIS_w1
names(So3_R)[length(So3_R)] <- "So3 Model IS REML RIS W"
ML_Resum(So3_R, "So3")
...

```

Els indicadors donen prova del que s'ha anat veient en els diferents models:

- + So1: Els models tenen poca diferència de qualitat en variància residual. Possiblement les diferències en AIC / BIC són pel tema de la distribució normal que es veurà també en els diagnòstics següents.
- + So2: S'aprecia clarament tant en els indicadors AIC/BIC com l'error estàndard residual la reducció al afegir el factor aleatori. Dins els models amb factor aleatori en canvi no s'aprecia diferència entre si és l'efecte d'intercepció aleatòria o intercepció/pendents aleatoris. Finalment les diferències amb AIC/BIC entre el tipus de càlcul de model s'observa possiblement pel compliment o no de normalitat.
- + So3: Disminució significativa al afegir els models amb efecte RI i encara més significativa al afegir els RIS. Quedaria bastant clar quin és el camí per enfocar aquest tipus de models amb efectes aleatoris. Tot i la aparent millora que afegeix el terme *\*varPower\** s'estudiarà cada cas concret si realment aquesta funció dona una millora a l'ajust del model.

Es visualitzen els gràfics de diagnòstic dels models base i REML:

```

```{r Diag Models So1}
So1_Diag0 <- list(So1_R[c(1,4,7)])

So1_Diag <- ConvList(So1_Diag0)

names(So1_Diag) <-
paste(rep("So1", each=length(So1_R[c(1,4,7)])), names(So1_Diag))

lapply(1:length(So1_Diag), function(QQ){
  if (class(So1_Diag[[QQ]])=="lme") {
    list(qqnorm(So1_Diag[[QQ]], abline = c(0,1), main=names(So1_Diag)
[QQ]), plot(So1_Diag[[QQ]], main=names(So1_Diag)[QQ]))
  } else {

```

```

        list(plot(So1_Diag[[QQ]],which=1,main=names(So1_Diag
[QQ]),plot(So1_Diag[[QQ]],which=2,main=names(So1_Diag)[QQ]))
    }
  })

```

...

```

```{r Diag Models So2}
So2_Diag0 <- list(So2_R[c(1,4,7)])

So2_Diag <- ConvList(So2_Diag0)

names(So2_Diag)
paste(rep("So2",each=length(So2_R[c(1,4,7)])),names(So2_Diag)) <-

lapply(1:length(So2_Diag),function(QQ){
 if (class(So2_Diag[[QQ]])=="lme") {
 list(qqnorm(So2_Diag[[QQ]],abline = c(0,1),main=names(So2_Diag
[QQ]),plot(So2_Diag[[QQ]],main=names(So2_Diag)[QQ]))
 } else {
 list(plot(So2_Diag[[QQ]],which=1,main=names(So2_Diag
[QQ]),plot(So2_Diag[[QQ]],which=2,main=names(So2_Diag)[QQ]))
 }
})

```

...

```

```{r Diag Models So3}
So3_Diag0 <- list(So3_R[c(1,4,7)])

So3_Diag <- ConvList(So3_Diag0)

names(So3_Diag)
paste(rep("So3",each=length(So3_R[c(1,4,7)])),names(So3_Diag)) <-

lapply(1:length(So3_Diag),function(QQ){
  if (class(So3_Diag[[QQ]])=="lme") {
    list(qqnorm(So3_Diag[[QQ]],abline = c(0,1),main=names(So3_Diag
[QQ]),plot(So3_Diag[[QQ]],main=names(So3_Diag)[QQ]))
  } else {
    list(plot(So3_Diag[[QQ]],which=1,main=names(So3_Diag
[QQ]),plot(So3_Diag[[QQ]],which=2,main=names(So3_Diag)[QQ]))
  }
})

```

...

Sobretot en els models So2 i So3 s'observa que al anar afegint paràmetres de millora al model els residus tenen menys desviació respecte a la distribució normal, així com la repartició en el gràfic de residus per homogeneïtat de variància ja que d'alguna manera, el model aconsegueix corregir els comportaments addicionals que el model simple no era capaç.

Prenent els models més efectius de la tècnica de màxima versemblança s'extreuen les matrius internes d'interès per analitzar més el detall:

```

```{r D Matrix So}
So_B
list(Comp_IS_RI_So1[[6]],Comp_IS_RI_So2[[6]],Comp_IS_RIS_So3[[6]]) <-

```



```

names(So_B) <- c(names(Comp_IS_RI_So1)[6], names(Comp_IS_RI_So2)
[6], names(Comp_IS_RIS_So3)[6])

lapply(1:length(So_B), function(B){
 VarCorr(So_B[[B]])
})

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

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

```

Comentaris sobre les matrius extrems dels conjunts So:

+ En els primers resultats s'observen a les matrius D donant poca informació, simplement es veu l'variància residual obtinguda en cada conjunt. Com a detall interessant veure que el tercer conjunt que podria ser el més complex s'ha aconseguit reduir l'variància residual lleugerament per sota dels dos primers conjunts que en teoria semblarien més fàcils de tractar.

+ 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 l'variància residual homogènia i per tant es veu com és la mateix per tots els temps i correspon a la diagonal.

+ La tercera sèria de matrius mostra:

+ La matriu primera de cada punt de la sèria és de tipus marginal on es suma l'variància residual i l'variància degut a l'efecte aleatori en la diagonal, mentre que l'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 l'variància total dins un dels individus (equivalent per tots els altres) que té l'variància de l'efecte aleatori sumat a l'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 l'variància dins el mateix lot.

+ A la segona matriu es plasma la correlació de variàncies i es reflexa la correlació entre els diferents temps del model per un individu (equivalent als altres).

+ Alguns comentaris d'aquestes 2 matrius:

+ En el primer conjunt So1 sembla tenir poca significació la covariant corresponent a l'efecte aleatori i de fet la correlació entre els temps és molt baixa propera a zero. Contràriament en els conjunts So2 i So3 augmenta considerablement l'efecte aleatori i també es reflexa en la correlació entre els temps que és relativament gran. Aquest fet en principi seria causat pel fet que al primer conjunt possiblement no és necessari l'efecte aleatori per explicar el model al tenir un model molt centrat i paral·lel, cosa que no passaria en els models So2 i So3 on (de diferent manera) hi ha una major dispersió que requeriria de l'efecte aleatori per ser explicada.

+ Tot i que en el conjunt So1 i So2 no varia la covariància i correlació entre temps, sí que ho fa en el conjunt So3, pel que en aquest cas sí que la funció ha aconseguit trobar certa correlació per poder reflectir en aquesta matriu. En els conjunts So1 i So2 la

variació s'entén que és de intercepció i per tant no hi hauria diferències segons el temps. El conjunt So3 on s'aplica també l'efecte aleatori de la pendent dona per fet que els diferents temps estaran correlacionats per poder construir la pendent que toqui en cada moment (de fet ja es veu l'estructura típica on els temps més propers estan més correlacionats que els temps més llunyans i que conicidiria amb un comportament habitual dels models en els estudis d'estabilitat).

```
PROVES DE VARIACIÓ DE LOTS/TEMPS So
```

S'utilitzen les fòrmules ja generades per graficar el heatmap de variació de variants sobre el model trobat com a òptim dins les proves realitzades amb el càlcul de màxima versemblança:

```
```{r Heatmaps So1}
set.seed(7982)
i <- seq(i3,i10)
j <- seq(j4,j10)

DS_So1_df <- sapply(1:i[length(i)],function(i){
  sapply(1:j[length(j)], function(j){
    runif(n=1, min=(TA[length(TA)]-TA[j])-1, max=(TA[length(TA)]-
TA[j])+1) #Observacions constant de rang petit entre -1 i 1 per
simular que no hi Ha tendències ni diferències
  })
})

#Es considera un Dataset amb el màxim de dades i després s'extreuen
les que interessin a cada punt del rang a testar

fm_ML_So1 <- formula(So1 ~ 1 + Temps)
re_RI <- formula(~1|Lot)

meREML <- "REML"
HeatMapRange_ML("So1",DS_So1_df, fm_ML_So1, re_RI, meREML, 0, 0)
```
```

El cas més perfecte dels lots centrats en la pendent de So1 reflexa com l'eficàcia del model depèn totalment de la variació del nombre de temps i no depèn del nombre de lots utilitzats. Possiblement es dona això degut a que la mitja dels lots sempre es situa allà mateix i únicament queda més o menys definida segons si conté més o menys repeticions o temps d'anàlisi.

```
```{r Heatmaps So2}
set.seed(7982)
DS_So2_df <- sapply(1:i[length(i)], function(i){
  m_lot <- runif(1, min=TA[length(TA)]-30, max=TA[length(TA)]+30)
  sapply(1:j[length(j)],function(j){
    runif(n=1, min=(m_lot-TA[j])-1, max=(m_lot-TA[j])+1)
  })
})

#Es considera un Dataset amb el màxim de dades i després s'extreuen
les que interessin a cada punt del rang a testar

fm_ML_So2 <- formula(So2 ~ 1 + Temps)

meREML <- "REML"
HeatMapRange_ML("So2",DS_So2_df, fm_ML_So2, re_RI, meREML, 0, 0)
```
```

```

```{r Heatmaps So3}
set.seed(7982)
DS_So3_df <- sapply(1:i[length(i)], function(i){
  m_lot <- runif(1, min=TA[length(TA)]-10, max=TA[length(TA)]+10)
  s_lot <- runif(1, min=0.1, max=2)
  sapply(1:j[length(j)], function(j){
    runif(n=1, min=(m_lot-(s_lot*TA[j]))-1, max=(m_lot-(s_lot*TA[j]))
+1)
  })
})

```

#Es considera un Dataset amb el màxim de dades i després s'extreuen les que interessin a cada punt del rang a testar

```

fm_ML_So3 <- formula(So3 ~ 1 + Temps)
re_RIS <- formula(~Temps|Lot)

```

```

meREML <- "REML"
HeatMapRange_ML("So3", DS_So3_df, fm_ML_So3, re_RIS, meREML, 0, 0)
```

```

A part de les diferències ja vistes entre els 3 datasets en aquest cas s'observa que:

- + Pels paràmetres indicatius de la qualitat del model AIC i BIC sembla que al tenir més lots/Temps augmenta aquest paràmetre disminuint l'eficàcia del model. Possiblement com en altres conjunts l'efecte de dispersió augmenta al afegir més lots i temps i per tant fa més difícil l'ajust d'un model eficient amb els recursos que s'han utilitzat fins ara.
- + El paràmetre de l'error estàndard residual no sembla seguir un patró concret, probablement degut a que tenim una mida mostral no molt gran i el càlcul d'aquests varia molt segons quines dades es posen o treuen.

Com en casos anteriors tot i que en aquest apartat es podria treure la conclusió que és millor tenir un nombre més limitat de lots i temps, en la realitat sempre és millor disposar del màxim de dades possibles ja que habitualment les dades d'un mateix lot no es comportaran amb una dispersió tan exagerada com s'ha simulat en aquests models. En tot cas és millor intentar millorar el model per adaptar-se a aquest augment de dades.