

TFM_ANNEX- 2_SIMULACIÓ_DADES_PERDUDES

Francesc Bernad Martin

May 31, 2019

Contents

FUNCIONS DE SIMULACIÓ DE MODELS MIXTOS	2
RECUPERACIÓ DATASETS GENERATS (Ho, He i So)	7
GENERACIÓ DATASET BASE MIG	7
GENERACIÓ DATASETS Ho	7
GENERACIÓ DATASETS He	7
GENERACIÓ DATASETS So	8
GENERACIÓ DATASETS AMB DADES PERDUDES	8
VISUALITZACIÓ DATASETS GENERATS	9
GRÀFICS GENERALS DADES PERDUDES	9
Spinegrams	9
Diagrames de Mosaic	12
Diagrames de caixa	15
Relació de paràmetres	18
MatrixPlot	19
GRÀFICS DE DADES PERDUDES ESPECÍFICS PER CONJUNT	20
MODELITZACIÓ AMB DADES PERDUDES	65
AJUST MODELS I CÀLCUL CONTRAST RLRT	65
AIC/BIC/Sigma	67
Contrast RLRT	76
Paràmetres Efectes Aleatoris	80
Suma de Quadrats	85
Intervals de Confiança de la predicció de mitges	85
IMPUTACIÓ DE DADES PERDUDES	92
MODELS D'IMPUTACIÓ	92
kNN	92
Regressió estocàstica	93
Ajust de models lineals	93
VISUALITZACIÓ RESULTATS	94
Scatterplots i Density plots	94
AIC/BIC/Sigma	110
Contrast RLRT	115
Paràmetres Efectes Aleatoris	118
Intervals de Confiança de la predicció de mitges	125

```
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("VIM")
usePackage("gridExtra")
usePackage("plyr")
usePackage("dplyr")
usePackage("naniar")
usePackage("mixlm")
usePackage("mice")

library(bootpredictlme4) # Si no està instal·lat fer servir: devtools::install_github("remkoduursma/bootpredictlme4")

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

```

Prenent de referència els datasets i models que s'han generat en la simulació corresponent als models mixtos, s'aprofita el codi de programació per fer la simulació de dades perdudes. En aquesta simulació s'eliminaran valors dels mateixos datasets generats de manera aleatòria per simular la pèrdua real de valors en un estudi.

Per reproduir alguns dels resultats de referència prenem la mateixa llavor ja utilitzada en el codi dels models mixtos essent aquesta 7982.

FUNCIONS DE SIMULACIÓ DE MODELS MIXTOS

Es recuperen algunes de les funcions utilitzades en la simulació de models mixtes per poder veure el comparatiu amb les dades perdudes i es generen algunes de noves (algunes es modifiquen per contemplar la presència de valors perduts):

```

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",na.action=na.omit))

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

  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)
}

```

```

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 adiat
    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 estàndard res. (Sigma)")) +
    theme(text = element_text(size=8),axis.text.x = element_text(angle=80,size = 10,hjust=0))

  return(list(R1,R2))
}

ML_Resum_RSS <- function(R,Key) {
  ML_table <- data.frame(TSS = sapply(1:length(R),function(s){
    round(sum((R[[s]]$residuals)^2),4)
  })
  )
  )

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

```

```

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))

Rp <- ggplot(data=ML_table_p, aes(x=Model, y=ParQ, fill=Par)) +
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Purples") + theme_minimal() +
  labs(title=Key,y = "Suma de quadrats residual (RSS)") +
  theme(text = element_text(size=10),axis.text.x = element_text(angle=80,size = 10,hjust=0))

return(Rp)
}

```

#Es genera una funció amb la simple instrucció de quan es facin llistats dintre llistats converteixi to

```

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)
}

```

Comp_W_LM <- function(fML,fRE,ds,WVI,WVE,WVP) # WVI;WVE;WVP Per decidir quines funcions s'apliquen esse

```

{

baseML <- lme(fML, random = fRE, control=ctrl, data = ds,method="ML",na.action=na.omit)
baseREML <- lme(fML, random = fRE, control=ctrl, data = ds,method="REML",na.action=na.omit)
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 tenim

```

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)
}
```

RECUPERACIÓ DATASETS GENERATS (Ho, He i So)

GENERACIÓ DATASET BASE MIG

Es torna a generar el dataset amb paràmetres de lots/temps mitjos:

```
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
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)
```

GENERACIÓ DATASETS Ho

```
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 si

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)
}))
```

GENERACIÓ DATASETS He

```
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))
  })
}))

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))
  })
}))
```

```

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)))
  })
}))

```

GENERACIÓ DATASETS So

```

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 ra
  })
}))

```

```

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)
  })
}))

```

```

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)
  })
}))

```

GENERACIÓ DATASETS AMB DADES PERDUDES

Per la generació dels datasets amb dades perdudes es tenen en compte varis nivells segons el % de dades perdudes i en base a cada nivell es generen dos vectors, un a nivell de lots i l'altre de temps d'anàlisi (TA) amb els punts a eliminar. Per fer-ho d'una manera semialeatòria s'utilitza la funció de R `sample_frac` del paquet `dplyr` amb la llavor comentada (generem una funció personalitzada per poder-ho fer més automàtic).

```

MD_list <- function(DS,percmin,percmax,percseq){ #La funció pren la matriu de dades i genera tantes mat
  perc_rang <- seq(from=percmin,to=percmax,by=percseq)
  MDlist <- lapply(c(0,perc_rang),function(pe){
    set.seed(7982)
    md_vec <- sample(nrow(DS),size=round(nrow(DS)*pe),replace=F)
    DS[md_vec,4:ncol(DS)] <- NA
    DS
  })
  names(MDlist) <- paste(100*c(0,perc_rang),"% VP",sep=" ")
}

```



```

    return(MDlist)
}

perc <- c(0.2,0.6,0.2)
perc_rang0 <- seq(from=perc[1],to=perc[2],by=perc[3])
Av_DS_MD_list <- MD_list(Av_DS,perc[1],perc[2],perc[3])

```

VISUALITZACIÓ DATASETS GENERATS

GRÀFICS GENERALS DADES PERDUDES

Els següents gràfics mostren les proporcions en general que s'apliquen de dades perdudes als punts sense tenir en compte a quina variable s'aplica. Encara que es vegi aplicat a Ho1 són els mateixos per la resta de variables.

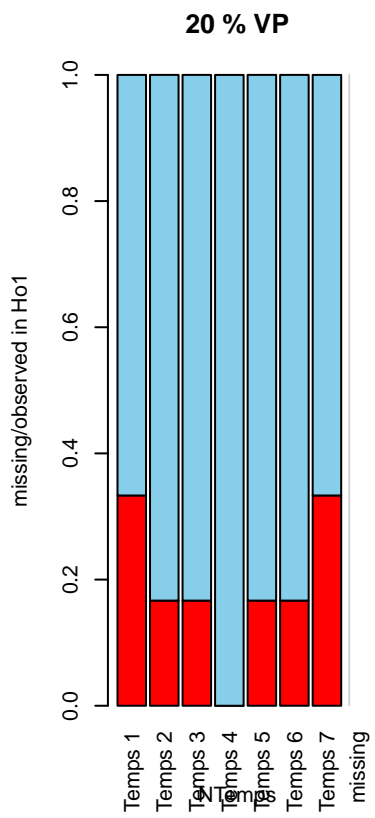
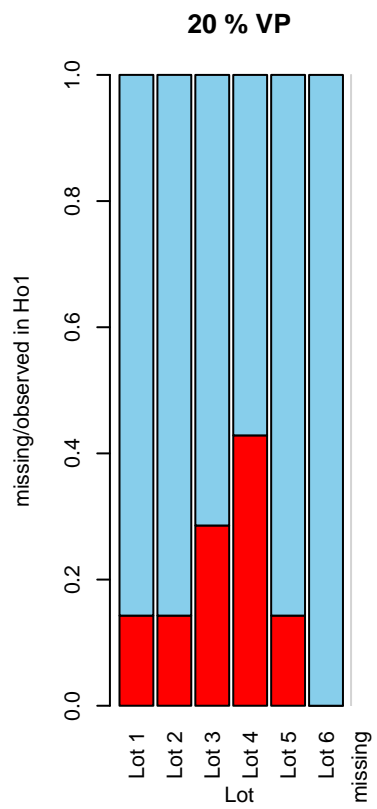
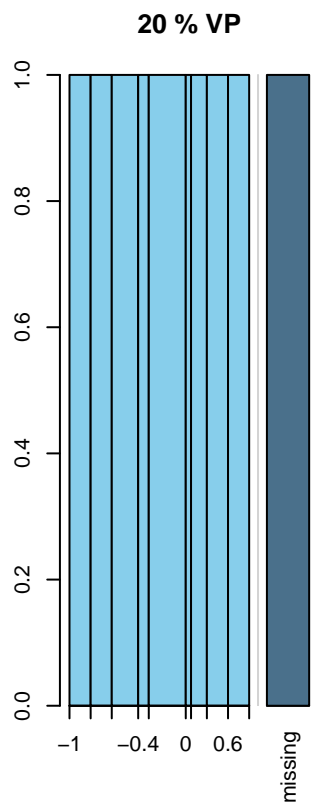
Spinegrams

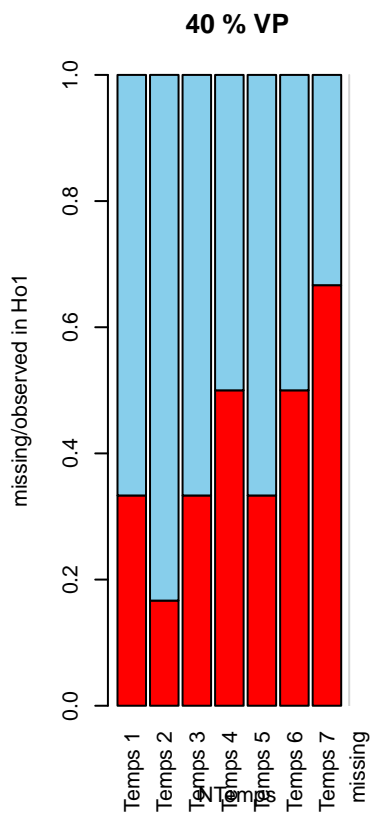
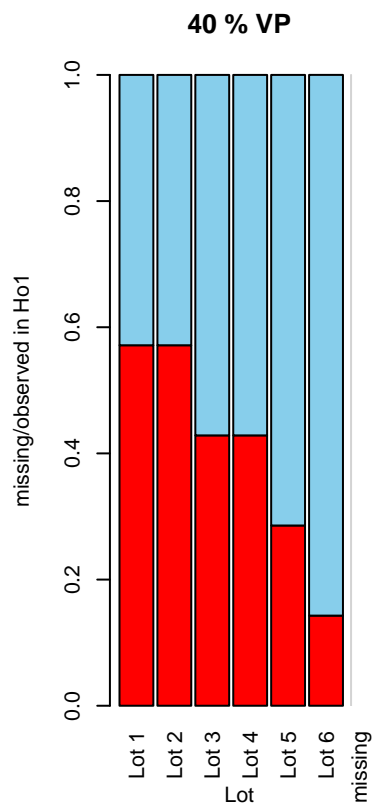
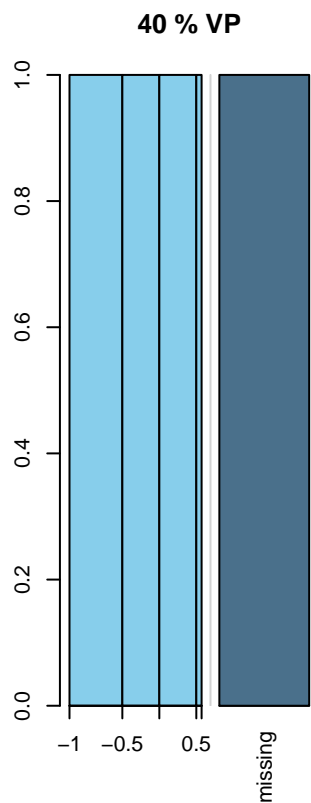
Veiem els gràfics anomenats spineplots que ajuden a visualitzar els percentatges de dades perdudes i les proporcions dins els factors de la matriu Lot i Temps:

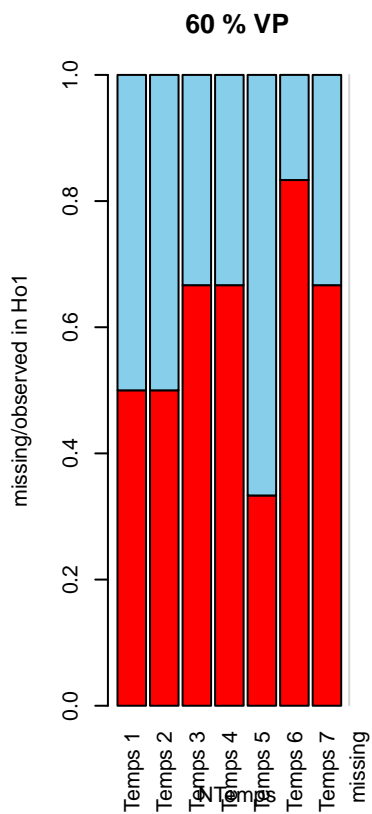
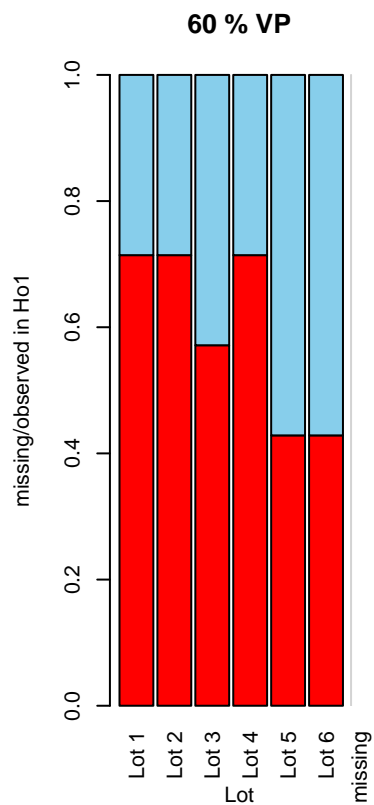
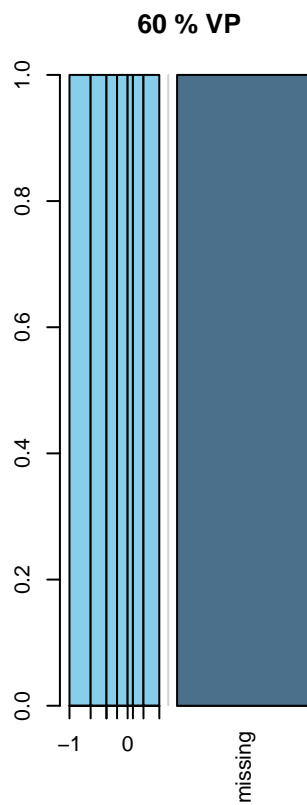
```

par(mfrow=c(1,length(Av_DS_MD_list)-1))
Av_DS_MD_spm <- lapply(2:length(Av_DS_MD_list),function(p){
  list(spineMiss(Av_DS_MD_list[[p]][,c("Ho1")],main=names(Av_DS_MD_list)[p]),
       spineMiss(Av_DS_MD_list[[p]][,c("Lot", "Ho1")],main=names(Av_DS_MD_list)[p]),
       spineMiss(Av_DS_MD_list[[p]][,c("NTemps", "Ho1")],main=names(Av_DS_MD_list)[p])
    )
})

```



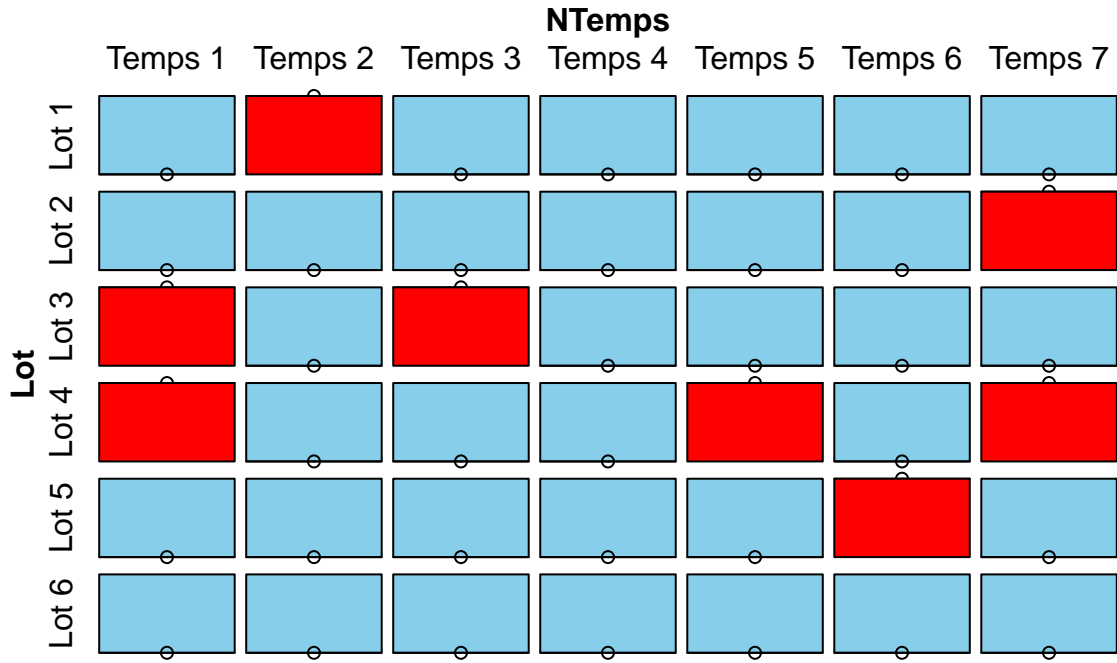




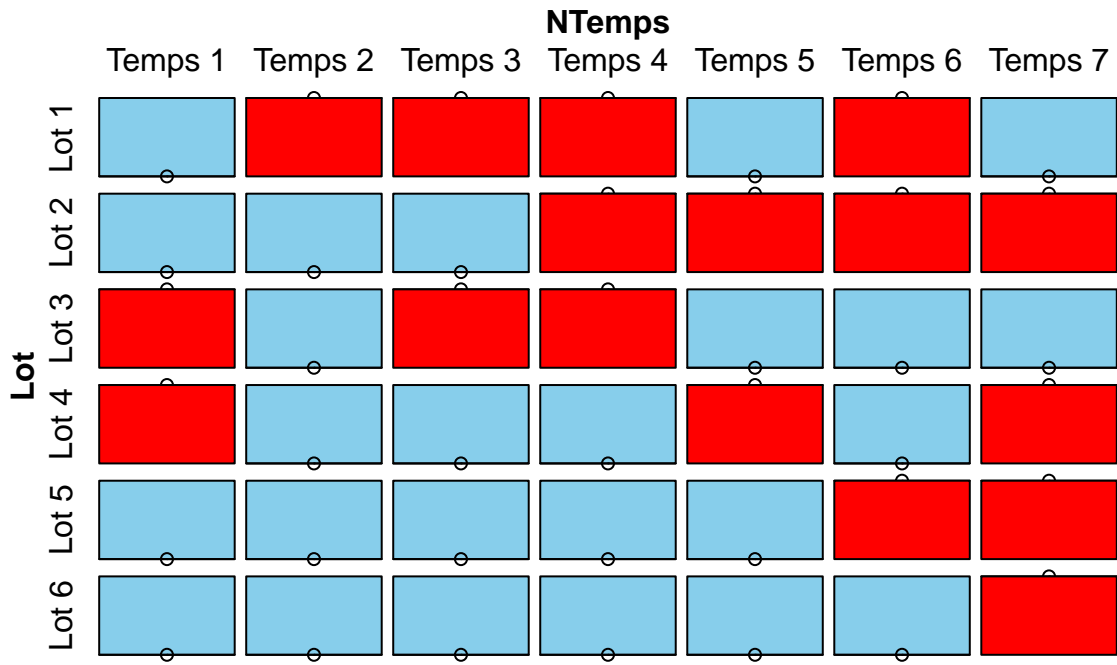
Diagrames de Mosaic

```
Av_DS_MD_mos <- lapply(2:length(Av_DS_MD_list),function(p){
  list(mosaicMiss(Av_DS_MD_list[[p]][, c("Lot","NTemps","Ho1")], highlight = 3,
    plotvars = 1:2, miss.labels = FALSE,main=names(Av_DS_MD_list)[p]))
})
```

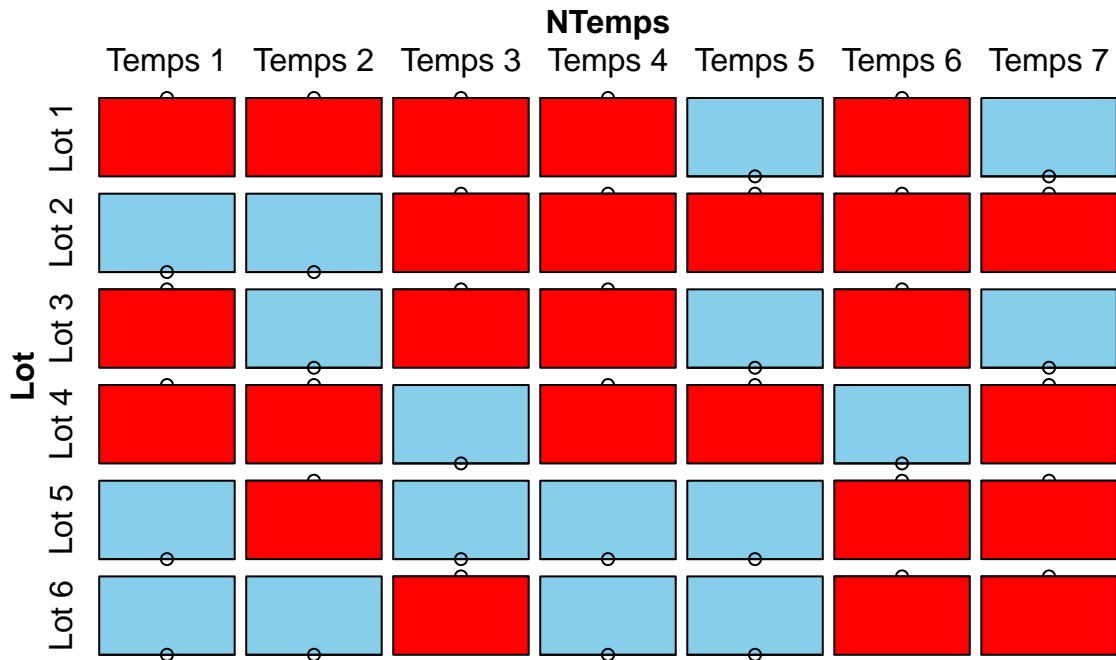
20 % VP



40 % VP



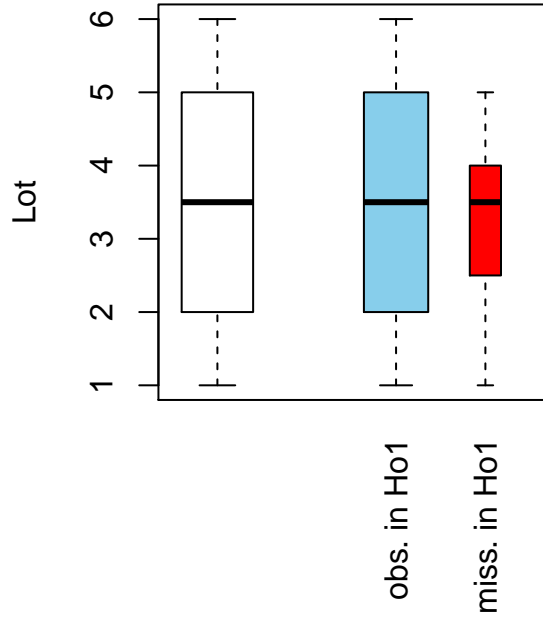
60 % VP



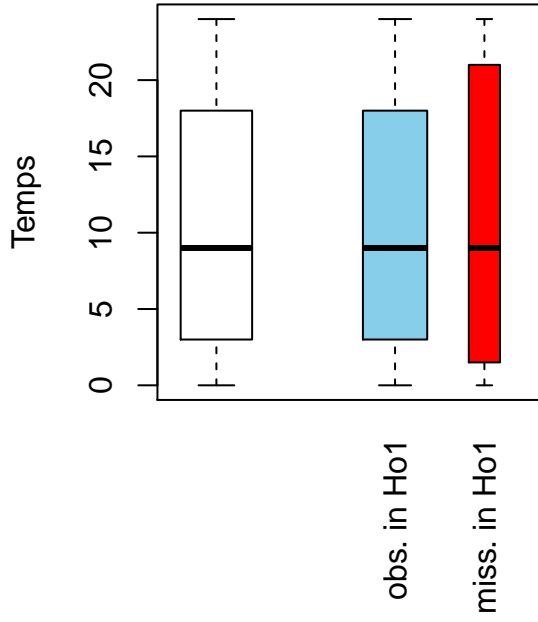
Diagrames de caixa

```
par(mfrow=c(1,2))
Av_DS_MD_pbox <- lapply(2:length(Av_DS_MD_list),function(p){
  list(pbox(Av_DS_MD_list[[p]][, c("Lot", "Ho1")],main=names(Av_DS_MD_list)[p]),
       pbox(Av_DS_MD_list[[p]][, c("Temps", "Ho1")],main=names(Av_DS_MD_list)[p]))
})
```

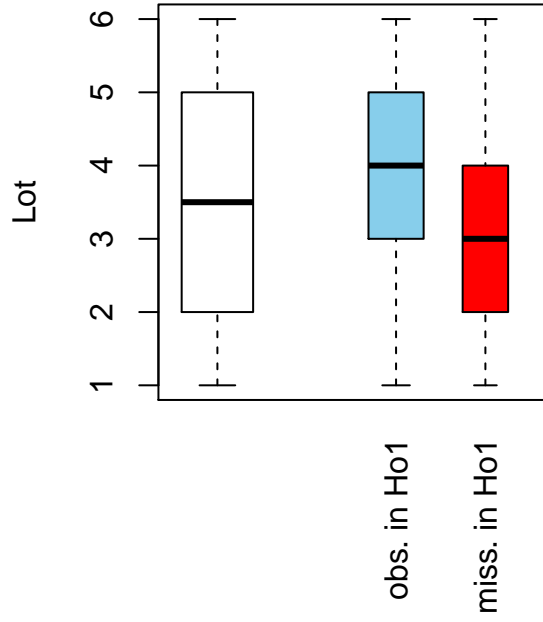
20 % VP



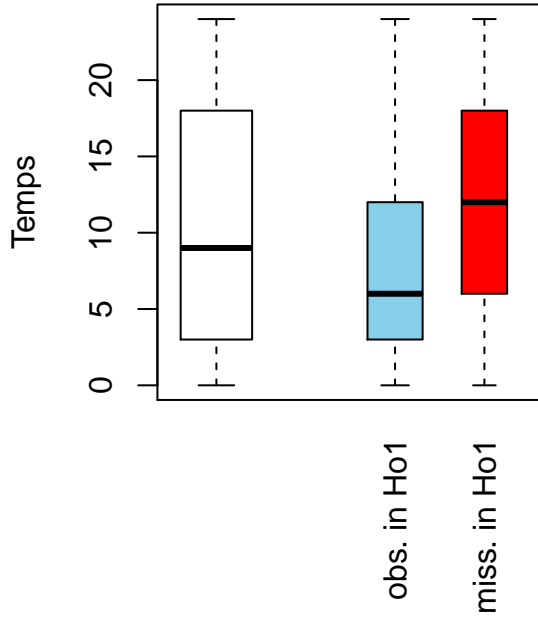
20 % VP

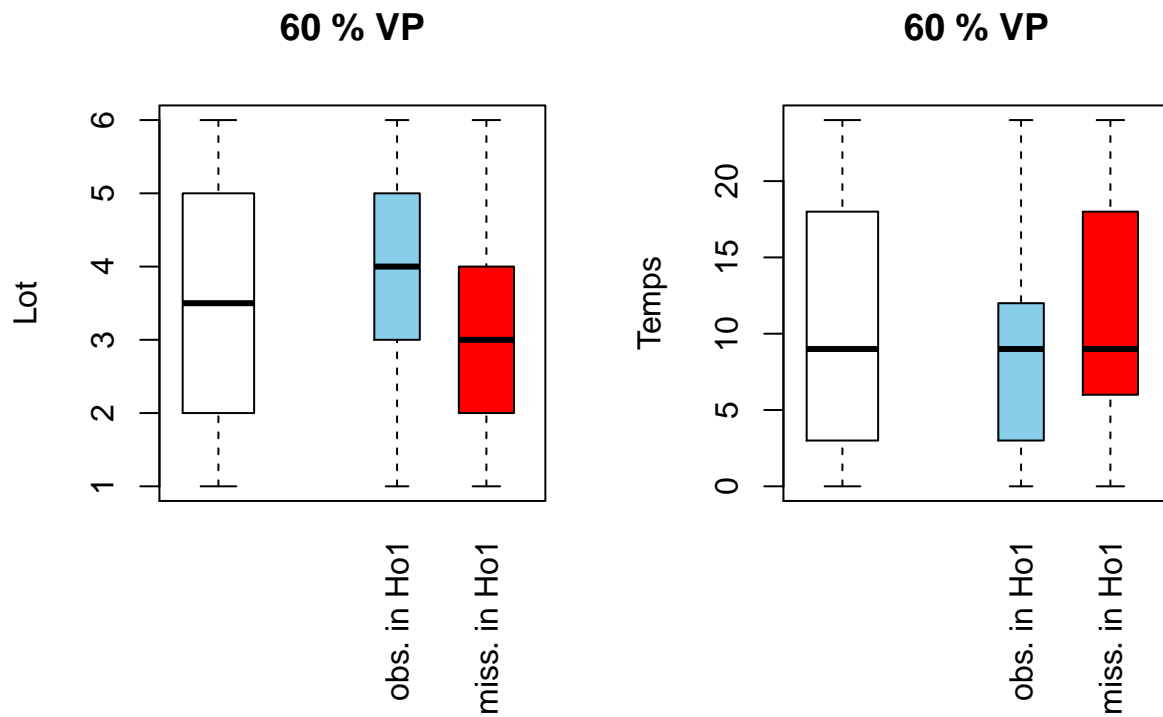


40 % VP



40 % VP

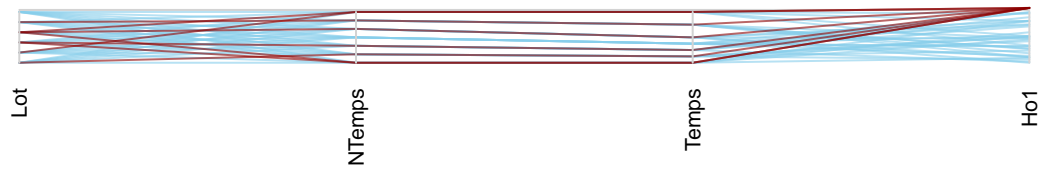




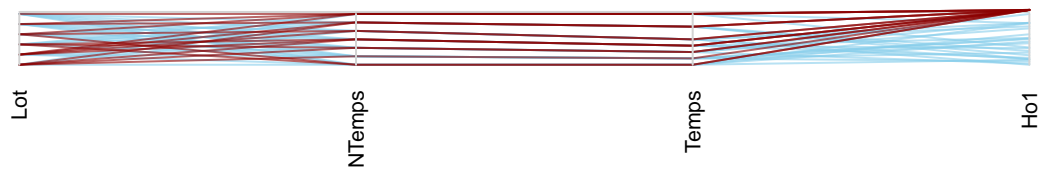
Relació de paràmetres

```
par(mfrow=c(length(Av_DS_MD_list)-1,1))
Av_DS_MD_pcor <- lapply(2:length(Av_DS_MD_list),function(p){
parcoordMiss(Av_DS_MD_list[[p]][, c("Lot","NTemps","Temps", "Ho1")], highlight = 'Ho1', alpha = 0.6,main
})
```

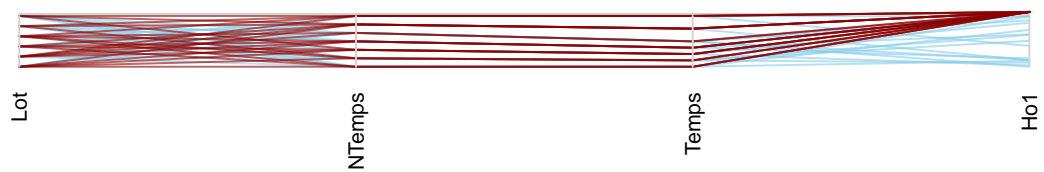
20 % VP



40 % VP

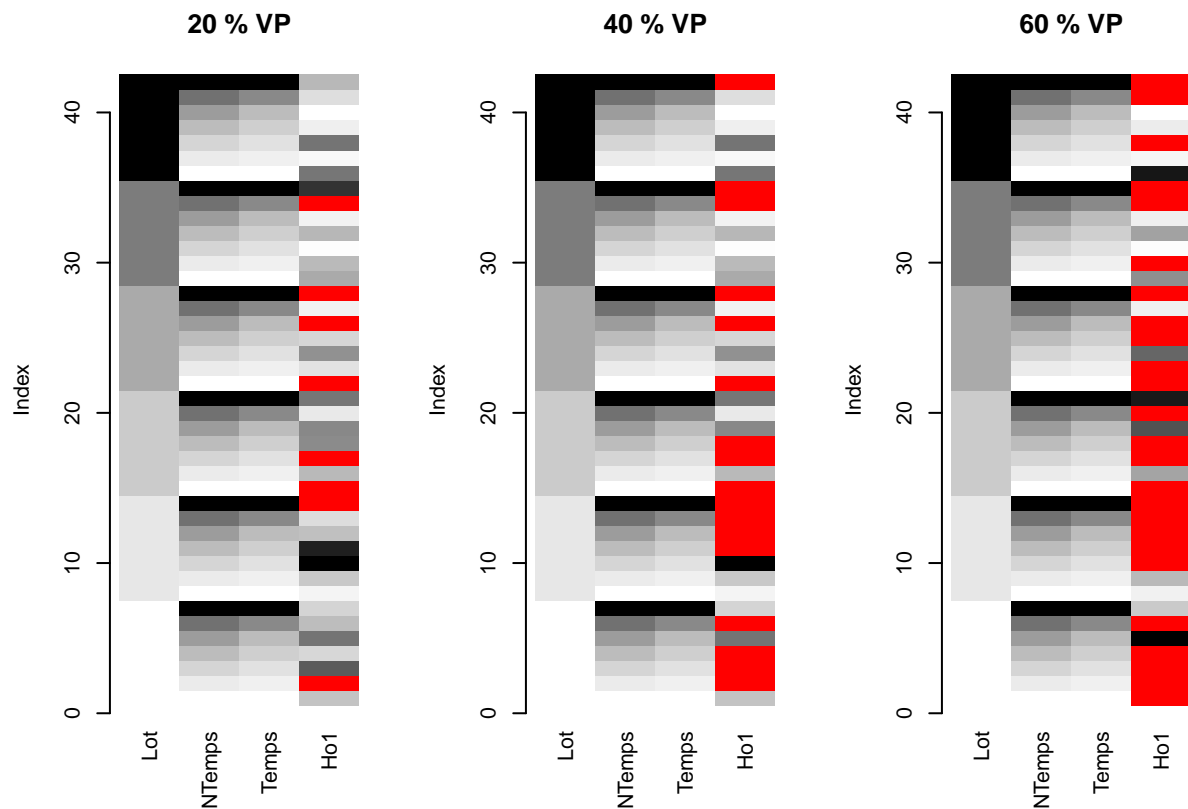


60 % VP



MatrixPlot

```
par(mfrow=c(1,length(Av_DS_MD_list)-1))
Av_DS_MD_matp <- lapply(2:length(Av_DS_MD_list),function(p){
matrixplot(Av_DS_MD_list[[p]][, c("Lot","NTemps","Temps", "Ho1")], sortBy=c('Lot'),main=names(Av_DS_MD_
})
```



En els diferents gràfics es mostra com s'ha aplicat d'una manera aproximadament igualada les absències de dades. Evidentment com més petita és la mida mostral és més fàcil que es donin diferències i veure possibles patrons que realment amb una mida mostral més gran no hi serien.

GRÀFICS DE DADES PERDUDES ESPECÍFICS PER CONJUNT

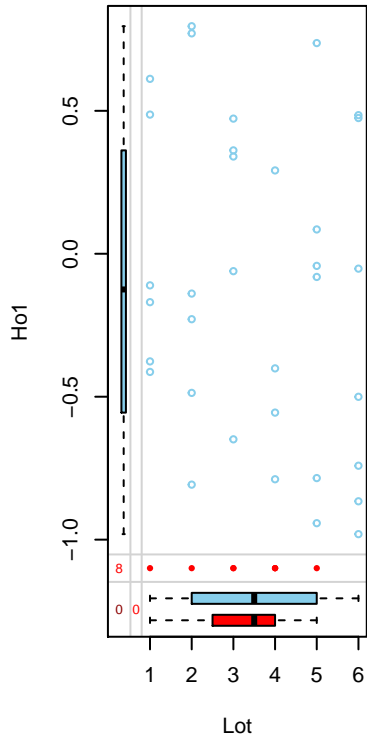
```

DS_Vars <- c("Ho1", "Ho2", "He1", "He2", "He3", "So1", "So2", "So3")

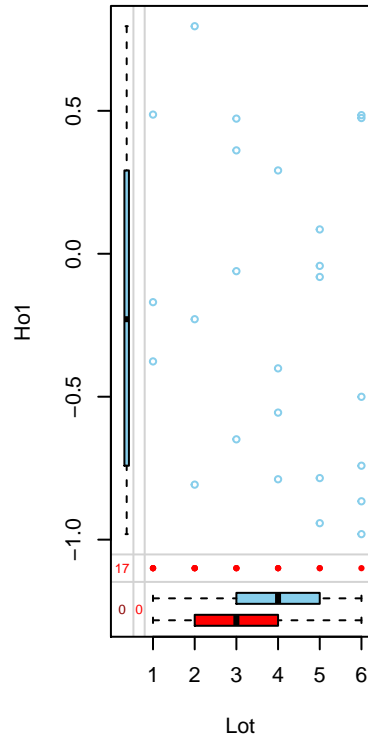
lapply(DS_Vars, function(V){
  par(mfrow=c(1, length(Av_DS_MD_list)-1))
  Lot_MD_marp <- lapply(2:length(Av_DS_MD_list), function(p){
    marginplot(Av_DS_MD_list[[p]][, c("Lot", V)], main=names(Av_DS_MD_list)[p])
  })
  Temps_MD_marp <- lapply(2:length(Av_DS_MD_list), function(p){
    marginplot(Av_DS_MD_list[[p]][, c("NTemps", V)], main=names(Av_DS_MD_list)[p])
  })
})

```

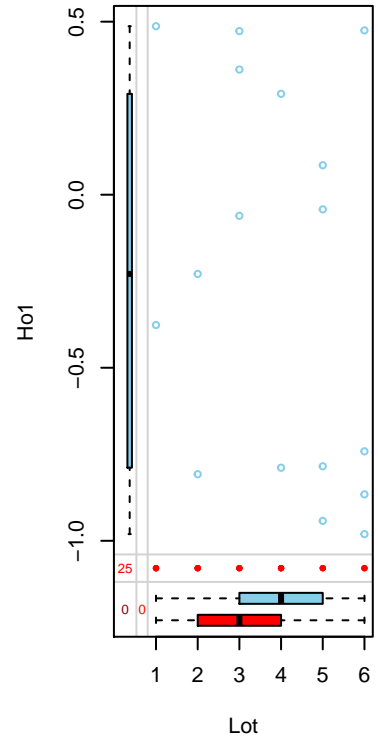
20 % VP

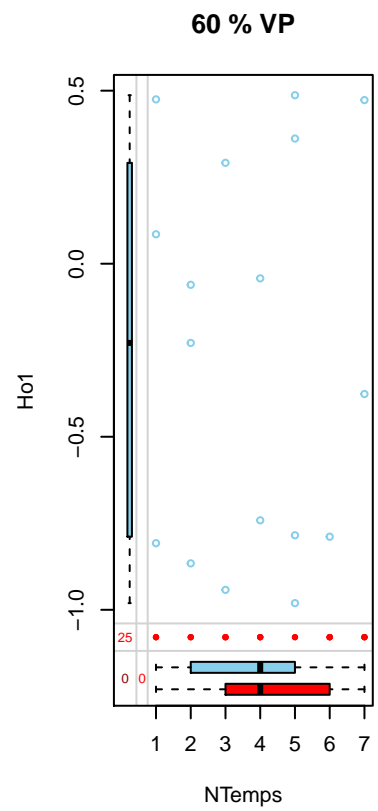
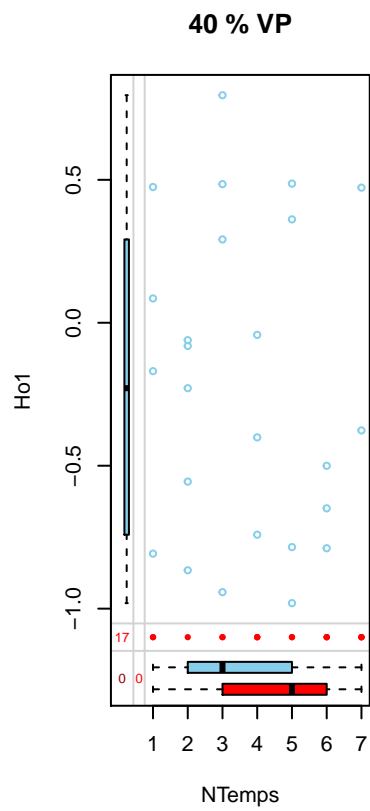
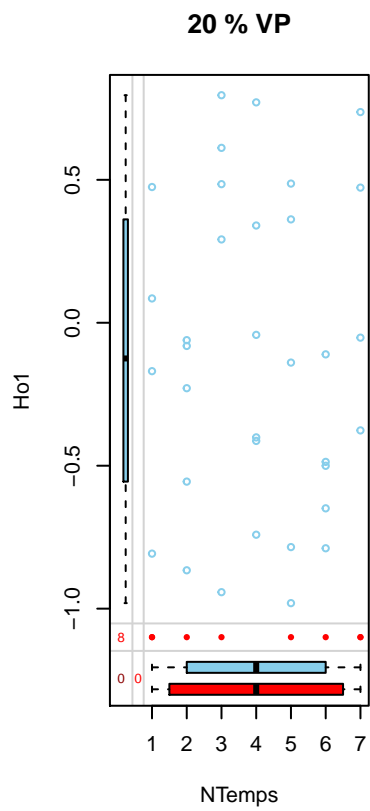


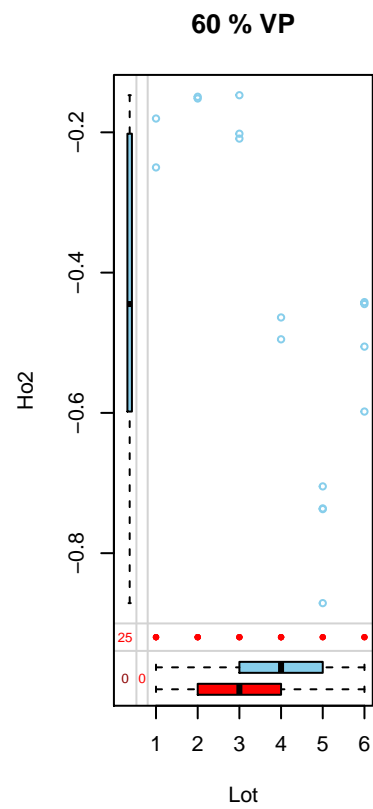
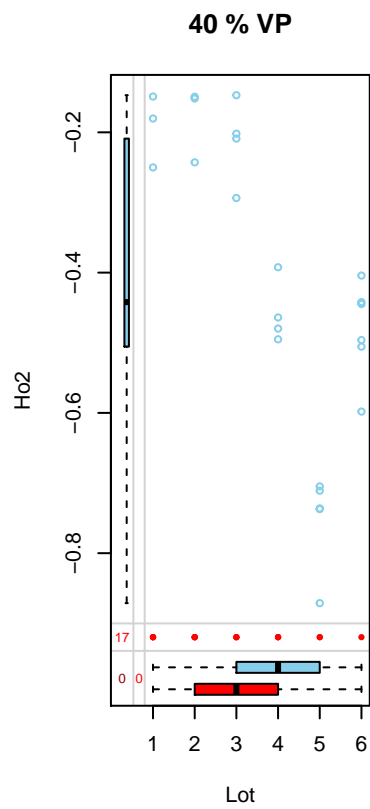
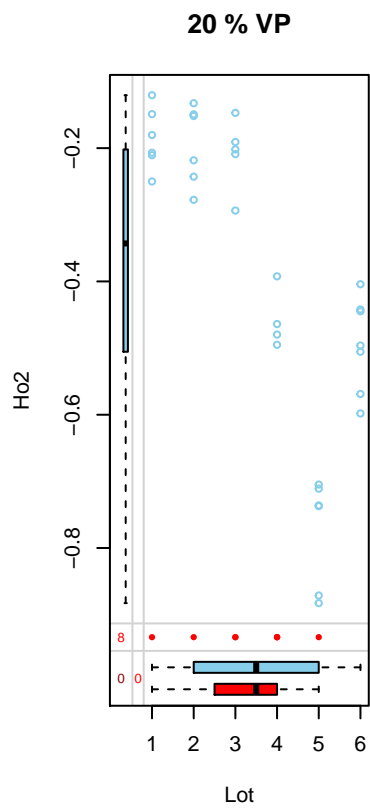
40 % VP

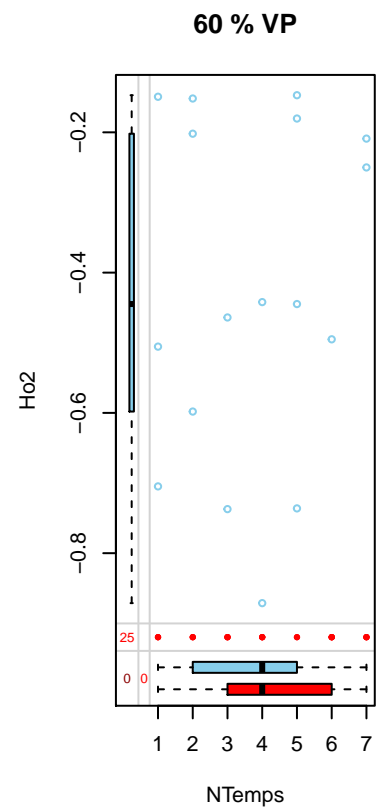
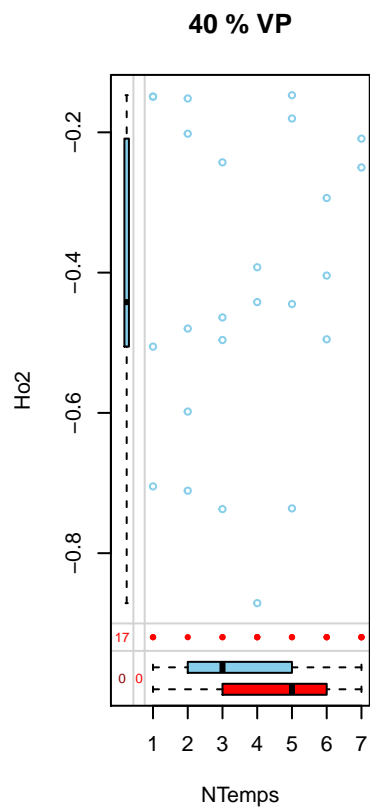
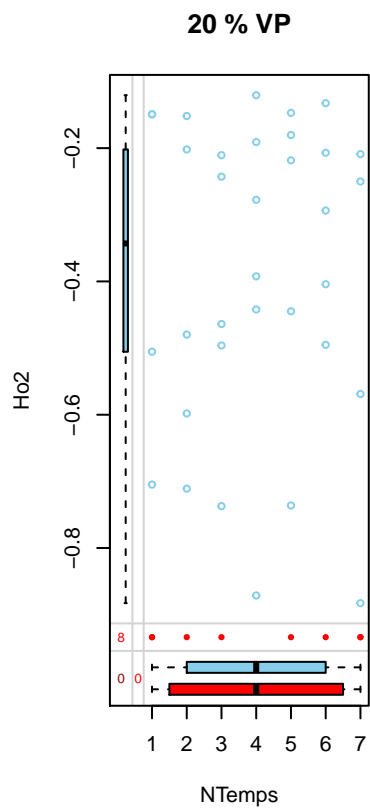


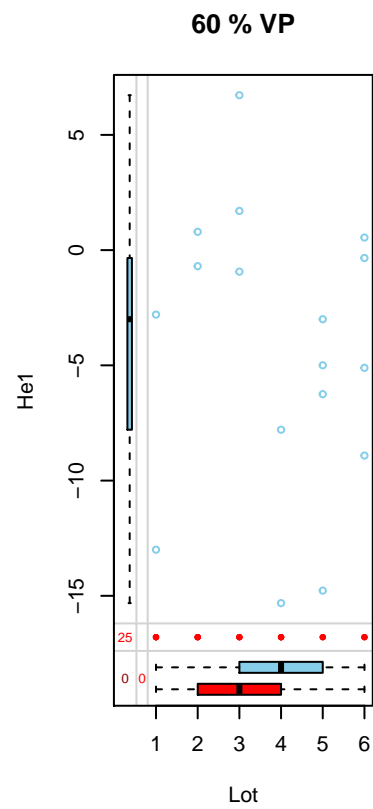
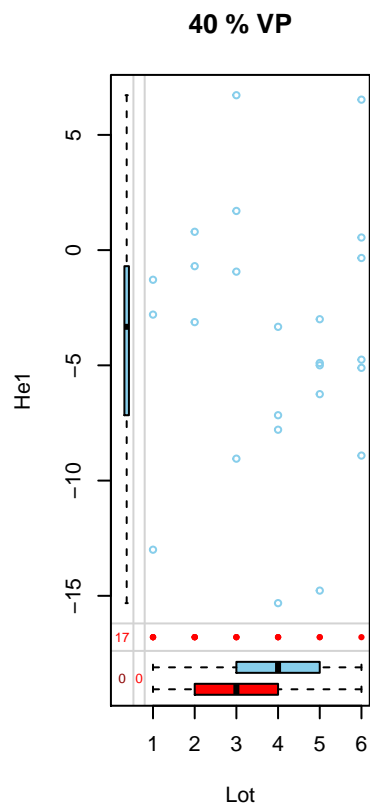
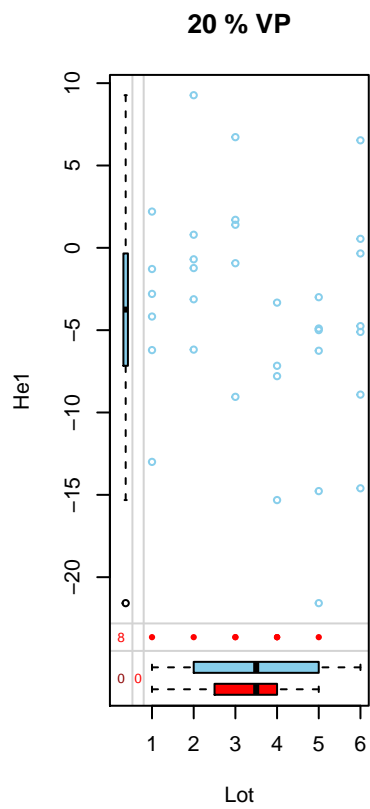
60 % VP

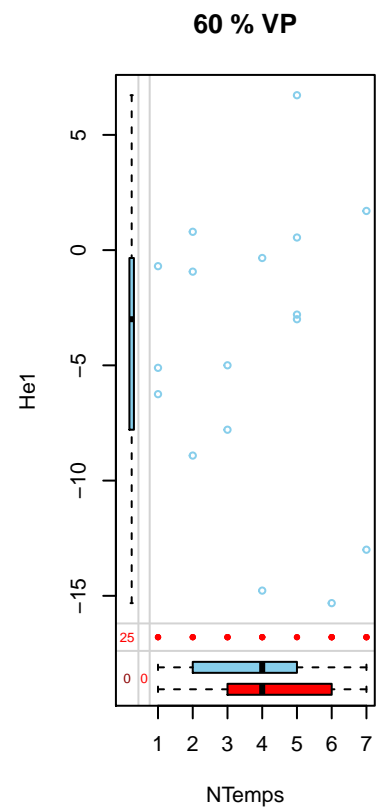
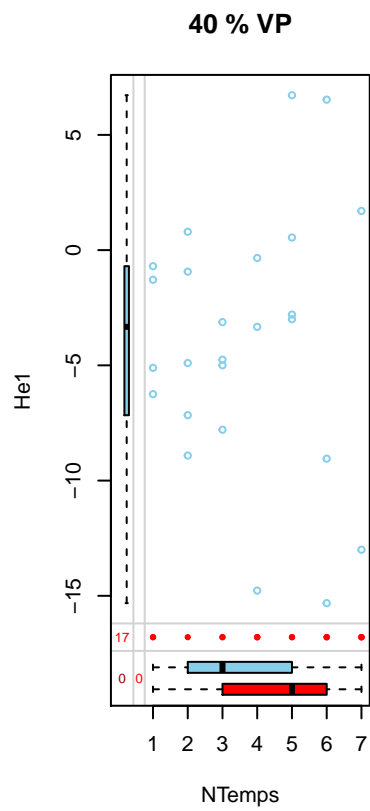
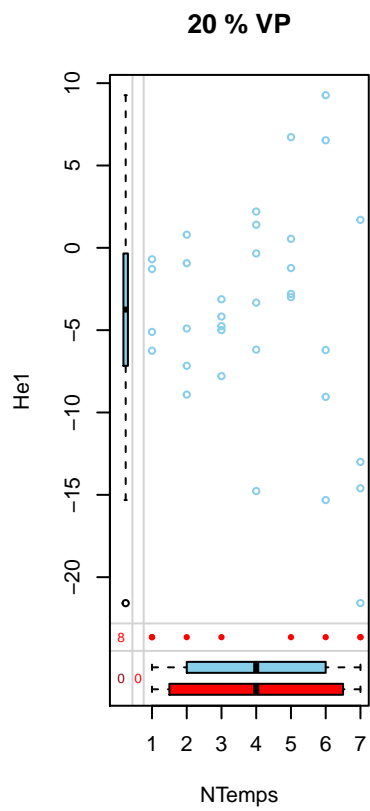




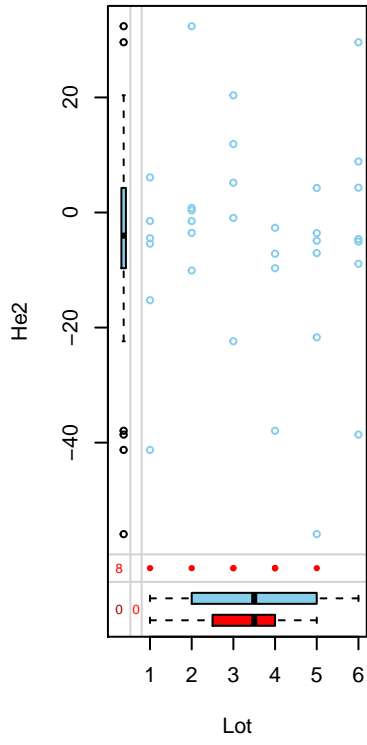




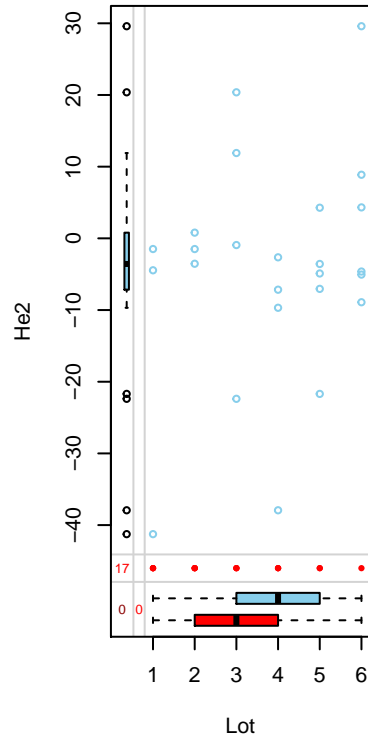




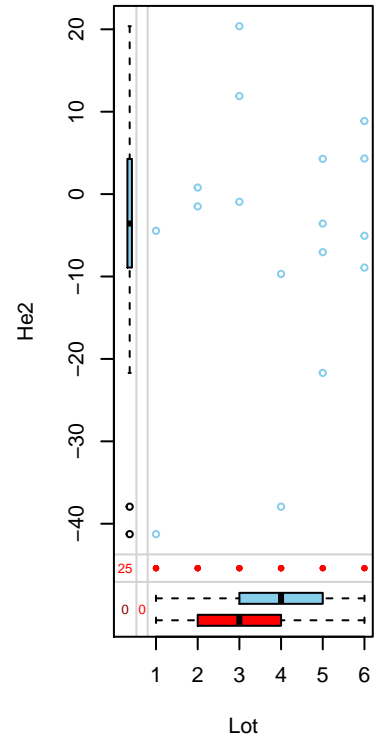
20 % VP



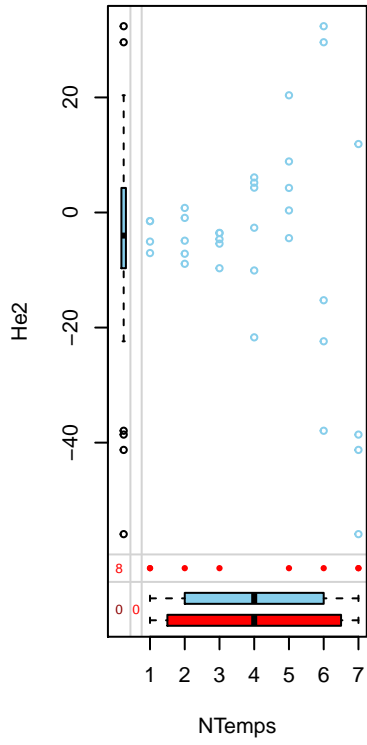
40 % VP



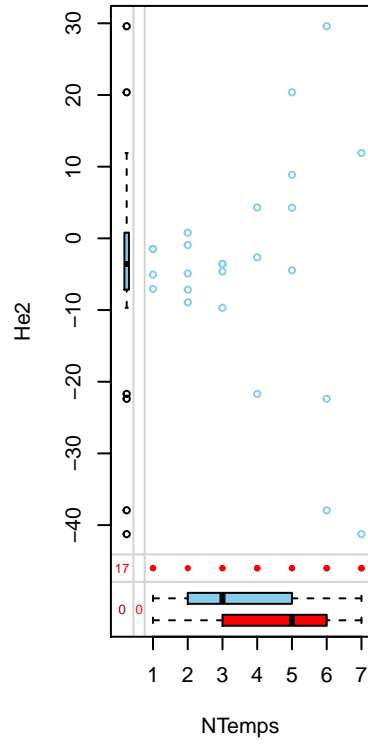
60 % VP



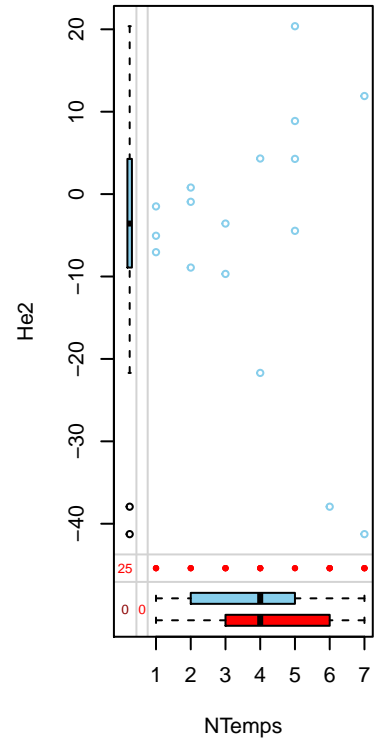
20 % VP

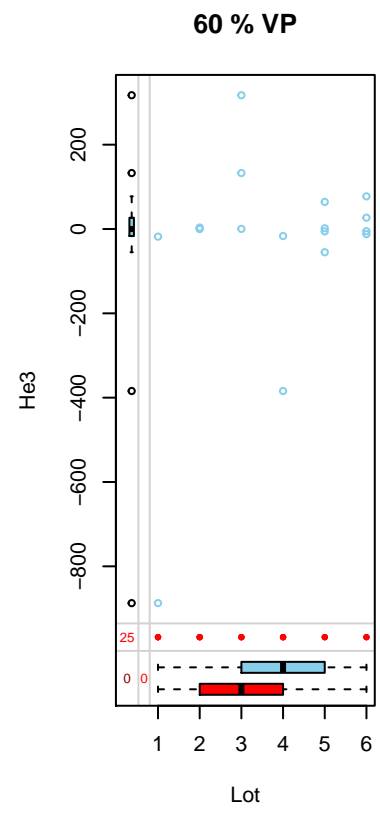
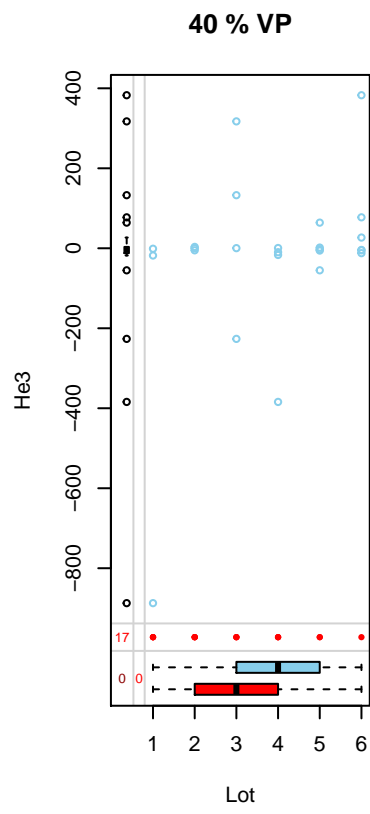
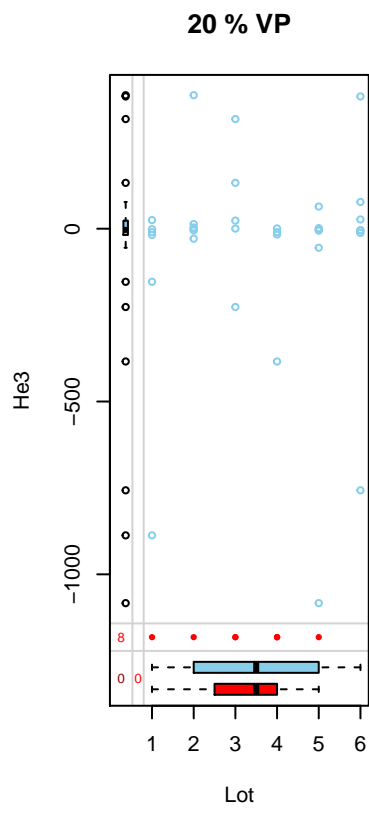


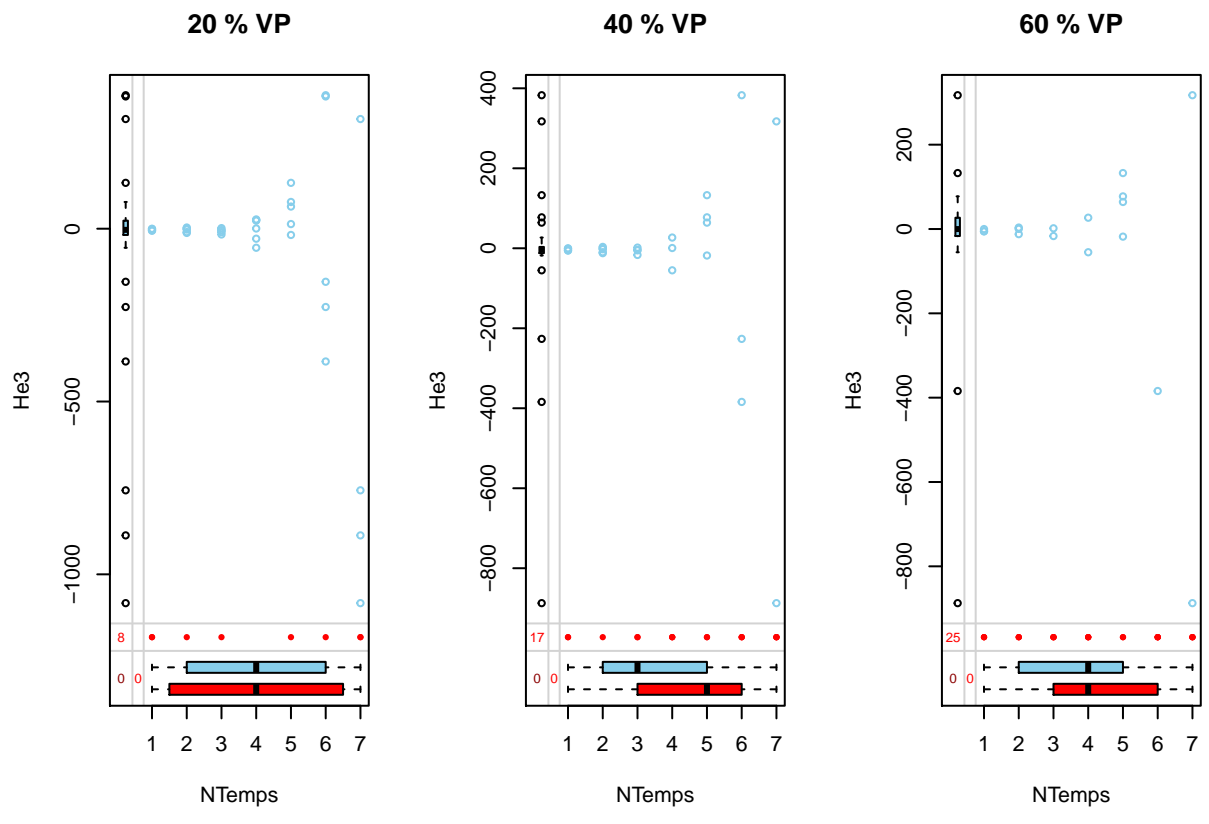
40 % VP



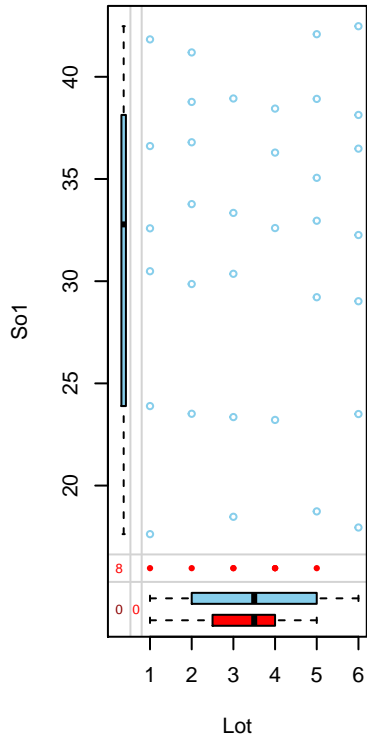
60 % VP



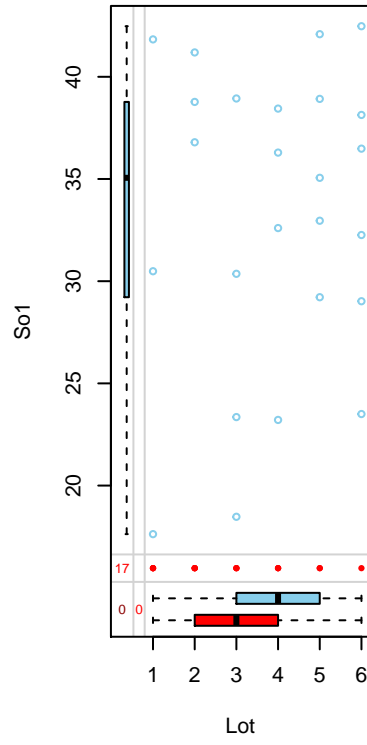




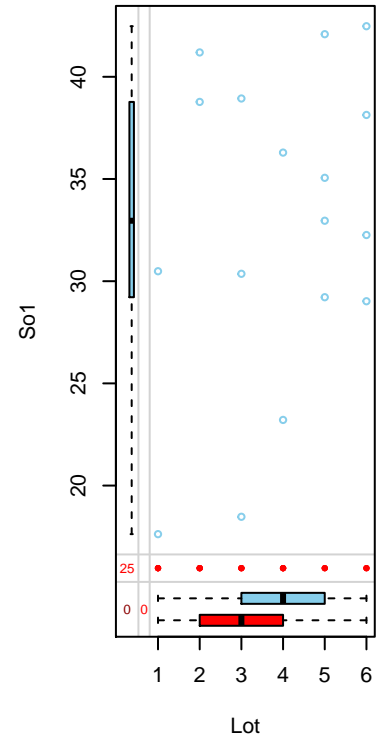
20 % VP



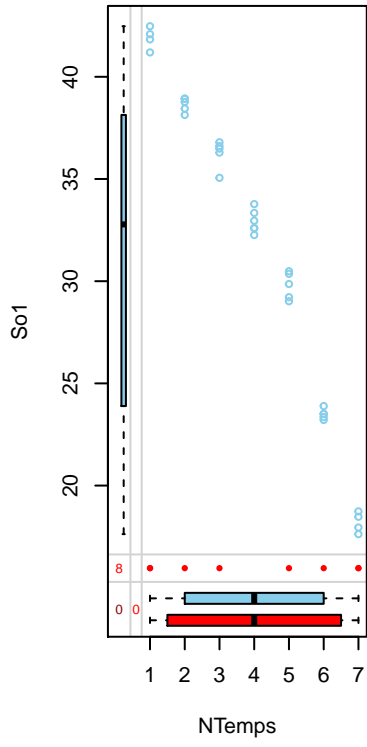
40 % VP



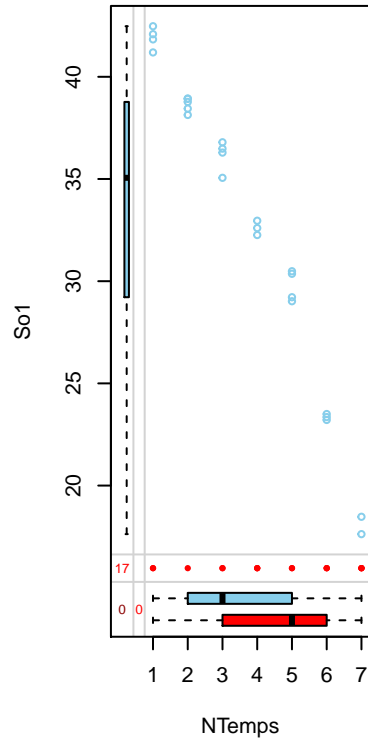
60 % VP



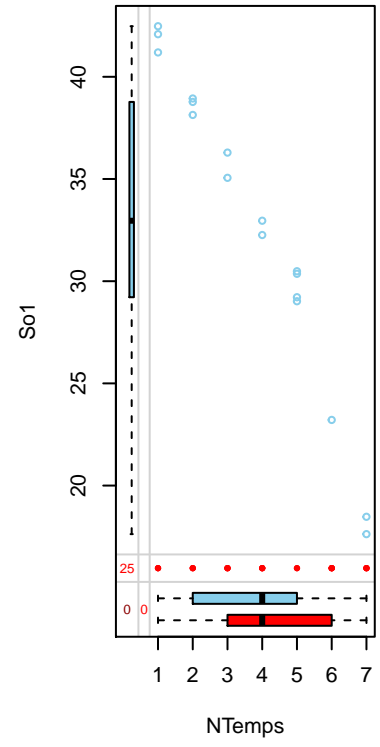
20 % VP



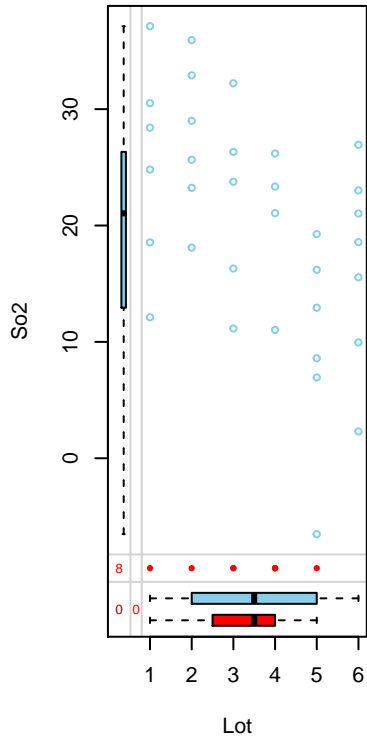
40 % VP



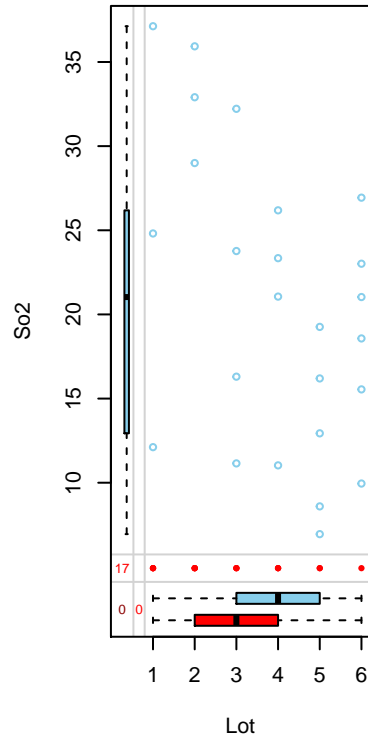
60 % VP



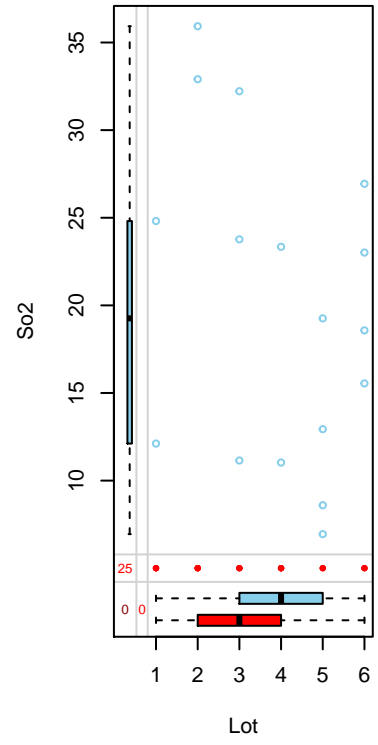
20 % VP



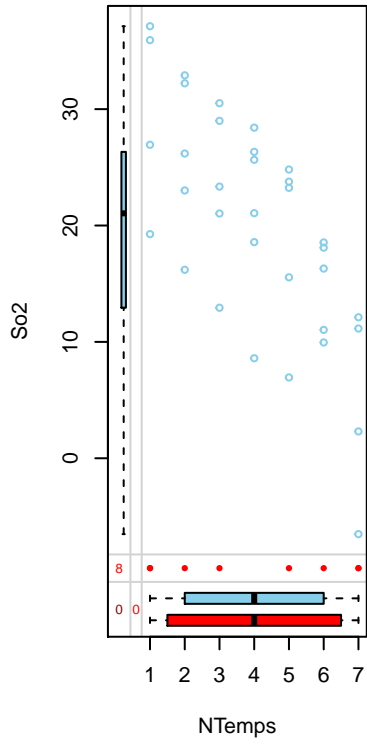
40 % VP



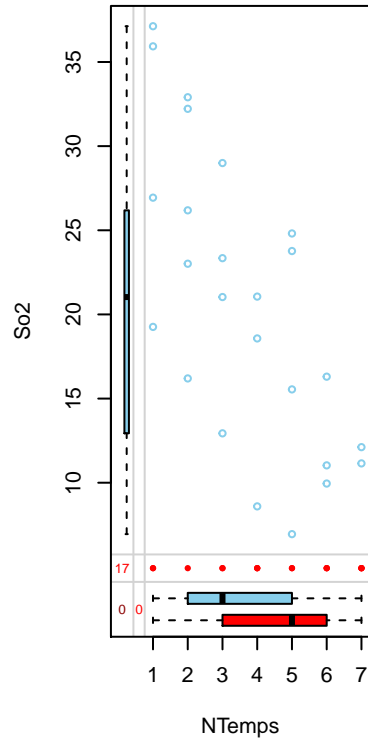
60 % VP



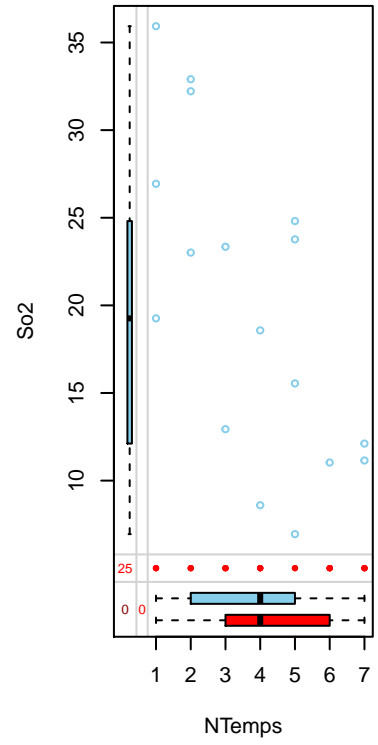
20 % VP

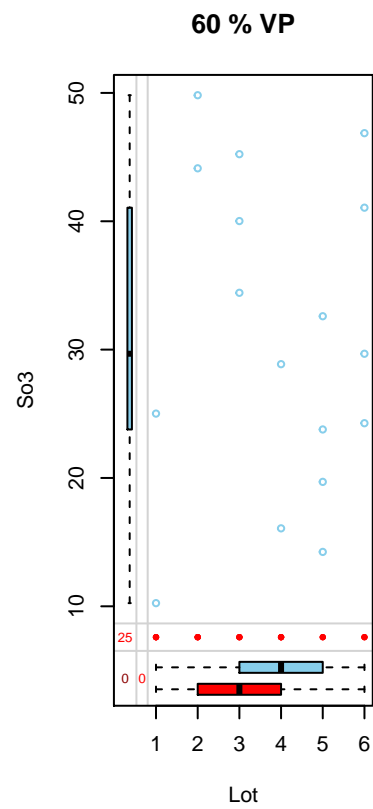
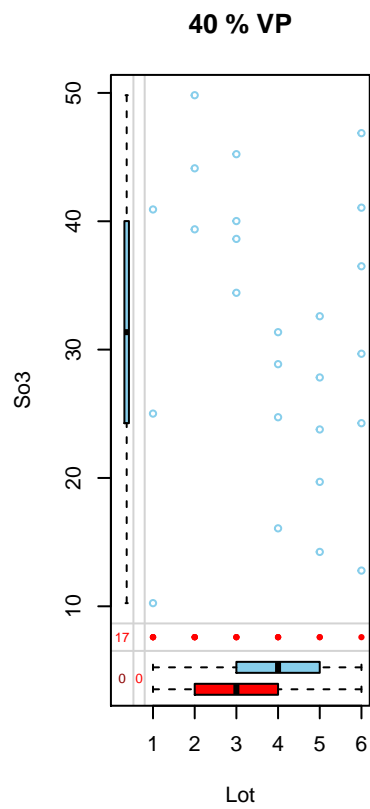
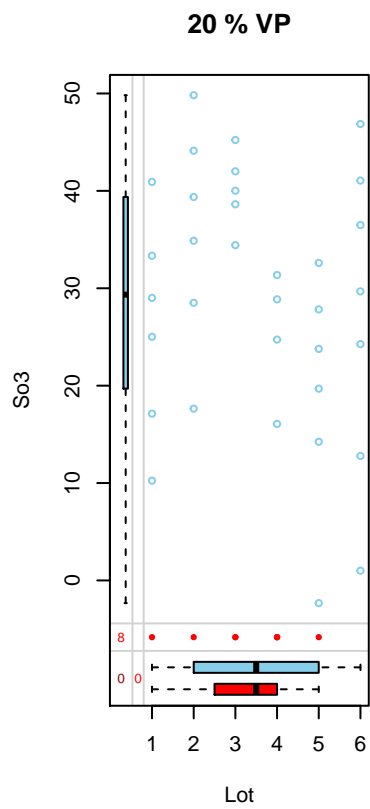


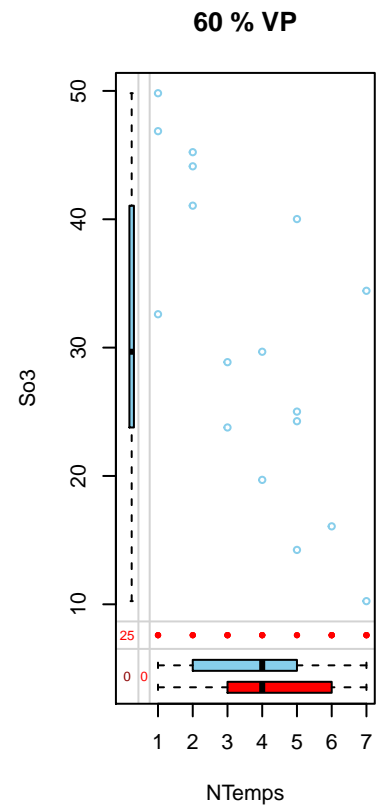
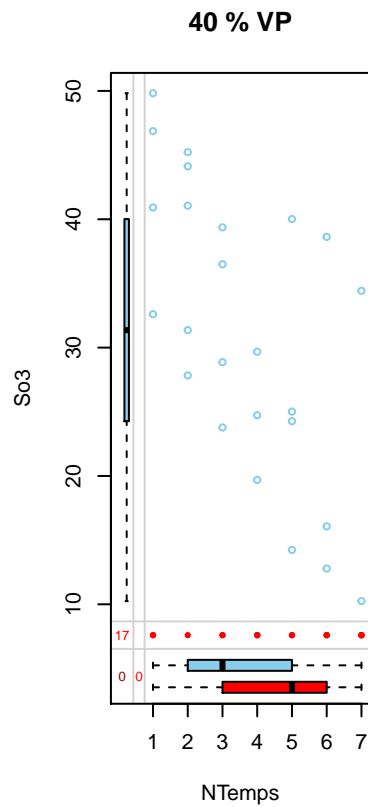
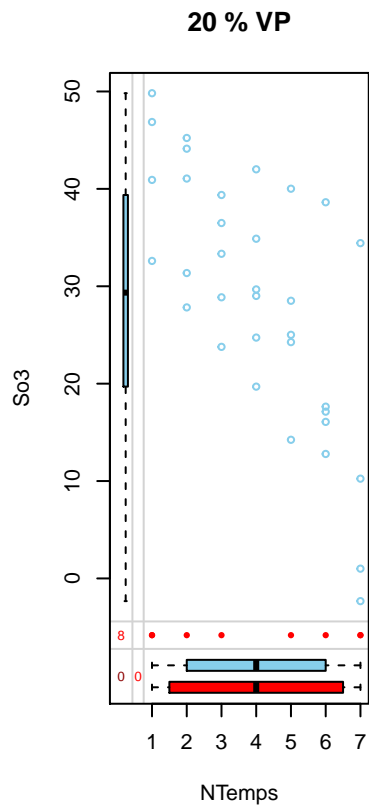
40 % VP



60 % VP







```
## [[1]]
## [[1]][[1]]
## NULL
##
## [[1]][[2]]
## NULL
##
## [[1]][[3]]
## NULL
##
##
## [[2]]
## [[2]][[1]]
## NULL
##
## [[2]][[2]]
## NULL
##
## [[2]][[3]]
## NULL
##
##
## [[3]]
## [[3]][[1]]
## NULL
##
```

```
## [[3]][[2]]
## NULL
##
## [[3]][[3]]
## NULL
##
##
## [[4]]
## [[4]][[1]]
## NULL
##
## [[4]][[2]]
## NULL
##
## [[4]][[3]]
## NULL
##
##
## [[5]]
## [[5]][[1]]
## NULL
##
## [[5]][[2]]
## NULL
##
## [[5]][[3]]
## NULL
##
##
## [[6]]
## [[6]][[1]]
## NULL
##
## [[6]][[2]]
## NULL
##
## [[6]][[3]]
## NULL
##
##
## [[7]]
## [[7]][[1]]
## NULL
##
## [[7]][[2]]
## NULL
##
## [[7]][[3]]
## NULL
##
##
## [[8]]
## [[8]][[1]]
## NULL
```

```
##
## [[8]][[2]]
## NULL
##
## [[8]][[3]]
## NULL
```

```
par(mfrow=c(1,1))
```

Els gràfics específics per variable ens mostren com canvia la distribució de dades perdudes en front del factor Lot i del factor Temps. Al estar aplicant una simulació de MCAR pur es pot veure que:

- En els gràfics sense patró lògic al tenir absència de dades segueix aproximadament una distribució similar, excepte potser algun cas del % més alt de dades perdudes.
- En els gràfics que segueixen un patró també aparentment es manté la tendència tot i perdre observacions.

En general es veu que per MCAR fins i tot agafant pèrdues de dades de fins al 60 % no s'observen casos molt preocupants a nivell visual (una altra cosa és a nivell de càlcul que es veu més endavant).

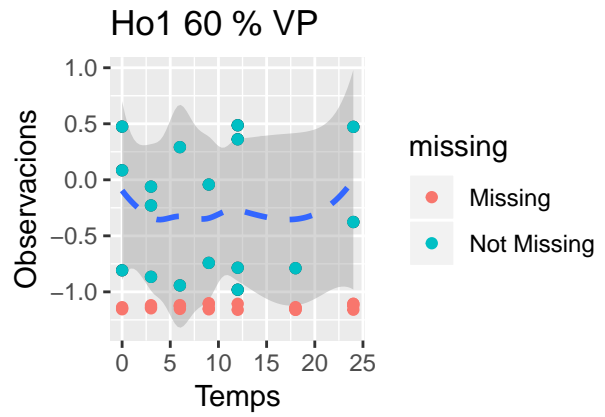
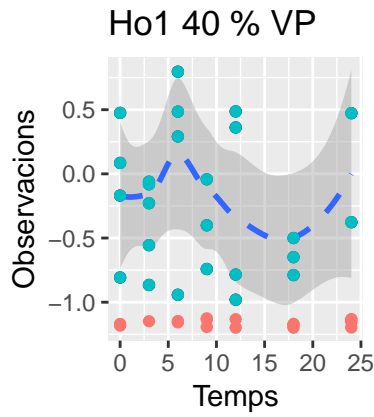
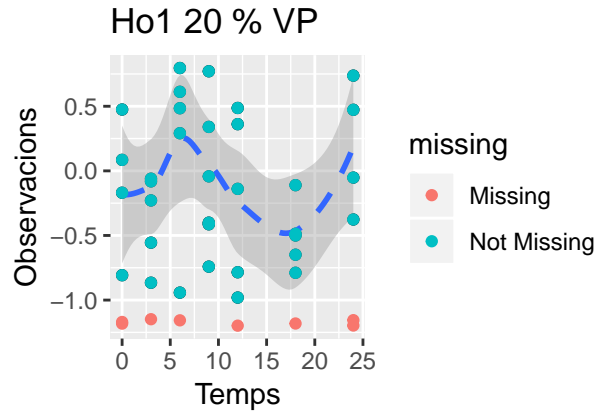
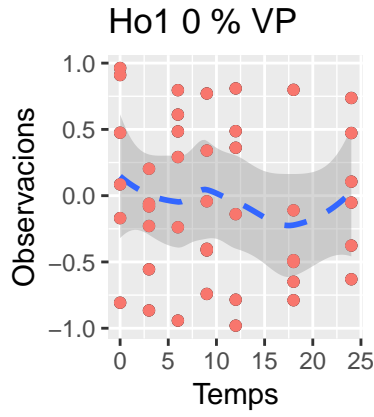
```
labs_1 <- matrix(paste(rep(DS_Vars,each=length(Av_DS_MD_list)),names(Av_DS_MD_list),sep=" "),ncol=length(DS_Vars),nrow=length(Av_DS_MD_list))
labs_2 <- matrix(paste(rep(DS_Vars,each=length(Av_DS_MD_list)),"s/ Lot",names(Av_DS_MD_list),sep=" "),ncol=length(DS_Vars),nrow=length(Av_DS_MD_list))
```

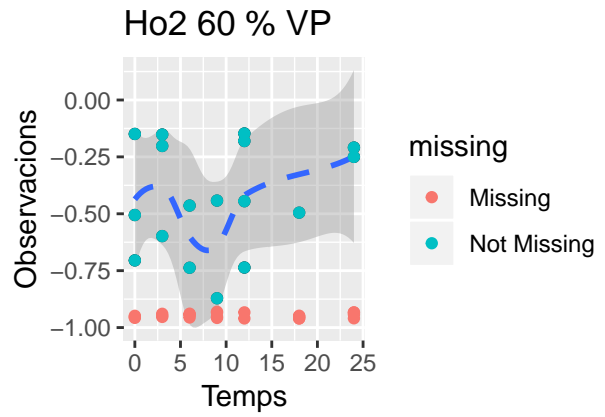
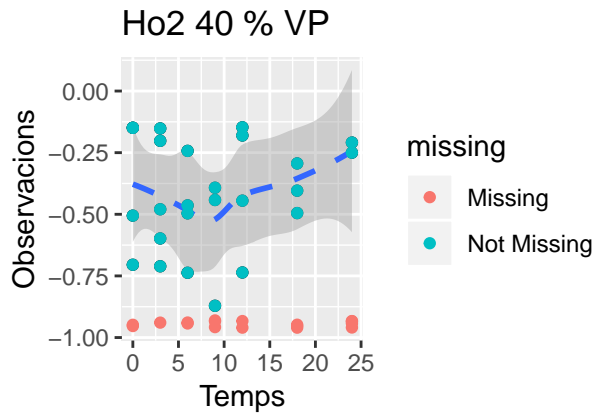
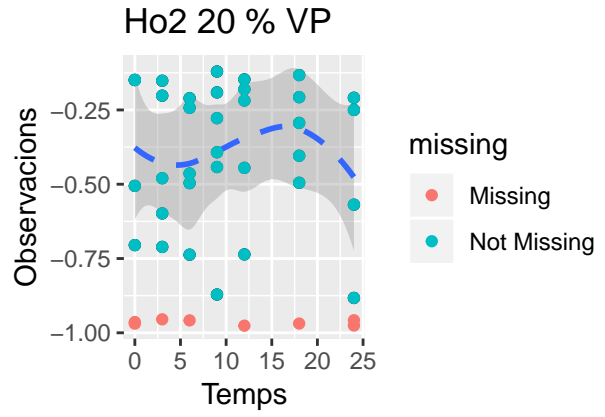
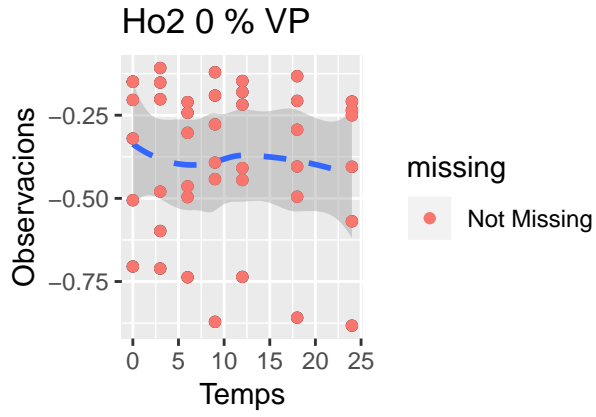
```
Av_DS_MD_plot_sc1 <- lapply(1:length(DS_Vars),function(g){
  lapply(1:length(Av_DS_MD_list),function(p){
    ggplot(Av_DS_MD_list[[p]], aes(x=Temps, y=Av_DS_MD_list[[p]][,paste(DS_Vars[g])])) + geom_point() +
      geom_smooth(linetype="dashed") + labs(title=labs_1[p,g], y = "Observacions") +
      geom_miss_point()
  })
})
```

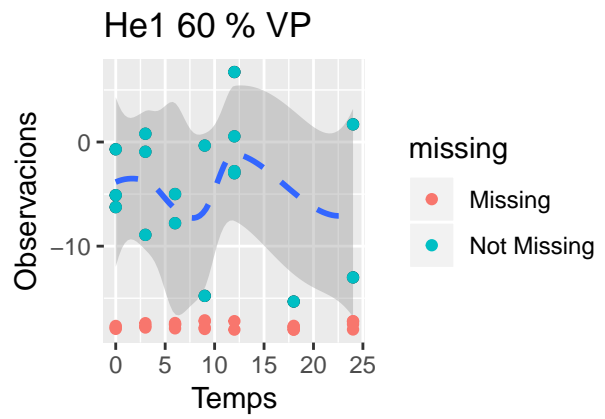
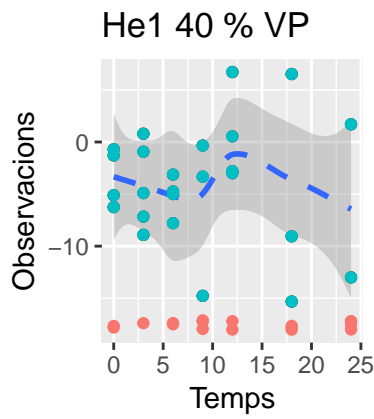
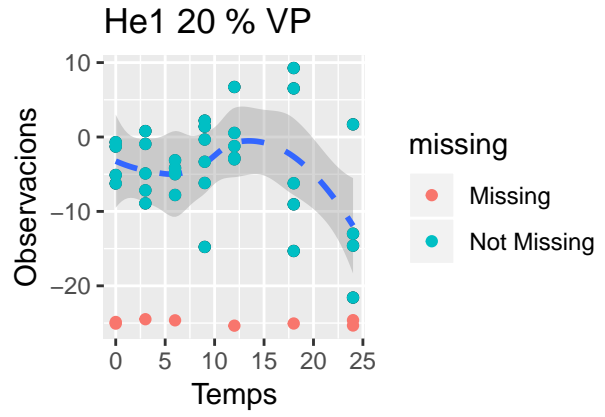
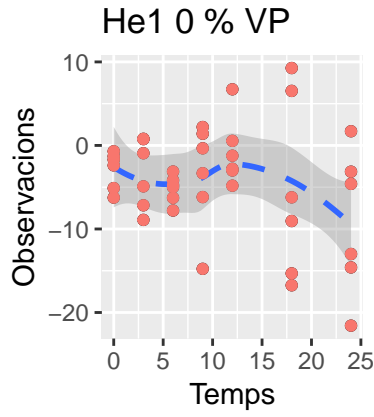
```
Av_DS_MD_plot_sc2 <- lapply(1:length(DS_Vars),function(g){
  lapply(1:length(Av_DS_MD_list),function(p){
    ggplot(Av_DS_MD_list[[p]], aes(x=Temps, y=Av_DS_MD_list[[p]][,paste(DS_Vars[g])], color=Lot, group=Lot)) +
      geom_point() + geom_smooth(method="lm",se=F) +
      labs(title=labs_2[p,g], y = "Observacions") +
      geom_miss_point()
  })
})
```

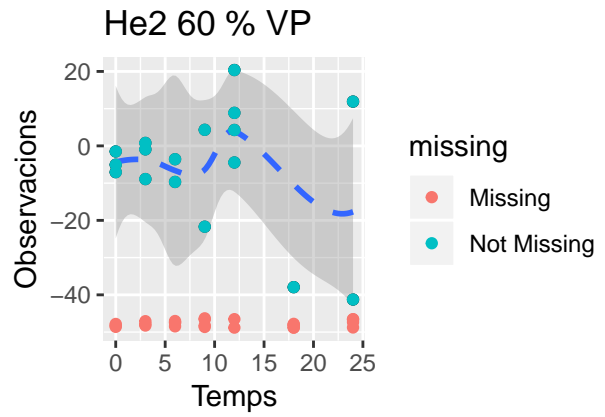
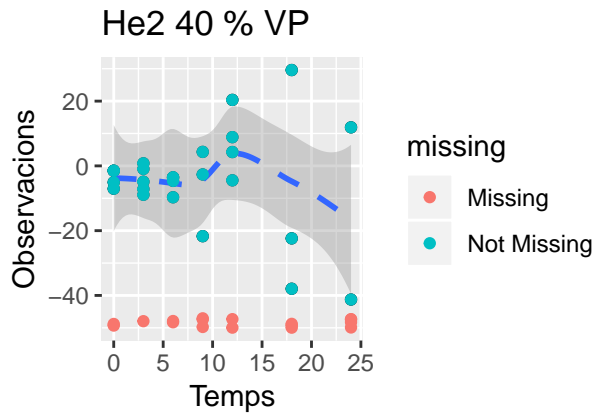
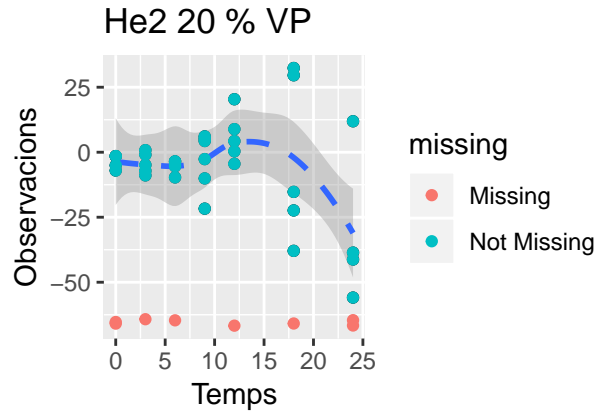
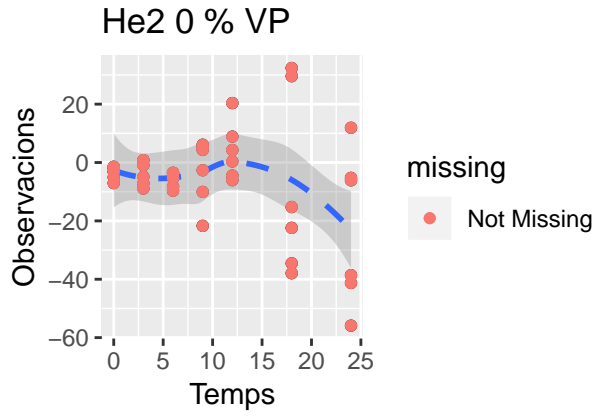
```
Av_DS_MD_ydens_plot <- lapply(1:length(DS_Vars),function(g){
  lapply(1:length(Av_DS_MD_list),function(p){
    ggplot(Av_DS_MD_list[[p]], aes(x=Av_DS_MD_list[[p]][,paste(DS_Vars[g])])) +
      geom_density(color="darkblue", fill="lightblue") +
      geom_vline(aes(xintercept=mean(DS_Vars[g])), color="blue", linetype="dashed", size=1)+
      labs(title=labs_1[p,g], x = "Observacions")
  })
})
```

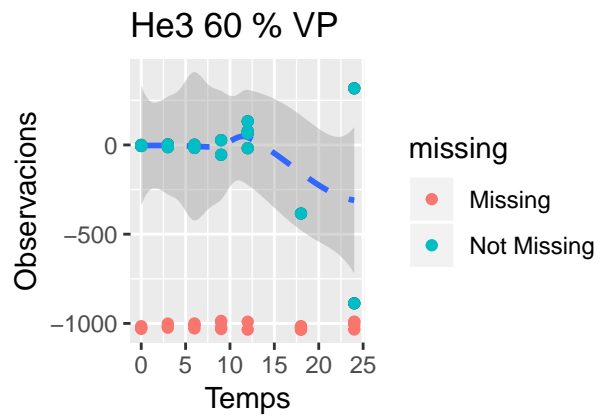
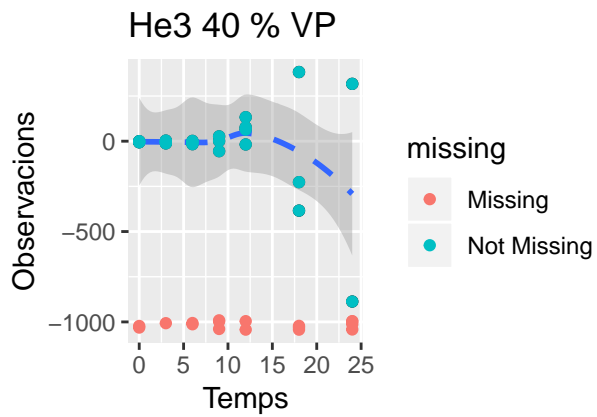
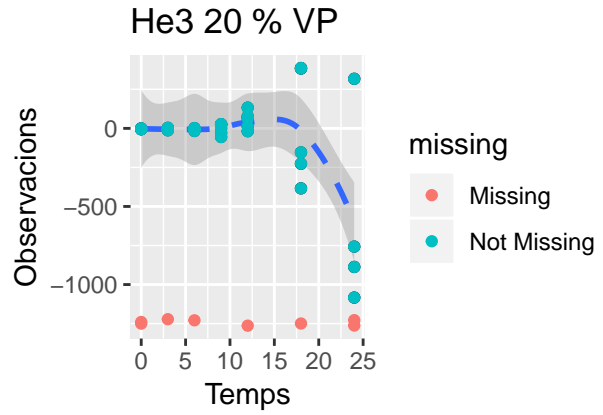
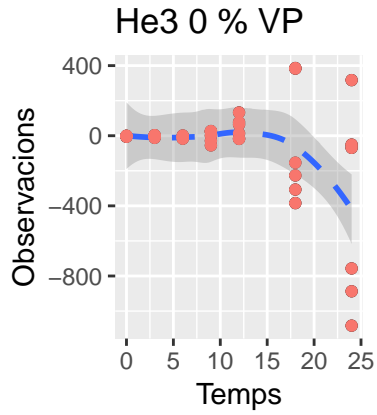
```
lapply(1:length(DS_Vars),function(g){
  grid.arrange(grobs = Av_DS_MD_plot_sc1[[g]])
})
```

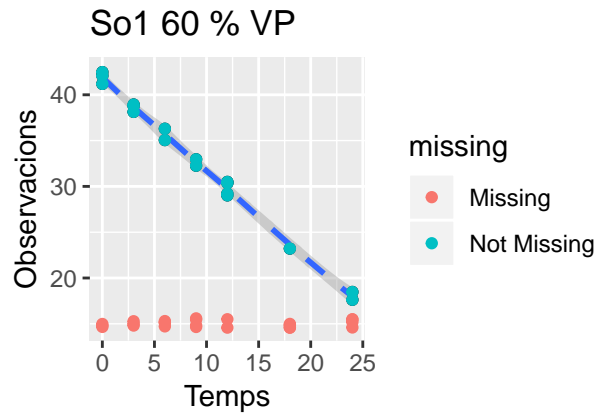
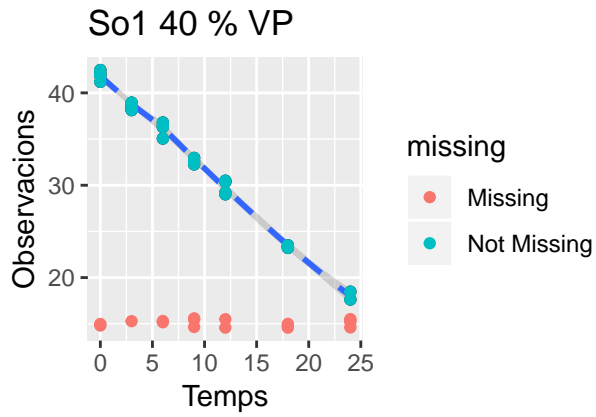
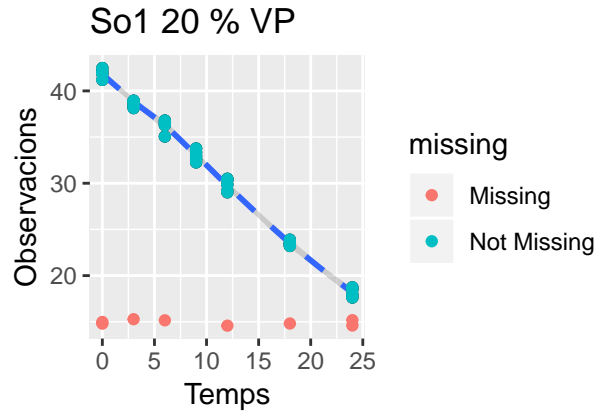
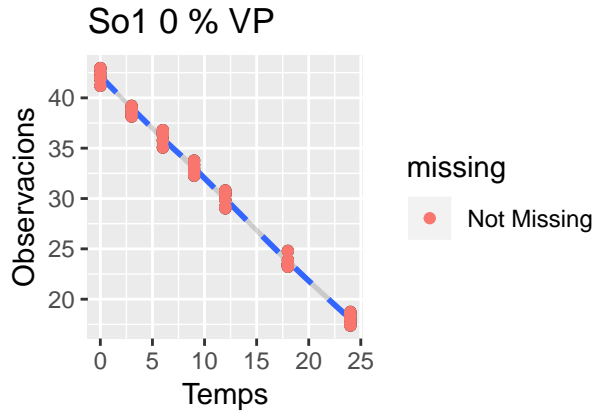


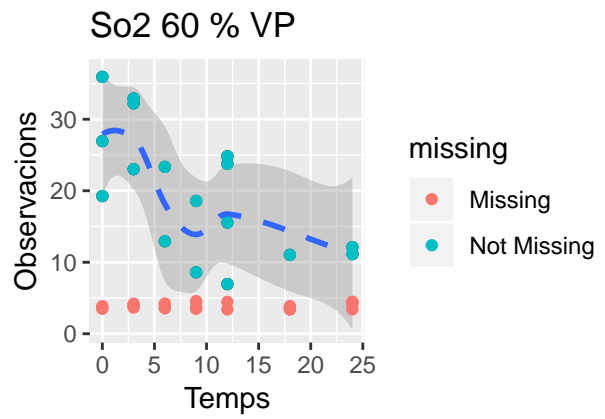
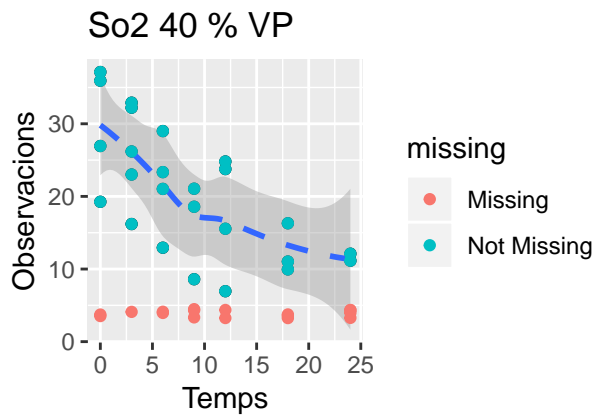
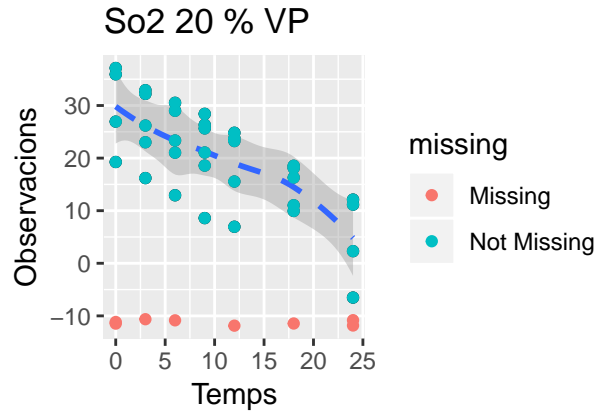
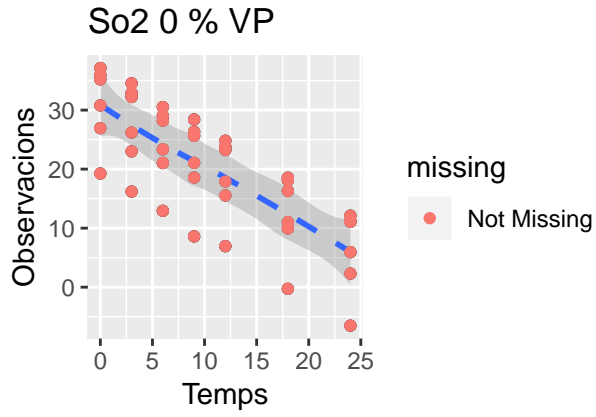


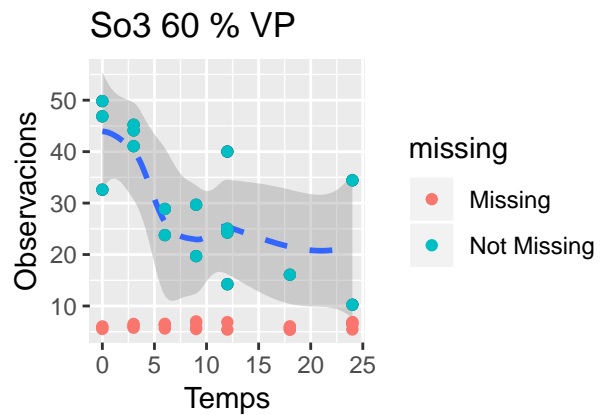
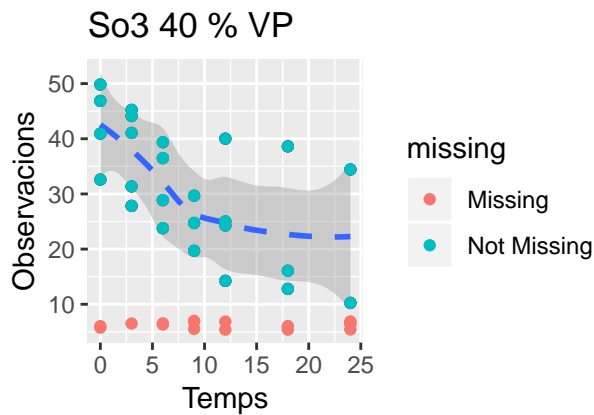
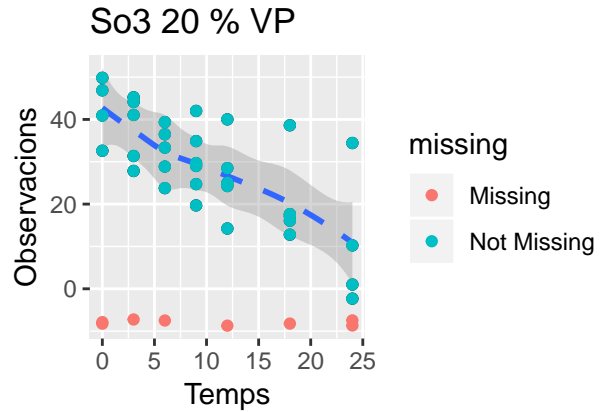
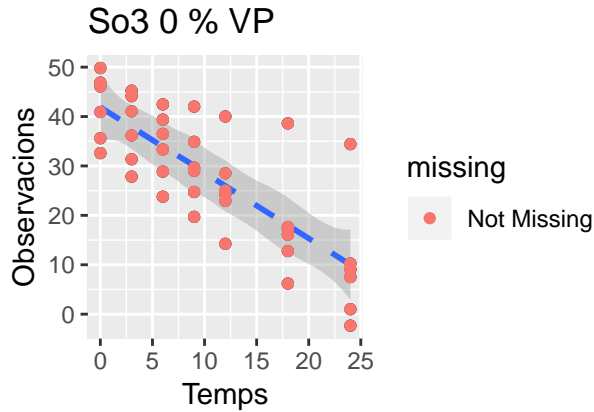










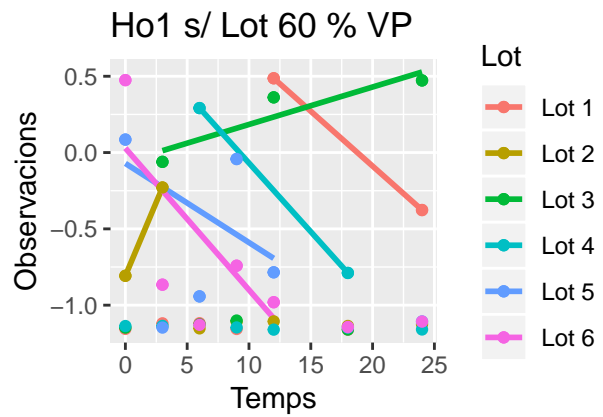
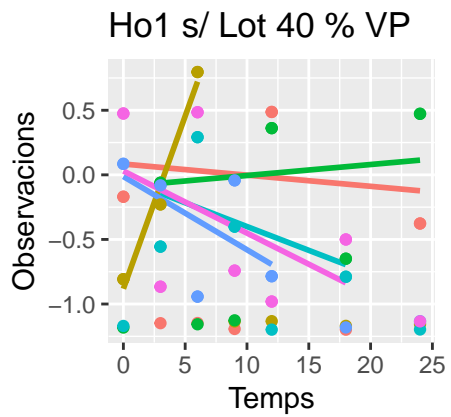
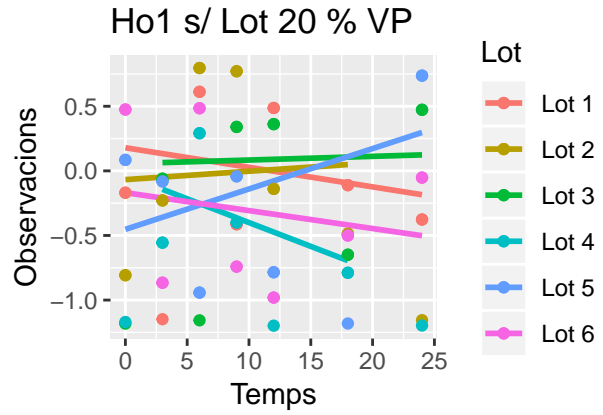
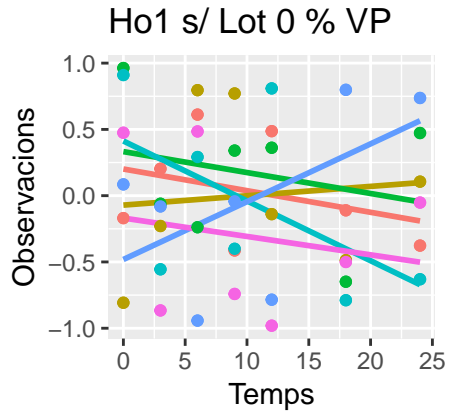


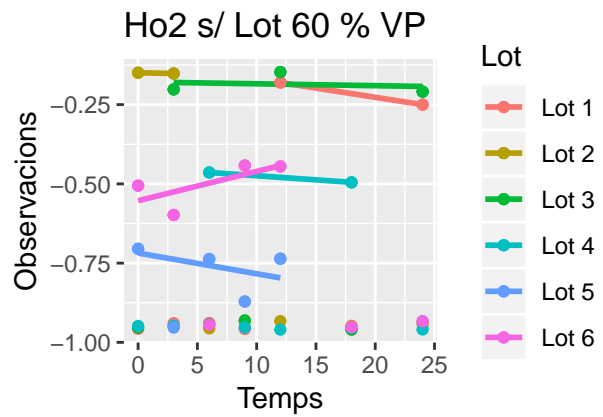
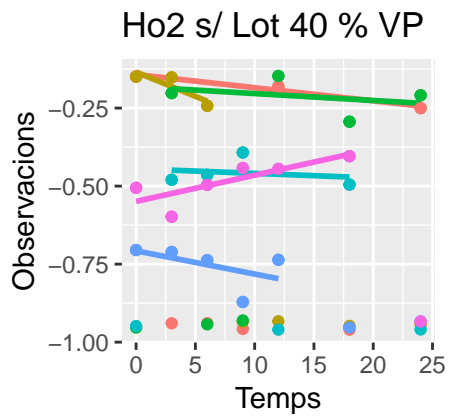
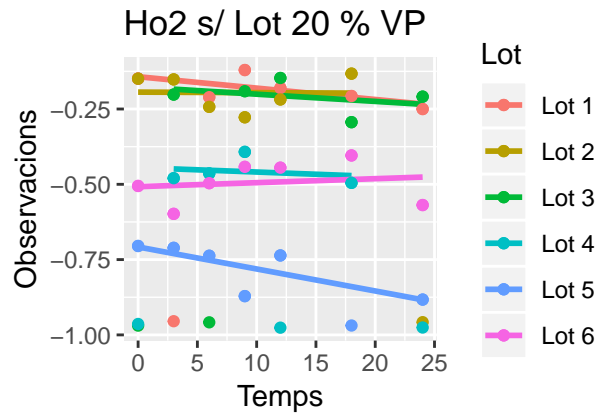
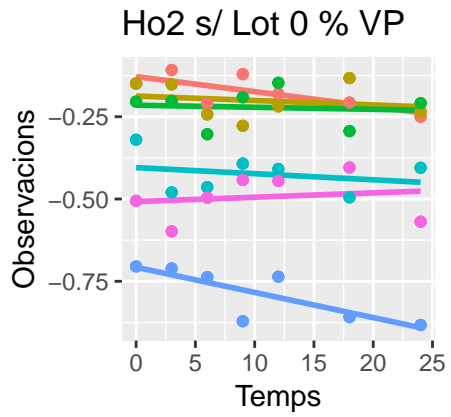
```
## [[1]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z    cells  name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[2]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z    cells  name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[3]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z    cells  name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[4]]
## TableGrob (2 x 2) "arrange": 4 grobs
```

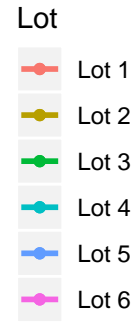
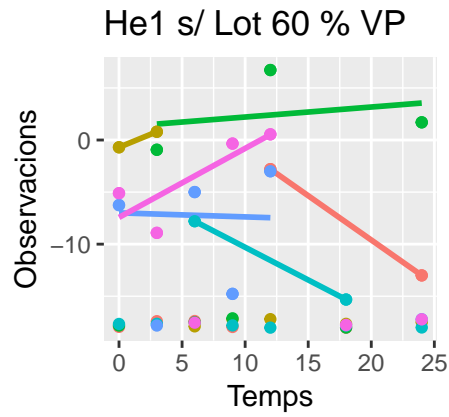
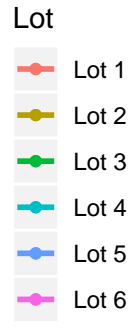
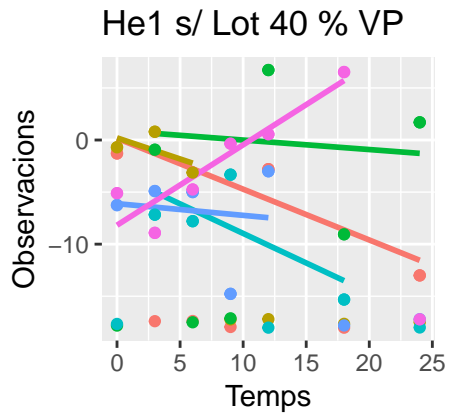
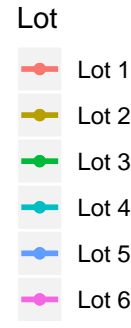
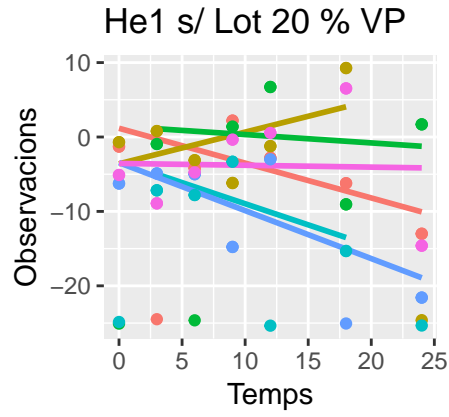
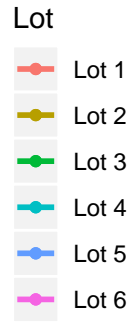
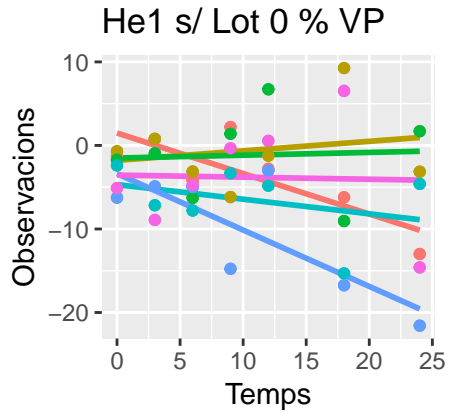
```

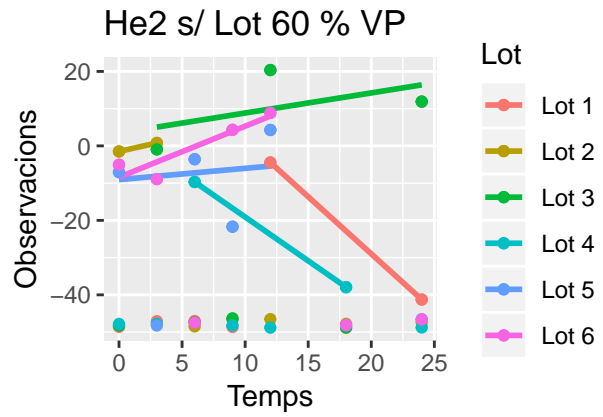
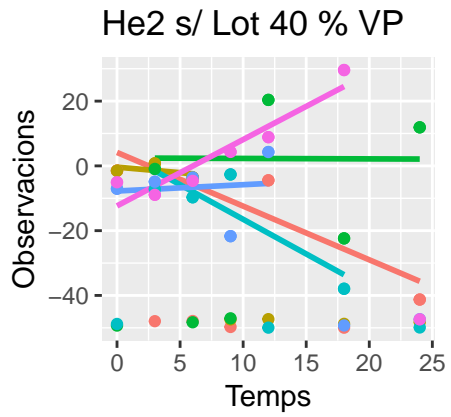
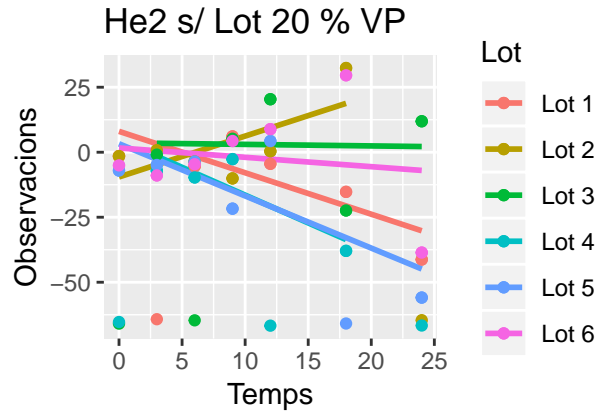
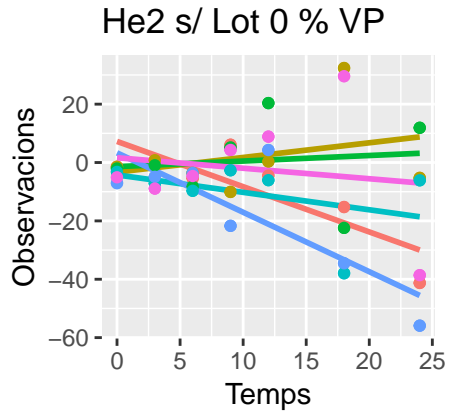
## z cells name grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[5]]
## TableGrob (2 x 2) "arrange": 4 grobs
## z cells name grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[6]]
## TableGrob (2 x 2) "arrange": 4 grobs
## z cells name grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[7]]
## TableGrob (2 x 2) "arrange": 4 grobs
## z cells name grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[8]]
## TableGrob (2 x 2) "arrange": 4 grobs
## z cells name grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
lapply(1:length(DS_Vars),function(g){ #Per aquest gràfic només prenem
  grid.arrange(grobs = Av_DS_MD_plot_sc2[[g]])
})

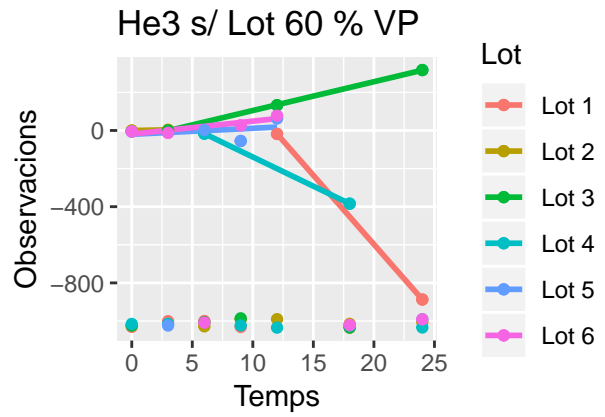
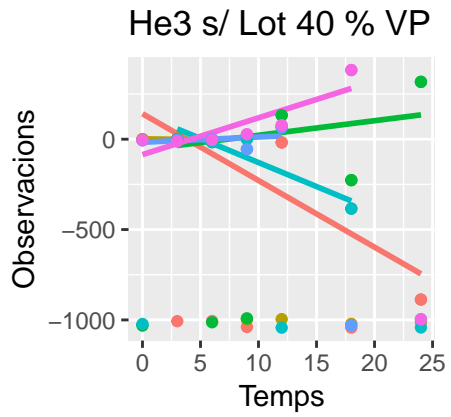
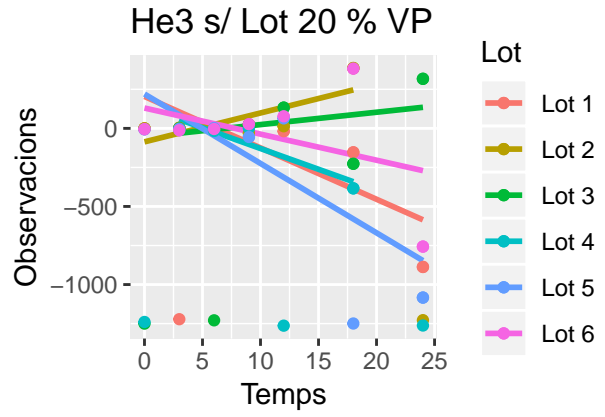
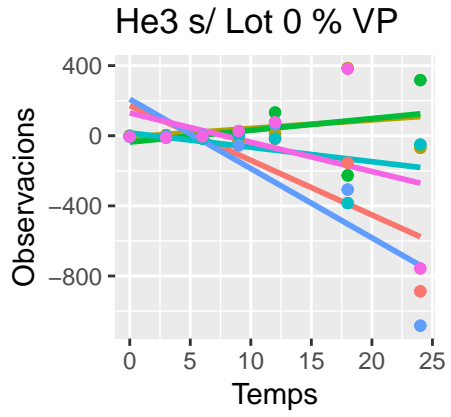
```

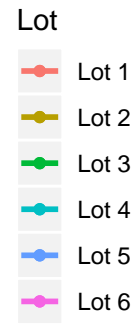
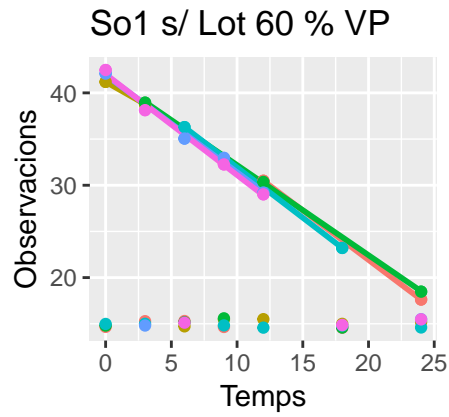
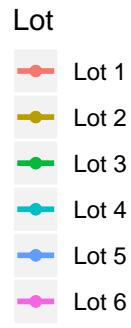
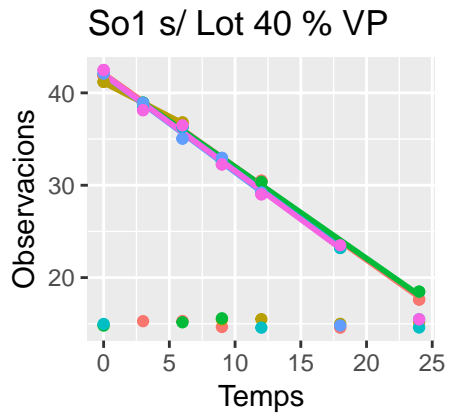
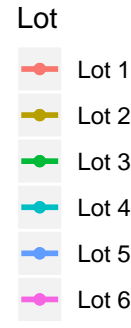
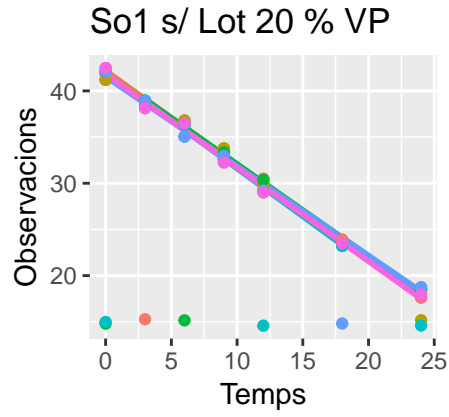
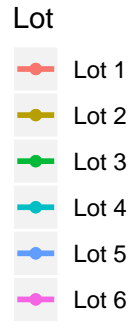
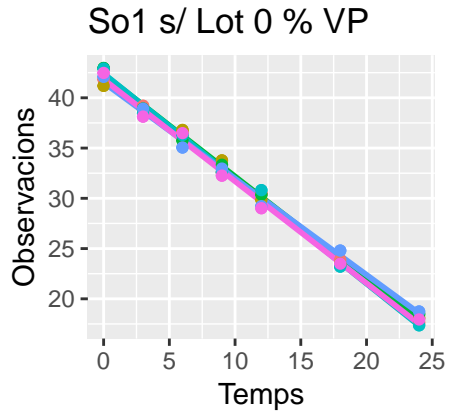


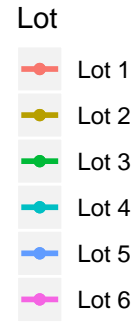
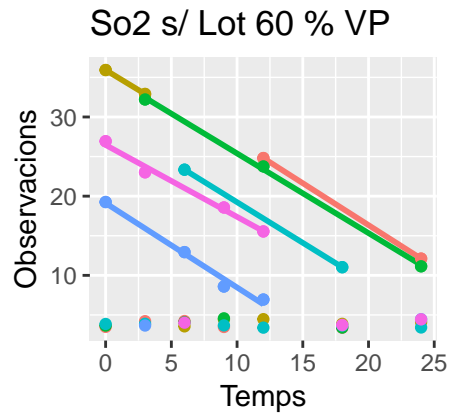
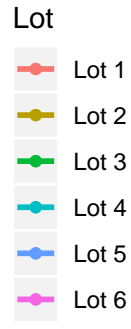
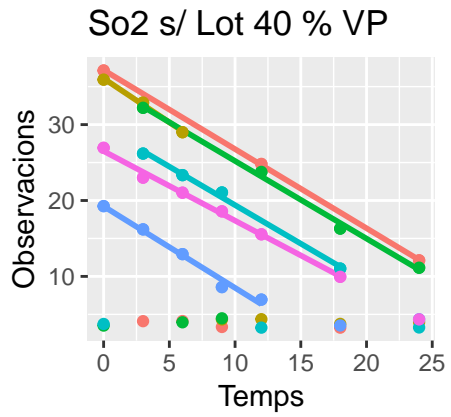
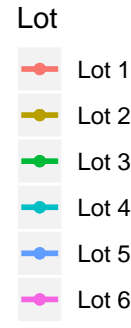
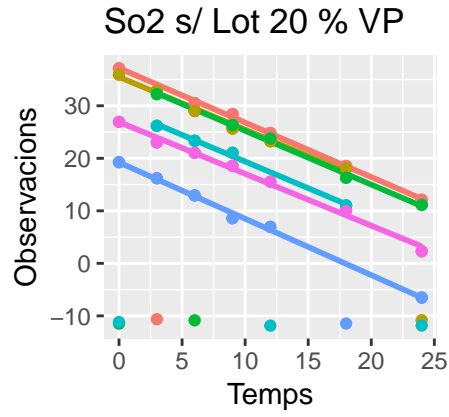
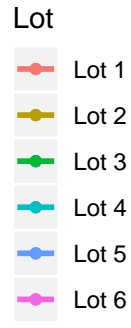
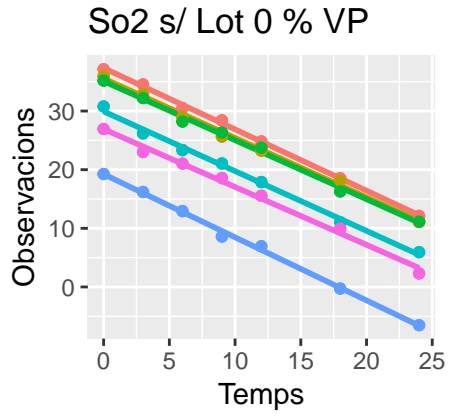


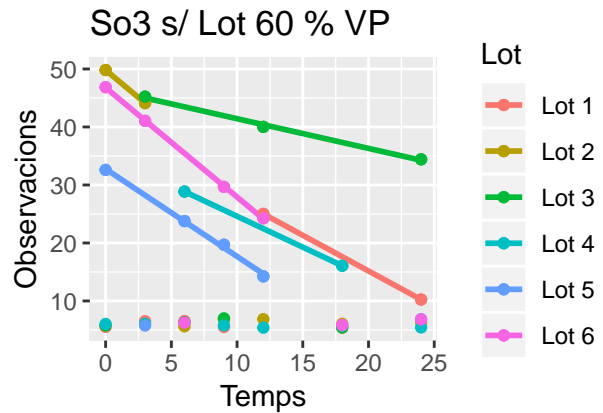
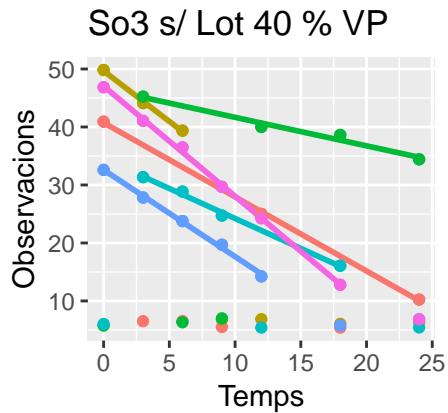
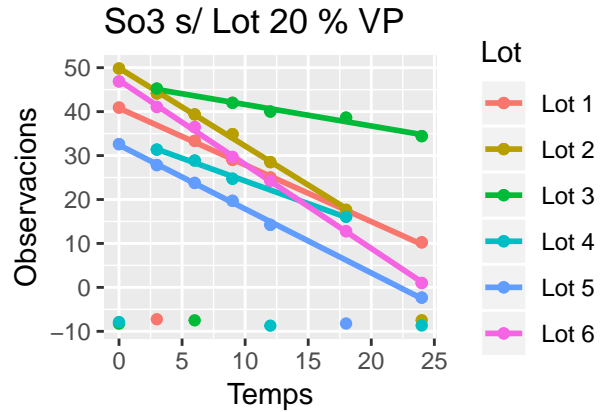
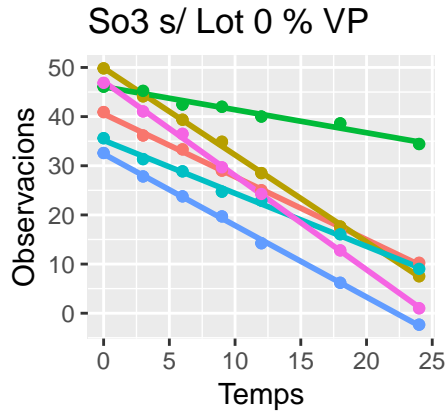












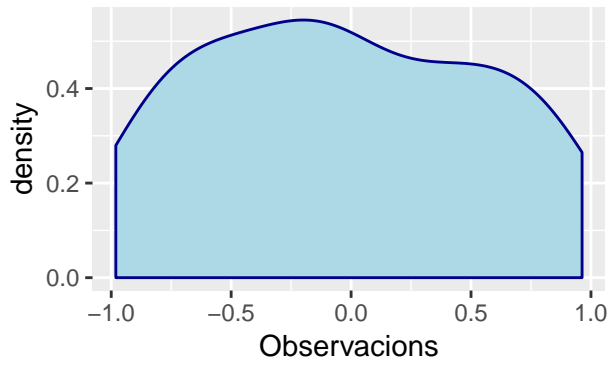
```
## [[1]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z   cells   name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[2]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z   cells   name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[3]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z   cells   name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[4]]
## TableGrob (2 x 2) "arrange": 4 grobs
```

```

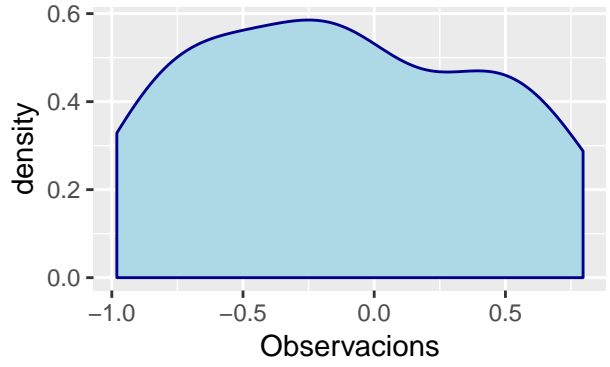
##   z     cells   name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[5]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z     cells   name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[6]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z     cells   name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[7]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z     cells   name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[8]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z     cells   name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
lapply(1:length(DS_Vars),function(g){
  grid.arrange(grobs = Av_DS_MD_ydens_plot[[g]])
})

```

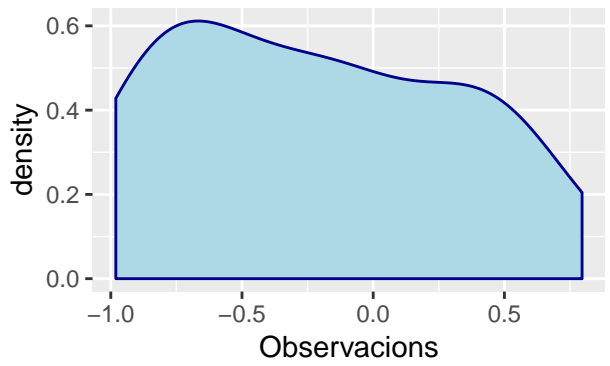

Ho1 0 % VP



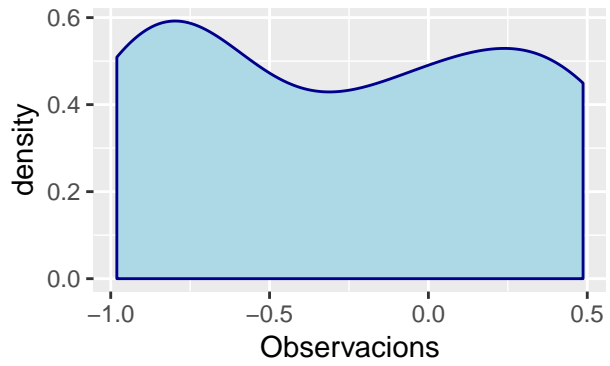
Ho1 20 % VP

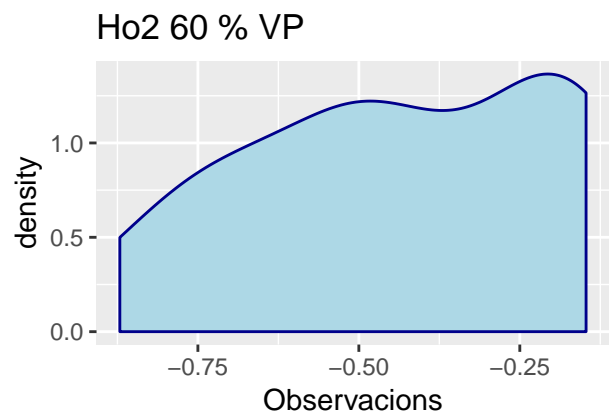
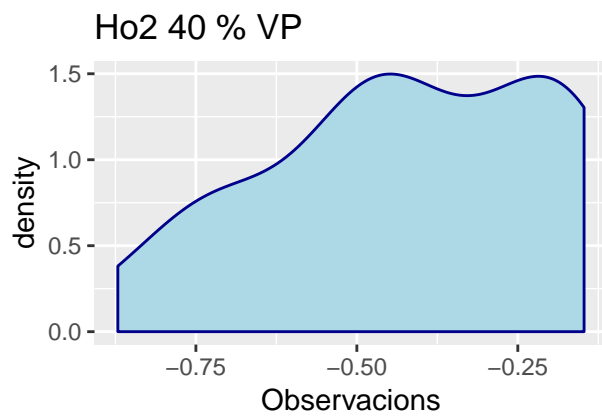
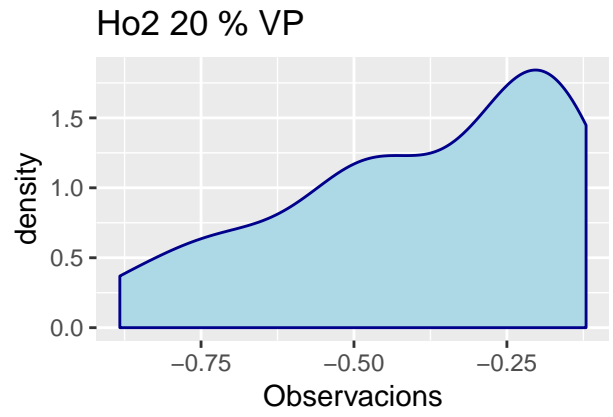
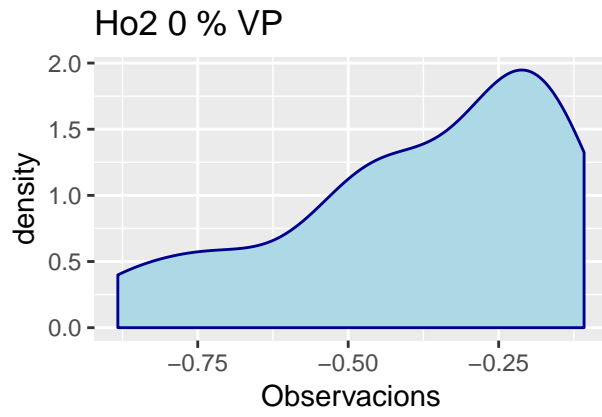


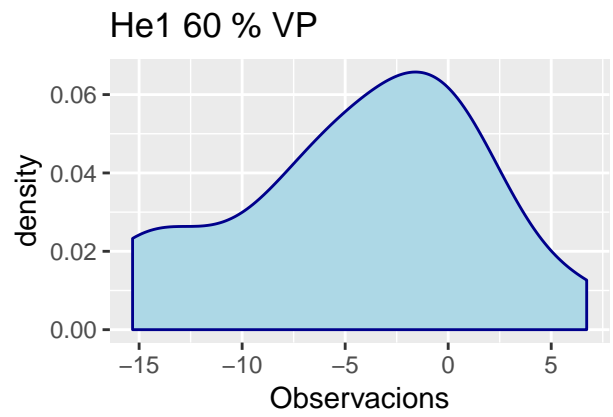
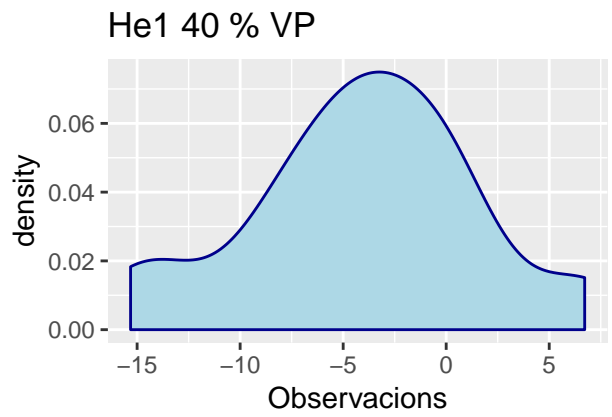
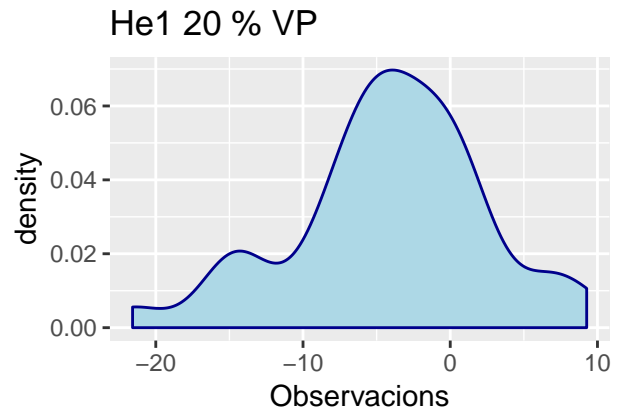
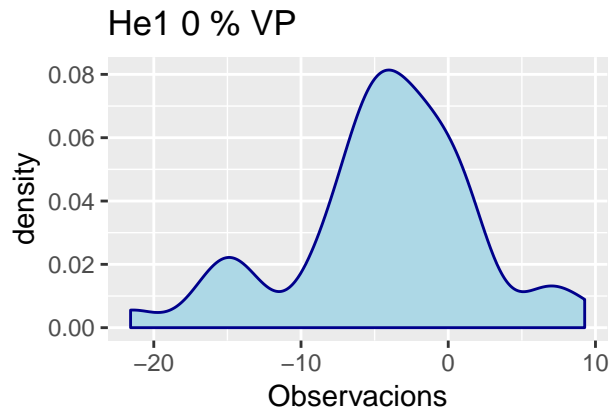
Ho1 40 % VP

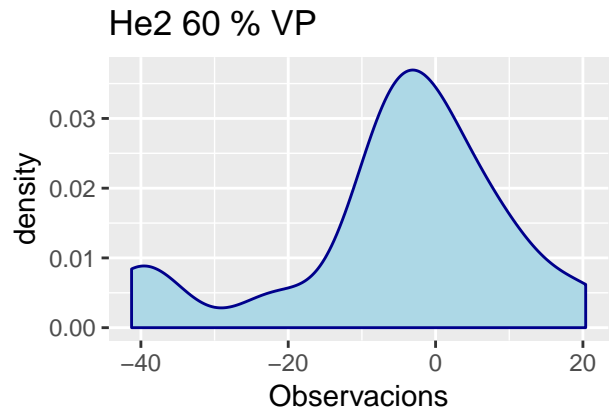
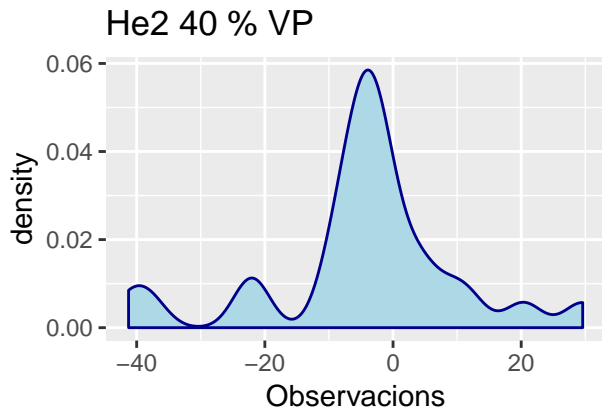
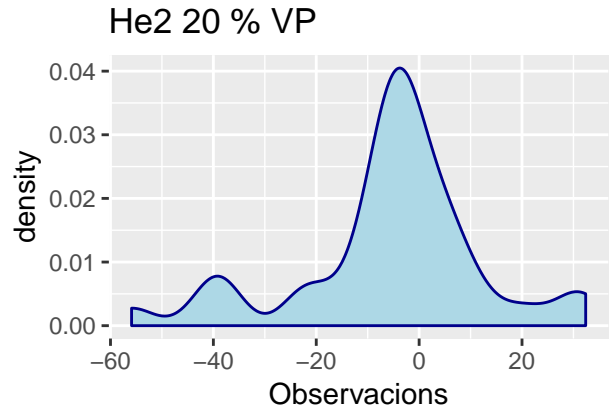
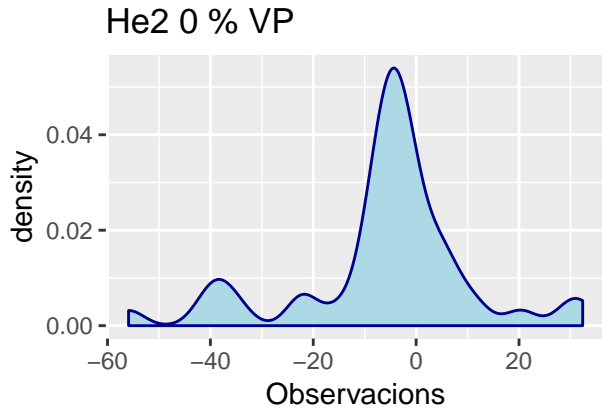


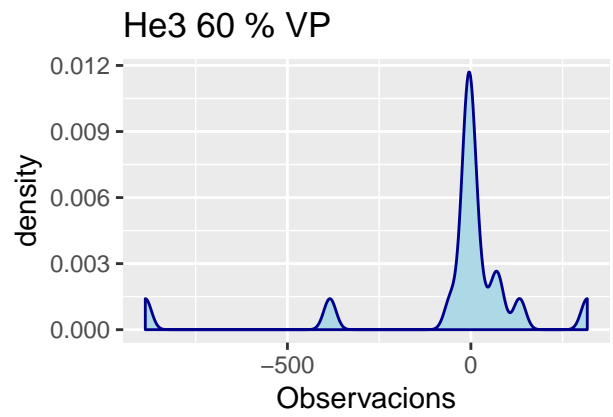
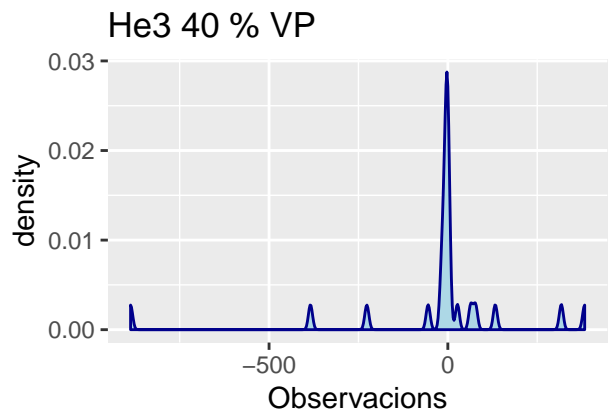
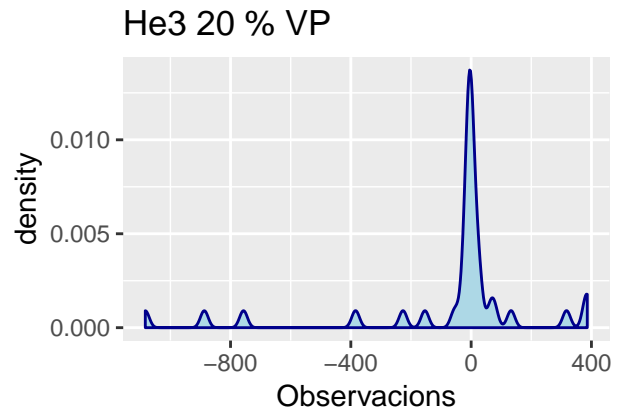
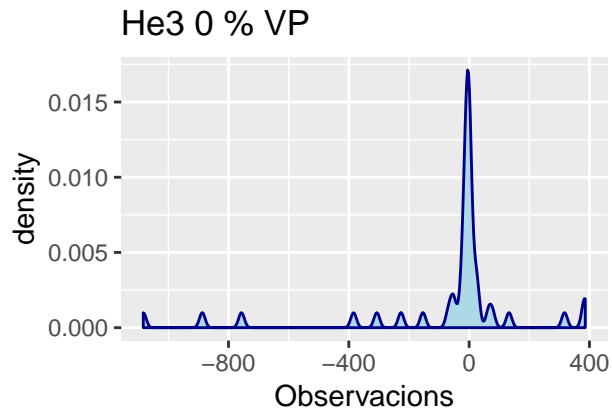
Ho1 60 % VP

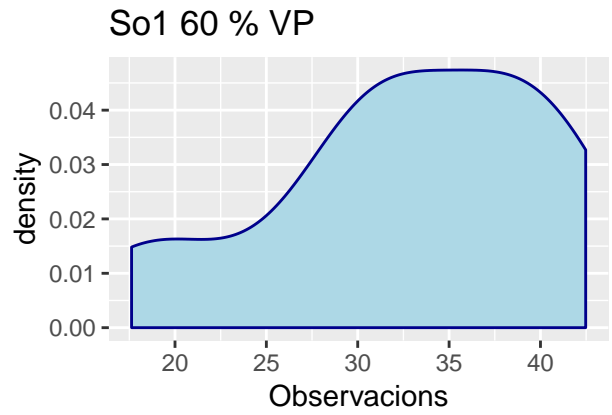
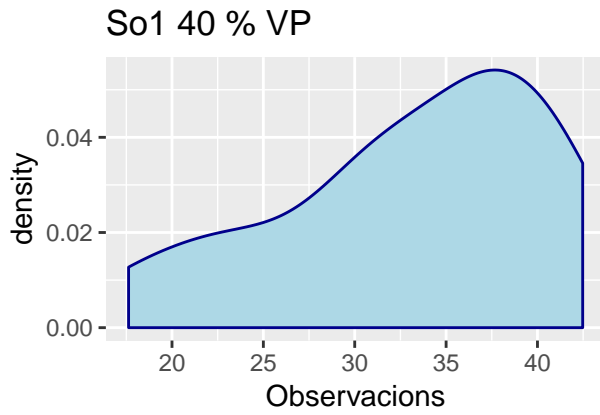
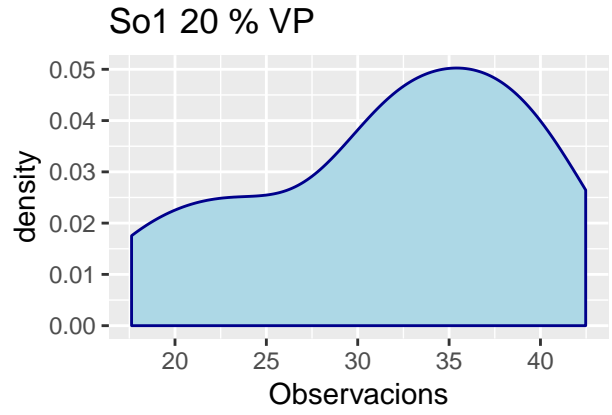
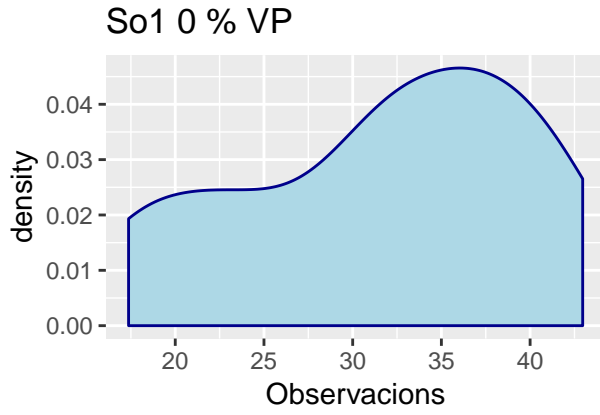


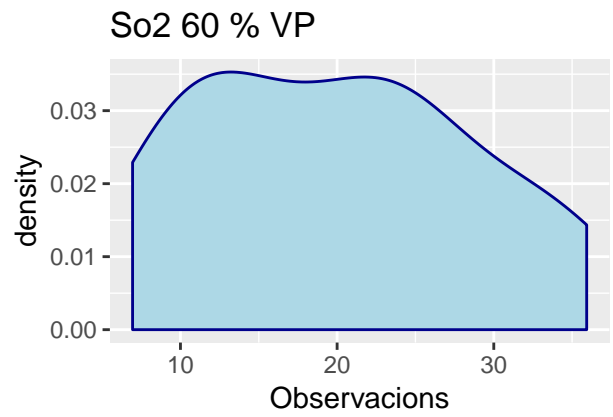
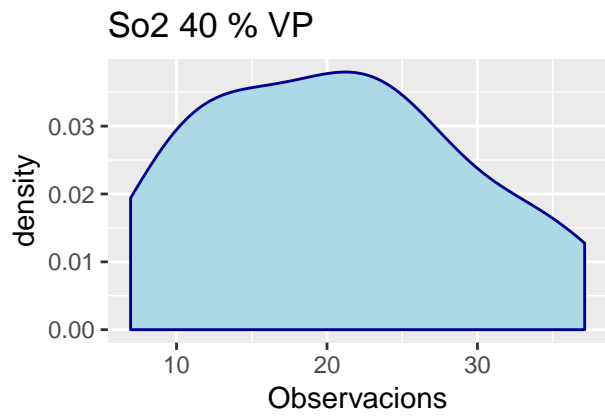
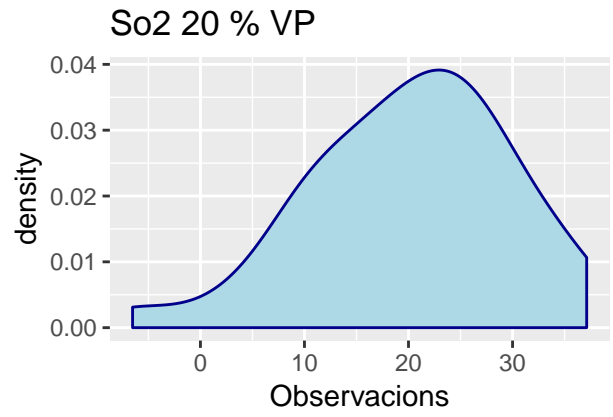
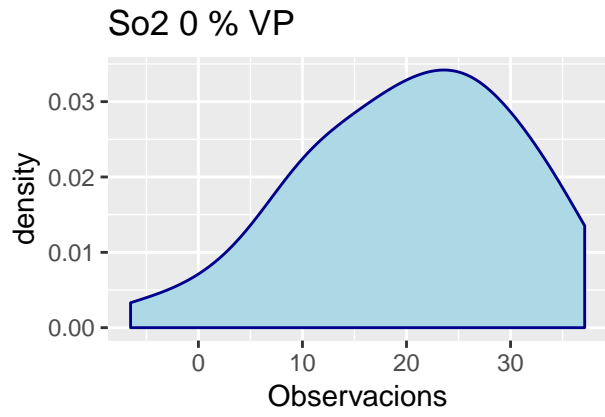


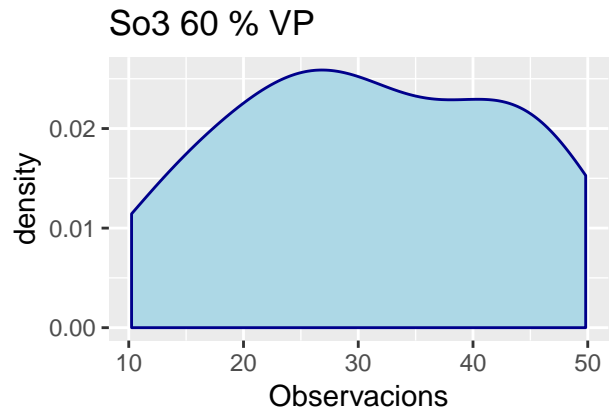
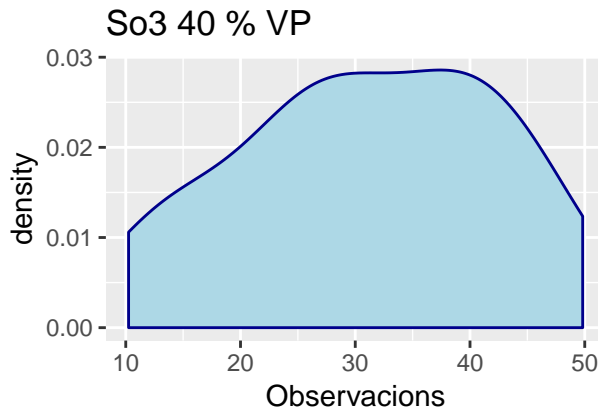
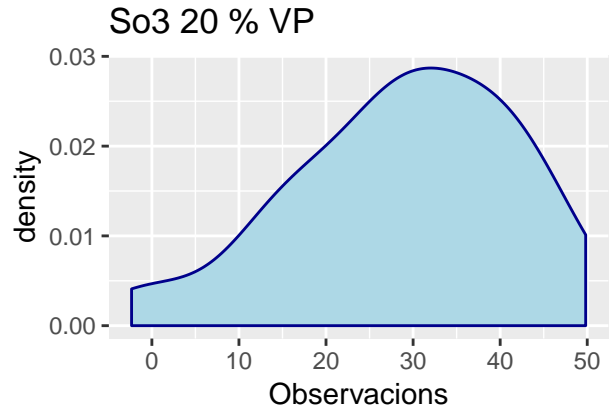
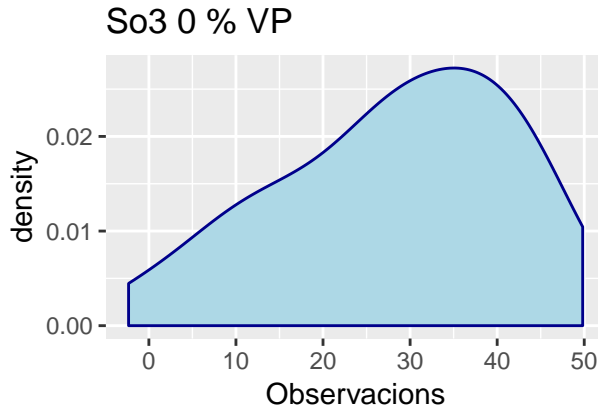












```
## [[1]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z    cells  name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[2]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z    cells  name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[3]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z    cells  name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[4]]
## TableGrob (2 x 2) "arrange": 4 grobs
```



```

## z cells name grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[5]]
## TableGrob (2 x 2) "arrange": 4 grobs
## z cells name grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[6]]
## TableGrob (2 x 2) "arrange": 4 grobs
## z cells name grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[7]]
## TableGrob (2 x 2) "arrange": 4 grobs
## z cells name grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[8]]
## TableGrob (2 x 2) "arrange": 4 grobs
## z cells name grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]

```

Dins la quantitat d'informació generada com a idees generals podria semblar que:

- Les dades perdudes afectarien menys quan les dades estan realment molt centrades.
- En els casos de mesures ja disperses potser tampoc tindrien un efecte rellevant al ja seguir disperses.
- En els casos amb patrons molt marcats, però amb dades més disperses possiblement si que podrien generar patrons diferents que falsejarien les prediccions (per exemple a So2 o Ho2).

MODELITZACIÓ AMB DADES PERDUDES

Es prenen models que hagin donat bons resultats en la modelització de models amb dades mixtes per fer els comparatius dels conjunts generats en aquesta simulació.

AJUST MODELS I CÀLCUL CONTRAST RLRT

```

Comp_I_RI_MD_Ho1 <- lapply(1:length(Av_DS_MD_list),function(p){
  Comp_I_RI_LM("Ho1",Av_DS_MD_list[[p]])

```

```

})

names(Comp_I_RI_MD_Ho1) <- names(Av_DS_MD_list)

fmHo1 <- formula(paste("Ho1", "~", 1))
Comp_I_RI_MD_Ho1_RLRT <- lapply(1:length(Av_DS_MD_list), function(p){
  eREML_Ho1<-exactRLRT(lme(fmHo1, random = ~1|Lot, data = Av_DS_MD_list[[p]], method="REML", na.action=na.omit))
})

names(Comp_I_RI_MD_Ho1_RLRT) <- names(Av_DS_MD_list)

Comp_I_RI_MD_Ho2 <- lapply(1:length(Av_DS_MD_list), function(p){
  Comp_I_RI_LM("Ho2", Av_DS_MD_list[[p]])
})

names(Comp_I_RI_MD_Ho2) <- names(Av_DS_MD_list)

fmHo2 <- formula(paste("Ho2", "~", 1))
Comp_I_RI_MD_Ho2_RLRT <- lapply(1:length(Av_DS_MD_list), function(p){
  eREML_Ho2<-exactRLRT(lme(fmHo2, random = ~1|Lot, data = Av_DS_MD_list[[p]], method="REML", na.action=na.omit))
})

names(Comp_I_RI_MD_Ho2_RLRT) <- names(Av_DS_MD_list)

#Ajustarem per varExp al ser una funció amb menys condicions que varPower encara que menys potent

fML_I_He1 <- formula(He1 ~ 1)
fRE_RI <- formula(~1|Lot)
Comp_I_RI_W_MD_He1 <- lapply(1:length(Av_DS_MD_list), function(p){
  Comp_W_LM(fML_I_He1, fRE_RI, Av_DS_MD_list[[p]], 0, 2, 0)
})
names(Comp_I_RI_W_MD_He1) <- names(Av_DS_MD_list)
Comp_I_RI_W_MD_He1_RLRT <- lapply(1:length(Av_DS_MD_list), function(p){
  eREML_He1<-exactRLRT(lme(fML_I_He1, random = ~1|Lot, data = Av_DS_MD_list[[p]], method="REML", na.action=na.omit))
})
names(Comp_I_RI_W_MD_He1_RLRT) <- names(Av_DS_MD_list)

fML_I_He2 <- formula(He2 ~ 1)
fRE_RI <- formula(~1|Lot)
Comp_I_RI_W_MD_He2 <- lapply(1:length(Av_DS_MD_list), function(p){
  Comp_W_LM(fML_I_He2, fRE_RI, Av_DS_MD_list[[p]], 0, 2, 0)
})
names(Comp_I_RI_W_MD_He2) <- names(Av_DS_MD_list)
Comp_I_RI_W_MD_He2_RLRT <- lapply(1:length(Av_DS_MD_list), function(p){
  eREML_He2<-exactRLRT(lme(fML_I_He2, random = ~1|Lot, data = Av_DS_MD_list[[p]], method="REML", na.action=na.omit))
})
names(Comp_I_RI_W_MD_He2_RLRT) <- names(Av_DS_MD_list)

fML_I_He3 <- formula(He3 ~ 1)
fRE_RI <- formula(~1|Lot)
Comp_I_RI_W_MD_He3 <- lapply(1:length(Av_DS_MD_list), function(p){
  Comp_W_LM(fML_I_He3, fRE_RI, Av_DS_MD_list[[p]], 0, 2, 0)
})
names(Comp_I_RI_W_MD_He3) <- names(Av_DS_MD_list)

```

```

Comp_I_RI_W_MD_He3_RLRT <- lapply(1:length(Av_DS_MD_list),function(p){
  eREML_He3<-exactRLRT(lme(fML_I_He3, random = ~1|Lot, data = Av_DS_MD_list[[p]], method="REML",na.action=na.omit)
})
names(Comp_I_RI_W_MD_He3_RLRT) <- names(Av_DS_MD_list)

```

#Segons els models trobats s'ajusta per So1 IS RI, So2 IS RI i So3 IS RIS. ENS centrem directament en e

```

Comp_IS_RI_MD_So1 <- lapply(1:length(Av_DS_MD_list),function(p){
  lme(So1 ~ 1 + Temps, random = ~1|Lot,control=ctrl,
      method="REML",data=Av_DS_MD_list[[p]],na.action=na.omit)
})
names(Comp_IS_RI_MD_So1) <- names(Av_DS_MD_list)
Comp_IS_RI_MD_So1_RLRT <- lapply(1:length(Av_DS_MD_list),function(p){
  eREML_So1<-exactRLRT(Comp_IS_RI_MD_So1[[p]])
})
names(Comp_IS_RI_MD_So1_RLRT) <- names(Av_DS_MD_list)

```

```

Comp_IS_RI_MD_So2 <- lapply(1:length(Av_DS_MD_list),function(p){
  lme(So3 ~ 1 + Temps, random = ~1|Lot,control=ctrl,
      method="REML",data=Av_DS_MD_list[[p]],na.action=na.omit)
})
names(Comp_IS_RI_MD_So2) <- names(Av_DS_MD_list)
Comp_IS_RI_MD_So2_RLRT <- lapply(1:length(Av_DS_MD_list),function(p){
  eREML_So2<-exactRLRT(Comp_IS_RI_MD_So2[[p]])
})
names(Comp_IS_RI_MD_So2_RLRT) <- names(Av_DS_MD_list)

```

#En el cas de múltiples efectes aleatoris fem el resultat de l'anova de comparació entre el model RI i

```

Comp_IS_RI_MD_So3 <- lapply(1:length(Av_DS_MD_list),function(p){
  lme(So3 ~ 1 + Temps, random = ~1|Lot,control=ctrl,
      method="REML",data=Av_DS_MD_list[[p]],na.action=na.omit)
})
names(Comp_IS_RI_MD_So3) <- names(Av_DS_MD_list)
Comp_IS_RIS_MD_So3 <- lapply(1:length(Av_DS_MD_list),function(p){
  lme(So3 ~ 1 + Temps, random = ~1+Temps|Lot,control=ctrl,
      method="REML",data=Av_DS_MD_list[[p]],na.action=na.omit)
})
names(Comp_IS_RIS_MD_So3) <- names(Av_DS_MD_list)

Comp_IS_RIS_MD_So3_RLRT <- lapply(1:length(Av_DS_MD_list),function(p){
  anova(Comp_IS_RI_MD_So3[[p]],Comp_IS_RIS_MD_So3[[p]])
})
names(Comp_IS_RIS_MD_So3_RLRT) <- names(Av_DS_MD_list)
Comp_IS_RI_MD_So3_RLRT <- lapply(1:length(Av_DS_MD_list),function(p){
  eREML_So3<-exactRLRT(Comp_IS_RI_MD_So3[[p]])
})
names(Comp_IS_RI_MD_So3_RLRT) <- names(Av_DS_MD_list)

```

AIC/BIC/Sigma

```

Ho1_MD_R <- ConvList(Comp_I_RI_MD_Ho1)
s<-seq(1:3)

```

```

repeat {
  s<-c(s,s[seq(from=(length(s)-2),to=length(s))]+6)
  if (length(s)>=length(Ho1_MD_R)/2){
    break
  }
}
Ho1_MD_R <- Ho1_MD_R[s]
names(Ho1_MD_R) <- paste(rep(names(Comp_I_RI_MD_Ho1),each=3),names(Ho1_MD_R))

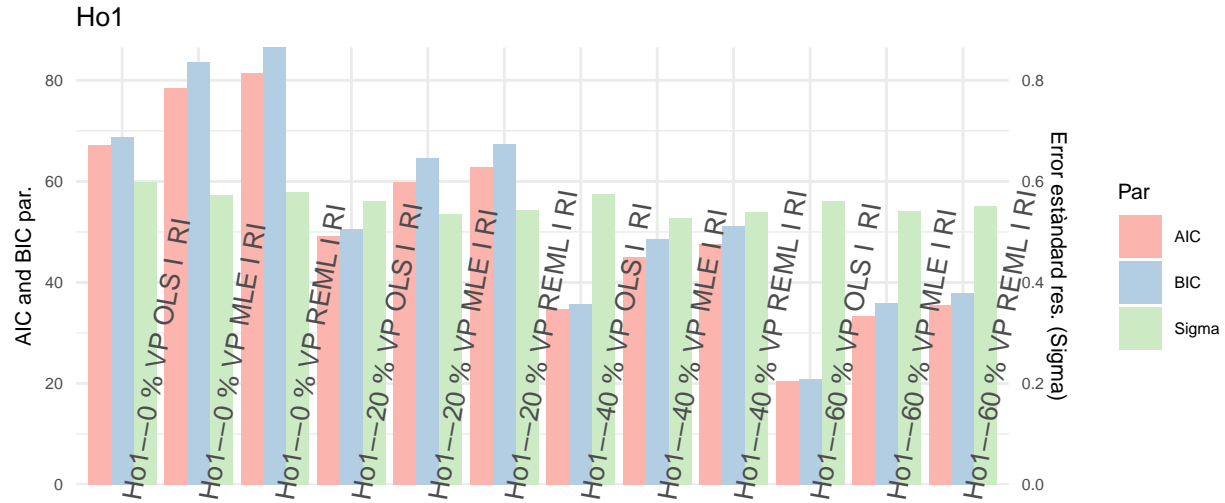
ML_Resum(Ho1_MD_R,"Ho1")

```

```

## [[1]]
##
##           Sigma      AIC      BIC
## Ho1--0 % VP OLS I  RI  0.5987 67.2228 68.8063
## Ho1--0 % VP MLE I  RI   0.5727 78.3682 83.5812
## Ho1--0 % VP REML I RI   0.5796 81.3709 86.5116
## Ho1--20 % VP OLS I  RI  0.5617 49.1585 50.4908
## Ho1--20 % VP MLE I  RI   0.5349 59.9400 64.5191
## Ho1--20 % VP REML I RI   0.5429 62.8651 67.3546
## Ho1--40 % VP OLS I  RI  0.5741 34.8299 35.7744
## Ho1--40 % VP MLE I  RI   0.5281 45.0254 48.6820
## Ho1--40 % VP REML I RI   0.5390 47.6630 51.1972
## Ho1--60 % VP OLS I  RI  0.5605 20.4814 20.8793
## Ho1--60 % VP MLE I  RI   0.5415 33.3869 35.8866
## Ho1--60 % VP REML I RI   0.5508 35.5645 37.8822
##
## [[2]]

```



Model

```

Ho2_MD_R <- ConvList(Comp_I_RI_MD_Ho2)
s<-seq(1:3)
repeat {
  s<-c(s,s[seq(from=(length(s)-2),to=length(s))]+6)
  if (length(s)>=length(Ho2_MD_R)/2){
    break
  }
}
Ho2_MD_R <- Ho2_MD_R[s]
names(Ho2_MD_R) <- paste(rep(names(Comp_I_RI_MD_Ho2),each=3),names(Ho2_MD_R))

ML_Resum(Ho2_MD_R,"Ho2")

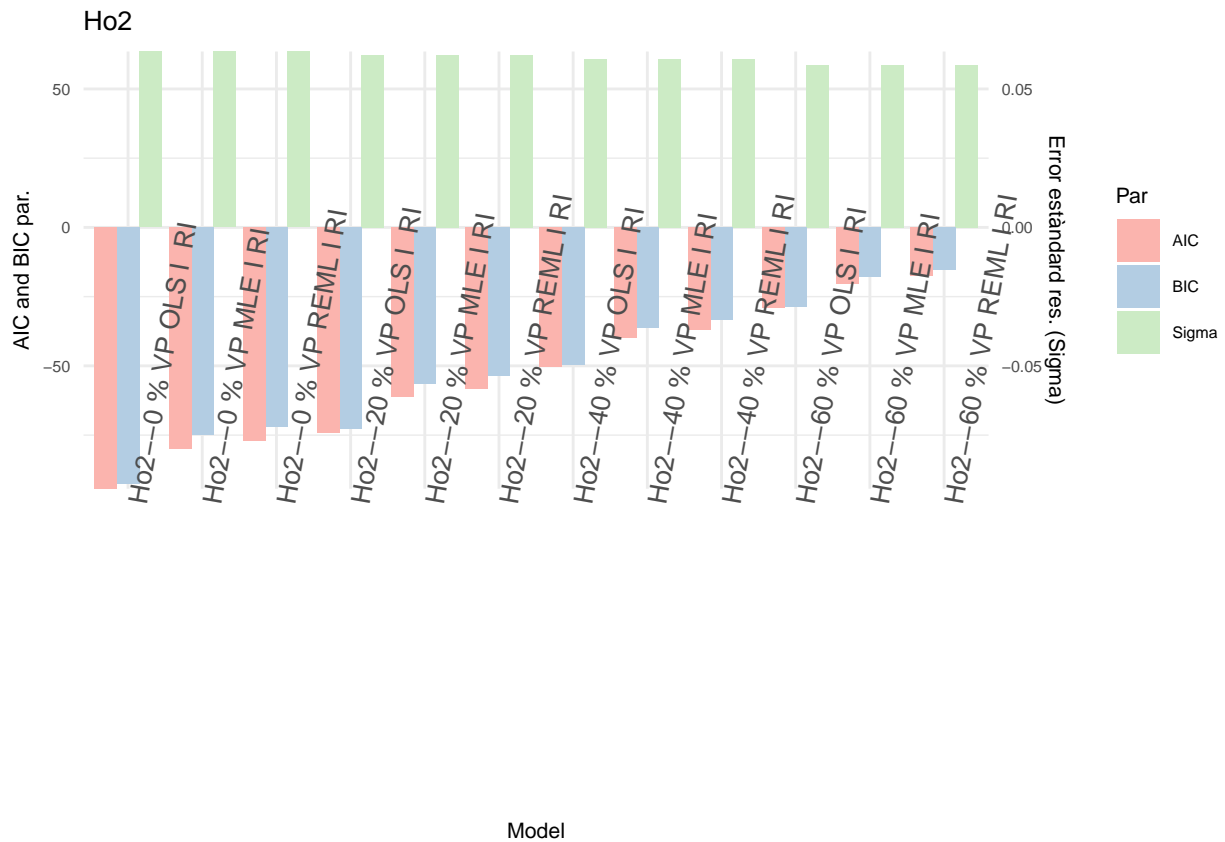
```

```

## [[1]]
##           Sigma      AIC      BIC
## Ho2--0 % VP OLS I RI  0.0635 -94.3562 -92.7727
## Ho2--0 % VP MLE I RI  0.0635 -80.0659 -74.8528
## Ho2--0 % VP REML I RI 0.0635 -77.1316 -71.9909
## Ho2--20 % VP OLS I RI  0.0622 -74.0342 -72.7019
## Ho2--20 % VP MLE I RI  0.0622 -61.1799 -56.6008
## Ho2--20 % VP REML I RI 0.0622 -58.2336 -53.7440
## Ho2--40 % VP OLS I RI  0.0607 -50.5295 -49.5850
## Ho2--40 % VP MLE I RI  0.0607 -40.0235 -36.3669
## Ho2--40 % VP REML I RI 0.0607 -37.0101 -33.4760
## Ho2--60 % VP OLS I RI  0.0586 -29.2089 -28.8110
## Ho2--60 % VP MLE I RI  0.0585 -20.4427 -17.9431

```

```
## Ho2--60 % VP REML I RI 0.0585 -17.5448 -15.2270
##
## [[2]]
```



Comentaris:

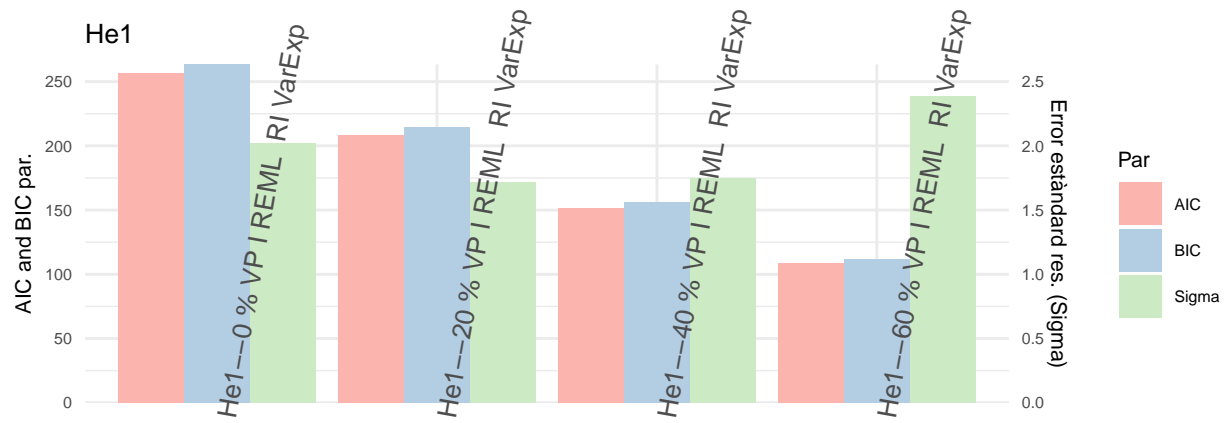
- Ho1: El núvol de punts aparentment ho segueix essent encara que eliminis valors. Les possibles variacions en el model poden ser per l'eliminació concreta que s'ha obtingut en aquesta simulació i podria ser una altra en un altre cas (es podria confirmar fent bootstrap en futurs anàlisis).
- Ho2: Aparentment al tenir més dades perdudes hi ha la mateixa error estàndard residual, però augmenta el AIC i BIC (els models es tornen més ineficients) a nivells significatius.

```
He1_MD_R <- ConvList(Comp_I_RI_W_MD_He1)
s<-seq(from=7, to=length(He1_MD_R), by=12)

He1_MD_R <- He1_MD_R[s]
names(He1_MD_R) <- paste(names(Comp_I_RI_W_MD_He1),names(He1_MD_R))

ML_Resum(He1_MD_R,"He1")
```

```
## [[1]]
##
##           Sigma      AIC      BIC
## He1--0 % VP I REML RI VarExp 2.0215 256.5681 263.4224
## He1--20 % VP I REML RI VarExp 1.7206 208.6839 214.6699
## He1--40 % VP I REML RI VarExp 1.7460 151.5126 156.2249
## He1--60 % VP I REML RI VarExp 2.3904 108.7557 111.8461
##
## [[2]]
```



Model

```

He2_MD_R <- ConvList(Comp_I_RI_W_MD_He2)
s<-seq(from=7, to=length(He2_MD_R), by=12)

He2_MD_R <- He2_MD_R[s]
names(He2_MD_R) <- paste(names(Comp_I_RI_W_MD_He2), names(He2_MD_R))

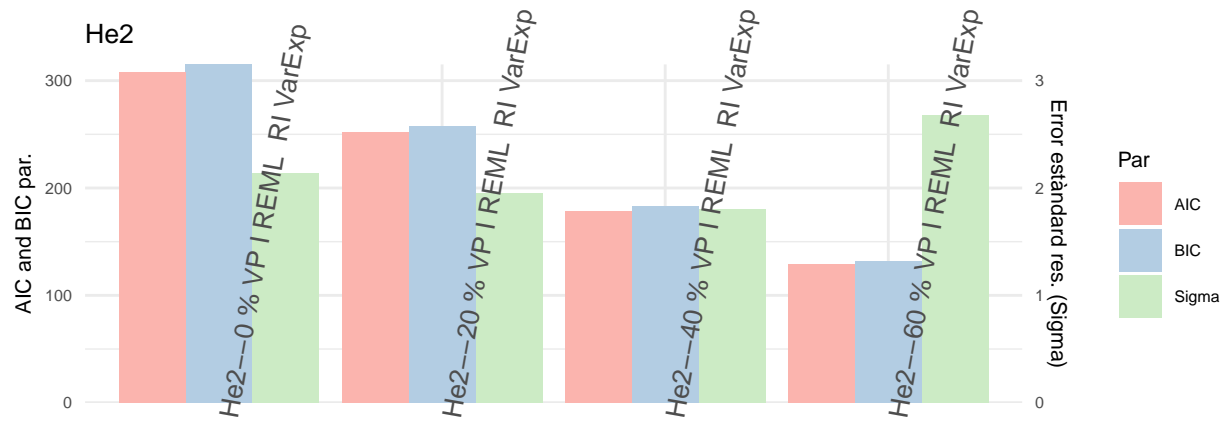
ML_Resum(He2_MD_R, "He2")

```

```

## [[1]]
##              Sigma      AIC      BIC
## He2--0 % VP I REML RI VarExp  2.1342 308.2841 315.1383
## He2--20 % VP I REML RI VarExp  1.9479 251.9775 257.9636
## He2--40 % VP I REML RI VarExp  1.8052 178.6783 183.3905
## He2--60 % VP I REML RI VarExp  2.6789 128.5652 131.6556
##
## [[2]]

```



Model

```

He3_MD_R <- ConvList(Comp_I_RI_W_MD_He3)
s<-seq(from=7, to=length(He3_MD_R), by=12)

He3_MD_R <- He3_MD_R[s]
names(He3_MD_R) <- paste(names(Comp_I_RI_W_MD_He3), names(He3_MD_R))

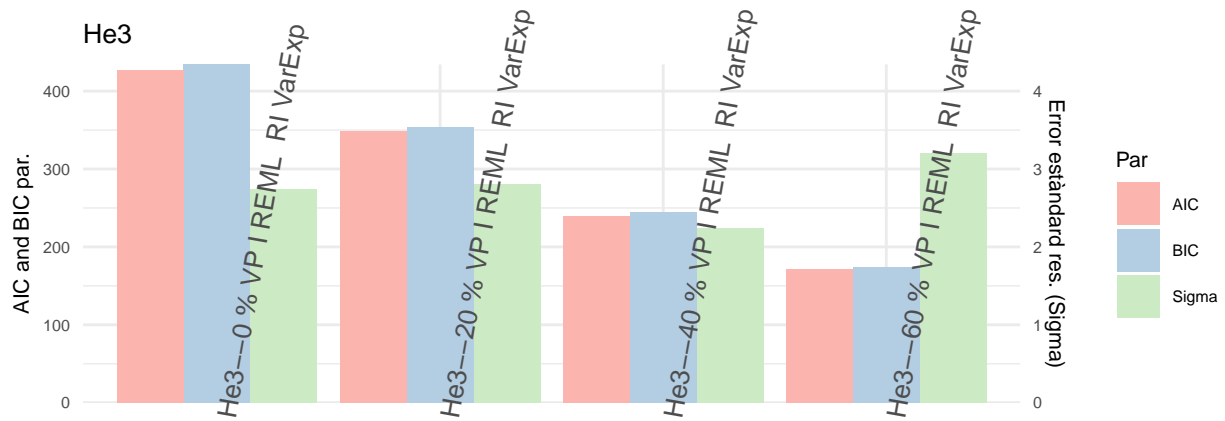
ML_Resum(He3_MD_R, "He3")

```

```

## [[1]]
##              Sigma      AIC      BIC
## He3--0 % VP I REML RI VarExp  2.7460 427.4230 434.2773
## He3--20 % VP I REML RI VarExp  2.8067 348.0627 354.0487
## He3--40 % VP I REML RI VarExp  2.2340 239.8691 244.5813
## He3--60 % VP I REML RI VarExp  3.2087 171.3179 174.4082
##
## [[2]]

```

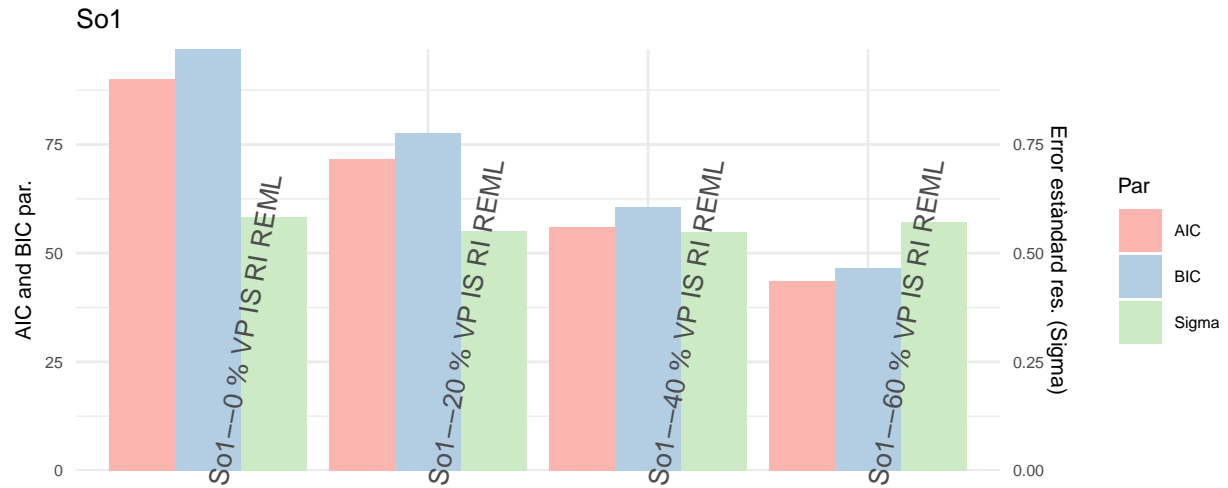
Model

Comentaris:

- He1/He2/He3: Mentre que la error estàndard residual no sembla seguir un patró concret (possiblement té a veure amb l'aleatorietat concreta que es faci), AIC i BIC semblen disminuir al tenir més dades perdudes. És contradictori, però possiblement es deu a que d'alguna manera l'eliminació de dades fa disminuir l'heterogeneïtat de valors acotant models més exactes (falsament).

```
So1_MD_R <- Comp_IS_RI_MD_So1
names(So1_MD_R) <- paste(names(Comp_IS_RI_MD_So1), c("IS RI REML"), sep=" ")
ML_Resum(So1_MD_R, "So1")
```

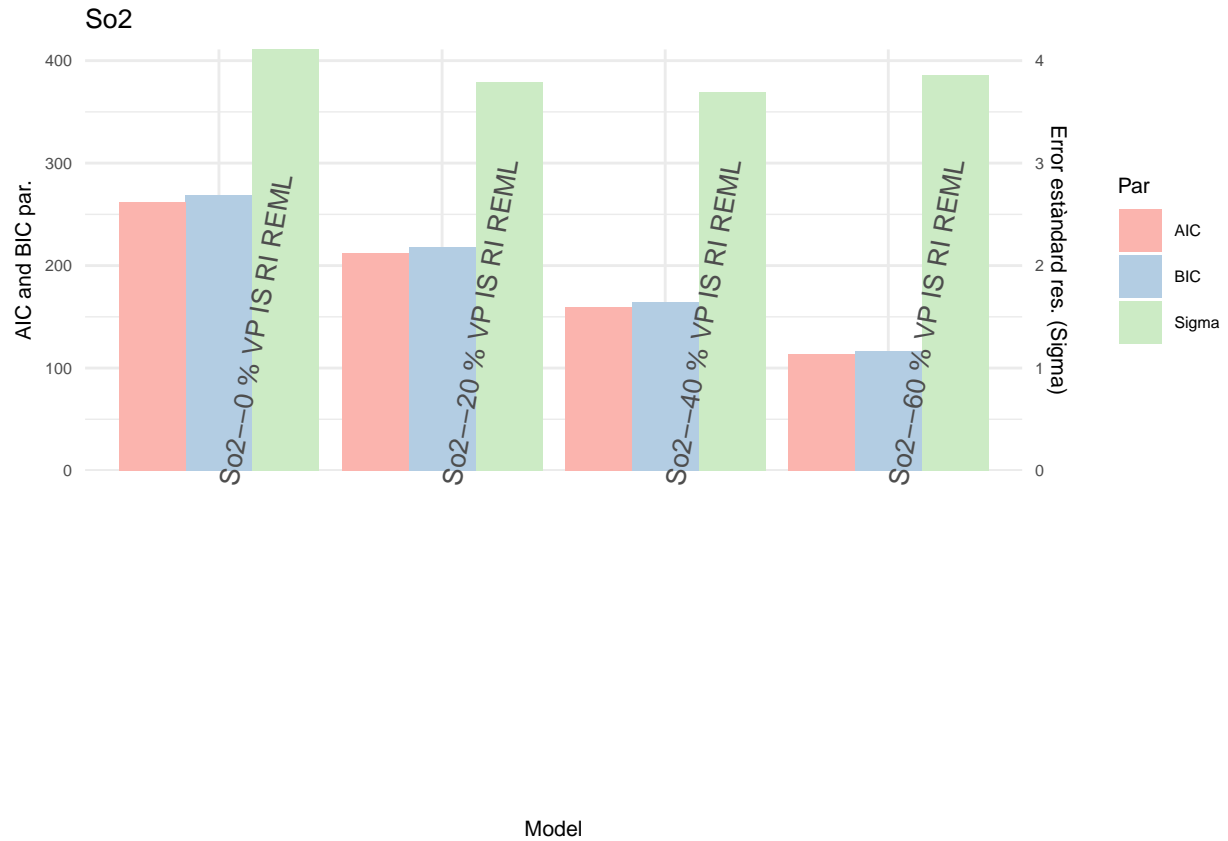
```
## [[1]]
##              Sigma      AIC      BIC
## So1--0 % VP IS RI REML 0.5843 90.1251 96.8806
## So1--20 % VP IS RI REML 0.5513 71.7565 77.6195
## So1--40 % VP IS RI REML 0.5485 56.0122 60.5542
## So1--60 % VP IS RI REML 0.5727 43.6928 46.5250
##
## [[2]]
```



Model

```
So2_MD_R <- Comp_IS_RI_MD_So2
names(So2_MD_R) <- paste(names(Comp_IS_RI_MD_So2), c("IS RI REML"), sep=" ")
ML_Resum(So2_MD_R, "So2")
```

```
## [[1]]
##           Sigma      AIC      BIC
## So2--0 % VP IS RI REML 4.1099 262.5228 269.2783
## So2--20 % VP IS RI REML 3.7953 212.4042 218.2672
## So2--40 % VP IS RI REML 3.6899 159.4850 164.0270
## So2--60 % VP IS RI REML 3.8563 113.6394 116.4716
##
## [[2]]
```



```

So3_MD_R <- Comp_IS_RIS_MD_So3
names(So3_MD_R) <- paste(names(Comp_IS_RIS_MD_So3), c("IS RIS REML"), sep=" ")

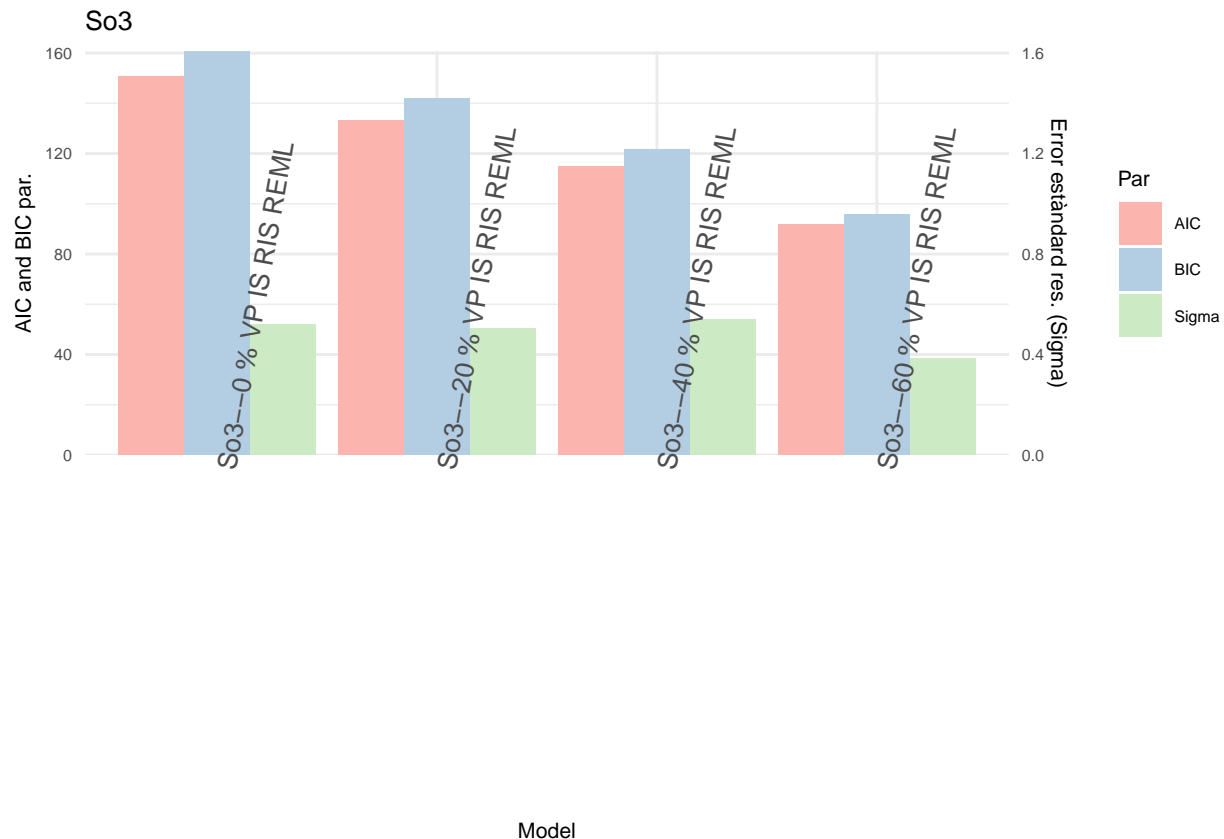
ML_Resum(So3_MD_R, "So3")

```

```

## [[1]]
##                Sigma      AIC      BIC
## So3--0 % VP IS RIS REML 0.5188 150.5445 160.6778
## So3--20 % VP IS RIS REML 0.5028 133.0514 141.8458
## So3--40 % VP IS RIS REML 0.5401 114.7276 121.5405
## So3--60 % VP IS RIS REML 0.3829  91.6770  95.9253
##
## [[2]]

```



Comentaris: Sembla haver un patró clar de baixada dels indicadors AIC/BIC en els models a mesura que hi ha més absència de dades. No hauria de ser el normal ja que l'absència de dades habitualment és un impediment per tenir un bon model. Possiblement en aquest cas el model s'ajusta més correctament a les dades que queden tot i que és probable que el resultat sigui més esbiaixat.

Contrast RLRT

Comparatiu de la prova de significació de l'efecte aleatori prenent com a model vàlid el model calcular per ML amb REML:

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

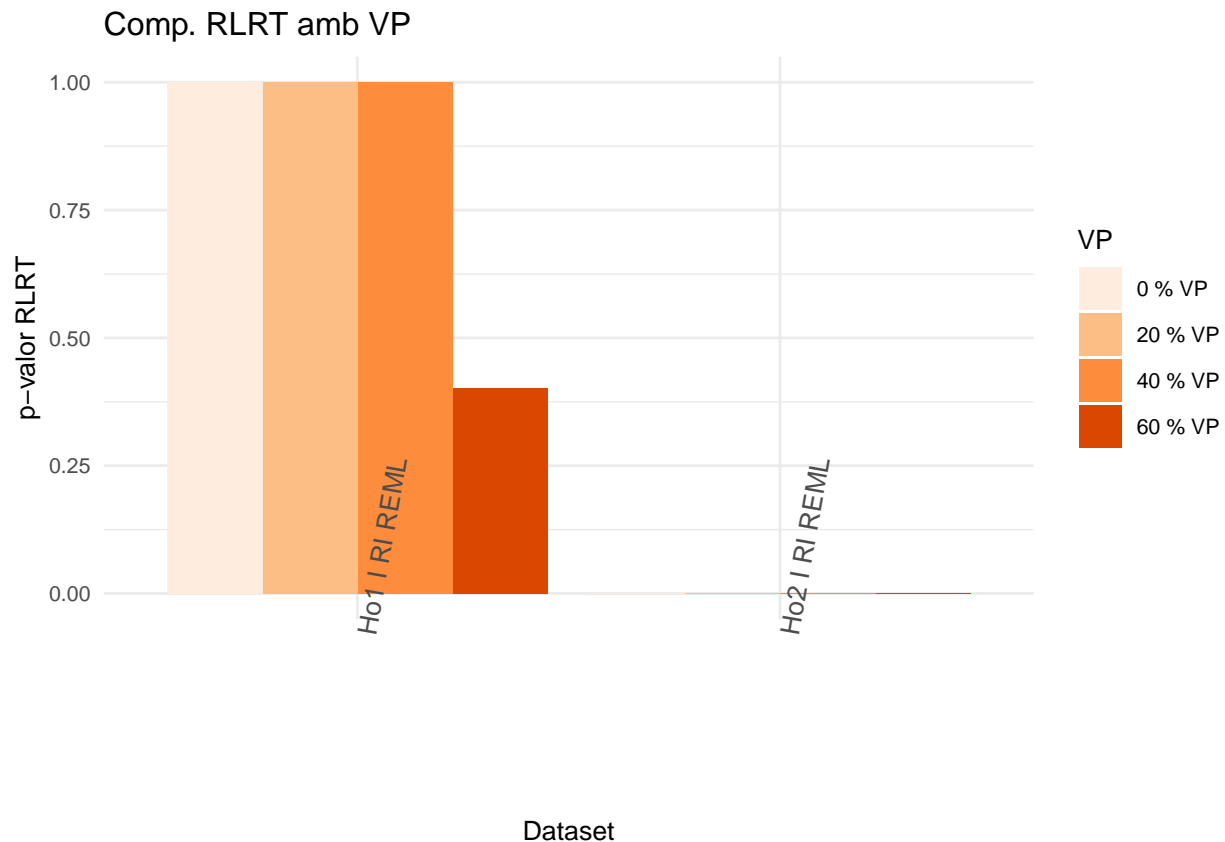
MD_I_RI_Ho_REML_RLRT <- data.frame(Dataset = paste(rep(HoVars,each=length(Comp_I_RI_MD_Ho1_RLRT)),
                                                paste("I RI","REML",sep=" ")))

MD_I_RI_Ho_REML_RLRT$VP <- rep(names(Comp_I_RI_MD_Ho1_RLRT),length(HoVars))

Comp_MD_RI_Ho_0 <- list(Comp_I_RI_MD_Ho1_RLRT,Comp_I_RI_MD_Ho2_RLRT)
Comp_MD_RI_Ho <- ConvList(Comp_MD_RI_Ho_0)

MD_I_RI_Ho_REML_RLRT$p_valor <- sapply(1:length(Comp_MD_RI_Ho),function(i){
  if (class(Comp_MD_RI_Ho[[i]])!="htest"){
    return("-")
  } else {
    round(Comp_MD_RI_Ho[[i]]$p.value,4)
  }
})
```

```
ggplot(data=MD_I_RI_Ho_REML_RLRT, aes(x=Dataset, y=p_valor, fill=VP)) +
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Oranges") + theme_minimal() +
  labs(title="Comp. RLRT amb VP",y="p-valor RLRT") +
  theme(text = element_text(size=10),axis.text.x = element_text(angle=80,size = 10,hjust=0))
```



Comentaris:

- Ho1: L'efecte aleatori per estrat de Lot és no significatiu en general al haver núvol de punts aleatori. Possiblement disminueix al augmentar els valors perduts pel fals efecte de aleatorietat seguint patrons concrets.
- Ho2: L'efecte aleatori resulta significatiu en tots els casos. Fins i tot amb pèrdua de dades de fins al 60% aquest conjunt resulta altament diferenciat per lot la mitja d'observacions.

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

MD_I_RI_W_He_REML_RLRT <- data.frame(Dataset = paste(rep(HeVars,each=length(Comp_I_RI_W_MD_He1_RLRT)),
  paste("I RI W", "REML", sep=" ")))

MD_I_RI_W_He_REML_RLRT$VP <- rep(names(Comp_I_RI_W_MD_He1_RLRT),length(HeVars))

Comp_MD_RI_W_He_0 <- list(Comp_I_RI_W_MD_He1_RLRT,Comp_I_RI_W_MD_He2_RLRT,Comp_I_RI_W_MD_He3_RLRT)
Comp_MD_RI_W_He <- ConvList(Comp_MD_RI_W_He_0)

MD_I_RI_W_He_REML_RLRT$p_valor <- sapply(1:length(Comp_MD_RI_W_He),function(l1){
  if (class(Comp_MD_RI_W_He[[l1]])!="htest"){
```

```

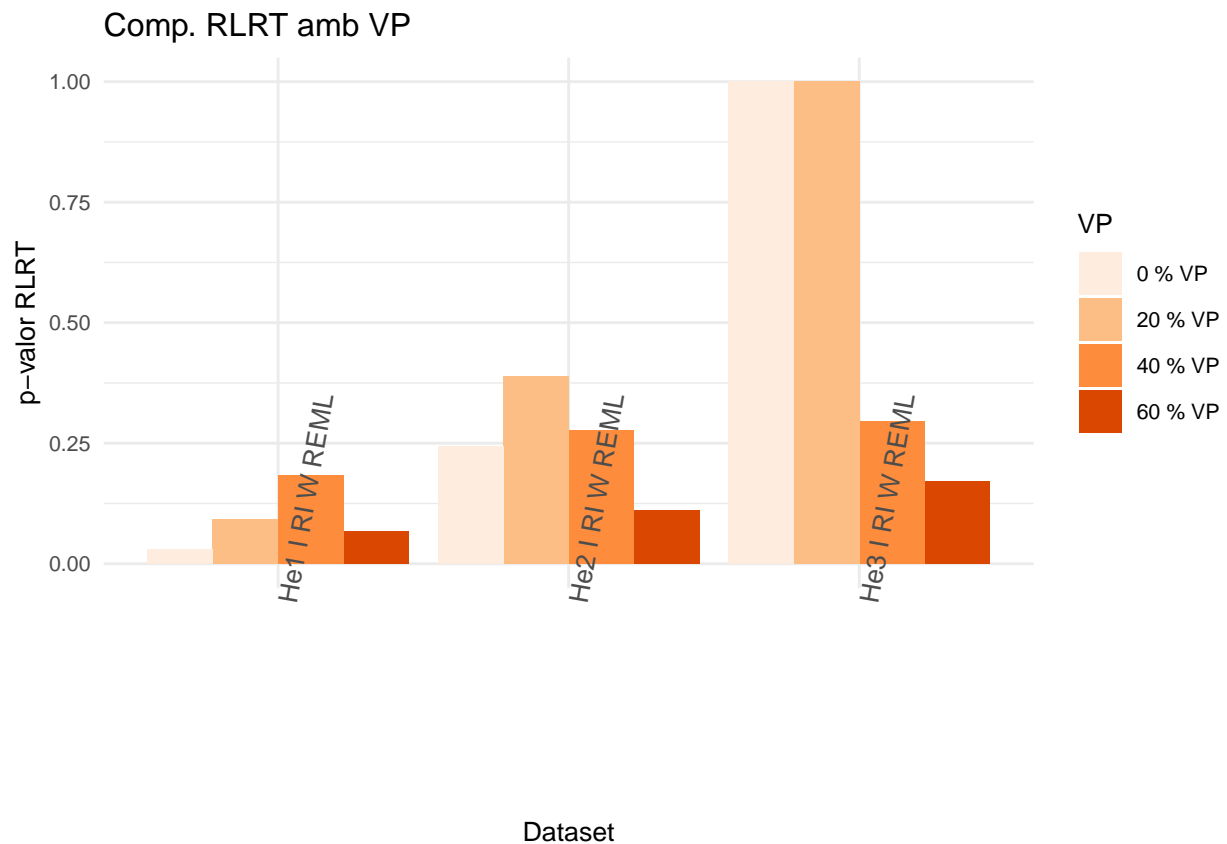
return("-")
} else {
  round(Comp_MD_RI_W_He[[11]]$p.value,4)
}
})

```

```

ggplot(data=MD_I_RI_W_He_REML_RLRT, aes(x=Dataset, y=p_valor, fill=VP)) +
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Oranges") + theme_minimal() +
  labs(title="Comp. RLRT amb VP",y="p-valor RLRT") +
  theme(text = element_text(size=10),axis.text.x = element_text(angle=80,size = 10,hjust=0))

```



Comentaris:

- He1: L'efecte aleatori perd significació al perdre dades el conjunt. El patró que detecta el model de la lleugera dependència del lot no es manté com en el model Ho2 al haver una dispersió molt més forta.
- He2/He3: L'efecte aleatori ja no era significatiu degut probablement a la alta dispersió dels punts i el canvi de punts fa variar aquesta significació. No es considera que sigui un patró genèric de significació en relació a la pèrdua de valors ja que pot ser més fruit de l'aleatorietat concreta dels punts retirats que ajusten millor l'efecte aleatori dels lots o pitjor.

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

```

MD_I_RI_S_So_REML_RLRT <- data.frame(Dataset = paste(rep(c(SoVars, "So3(RIS)"),
  each=length(Comp_IS_RI_MD_So1_RLRT)),
  paste("IS RI/RIS", "REML", sep=" ")))

```

```

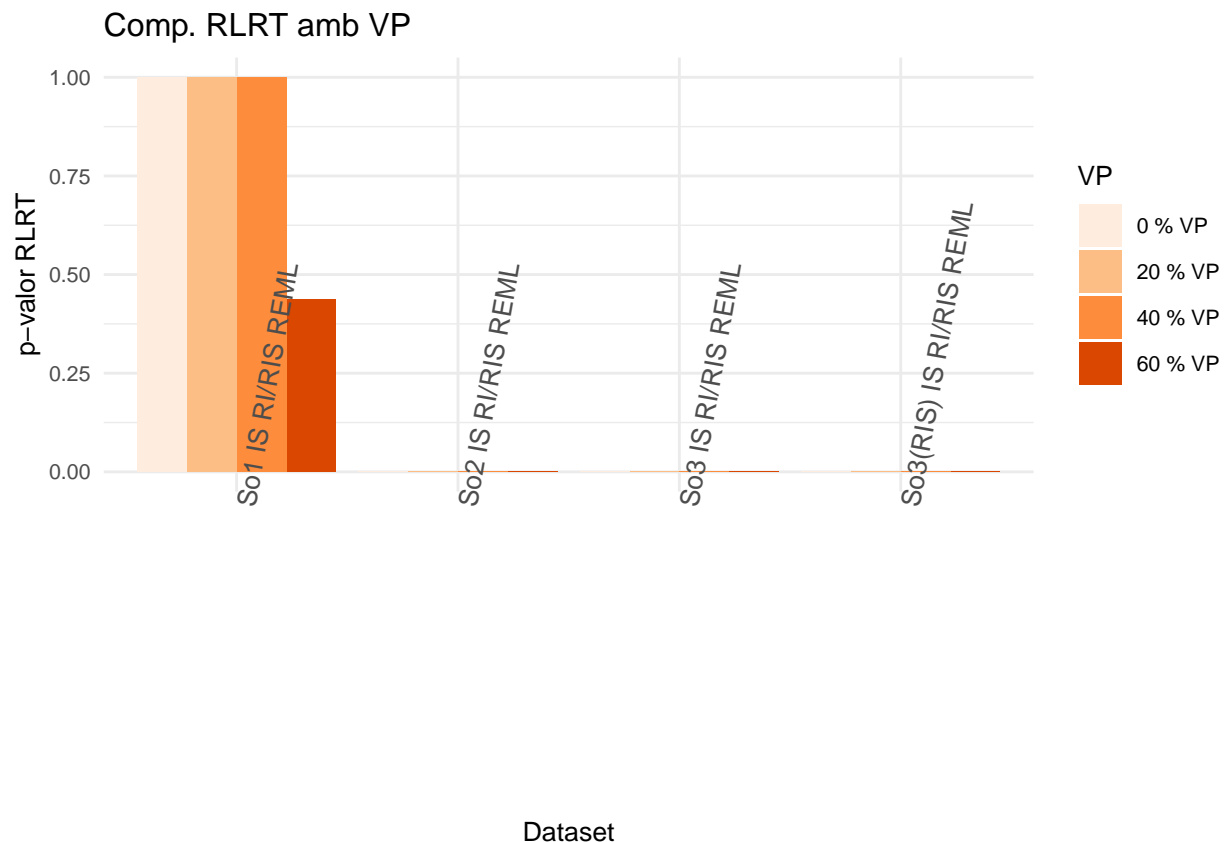
MD_I_RI_S_So_REML_RLRT$VP <- rep(names(Comp_IS_RI_MD_So1_RLRT), (length(SoVars)+1))

Comp_MD_RI_S_So_0 <- list(Comp_IS_RI_MD_So1_RLRT, Comp_IS_RI_MD_So2_RLRT,
                          Comp_IS_RI_MD_So3_RLRT, Comp_IS_RIS_MD_So3_RLRT)
Comp_MD_RI_S_So <- ConvList(Comp_MD_RI_S_So_0)

MD_I_RI_S_So_REML_RLRT$p_valor <- sapply(1:length(Comp_MD_RI_S_So), function(l1){
  if (class(Comp_MD_RI_S_So[[l1]])[1]=="htest"){
    round(Comp_MD_RI_S_So[[l1]]$p.value,4)
  } else {
    round(Comp_MD_RI_S_So[[l1]]$"p-value"[2],4)
  }
})

ggplot(data=MD_I_RI_S_So_REML_RLRT, aes(x=Dataset, y=p_valor, fill=VP)) +
  geom_bar(stat="identity", position=position_dodge()) +
  scale_fill_brewer(palette="Oranges") + theme_minimal() +
  labs(title="Comp. RLRT amb VP", y="p-valor RLRT") +
  theme(text = element_text(size=10), axis.text.x = element_text(angle=80, size = 10, hjust=0))

```



Comentaris: Tots els efectes semblen significatius en els conjunts So2 i So3 independentment de la pèrdua de dades. Semblen doncs necessaris igualment per explicar la variabilitat. En el model So1 segueix essent poc significatiu. La baixada segurament es deu a la pròpia aleatorietat de la simulació de pèrdua de dades.

Paràmetres Efectes Aleatoris

Prenent igualment el model REML com a vàlid, veiem els canvis en els components de la variància de l'efecte aleatori (covariància de la matriu de variància-covariància entre repeticions) i la correlació que en resulta entre repeticions:

```
#Generació de les llistes a mida de models per poder fer la taula de paràmetres amb el model REML
Ho1_MD_R_2 <- ConvList(Comp_I_RI_MD_Ho1)
Ho2_MD_R_2 <- ConvList(Comp_I_RI_MD_Ho2)
Ho_MD_R_2_0 <- list(Ho1_MD_R_2,Ho2_MD_R_2)
Ho_MD_R_2 <- ConvList(Ho_MD_R_2_0)

s<-seq(from=6, to=length(Ho_MD_R_2), by=6)
Ho_MD_R_2r <- Ho_MD_R_2[s]

Ho_MD_RIpar <- data.frame(Dataset = paste(rep(HoVars,each=length(Comp_I_RI_MD_Ho1)),rep(names(Comp_I_RI_MD_Ho1),length(HoVars))),
                          VarRI = rep(VarRI,each=length(HoVars)),
                          CorrRI = rep(CorrRI,each=length(HoVars)))

Ho_MD_RIpar <- rbind(Ho_MD_RIpar,Ho_MD_RIpar)

Ho_MD_RIpar$Par_Tipus <- factor(rep(c("RI Variància","RI Correlació"),each=(nrow(Ho_MD_RIpar)/2)),
                               levels=c("RI Variància","RI Correlació"))

Ho_MD_RIpar_VarRI <- sapply(1:length(Ho_MD_R_2r),function(B){
  round(as.numeric(VarCorr(Ho_MD_R_2r[[B]])[1]),4)
})

Ho_MD_RIpar_CorrRI <- sapply(1:length(Ho_MD_R_2r),function(B){
  a<-getVarCov(Ho_MD_R_2r[[B]],type="marginal")
  b<-cov2cor(a[[1]])
  return(round(b[1,2],4))
})

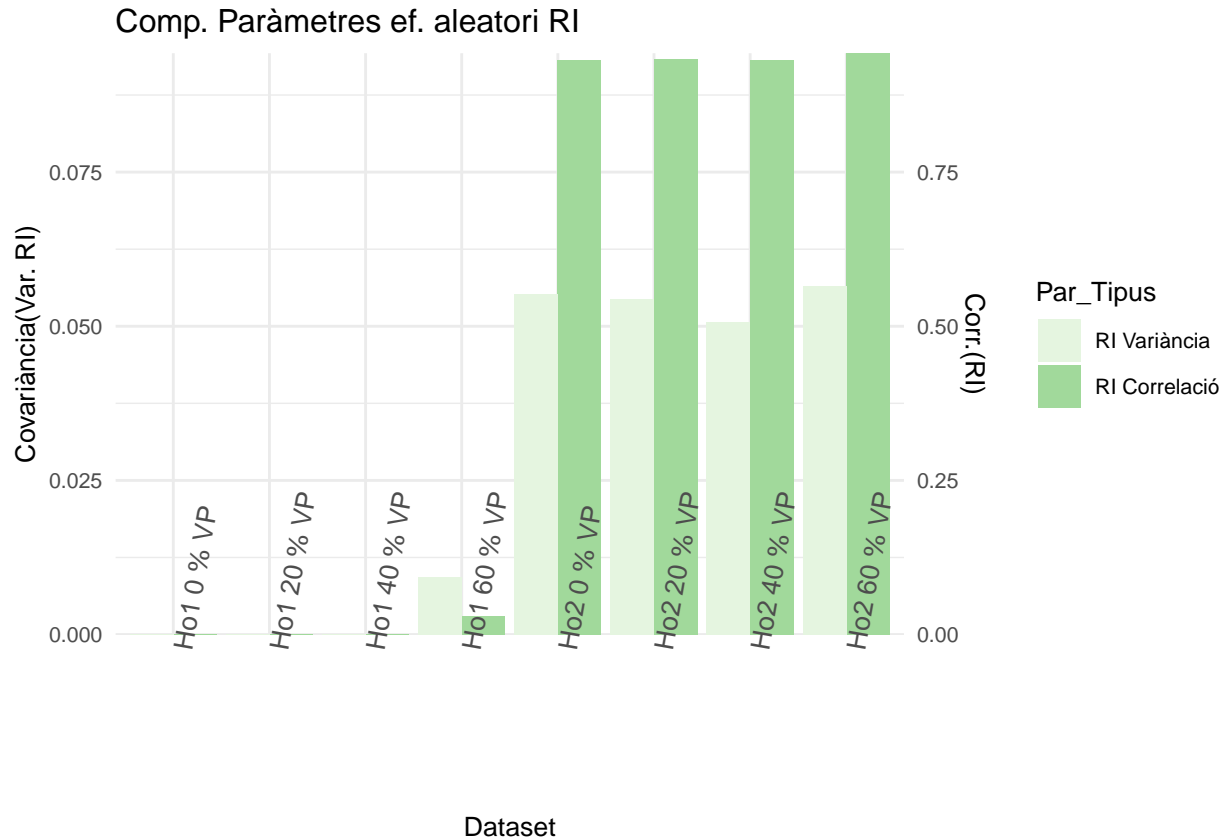
fact <- if(mean(abs(Ho_MD_RIpar_VarRI)) >= mean(abs(Ho_MD_RIpar_CorrRI))){ #Es fa un càlcul del factor de correcció
  10^(round(log10(abs(mean(Ho_MD_RIpar_VarRI))/abs(mean(Ho_MD_RIpar_CorrRI))))
} else {
  1/(10^(round(log10(abs(mean(Ho_MD_RIpar_CorrRI))/abs(mean(Ho_MD_RIpar_VarRI))))))
}

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

Ho_MD_RIpar_CorrRI <- Ho_MD_RIpar_CorrRI*fact

Ho_MD_RIpar$Par <- c(Ho_MD_RIpar_VarRI,Ho_MD_RIpar_CorrRI)

ggplot(data=Ho_MD_RIpar, aes(x=Dataset, y=Par, fill=Par_Tipus)) +
  geom_bar(stat="identity", position=position_dodge())+
  scale_fill_brewer(palette="Greens") + theme_minimal() +
  labs(title="Comp. Paràmetres ef. aleatori RI",y="Covariància(Var. RI)" ) +
  scale_y_continuous(expand = c(0, 0),sec.axis = sec_axis(~./fact, name = "Corr.(RI)") ) +
  theme(text = element_text(size=10),axis.text.x = element_text(angle=80,size = 10,hjust=0))
```

Comentaris:

- Ho1: Variacions probablement degut a l'aleatorietat concreta de pèrdua de dades.
- Ho2: La centralitat de punts per Lot fa que tot i la pèrdua de dades es mantingui en el model el grau de covariància/correlació entre repeticions degut a l'efecte aleatori.

```
#Generació de les llistes a mida de models per poder fer la taula de paràmetres amb el model REML
He1_MD_R_2 <- ConvList(Comp_I_RI_W_MD_He1)
He2_MD_R_2 <- ConvList(Comp_I_RI_W_MD_He2)
He3_MD_R_2 <- ConvList(Comp_I_RI_W_MD_He3)
He_MD_R_2_0 <- list(He1_MD_R_2,He2_MD_R_2,He3_MD_R_2)
He_MD_R_2 <- ConvList(He_MD_R_2_0)

s<-seq(from=8, to=length(He_MD_R_2), by=12)
He_MD_R_2r <- He_MD_R_2[s]

He_MD_RIpar <- data.frame(Dataset = paste(rep(HeVars,each=length(Comp_I_RI_W_MD_He1)),rep(names(Comp_I_RI_W_MD_He1),length(Comp_I_RI_W_MD_He1))),
                          VarRI = rep(0,length(Comp_I_RI_W_MD_He1)),
                          CorrRI = rep(0,length(Comp_I_RI_W_MD_He1)))

He_MD_RIpar <- rbind(He_MD_RIpar,He_MD_RIpar)

He_MD_RIpar$Par_Tipus <- factor(rep(c("RI Variància","RI Correlació"),each=(nrow(He_MD_RIpar)/2)),
                               levels=c("RI Variància","RI Correlació"))

He_MD_RIpar_VarRI <- sapply(1:length(He_MD_R_2r),function(B){
  round(as.numeric(VarCorr(He_MD_R_2r[[B]])[1]),4)
})
```

```

He_MD_RIpar_CorrRI <- sapply(1:length(He_MD_R_2r),function(B){
  a<-getVarCov(He_MD_R_2r[[B]],type="marginal")
  b<-cov2cor(a[[1]])
  return(round(b[1,2],4))
})

fact <- if(mean(abs(He_MD_RIpar_VarRI)) >= mean(abs(He_MD_RIpar_CorrRI))){ #Es fa un càlcul del factor
  10^(round(log10(abs(mean(He_MD_RIpar_VarRI))/abs(mean(He_MD_RIpar_CorrRI))))
} else {
  1/(10^(round(log10(abs(mean(He_MD_RIpar_CorrRI))/abs(mean(He_MD_RIpar_VarRI))))))
}

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

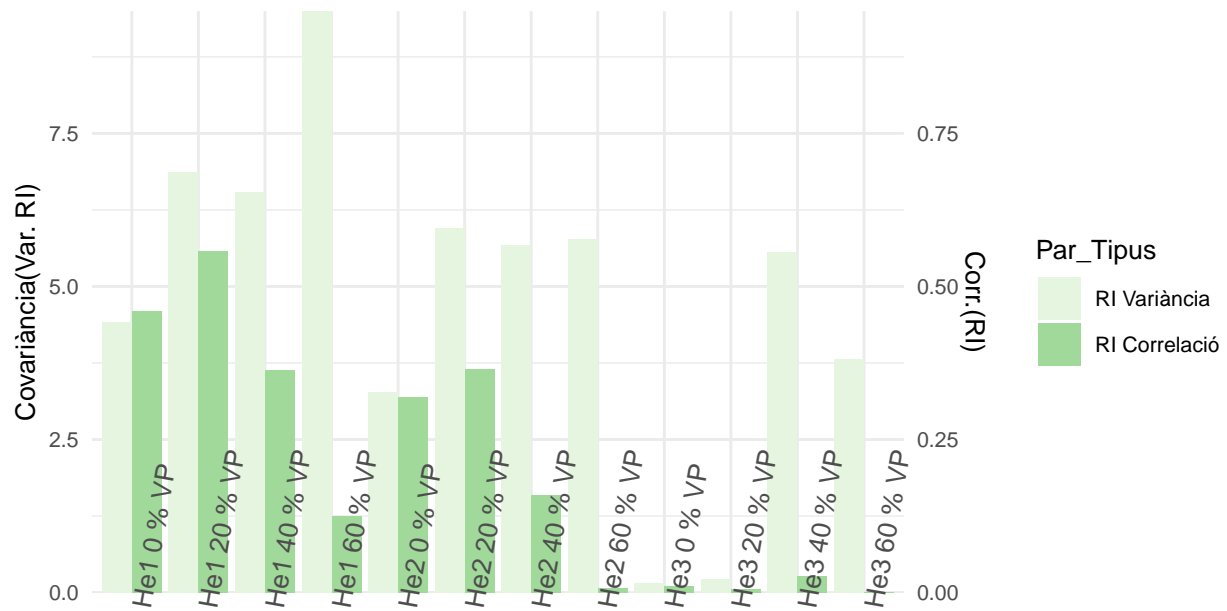
He_MD_RIpar_CorrRI <- He_MD_RIpar_CorrRI*fact

He_MD_RIpar$Par <- c(He_MD_RIpar_VarRI,He_MD_RIpar_CorrRI)

ggplot(data=He_MD_RIpar, aes(x=Dataset, y=Par, fill=Par_Tipus)) +
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Greens") + theme_minimal() +
  labs(title="Comp. Paràmetres ef. aleatori RI",y="Covariància(Var. RI)" ) +
  scale_y_continuous(expand = c(0, 0),sec.axis = sec_axis(~./fact, name = "Corr.(RI)") ) +
  theme(text = element_text(size=10),axis.text.x = element_text(angle=80,size = 10,hjust=0))

```

Comp. Paràmetres ef. aleatori RI



Dataset

Comentaris:

- He1: El conjunt que té certa significància l'efecte aleatori mostra un nivell de covariància més alt i de correlació. Curiosament semblaria que la covariància augmenta amb la pèrdua de valors i la correlació disminueix. És a dir que possiblement el model dona una càrrega de l'explicació de la variabilitat més alta a l'efecte aleatori, però és incapaç d'explicar-lo a nivell de correlació entre repeticions. Hem de tenir en compte que en aquest cas també s'afegeix la funció de modulació de la variància.
- He2: Efecte semblant a He1.
- He3: La dispersió d'aquest conjunt fa que la correlació entre repeticions es mantingui sempre baixa tot i que en els models amb més pèrdua de dades el model calcula una covariància de l'efecte aleatori alta. Possiblement pot ser un fals patró com s'ha anat comentat en alguns casos.

```
#Prenem la correlació entre els punts més propers com a referència
#Generació de les llistes a mida de models per poder fer la taula de paràmetres amb el model REML
So1_MD_R_2 <- ConvList(Comp_IS_RI_MD_So1)
So2_MD_R_2 <- ConvList(Comp_IS_RI_MD_So2)
So3_MD_R_2 <- ConvList(Comp_IS_RIS_MD_So3)
So_MD_R_2_0 <- list(So1_MD_R_2, So2_MD_R_2, So3_MD_R_2, So3_MD_R_2)
So_MD_R_2 <- ConvList(So_MD_R_2_0)

#A So3 s'ha de prendre la part de l'efecte aleatori de intercepció la del temps per separat:
So_MD_RIpar <- data.frame(Dataset = paste(rep(c(SoVars, "So3(t)"), each=length(Comp_IS_RI_MD_So1)),
                                         rep(names(Comp_IS_RI_MD_So1), (length(SoVars)+1))))

So_MD_RIpar <- rbind(So_MD_RIpar, So_MD_RIpar)

So_MD_RIpar$Par_Tipus <- factor(rep(c("RI Variància", "RI Correlació"), each=(nrow(So_MD_RIpar)/2)),
```

```

                                levels=c("RI Variància","RI Correlació"))

So_MD_RIpar_VarRI <- c(
  sapply(1:(length(So_MD_R_2)-4),function(B){
    round(as.numeric(VarCorr(So_MD_R_2[[B]])[1]),4)
  }),
  sapply((length(So_MD_R_2)-3):length(So_MD_R_2),function(B){
    round(as.numeric(VarCorr(So_MD_R_2[[B]])[2]),4)
  })
)

So_MD_RIpar_CorrRI <- sapply(1:length(So_MD_R_2),function(B){
  a<-getVarCov(So_MD_R_2[[B]],type="marginal")
  b<-cov2cor(a[[1]])
  return(round(b[1,2],4))
})

fact <- if(mean(abs(So_MD_RIpar_VarRI)) >= mean(abs(So_MD_RIpar_CorrRI)){ #Es fa un càlcul del factor
  10^(round(log10(abs(mean(So_MD_RIpar_VarRI))/abs(mean(So_MD_RIpar_CorrRI))))
} else {
  1/(10^(round(log10(abs(mean(So_MD_RIpar_CorrRI))/abs(mean(So_MD_RIpar_VarRI))))))
}

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

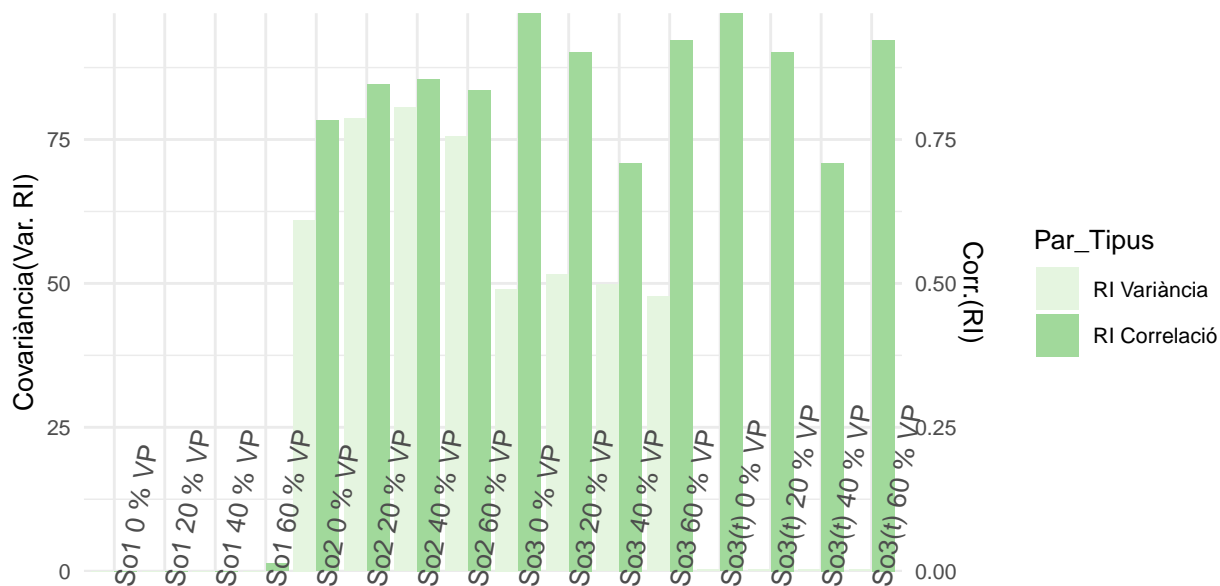
So_MD_RIpar_CorrRI <- So_MD_RIpar_CorrRI*fact

So_MD_RIpar$Par <- c(So_MD_RIpar_VarRI,So_MD_RIpar_CorrRI)

ggplot(data=So_MD_RIpar, aes(x=Dataset, y=Par, fill=Par_Tipus)) +
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Greens") + theme_minimal() +
  labs(title="Comp. Paràmetres ef. aleatori RI",y="Covariància(Var. RI)" ) +
  scale_y_continuous(expand = c(0, 0),sec.axis = sec_axis(~./fact, name = "Corr.(RI)") ) +
  theme(text = element_text(size=10),axis.text.x = element_text(angle=80,size = 10,hjust=0))

```

Comp. Paràmetres ef. aleatori RI



Dataset

No s'observa un patró claríssim entre els diferents conjunts en quant als efectes en la matriu de variància-covariància dels efectes aleatoris.

Suma de Quadrats

Després de algunes proves, no es considera aquest apartat al estar tinguent en compte conjuntament models calculats per mínims quadrats i màxima versemblança.

Intervals de Confiança de la predicció de mitges

No és senzill calcular els intervals de confiança de predicció en els models mixtes ja que la variabilitat depèn de més factors a part del factor residual a part de tenir el factor o factors aleatoris. Per duu a terme aquesta tasca dins el temps disponible per l'anàlisi estadístic es segueix les recomanacions de <http://www.remkoduursma.com/post/2017-06-15-bootpredictlme4/> i per fer-ho s'ajusten tots els models amb la funció `lmer`:

```
#A partir de la tècnica bootstrap amb l'estadístic t es pot deduir la possible distribució i calcular e
set.seed(7982)
lmer_I_RI_MD_Ho1 <- lapply(1:length(Av_DS_MD_list),function(p){
  lmer(Ho1 ~ 1 + (1|Lot),data=Av_DS_MD_list[[p]],REML=T,na.action=na.omit)
})
pred_lmer_I_RI_MD_Ho1 <- lapply(1:length(lmer_I_RI_MD_Ho1),function(p){
  predict(lmer_I_RI_MD_Ho1[[p]], newdata=data.frame(Lot="Lot 1"), se.fit=TRUE, nsim=100)
})

set.seed(7982)
lmer_I_RI_MD_Ho2 <- lapply(1:length(Av_DS_MD_list),function(p){
```

```

  lmer(Ho2 ~ 1 + (1|Lot),data=Av_DS_MD_list[[p]],REML=T,na.action=na.omit)
})
pred_lmer_I_RI_MD_Ho2 <- lapply(1:length(lmer_I_RI_MD_Ho2),function(p){
  predict(lmer_I_RI_MD_Ho2[[p]], newdata=data.frame(Lot="Lot 1"), se.fit=TRUE, nsim=100)
})

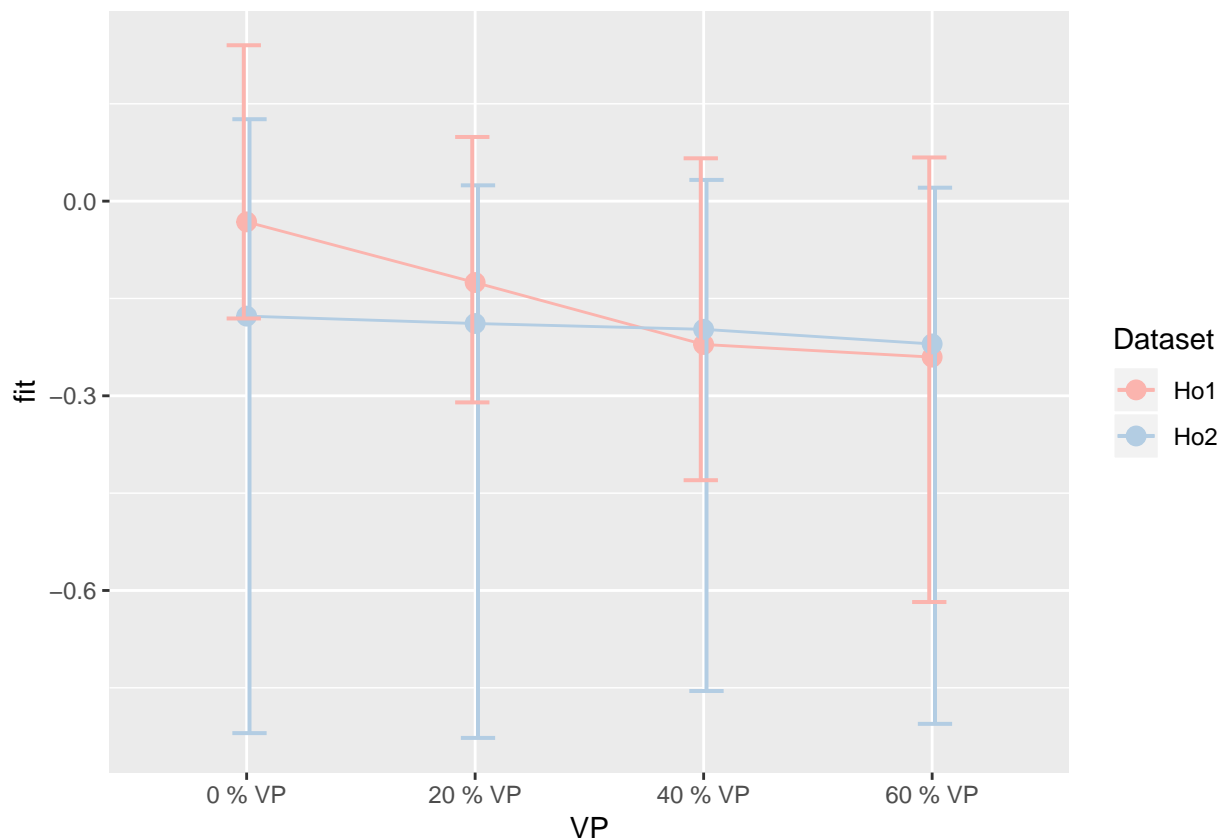
pred_lmer_Ho <- data.frame(Dataset = rep(HoVars,each=length(Av_DS_MD_list)))
pred_lmer_Ho$VP <- rep(names(Av_DS_MD_list),length(HoVars))
pred_lmer_Ho$fit <- c(
  sapply(1:length(pred_lmer_I_RI_MD_Ho1),function(p){
    pred_lmer_I_RI_MD_Ho1[[p]]$fit
  }),
  sapply(1:length(pred_lmer_I_RI_MD_Ho2),function(p){
    pred_lmer_I_RI_MD_Ho2[[p]]$fit
  })
)

pred_lmer_Ho$LCI <- c(
  sapply(1:length(pred_lmer_I_RI_MD_Ho1),function(p){
    pred_lmer_I_RI_MD_Ho1[[p]]$ci.fit[1,]
  }),
  sapply(1:length(pred_lmer_I_RI_MD_Ho2),function(p){
    pred_lmer_I_RI_MD_Ho2[[p]]$ci.fit[1,]
  })
)

pred_lmer_Ho$HCI <- c(
  sapply(1:length(pred_lmer_I_RI_MD_Ho1),function(p){
    pred_lmer_I_RI_MD_Ho1[[p]]$ci.fit[2,]
  }),
  sapply(1:length(pred_lmer_I_RI_MD_Ho2),function(p){
    pred_lmer_I_RI_MD_Ho2[[p]]$ci.fit[2,]
  })
)

ggplot(pred_lmer_Ho, aes(x=VP, y=fit, group=Dataset, color=Dataset)) +
  geom_line() +
  geom_point(size=3)+
  geom_errorbar(aes(ymin=LCI, ymax=HCI), width=.3,size=.7,
               position=position_dodge(0.05))+
  scale_color_brewer(palette="Pastel1")

```



Comentaris:

Tot i la variació de conjunt per les dades perdudes semblaria que l'interval de confiança per predir la mitja d'un lot qualsevol no variaria enormement. Ho sempre sembla variar més probablement pel factor aleatori aplicat que s'ha anat comentant.

```
#A partir de la tècnica bootstrap amb l'estadístic t es pot deduir la possible distribució i calcular e
set.seed(7982)
lmer_I_RI_W_MD_He1 <- lapply(1:length(Av_DS_MD_list),function(p){
  lmer(He1 ~ 1 + (1|Lot),data=Av_DS_MD_list[[p]],
    weights = Temps, REML=T,na.action=na.omit)
})

pred_lmer_I_RI_W_MD_He1 <- lapply(1:length(lmer_I_RI_W_MD_He1),function(p){
  predict(lmer_I_RI_W_MD_He1[[p]], newdata=data.frame(Lot="Lot 1", Temps = 24), se.fit=TRUE, nsim=100)
})

set.seed(7982)
lmer_I_RI_W_MD_He2 <- lapply(1:length(Av_DS_MD_list),function(p){
  lmer(He2 ~ 1 + (1|Lot),data=Av_DS_MD_list[[p]],
    weights = Temps, REML=T,na.action=na.omit)
})

pred_lmer_I_RI_W_MD_He2 <- lapply(1:length(lmer_I_RI_W_MD_He2),function(p){
  predict(lmer_I_RI_W_MD_He2[[p]], newdata=data.frame(Lot="Lot 1", Temps = 24), se.fit=TRUE, nsim=100)
})

set.seed(7982)
```

```

lmer_I_RI_W_MD_He3 <- lapply(1:length(Av_DS_MD_list),function(p){
  lmer(He3 ~ 1 + (1|Lot),data=Av_DS_MD_list[[p]],
      weights = Temps, REML=T,na.action=na.omit)
})
pred_lmer_I_RI_W_MD_He3 <- lapply(1:length(lmer_I_RI_W_MD_He3),function(p){
  predict(lmer_I_RI_W_MD_He3[[p]], newdata=data.frame(Lot="Lot 1", Temps = 24), se.fit=TRUE, nsim=100)
})

pred_lmer_He <- data.frame(Dataset = rep(HeVars,each=length(Av_DS_MD_list)))
pred_lmer_He$VP <- rep(names(Av_DS_MD_list),length(HeVars))
pred_lmer_He$fit <- c(
  sapply(1:length(pred_lmer_I_RI_W_MD_He1),function(p){
    pred_lmer_I_RI_W_MD_He1[[p]]$fit
  }),
  sapply(1:length(pred_lmer_I_RI_W_MD_He2),function(p){
    pred_lmer_I_RI_W_MD_He2[[p]]$fit
  }),
  sapply(1:length(pred_lmer_I_RI_W_MD_He3),function(p){
    pred_lmer_I_RI_W_MD_He3[[p]]$fit
  })
)

pred_lmer_He$LCI <- c(
  sapply(1:length(pred_lmer_I_RI_W_MD_He1),function(p){
    pred_lmer_I_RI_W_MD_He1[[p]]$ci.fit[1,]
  }),
  sapply(1:length(pred_lmer_I_RI_W_MD_He2),function(p){
    pred_lmer_I_RI_W_MD_He2[[p]]$ci.fit[1,]
  }),
  sapply(1:length(pred_lmer_I_RI_W_MD_He3),function(p){
    pred_lmer_I_RI_W_MD_He3[[p]]$ci.fit[1,]
  })
)

pred_lmer_He$HCI <- c(
  sapply(1:length(pred_lmer_I_RI_W_MD_He1),function(p){
    pred_lmer_I_RI_W_MD_He1[[p]]$ci.fit[2,]
  }),
  sapply(1:length(pred_lmer_I_RI_W_MD_He2),function(p){
    pred_lmer_I_RI_W_MD_He2[[p]]$ci.fit[2,]
  }),
  sapply(1:length(pred_lmer_I_RI_W_MD_He3),function(p){
    pred_lmer_I_RI_W_MD_He3[[p]]$ci.fit[2,]
  })
)

p1<-ggplot(pred_lmer_He[pred_lmer_He$Dataset=="He1",], aes(x=VP, y=fit, group=Dataset, color=Dataset)) +
  geom_line() +
  geom_point(size=3)+
  geom_errorbar(aes(ymin=LCI, ymax=HCI), width=.3,size=.7,
               position=position_dodge(0.05))
p2<-ggplot(pred_lmer_He[pred_lmer_He$Dataset=="He2",], aes(x=VP, y=fit, group=Dataset, color=Dataset)) +
  geom_line() +
  geom_point(size=3)+

```



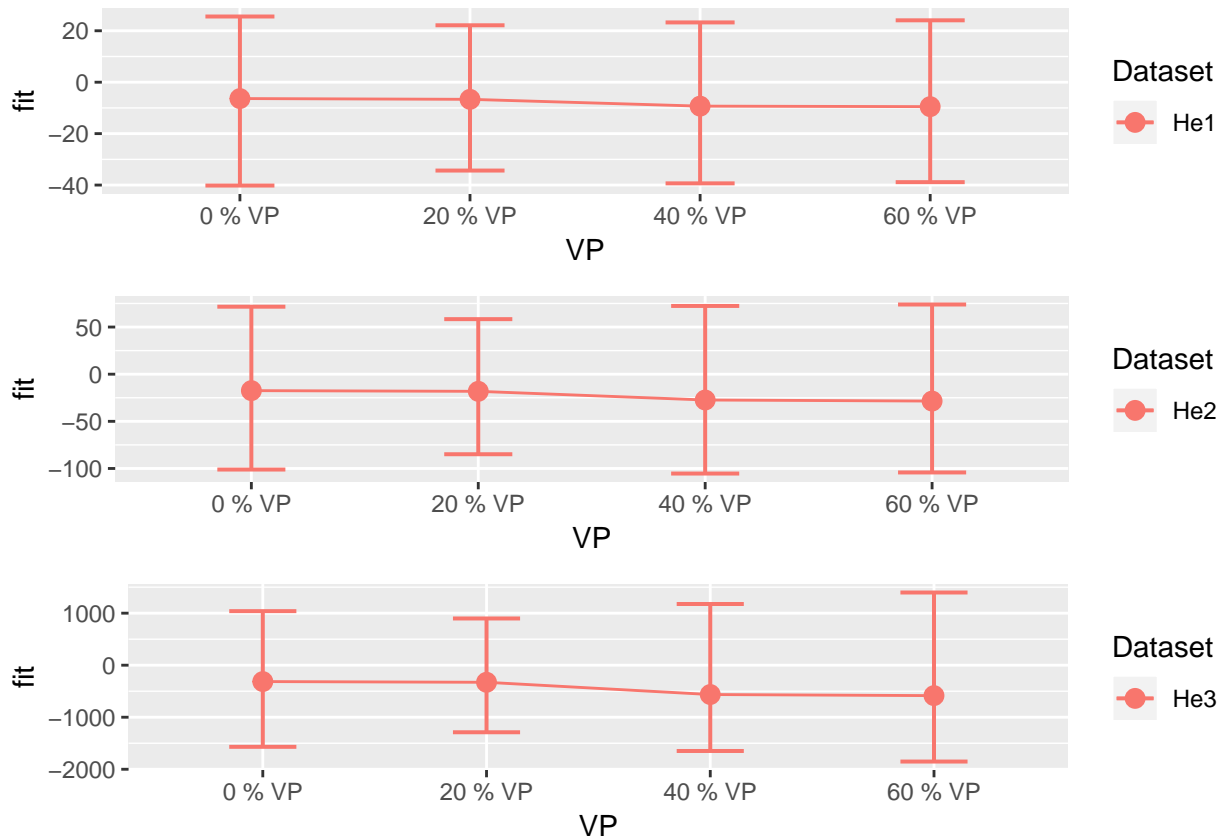
```

geom_errorbar(aes(ymin=LCI, ymax=HCI), width=.3,size=.7,
              position=position_dodge(0.05))

p3<-ggplot(pred_lmer_He[pred_lmer_He$Dataset=="He3",], aes(x=VP, y=fit, group=Dataset, color=Dataset)) +
  geom_line() +
  geom_point(size=3)+
  geom_errorbar(aes(ymin=LCI, ymax=HCI), width=.3,size=.7,
              position=position_dodge(0.05))

grid.arrange(p1,p2,p3)

```



Comentaris:

No s'aprecia una variació molt gran entre els intervals de confiança de les prediccions tot i que seria recomanable repetir aquest anàlisi amb més detall al haver hagut d'alternar les funcions lme i lmer per practicitat de codi i temps disponible.

```

#A partir de la tècnica bootstrap amb l'estadístic t es pot deduir la possible distribució i calcular e
set.seed(7982)
lmer_IS_RI_MD_So1 <- lapply(1:length(Av_DS_MD_list),function(p){
  lmer(So1 ~ 1 + Temps + (1|Lot),data=Av_DS_MD_list[[p]],
      REML=T,na.action=na.omit)
})

pred_lmer_IS_RI_MD_So1 <- lapply(1:length(lmer_IS_RI_MD_So1),function(p){
  predict(lmer_IS_RI_MD_So1[[p]], newdata=data.frame(Lot="Lot 1", Temps = 24), se.fit=TRUE, nsim=100)
})

```

```

set.seed(7982)
lmer_IS_RI_MD_So2 <- lapply(1:length(Av_DS_MD_list),function(p){
  lmer(So2 ~ 1 + Temps + (1|Lot),data=Av_DS_MD_list[[p]],
      REML=T,na.action=na.omit)
})
pred_lmer_IS_RI_MD_So2 <- lapply(1:length(lmer_IS_RI_MD_So2),function(p){
  predict(lmer_IS_RI_MD_So2[[p]], newdata=data.frame(Lot="Lot 1", Temps = 24), se.fit=TRUE, nsim=100)
})

set.seed(7982)
lmer_IS_RIS_MD_So3 <- lapply(1:length(Av_DS_MD_list),function(p){
  lmer(So3 ~ 1 + Temps + (1+Temps|Lot),data=Av_DS_MD_list[[p]],
      REML=T,na.action=na.omit,control = lmerControl(optimizer = "Nelder_Mead"))
})
pred_lmer_IS_RIS_MD_So3 <- lapply(1:length(lmer_IS_RIS_MD_So3),function(p){
  predict(lmer_IS_RIS_MD_So3[[p]], newdata=data.frame(Lot="Lot 1", Temps = 24), se.fit=TRUE, nsim=100)
})

pred_lmer_So <- data.frame(Dataset = rep(SoVars,each=length(Av_DS_MD_list)))
pred_lmer_So$VP <- rep(names(Av_DS_MD_list),length(SoVars))
pred_lmer_So$fit <- c(
  sapply(1:length(pred_lmer_IS_RI_MD_So1),function(p){
    pred_lmer_IS_RI_MD_So1[[p]]$fit
  }),
  sapply(1:length(pred_lmer_IS_RI_MD_So2),function(p){
    pred_lmer_IS_RI_MD_So2[[p]]$fit
  }),
  sapply(1:length(pred_lmer_IS_RIS_MD_So3),function(p){
    pred_lmer_IS_RIS_MD_So3[[p]]$fit
  })
)

pred_lmer_So$LCI <- c(
  sapply(1:length(pred_lmer_IS_RI_MD_So1),function(p){
    pred_lmer_IS_RI_MD_So1[[p]]$ci.fit[1,]
  }),
  sapply(1:length(pred_lmer_IS_RI_MD_So2),function(p){
    pred_lmer_IS_RI_MD_So2[[p]]$ci.fit[1,]
  }),
  sapply(1:length(pred_lmer_IS_RIS_MD_So3),function(p){
    pred_lmer_IS_RIS_MD_So3[[p]]$ci.fit[1,]
  })
)

pred_lmer_So$HCI <- c(
  sapply(1:length(pred_lmer_IS_RI_MD_So1),function(p){
    pred_lmer_IS_RI_MD_So1[[p]]$ci.fit[2,]
  }),
  sapply(1:length(pred_lmer_IS_RI_MD_So2),function(p){
    pred_lmer_IS_RI_MD_So2[[p]]$ci.fit[2,]
  }),
  sapply(1:length(pred_lmer_IS_RIS_MD_So3),function(p){
    pred_lmer_IS_RIS_MD_So3[[p]]$ci.fit[2,]
  })
)

```

```

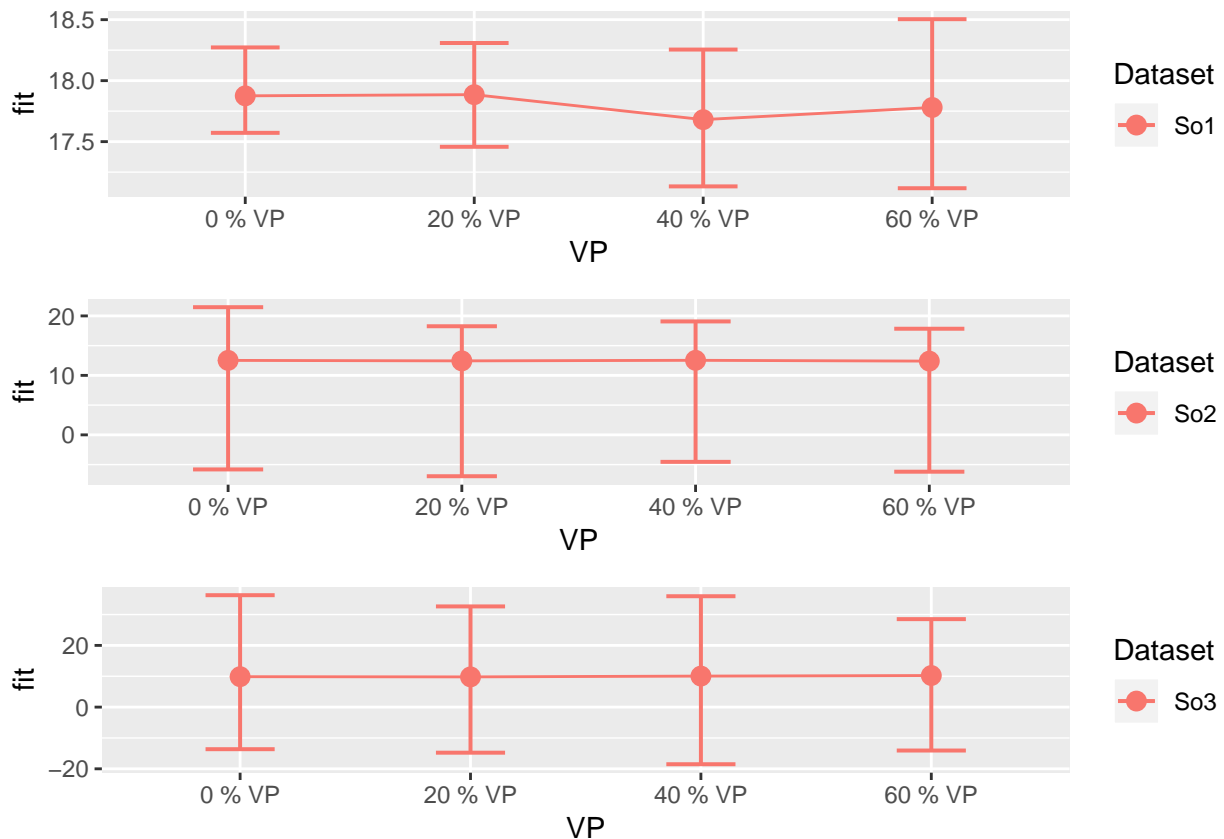
})
)

p1<-ggplot(pred_lmer_So[pred_lmer_So$Dataset=="So1",], aes(x=VP, y=fit, group=Dataset, color=Dataset)) +
  geom_line() +
  geom_point(size=3)+
  geom_errorbar(aes(ymin=LCI, ymax=HCI), width=.3,size=.7,
                position=position_dodge(0.05))
p2<-ggplot(pred_lmer_So[pred_lmer_So$Dataset=="So2",], aes(x=VP, y=fit, group=Dataset, color=Dataset)) +
  geom_line() +
  geom_point(size=3)+
  geom_errorbar(aes(ymin=LCI, ymax=HCI), width=.3,size=.7,
                position=position_dodge(0.05))

p3<-ggplot(pred_lmer_So[pred_lmer_So$Dataset=="So3",], aes(x=VP, y=fit, group=Dataset, color=Dataset)) +
  geom_line() +
  geom_point(size=3)+
  geom_errorbar(aes(ymin=LCI, ymax=HCI), width=.3,size=.7,
                position=position_dodge(0.05))

grid.arrange(p1,p2,p3)

```



Lleugeres diferències en la predicció, però sembla que l'efectivitat del model compensa la pèrdua de dades en els algoritmes utilitzats en el codi almenys.

IMPUTACIÓ DE DADES PERDUDES

Es consideren alguns mètodes d'imputació per veure com compensen el model aprofitant les dades que s'han calculat de l'efecte de no tenir aquestes dades. Per simplificar es tindrà en compte únicament alguns dels mètodes que s'han fet servir per diagnosticar els efectes dels resultats:

- Scatterplots i Gràfics de densitat inicials on es comparen les observacions.
- Anàlisi de indicadors de qualitat AIC, BIC i error estàndard residual.
- Predicció de valors en el temps final.

Es simplifica també les bases tractades prenent els casos on s'han vist possibles desviacions interessants de tractar:

- Ho2
- He2
- He3
- So2
- So3

Es pren únicament un dels casos de pèrdua de dades prenent el del 40%. Un cas exagerat, però no tan inversemblant com el del 60% on habitualment ja no es continuaria l'estudi.

Tot i que es podrà veure en certa mesura, en aquest exercici no podem esperar comparar de manera molt detallada els diferents sistemes ja que cada un d'ells té moltes variables possibles a aplicar segons el tipus de context on estigui imputant dades, pel que es té en compte més en general l'eficàcia de imputar dades en relació a no fer-ho.

Es generen els objectes necessaris inicials:

```
Av_DS_MDIMP_list0 <- Av_DS_MD_list[c(1,3)]
Av_DS_MDIMP_list <- Av_DS_MDIMP_list0
IMP_Vars <- c("Ho2", "He2", "He3", "So2", "So3")

for(p in 1:(length(Av_DS_MDIMP_list0))){
  Av_DS_MDIMP_list[[p]] <- subset(Av_DS_MDIMP_list[[p]], select=-c(Ho1, He1, So1))
}

VP_DS <- Av_DS_MDIMP_list[[length(Av_DS_MDIMP_list0)]]
```

MODELS D'IMPUTACIÓ

kNN

L'algoritme utilitza la fórmula de la distància i concretament escollim la opció de tenir en compte la distància per calcular el pes del veí en el càlcul:

```
#S'aplica individualment per cada variable de resposta i després s'ajunten
kNNO <- lapply(1:length(IMP_Vars), function(v){
  kNN(VP_DS[,c(1:3, (3+v))], k = 5, dist_var = colnames(VP_DS[,c(1:3, (3+v))]) ,weight="auto",
      weightDist = T, imp_var=F, numFun = median)
})

VP_DS_kNN <- cbind(kNNO[[1]], sapply(2:length(IMP_Vars), function(v){
  kNNO[[v]][,4]
}))
colnames(VP_DS_kNN)[4:ncol(VP_DS_kNN)] <- IMP_Vars
```

```
Av_DS_MDIMP_list[[length(Av_DS_MDIMP_list)+1]] <- VP_DS_kNN
names(Av_DS_MDIMP_list)[length(Av_DS_MDIMP_list)] <- "40 % VP kNN Imp."
```

Regressió estocàstica

S'utilitza l'algoritme dins el paquet MICE:

```
BSRO <- lapply(1:length(IMP_Vars),function(v){
  complete(mice(VP_DS[,c(1:3,(3+v))],method = "norm.nob", m = 1,print=F))
})
```

```
VP_DS_BSR <- cbind(BSRO[[1]],sapply(2:length(IMP_Vars),function(v){
  BSRO[[v]][,4]
}))
colnames(VP_DS_BSR)[4:ncol(VP_DS_BSR)] <- IMP_Vars
```

```
Av_DS_MDIMP_list[[length(Av_DS_MDIMP_list)+1]] <- VP_DS_BSR
names(Av_DS_MDIMP_list)[length(Av_DS_MDIMP_list)] <- "40 % VP BSR Imp."
```

Ajust de models lineals

Repetim el mateix que abans però amb la nova llista de conjunts de dades:

```
Comp_I_RI_MDIMP_Ho2 <- lapply(1:length(Av_DS_MDIMP_list),function(p){
  Comp_I_RI_LM("Ho2",Av_DS_MDIMP_list[[p]])
})
```

```
names(Comp_I_RI_MDIMP_Ho2) <- names(Av_DS_MDIMP_list)
```

```
fmHo2 <- formula(paste("Ho2", "~", 1))
```

```
Comp_I_RI_MDIMP_Ho2_RLRT <- lapply(1:length(Av_DS_MDIMP_list),function(p){
  eREML_Ho2<-exactRLRT(lme(fmHo2, random = ~1|Lot, data = Av_DS_MDIMP_list[[p]], method="REML",na.action=na.omit))
})
```

```
names(Comp_I_RI_MDIMP_Ho2_RLRT) <- names(Av_DS_MDIMP_list)
```

#Ajustarem per varExp al ser una funció amb menys condicions que varPower encara que menys potent

```
fML_I_He2 <- formula(He2 ~ 1)
```

```
fRE_RI <- formula(~1|Lot)
```

```
Comp_I_RI_W_MDIMP_He2 <- lapply(1:length(Av_DS_MDIMP_list),function(p){
  Comp_W_LM(fML_I_He2,fRE_RI,Av_DS_MDIMP_list[[p]],0,2,0)
})
```

```
names(Comp_I_RI_W_MDIMP_He2) <- names(Av_DS_MDIMP_list)
```

```
Comp_I_RI_W_MDIMP_He2_RLRT <- lapply(1:length(Av_DS_MDIMP_list),function(p){
  eREML_He2<-exactRLRT(lme(fML_I_He2, random = ~1|Lot, data = Av_DS_MDIMP_list[[p]], method="REML",na.action=na.omit))
})
```

```
names(Comp_I_RI_W_MDIMP_He2_RLRT) <- names(Av_DS_MDIMP_list)
```

```
fML_I_He3 <- formula(He3 ~ 1)
```

```
fRE_RI <- formula(~1|Lot)
```

```
Comp_I_RI_W_MDIMP_He3 <- lapply(1:length(Av_DS_MDIMP_list),function(p){
  Comp_W_LM(fML_I_He3,fRE_RI,Av_DS_MDIMP_list[[p]],0,2,0)
})
```

```
names(Comp_I_RI_W_MDIMP_He3) <- names(Av_DS_MDIMP_list)
```

```

Comp_I_RI_W_MDIMP_He3_RLRT <- lapply(1:length(Av_DS_MDIMP_list),function(p){
  eREML_He3<-exactRLRT(lme(fML_I_He3, random = ~1|Lot, data = Av_DS_MDIMP_list[[p]], method="REML",na.a
})
names(Comp_I_RI_W_MDIMP_He3_RLRT) <- names(Av_DS_MDIMP_list)

#Segons els models trobats s'ajusta per So1 IS RI, So2 IS RI i So3 IS RIS. ENS centrem directament en e

Comp_IS_RI_MDIMP_So2 <- lapply(1:length(Av_DS_MDIMP_list),function(p){
  lme(So3 ~ 1 + Temps, random = ~1|Lot,control=ctrl,
      method="REML",data=Av_DS_MDIMP_list[[p]],na.action=na.omit)
})
names(Comp_IS_RI_MDIMP_So2) <- names(Av_DS_MDIMP_list)
Comp_IS_RI_MDIMP_So2_RLRT <- lapply(1:length(Av_DS_MDIMP_list),function(p){
  eREML_So2<-exactRLRT(Comp_IS_RI_MDIMP_So2[[p]])
})
names(Comp_IS_RI_MDIMP_So2_RLRT) <- names(Av_DS_MDIMP_list)

#En el cas de múltiples efectes aleatoris fem el resultat de l'anova de comparació entre el model RI i

Comp_IS_RI_MDIMP_So3 <- lapply(1:length(Av_DS_MDIMP_list),function(p){
  lme(So3 ~ 1 + Temps, random = ~1|Lot,control=ctrl,
      method="REML",data=Av_DS_MDIMP_list[[p]],na.action=na.omit)
})
names(Comp_IS_RI_MDIMP_So3) <- names(Av_DS_MDIMP_list)
Comp_IS_RIS_MDIMP_So3 <- lapply(1:length(Av_DS_MDIMP_list),function(p){
  lme(So3 ~ 1 + Temps, random = ~1+Temps|Lot,control=ctrl,
      method="REML",data=Av_DS_MDIMP_list[[p]],na.action=na.omit)
})
names(Comp_IS_RIS_MDIMP_So3) <- names(Av_DS_MDIMP_list)

Comp_IS_RIS_MDIMP_So3_RLRT <- lapply(1:length(Av_DS_MDIMP_list),function(p){
  anova(Comp_IS_RI_MDIMP_So3[[p]],Comp_IS_RIS_MDIMP_So3[[p]])
})
names(Comp_IS_RIS_MDIMP_So3_RLRT) <- names(Av_DS_MDIMP_list)
Comp_IS_RI_MDIMP_So3_RLRT <- lapply(1:length(Av_DS_MDIMP_list),function(p){
  eREML_So3<-exactRLRT(Comp_IS_RI_MDIMP_So3[[p]])
})
names(Comp_IS_RI_MDIMP_So3_RLRT) <- names(Av_DS_MDIMP_list)

```

VISUALITZACIÓ RESULTATS

Scatterplots i Density plots

```

Av_DS_MDIMP_list <- list(Av_DS_MDIMP_list[[1]],Av_DS_MDIMP_list[[2]],VP_DS_kNN,VP_DS_BSR)
names(Av_DS_MDIMP_list) <- c("DS Original 0% VP",paste("DS",names(Av_DS_MD_list)[3]),
                             paste("DS",names(Av_DS_MD_list)[3],"kNN"),
                             paste("DS",names(Av_DS_MD_list)[3],"BSR"))

labs_1 <- matrix(paste(rep(IMP_Vars,each=length(Av_DS_MDIMP_list)),names(Av_DS_MDIMP_list),sep=" "),ncol=
labs_2 <- matrix(paste(rep(IMP_Vars,each=length(Av_DS_MDIMP_list)),"s/ Lot",names(Av_DS_MDIMP_list),sep=

Av_DS_MDIMP_plot_sc1 <- lapply(1:length(IMP_Vars),function(g){
  lapply(1:length(Av_DS_MDIMP_list),function(p){
    ggplot(Av_DS_MDIMP_list[[p]], aes(x=Temps, y=Av_DS_MDIMP_list[[p]][,paste(IMP_Vars[g])])) + geom_po

```

```

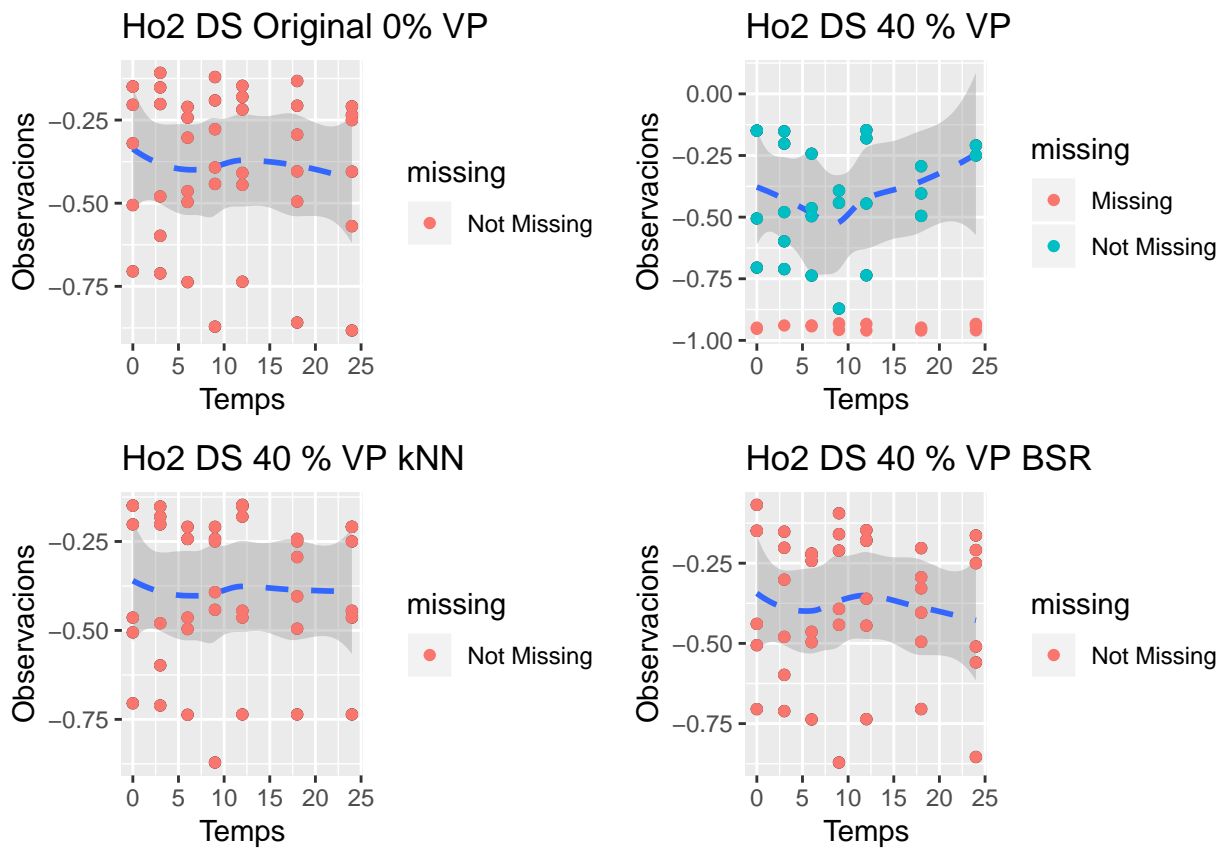
    geom_smooth(linetype="dashed") + labs(title=labs_1[p,g], y = "Observacions") +
    geom_miss_point()
  })
})

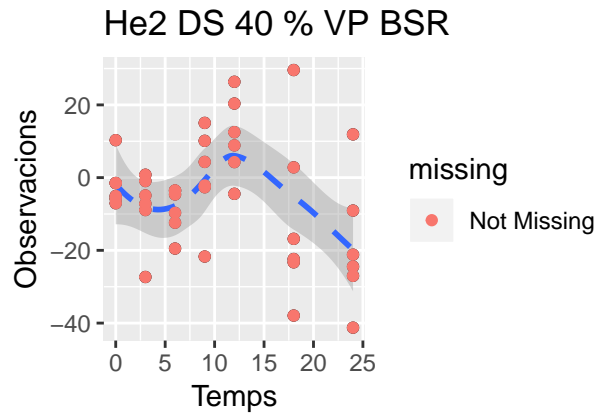
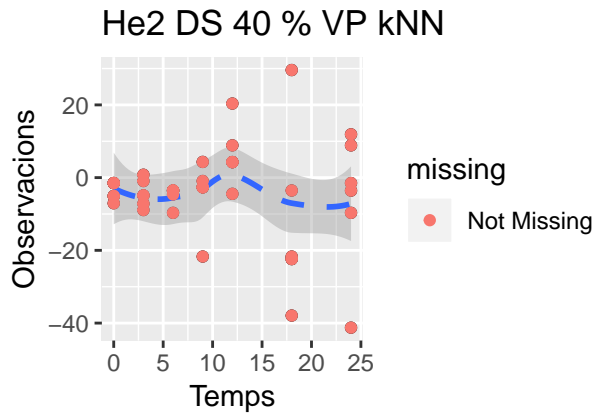
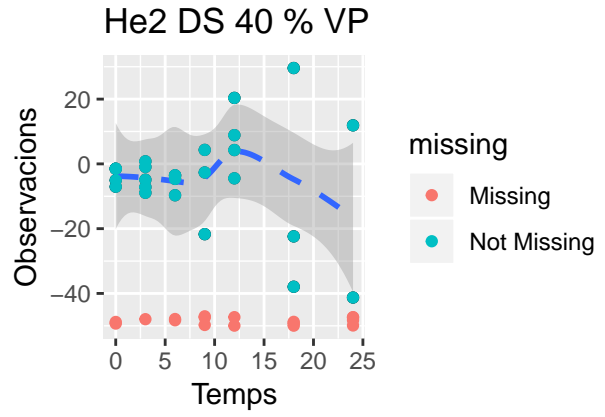
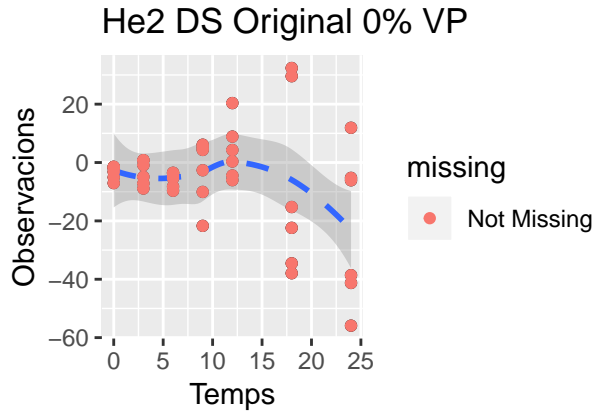
Av_DS_MDIMP_plot_sc2 <- lapply(1:length(IMP_Vars),function(g){
  lapply(1:length(Av_DS_MDIMP_list),function(p){
    ggplot(Av_DS_MDIMP_list[[p]], aes(x=Temps, y=Av_DS_MDIMP_list[[p]][,paste(IMP_Vars[g])], color=Lot,
    geom_point() + geom_smooth(method="lm",se=F) +
    labs(title=labs_2[p,g], y = "Observacions") +
    geom_miss_point()
  })
})

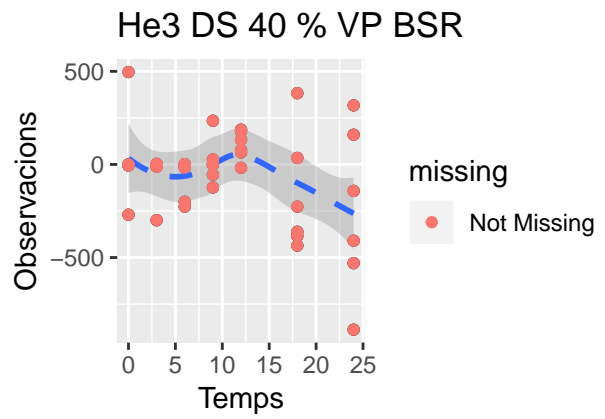
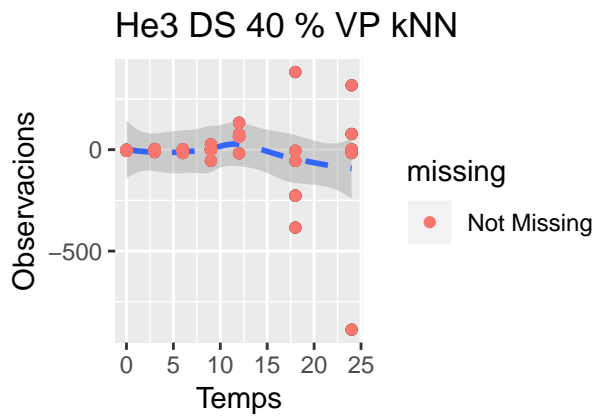
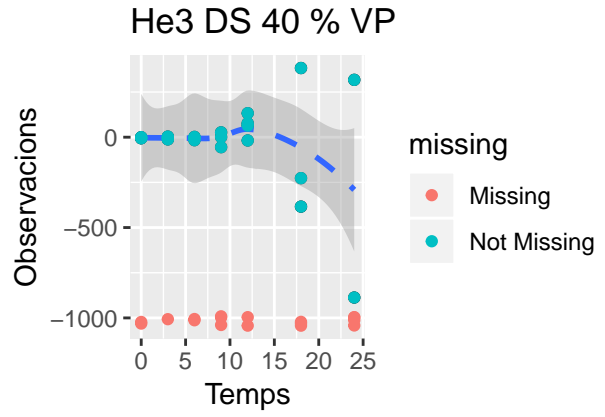
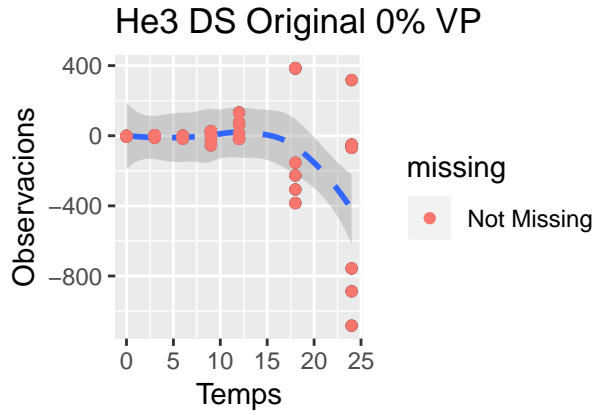
Av_DS_MDIMP_ydens_plot <- lapply(1:length(IMP_Vars),function(g){
  lapply(1:length(Av_DS_MDIMP_list),function(p){
    ggplot(Av_DS_MDIMP_list[[p]], aes(x=Av_DS_MDIMP_list[[p]][,paste(IMP_Vars[g])])) +
    geom_density(color="darkblue", fill="lightblue") +
    geom_vline(aes(xintercept=mean(IMP_Vars[g])), color="blue", linetype="dashed", size=1)+
    labs(title=labs_1[p,g], x = "Observacions")
  })
})

lapply(1:length(IMP_Vars),function(g){
  grid.arrange(grobs = Av_DS_MDIMP_plot_sc1[[g]])
})

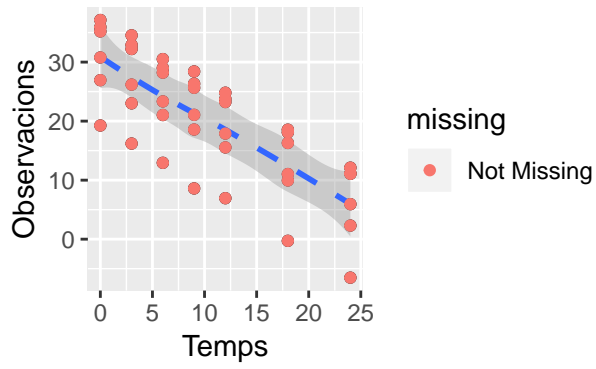
```



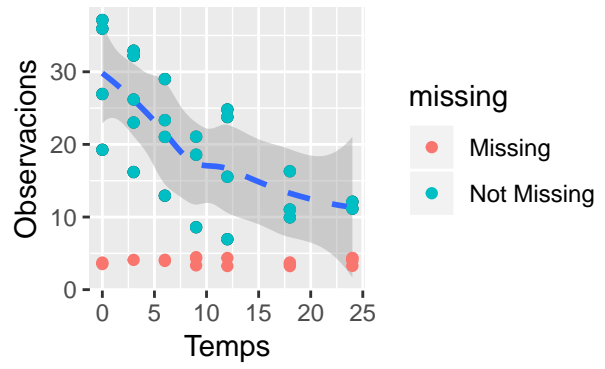




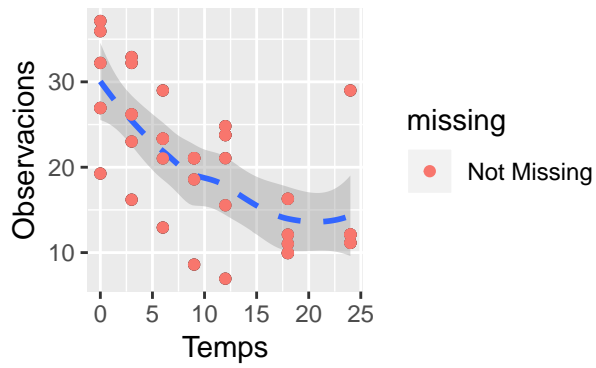
So2 DS Original 0% VP



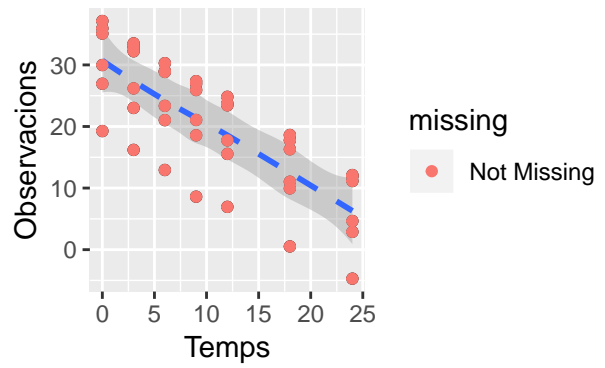
So2 DS 40 % VP



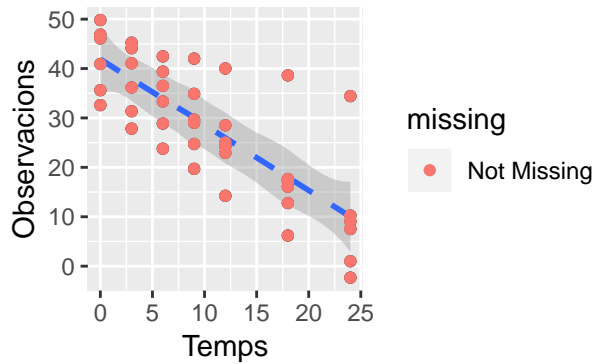
So2 DS 40 % VP kNN



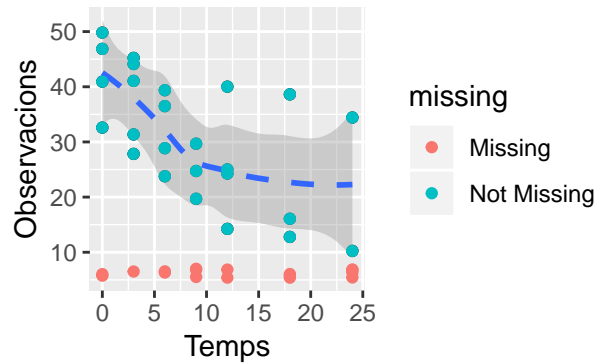
So2 DS 40 % VP BSR



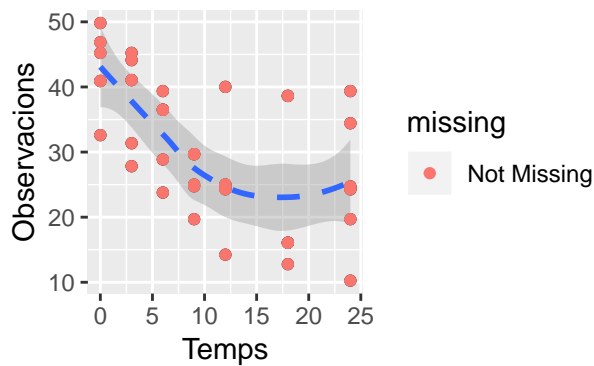
So3 DS Original 0% VP



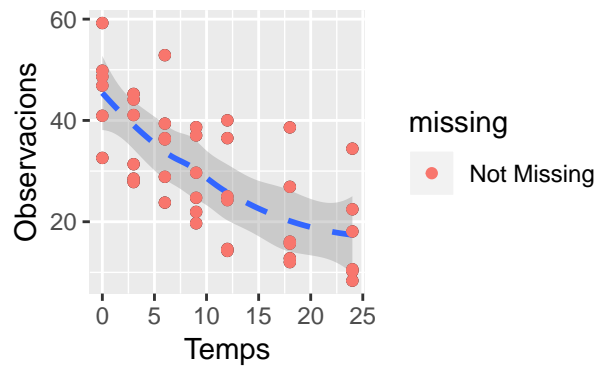
So3 DS 40 % VP



So3 DS 40 % VP kNN



So3 DS 40 % VP BSR



```
## [[1]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z    cells  name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[2]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z    cells  name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[3]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z    cells  name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[4]]
## TableGrob (2 x 2) "arrange": 4 grobs
```

```

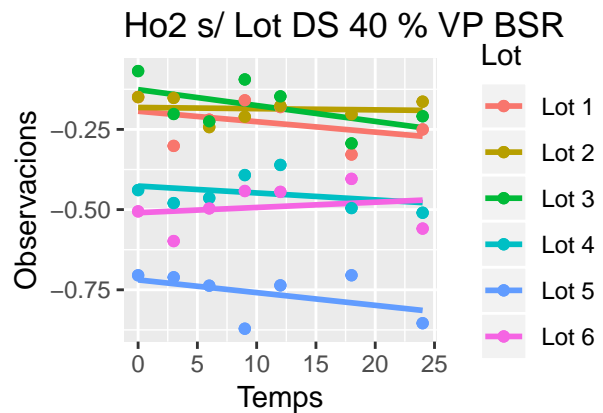
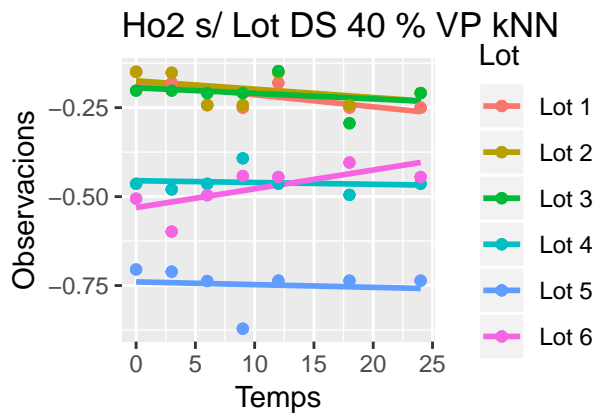
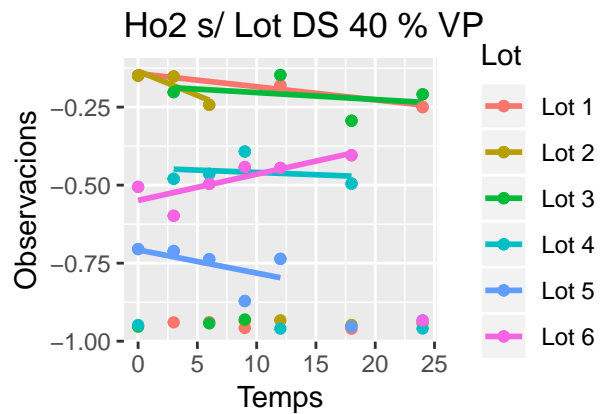
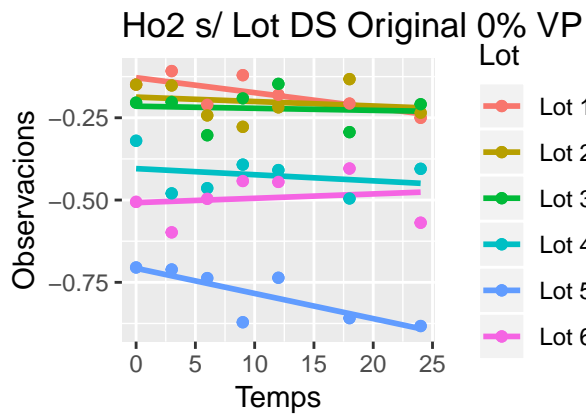
##      z      cells      name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[5]]
## TableGrob (2 x 2) "arrange": 4 grobs
##      z      cells      name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]

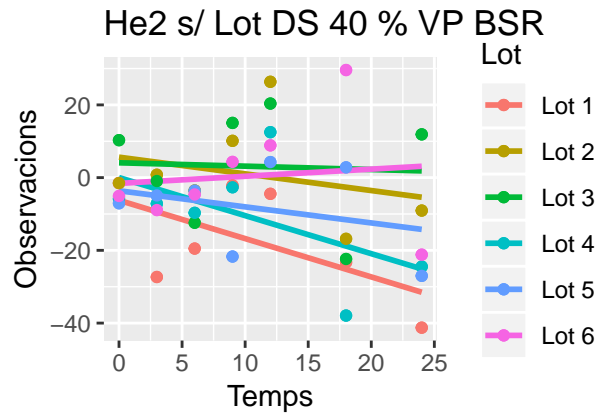
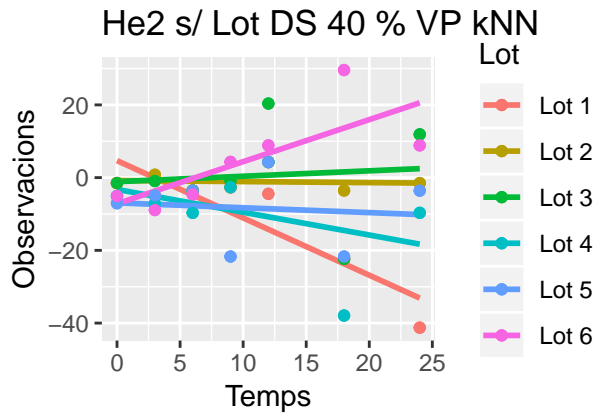
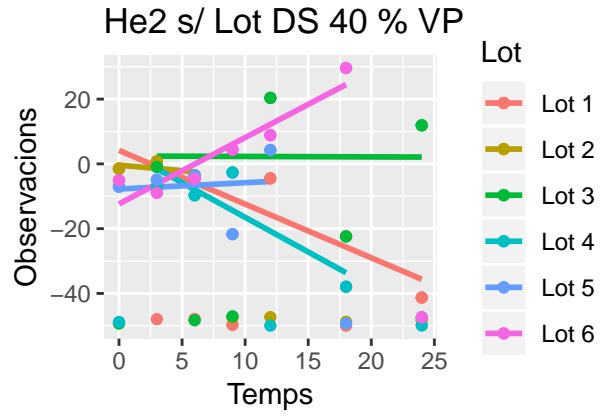
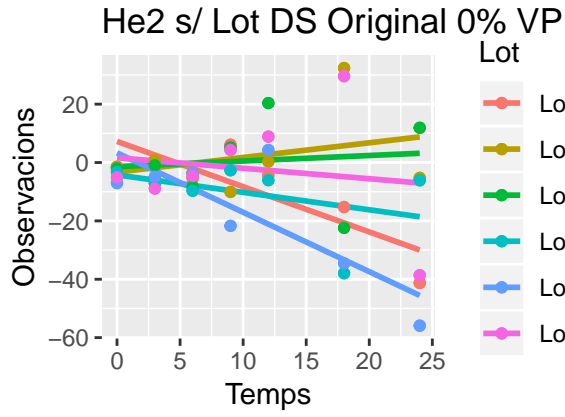
```

```

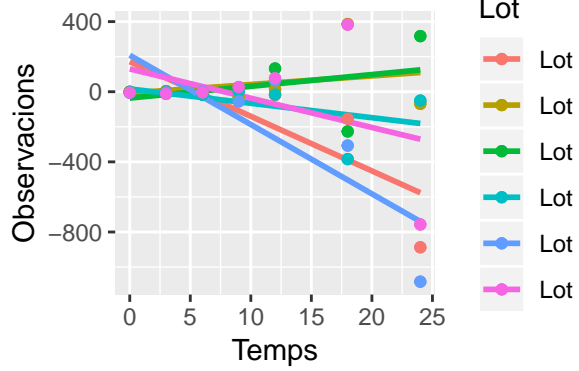
lapply(1:length(IMP_Vars),function(g){ #Per aquest gràfic només prenem
  grid.arrange(grobs = Av_DS_MDIMP_plot_sc2[[g]])
})

```

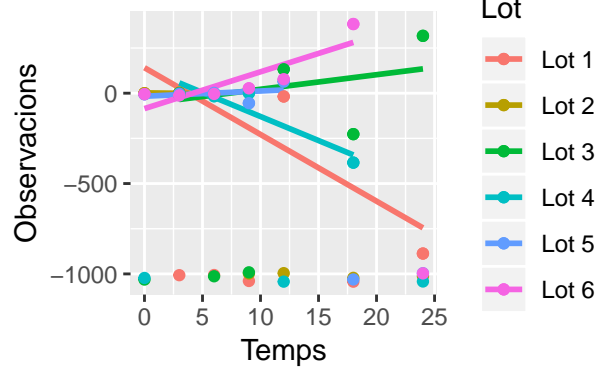




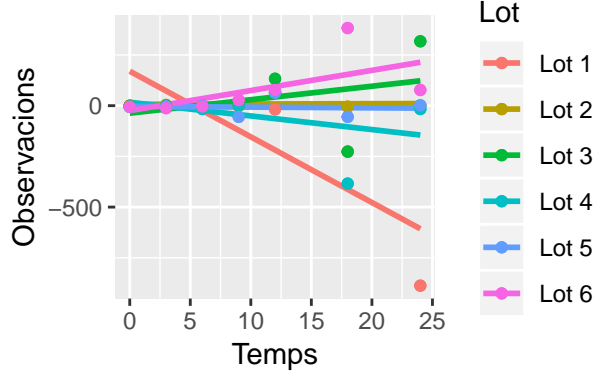
He3 s/ Lot DS Original 0% VP



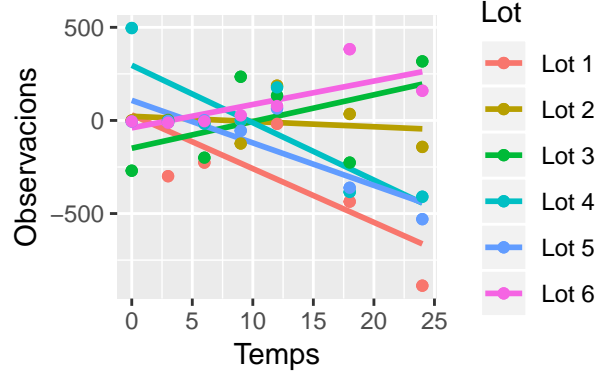
He3 s/ Lot DS 40 % VP



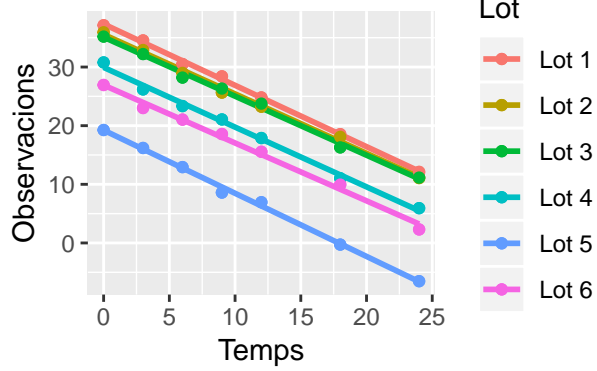
He3 s/ Lot DS 40 % VP kNN



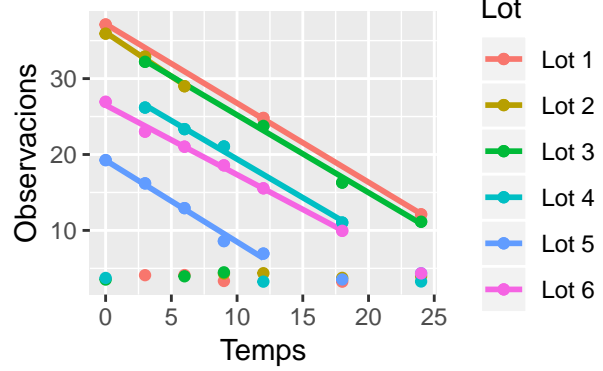
He3 s/ Lot DS 40 % VP BSR



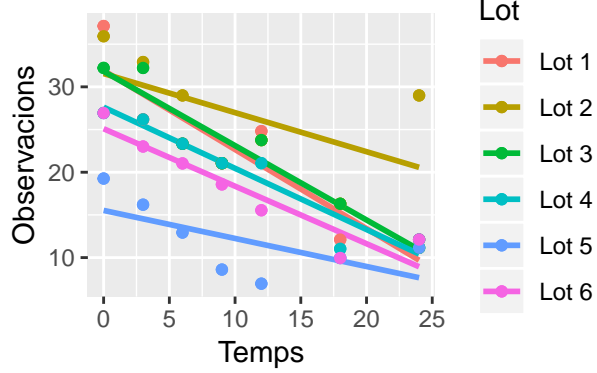
So2 s/ Lot DS Original 0% VP



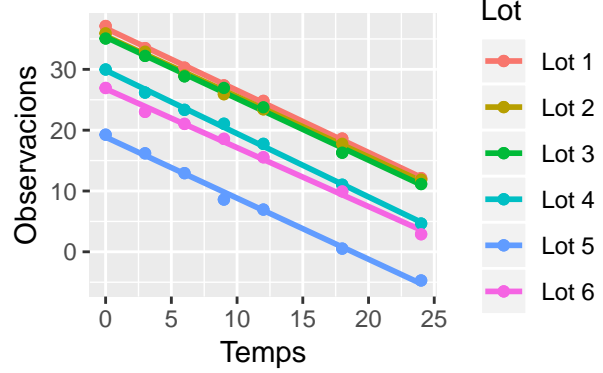
So2 s/ Lot DS 40 % VP

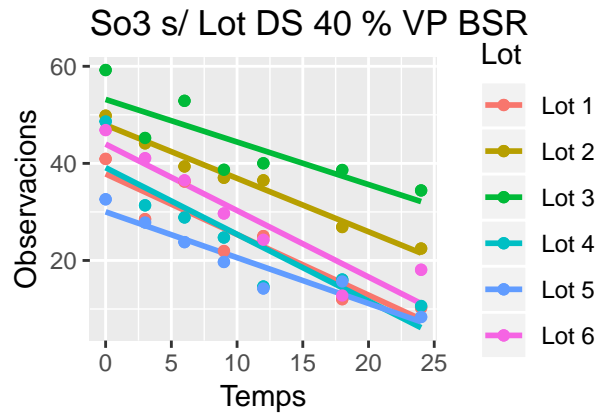
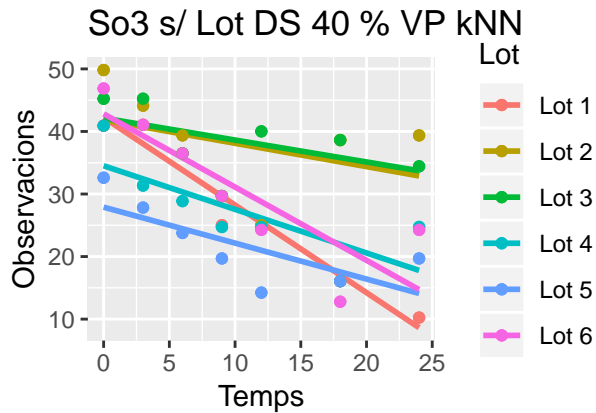
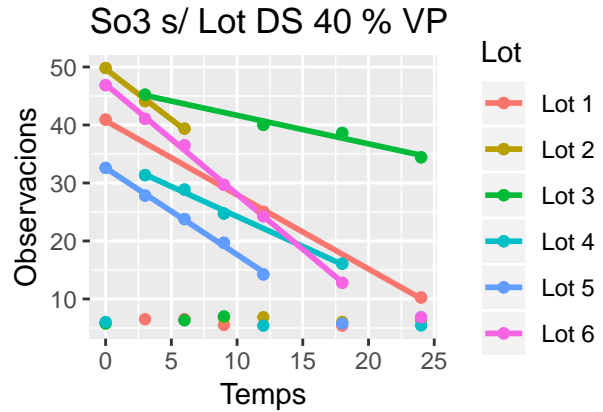
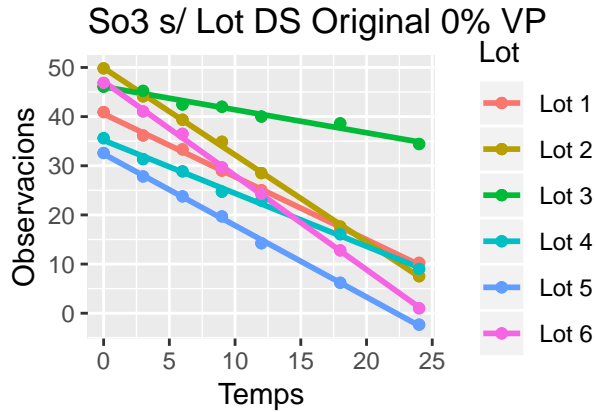


So2 s/ Lot DS 40 % VP kNN



So2 s/ Lot DS 40 % VP BSR





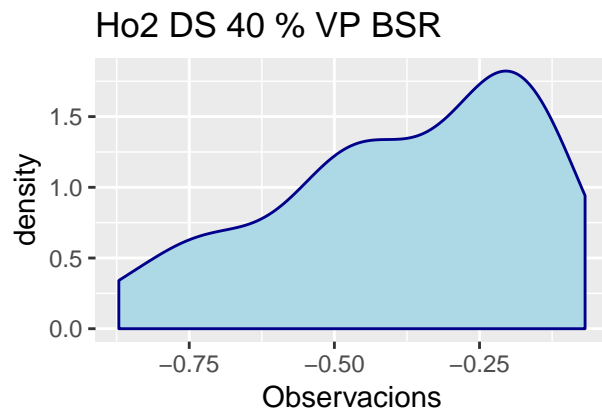
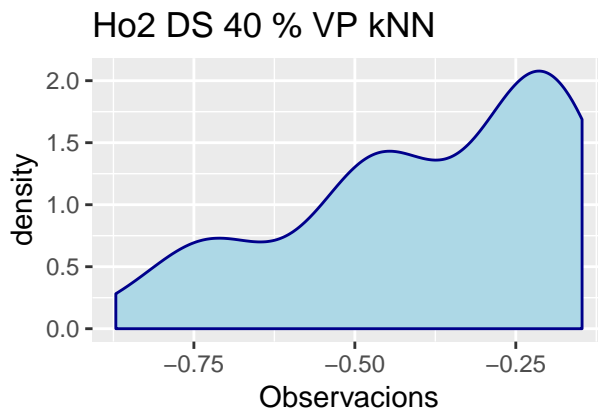
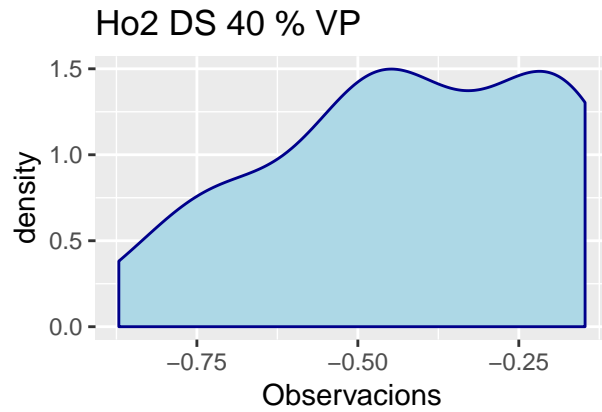
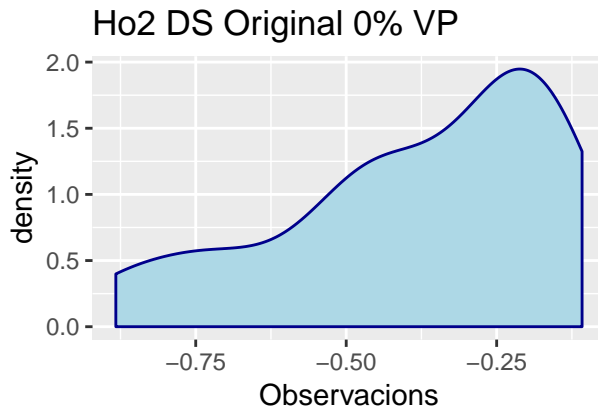
```
## [[1]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z    cells  name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[2]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z    cells  name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[3]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z    cells  name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[4]]
## TableGrob (2 x 2) "arrange": 4 grobs
```

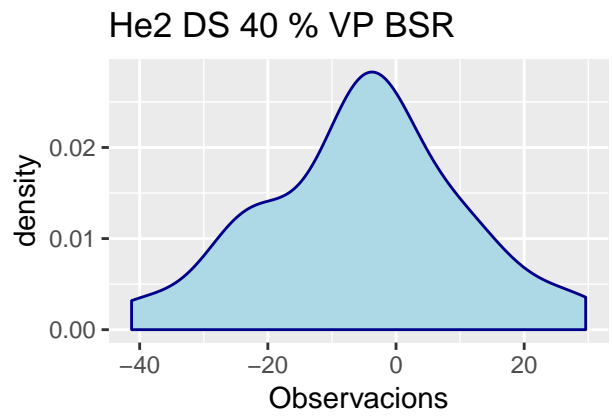
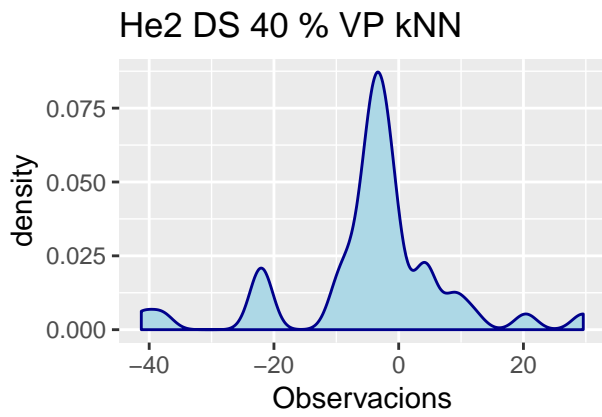
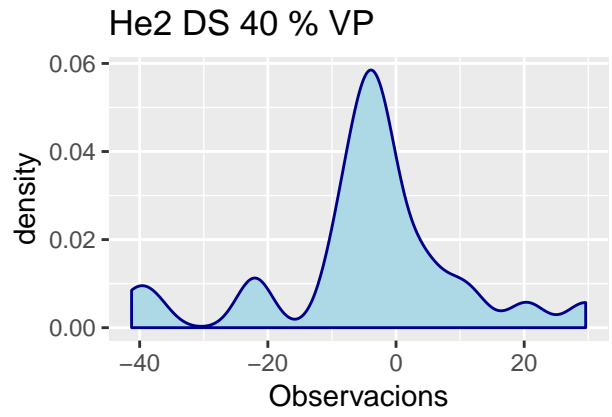
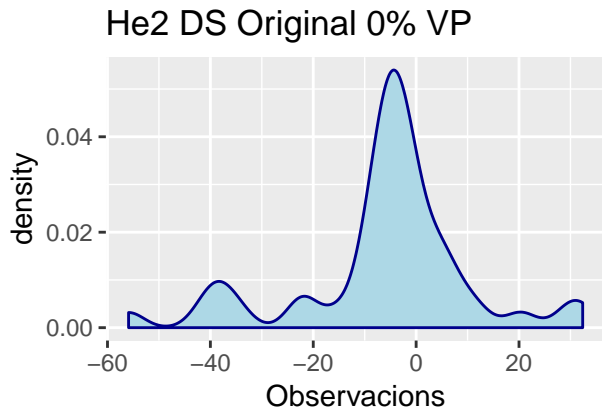


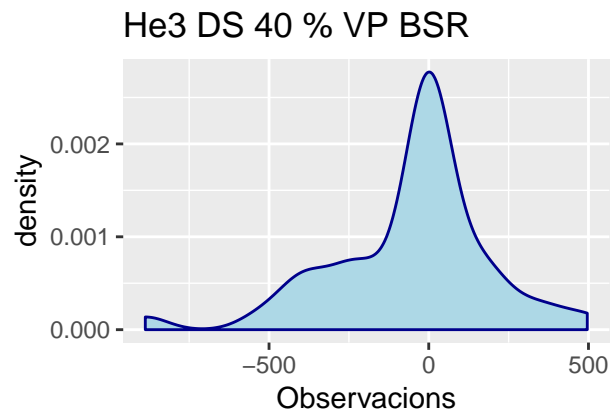
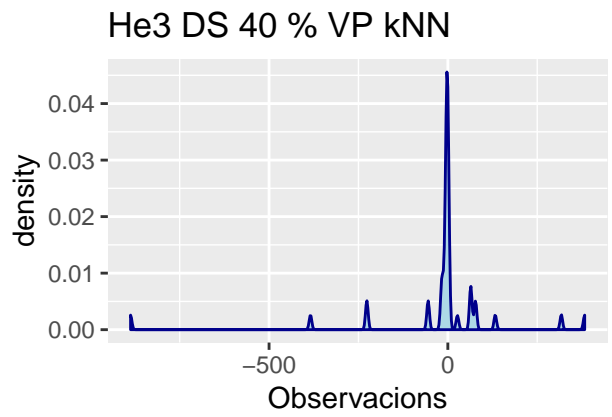
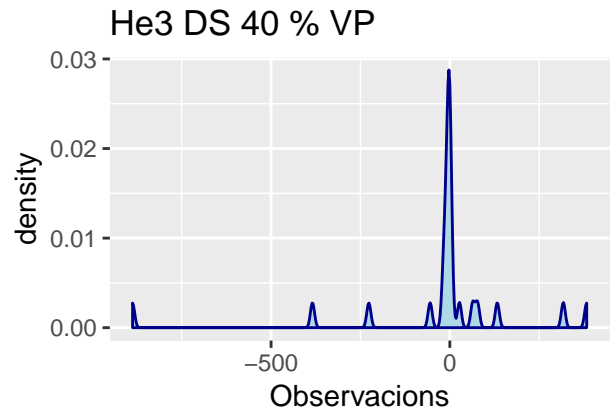
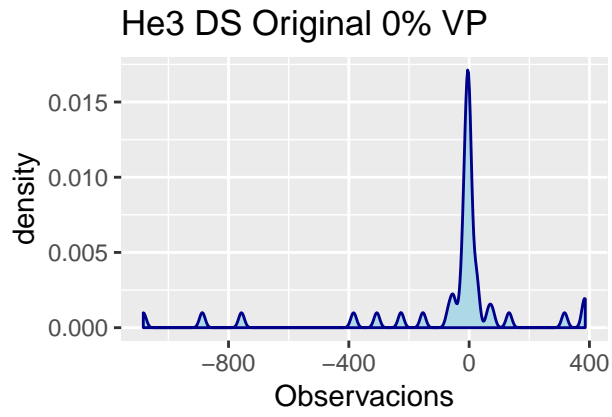
```

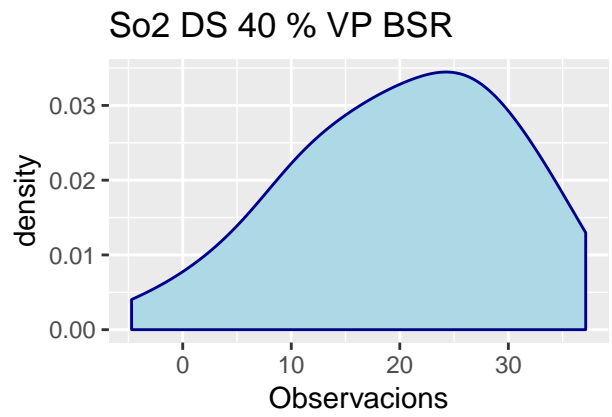
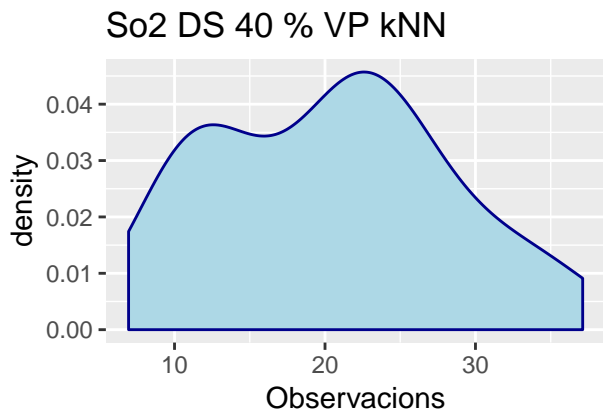
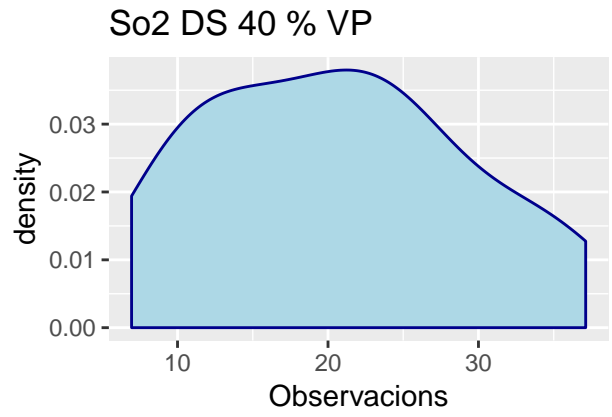
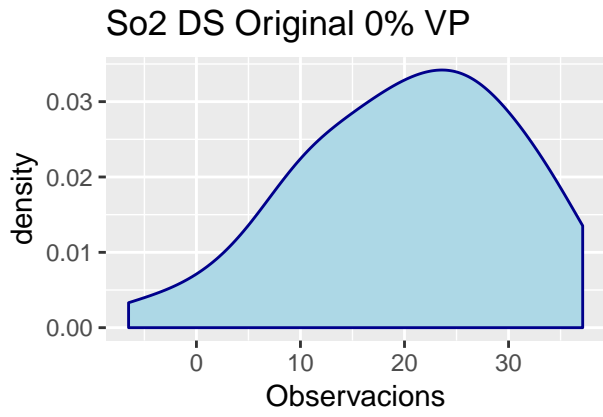
##      z      cells      name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[5]]
## TableGrob (2 x 2) "arrange": 4 grobs
##      z      cells      name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
lapply(1:length(IMP_Vars),function(g){
  grid.arrange(grobs = Av_DS_MDIMP_ydens_plot[[g]])
})

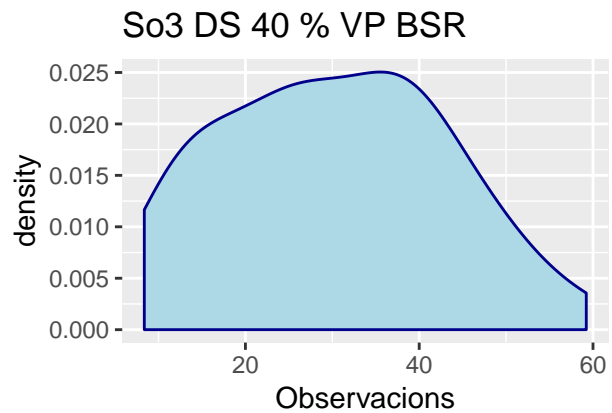
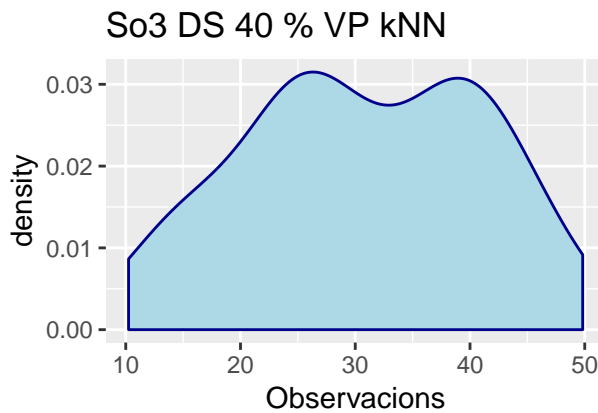
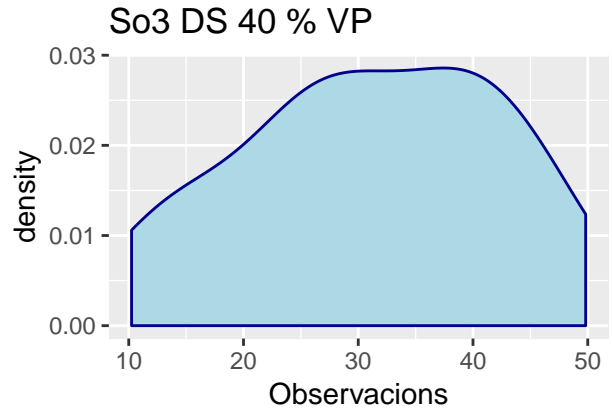
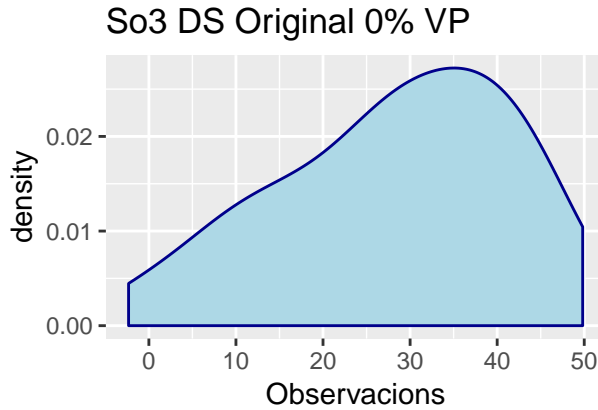
```











```
## [[1]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z   cells   name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[2]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z   cells   name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[3]]
## TableGrob (2 x 2) "arrange": 4 grobs
##   z   cells   name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[4]]
## TableGrob (2 x 2) "arrange": 4 grobs
```

```
## z cells name grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
##
## [[5]]
## TableGrob (2 x 2) "arrange": 4 grobs
## z cells name grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
```

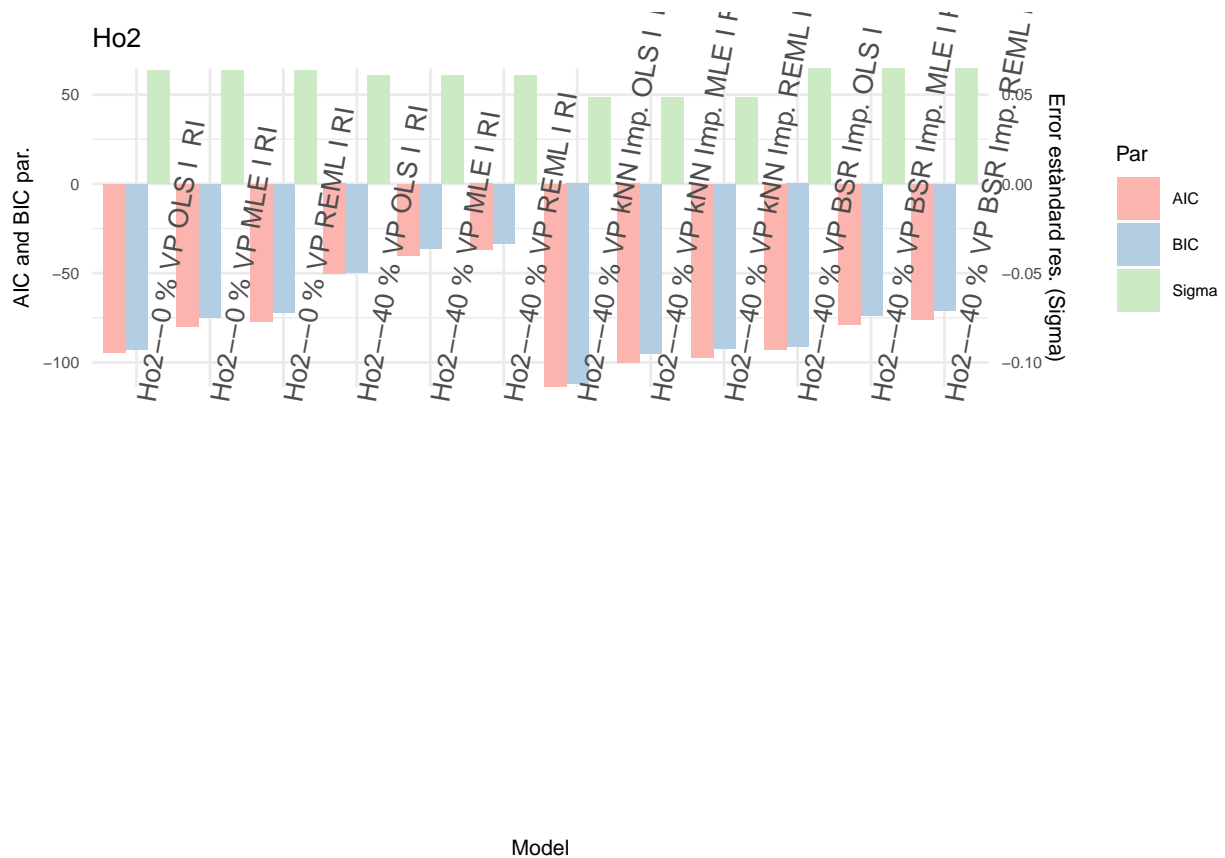
És interessant veure com es resolen els punts a vegades de manera més encertada o menys. Donaria la impressió que en els conjunts amb patrons de regressió més definits o més centrats en cada lot, l'aproximació per BSR dona més bon efecte, mentre que en altres conjunts sense patrons de regressió dona més bon resultat kNN

AIC/BIC/Sigma

```
Ho2_MDIMP_R <- ConvList(Comp_I_RI_MDIMP_Ho2)
s<-seq(1:3)
repeat {
  s<-c(s,s[seq(from=(length(s)-2),to=length(s))]+6)
  if (length(s)>=length(Ho2_MDIMP_R)/2){
    break
  }
}
Ho2_MDIMP_R <- Ho2_MDIMP_R[s]
names(Ho2_MDIMP_R) <- paste(rep(names(Comp_I_RI_MDIMP_Ho2),each=3),names(Ho2_MDIMP_R))

ML_Resum(Ho2_MDIMP_R,"Ho2")
```

```
## [[1]]
##
## Sigma AIC BIC
## Ho2--0 % VP OLS I RI 0.0635 -94.3562 -92.7727
## Ho2--0 % VP MLE I RI 0.0635 -80.0659 -74.8528
## Ho2--0 % VP REML I RI 0.0635 -77.1316 -71.9909
## Ho2--40 % VP OLS I RI 0.0607 -50.5295 -49.5850
## Ho2--40 % VP MLE I RI 0.0607 -40.0235 -36.3669
## Ho2--40 % VP REML I RI 0.0607 -37.0101 -33.4760
## Ho2--40 % VP kNN Imp. OLS I RI 0.0485 -113.6925 -112.1090
## Ho2--40 % VP kNN Imp. MLE I RI 0.0485 -100.3080 -95.0950
## Ho2--40 % VP kNN Imp. REML I RI 0.0485 -97.2228 -92.0821
## Ho2--40 % VP BSR Imp. OLS I RI 0.0647 -92.9336 -91.3501
## Ho2--40 % VP BSR Imp. MLE I RI 0.0647 -78.9671 -73.7541
## Ho2--40 % VP BSR Imp. REML I RI 0.0647 -75.9789 -70.8382
##
## [[2]]
```



Comentaris:

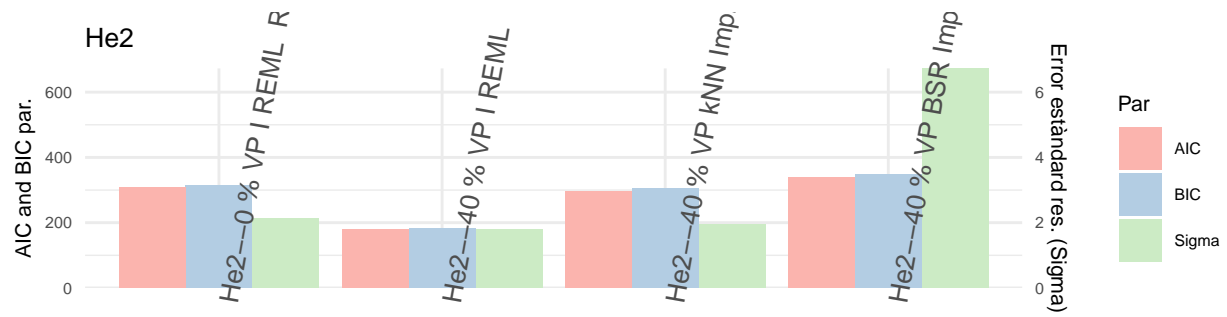
- Ho2: Tant per la part de AIC i BIC com la error estàndard residual, sembla que en el cas de BSR compensa millor les dades perdudes en aquest cas, cosa que ja semblava en els gràfics inicials.

```
He2_MDIMP_R <- ConvList(Comp_I_RI_W_MDIMP_He2)
s<-seq(from=7, to=length(He2_MDIMP_R), by=12)
```

```
He2_MDIMP_R <- He2_MDIMP_R[s]
names(He2_MDIMP_R) <- paste(names(Comp_I_RI_W_MDIMP_He2), names(He2_MDIMP_R))
```

```
ML_Resum(He2_MDIMP_R, "He2")
```

```
## [[1]]
##           Sigma      AIC      BIC
## He2--0 % VP I REML RI VarExp      2.1342 308.2841 315.1383
## He2--40 % VP I REML RI VarExp      1.8052 178.6783 183.3905
## He2--40 % VP kNN Imp. I REML RI VarExp 1.9333 296.9225 303.7768
## He2--40 % VP BSR Imp. I REML RI VarExp 6.7276 339.5139 346.3682
##
## [[2]]
```



Model

```

He3_MDIMP_R <- ConvList(Comp_I_RI_W_MDIMP_He3)
s<-seq(from=7, to=length(He3_MDIMP_R), by=12)

He3_MDIMP_R <- He3_MDIMP_R[s]
names(He3_MDIMP_R) <- paste(names(Comp_I_RI_W_MDIMP_He3), names(He3_MDIMP_R))

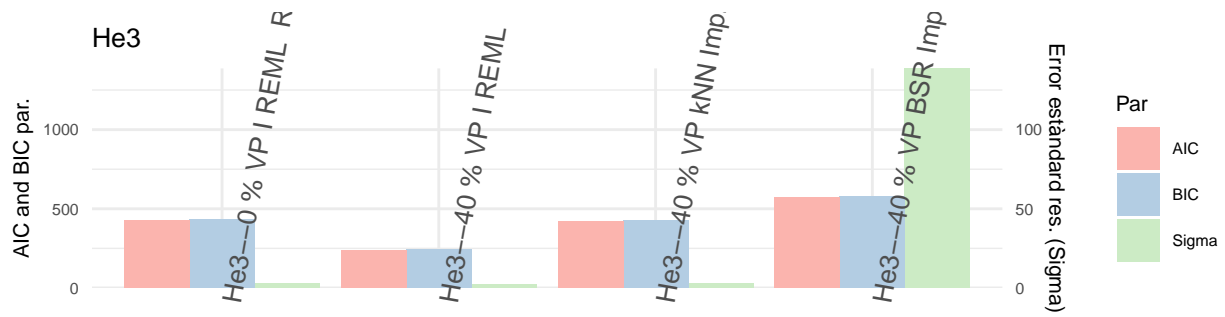
ML_Resum(He3_MDIMP_R, "He3")

```

```

## [[1]]
##              Sigma      AIC      BIC
## He3--0 % VP I REML RI VarExp    2.7460 427.4230 434.2773
## He3--40 % VP I REML RI VarExp    2.2340 239.8691 244.5813
## He3--40 % VP kNN Imp. I REML RI VarExp 2.7302 419.8816 426.7359
## He3--40 % VP BSR Imp. I REML RI VarExp 138.6648 569.7275 576.5818
##
## [[2]]

```

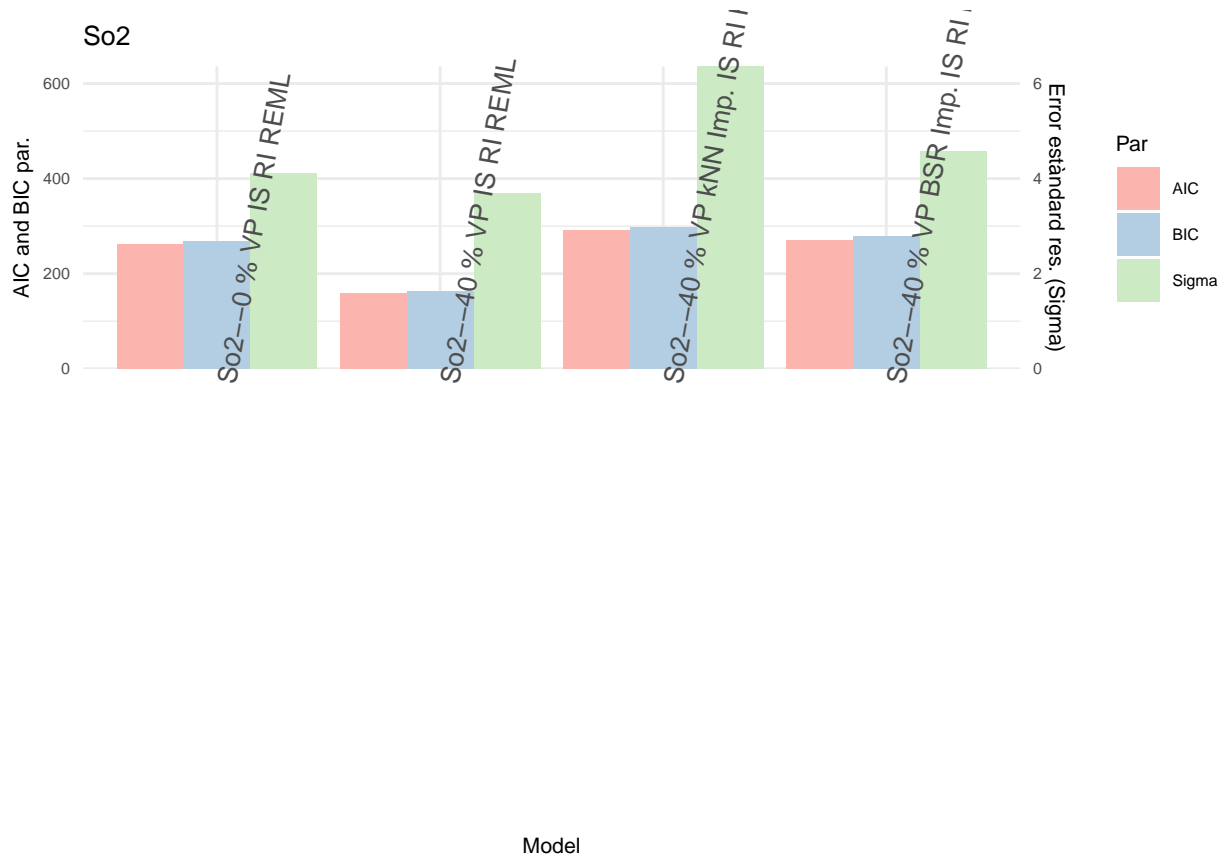
Model

Comentaris:

- He2/He3: Semblaria ajustar millor en aquest aspecte el model amb kNN a nivell de error estàndard residual, tot i que per part dels indicadors AIC i BIC estarien situats a nivells similars.

```
So2_MDIMP_R <- Comp_IS_RI_MDIMP_So2
names(So2_MDIMP_R) <- paste(names(Comp_IS_RI_MDIMP_So2), c("IS RI REML"), sep=" ")
ML_Resum(So2_MDIMP_R, "So2")
```

```
## [[1]]
##              Sigma      AIC      BIC
## So2--0 % VP IS RI REML    4.1099 262.5228 269.2783
## So2--40 % VP IS RI REML    3.6899 159.4850 164.0270
## So2--40 % VP kNN Imp. IS RI REML 6.3607 291.0333 297.7889
## So2--40 % VP BSR Imp. IS RI REML 4.5714 270.9799 277.7354
##
## [[2]]
```



```

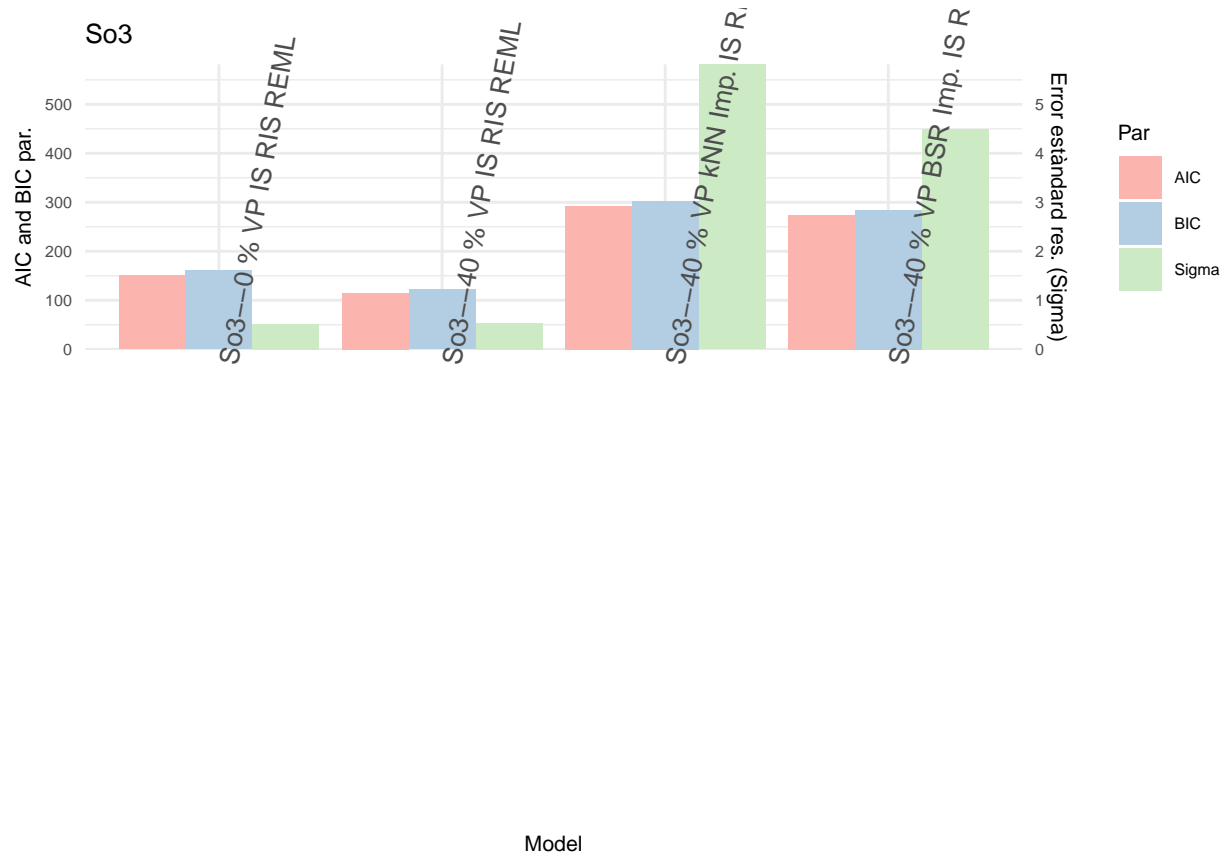
So3_MDIMP_R <- Comp_IS_RIS_MDIMP_So3
names(So3_MDIMP_R) <- paste(names(Comp_IS_RIS_MDIMP_So3), c("IS RIS REML"), sep=" ")
ML_Resum(So3_MDIMP_R, "So3")

```

```

## [[1]]
##                               Sigma      AIC      BIC
## So3--0 % VP IS RIS REML      0.5188 150.5445 160.6778
## So3--40 % VP IS RIS REML      0.5401 114.7276 121.5405
## So3--40 % VP kNN Imp. IS RIS REML 5.8154 292.0694 302.2027
## So3--40 % VP BSR Imp. IS RIS REML 4.4938 274.3575 284.4907
##
## [[2]]

```



Comentaris:

- So2: Semblaria clar en aquesta part que el model que ajusta millor és el corregit per BSR i de fet és una correcció basada en una regressió, cosa que tindria sentit al ser un model amb marcada regressió per lot.
- So3: La aleatorietat de pendents fa que els mètodes utilitzats per suplir les dades faltants no siguin efectius en aquest cas donant resultats dels 3 indicadors significativament més diferents que l'ajustat amb eliminació dels valors perduts.

Contrast RLRT

Comparatiu de la prova de significació de l'efecte aleatori prenent com a model vàlid el model calcular per ML amb REML:

```

HoVars <- c("Ho2")

MDIMP_I_RI_Ho_REML_RLRT <- data.frame(Dataset = paste(rep(HoVars,each=length(Comp_I_RI_MDIMP_Ho2_RLRT))
                                                    paste("I RI","REML",sep=" ")))

MDIMP_I_RI_Ho_REML_RLRT$VP <- rep(names(Comp_I_RI_MDIMP_Ho2_RLRT),length(HoVars))

Comp_MDIMP_RI_Ho_0 <- list(Comp_I_RI_MDIMP_Ho2_RLRT)
Comp_MDIMP_RI_Ho <- ConvList(Comp_MDIMP_RI_Ho_0)

MDIMP_I_RI_Ho_REML_RLRT$p_valor <- sapply(1:length(Comp_MDIMP_RI_Ho),function(l1){
  if (class(Comp_MDIMP_RI_Ho[[l1]])!="htest"){
    return("-")
  } else {

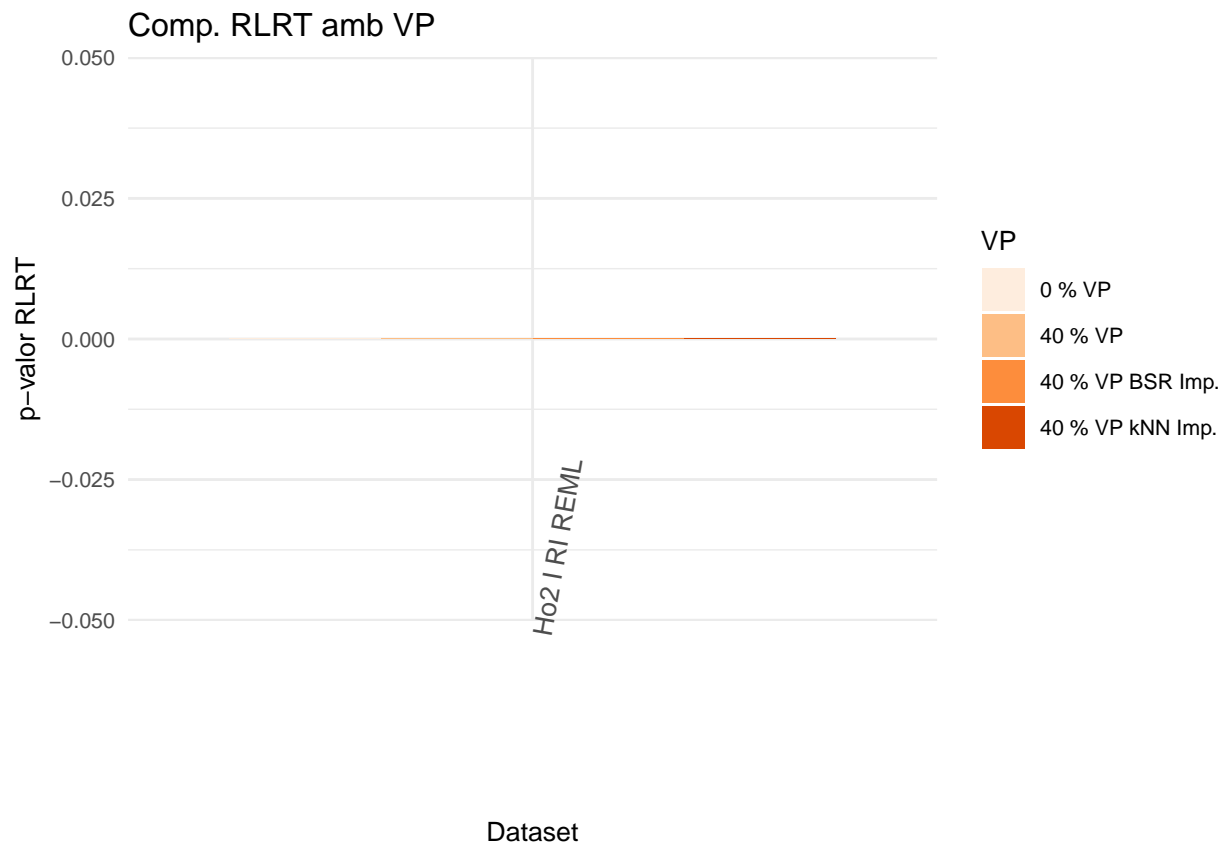
```

```

    round(Comp_MDIMP_RI_Ho[[11]]$p.value,4)
  }
})

ggplot(data=MDIMP_I_RI_Ho_REML_RLRT, aes(x=Dataset, y=p_valor, fill=VP)) +
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Oranges") + theme_minimal() +
  labs(title="Comp. RLRT amb VP",y="p-valor RLRT") +
  theme(text = element_text(size=10),axis.text.x = element_text(angle=80,size = 10,hjust=0))

```



Comentaris:

- Ho2: L'efecte aleatori resulta significatiu en tots els casos.

```

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

MDIMP_I_RI_W_He_REML_RLRT <- data.frame(Dataset = paste(rep(HeVars,each=length(Comp_I_RI_W_MDIMP_He2_RLRT),
  paste("I RI W","REML",sep=" ")))

MDIMP_I_RI_W_He_REML_RLRT$VP <- rep(names(Comp_I_RI_W_MDIMP_He2_RLRT),length(HeVars))

Comp_MDIMP_RI_W_He_0 <- list(Comp_I_RI_W_MDIMP_He2_RLRT,Comp_I_RI_W_MDIMP_He3_RLRT)
Comp_MDIMP_RI_W_He <- ConvList(Comp_MDIMP_RI_W_He_0)

MDIMP_I_RI_W_He_REML_RLRT$p_valor <- sapply(1:length(Comp_MDIMP_RI_W_He),function(l1){
  if (class(Comp_MDIMP_RI_W_He[[l1]])!="htest"){
    return("-")
  }
})

```

```

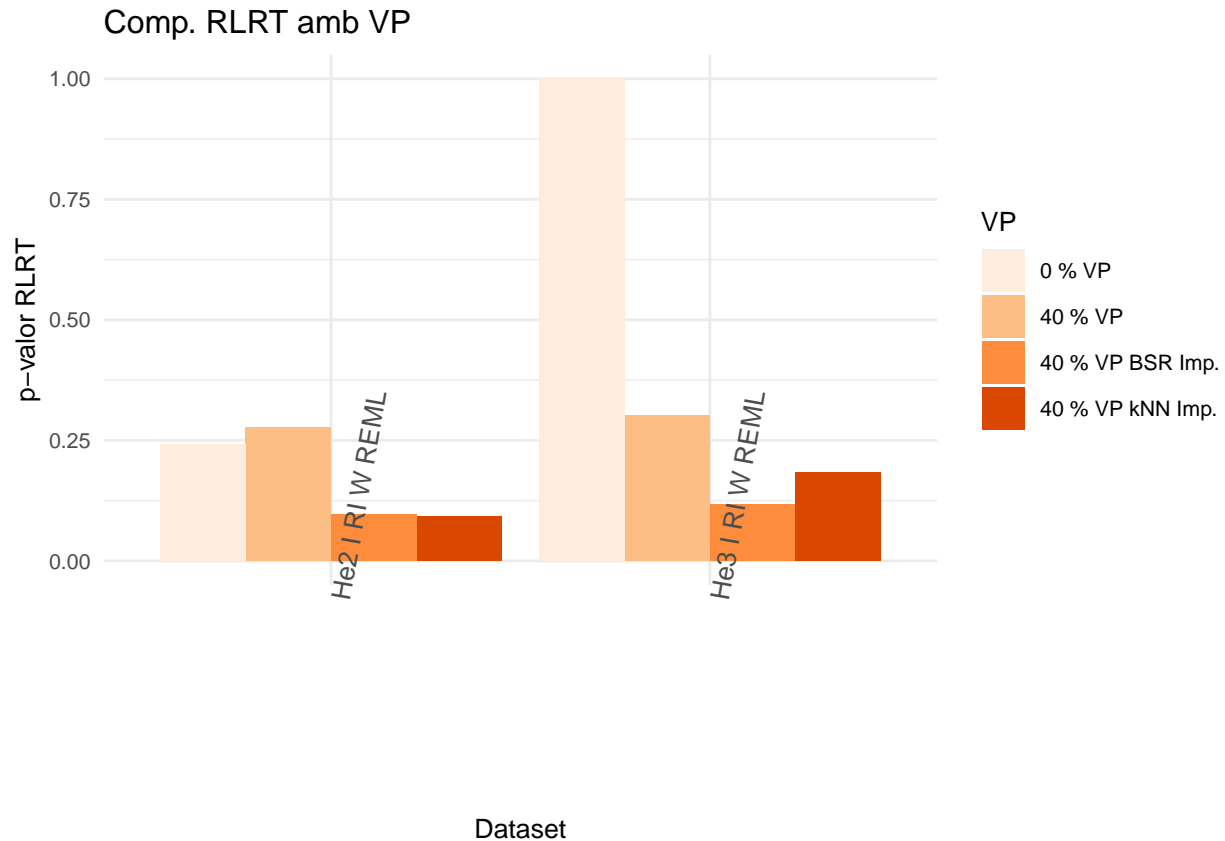
} else {
  round(Comp_MDIMP_RI_W_He[[11]]$p.value,4)
}
})

```

```

ggplot(data=MDIMP_I_RI_W_He_REML_RLRT, aes(x=Dataset, y=p_valor, fill=VP)) +
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Oranges") + theme_minimal() +
  labs(title="Comp. RLRT amb VP",y="p-valor RLRT") +
  theme(text = element_text(size=10),axis.text.x = element_text(angle=80,size = 10,hjust=0))

```



Comentaris:

- He2/He3: Cap dels 2 mètodes per suplir valors semblaria igualar la original i inclús el model amb els valors eliminats semblaria tenir més semblança.

```

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

MDIMP_I_RI_S_So_REML_RLRT <- data.frame(Dataset = paste(rep(c(SoVars, "So3(RIS)"),
  each=length(Comp_IS_RI_MDIMP_So2_RLRT)),
  paste("IS RI/RIS", "REML", sep=" ")))

MDIMP_I_RI_S_So_REML_RLRT$VP <- rep(names(Comp_IS_RI_MDIMP_So2_RLRT), (length(SoVars)+1))

Comp_MDIMP_RI_S_So_0 <- list(Comp_IS_RI_MDIMP_So2_RLRT,
  Comp_IS_RI_MDIMP_So3_RLRT, Comp_IS_RIS_MDIMP_So3_RLRT)
Comp_MDIMP_RI_S_So <- ConvList(Comp_MDIMP_RI_S_So_0)

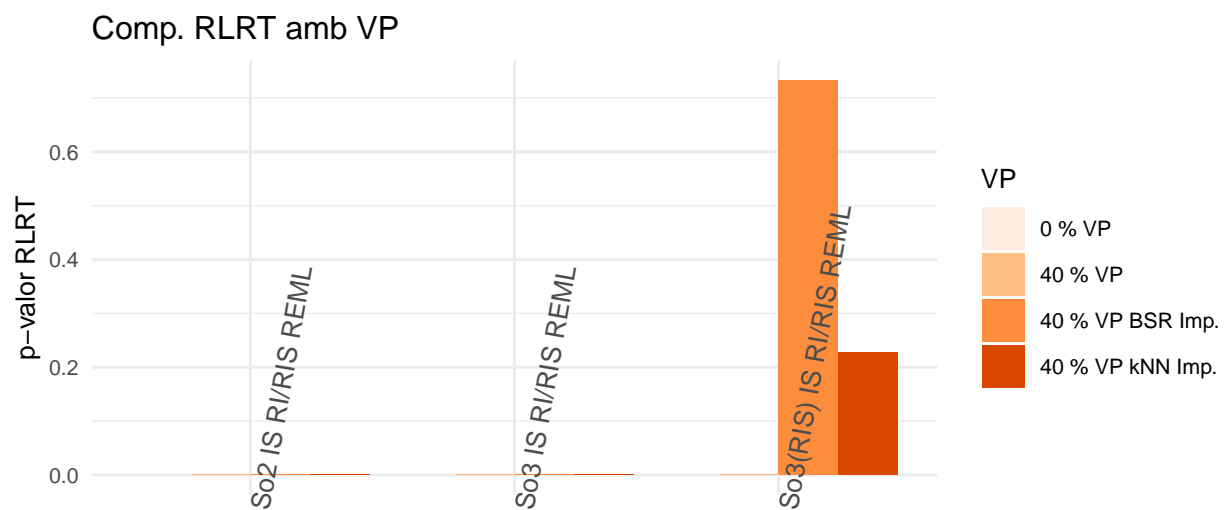
```

```

MDIMP_I_RI_S_So_REML_RLRT$p_valor <- sapply(1:length(Comp_MDIMP_RI_S_So),function(l1){
  if (class(Comp_MDIMP_RI_S_So[[l1]])[1]=="htest"){
    round(Comp_MDIMP_RI_S_So[[l1]]$p.value,4)
  } else {
    round(Comp_MDIMP_RI_S_So[[l1]]$"p-value"[2],4)
  }
})

ggplot(data=MDIMP_I_RI_S_So_REML_RLRT, aes(x=Dataset, y=p_valor, fill=VP)) +
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Oranges") + theme_minimal() +
  labs(title="Comp. RLRT amb VP",y="p-valor RLRT") +
  theme(text = element_text(size=10),axis.text.x = element_text(angle=80,size = 10,hjust=0))

```



Dataset

Comentaris: Tots els efectes semblen significatius en els conjunts So2 i So3 independentment de la pèrdua de dades. Semblen doncs necessaris igualment per explicar la variabilitat. Els dos sistemes de correcció de dades donen models encara més diferents que el de l'eliminació de dades perdudes en el cas de So3. Possiblement al tenir una estructura més complexa caldria un mètode més addient per substitució de valors.

Paràmetres Efectes Aleatoris

Prenent igualment el model REML com a vàlid, veiem els canvis en els components de la variància de l'efecte aleatori (covariància de la matriu de variància-covariància entre repeticions) i la correlació que en resulta entre repeticions:

```

#Generació de les llistes a mida de models per poder fer la taula de paràmetres amb el model REML
Ho2_MDIMP_R_2 <- ConvList(Comp_I_RI_MDIMP_Ho2)

```

```

Ho_MDIMP_R_2_0 <- list(Ho2_MDIMP_R_2)
Ho_MDIMP_R_2 <- ConvList(Ho_MDIMP_R_2_0)

s<-seq(from=6, to=length(Ho_MDIMP_R_2), by=6)
Ho_MDIMP_R_2r <- Ho_MDIMP_R_2[s]

Ho_MDIMP_RIpar <- data.frame(Dataset = paste(rep(HoVars,each=length(Comp_I_RI_MDIMP_Ho2)),rep(names(Comp_I_RI_MDIMP_Ho2),each=length(Comp_I_RI_MDIMP_Ho2))),rep(names(Comp_I_RI_MDIMP_Ho2),each=length(Comp_I_RI_MDIMP_Ho2)))

Ho_MDIMP_RIpar <- rbind(Ho_MDIMP_RIpar,Ho_MDIMP_RIpar)

Ho_MDIMP_RIpar$Par_Tipus <- factor(rep(c("RI Variància","RI Correlació"),each=(nrow(Ho_MDIMP_RIpar)/2)),
                                levels=c("RI Variància","RI Correlació"))

Ho_MDIMP_RIpar_VarRI <- sapply(1:length(Ho_MDIMP_R_2r),function(B){
  round(as.numeric(VarCorr(Ho_MDIMP_R_2r[[B]])[1]),4)
})

Ho_MDIMP_RIpar_CorrRI <- sapply(1:length(Ho_MDIMP_R_2r),function(B){
  a<-getVarCov(Ho_MDIMP_R_2r[[B]],type="marginal")
  b<-cov2cor(a[[1]])
  return(round(b[1,2],4))
})

fact <- if(mean(abs(Ho_MDIMP_RIpar_VarRI)) >= mean(abs(Ho_MDIMP_RIpar_CorrRI))){ #Es fa un càlcul del f
  10^(round(log10(abs(mean(Ho_MDIMP_RIpar_VarRI))/abs(mean(Ho_MDIMP_RIpar_CorrRI))))
} else {
  1/(10^(round(log10(abs(mean(Ho_MDIMP_RIpar_CorrRI))/abs(mean(Ho_MDIMP_RIpar_VarRI))))))
}

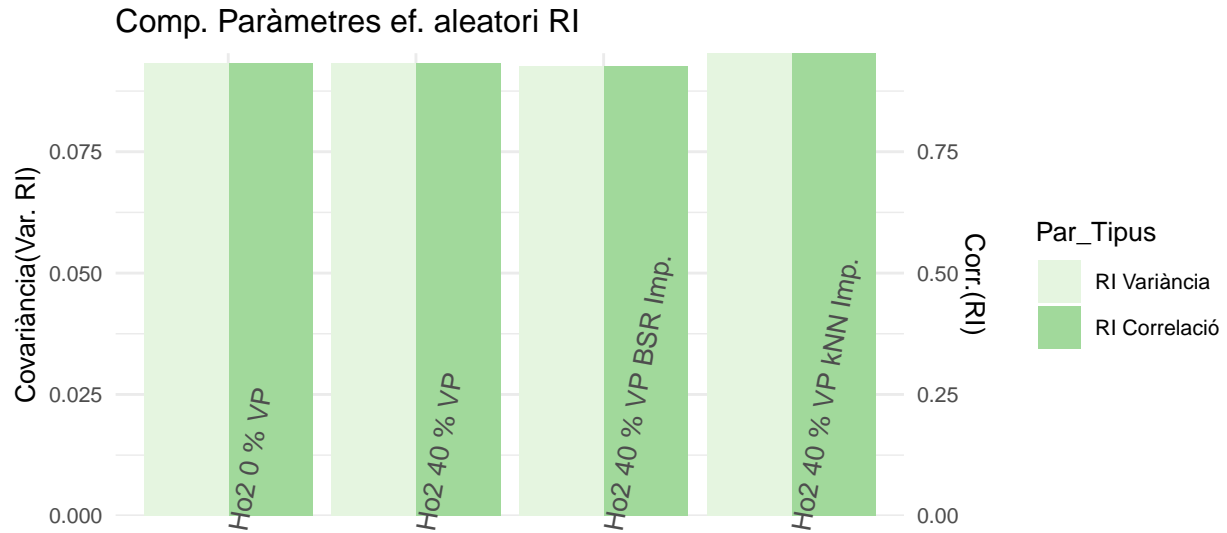
if (fact==0){
  fact <-0.0001
}

Ho_MDIMP_RIpar_CorrRI <- Ho_MDIMP_RIpar_CorrRI*fact

Ho_MDIMP_RIpar$Par <- c(Ho_MDIMP_RIpar_VarRI,Ho_MDIMP_RIpar_CorrRI)

ggplot(data=Ho_MDIMP_RIpar, aes(x=Dataset, y=Par, fill=Par_Tipus)) +
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Greens") + theme_minimal() +
  labs(title="Comp. Paràmetres ef. aleatori RI",y="Covariància(Var. RI)" ) +
  scale_y_continuous(expand = c(0, 0),sec.axis = sec_axis(~./fact, name = "Corr.(RI)") ) +
  theme(text = element_text(size=10),axis.text.x = element_text(angle=80,size = 10,hjust=0))

```



Dataset

Comentaris:

- Ho2: La centralitat de punts per Lot fa que tot i la pèrdua de dades es mantingui en el model el grau de covariància/correlació entre repeticions degut a l'efecte aleatori.

#Generació de les llistes a mida de models per poder fer la taula de paràmetres amb el model REML

```

He2_MDIMP_R_2 <- ConvList(Comp_I_RI_W_MDIMP_He2)
He3_MDIMP_R_2 <- ConvList(Comp_I_RI_W_MDIMP_He3)
He_MDIMP_R_2_0 <- list(He2_MDIMP_R_2,He3_MDIMP_R_2)
He_MDIMP_R_2 <- ConvList(He_MDIMP_R_2_0)

s<-seq(from=8, to=length(He_MDIMP_R_2), by=12)
He_MDIMP_R_2r <- He_MDIMP_R_2[s]

He_MDIMP_RIpar <- data.frame(Dataset = paste(rep(HeVars,each=length(Comp_I_RI_W_MDIMP_He2)),rep(names(C
He_MDIMP_RIpar <- rbind(He_MDIMP_RIpar,He_MDIMP_RIpar)

He_MDIMP_RIpar$Par_Tipus <- factor(rep(c("RI Variància","RI Correlació"),each=(nrow(He_MDIMP_RIpar)/2))
                                levels=c("RI Variància","RI Correlació"))

He_MDIMP_RIpar_VarRI <- sapply(1:length(He_MDIMP_R_2r),function(B){
  round(as.numeric(VarCorr(He_MDIMP_R_2r[[B]])[1]),4)
})

He_MDIMP_RIpar_CorrRI <- sapply(1:length(He_MDIMP_R_2r),function(B){

```



```

a<-getVarCov(He_MDIMP_R_2r[[B]],type="marginal")
b<-cov2cor(a[[1]])
return(round(b[1,2],4))
})

fact <- if(mean(abs(He_MDIMP_RIpar_VarRI)) >= mean(abs(He_MDIMP_RIpar_CorrRI))){ #Es fa un càlcul del f
  10^(round(log10(abs(mean(He_MDIMP_RIpar_VarRI))/abs(mean(He_MDIMP_RIpar_CorrRI))))
} else {
  1/(10^(round(log10(abs(mean(He_MDIMP_RIpar_CorrRI))/abs(mean(He_MDIMP_RIpar_VarRI))))))
}

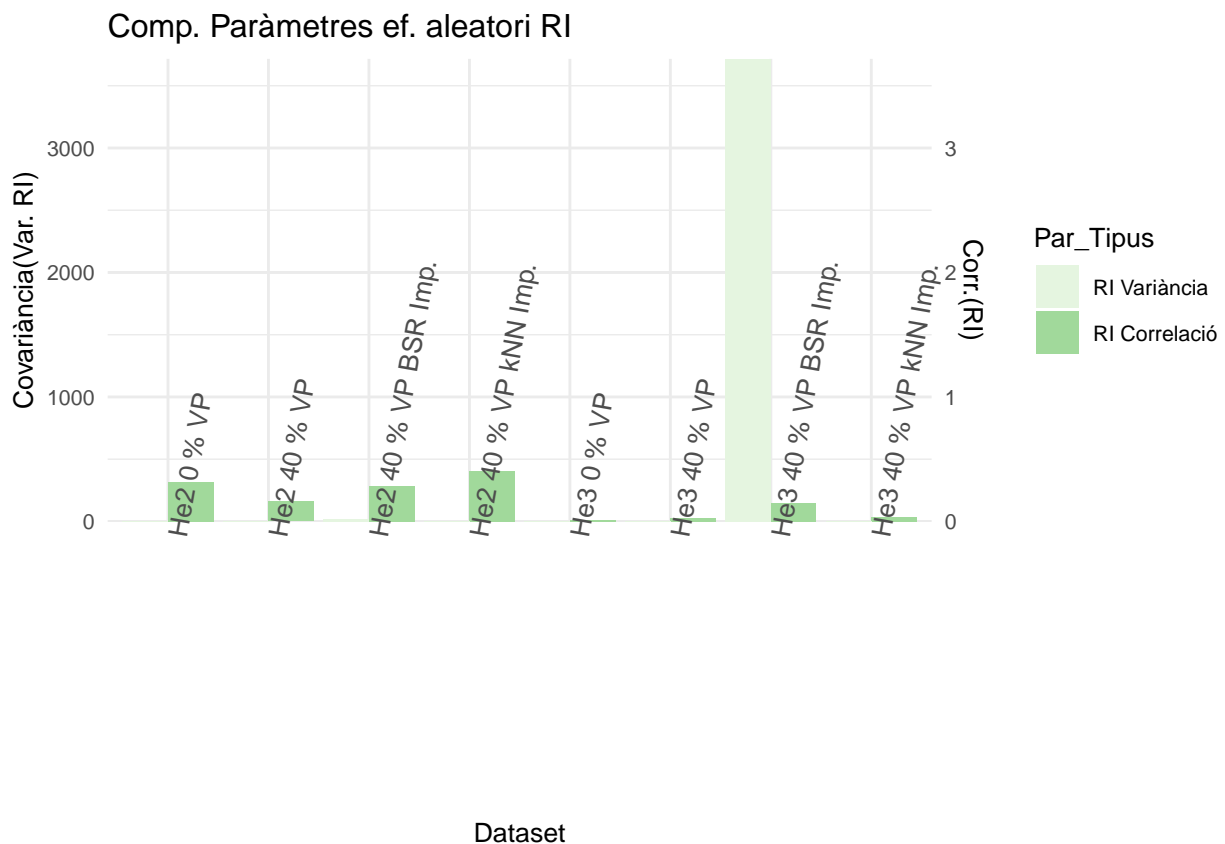
if (fact==0){
  fact <-0.0001
}

He_MDIMP_RIpar_CorrRI <- He_MDIMP_RIpar_CorrRI*fact

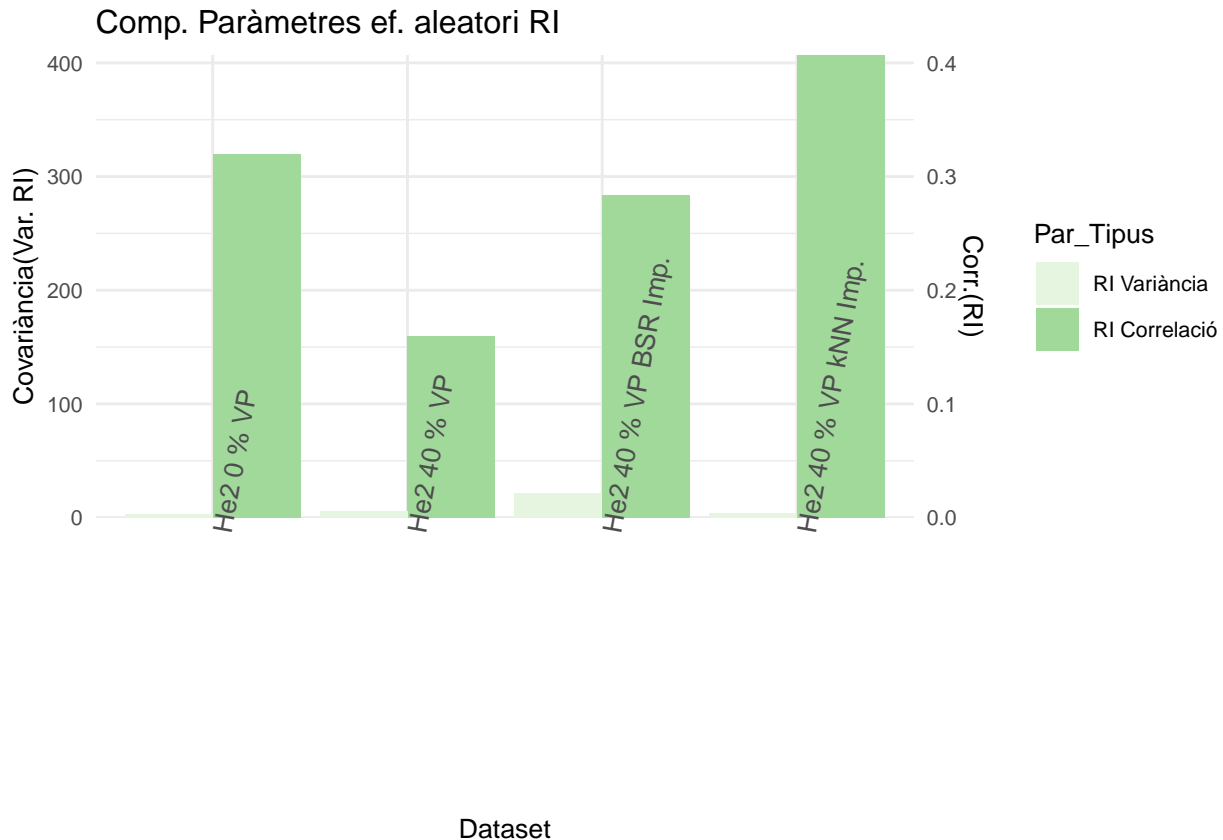
He_MDIMP_RIpar$Par <- c(He_MDIMP_RIpar_VarRI,He_MDIMP_RIpar_CorrRI)

ggplot(data=He_MDIMP_RIpar, aes(x=Dataset, y=Par, fill=Par_Tipus)) +
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Greens") + theme_minimal() +
  labs(title="Comp. Paràmetres ef. aleatori RI",y="Covariància(Var. RI)" +
  scale_y_continuous(expand = c(0, 0),sec.axis = sec_axis(~./fact, name = "Corr.(RI)")) +
  theme(text = element_text(size=10),axis.text.x = element_text(angle=80,size = 10,hjust=0))

```



```
ggplot(data=He_MDIMP_RIpar[~grep("He3", He_MDIMP_RIpar$Dataset),],
  aes(x=Dataset, y=Par, fill=Par_Tipus)) +
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Greens") + theme_minimal() +
  labs(title="Comp. Paràmetres ef. aleatori RI",y="Covariància(Var. RI)" +
  scale_y_continuous(expand = c(0, 0),sec.axis = sec_axis(~./fact, name = "Corr.(RI)")) +
  theme(text = element_text(size=10),axis.text.x = element_text(angle=80,size = 10,hjust=0))
```



Comentaris:

Dels 2 models semblaria que únicament milloren en aquests 2 paràmetres els mètodes de dades perdudes el cas de He2. Possiblement en el cas He3 es milloraria molt lleugerament el kNN, però no seria molt significatiu.

```
#Prenem la correlació entre els punts més propers com a referencia
#Generació de les llistes a mida de models per poder fer la taula de paràmetres amb el model REML
So2_MDIMP_R_2 <- ConvList(Comp_IS_RI_MDIMP_So2)
So3_MDIMP_R_2 <- ConvList(Comp_IS_RIS_MDIMP_So3)
So_MDIMP_R_2_0 <- list(So2_MDIMP_R_2,So3_MDIMP_R_2,So3_MDIMP_R_2)
So_MDIMP_R_2 <- ConvList(So_MDIMP_R_2_0)

So_MDIMP_RIpar <- data.frame(Dataset = paste(rep(c(SoVars, "So3(t)"),each=length(Comp_IS_RI_MDIMP_So2)),
  rep(names(Comp_IS_RI_MDIMP_So2), (length(SoVars)+1))))

So_MDIMP_RIpar <- rbind(So_MDIMP_RIpar,So_MDIMP_RIpar)

So_MDIMP_RIpar$Par_Tipus <- factor(rep(c("RI Variància","RI Correlació"),each=(nrow(So_MDIMP_RIpar)/2))
  levels=c("RI Variància","RI Correlació"))
```

```

So_MDIMP_RIpar_VarRI <- c(
  sapply(1:(length(So_MDIMP_R_2)-4),function(B){
    round(as.numeric(VarCorr(So_MDIMP_R_2[[B]])[1]),4)
  }),
  sapply((length(So_MDIMP_R_2)-3):length(So_MDIMP_R_2),function(B){
    round(as.numeric(VarCorr(So_MDIMP_R_2[[B]])[2]),4)
  })
)

So_MDIMP_RIpar_CorrRI <- sapply(1:length(So_MDIMP_R_2),function(B){
  a<-getVarCov(So_MDIMP_R_2[[B]],type="marginal")
  b<-cov2cor(a[[1]])
  return(round(b[1,2],4))
})

fact <- if(mean(abs(So_MDIMP_RIpar_VarRI)) >= mean(abs(So_MDIMP_RIpar_CorrRI))){ #Es fa un càlcul del f
  10^(round(log10(abs(mean(So_MDIMP_RIpar_VarRI))/abs(mean(So_MDIMP_RIpar_CorrRI))))
} else {
  1/(10^(round(log10(abs(mean(So_MDIMP_RIpar_CorrRI))/abs(mean(So_MDIMP_RIpar_VarRI))))))
}

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

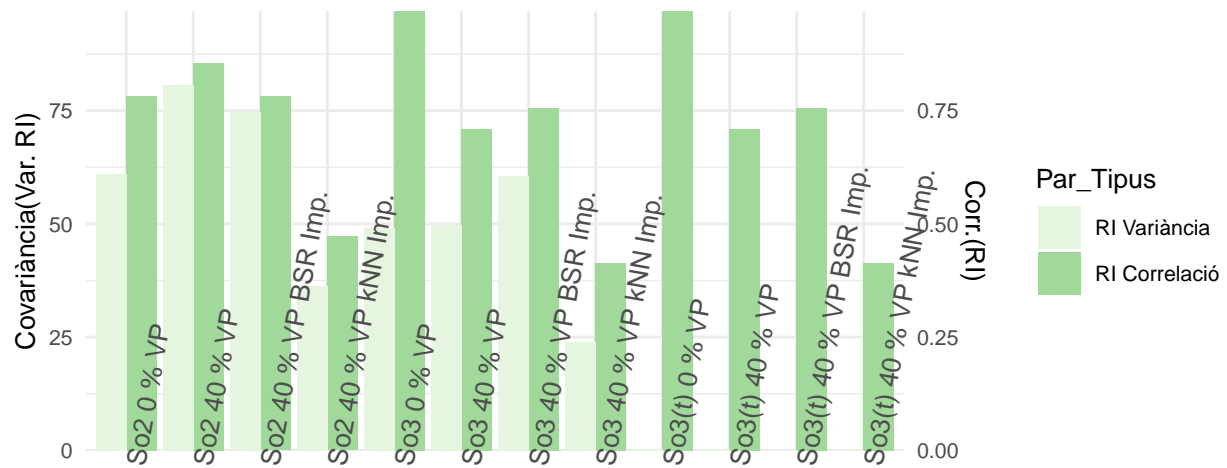
So_MDIMP_RIpar_CorrRI <- So_MDIMP_RIpar_CorrRI*fact

So_MDIMP_RIpar$Par <- c(So_MDIMP_RIpar_VarRI,So_MDIMP_RIpar_CorrRI)

ggplot(data=So_MDIMP_RIpar, aes(x=Dataset, y=Par, fill=Par_Tipus)) +
  geom_bar(stat="identity", position=position_dodge()+
  scale_fill_brewer(palette="Greens") + theme_minimal() +
  labs(title="Comp. Paràmetres ef. aleatori RI",y="Covariància(Var. RI)" ) +
  scale_y_continuous(expand = c(0, 0),sec.axis = sec_axis(~./fact, name = "Corr.(RI)") ) +
  theme(text = element_text(size=10),axis.text.x = element_text(angle=80,size = 10,hjust=0))

```

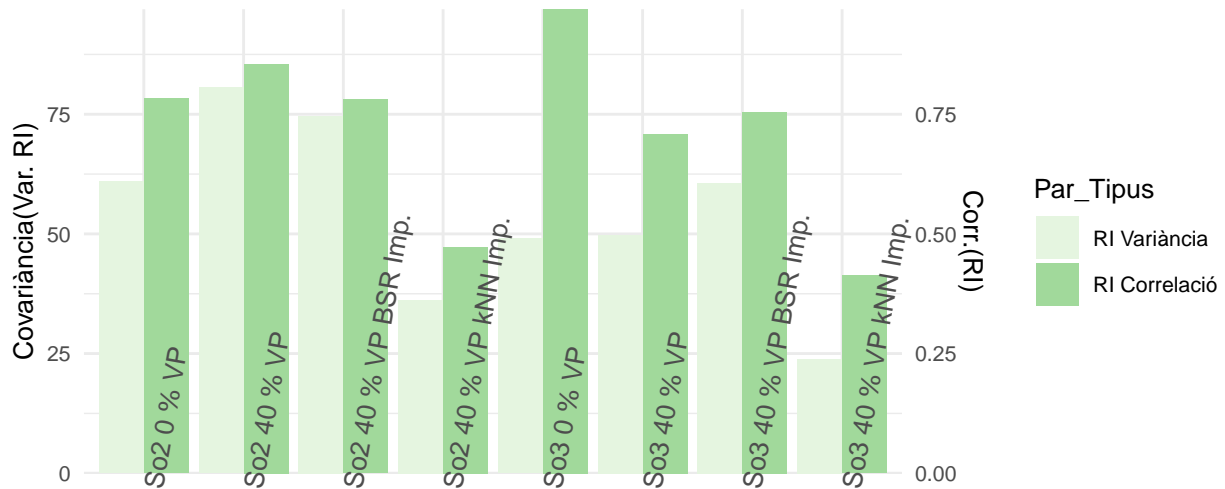
Comp. Paràmetres ef. aleatori RI



Dataset

```
ggplot(data=So_MDIMP_RIpar[-grep("t", So_MDIMP_RIpar$Dataset),]
, aes(x=Dataset, y=Par, fill=Par_Tipus)) +
  geom_bar(stat="identity", position=position_dodge())+
  scale_fill_brewer(palette="Greens") + theme_minimal() +
  labs(title="Comp. Paràmetres ef. aleatori RI",y="Covariància(Var. RI)") +
  scale_y_continuous(expand = c(0, 0),sec.axis = sec_axis(~./fact, name = "Corr.(RI)") ) +
  theme(text = element_text(size=10),axis.text.x = element_text(angle=80,size = 10,hjust=0))
```

Comp. Paràmetres ef. aleatori RI



Dataset

Dona la impressió que les millores només es produeixen en el factor de correlació al aplicar els mètodes en els 3 casos, millorant respecte al conjunt amb els valors perduts eliminats.

Intervals de Confiància de la predicció de mitges

No és senzill calcular els intervals de confiança de predicció en els models mixtes ja que la variabilitat depèn de més factors a part del factor residual a part de tenir el factor o factors aleatoris. Per duu a terme aquesta tasca dins el temps disponible per l'anàlisi estadístic es segueix les recomanacions de <http://www.remkoduursma.com/post/2017-06-15-bootpredictlme4/> i per fer-ho s'ajusten tots els models amb la funció `lmer`:

#A partir de la tècnica bootstrap amb l'estadístic t es pot deduir la possible distribució i calcular e

```
set.seed(7982)
lmer_I_RI_MDIMP_Ho2 <- lapply(1:length(Av_DS_MDIMP_list),function(p){
  lmer(Ho2 ~ 1 + (1|Lot),data=Av_DS_MDIMP_list[[p]],REML=T,na.action=na.omit)
})
pred_lmer_I_RI_MDIMP_Ho2 <- lapply(1:length(lmer_I_RI_MDIMP_Ho2),function(p){
  predict(lmer_I_RI_MDIMP_Ho2[[p]], newdata=data.frame(Lot="Lot 1"), se.fit=TRUE, nsim=100)
})

pred_lmer_Ho <- data.frame(Dataset = rep(HoVars,each=length(Av_DS_MDIMP_list)))
pred_lmer_Ho$VP <- rep(names(Av_DS_MDIMP_list),length(HoVars))
pred_lmer_Ho$fit <- c(
  sapply(1:length(pred_lmer_I_RI_MDIMP_Ho2),function(p){
    pred_lmer_I_RI_MDIMP_Ho2[[p]]$fit
  })
)
```

```

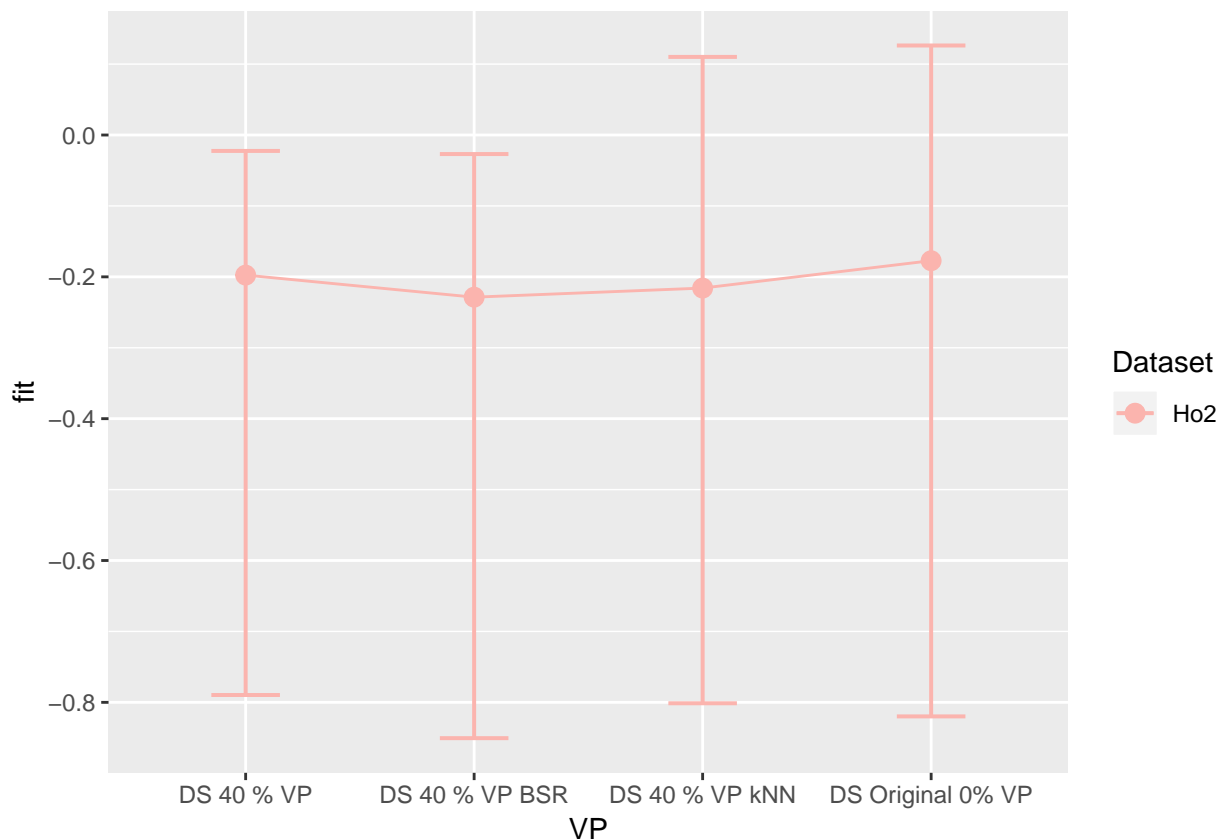
)

pred_lmer_Ho$LCI <- c(
  sapply(1:length(pred_lmer_I_RI_MDIMP_Ho2),function(p){
    pred_lmer_I_RI_MDIMP_Ho2[[p]]$ci.fit[1,]
  })
)

pred_lmer_Ho$HCI <- c(
  sapply(1:length(pred_lmer_I_RI_MDIMP_Ho2),function(p){
    pred_lmer_I_RI_MDIMP_Ho2[[p]]$ci.fit[2,]
  })
)

ggplot(pred_lmer_Ho, aes(x=VP, y=fit, group=Dataset, color=Dataset)) +
  geom_line() +
  geom_point(size=3) +
  geom_errorbar(aes(ymin=LCI, ymax=HCI), width=.3,size=.7,
               position=position_dodge(0.05)) +
  scale_color_brewer(palette="Pastel1")

```



Comentaris:

En aquest sentit sembla que el sistema kNN aconsegueix generar una predicció molt semblant al model original.

```

#A partir de la tècnica bootstrap amb l'estadístic t es pot deduir la possible distribució i calcular e
set.seed(7982)
lmer_I_RI_W_MDIMP_He2 <- lapply(1:length(Av_DS_MDIMP_list),function(p){
  lmer(He2 ~ 1 + (1|Lot),data=Av_DS_MDIMP_list[[p]],
      weights = Temps, REML=T,na.action=na.omit)
})
pred_lmer_I_RI_W_MDIMP_He2 <- lapply(1:length(lmer_I_RI_W_MDIMP_He2),function(p){
  predict(lmer_I_RI_W_MDIMP_He2[[p]], newdata=data.frame(Lot="Lot 1", Temps = 24), se.fit=TRUE, nsim=100)
})

set.seed(7982)
lmer_I_RI_W_MDIMP_He3 <- lapply(1:length(Av_DS_MDIMP_list),function(p){
  lmer(He3 ~ 1 + (1|Lot),data=Av_DS_MDIMP_list[[p]],
      weights = Temps, REML=T,na.action=na.omit)
})
pred_lmer_I_RI_W_MDIMP_He3 <- lapply(1:length(lmer_I_RI_W_MDIMP_He3),function(p){
  predict(lmer_I_RI_W_MDIMP_He3[[p]], newdata=data.frame(Lot="Lot 1", Temps = 24), se.fit=TRUE, nsim=100)
})

pred_lmer_He <- data.frame(Dataset = rep(HeVars,each=length(Av_DS_MDIMP_list)))
pred_lmer_He$VP <- rep(names(Av_DS_MDIMP_list),length(HeVars))
pred_lmer_He$fit <- c(
  sapply(1:length(pred_lmer_I_RI_W_MDIMP_He2),function(p){
    pred_lmer_I_RI_W_MDIMP_He2[[p]]$fit
  }),
  sapply(1:length(pred_lmer_I_RI_W_MDIMP_He3),function(p){
    pred_lmer_I_RI_W_MDIMP_He3[[p]]$fit
  })
)

pred_lmer_He$LCI <- c(
  sapply(1:length(pred_lmer_I_RI_W_MDIMP_He2),function(p){
    pred_lmer_I_RI_W_MDIMP_He2[[p]]$ci.fit[1,]
  }),
  sapply(1:length(pred_lmer_I_RI_W_MDIMP_He3),function(p){
    pred_lmer_I_RI_W_MDIMP_He3[[p]]$ci.fit[1,]
  })
)

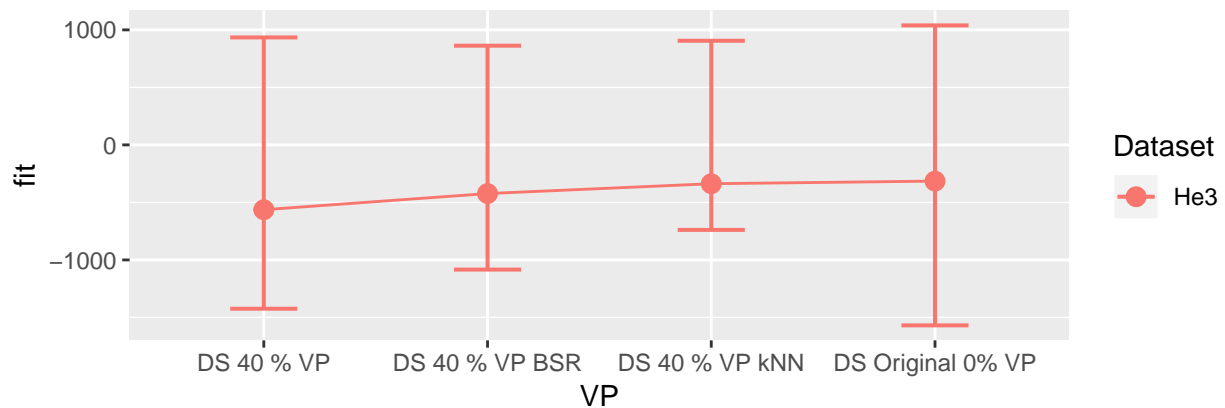
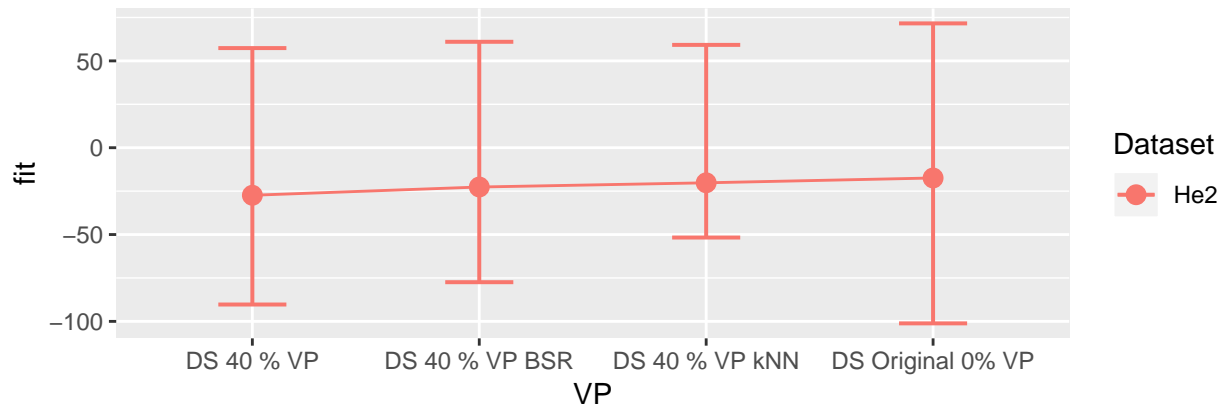
pred_lmer_He$HCI <- c(
  sapply(1:length(pred_lmer_I_RI_W_MDIMP_He2),function(p){
    pred_lmer_I_RI_W_MDIMP_He2[[p]]$ci.fit[2,]
  }),
  sapply(1:length(pred_lmer_I_RI_W_MDIMP_He3),function(p){
    pred_lmer_I_RI_W_MDIMP_He3[[p]]$ci.fit[2,]
  })
)

p2<-ggplot(pred_lmer_He[pred_lmer_He$Dataset=="He2",], aes(x=VP, y=fit, group=Dataset, color=Dataset)) +
  geom_line() +
  geom_point(size=3)+
  geom_errorbar(aes(ymin=LCI, ymax=HCI), width=.3,size=.7,
               position=position_dodge(0.05))

```

```
p3<-ggplot(pred_lmer_He[pred_lmer_He$Dataset=="He3",], aes(x=VP, y=fit, group=Dataset, color=Dataset)) +
  geom_line() +
  geom_point(size=3)+
  geom_errorbar(aes(ymin=LCI, ymax=HCI), width=.3,size=.7,
                position=position_dodge(0.05))
```

```
grid.arrange(p2,p3)
```



Comentaris:

Tot i no haver molta diferència amb el conjunt amb eliminació de valors perduts, possiblement en aquest aspecte el mètode BSR ha estat el mètode més proper al dataset original.

```
#A partir de la tècnica bootstrap amb l'estadístic t es pot deduir la possible distribució i calcular e
set.seed(7982)
lmer_IS_RI_MDIMP_So2 <- lapply(1:length(Av_DS_MDIMP_list),function(p){
  lmer(So2 ~ 1 + Temps + (1|Lot),data=Av_DS_MDIMP_list[[p]],
      REML=T,na.action=na.omit)
})
pred_lmer_IS_RI_MDIMP_So2 <- lapply(1:length(lmer_IS_RI_MDIMP_So2),function(p){
  predict(lmer_IS_RI_MDIMP_So2[[p]], newdata=data.frame(Lot="Lot 1", Temps = 24), se.fit=TRUE, nsim=100)
})

set.seed(7982)
lmer_IS_RIS_MDIMP_So3 <- lapply(1:length(Av_DS_MDIMP_list),function(p){
  lmer(So3 ~ 1 + Temps + (1+Temps|Lot),data=Av_DS_MDIMP_list[[p]],
      REML=T,na.action=na.omit,control = lmerControl(optimizer = "Nelder_Mead"))
```



```

})
pred_lmer_IS_RIS_MDIMP_So3 <- lapply(1:length(lmer_IS_RIS_MDIMP_So3),function(p){
  predict(lmer_IS_RIS_MDIMP_So3[[p]], newdata=data.frame(Lot="Lot 1", Temps = 24), se.fit=TRUE, nsim=1000)
})

pred_lmer_So <- data.frame(Dataset = rep(SoVars,each=length(Av_DS_MDIMP_list)))
pred_lmer_So$VP <- rep(names(Av_DS_MDIMP_list),length(SoVars))
pred_lmer_So$fit <- c(
  sapply(1:length(pred_lmer_IS_RI_MDIMP_So2),function(p){
    pred_lmer_IS_RI_MDIMP_So2[[p]]$fit
  }),
  sapply(1:length(pred_lmer_IS_RIS_MDIMP_So3),function(p){
    pred_lmer_IS_RIS_MDIMP_So3[[p]]$fit
  })
)

pred_lmer_So$LCI <- c(
  sapply(1:length(pred_lmer_IS_RI_MDIMP_So2),function(p){
    pred_lmer_IS_RI_MDIMP_So2[[p]]$ci.fit[1,]
  }),
  sapply(1:length(pred_lmer_IS_RIS_MDIMP_So3),function(p){
    pred_lmer_IS_RIS_MDIMP_So3[[p]]$ci.fit[1,]
  })
)

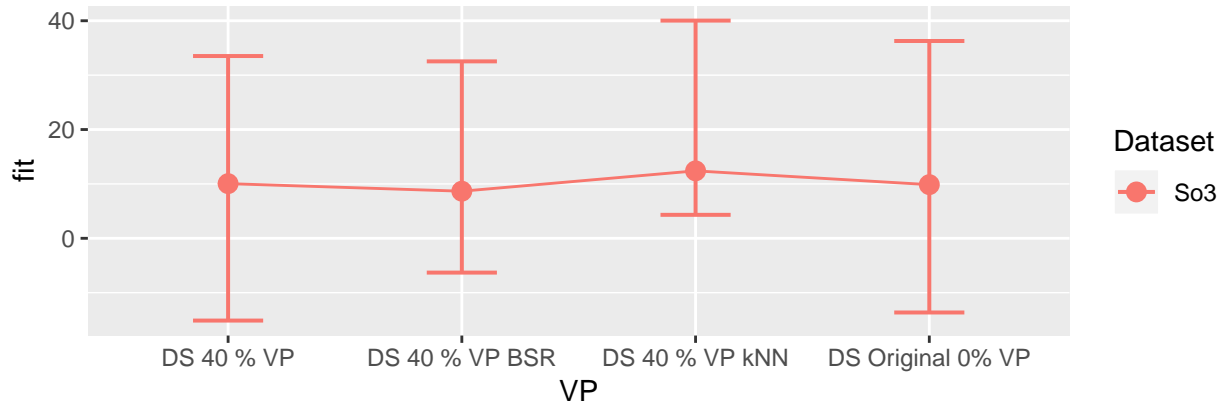
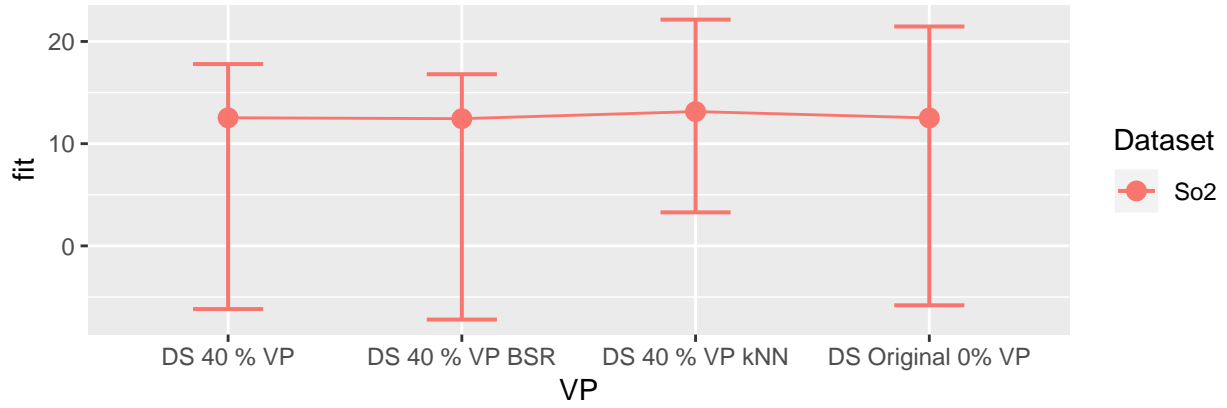
pred_lmer_So$HCI <- c(
  sapply(1:length(pred_lmer_IS_RI_MDIMP_So2),function(p){
    pred_lmer_IS_RI_MDIMP_So2[[p]]$ci.fit[2,]
  }),
  sapply(1:length(pred_lmer_IS_RIS_MDIMP_So3),function(p){
    pred_lmer_IS_RIS_MDIMP_So3[[p]]$ci.fit[2,]
  })
)

p2<-ggplot(pred_lmer_So[pred_lmer_So$Dataset=="So2",], aes(x=VP, y=fit, group=Dataset, color=Dataset)) +
  geom_line() +
  geom_point(size=3)+
  geom_errorbar(aes(ymin=LCI, ymax=HCI), width=.3,size=.7,
    position=position_dodge(0.05))

p3<-ggplot(pred_lmer_So[pred_lmer_So$Dataset=="So3",], aes(x=VP, y=fit, group=Dataset, color=Dataset)) +
  geom_line() +
  geom_point(size=3)+
  geom_errorbar(aes(ymin=LCI, ymax=HCI), width=.3,size=.7,
    position=position_dodge(0.05))

grid.arrange(p2,p3)

```



Realment no es pot apreciar una millora significativa en aquest aspecte pels dos conjunts corregits en front el que ha eliminat els valors perduts, inclús potser estan més allunyats que el ocnjunt original.