

# TFM\_ANNEX-3\_CAS\_PRÀCTIC\_LAKE

Francesc Bernad Martin

May 31, 2019

## Contents

<b>ANÀLISI DESCRIPTIU DATASET</b>	<b>2</b>
Importació Dataset . . . . .	2
Estructura y selecció . . . . .	3
Anàlisi descriptiu Dataset . . . . .	8
<b>AJUST MODEL SIMPLE</b>	<b>20</b>
Model mínims quadrats <i>Stepwise</i> . . . . .	20
Diagnòstic model simple . . . . .	23
Normalitat / Gràfics Residuals . . . . .	23
Homoscedasticitat . . . . .	33
Diagnòstic possibles outliers . . . . .	41
Colinealitat . . . . .	55
<b>COMPROVACIÓ INTERACCIONS MODEL</b>	<b>59</b>
Paralelisme o Interacció Individu-Temps . . . . .	59
Altres Interaccions . . . . .	61
Diagnòstics models interaccions . . . . .	66
<b>AJUST MODEL EFECTES ALEATORIS</b>	<b>80</b>
Intercepció Aleatòria . . . . .	80
Ajust models . . . . .	80
Diagnòstic models . . . . .	83
Pendent Aleatòria . . . . .	97
Ajust models . . . . .	97
Diagnòstic models . . . . .	102
<b>AJUST FUNCIONS DE COVARIÀNCIA</b>	<b>138</b>
Ajust models . . . . .	138
Diagnòstic models . . . . .	142
Optimització . . . . .	180

```
usePackage <- function(p) {
  if (!is.element(p, installed.packages()[,1]))
    install.packages(p, dep = TRUE)
  require(p, character.only = TRUE)
}
usePackage("nlme")
usePackage("nlmeU")
usePackage("lme4")
usePackage("ggplot2")
usePackage("gridExtra")
usePackage("ggExtra")
usePackage("kableExtra")
usePackage("RLRsim")
usePackage("mixlm")
usePackage("magrittr")
```

```

usePackage("evaluate")
usePackage("haven")
usePackage("summarytools")
usePackage("psych")
usePackage("stringr")
usePackage("randomcolor")
usePackage("MASS")
usePackage("stats")
usePackage("lmtest")
usePackage("faraway")
usePackage("piecewiseSEM")

```

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

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

*#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)
}

```

## ANÀLISI DESCRIPTIU DATASET

### Importació Dataset

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

```

workingDir <-getwd()
setwd(workingDir)
LAKEO <- read_sav(paste(workingDir,"/BdD Clinical Trial/LAKE/lake.sav",sep=""))
head(LAKEO[,1:10])

```

```

## # A tibble: 6 x 10
##   proc nusuario npac  nvisita fecha_nac      sexo factorriesgo_AD~
##   <dbl+lbl> <chr>   <chr>   <dbl> <date>         <dbl+lbl>         <dbl+lbl>
## 1 NA      aarco    002     NA NA           NA                NA
## 2 1 [Dem~ amariño 003     1 1956-10-14 1 [Hom~           1 [Si~
## 3 1 [Dem~ amariño 004     1 1966-11-01 1 [Hom~           0 [No~
## 4 1 [Dem~ aocampo 002     1 1976-02-11 1 [Hom~           0 [No~
## 5 1 [Dem~ aocampo 003     1 1972-04-24 1 [Hom~           0 [No~
## 6 1 [Dem~ aocampo 004     1 1968-07-14 1 [Hom~           0 [No~
## # ... with 3 more variables: especificar <chr>, a19 <dbl>,

```

```
## # estadio_VIH_20 <dbl+lbl>
```

## Estructura y selecció

Es mostra el tipus de variables que formen el model:

```
dim(LAKE0)
```

```
## [1] 116 219
```

```
table(sapply(LAKE0,class))
```

```
##
##      character      Date haven_labelled      numeric
##           3           8           30           178
```

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

```
##      nusuario      npac especificar
##           2           3           8
```

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

```
##      fecha_nac fecha_ini_lake      fecha_vih      Fecha_0      Fecha_12
##           5           11           19           24           61
##      Fecha_24      Fecha_36      Fecha_48
##           98           138           175
```

La quantitat d'individus inicial és gran i suficient per analitzar de manera correcta semblaria (116 individus). HI ha també una quantitat enorme de variables (219). És d'interès pel model que es vol testar modelitzar amb poques variables d'inici ja que l'objectiu no és l'anàlisi multivariable en sí.

Respecte a la identificació del tipus de variables hi ha:

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

Si s'observa quines són les variables numèriques:

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

```
##      nvisita      a19      a28
##           4           9           16
##      estadio_VIH_31      a32      Estado
##           17           18           20
##      edad      week_0      CargaViral_0
```

##	21	23	25
##	CD4A_0	CD4P_0	CD8A_0
##	26	27	28
##	CD8P_0	Hematocrito_0	Hemoglobina_0
##	29	30	31
##	Plaquetas_0	Leucocitos_0	LinfosTotales_0
##	32	33	34
##	Glucosa_mg_0	Urea_mg_0	Creatinina_mumol_0
##	35	36	37
##	Sodio_0	Potasio_0	Cloro_0
##	38	39	40
##	Calcio_0	Bilirrubina_mumol_0	GPT_0
##	41	42	43
##	GOT_0	GGT_0	ProteinasTotales_0
##	44	45	46
##	Albumina_0	Colesterol_mg_0	LDL_mg_0
##	47	48	49
##	HDL_mg_0	Trigliceridos_mg_0	Amilasa_0
##	50	51	52
##	pH_0	Bicarbonato_0	AcidoLactico_0
##	53	54	55
##	AcidoPiruvico_0	week_12	CargaViral_12
##	56	60	62
##	CD4A_12	CD4P_12	CD8A_12
##	63	64	65
##	CD8P_12	Hematocrito_12	Hemoglobina_12
##	66	67	68
##	Plaquetas_12	Leucocitos_12	LinfosTotales_12
##	69	70	71
##	Glucosa_mg_12	Urea_mg_12	Creatinina_mumol_12
##	72	73	74
##	Sodio_12	Potasio_12	Cloro_12
##	75	76	77
##	Calcio_12	Bilirrubina_mumol_12	GPT_12
##	78	79	80
##	GOT_12	GGT_12	ProteinasTotales_12
##	81	82	83
##	Albumina_12	Colesterol_mg_12	LDL_mg_12
##	84	85	86
##	HDL_mg_12	Trigliceridos_mg_12	Amilasa_12
##	87	88	89
##	pH_12	Bicarbonato_12	AcidoLactico_12
##	90	91	92
##	AcidoPiruvico_12	week_24	CargaViral_24
##	93	97	99
##	CD4A_24	CD4P_24	CD8A_24
##	100	101	102
##	CD8P_24	Hematocrito_24	Hemoglobina_24
##	103	104	105
##	Plaquetas_24	Leucocitos_24	LinfosTotales_24
##	106	107	108
##	Glucosa_mg_24	Urea_mg_24	Creatinina_mumol_24
##	109	110	111
##	Sodio_24	Potasio_24	Cloro_24

##	112	113	114
##	Calcio_24	Bilirrubina_mumol_24	GPT_24
##	115	116	117
##	GOT_24	GGT_24	ProteinasTotales_24
##	118	119	120
##	Albumina_24	Colesterol_mg_24	LDL_mg_24
##	121	122	123
##	HDL_mg_24	Trigliceridos_mg_24	Amilasa_24
##	124	125	126
##	pH_24	Bicarbonato_24	AcidoLactico_24
##	127	128	129
##	AcidoPiruvico_24	week_36	CargaViral_36
##	130	137	139
##	CD4A_36	CD4P_36	CD8A_36
##	140	141	142
##	CD8P_36	Hematocrito_36	Hemoglobina_36
##	143	144	145
##	Plaquetas_36	Leucocitos_36	LinfosTotales_36
##	146	147	148
##	Glucosa_mg_36	Urea_mg_36	Creatinina_mumol_36
##	149	150	151
##	Sodio_36	Potasio_36	Cloro_36
##	152	153	154
##	Calcio_36	Bilirrubina_mumol_36	GPT_36
##	155	156	157
##	GOT_36	GGT_36	ProteinasTotales_36
##	158	159	160
##	Albumina_36	Colesterol_mg_36	LDL_mg_36
##	161	162	163
##	HDL_mg_36	Trigliceridos_mg_36	Amilasa_36
##	164	165	166
##	pH_36	Bicarbonato_36	AcidoLactico_36
##	167	168	169
##	AcidoPiruvico_36	week_48	CargaViral_48
##	170	174	176
##	CD4A_48	CD4P_48	CD8A_48
##	177	178	179
##	CD8P_48	Hematocrito_48	Hemoglobina_48
##	180	181	182
##	Plaquetas_48	Leucocitos_48	LinfosTotales_48
##	183	184	185
##	Glucosa_mg_48	Urea_mg_48	Creatinina_mumol_48
##	186	187	188
##	Sodio_48	Potasio_48	Cloro_48
##	189	190	191
##	Calcio_48	Bilirrubina_mumol_48	GPT_48
##	192	193	194
##	GOT_48	GGT_48	ProteinasTotales_48
##	195	196	197
##	Albumina_48	Colesterol_mg_48	LDL_mg_48
##	198	199	200
##	HDL_mg_48	Trigliceridos_mg_48	Amilasa_48
##	201	202	203
##	pH_48	Bicarbonato_48	AcidoLactico_48

```
##           204           205           206
##   AcidoPiruvico_48   tpo_vih_meses   diff_cd4_48_0
##           207           213           215
##   diff_cd4p_48_0   diff_col_48_0   diff_HDL_48_0
##           216           217           218
##   diff_LDL_48_0
##           219
```

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

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

```
LAKE0a<- LAKE0[,which(sapply(LAKE0,class)!="haven_labelled" & sapply(LAKE0,class)!="Date")]
LAKE0a$Grupo <- LAKE0$Grupo
LAKE0a$sexo <- LAKE0$sexo
#De les variables numèriques es deixa només amb edad i grup com a idenfificadors i tractament, i amb le

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

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

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

#Var depenents del temps
LAKE1b0 <- LAKE0a[,which(sapply(LAKE0a,class)=="numeric")]
LAKE1b0 <- LAKE1b0[,-grep("diff_",colnames(LAKE1b0))]#S'elimina les variables diff_ que poden fer conf

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

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

## [1] "CargaViral" "Albumina" "CD4P" "Cloro"
```

```

nvar_sel <- length(var_names_sel)

LAKE1b0_sel <- LAKE1b0[,c(paste(var_names_sel,rep(vart_string,each=length(var_names_sel)),sep=""))]
LAKE1ab0_prov <- cbind(LAKE1a0,LAKE1b0_sel)
LAKE1ab0_prov_noNA <- na.omit(LAKE1ab0_prov, cols=((ncol(LAKE1a0)+1):ncol(LAKE1ab0_prov_noNA))) #només

LAKE1a0_noNA <- LAKE1ab0_prov_noNA[,1:ncol(LAKE1a0)]
LAKE1b0_noNA <- LAKE1ab0_prov_noNA[, (ncol(LAKE1a0)+1):ncol(LAKE1ab0_prov_noNA)]

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

reorgLAKE1b0_noNA <- reorg(LAKE1b0_noNA,nvar_sel,nvart_string)

LAKE1b_noNA <- reorgLAKE1b0_noNA
colnames(LAKE1b_noNA) <- var_names_sel

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

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

```

```

##      nusuario npac edad   Grupo  sexo      Ident Tiempo CargaViral
## 4      aocampo  002   29 Treatm.1 Hombre aocampo 002      0      48400
## 4.1    aocampo  002   29 Treatm.1 Hombre aocampo 002     12       40
## 4.2    aocampo  002   29 Treatm.1 Hombre aocampo 002     24       40
## 4.3    aocampo  002   29 Treatm.1 Hombre aocampo 002     36       40
## 4.4    aocampo  002   29 Treatm.1 Hombre aocampo 002     48       40
## 5      aocampo  003   33 Treatm.1 Hombre aocampo 003      0     190000
##      Albumina CD4P Cloro
## 4          4.39 12.6  107
## 4.1        4.54 15.0  103
## 4.2        4.51 26.0  104
## 4.3        4.67 25.0  104
## 4.4        4.59 26.0  108
## 5          4.09 19.0  105

```

```
str(LAKE1_noNA)
```

```
## 'data.frame': 40 obs. of 11 variables:
## $ nusuari : Factor w/ 2 levels "aocampo","gcarosi": 1 1 1 1 1 1 1 1 1 1 ...
## $ npac : Factor w/ 7 levels "001","002","003",...: 2 2 2 2 2 3 3 3 3 3 ...
## $ edad : num 29 29 29 29 29 33 33 33 33 33 ...
## $ Grupo : Factor w/ 2 levels "Treatment.1","Treatment.2": 1 1 1 1 1 1 1 1 1 1 ...
## $ sexo : Factor w/ 2 levels "Hombre","Mujer": 1 1 1 1 1 1 1 1 1 1 ...
## $ Ident : Factor w/ 8 levels "aocampo 001",...: 2 2 2 2 2 3 3 3 3 3 ...
## $ Tiempo : num 0 12 24 36 48 0 12 24 36 48 ...
## $ CargaViral: num 48400 40 40 40 40 190000 87 40 47 40 ...
## $ Albumina : num 4.39 4.54 4.51 4.67 4.59 4.09 4.06 4.08 3.94 4.27 ...
## $ CD4P : num 12.6 15 26 25 26 19 22 24 29 24 ...
## $ Cloro : num 107 103 104 104 108 105 104 103 106 103 ...
```

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

Al depurar el *dataset* per acomodar-lo als propòsits d'aquesta part pràctica s'ha quedat en un conjunt de dades amb 8 subjectes i 9 variables.

## Anàlisi descriptiu Dataset

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

```
summary(LAKE1)
```

```
##      edad      Grupo      sexo      Ident      Tiempo
## Min.   :29.00   Treatm.1:20   Hombre:35   aocampo 001: 5   Min.    : 0
## 1st Qu.:32.00   Treatm.2:20   Mujer : 5   aocampo 002: 5   1st Qu.:12
## Median :34.00                                     aocampo 003: 5   Median  :24
## Mean   :35.12                                     aocampo 004: 5   Mean    :24
## 3rd Qu.:37.50                                     gcarosi 003: 5   3rd Qu.:36
## Max.   :42.00                                     gcarosi 006: 5   Max.    :48
##                                           (Other)   :10
##      CargaViral      Albumina      CD4P      Cloro
## Min.   : 40.0   Min.   :3.100   Min.   : 3.90   Min.   : 96.0
## 1st Qu.: 40.0   1st Qu.:4.075   1st Qu.:17.00   1st Qu.:101.0
## Median : 50.0   Median :4.400   Median :23.30   Median :104.0
## Mean   :38735.7   Mean :4.302   Mean :22.11   Mean :103.1
## 3rd Qu.: 194.2   3rd Qu.:4.598   3rd Qu.:26.00   3rd Qu.:105.0
## Max.   :500000.0   Max.   :5.060   Max.   :35.00   Max.   :108.0
##
```

```
by(LAKE1, LAKE1$Grupo, summary)
```

```
## LAKE1$Grupo: Treatm.1
##      edad      Grupo      sexo      Ident      Tiempo
## Min.   :29.0   Treatm.1:20   Hombre:20   aocampo 002:5   Min.    : 0
## 1st Qu.:32.0   Treatm.2: 0   Mujer : 0   aocampo 003:5   1st Qu.:12
## Median :34.5                                     aocampo 004:5   Median  :24
## Mean   :35.0                                     gcarosi 007:5   Mean    :24
## 3rd Qu.:37.5                                     aocampo 001:0   3rd Qu.:36
## Max.   :42.0                                     gcarosi 003:0   Max.    :48
```



```

##                               (Other)   :0
##   CargaViral      Albumina      CD4P      Cloro
##   Min.   : 40.0   Min.   :3.600   Min.   : 3.90   Min.   : 96
##   1st Qu.: 40.0   1st Qu.:4.075   1st Qu.:17.00   1st Qu.:103
##   Median : 43.5   Median :4.390   Median :22.15   Median :104
##   Mean   : 37051.2   Mean   :4.327   Mean   :20.77   Mean   :104
##   3rd Qu.: 136.5   3rd Qu.:4.560   3rd Qu.:24.32   3rd Qu.:106
##   Max.   :500000.0   Max.   :4.890   Max.   :29.00   Max.   :108
##
## -----

```

```
## LAKE1$Grupo: Treatm.2
```

```

##   edad      Grupo      sexo      Ident      Tiempo
##   Min.   :32.00   Treatm.1: 0   Hombre:15   aocampo 001:5   Min.   : 0
##   1st Qu.:32.00   Treatm.2:20   Mujer : 5   gcarosi 003:5   1st Qu.:12
##   Median :33.50                                     gcarosi 006:5   Median :24
##   Mean   :35.25                                     gcarosi 010:5   Mean   :24
##   3rd Qu.:36.75                                     aocampo 002:0   3rd Qu.:36
##   Max.   :42.00                                     aocampo 003:0   Max.   :48
##
##                               (Other)   :0

```

```

##   CargaViral      Albumina      CD4P      Cloro
##   Min.   : 40.0   Min.   :3.100   Min.   : 8.40   Min.   : 98.0
##   1st Qu.: 50.0   1st Qu.:4.067   1st Qu.:18.70   1st Qu.:100.0
##   Median : 50.0   Median :4.470   Median :23.90   Median :102.0
##   Mean   : 40420.2   Mean   :4.277   Mean   :23.45   Mean   :102.2
##   3rd Qu.: 194.2   3rd Qu.:4.707   3rd Qu.:28.75   3rd Qu.:104.0
##   Max.   :303409.0   Max.   :5.060   Max.   :35.00   Max.   :107.0
##
##

```

```
by(LAKE1,LAKE1$sexo,summary)
```

```
## LAKE1$sexo: Hombre
```

```

##   edad      Grupo      sexo      Ident      Tiempo
##   Min.   :29.00   Treatm.1:20   Hombre:35   aocampo 002:5   Min.   : 0
##   1st Qu.:32.00   Treatm.2:15   Mujer : 0   aocampo 003:5   1st Qu.:12
##   Median :35.00                                     aocampo 004:5   Median :24
##   Mean   :35.57                                     gcarosi 003:5   Mean   :24
##   3rd Qu.:42.00                                     gcarosi 006:5   3rd Qu.:36
##   Max.   :42.00                                     gcarosi 007:5   Max.   :48
##
##                               (Other)   :5

```

```

##   CargaViral      Albumina      CD4P      Cloro
##   Min.   : 40   Min.   :3.100   Min.   : 3.90   Min.   : 96.0
##   1st Qu.: 40   1st Qu.:4.035   1st Qu.:16.50   1st Qu.:100.5
##   Median : 50   Median :4.400   Median :22.30   Median :104.0
##   Mean   : 41060   Mean   :4.303   Mean   :21.01   Mean   :103.1
##   3rd Qu.: 195   3rd Qu.:4.645   3rd Qu.:25.20   3rd Qu.:105.0
##   Max.   :500000   Max.   :5.060   Max.   :31.60   Max.   :108.0
##
## -----

```

```
## LAKE1$sexo: Mujer
```

```

##   edad      Grupo      sexo      Ident      Tiempo
##   Min.   :32   Treatm.1:0   Hombre:0   aocampo 001:5   Min.   : 0
##   1st Qu.:32   Treatm.2:5   Mujer :5   aocampo 002:0   1st Qu.:12
##   Median :32                                     aocampo 003:0   Median :24
##   Mean   :32                                     aocampo 004:0   Mean   :24
##   3rd Qu.:32                                     gcarosi 003:0   3rd Qu.:36

```

```
## Max.      :32                                gcarosi 006:0 Max.      :48
##                                                (Other)   :0
##   CargaViral      Albumina      CD4P      Cloro
## Min.      : 40   Min.      :4.16   Min.      :23.0   Min.      :102.0
## 1st Qu.: 40   1st Qu.:4.19   1st Qu.:26.0   1st Qu.:102.0
## Median : 84   Median :4.20   Median :31.0   Median :104.0
## Mean   : 22467 Mean   :4.30   Mean   :29.8   Mean   :103.4
## 3rd Qu.: 170   3rd Qu.:4.46   3rd Qu.:34.0   3rd Qu.:104.0
## Max.   :112000 Max.   :4.49   Max.   :35.0   Max.   :105.0
##
```

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

```
##
## Hombre  Mujer
##      20      0
```

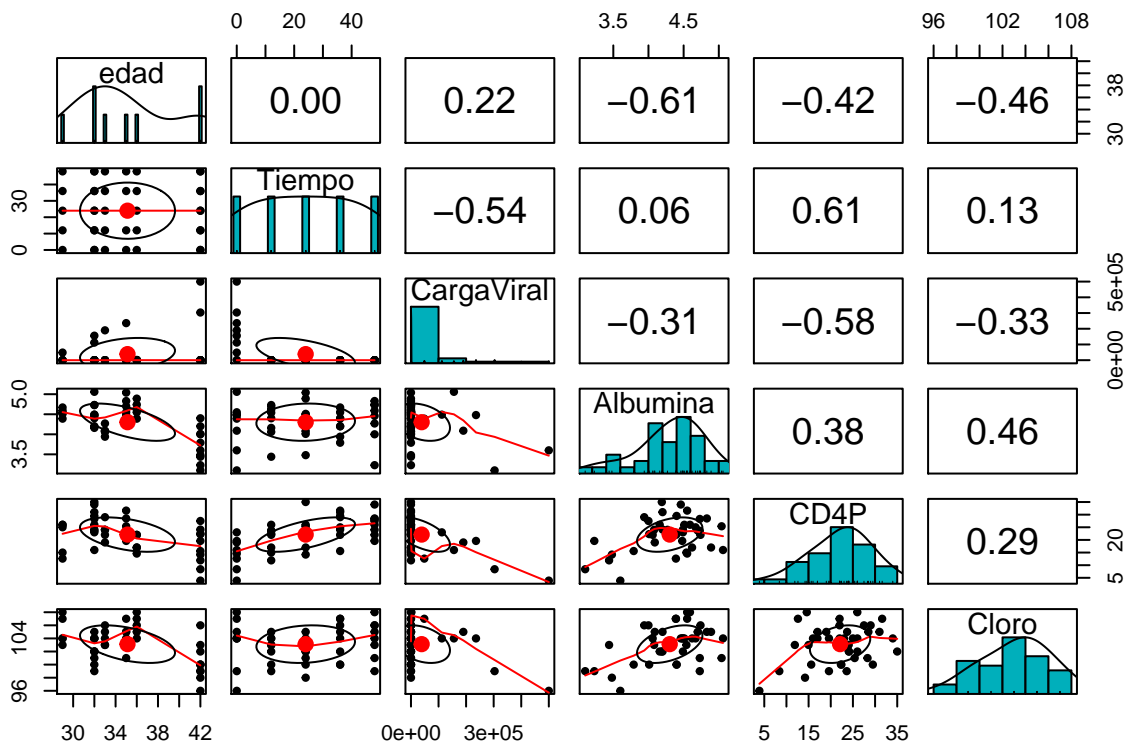
```
table(subset(LAKE1, LAKE1$Grupo=="Treatm.2")$sexo)
```

```
##
## Hombre  Mujer
##      15      5
```

Excepte la variable de càrrega viral que sembla moure's d'una manera diferent per cada individu, la resta de variables numèriques no mostra en aquest punt una diferència que es percebi amb claredat en el resum de dades. Sí que es veu en aquest cas que la proporció de sexes és 100 %, 0 % (Homes/Dones) pel tractament 1 i 75 %, 25 % (Homes/Dones) pel cas del tractament 2 (en cas que no hi hagi repartiment equitatiu pot ser un problema per analitzar el significat real del factor sexe).

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

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

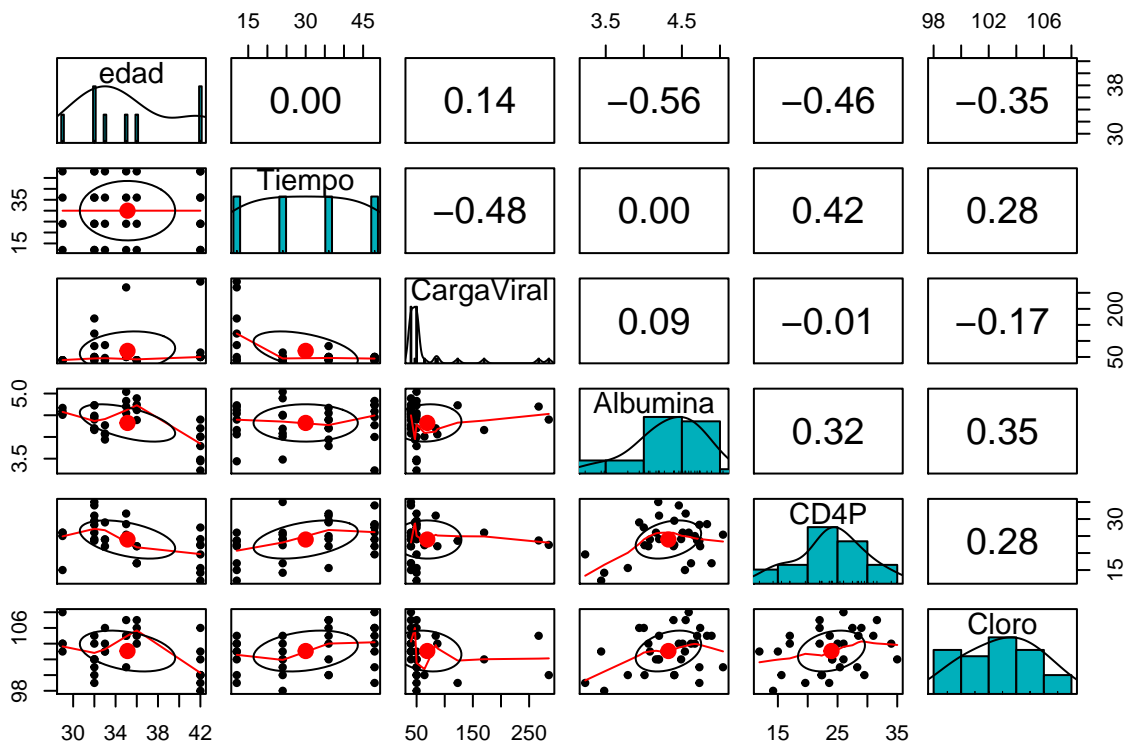


```
LAKE1cor <- as.data.frame(cor(LAKE1[,sapply(LAKE1,class)== "numeric"])))
LAKE1cor2 <- LAKE1cor[which(LAKE1cor$CargaViral!=1),]
LAKE1maxpred <- rownames(LAKE1cor2[which.max(abs(LAKE1cor2$CargaViral)),])
```

D'aquest gràfic i càlculs interessa saber quines són les variables que tenen una correlació més alta amb la resposta de la càrrega viral que en aquest cas en resulta una correlació amb les variables *edad*, *Tiempo*, *Albumina*, *CD4P*, *Cloro* de 0.22, -0.54, -0.31, -0.58, -0.33 respectivament, prenent com a predictor principal la variable de correlació més alta en valor absolut: *CD4P*.

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

```
LAKE1r <- subset(LAKE1,LAKE1$Tiempo!=0)
pairs.panels(LAKE1r[,sapply(LAKE1r,class)== "numeric"],
  method = "pearson", # correlation method
  hist.col = "#00AFBB",
  density = TRUE, # show density plots
  ellipses = TRUE # show correlation ellipses
)
```



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

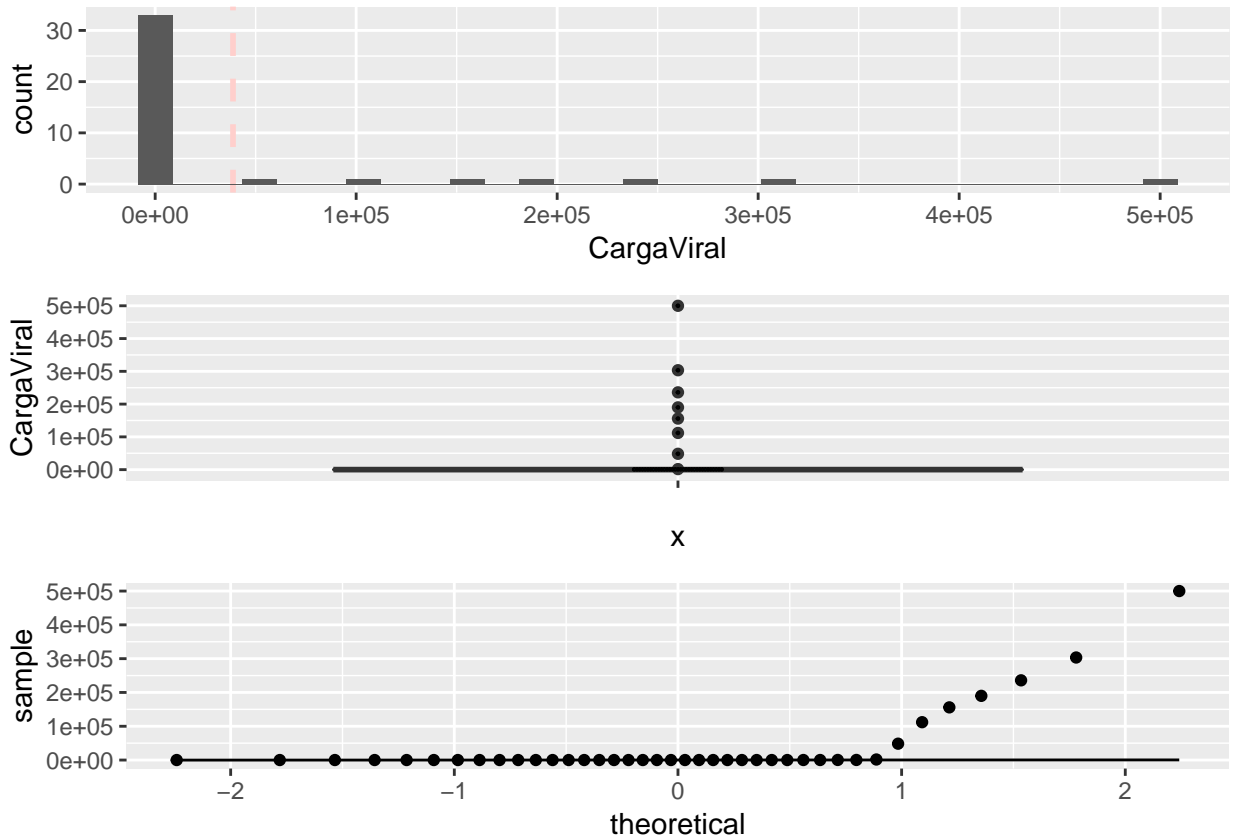
```
p01<-ggplot(LAKE1, aes(x=CargaViral)) +
  geom_vline(aes(xintercept=mean(LAKE1$CargaViral)),
    color=randomColor(), linetype="dashed", size=1) +
  geom_histogram()

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

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

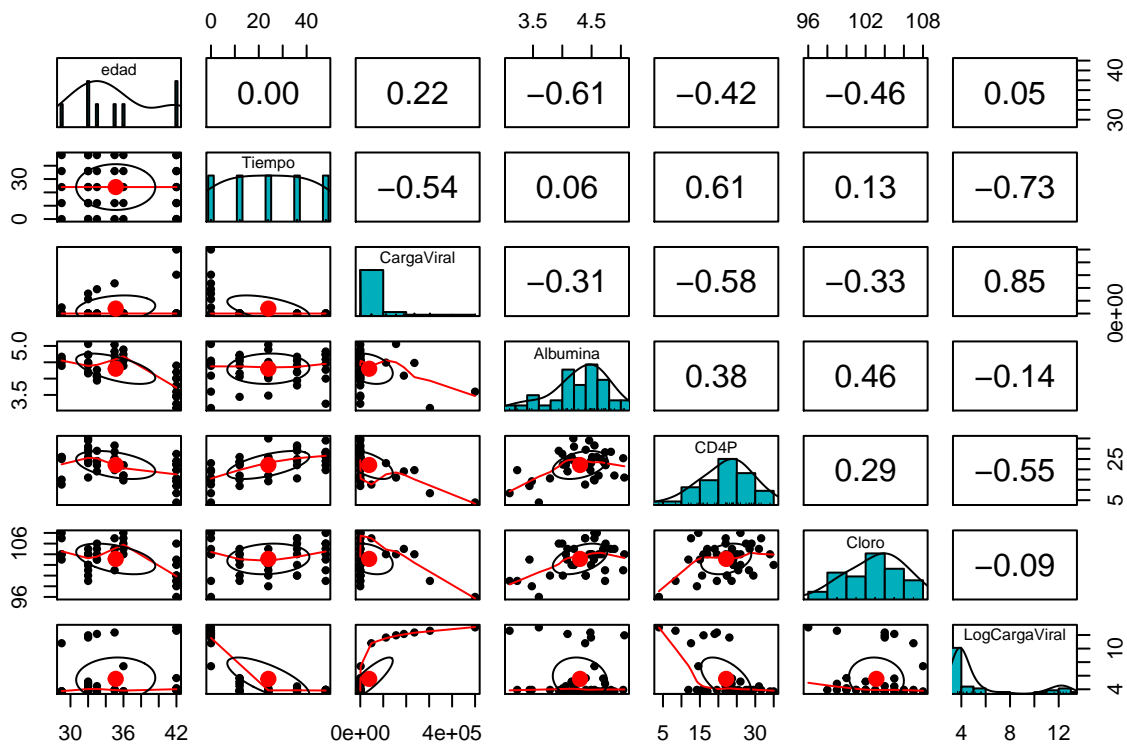
grid.arrange(p01,p02,p03)

## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bindot()' using 'bins = 30'. Pick better value with 'binwidth'.
```



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

```
LAKE1$LogCargaViral <- log(LAKE1$CargaViral)
pairs.panels(LAKE1[,sapply(LAKE1,class)=="numeric"],
  method = "pearson", # correlation method
  hist.col = "#00AFBB",
  density = TRUE, # show density plots
  ellipses = TRUE # show correlation ellipses
)
```



```

p11<-ggplot(LAKE1, aes(x=LogCargaViral)) +
  geom_vline(aes(xintercept=mean(LAKE1$LogCargaViral)),
             color=randomColor(), linetype="dashed", size=1) +
  geom_histogram()

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

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

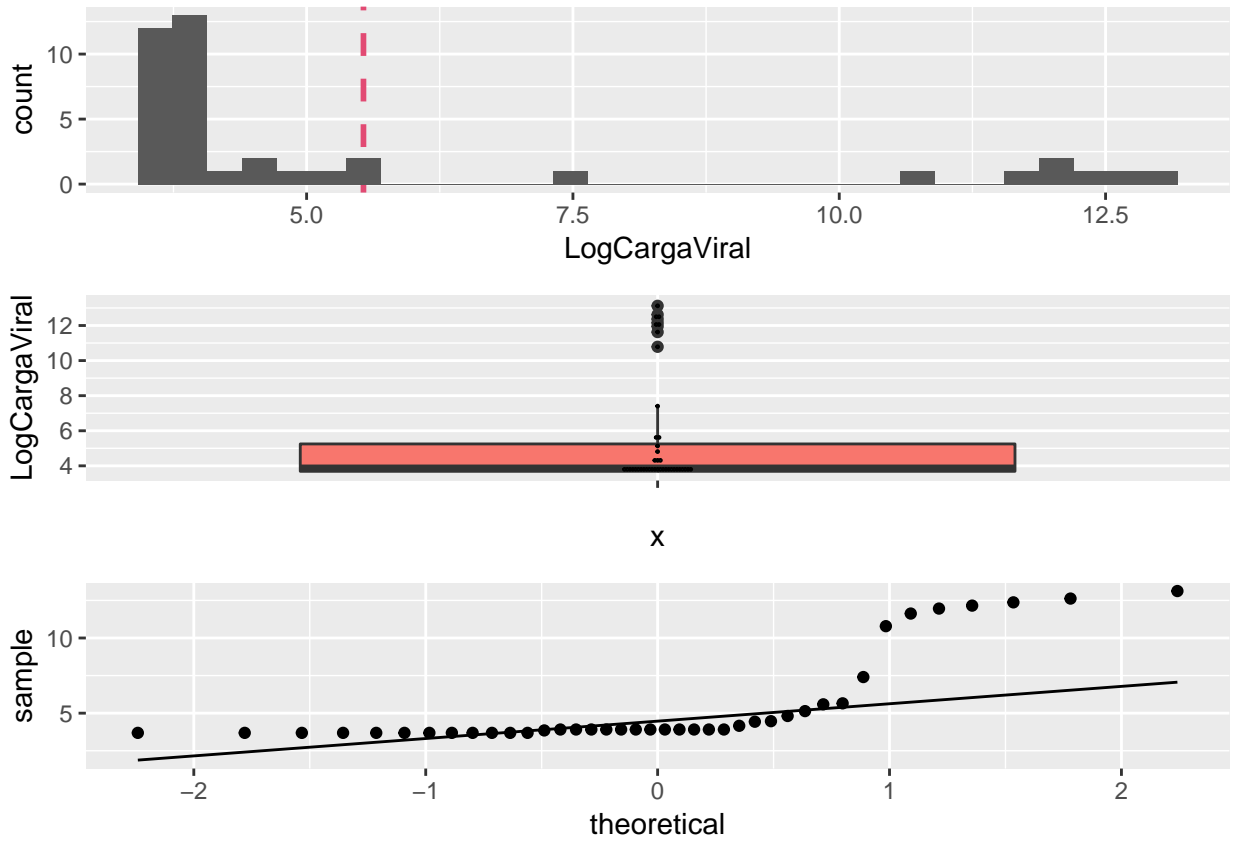
grid.arrange(p11,p12,p13)

```

```

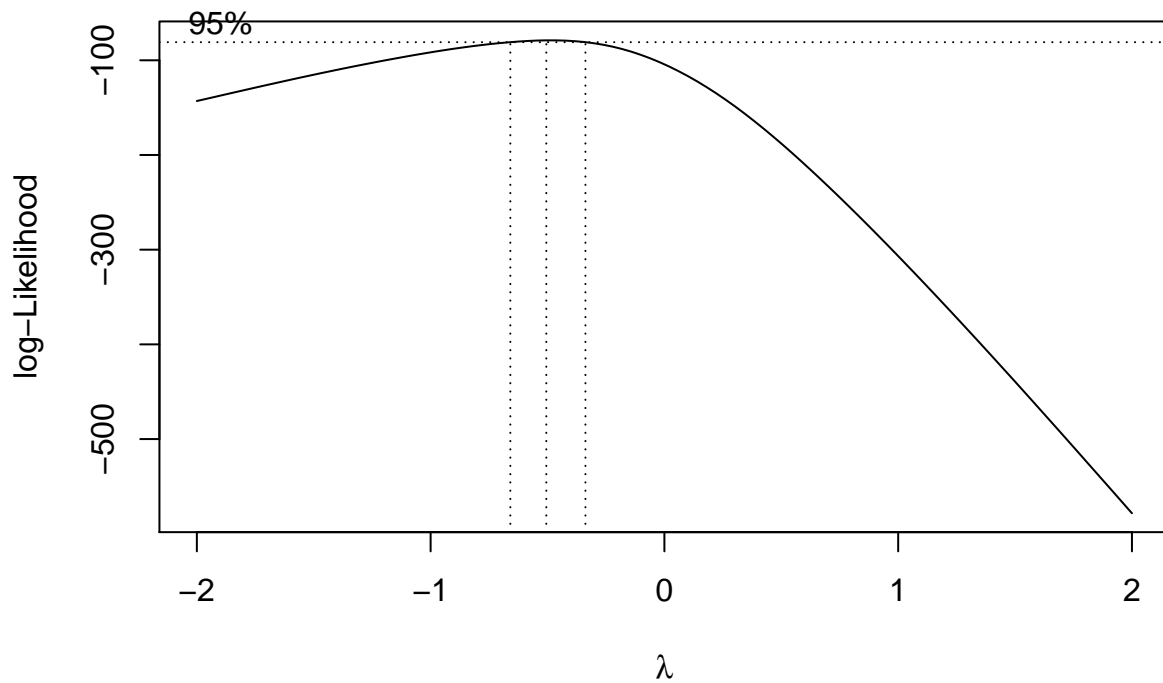
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bindot()' using 'bins = 30'. Pick better value with 'binwidth'.

```



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

```
bct<-boxcox(CargaViral~Tiempo,lambda = seq(-2, 2, length = 10),data=LAKE1,plotit=T)
```



```
lambda<-bct$x[which.max(bct$y)]
lambda
```

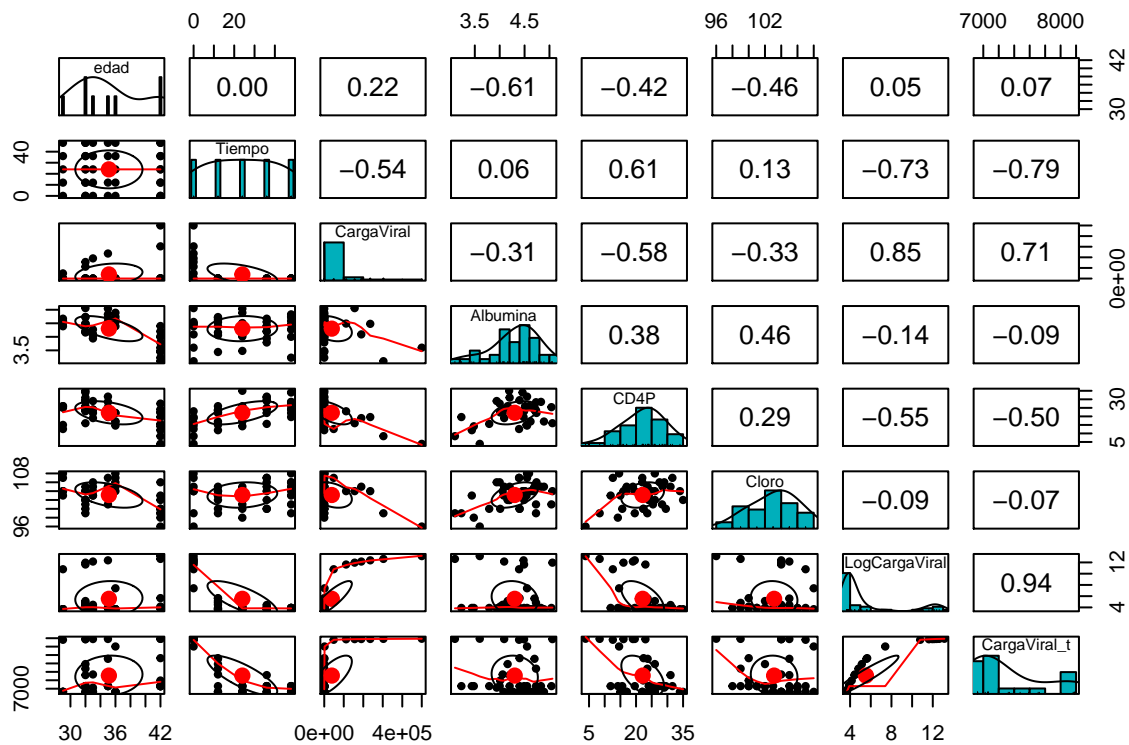
```
## [1] -0.5050505
```

```
y <- LAKE1$CargaViral
gm<-geometric.mean(y)
```

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

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





```

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

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

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

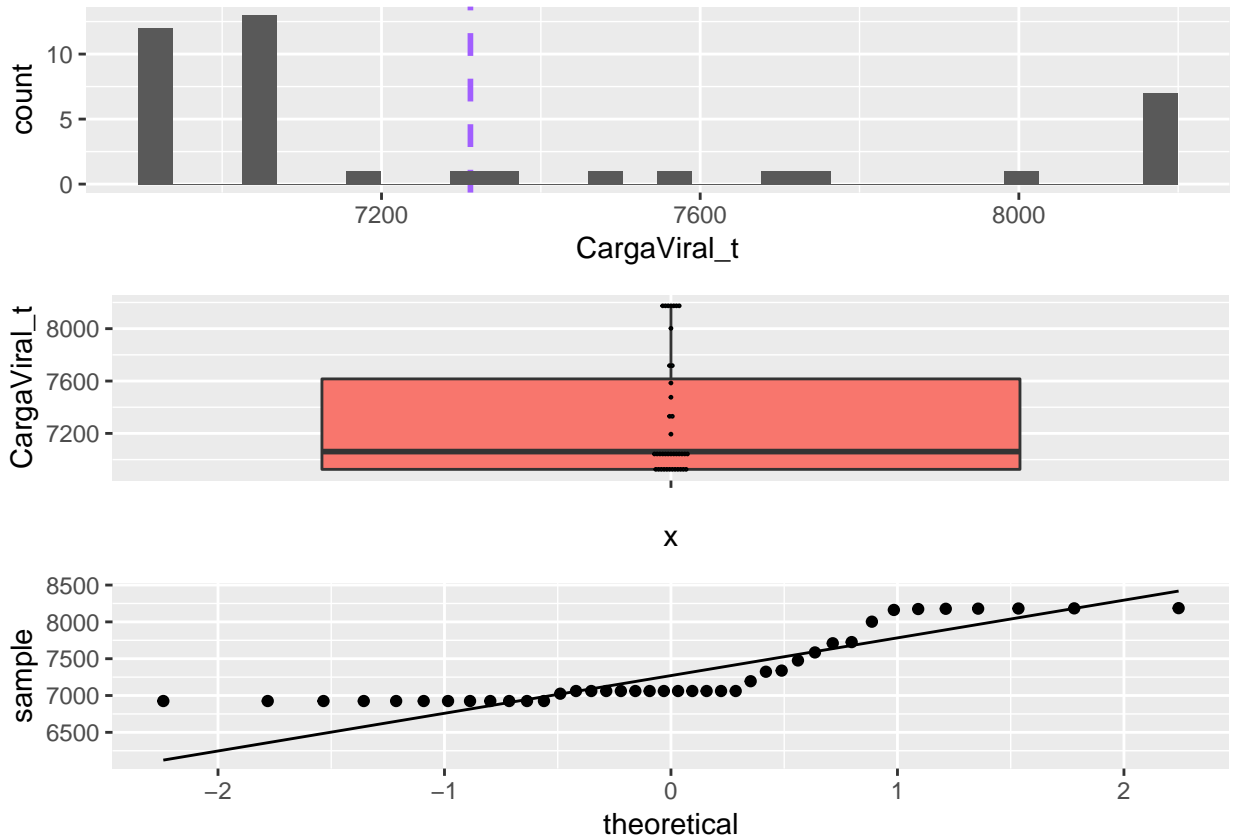
grid.arrange(p21,p22,p23)

```

```

## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bindot()' using 'bins = 30'. Pick better value with 'binwidth'.

```



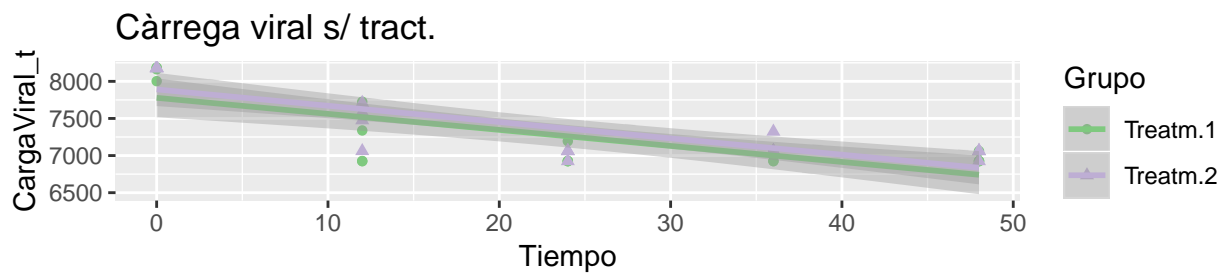
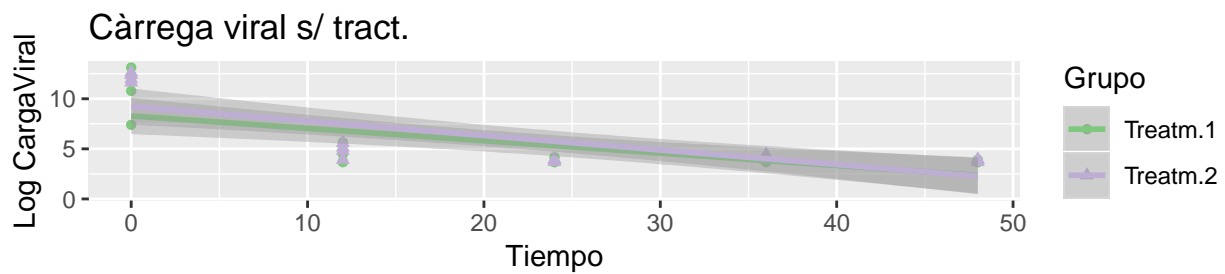
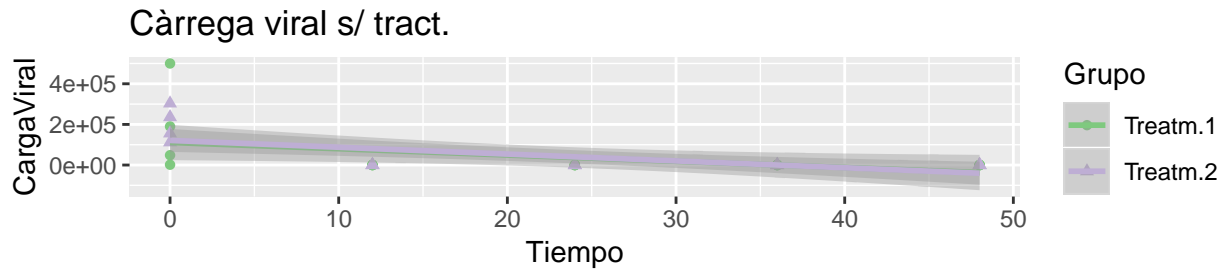
Millora lleugerament respecte a la transformació logarítmica, tot i que els inconvenients potser no justifiquen mantenir aquesta transformació. En principi es mantindrà únicament la logarítmica i en algun cas es pot fer alguna prova per si aquesta transformació aporta una millora significativa.

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

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

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

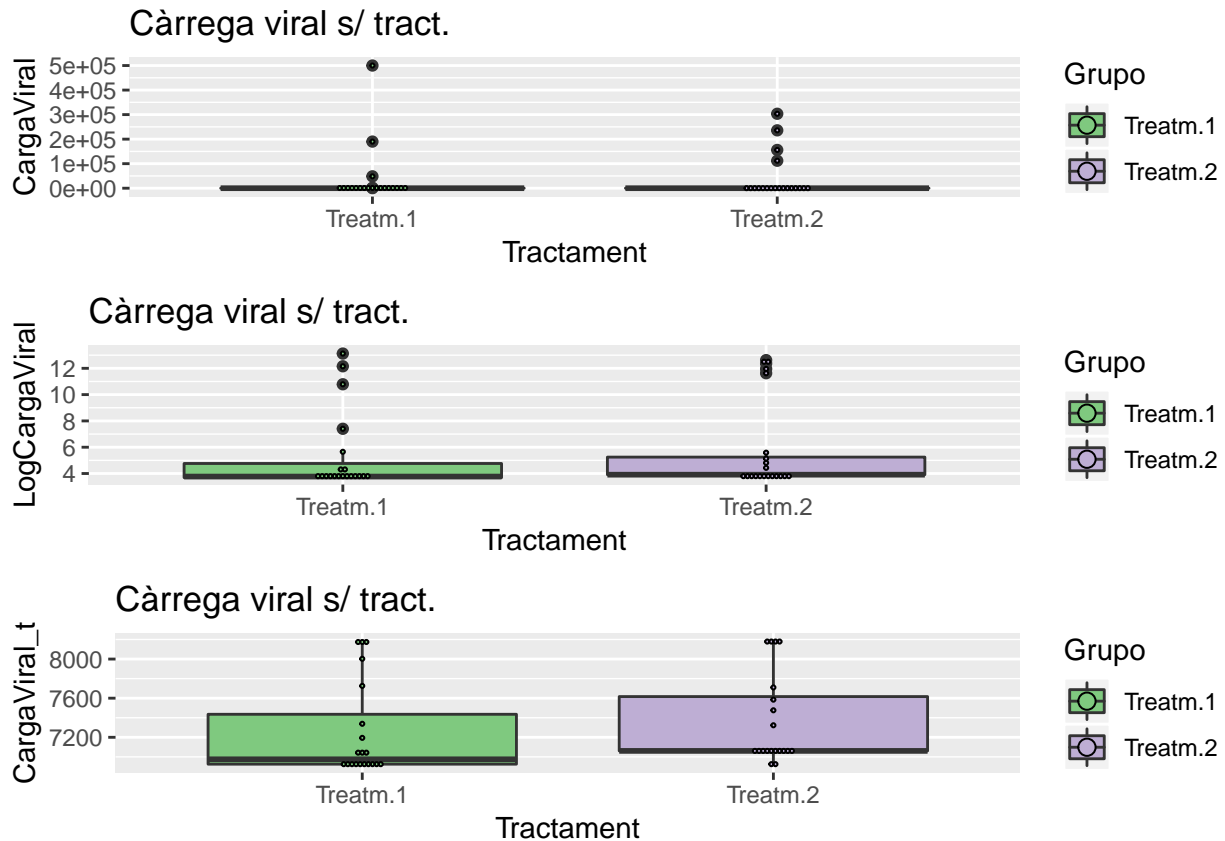
grid.arrange(Lake1SCPlot.a,Lake1SCPlot.b,Lake1SCPlot.c)
```



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

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

```
## 'stat_bindot()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bindot()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bindot()' using 'bins = 30'. Pick better value with 'binwidth'.
```



## AJUST MODEL SIMPLE

### Model mínims quadrats *Stepwise*

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

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

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

```
##
## Call:
## lm(formula = CargaViral ~ Tiempo + CD4P + Cloro, data = LAKE1)
##
## Residuals:
```

```
##      Min      1Q  Median      3Q      Max
## -113727 -55558  -7185   42815  276156
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   910289    478303   1.903  0.0650 .
## Tiempo        -1868       958  -1.950  0.0591 .
## CD4P          -4985       2505  -1.990  0.0542 .
## Cloro         -6948       4748  -1.463  0.1521
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## s: 81390 on 36 degrees of freedom
## Multiple R-squared:  0.4226,
## Adjusted R-squared:  0.3745
## F-statistic: 8.783 on 3 and 36 DF,  p-value: 0.0001673
```

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

```
## Analysis of Variance Table
##
## Response: CargaViral
##           Df      Sum Sq   Mean Sq F value    Pr(>F)
## Tiempo     1 1.1969e+11 1.1969e+11 18.0699 0.0001441 ***
## CD4P       1 4.0649e+10 4.0649e+10  6.1369 0.0180686 *
## Cloro      1 1.4181e+10 1.4181e+10  2.1409 0.1520937
## Residuals 36 2.3845e+11 6.6237e+09
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Fit the full model
```

```
Lake1.fm1 <- lm(LogCargaViral ~ . - CargaViral - CargaViral_t - Ident, data = LAKE1)
```

```
# Stepwise regression model
```

```
Lake1.step.model1 <- stepAIC(Lake1.fm1, direction = "both",
                             trace = F)
```

```
(Lake1.step.model1s <- summary(Lake1.step.model1))
```

```
##
## Call:
## lm(formula = LogCargaViral ~ Tiempo, data = LAKE1)
##
## Residuals:
##      Min      1Q  Median      3Q      Max
## -3.4563 -1.6775 -0.0412  1.6012  4.3657
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.7566     0.5966  14.678 < 2e-16 ***
## Tiempo        -0.1343     0.0203  -6.616 8.16e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## s: 2.178 on 38 degrees of freedom
## Multiple R-squared:  0.5353,
## Adjusted R-squared:  0.5231
```

```
## F-statistic: 43.78 on 1 and 38 DF, p-value: 8.162e-08
```

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

```
##
## Call:
## lm(formula = CargaViral_t ~ Tiempo, data = LAKE1)
##
## Residuals:
##   Min     1Q  Median     3Q    Max
## -647.6 -238.6   11.4  272.2  352.4
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7834.432     81.191  96.494 < 2e-16 ***
## Tiempo      -21.784      2.762  -7.887 1.61e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## s: 296.5 on 38 degrees of freedom
## Multiple R-squared:  0.6208,
## Adjusted R-squared:  0.6108
## F-statistic: 62.2 on 1 and 38 DF, p-value: 1.606e-09
```

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

	M.Simple sense transf. 1			M.Simple log transf			M.Simple Box-Cox t		
	Estimate	Std. Error	Pr(> t )	Estimate	Std. Error	Pr(> t )	Estimate	Std. Error	Pr(> t )
Tiempo	-1867.5819	957.9577	0.0591	-0.1343	0.0203	0	-21.7842	2.7622	0
CD4P	-4985.2783	2505.0233	0.0542	-	-	-	-	-	-
Cloro	-6947.9474	4748.509	0.1521	-	-	-	-	-	-

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

Model	Mediana residual
M.Simple sense transf	-7184.6798137
M.Simple log transf	-0.0411551
M.Simple Box-Cox t	11.4023266

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

- El model amb variables originals s'ha considerat òptima la utilització de 3 predictors (*Tiempo*, *CD4P*, *Cloro*) per explicar la resposta, els 3 amb una prova de significació no significativa (p-valors de 0.0591, 0.0542, 0.1521 respectivament). En el cas de l'ANOVA es calcula la significació si es treuen del model 1 a 1 veient l'ordre de importància dels termes *Tiempo*, *CD4P*, *Cloro* de més a menys significatiu.
- En els 2 models transformats, el sistema decideix pels 2 un model més simple que depèn només del temps essent aquest molt significatiu per explicar la resposta (p-valors de 0 i 0 respectivament per transformació Log i *Box-Cox*).

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

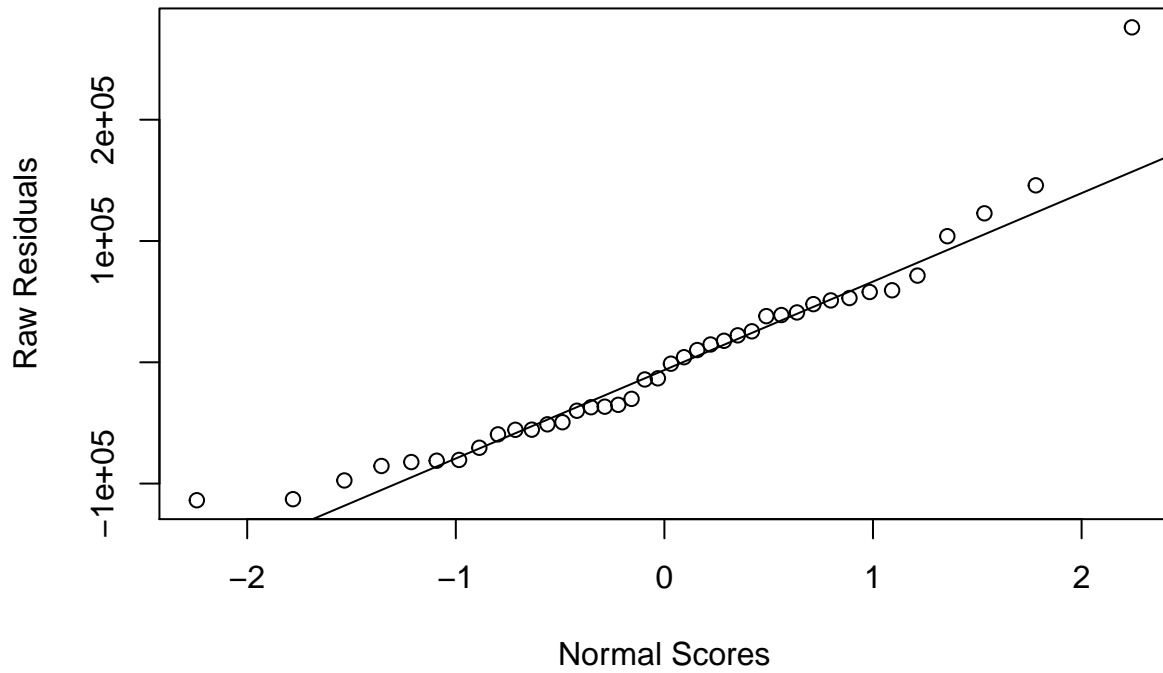
## Diagnòstic model simple

### Normalitat / Gràfics Residuals

```
Lake1.SM0.sdres <- rstandard(Lake1.step.model0)
Lake1.SM0.studres <- studres(Lake1.step.model0)
par(mfrow=c(1,1))

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

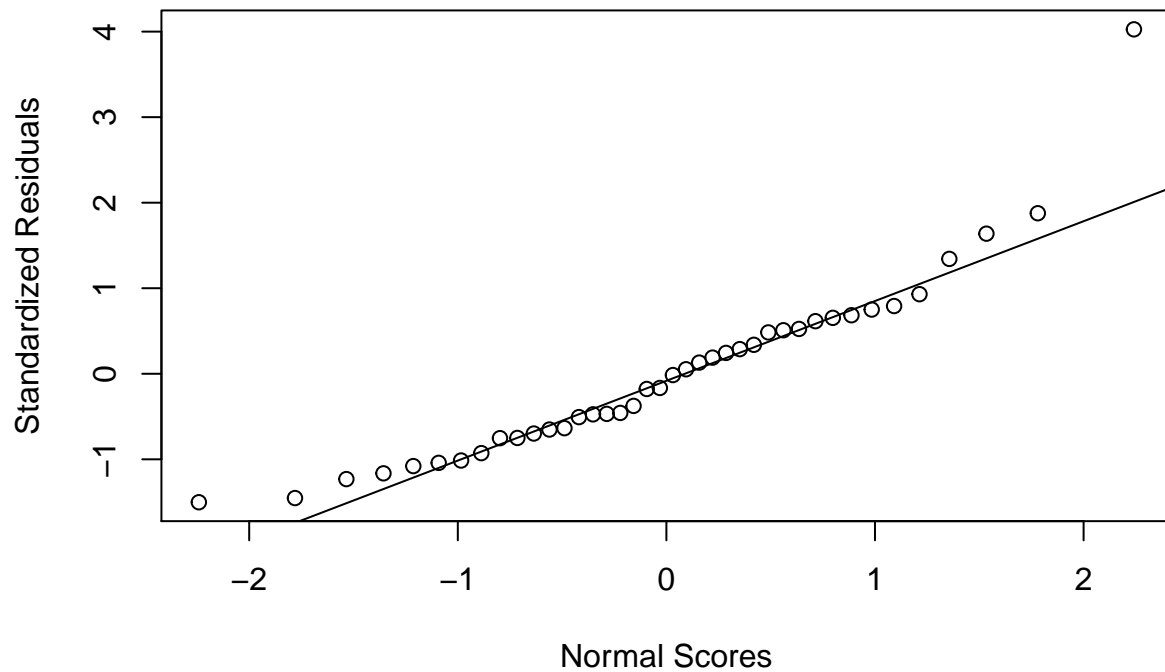
## Model Original – NPP of Raw Residuals



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

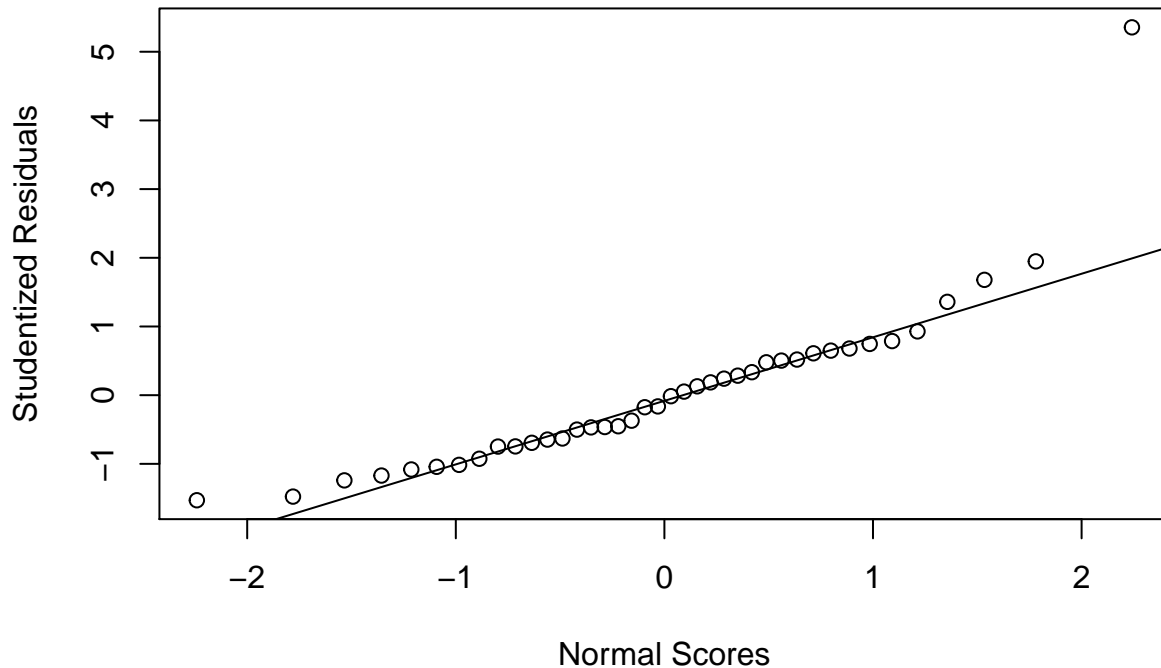


## Model Original – NPP of Standardized Residuals



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

## Model Original – NPP of Studentized Residuals



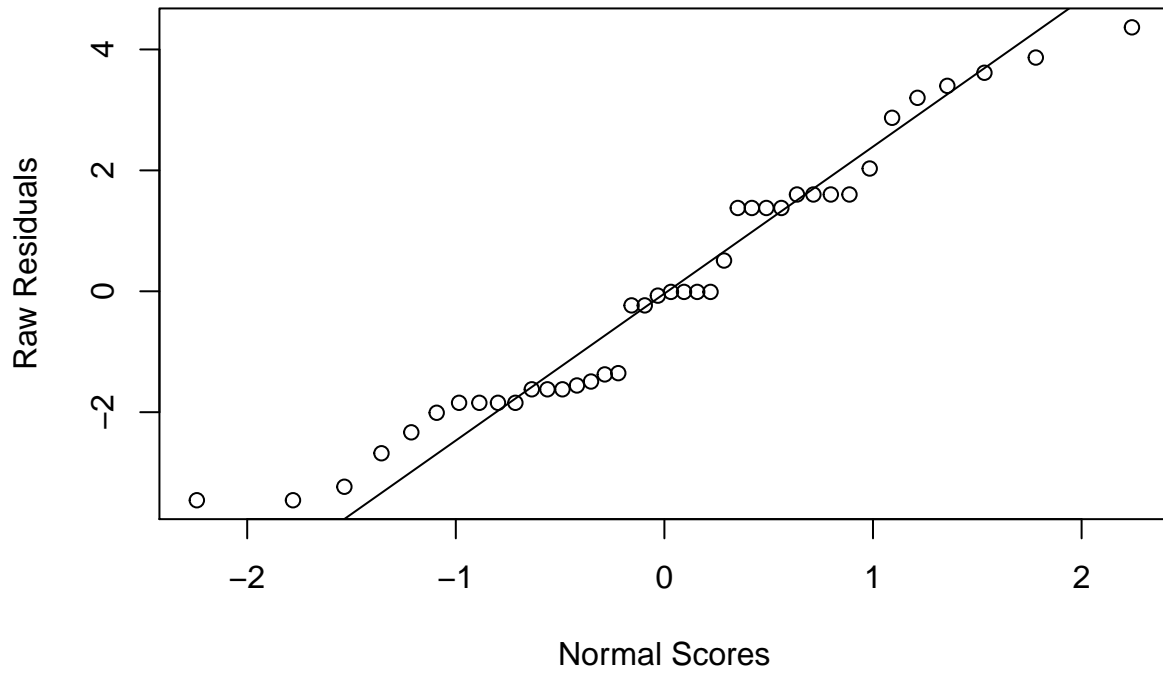
```
S0 <- shapiro.test(Lake1.step.model0$residuals)
S0
```

```
##
## Shapiro-Wilk normality test
##
## data: Lake1.step.model0$residuals
## W = 0.92419, p-value = 0.01046
```

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

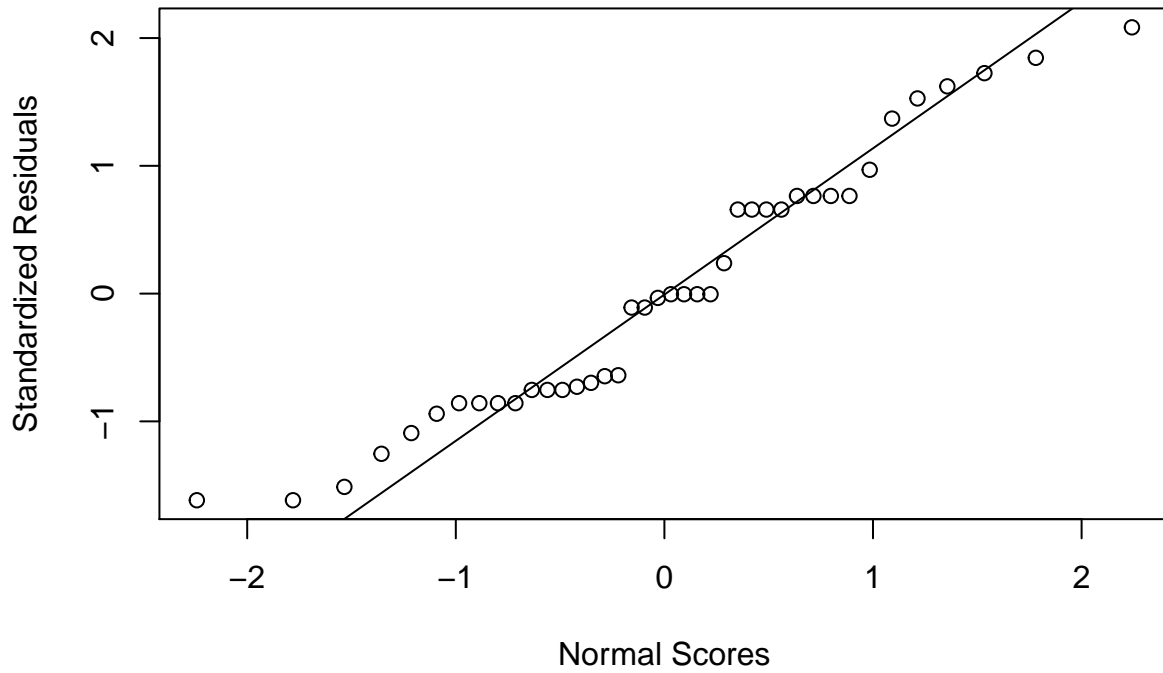
```
qqnorm(Lake1.step.model1$residuals,
       ylab="Raw Residuals",
       xlab="Normal Scores",
       main="Model Log - NPP of Raw Residuals")
qqline(Lake1.step.model1$residuals)
```

## Model Log – NPP of Raw Residuals



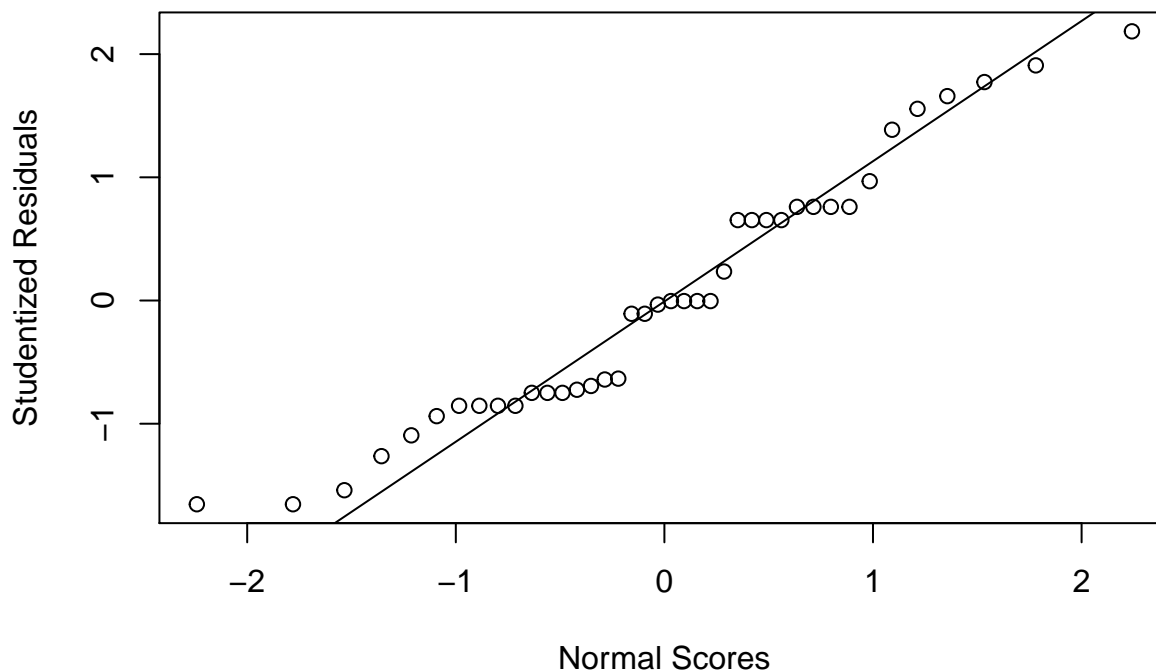
```
qqnorm(Lake1.SM1.sdres,  
        ylab="Standardized Residuals",  
        xlab="Normal Scores",  
        main="Model Log - NPP of Standardized Residuals")  
qqline(Lake1.SM1.sdres)
```

## Model Log – NPP of Standardized Residuals



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

## Model Log – NPP of Studentized Residuals



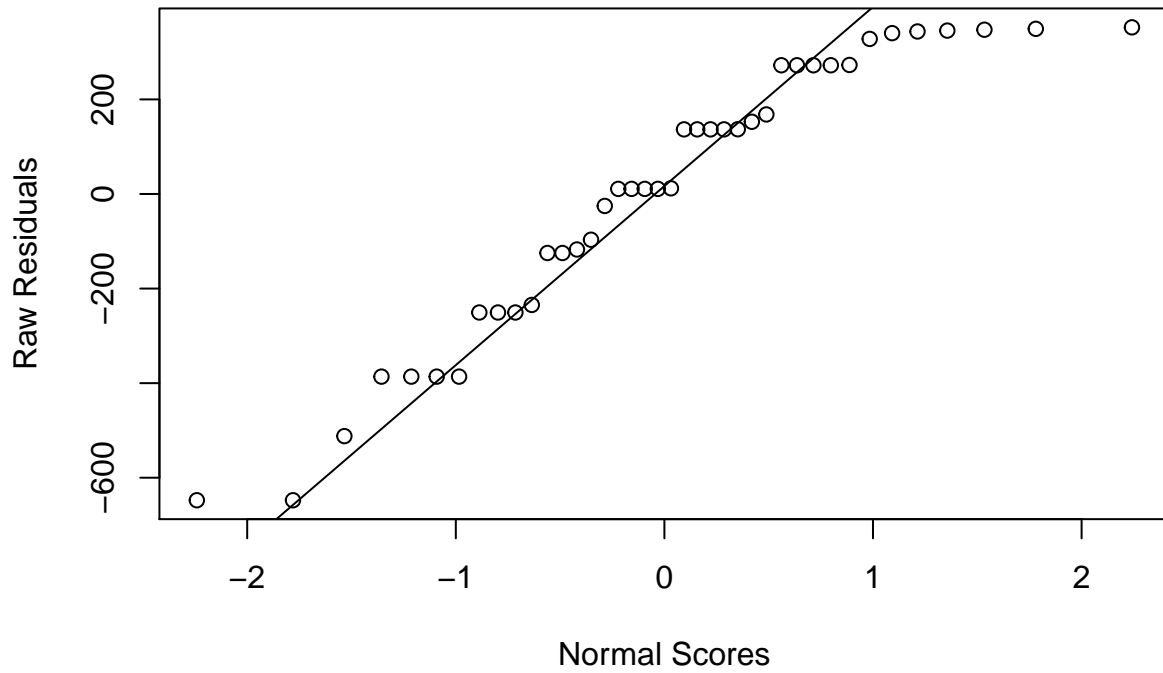
```
S1 <- shapiro.test(Lake1.step.model1$residuals)
S1

##
## Shapiro-Wilk normality test
##
## data:  Lake1.step.model1$residuals
## W = 0.94994, p-value = 0.07547

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

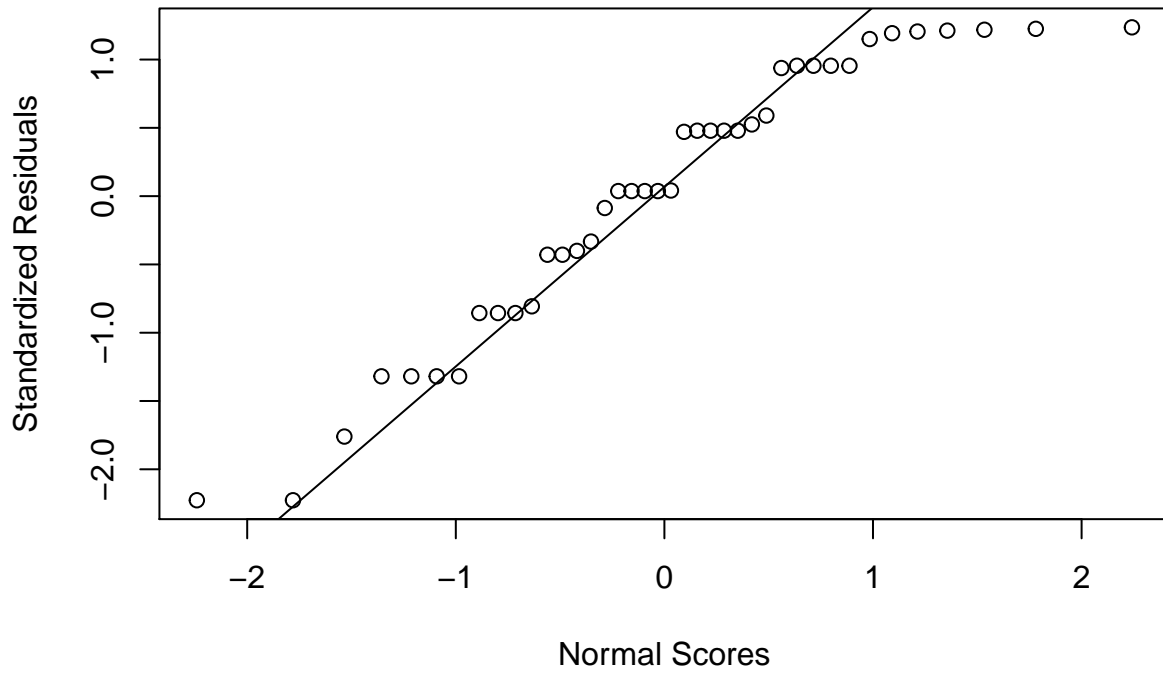
qqnorm(Lake1.step.model2$residuals,
       ylab="Raw Residuals",
       xlab="Normal Scores",
       main="Model Box-Cox – NPP of Raw Residuals")
qqline(Lake1.step.model2$residuals)
```

## Model Box-Cox - NPP of Raw Residuals



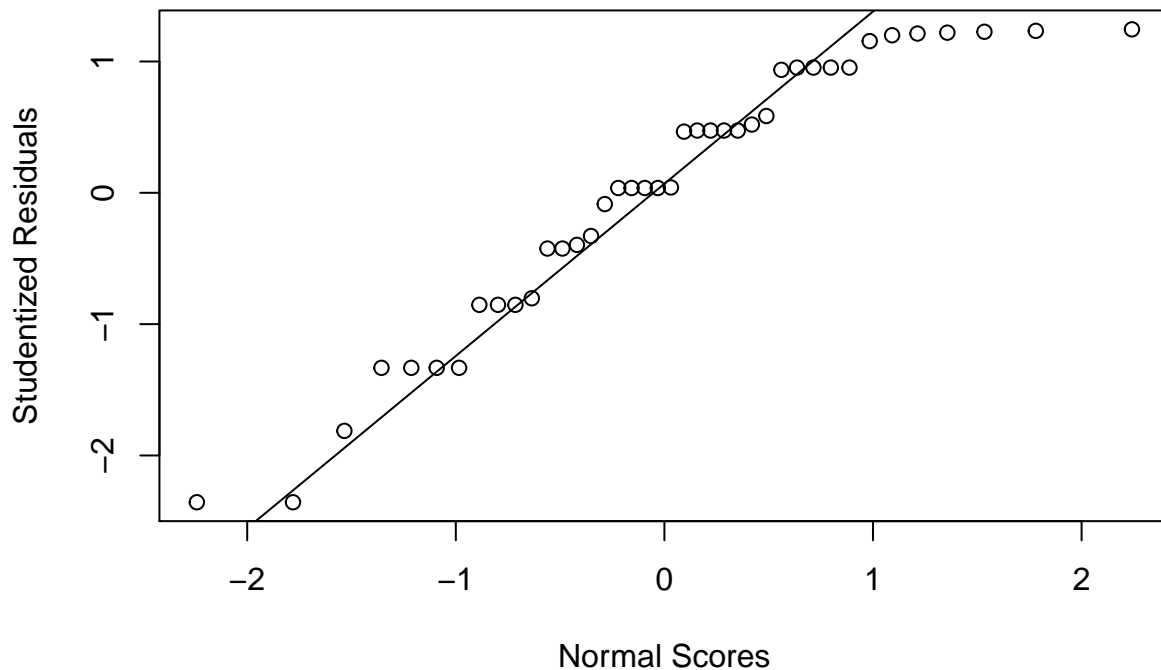
```
qqnorm(Lake1.SM2.sdres,  
       ylab="Standardized Residuals",  
       xlab="Normal Scores",  
       main="Model Box-Cox - NPP of Standardized Residuals")  
qqline(Lake1.SM2.sdres)
```

## Model Box-Cox - NPP of Standardized Residuals



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

## Model Box-Cox – NPP of Studentized Residuals



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

```
##
## Shapiro-Wilk normality test
##
## data: Lake1.step.model2$residuals
## W = 0.91901, p-value = 0.00717
```

```
par(mfrow=c(1,3))
```

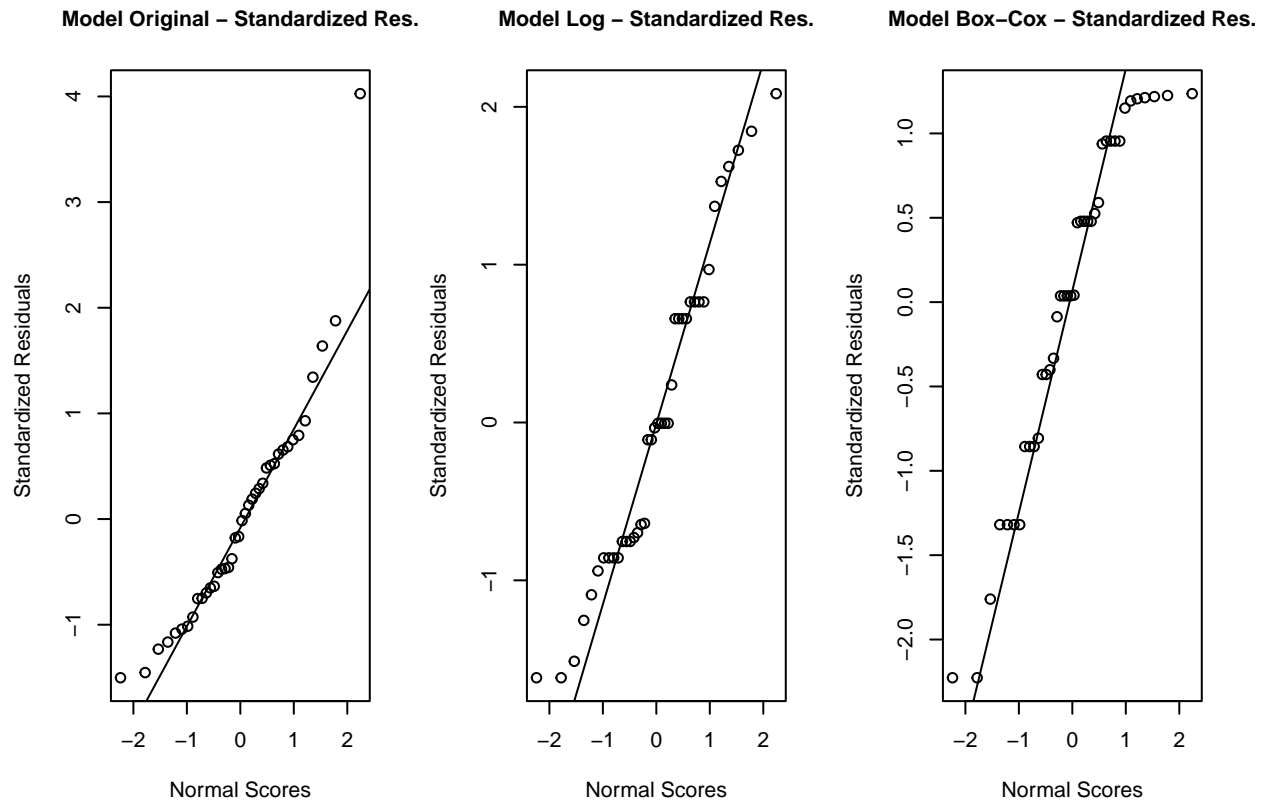
```
qqnorm(Lake1.SM0.sdres,
       ylab="Standardized Residuals",
       xlab="Normal Scores",
       main="Model Original - Standardized Res.",cex.main=1)
qqline(Lake1.SM0.sdres)
```

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

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



```
qqline(Lake1.SM2.sdres)
```



Sorprenentment els 3 models entren en el que es podria classificar com una distribuci6 normal dels residus del model tot i que no queda suportat per la prova de *shapiro* amb p-valors del model original, logtransformat i *box-cox* de 0.0105, 0.0755 i 0.0072 respectivament.

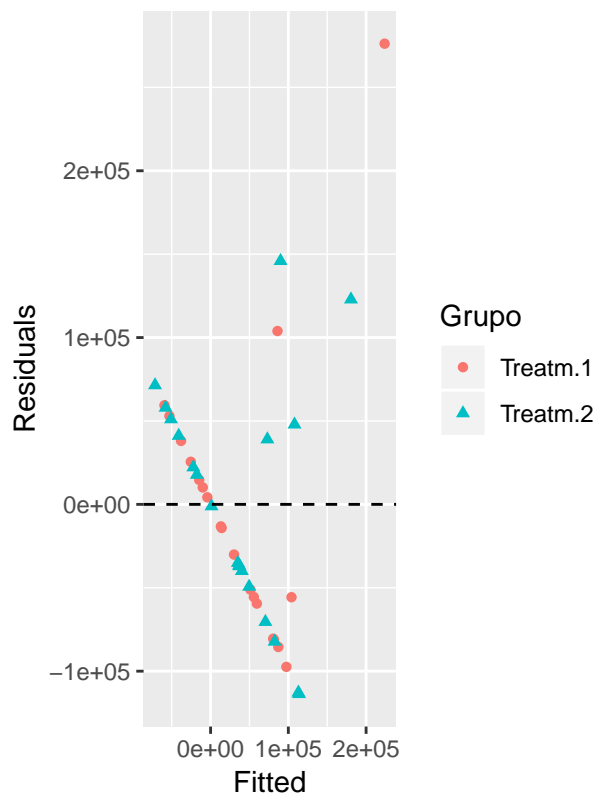
## Homoscedasticitat

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

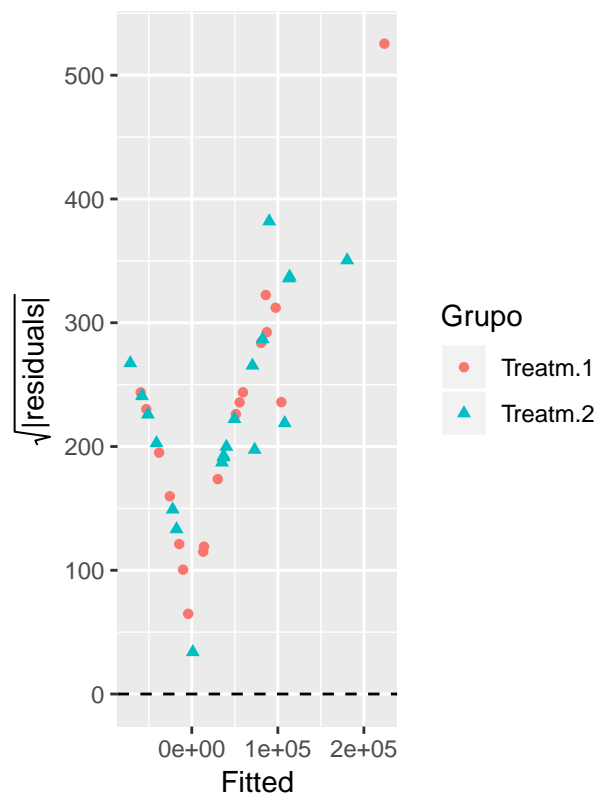
pH02 <-ggplot(LAKE1, aes(x=fitted(Lake1.step.model0), y=sqrt(abs(residuals(Lake1.step.model0))),
                        shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals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

grid.arrange(pH01, pH02,ncol=2)
```

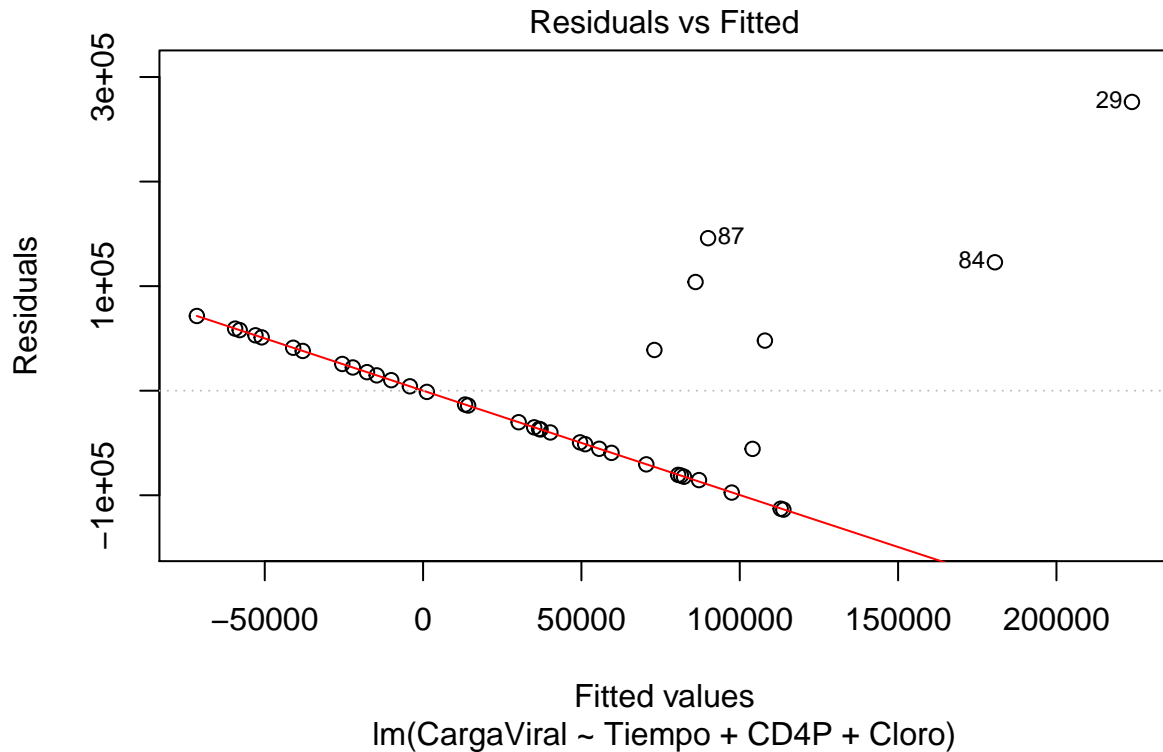
Fitted vs Residuals (Homoceda



Fitted vs Residuals (Homocedast



```
plot(Lake1.step.model0,which=1)
```



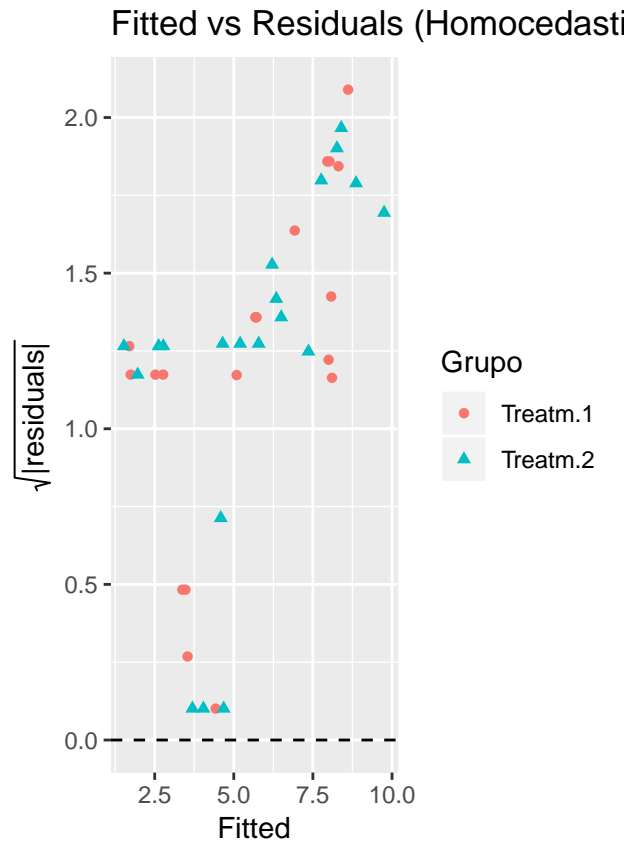
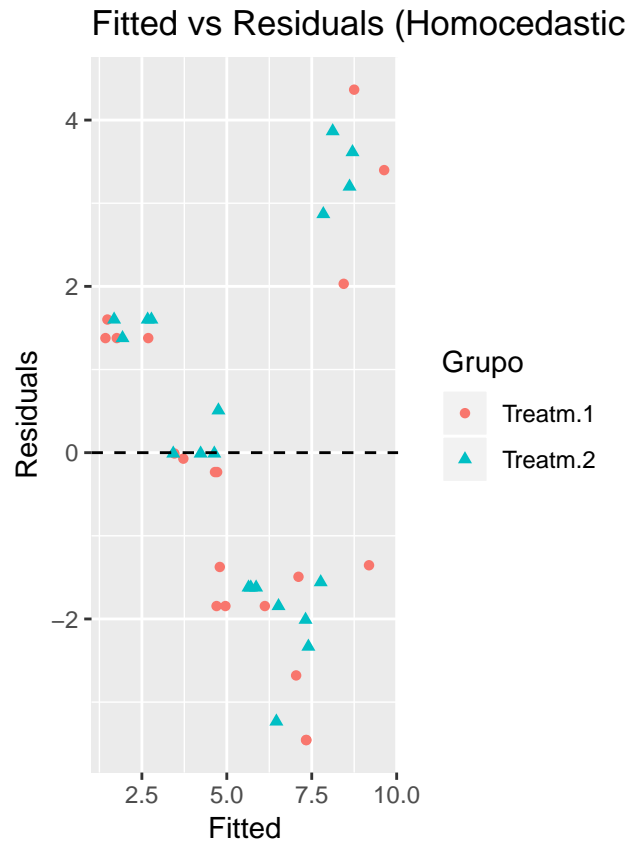
```
BP01<-bptest(Lake1.step.model0)
BP01
```

```
##
## studentized Breusch-Pagan test
##
## data: Lake1.step.model0
## BP = 17.345, df = 3, p-value = 6e-04
```

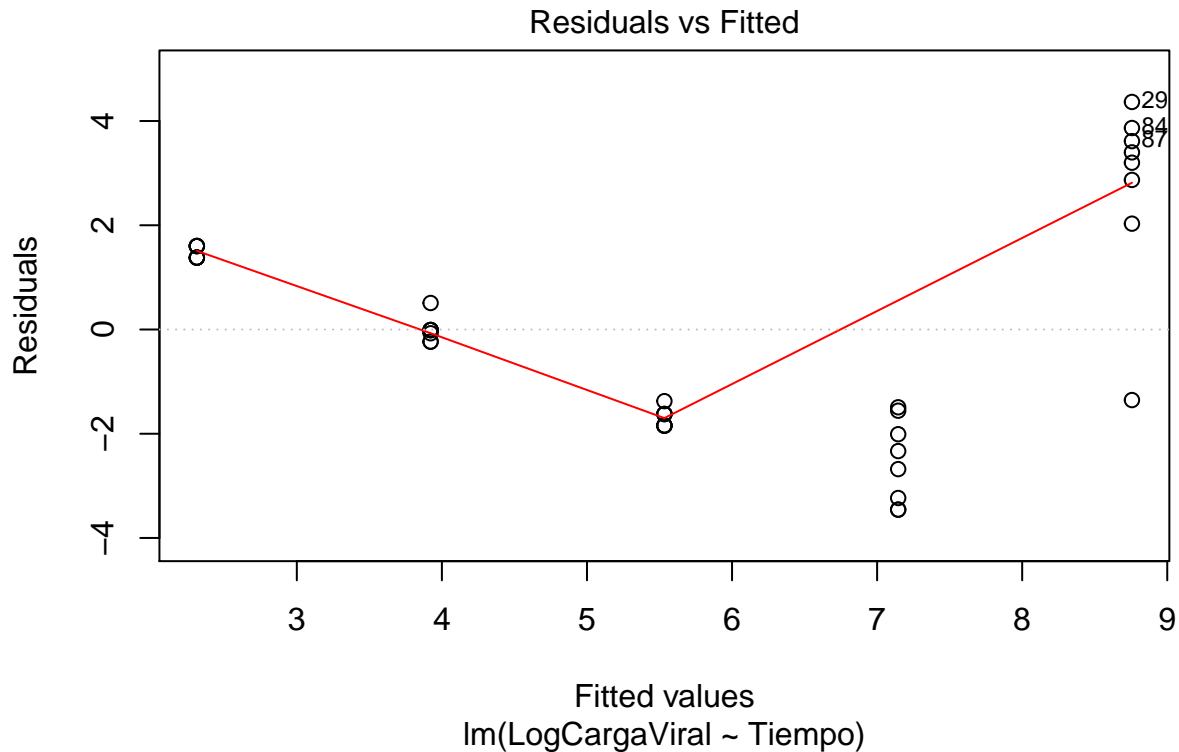
```
pH11 <- ggplot(LAKE1, aes(x=fitted(Lake1.step.model1), y=residuals(Lake1.step.model1),
                          shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Homocedasticity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

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

```
grid.arrange(pH11, pH12,ncol=2)
```



```
plot(Lake1.step.model1, which=1)
```



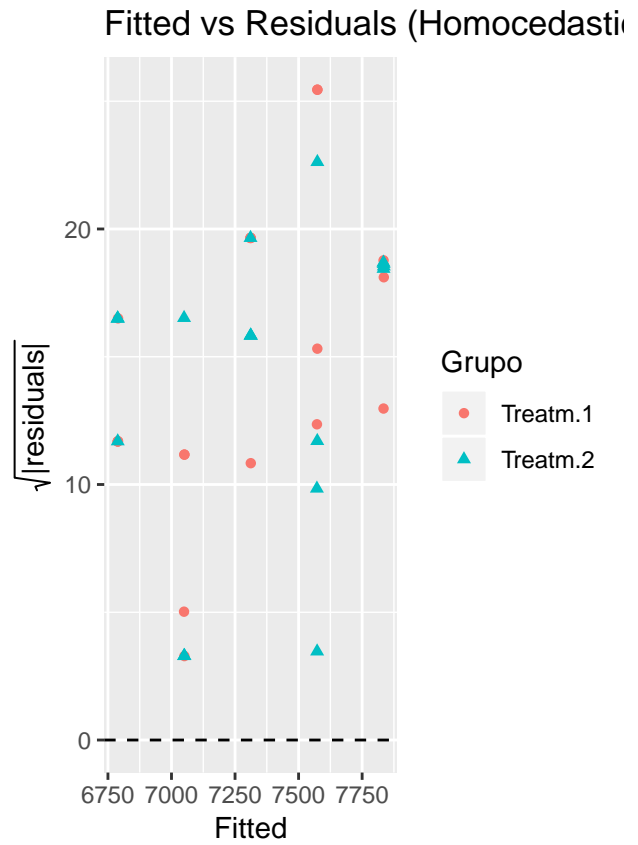
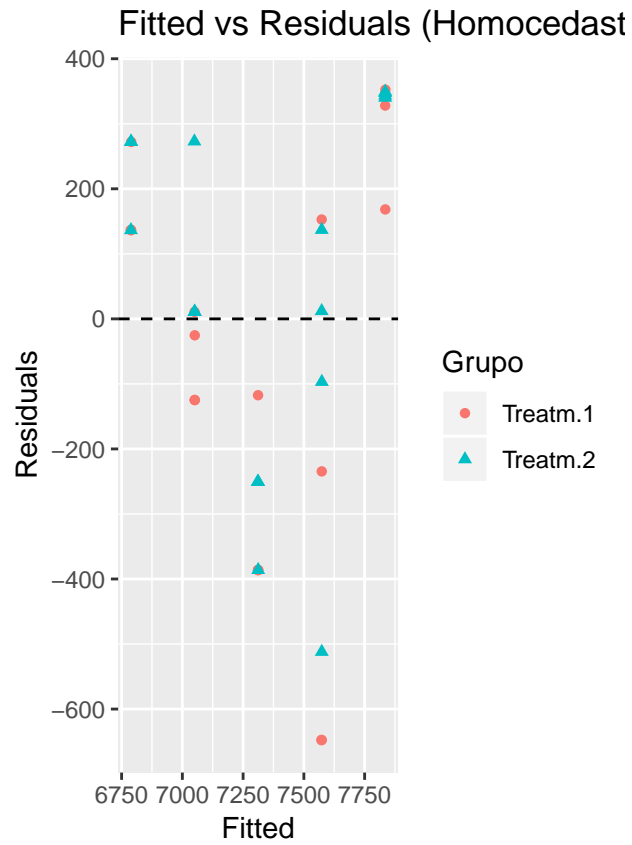
```
BP11<-bptest(Lake1.step.model1)
BP11
```

```
##
## studentized Breusch-Pagan test
##
## data: Lake1.step.model1
## BP = 19.474, df = 1, p-value = 1.02e-05
```

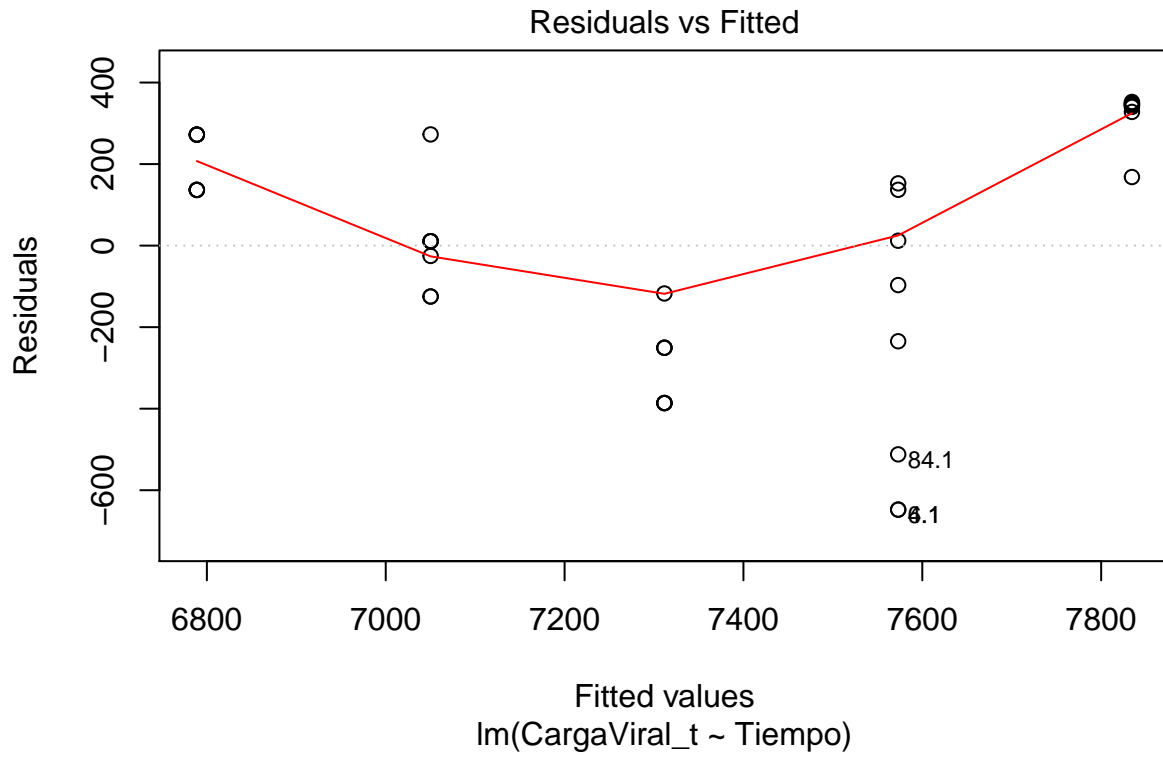
```
pH21 <- ggplot(LAKE1, aes(x=fitted(Lake1.step.model2), y=residuals(Lake1.step.model2),
                          shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Homocedasticity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

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

```
grid.arrange(pH21, pH22,ncol=2)
```



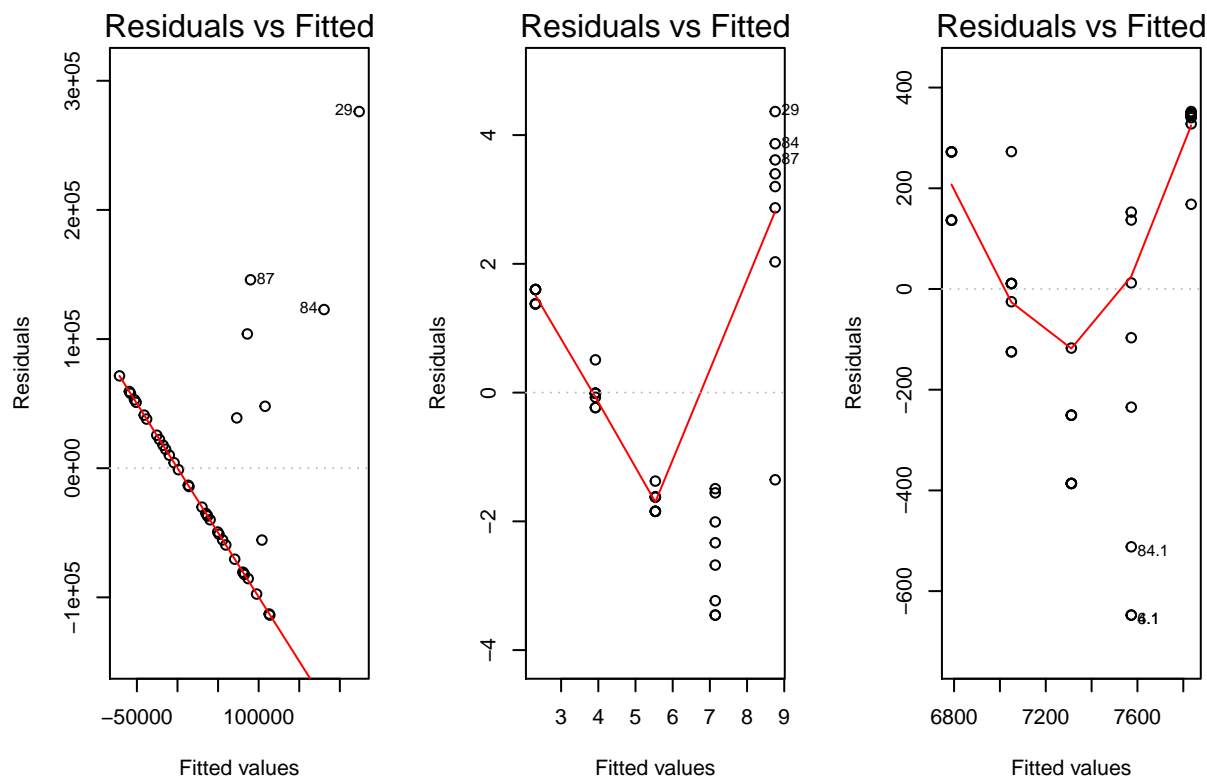
```
plot(Lake1.step.model2, which=1)
```



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

```
##
## studentized Breusch-Pagan test
##
## data: Lake1.step.model2
## BP = 5.718, df = 1, p-value = 0.01679
```

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



Els gràfics es veu que:

- Semblaria clarament que al model original es té un problema de heteroscedasticitat amb els residus que semblen seguir clarament una tendència negativa excepte per uns quants punts. També ho suporta el test de *Breusch-Pagan* amb un p-valor de  $6e-04$ .
- Al transformar logarímicament sembla que es compensa tot i donar també aparentment un patró de comportament negatiu i amb un p-valor de BP de 0.
- En la transformació Cox-Box seria potser l'únic model que donaria una homogeneïtat aparent residual tot i que també sembla seguir una certa forma no lineal, i amb un p-valor de BP de 0.0168.

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

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

```
by(residuals(Lake1.step.model0), LAKE1$Grupo, var)
```

```
## LAKE1$Grupo: Treatm.1
## [1] 7328369519
## -----
## LAKE1$Grupo: Treatm.2
## [1] 5213866561
```

```
by(residuals(Lake1.step.model1), LAKE1$Grupo, var)
```

```
## LAKE1$Grupo: Treatm.1
```



```
## [1] 4.704479
## -----
## LAKE1$Grupo: Treatm.2
## [1] 4.686232
by(residuals(Lake1.step.model2),LAKE1$Grupo,var)

## LAKE1$Grupo: Treatm.1
## [1] 98189.66
## -----
## LAKE1$Grupo: Treatm.2
## [1] 71892.55
```

### Diagnòstic possibles outliers

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

```
LAKE1[which.max(qq01$y),]
```

```
##      edad      Grupo      sexo      Ident Tiempo CargaViral Albumina CD4P Cloro
## 29      42 Treatm.1 Hombre gcarosi 007      0      5e+05      3.6 3.9 96
##      LogCargaViral CargaViral_t
## 29      13.12236      8186.802
```

```
LAKE1[which.max(sqrt(abs(residuals(Lake1.step.model0))))],]
```

```
##      edad      Grupo      sexo      Ident Tiempo CargaViral Albumina CD4P Cloro
## 29      42 Treatm.1 Hombre gcarosi 007      0      5e+05      3.6 3.9 96
##      LogCargaViral CargaViral_t
## 29      13.12236      8186.802
```

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

```
Lake1.step.model0s
```

```
##
## Call:
## lm(formula = CargaViral ~ Tiempo + CD4P + Cloro, data = LAKE1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -113727 -55558  -7185   42815  276156
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   910289    478303   1.903  0.0650 .
## Tiempo        -1868         958  -1.950  0.0591 .
## CD4P          -4985        2505  -1.990  0.0542 .
## Cloro         -6948        4748  -1.463  0.1521
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## s: 81390 on 36 degrees of freedom
## Multiple R-squared: 0.4226,
```

```
## Adjusted R-squared: 0.3745
## F-statistic: 8.783 on 3 and 36 DF, p-value: 0.0001673
LAKE1c <- LAKE1[-which.max(qq01$y),]
# Fit the full model
Lake1c.fm0 <- lm(CargaViral ~ . - LogCargaViral - CargaViral_t - Ident, data = LAKE1c)
# Stepwise regression model
Lake1c.step.modelo0 <- stepAIC(Lake1c.fm0, direction = "both",
                             trace = F)
(Lake1c.step.modelo0s <- summary(Lake1c.step.modelo0))
```

```
##
## Call:
## lm(formula = CargaViral ~ Tiempo + Albumina, data = LAKE1c)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -82200 -40919  -7309   29408 183024
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 210820.1    92558.5   2.278 0.028785 *
## Tiempo      -2351.3      568.6  -4.135 0.000203 ***
## Albumina    -29172.7    21093.4  -1.383 0.175175
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## s: 59440 on 36 degrees of freedom
## Multiple R-squared: 0.3469,
## Adjusted R-squared: 0.3106
## F-statistic: 9.561 on 2 and 36 DF, p-value: 0.0004673
```

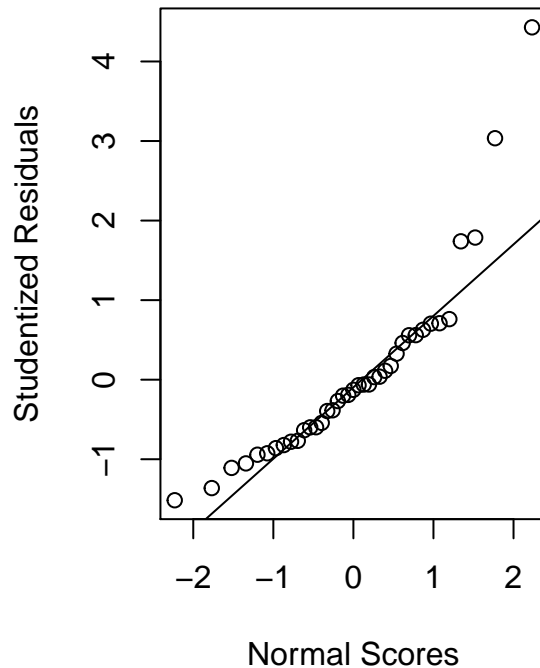
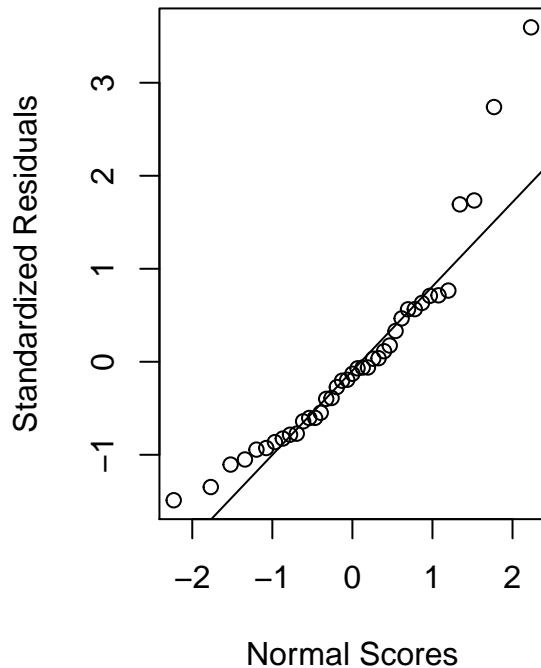
```
Lake1c.SM0.sdres <- rstandard(Lake1c.step.modelo0)
Lake1c.SM0.studres <- studres(Lake1c.step.modelo0)
```

```
par(mfrow=c(1,2))
```

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

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

## el Original – NPP of Standardized Residuals Original – NPP of Studentized Residuals



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

```
##
## Shapiro-Wilk normality test
##
## data: Lake1c.step.model0$residuals
## W = 0.89417, p-value = 0.001508
```

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

```
##      edad  Grupo  sexo      Ident Tiempo CargaViral Albumina CD4P Cloro
## 84    42 Treatm.2 Hombre gcarosi 006      0    303409      3.1  8.4   99
##      LogCargaViral CargaViral_t
## 84          12.62284      8183.689
```

Es mostra un gran canvi ja que el sistema deixa de tenir en compte els 2 predictors que tenia en compte abans *Tiempo*, *CD4P*, *Cloro* i té en compte ara *Tiempo*, *Albumina*, però es continua veient que al gràfic QQ estudentitzat i estandarditzat punts que estan exageradament per fora de la mostra. Aquí cal plantejar-se si aquests punts són tots *outliers*, si formen part d'una població diferent o si el model li falta un ajust addicional per contemplar aquesta diferenciació.

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

```
Lake1.step.model0s
```

```
##
## Call:
## lm(formula = CargaViral ~ Tiempo + CD4P + Cloro, data = LAKE1)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -113727  -55558   -7185   42815  276156
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   910289     478303   1.903  0.0650 .
## Tiempo        -1868         958  -1.950  0.0591 .
## CD4P          -4985         2505  -1.990  0.0542 .
## Cloro         -6948         4748  -1.463  0.1521
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## s: 81390 on 36 degrees of freedom
## Multiple R-squared:  0.4226,
## Adjusted R-squared:  0.3745
## F-statistic: 8.783 on 3 and 36 DF,  p-value: 0.0001673
```

```
Lake1c.step.model0s
```

```
##
## Call:
## lm(formula = CargaViral ~ Tiempo + Albumina, data = LAKE1c)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -82200  -40919   -7309   29408  183024
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 210820.1     92558.5   2.278 0.028785 *
## Tiempo      -2351.3         568.6  -4.135 0.000203 ***
## Albumina    -29172.7     21093.4  -1.383 0.175175
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## s: 59440 on 36 degrees of freedom
## Multiple R-squared:  0.3469,
## Adjusted R-squared:  0.3106
## F-statistic: 9.561 on 2 and 36 DF,  p-value: 0.0004673
```

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

```
##
## Call:
## lm(formula = CargaViral ~ Grupo + Tiempo + Cloro, data = LAKE1c2)
##
## Residuals:
```

```

##      Min      1Q Median      3Q      Max
## -73430 -27076 -6571  16815 155899
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -424385.0  337613.6  -1.257 0.217316
## Grupo(Treatm.1) -11930.9   8379.0  -1.424 0.163586
## Tiempo        -1808.5    464.3  -3.895 0.000437 ***
## Cloro          4735.1    3259.2   1.453 0.155435
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## s: 47120 on 34 degrees of freedom
## Multiple R-squared:  0.3507,
## Adjusted R-squared:  0.2934
## F-statistic: 6.122 on 3 and 34 DF,  p-value: 0.001913

```

```

Lake1c2.SMO.sdres <- rstandard(Lake1c2.step.modelo0)
Lake1c2.SMO.studres <- studres(Lake1c2.step.modelo0)

```

```
par(mfrow=c(1,2))
```

```

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

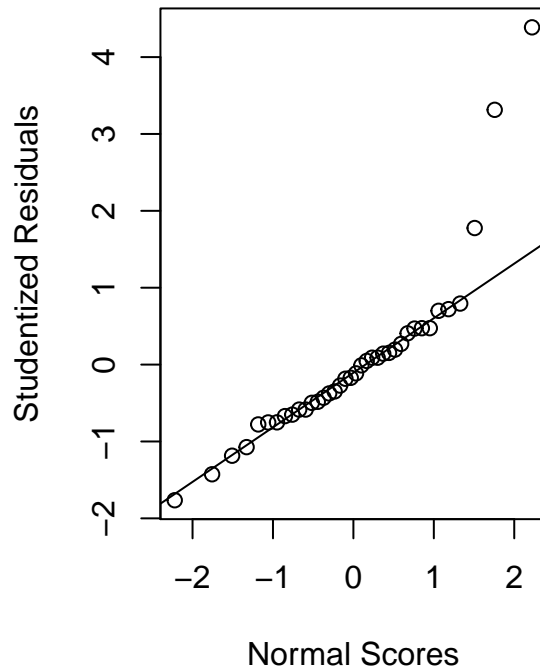
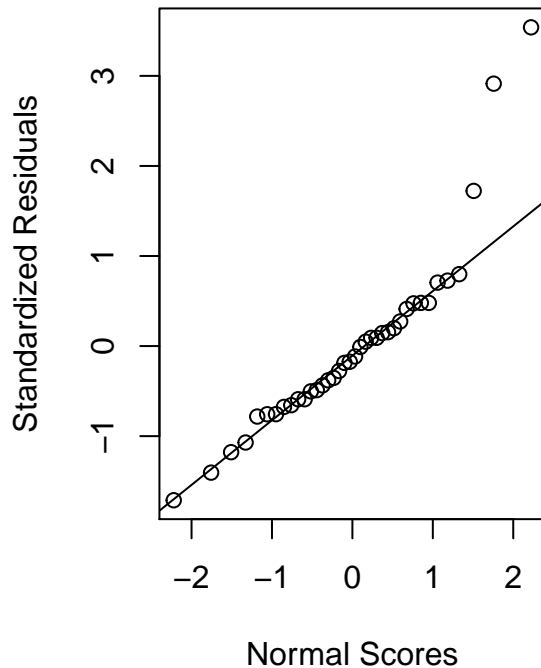
```

```

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

```

## el Original – NPP of Standardized Residuals Original – NPP of Studentized Residuals



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

```
##
## Shapiro-Wilk normality test
##
## data: Lake1c2.step.model0$residuals
## W = 0.86346, p-value = 0.000276
```

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

```
##   edad   Grupo  sexo      Ident Tiempo CargaViral Albumina CD4P Cloro
## 87   35 Treatm.2 Hombre gcarosi 010      0    235891     4.48 19.6  104
##   LogCargaViral CargaViral_t
## 87      12.37113      8181.796
```

```
LAKE1c3 <- LAKE1c2[-which.max(qq02$y),]
```

```
# Fit the full model
```

```
Lake1c3.fm0 <- lm(CargaViral ~ . - LogCargaViral - CargaViral_t - Ident, data = LAKE1c3)
```

```
# Stepwise regression model
```

```
Lake1c3.step.model0 <- stepAIC(Lake1c3.fm0, direction = "both",
                              trace = F)
```

```
(Lake1c3.step.model0s <- summary(Lake1c3.step.model0))
```

```
##
```

```
## Call:
```

```
## lm(formula = CargaViral ~ Grupo + Tiempo + Cloro, data = LAKE1c3)
```

```
##
```

```

## Residuals:
##   Min      1Q  Median      3Q      Max
## -75864 -30134  -4811  19688 150741
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -437829.3  335813.3  -1.304  0.201328
## Grupo(Treatm.1) -13618.4    8450.4  -1.612  0.116579
## Tiempo        -1882.1     465.7  -4.042  0.000299 ***
## Cloro          4897.7     3242.8   1.510  0.140481
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## s: 46850 on 33 degrees of freedom
## Multiple R-squared:  0.3751,
## Adjusted R-squared:  0.3183
## F-statistic: 6.604 on 3 and 33 DF,  p-value: 0.001282

```

```

Lake1c3.SMO.sdres <- rstandard(Lake1c3.step.modelo)
Lake1c3.SMO.studres <- studres(Lake1c3.step.modelo)

```

```
par(mfrow=c(1,2))
```

```

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

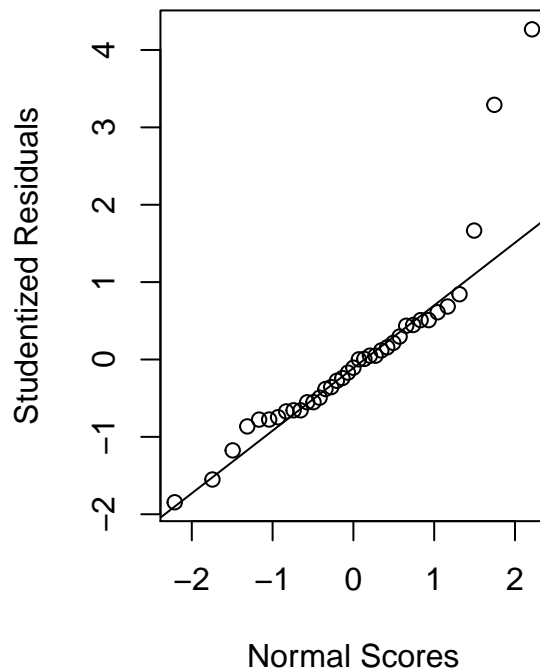
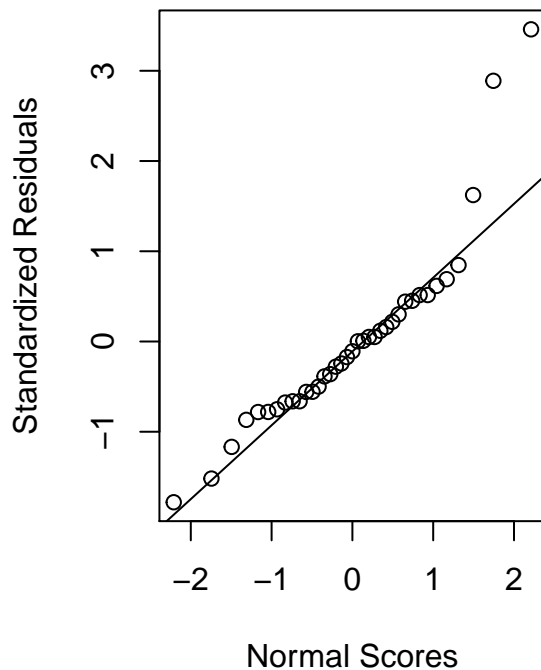
```

```

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

```

## el Original – NPP of Standardized Residuals Original – NPP of Studentized Residuals



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

```
##
## Shapiro-Wilk normality test
##
## data: Lake1c3.step.model0$residuals
## W = 0.87288, p-value = 0.0005645
```

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

```
##   edad   Grupo  sexo      Ident Tiempo CargaViral Albumina CD4P Cloro
## 87   35 Treatm.2 Hombre gcarosi 010      0    235891     4.48 19.6  104
##   LogCargaViral CargaViral_t
## 87      12.37113      8181.796
```

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

```
LAKE1_test <- LAKE1
LAKE1_test$nusuario <- LAKE1_noNA$nusuario
table(LAKE1_test$nusuario)
```

```
##
## aocampo gcarosi
##      20      20
```



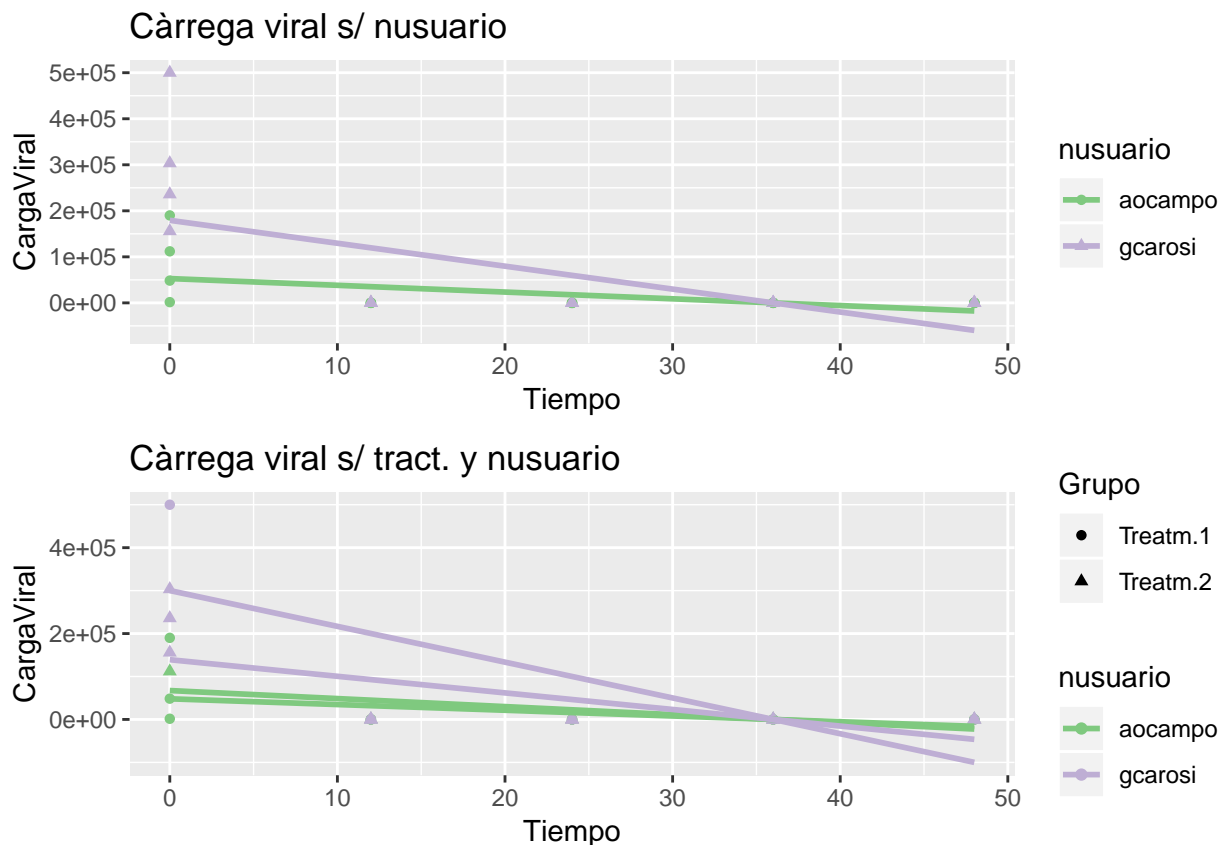
```

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

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

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

```



```

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

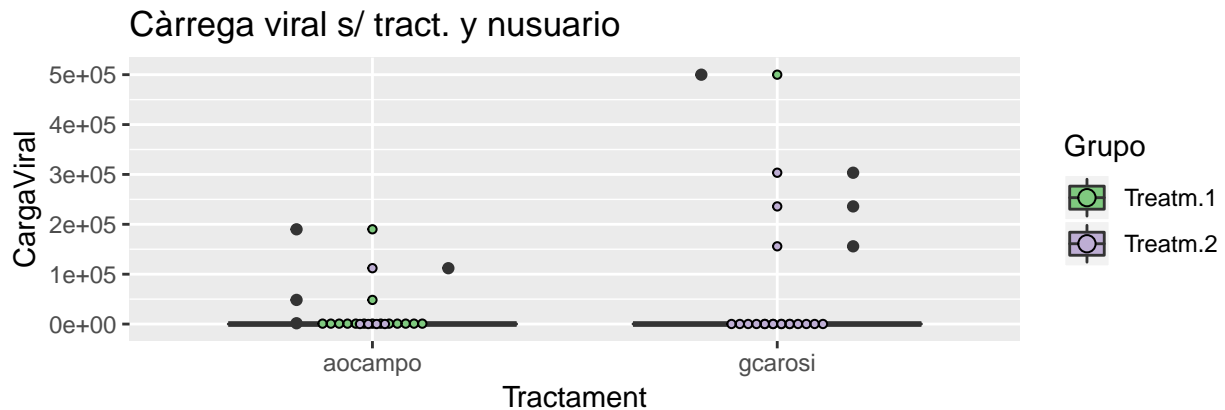
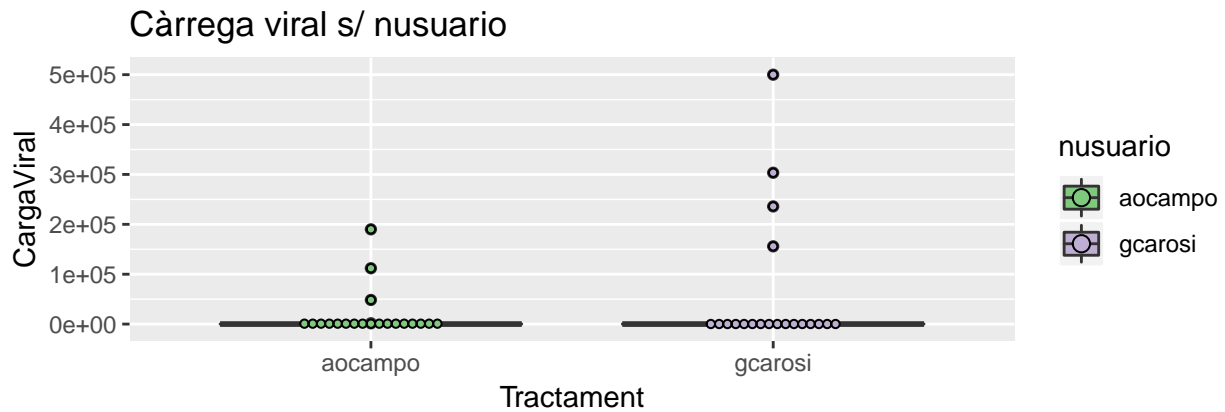
Lake1_testBPlot.b <- ggplot(LAKE1_test, aes(x=nusuario, y=CargaViral, fill=Grupo)) +
  geom_boxplot()+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1)+

```

```
labs(title="Càrrega viral s/ tract. y nusuario",
      x="Tractament", y = "CargaViral")+
scale_fill_brewer(palette="Accent")
```

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

```
## 'stat_bindot()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bindot()' using 'bins = 30'. Pick better value with 'binwidth'.
```



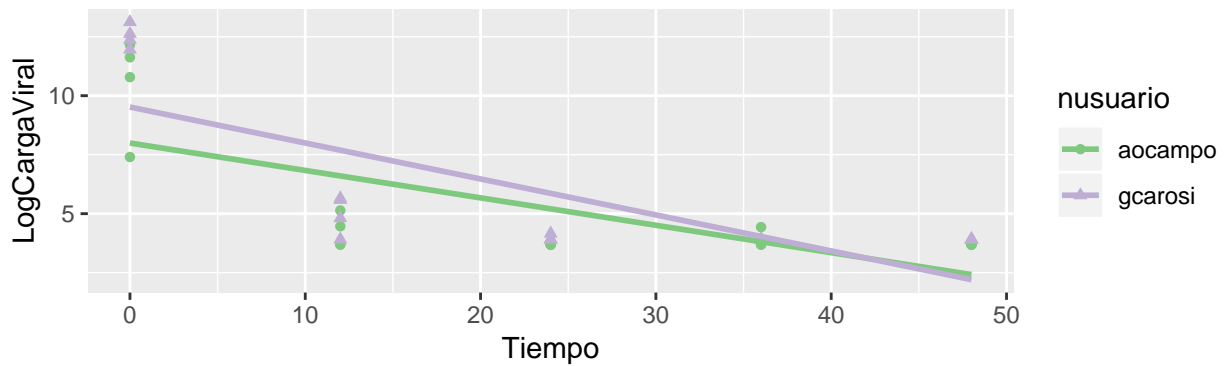
Log:

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

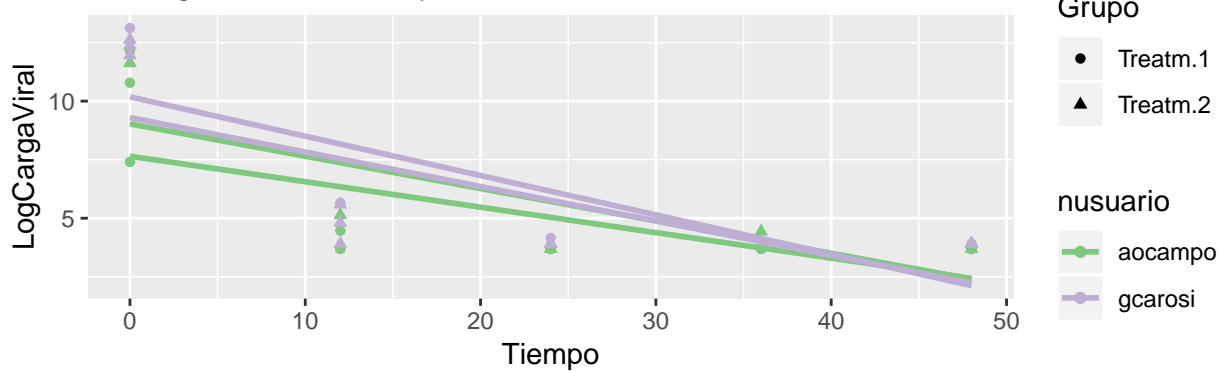
```
Lake1_testSCPlot.b <- ggplot(LAKE1_test, aes(x=Tiempo, y=LogCargaViral, color=nusuario, shape=Grupo)) +
  geom_point()+
  geom_smooth(method=lm,se=F)+
  labs(title="Càrrega viral s/ tract. y nusuario",
        x="Tiempo", y = "LogCargaViral")+
  scale_color_brewer(palette="Accent")
```

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

Càrrega viral s/ nusuario



Càrrega viral s/ tract. y nusuario



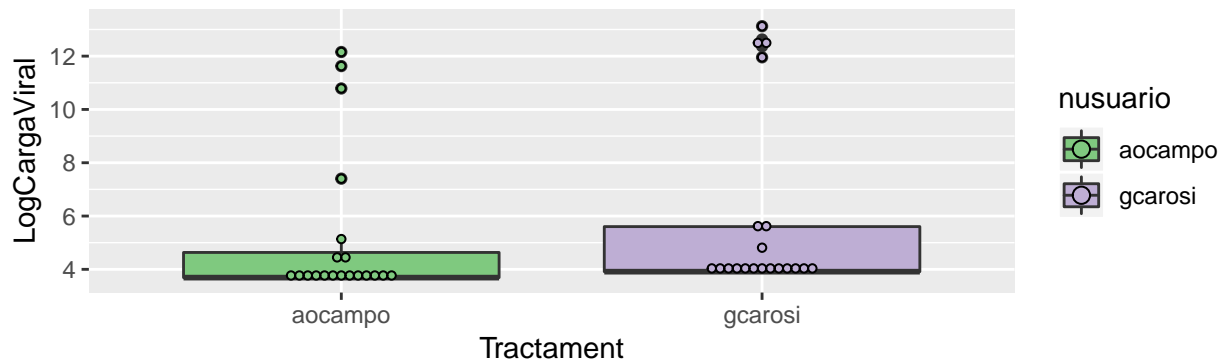
```
Lake1_testBPlot.a <- ggplot(LAKE1_test, aes(x=nusuario, y=LogCargaViral, fill=nusuario)) +
  geom_boxplot()+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1)+
  labs(title="Càrrega viral s/ nusuario",
        x="Tractament", y = "LogCargaViral")+
  scale_fill_brewer(palette="Accent")

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

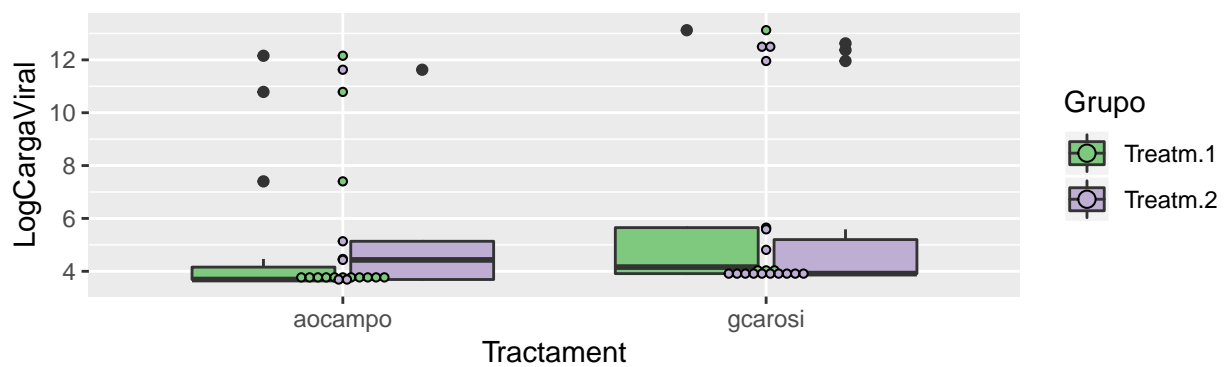
grid.arrange(Lake1_testBPlot.a,Lake1_testBPlot.b)
```

```
## 'stat_bindot()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bindot()' using 'bins = 30'. Pick better value with 'binwidth'.
```

### Càrrega viral s/ nusuariu



### Càrrega viral s/ tract. y nusuariu



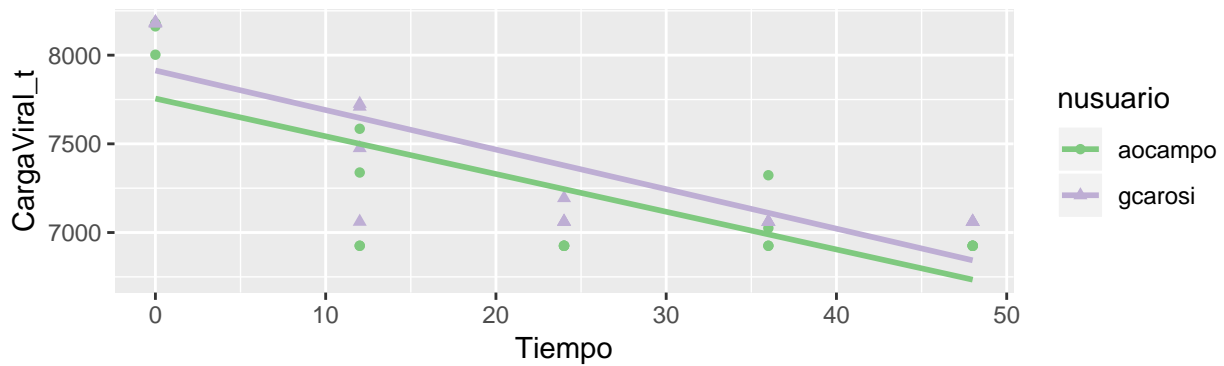
### Box-Cox

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

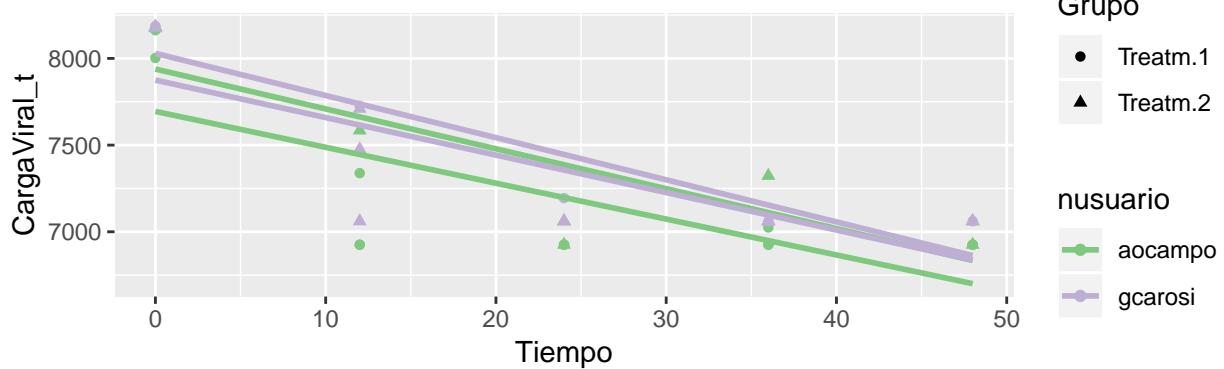
Lake1_testSCPlot.b <- ggplot(LAKE1_test, aes(x=Tiempo, y=CargaViral_t, color=nusuariu, shape=Grupo)) +
  geom_point()+
  geom_smooth(method=lm,se=F)+
  labs(title="Càrrega viral s/ tract. y nusuariu",
        x="Tiempo", y = "CargaViral_t")+
  scale_color_brewer(palette="Accent")

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

Càrrega viral s/ nusuario



Càrrega viral s/ tract. y nusuario



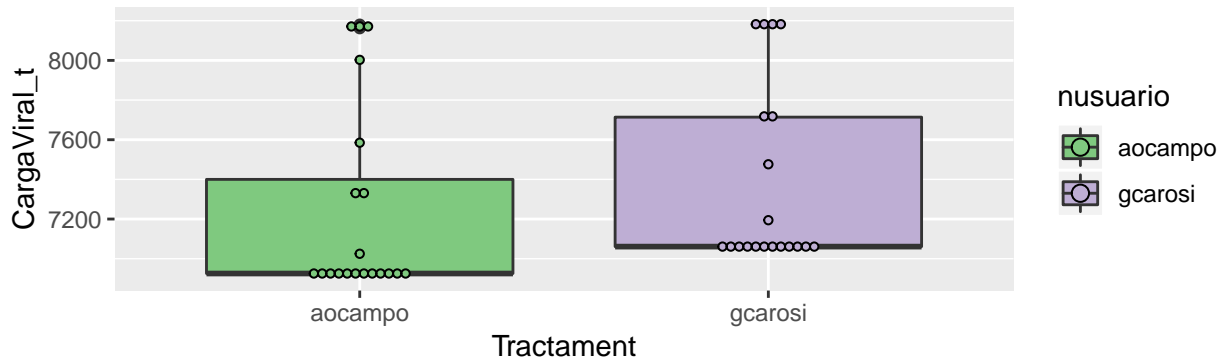
```
Lake1_testBPlot.a <- ggplot(LAKE1_test, aes(x=nusuario, y=CargaViral_t, fill=nusuario)) +
  geom_boxplot()+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1)+
  labs(title="Càrrega viral s/ nusuario",
        x="Tractament", y = "CargaViral_t")+
  scale_fill_brewer(palette="Accent")

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

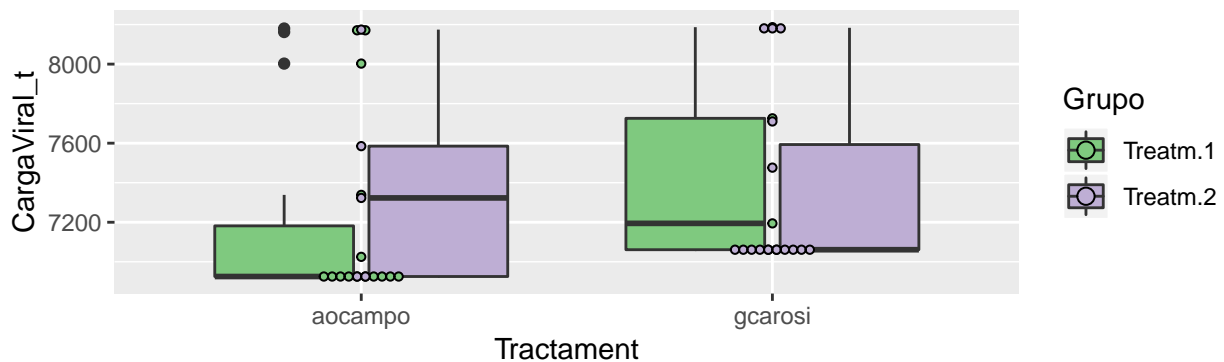
grid.arrange(Lake1_testBPlot.a,Lake1_testBPlot.b)
```

```
## 'stat_bindot()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bindot()' using 'bins = 30'. Pick better value with 'binwidth'.
```

### Càrrega viral s/ nusuario



### Càrrega viral s/ tract. y nusuario



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

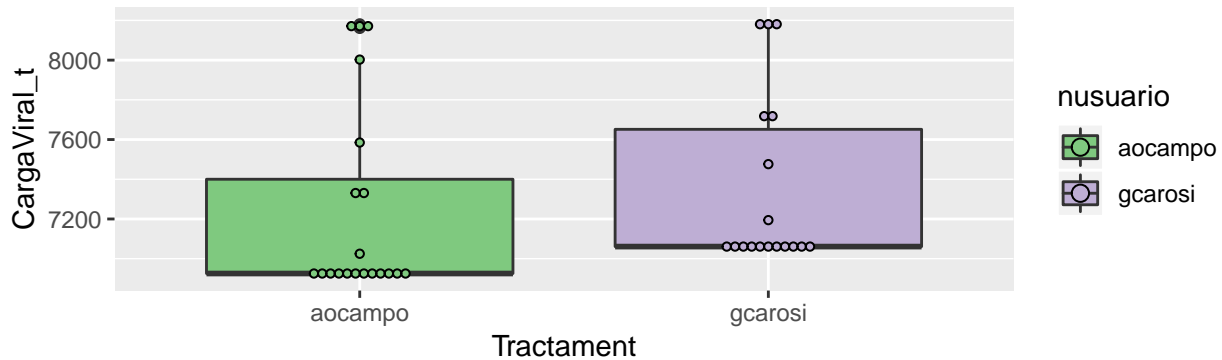
Lake1c2_testBPlot.a <- ggplot(LAKE1c2_test, aes(x=nusuario, y=CargaViral_t, fill=nusuario)) +
  geom_boxplot()+
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1)+
  labs(title="Càrrega viral s/ nusuario (- 2 outl.)",
        x="Tractament", y = "CargaViral_t")+
  scale_fill_brewer(palette="Accent")

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

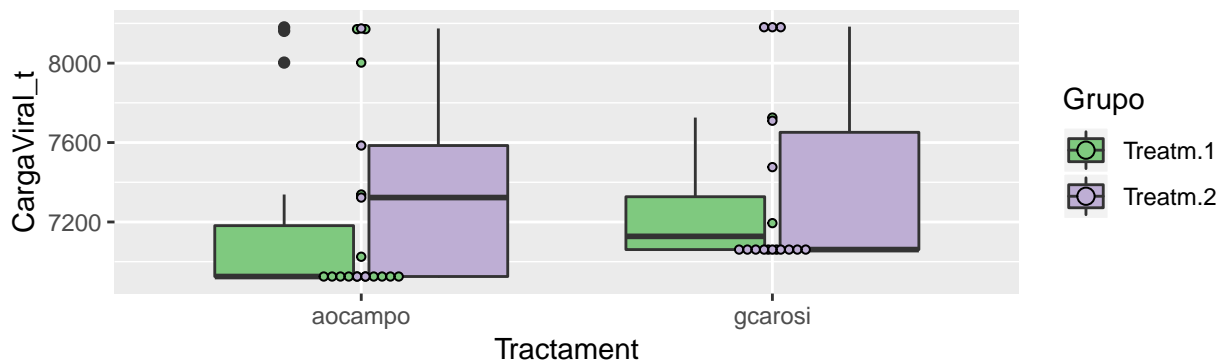
grid.arrange(Lake1c2_testBPlot.a,Lake1c2_testBPlot.b)

## 'stat_bindot()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bindot()' using 'bins = 30'. Pick better value with 'binwidth'.
```

Càrrega viral s/ nusuario (- 2 outl.)



Càrrega viral s/ tract. y nusuario (- 2 outl.)



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

- Pels resultats sembla que **gcarosi** és el grup de **nusuario** que té els valors més extremats i això explicaria possiblement perquè els 3 primers valors aberrants que s'eliminarien són d'aquest grup. Possiblement si es seguís eliminant valors d'aquesta categoria en algun moment es trobarien els primers aberrants de l'altre grup de **nusuario**.
- Quan s'avança en l'eliminació de punts del grup més extremat **gcarosi** l'altre grup **aocampo** pren més pes i també canvia la proporció entre tractaments pel primer grup. Tot i així es queda en un model no balancejat de temps i **nusuario** pel que també es perdria certesa.

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

### Colinealitat

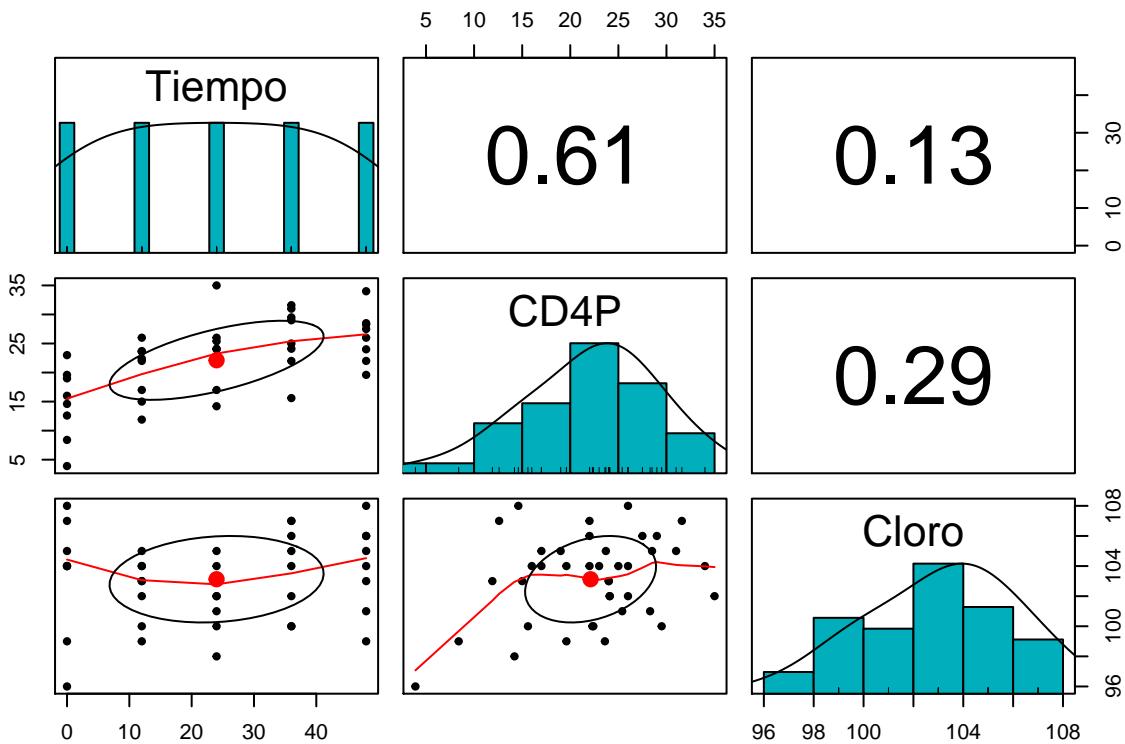
En els models on apareix més d'un predictor es pot donar el cas que algun predictor sigui una combinació lineal dels altres donant lloc a una matriu  $X^T X$  singular o encara que no hi hagi una combinació lineal perfecte una aproximació a la matriu singular. En aquests casos anomenats de multicolinealitat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

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

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

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

```
pairs.panels(LAKE1col,
  method = "pearson", # correlation method
  hist.col = "#00AFBB",
  density = TRUE, # show density plots
  ellipses = TRUE # show correlation ellipses
)
```



```
LAKE1colcor <- as.data.frame(cor(LAKE1col))
LAKE1colcor2 <- LAKE1colcor[which(LAKE1colcor$Tiempo!=1),] #Es mostra el més gran relacionat amb Temps
LAKE1colmaxpred <- rownames(LAKE1colcor2[which.max(abs(LAKE1colcor2$Tiempo)),])
```

Semblaria en aquest primer diagnòstic que la variable amb correlació més gran amb la principal covariant Tiempo seria CD4P.

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

```
(LAKE1col.lms<-summary(LAKE1col.lm <- lm(Tiempo ~ .,data=LAKE1col)))
```

```
##
## Call:
## lm(formula = Tiempo ~ ., data = LAKE1col)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```



```
## -25.132 -10.896 -1.341 9.187 26.683
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20.9199 82.0115 0.255 0.800
## CD4P 1.5745 0.3432 4.587 5.01e-05 ***
## Cloro -0.3077 0.8133 -0.378 0.707
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## s: 13.97 on 37 degrees of freedom
## Multiple R-squared: 0.3735,
## Adjusted R-squared: 0.3396
## F-statistic: 11.03 on 2 and 37 DF, p-value: 0.0001753
```

També ho s'observa en aquest resultat al comprovar els valors de significació dels diferents predictors secundaris *CD4P*, *Cloro* amb p-valors de 1e-04, 0.7073 respectivament.

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

```
x <- model.matrix(Lake1.step.model0)[,-1]
e <- eigen(t(x) %*% x)
e$val
```

```
## [1] 469390.468 11237.683 1009.149
```

```
(k<-sqrt(e$val[1]/e$val)[-1])
```

```
## [1] 6.462919 21.566989
```

Ratios bastant petits i essent per sota de 30 els ratios amb *CD4P*, *Cloro*, sobretot el ratio de *CD4P* amb un ratio de 6.46, indicant possible multicolinealitat.

- Prova de VIF: Segons la fórmula següent es pot veure una mesura de colinealitat en el càlcul de la variància del coeficient del predictor

$$\text{var}\hat{\beta}_j = \sigma^2 \left( \frac{1}{1 - R_j^2} \right) \Delta \frac{1}{\sum_i (x_{ij} - \bar{x}_j)^2}$$

El factor de inflació de variància corresponent a cada predictor  $\frac{1}{1-R_j^2}$  que provoca un augment de la variància del coeficient en cas que sigui gran, és el que indica la possible multicolinealitat del predictor:

```
vif(x)
```

```
## Tiempo CD4P Cloro
## 1.596046 1.714961 1.097428
```

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

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

```
Lake1.step.model0s
```

```
##
## Call:
## lm(formula = CargaViral ~ Tiempo + CD4P + Cloro, data = LAKE1)
```

```

##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -113727  -55558   -7185   42815  276156
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   910289     478303   1.903  0.0650 .
## Tiempo        -1868         958  -1.950  0.0591 .
## CD4P          -4985         2505  -1.990  0.0542 .
## Cloro         -6948         4748  -1.463  0.1521
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## s: 81390 on 36 degrees of freedom
## Multiple R-squared:  0.4226,
## Adjusted R-squared:  0.3745
## F-statistic: 8.783 on 3 and 36 DF,  p-value: 0.0001673
(Lake1.step.model0col.s<-summary(Lake1.step.model0col<-update(Lake1.step.model0,paste(". ~ . -",colnames(LAKE1)))
##
## Call:
## lm(formula = CargaViral ~ Tiempo + Cloro, data = LAKE1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -113986  -59944  -17035   36231  321205
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1090786.9   488052.3   2.235 0.031540 *
## Tiempo      -3015.5     794.8   -3.794 0.000532 ***
## Cloro       -9499.9     4751.5  -1.999 0.052954 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## s: 84580 on 37 degrees of freedom
## Multiple R-squared:  0.3591,
## Adjusted R-squared:  0.3244
## F-statistic: 10.36 on 2 and 37 DF,  p-value: 0.0002667
(Lake1.step.model0col.an<-anova(Lake1.step.model0col,Lake1.step.model0))
## Analysis of Variance Table
##
## Model 1: CargaViral ~ Tiempo + Cloro
## Model 2: CargaViral ~ Tiempo + CD4P + Cloro
##   Res.Df    RSS Df Sum of Sq   F Pr(>F)
## 1      37 2.6469e+11
## 2      36 2.3845e+11  1 2.6233e+10 3.9605 0.05421 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Per un marge molt estret no es consideren significativament diferents els dos models contrastats amb anova amb un p-valor de 0.0542, però sí que es pot veure que en el segon model la variable temps pren més significació

passant de un p-valor de 0.0591 a 5e-04, encara que també es redueix el coeficient de determinació  $R^2$  absolut i ajustat de 0.4226/0.3745 a 0.3591/0.3244, perdent % d'explicabilitat de la variància de la resposta.

S'està doncs en una situació de compromís on no hi ha una prova definitiva entre les fetes que doni una direcció molt clara. Com que l'objectiu és crear un model de predicció es sacrificarà tenir incertesa en els coeficients dels predictors en front de tenir un % d'explicabilitat més gran de la resposta agafant el model amb més variables, encara que hi hagi certa colinearitat present entre *Tiempo*, *CD4P*, *Cloro*. En tot cas en les properes passes s'ha de tenir en compte com una opció la d'eliminar aquest predictor amb colinearitat en cas que es compensi amb altres estructures més eficients.

## COMPROVACIÓ INTERACCIONS MODEL

### Paralelisme o Interacció Individu-Temps

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

*#Model original*

```
Lake1.step.model0Is <- summary(Lake1.step.model0I<-update(Lake1.step.model0,. ~ . + Ident*Tiempo)) #S'a
(Lake1.step.model0I.an<-Anova(Lake1.step.model0I))
```

```
## Anova Table (Type II tests)
```

```
##
```

```
## Response: CargaViral
```

```
##          Sum Sq Df F value    Pr(>F)
## Tiempo      3.7998e+09  1  0.7374 0.3997562
## CD4P         7.5535e+10  1 14.6586 0.0009152 ***
## Cloro        1.8871e+07  1  0.0037 0.9522913
## Ident        7.9799e+10  7  2.2123 0.0731654 .
## Tiempo:Ident 4.5289e+10  7  1.2556 0.3165258
## Residuals    1.1336e+11 22
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
##          Df    Sum Sq Mean Sq F value    Pr(>F)
## Tiempo      1 1.197e+11 1.197e+11  23.227 8.16e-05 ***
## CD4P         1 4.065e+10 4.065e+10   7.889  0.0102 *
## Cloro        1 1.418e+10 1.418e+10   2.752  0.1113
## Ident        7 7.980e+10 1.140e+10   2.212  0.0732 .
## Tiempo:Ident  7 4.529e+10 6.470e+09   1.256  0.3165
## Residuals   22 1.134e+11 5.153e+09
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

*#Model amb correcció per colinearitat*

```
Lake1.step.model0colIs <- summary(Lake1.step.model0colI<-update(Lake1.step.model0col,. ~ . + Ident*Tiempo)
(Lake1.step.model0colI.an<-Anova(Lake1.step.model0colI))
```

```
## Anova Table (Type II tests)
```

```
##
```

```
## Response: CargaViral
```

```
##          Sum Sq Df F value Pr(>F)
```

```
## Tiempo      1.0224e+11  1 12.4481 0.0018 **
## Cloro       7.1571e+07   1  0.0087 0.9264
## Ident       1.6199e+10   7  0.2818 0.9547
## Tiempo:Ident 5.9587e+10   7  1.0365 0.4335
## Residuals   1.8890e+11  23
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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

```
##           Df    Sum Sq  Mean Sq F value    Pr(>F)
## Tiempo      1 1.197e+11 1.197e+11  14.573 0.000884 ***
## Cloro       1 2.860e+10 2.860e+10   3.482 0.074851 .
## Ident       7 1.620e+10 2.314e+09   0.282 0.954660
## Tiempo:Ident 7 5.959e+10 8.512e+09   1.036 0.433518
## Residuals  23 1.889e+11 8.213e+09
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

#### *#Model Log*

```
Lake1.step.model1Is <- summary(Lake1.step.model1I<-update(Lake1.step.model1,. ~ . + Ident*Tiempo)) #S'a
(Lake1.step.model1I.an<-Anova(Lake1.step.model1I))
```

```
## Anova Table (Type II tests)
##
## Response: LogCargaViral
##           Sum Sq Df F value    Pr(>F)
## Tiempo      207.746  1 31.2398 9.431e-06 ***
## Ident       10.088  7  0.2167  0.9779
## Tiempo:Ident 10.641  7  0.2286  0.9744
## Residuals   159.601 24
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Tiempo      1 207.75  207.75  31.240 9.43e-06 ***
## Ident       7  10.09   1.44  0.217  0.978
## Tiempo:Ident 7  10.64   1.52  0.229  0.974
## Residuals  24 159.60   6.65
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

#### *#Model Box-Cox*

```
Lake1.step.model2Is <- summary(Lake1.step.model2I<-update(Lake1.step.model2,. ~ . + Ident*Tiempo)) #S'a
(Lake1.step.model2I.an<-Anova(Lake1.step.model2I))
```

```
## Anova Table (Type II tests)
##
## Response: CargaViral_t
##           Sum Sq Df F value    Pr(>F)
## Tiempo     5466812  1 46.1687 4.997e-07 ***
## Ident      438041  7  0.5285  0.8042
## Tiempo:Ident 60066  7  0.0725  0.9992
## Residuals  2841826 24
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
summary(aov(formula(paste(Lake1.step.model2I$call[2])),data=LAKE1))
```

```
##           Df Sum Sq Mean Sq F value Pr(>F)
## Tiempo      1 5466812 5466812  46.169 5e-07 ***
## Ident       7  438041   62577   0.528  0.804
## Tiempo:Ident 7   60066    8581   0.072  0.999
## Residuals   24 2841826  118409
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Es considera els resultats de p-valor del factor interacció que inferiors al 5% de significància com a significativa la variació de pendent per individu. En aquest cas els resultats dels p-valors dels diferents models Original, Original corregit per multicolinealitat, Log transformat i amb transformació Box-Cox són de 0.3165, 0.4335, 0.9744 i 0.9992 respectivament.

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

## Altres Interaccions

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

```
#Original
Inter.0 <- attr(Lake1.step.model0s$terms,"term.labels")
Lake1.step.model0int.s <- summary(Lake1.step.model0int <- update(Lake1.step.model0,paste(". ~ . +",paste(Lake1.step.model0int2.s2 <- summary(Lake1.step.model0int2 <- stepAIC(Lake1.step.model0int, direction =

##
## Call:
## lm(formula = CargaViral ~ Tiempo + CD4P + Cloro + Tiempo:CD4P +
##     Tiempo:Cloro + CD4P:Cloro + Tiempo:CD4P:Cloro, data = LAKE1)
##
## Residuals:
##   Min     1Q Median     3Q    Max
## -73263 -18903  -1618  24695 147784
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6447992.94 1022547.10   6.306 4.49e-07 ***
## Tiempo       -176830.68   58543.48  -3.021 0.00493 **
## CD4P         -343452.68   74069.49  -4.637 5.69e-05 ***
## Cloro        -60755.43   10234.76  -5.936 1.31e-06 ***
## Tiempo:CD4P    9148.16    2830.75   3.232 0.00285 **
## Tiempo:Cloro  1649.19     574.98   2.868 0.00725 **
## CD4P:Cloro    3282.16     732.55   4.480 8.93e-05 ***
## Tiempo:CD4P:Cloro -86.72     27.74  -3.127 0.00375 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## s: 47430 on 32 degrees of freedom
## Multiple R-squared:  0.8257,
## Adjusted R-squared:  0.7875
```

```
## F-statistic: 21.65 on 7 and 32 DF, p-value: 1.856e-10
```

```
(Lake1.step.model0int2.an2<-Anova(Lake1.step.model0int2))
```

```
## Anova Table (Type II tests)
```

```
##
```

```
## Response: CargaViral
```

```
##           Sum Sq Df F value    Pr(>F)
## Tiempo      5.7912e+10  1 25.7394 1.605e-05 ***
## CD4P         1.8873e+10  1  8.3881 0.006757 **
## Cloro        2.1648e+10  1  9.6215 0.003998 **
## Tiempo:CD4P  2.8046e+10  1 12.4651 0.001282 **
## Tiempo:Cloro 4.6017e+06  1  0.0020 0.964209
## CD4P:Cloro   2.3260e+10  1 10.3379 0.002975 **
## Tiempo:CD4P:Cloro 2.1994e+10  1  9.7751 0.003751 **
## Residuals    7.1999e+10 32
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Lake1.step.model0int2.an1_2<-anova(Lake1.step.model0int2,Lake1.step.model0))
```

```
## Analysis of Variance Table
```

```
##
```

```
## Model 1: CargaViral ~ Tiempo + CD4P + Cloro + Tiempo:CD4P + Tiempo:Cloro +
```

```
##   CD4P:Cloro + Tiempo:CD4P:Cloro
```

```
## Model 2: CargaViral ~ Tiempo + CD4P + Cloro
```

```
##   Res.Df      RSS Df Sum of Sq    F    Pr(>F)
```

```
## 1      32 7.1999e+10
```

```
## 2      36 2.3845e+11 -4 -1.6645e+11 18.495 5.808e-08 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
if (Lake1.step.model0int2.an1_2$`Pr(>F)`[2]>0.05){
```

```
  intM0 <- Lake1.step.model0
```

```
  intM0s <- Lake1.step.model0s
```

```
} else {
```

```
  intM0 <- Lake1.step.model0int2
```

```
  intM0s <- Lake1.step.model0int2.s2
```

```
}
```

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

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

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

```
#Log
```

```
#Step1
```

```
Lake1.step.model1int.s <- summary(Lake1.step.model1int <- update(Lake1.step.model1,paste(".", ~ ., "+", paste
```

```
(Lake1.step.model1int2.s2 <- summary(Lake1.step.model1int2 <- stepAIC(Lake1.step.model1int, direction =
```

```
##
```

```
## Call:
```

```
## lm(formula = LogCargaViral ~ Tiempo + CD4P + Tiempo:CD4P, data = LAKE1)
```

```
##
```

```

## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.7900 -1.0384 -0.2722  0.7064  4.2465
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.642765   1.468711   9.289 4.29e-11 ***
## Tiempo      -0.350318   0.069405  -5.047 1.30e-05 ***
## CD4P         -0.272305   0.078710  -3.460 0.00141 **
## Tiempo:CD4P  0.010529   0.002956   3.562 0.00106 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## s: 1.89 on 36 degrees of freedom
## Multiple R-squared:  0.6687,
## Adjusted R-squared:  0.6411
## F-statistic: 24.22 on 3 and 36 DF,  p-value: 9.352e-09
(Lake1.step.modellint2.an2<-Anova(Lake1.step.modellint2))

## Anova Table (Type II tests)
##
## Response: LogCargaViral
##              Sum Sq Df F value    Pr(>F)
## Tiempo        97.671  1 27.3491 7.435e-06 ***
## CD4P           6.459  1  1.8086 0.187087
## Tiempo:CD4P   45.306  1 12.6862 0.001059 **
## Residuals    128.566 36
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Lake1.step.modellint2.an1_2 <- anova(Lake1.step.modellint2,Lake1.step.model1))

## Analysis of Variance Table
##
## Model 1: LogCargaViral ~ Tiempo + CD4P + Tiempo:CD4P
## Model 2: LogCargaViral ~ Tiempo
##      Res.Df    RSS Df Sum of Sq    F    Pr(>F)
## 1         36 128.57
## 2         38 180.33 -2   -51.765 7.2474 0.002265 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
if (Lake1.step.modellint2.an1_2$`Pr(>F)`[2]>0.05){
  intM1 <- Lake1.step.model1
  intM1s <- Lake1.step.model1s
} else {
  intM1 <- Lake1.step.modellint2
  intM1s <- Lake1.step.modellint2.s2
}

##Box-Cox
##Step1
Lake1.step.model2int.s <- summary(Lake1.step.model2int <- update(Lake1.step.model2,paste(".", "~ . +",paste
(Lake1.step.model2int2.s2 <- summary(Lake1.step.model2int2 <- stepAIC(Lake1.step.model2int, direction =

```

```

##
## Call:
## lm(formula = CargaViral_t ~ Tiempo + CD4P + Tiempo:CD4P, data = LAKE1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -631.26 -157.19   31.58  187.95  448.34
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8267.5540   215.9970  38.276 < 2e-16 ***
## Tiempo      -47.3143    10.2070  -4.635 4.55e-05 ***
## CD4P        -23.5364    11.5756  -2.033 0.0494 *
## Tiempo:CD4P  1.1663     0.4347   2.683 0.0110 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## s: 277.9 on 36 degrees of freedom
## Multiple R-squared: 0.6843,
## Adjusted R-squared: 0.6579
## F-statistic: 26.01 on 3 and 36 DF,  p-value: 3.979e-09
(Lake1.step.model2int2.an2<-Anova(Lake1.step.model2int2))

## Anova Table (Type II tests)
##
## Response: CargaViral_t
##              Sum Sq Df F value    Pr(>F)
## Tiempo      3308923  1 42.8392 1.307e-07 ***
## CD4P          3352  1  0.0434  0.83615
## Tiempo:CD4P 555922  1  7.1973  0.01096 *
## Residuals  2780659 36
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Lake1.step.model2int2.an1_2 <- anova(Lake1.step.model2int2,Lake1.step.model2))

## Analysis of Variance Table
##
## Model 1: CargaViral_t ~ Tiempo + CD4P + Tiempo:CD4P
## Model 2: CargaViral_t ~ Tiempo
##      Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1         36 2780659
## 2         38 3339934 -2   -559274 3.6203 0.03693 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
if (Lake1.step.model2int2.an1_2$'Pr(>F)')[2]>0.05){
  intM2 <- Lake1.step.model2
  intM2s <- Lake1.step.model2s
} else {
  intM2 <- Lake1.step.model2int2
  intM2s <- Lake1.step.model2int2.s2
}

```



```

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

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

Lake1.it1 <- (cbind(round(intM0s$coefficients[-1,-3],4),
                    intM1s_mod,
                    intM2s_mod))
kable(Lake1.it1) %>%
  kable_styling(bootstrap_options = "striped", full_width = F) %>%
  add_header_above(c(" " = 1, "M.Int sense transf. 1" = 3, "M.Int log transf" = 3, "M.Int Box-Cox t" = 3))

```

	M.Int sense transf. 1			M.Int log transf			M.Int Box-Cox t	
	Estimate	Std. Error	Pr(> t )	Estimate	Std. Error	Pr(> t )	Estimate	Std. Error
Tiempo	-176830.6843	58543.4804	0.0049	-0.3503	0.0694	0.0000	-47.3143	10.2070
CD4P	-343452.6770	74069.4864	0.0001	-0.2723	0.0787	0.0014	-23.5364	11.5756
Cloro	-60755.4316	10234.7569	0.0000	NA	NA	NA	NA	NA
Tiempo:CD4P	9148.1570	2830.7465	0.0028	0.0105	0.0030	0.0011	1.1663	0.4347
Tiempo:Cloro	1649.1938	574.9848	0.0072	NA	NA	NA	NA	NA
CD4P:Cloro	3282.1583	732.5469	0.0001	NA	NA	NA	NA	NA
Tiempo:CD4P:Cloro	-86.7203	27.7370	0.0038	NA	NA	NA	NA	NA

```

Lake1.it2 <- data.frame(Model=c("M.Int sense transf","M.Int log transf","M.Int Box-Cox t"))
Lake1.it2$"Mediana residual" <- c(median(intM0s$residuals),
                                  median(intM1s$residuals),
                                  median(intM2s$residuals))
kable(Lake1.it2) %>%
  kable_styling("striped")

```

Model	Mediana residual
M.Int sense transf	-1617.8894864
M.Int log transf	-0.2722391
M.Int Box-Cox t	31.5771159

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

- Pel model original el model òptim seria  $Carga\ Viral \sim Tiempo + CD4P + Cloro + Tiempo:CD4P + Tiempo:Cloro + CD4P:Cloro + Tiempo:CD4P:Cloro$  i s'ha obtingut un p-valor entre models de 0.
- Pel model original el model òptim seria  $LogCarga\ Viral \sim Tiempo + CD4P + Tiempo:CD4P$  i s'ha obtingut un p-valor entre models de 0.0023.
- Pel model original el model òptim seria  $Carga\ Viral\_t \sim Tiempo + CD4P + Tiempo:CD4P$  i s'ha obtingut un p-valor entre models de 0.0369.

## Diagnòstics models interaccions

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

```
Lake1Int.SMO.sdres <- rstandard(intM0)
Lake1Int.SMO.studres <- studres(intM0)
par(mfrow=c(2,2))

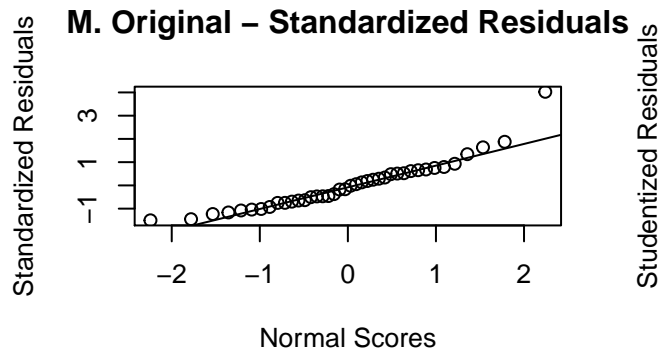
qqnorm(Lake1.SMO.sdres,
       ylab="Standardized Residuals",
       xlab="Normal Scores",
       main="M. Original - Standardized Residuals")
qqline(Lake1.SMO.sdres)

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

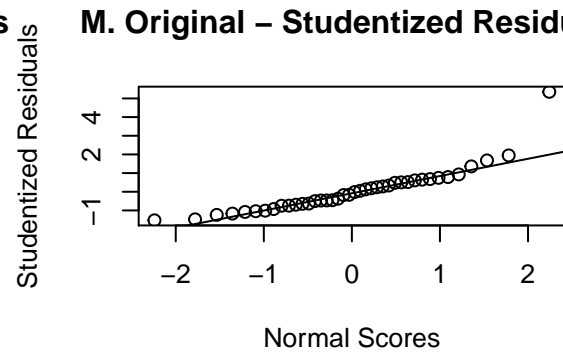
qqnorm(Lake1Int.SMO.sdres,
       ylab="Standardized Residuals",
       xlab="Normal Scores",
       main="M. Original Int. - Standardized Residuals")
qqline(Lake1Int.SMO.sdres)

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

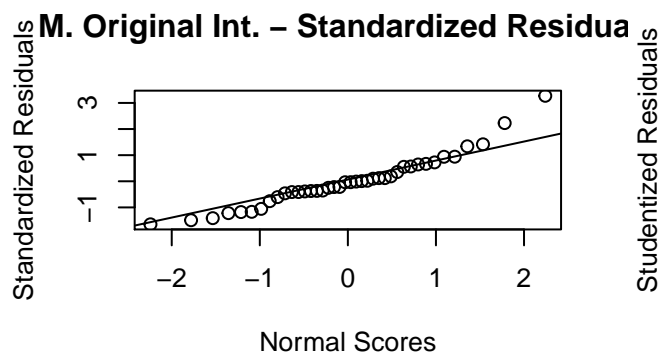
M. Original – Standardized Residuals



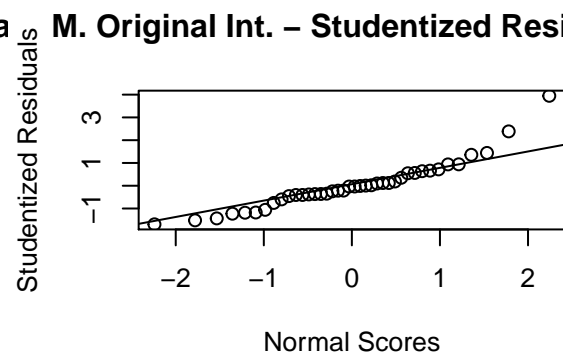
M. Original – Studentized Residuals



M. Original Int. – Standardized Residuals



M. Original Int. – Studentized Residuals



S0

```
##
## Shapiro-Wilk normality test
##
## data: Lake1c3.step.model0$residuals
## W = 0.87288, p-value = 0.0005645
(S0Int <- shapiro.test(intM0$residuals))
```

```
##
## Shapiro-Wilk normality test
##
## data: intM0$residuals
## W = 0.92773, p-value = 0.01359
```

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

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

```

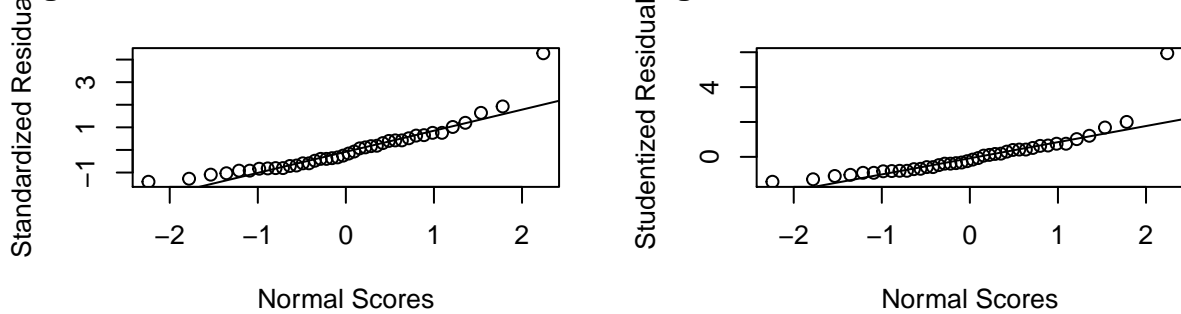
qqnorm(Lake1.SM0col.studres,
       ylab="Studentized Residuals",
       xlab="Normal Scores",
       main="M. Original Correcció Col. - Studentized Residuals")
qqline(Lake1.SM0.studres)

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

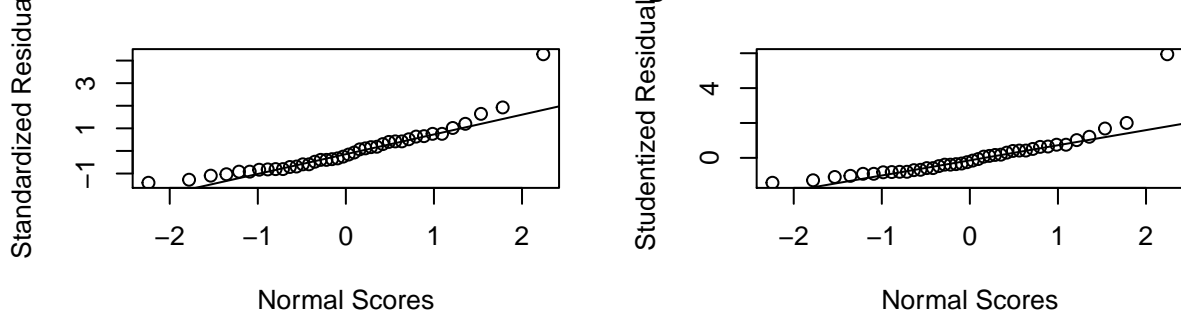
qqnorm(Lake1Int.SM0col.studres,
       ylab="Studentized Residuals",
       xlab="Normal Scores",
       main="M. Original Int. Correcció Col. - Studentized Residuals")
qqline(Lake1Int.SM0col.studres)

```

**Original Correcció Col. – Standardized Residuals**      **Original Correcció Col. – Studentized Residuals**



**Original Int. Correcció Col. – Standardized Residuals**      **Original Int. Correcció Col. – Studentized Residuals**



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

```

##
## Shapiro-Wilk normality test
##
## data: Lake1.step.model0col$residuals
## W = 0.88007, p-value = 0.0005286

```

```

(S0colInt <- shapiro.test(intM0col$residuals))

##
## Shapiro-Wilk normality test
##
## data: intM0col$residuals
## W = 0.88007, p-value = 0.0005286

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

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

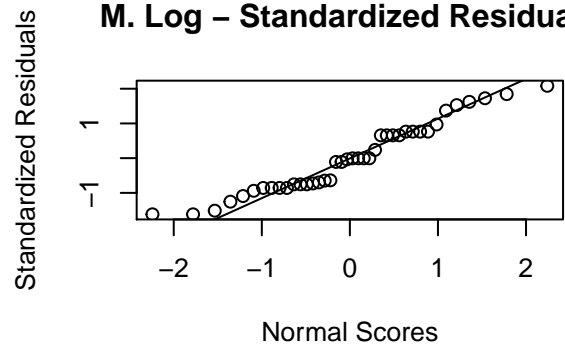
qqnorm(Lake1.SM1.studres,
       ylab="Studentized Residuals",
       xlab="Normal Scores",
       main="M. Log - Studentized Residuals")
qqline(Lake1.SM1.studres)

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

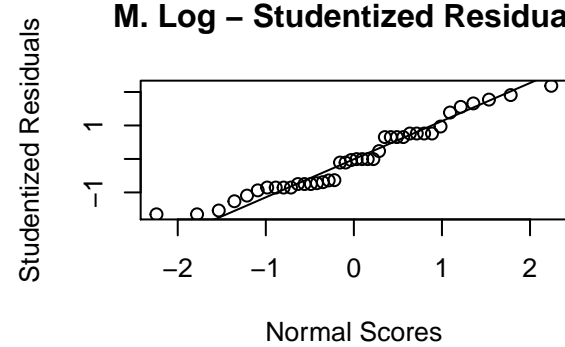
qqnorm(Lake1Int.SM1.studres,
       ylab="Studentized Residuals",
       xlab="Normal Scores",
       main="M. Log Int.- Studentized Residuals")
qqline(Lake1Int.SM1.studres)

```

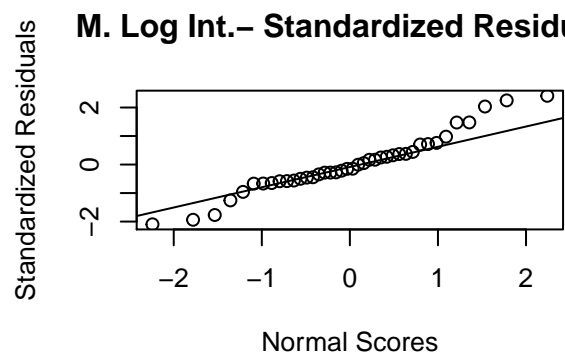
**M. Log – Standardized Residuals**



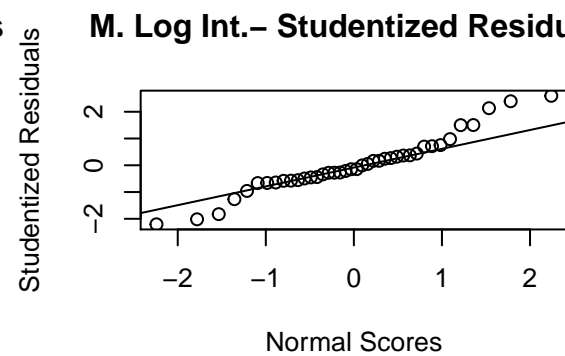
**M. Log – Studentized Residuals**



**M. Log Int.– Standardized Residuals**



**M. Log Int.– Studentized Residuals**



```
S1
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: Lake1.step.model1$residuals  
## W = 0.94994, p-value = 0.07547  
(S1Int <- shapiro.test(intM1$residuals))
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: intM1$residuals  
## W = 0.9557, p-value = 0.1192
```

```
Lake1Int.SM2.sdres <- rstandard(intM2)  
Lake1Int.SM2.studres <- studres(intM2)  
par(mfrow=c(2,2))  
  
qqnorm(Lake1.SM2.sdres,  
       ylab="Standardized Residuals",  
       xlab="Normal Scores",  
       main="M. Box-Cox - Standardized Residuals")  
qqline(Lake1.SM2.sdres)  
  
qqnorm(Lake1.SM2.studres,  
       ylab="Studentized Residuals",
```

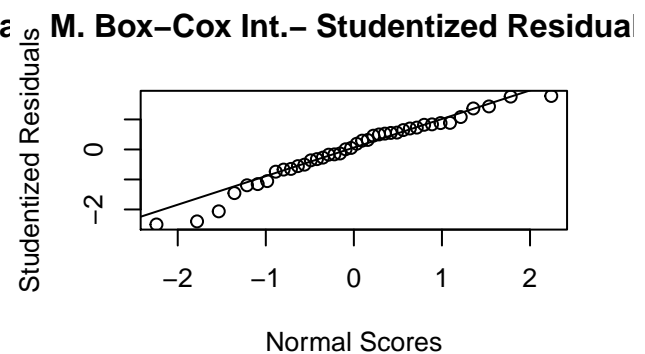
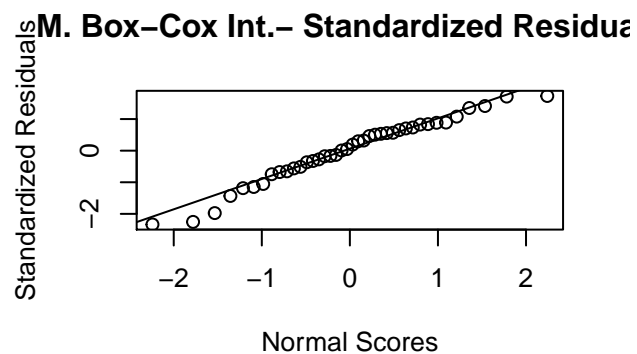
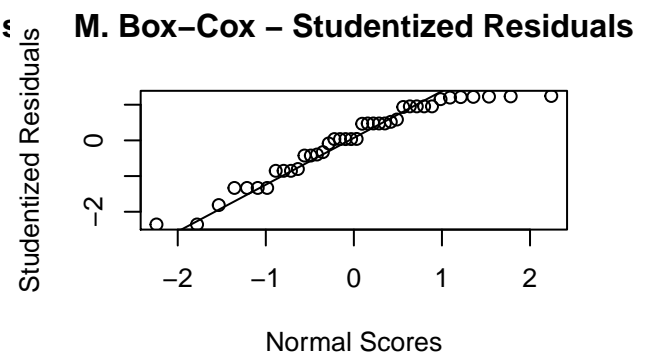
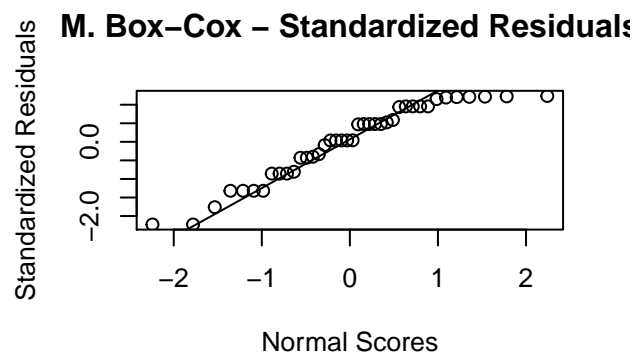
```

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

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

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

```



S2

```

##
## Shapiro-Wilk normality test
##
## data: Lake1.step.model2$residuals
## W = 0.91901, p-value = 0.00717
(S2Int <- shapiro.test(intM2$residuals))

##

```

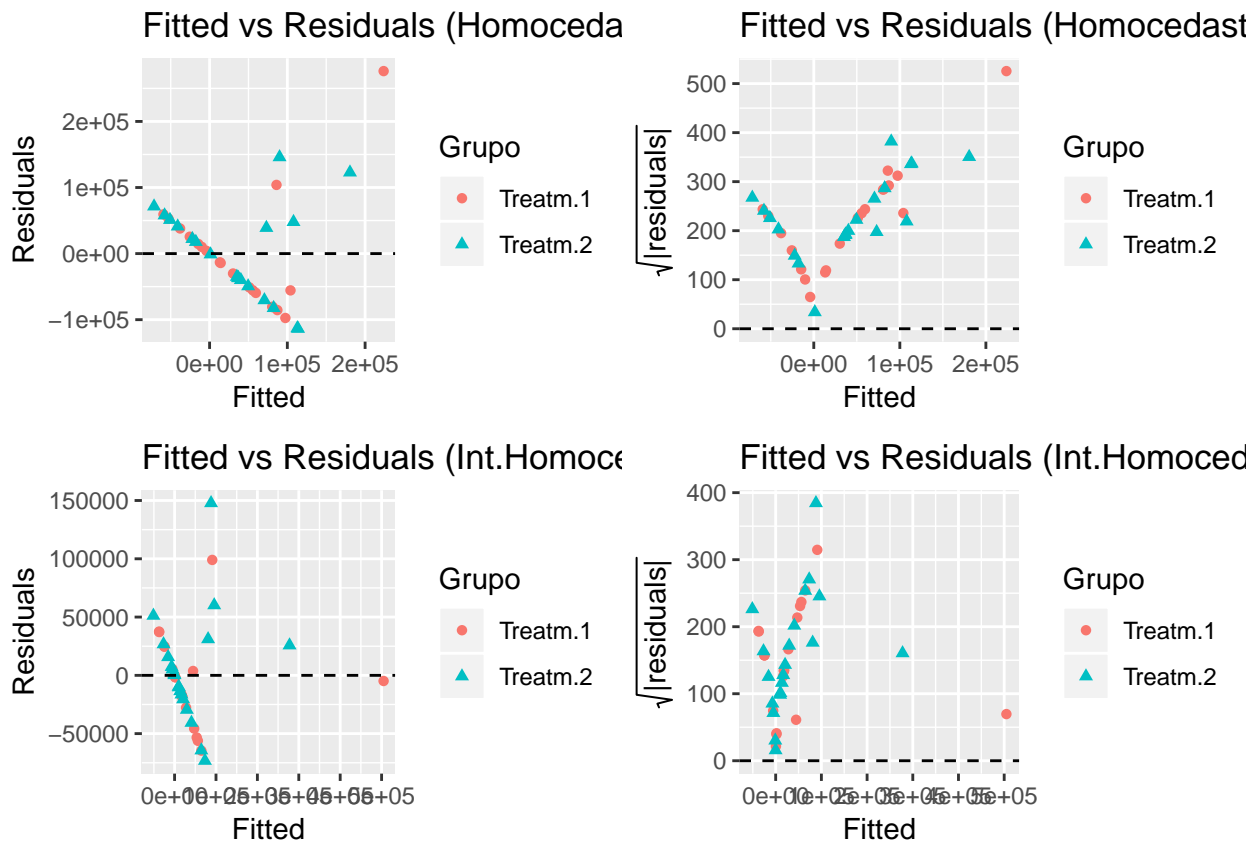
```
## Shapiro-Wilk normality test
##
## data: intM2$residuals
## W = 0.96332, p-value = 0.2172
```

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

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

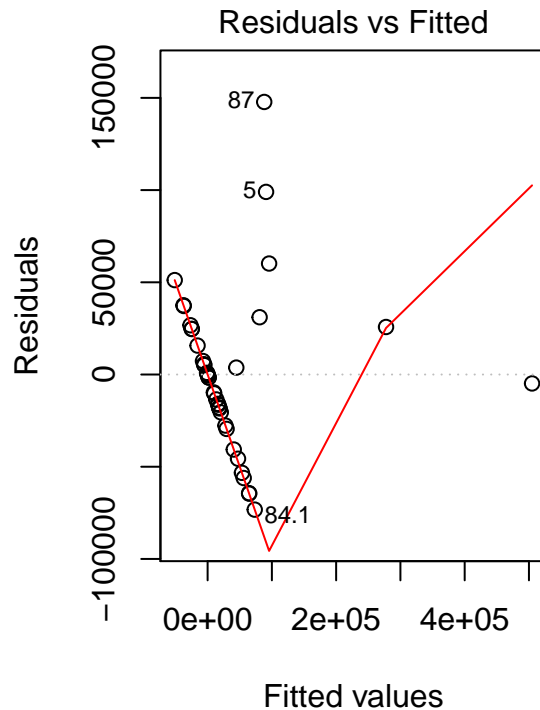
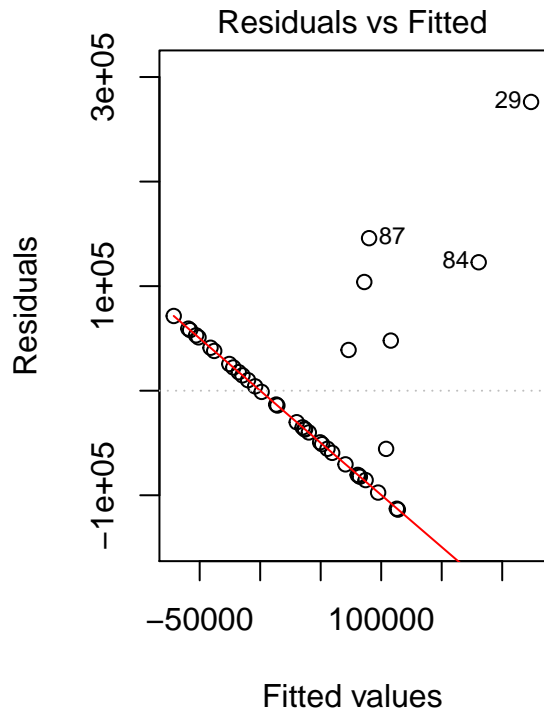
pHInt02 <- ggplot(LAKE1, aes(x=fitted(intM0), y=sqrt(abs(residuals(intM0))),
                             shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Int.Homocedasticity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

grid.arrange(pH01, pH02, pHInt01, pHInt02,ncol=2)
```



```
par(mfrow=c(1,2))
plot(Lake1.step.model0,which=1)
plot(intM0,which=1)
```





BP01

```
##
## studentized Breusch-Pagan test
##
## data: Lake1.step.model0
## BP = 17.345, df = 3, p-value = 6e-04
```

```
(BPInt01<-bptest(intM0))
```

```
##
## studentized Breusch-Pagan test
##
## data: intM0
## BP = 8.9796, df = 7, p-value = 0.2541
```

```
pHco101 <- ggplot(LAKE1, aes(x=fitted(Lake1.step.model0col), y=residuals(Lake1.step.model0col),
                             shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Col.Homocedasticity test)",
        x="Fitted", y = "Residuals")
```

```
pHco102 <-ggplot(LAKE1, aes(x=fitted(Lake1.step.model0col), y=sqrt(abs(residuals(Lake1.step.model0col))),
                             shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Col.Homocedasticity test)",
```

```

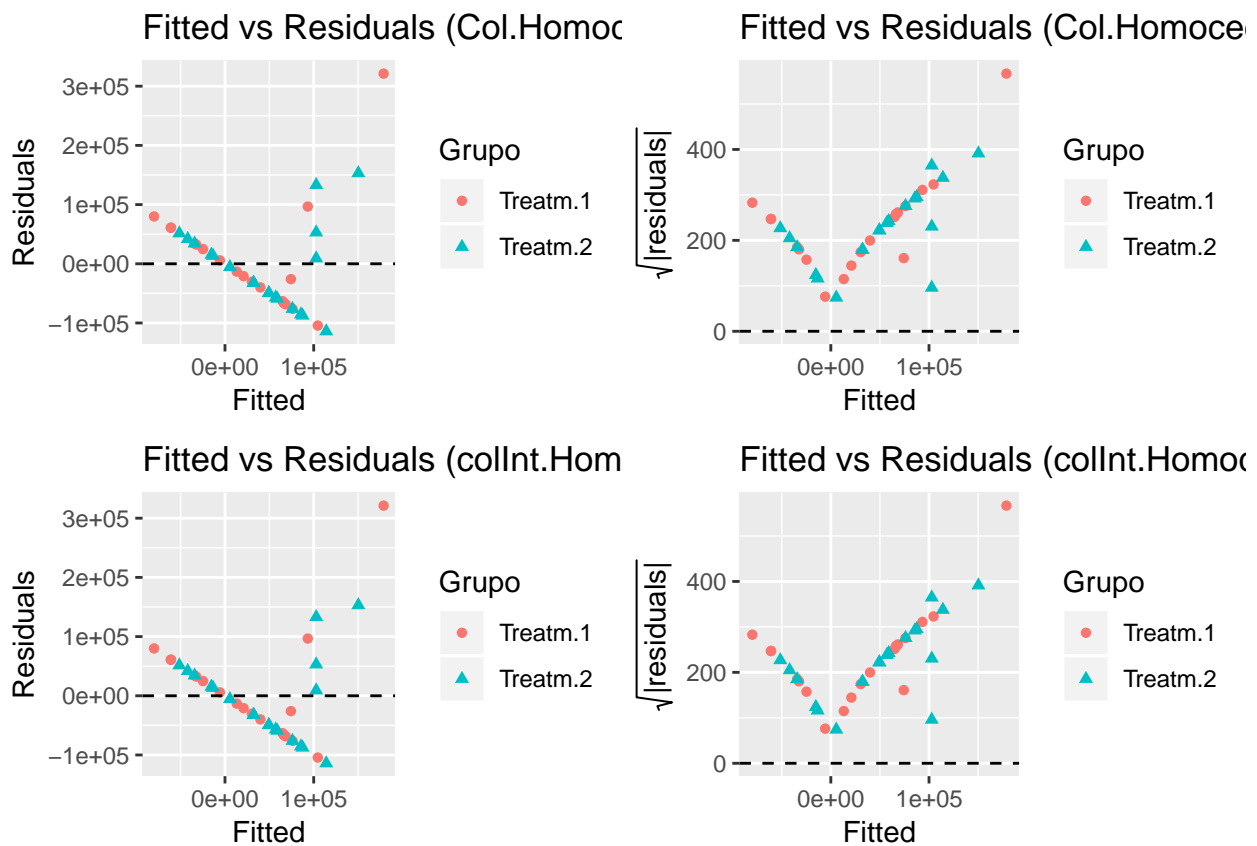
x="Fitted", y = expression(sqrt("|residuals|")))

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

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

grid.arrange(pHcol01, pHcol02, pHcolInt01, pHcolInt02,ncol=2)

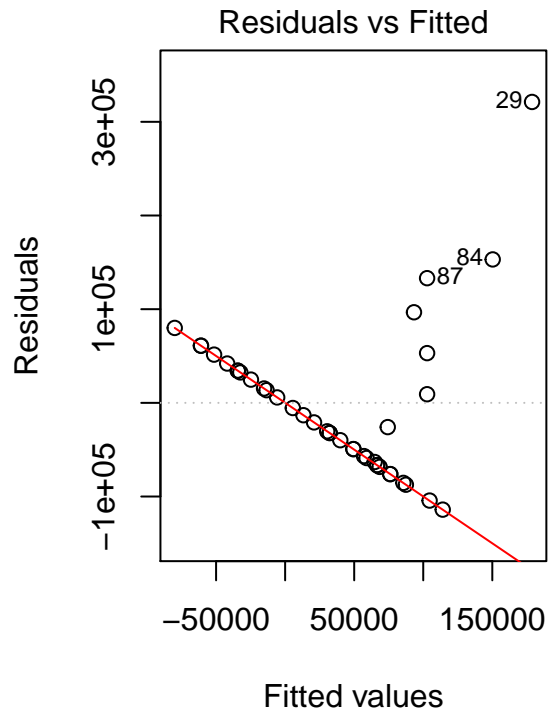
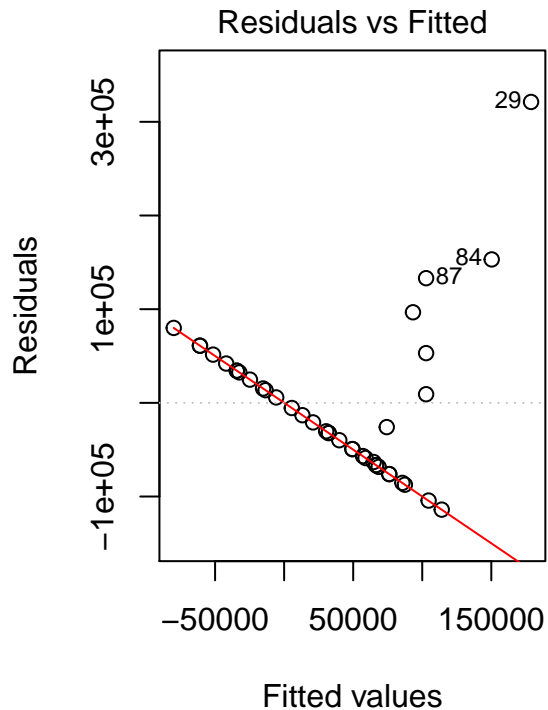
```



```

par(mfrow=c(1,2))
plot(Lake1.step.model0col,which=1)
plot(intM0col,which=1)

```



```
(BPcol01<-bptest(Lake1.step.model0col))
```

```
##
## studentized Breusch-Pagan test
##
## data: Lake1.step.model0col
## BP = 12.423, df = 2, p-value = 0.002006
```

```
(BPcolInt01<-bptest(intM0col))
```

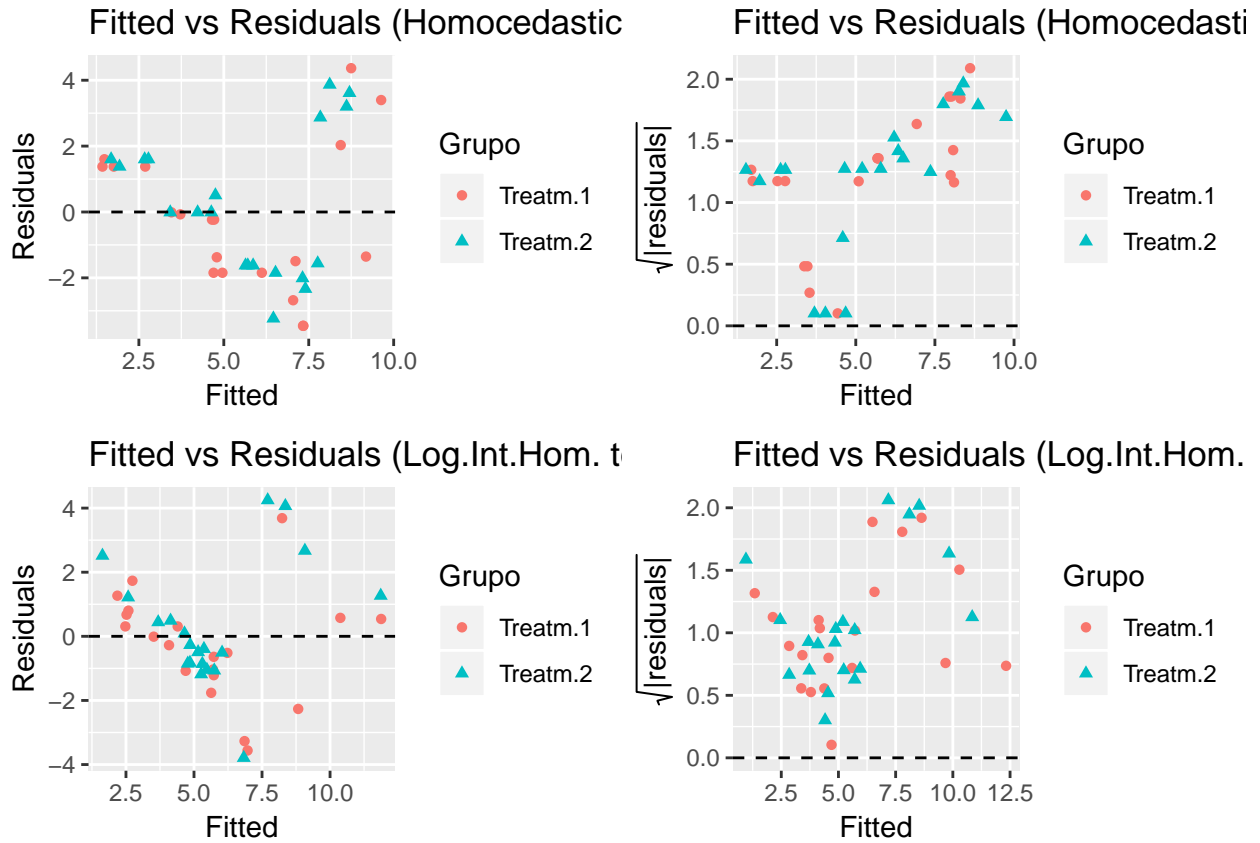
```
##
## studentized Breusch-Pagan test
##
## data: intM0col
## BP = 12.423, df = 2, p-value = 0.002006
```

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

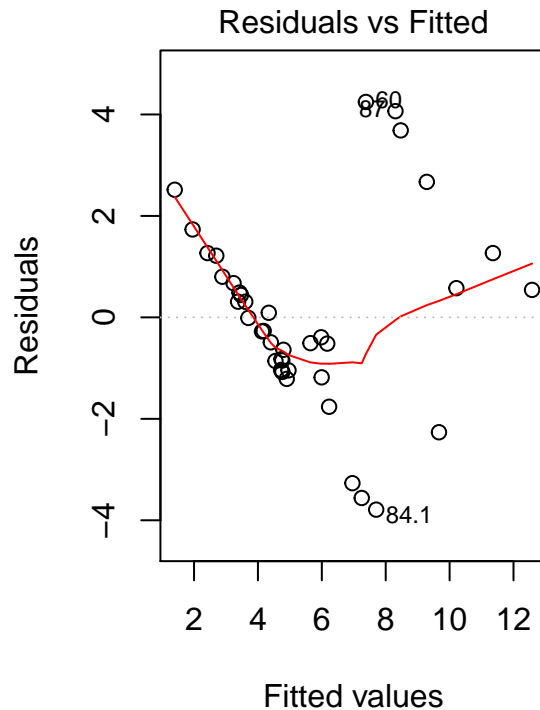
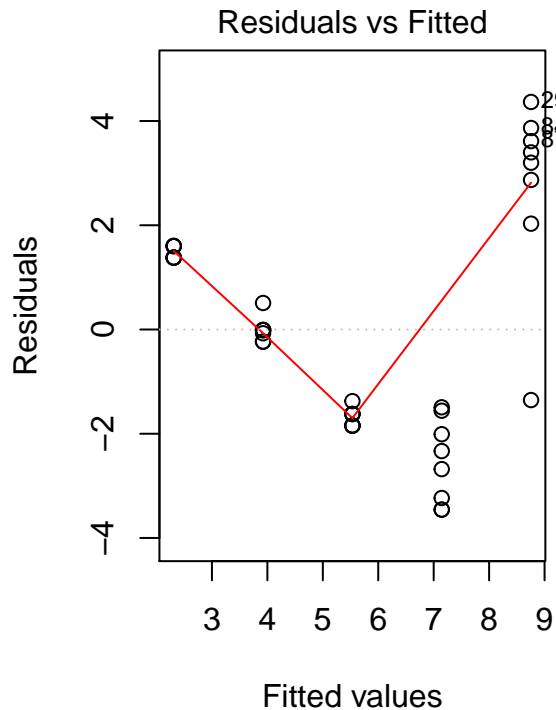
```
pHInt12 <-ggplot(LAKE1, aes(x=fitted(intM1), y=sqrt(abs(residuals(intM1))),
                           shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
```

```
labs(title="Fitted vs Residuals (Log.Int.Hom. test)",
      x="Fitted", y = expression(sqrt("|residuals|")))
```

```
grid.arrange(pH11, pH12,pHInt11,pHInt12,ncol=2)
```



```
plot(Lake1.step.model1,which=1)
plot(intM1,which=1)
```



BP11

```
##
## studentized Breusch-Pagan test
##
## data: Lake1.step.model1
## BP = 19.474, df = 1, p-value = 1.02e-05
```

```
(BPInt11<-bptest(intM1))
```

```
##
## studentized Breusch-Pagan test
##
## data: intM1
## BP = 10.752, df = 3, p-value = 0.01314
```

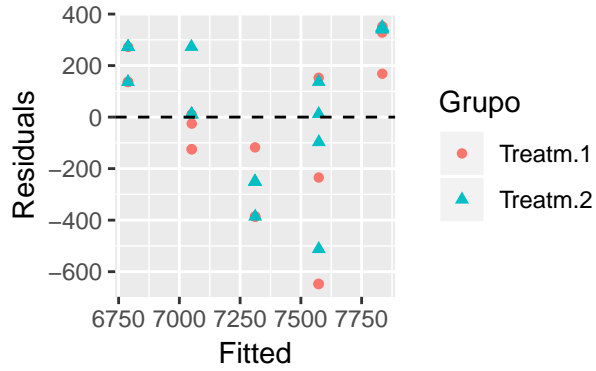
```
pHInt21 <- ggplot(LAKE1, aes(x=fitted(intM2), y=residuals(intM2),
                             shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Int.BC.Hom test)",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

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

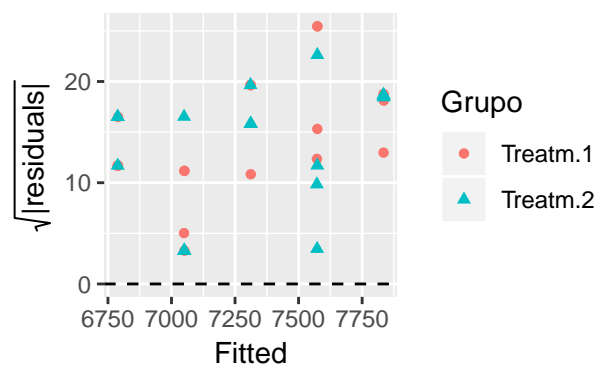
```
x="Fitted", y = expression(sqrt("|residuals|"))
```

```
grid.arrange(pH21, pH22,pHInt21,pHInt22,ncol=2)
```

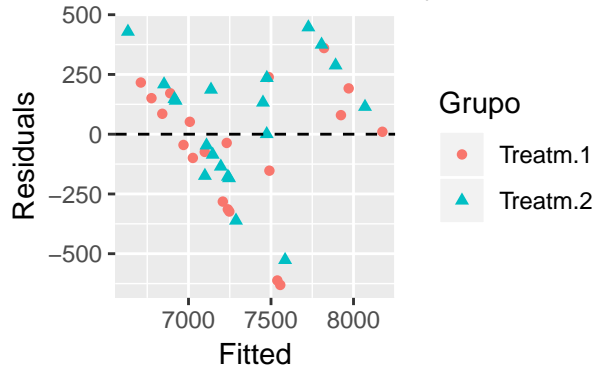
Fitted vs Residuals (Homocedast



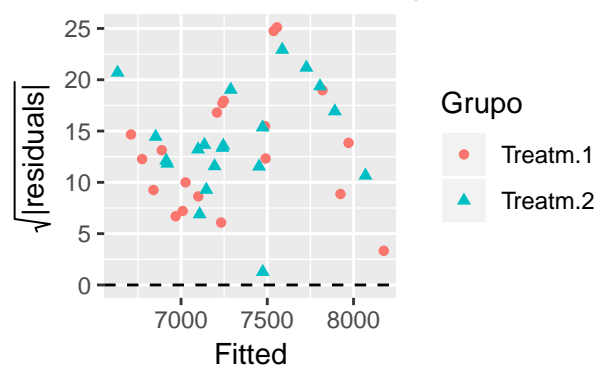
Fitted vs Residuals (Homocedasti



Fitted vs Residuals (Int.BC.Hom t

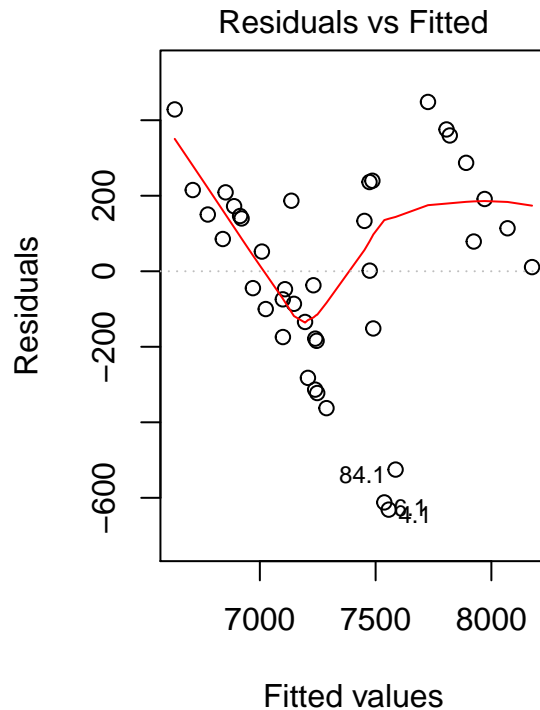
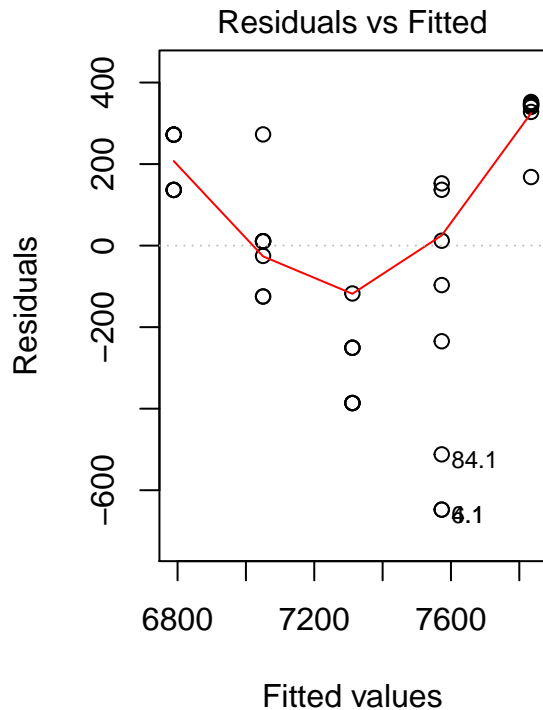


Fitted vs Residuals (Int.BC.Hom te



```
plot(Lake1.step.model2,which=1)
```

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



```
BP21
```

```
##
## studentized Breusch-Pagan test
##
## data: Lake1.step.model2
## BP = 5.718, df = 1, p-value = 0.01679
```

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

```
##
## studentized Breusch-Pagan test
##
## data: intM2
## BP = 5.8317, df = 3, p-value = 0.1201
```

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

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

```
#Es simplifica els noms dels models inicials per poder utilitzar-los més comodament en el codi
M0<-Lake1.step.model0
```

```

M0col<-Lake1.step.model0col
M1<-Lake1.step.model1
M2<-Lake1.step.model2

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

```

## AJUST MODEL EFECTES ALEATORIS

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

### Intercepció Aleatòria

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

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

### Ajust models

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

```

Update_LM_RI <- function(LM,RI_var,df) { #Objecte del tipus LM

  fmOLS <- update(LM$call$formula,paste(" ~ . + Error(",RI_var,""))
  ML_I_RI_sOLS <- summary(ML_I_RI_OLS <- aov(fmOLS, data = df))

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

  ML_I_RI_sMLE <- summary(ML_I_RI_MLE <- lme(fmMLf, random = fmMLr, data = df, method="ML"))
  ML_I_RI_sREML <- summary(ML_I_RI_REML <- lme(fmMLf, random = fmMLr, data = df, method="REML"))

  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 (Within)","MLE I RI","REML I RI",
               "OLS I RI base","MLE I RI base","REML I RI base")

```



```

    return(l)
  }

MO_RI<-Update_LM_RI(MO,"Ident",LAKE1)
eMLE_MO_RI<-exactRLRT(MO_RI_MLE<-lme(MO$call$formula, random = ~1|Ident, data = LAKE1, method="ML"))

## Using restricted likelihood evaluated at ML estimators.
## Refit with method="REML" for exact results.
## Warning in model.matrix.default(~m$groups[[n.levels - i + 1]] - 1,
## contrasts.arg = c("contr.treatment", : non-list contrasts argument ignored)
eREML_MO_RI<-exactRLRT(MO_RI_REML<-lme(MO$call$formula, random = ~1|Ident, data = LAKE1, method="REML"))

## Warning in model.matrix.default(~m$groups[[n.levels - i + 1]] - 1,
## contrasts.arg = c("contr.treatment", : non-list contrasts argument ignored)
intMO_RI<-Update_LM_RI(intMO,"Ident",LAKE1)
eMLE_intMO_RI<-exactRLRT(intMO_RI_MLE<-lme(intMO$call$formula, random = ~1|Ident, data = LAKE1, method=

## Using restricted likelihood evaluated at ML estimators.
## Refit with method="REML" for exact results.
## Warning in model.matrix.default(~m$groups[[n.levels - i + 1]] - 1,
## contrasts.arg = c("contr.treatment", : non-list contrasts argument ignored)
eREML_intMO_RI<-exactRLRT(intMO_RI_REML<-lme(intMO$call$formula, random = ~1|Ident, data = LAKE1, method=

## Warning in model.matrix.default(~m$groups[[n.levels - i + 1]] - 1,
## contrasts.arg = c("contr.treatment", : non-list contrasts argument ignored)
MOcol_RI<-Update_LM_RI(MOcol,"Ident",LAKE1)
eMLE_MOcol_RI<-exactRLRT(MOcol_RI_MLE<-lme(MOcol$call$formula, random = ~1|Ident, data = LAKE1, method=

## Using restricted likelihood evaluated at ML estimators.
## Refit with method="REML" for exact results.
## Warning in model.matrix.default(~m$groups[[n.levels - i + 1]] - 1,
## contrasts.arg = c("contr.treatment", : non-list contrasts argument ignored)
eREML_MOcol_RI<-exactRLRT(MOcol_RI_REML<-lme(MOcol$call$formula, random = ~1|Ident, data = LAKE1, method=

## Warning in model.matrix.default(~m$groups[[n.levels - i + 1]] - 1,
## contrasts.arg = c("contr.treatment", : non-list contrasts argument ignored)
intMOcol_RI<-Update_LM_RI(intMOcol,"Ident",LAKE1)
eMLE_intMOcol_RI<-exactRLRT(intMOcol_RI_MLE<-lme(intMOcol$call$formula, random = ~1|Ident, data = LAKE1

## Using restricted likelihood evaluated at ML estimators.
## Refit with method="REML" for exact results.
## Warning in model.matrix.default(~m$groups[[n.levels - i + 1]] - 1,
## contrasts.arg = c("contr.treatment", : non-list contrasts argument ignored)
eREML_intMOcol_RI<-exactRLRT(intMOcol_RI_REML<-lme(intMOcol$call$formula, random = ~1|Ident, data = LAKE1

## Warning in model.matrix.default(~m$groups[[n.levels - i + 1]] - 1,
## contrasts.arg = c("contr.treatment", : non-list contrasts argument ignored)
M1_RI<-Update_LM_RI(M1,"Ident",LAKE1)
eMLE_M1_RI<-exactRLRT(M1_RI_MLE<-lme(M1$call$formula, random = ~1|Ident, data = LAKE1, method="ML"))

```

```

## Using restricted likelihood evaluated at ML estimators.
## Refit with method="REML" for exact results.

## Warning in model.matrix.default(~m$groups[[n.levels - i + 1]] - 1,
## contrasts.arg = c("contr.treatment", : non-list contrasts argument ignored
eREML_M1_RI<-exactRLRT(M1_RI_REML<-lme(M1$call$formula, random = ~1|Ident, data = LAKE1, method="REML"))

## Warning in model.matrix.default(~m$groups[[n.levels - i + 1]] - 1,
## contrasts.arg = c("contr.treatment", : non-list contrasts argument ignored
intM1_RI<-Update_LM_RI(intM1,"Ident",LAKE1)
eMLE_intM1_RI<-exactRLRT(intM1_RI_MLE<-lme(intM1$call$formula, random = ~1|Ident, data = LAKE1, method=

## Using restricted likelihood evaluated at ML estimators.
## Refit with method="REML" for exact results.

## Warning in model.matrix.default(~m$groups[[n.levels - i + 1]] - 1,
## contrasts.arg = c("contr.treatment", : non-list contrasts argument ignored
eREML_intM1_RI<-exactRLRT(intM1_RI_REML<-lme(intM1$call$formula, random = ~1|Ident, data = LAKE1, method=

## Warning in model.matrix.default(~m$groups[[n.levels - i + 1]] - 1,
## contrasts.arg = c("contr.treatment", : non-list contrasts argument ignored
M2_RI<-Update_LM_RI(M2,"Ident",LAKE1)
eMLE_M2_RI<-exactRLRT(M2_RI_MLE<-lme(M2$call$formula, random = ~1|Ident, data = LAKE1, method="ML"))

## Using restricted likelihood evaluated at ML estimators.
## Refit with method="REML" for exact results.

## Warning in model.matrix.default(~m$groups[[n.levels - i + 1]] - 1,
## contrasts.arg = c("contr.treatment", : non-list contrasts argument ignored
eREML_M2_RI<-exactRLRT(M2_RI_REML<-lme(M2$call$formula, random = ~1|Ident, data = LAKE1, method="REML"))

## Warning in model.matrix.default(~m$groups[[n.levels - i + 1]] - 1,
## contrasts.arg = c("contr.treatment", : non-list contrasts argument ignored
intM2_RI<-Update_LM_RI(intM2,"Ident",LAKE1)
eMLE_intM2_RI<-exactRLRT(intM2_RI_MLE<-lme(intM2$call$formula, random = ~1|Ident, data = LAKE1, method=

## Using restricted likelihood evaluated at ML estimators.
## Refit with method="REML" for exact results.

## Warning in model.matrix.default(~m$groups[[n.levels - i + 1]] - 1,
## contrasts.arg = c("contr.treatment", : non-list contrasts argument ignored
eREML_intM2_RI<-exactRLRT(intM2_RI_REML<-lme(intM2$call$formula, random = ~1|Ident, data = LAKE1, method=

## Warning in model.matrix.default(~m$groups[[n.levels - i + 1]] - 1,
## contrasts.arg = c("contr.treatment", : non-list contrasts argument ignored
NomsModels <- c("M.Base","M.Base Int.",
               "M.Corr.Col","M.Corr.Col. Int.",
               "M.LogTrans. ","M.LogTrans.Int.",
               "M.Box-Cox","M.Box-Cox Int.")

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

RLRT_list_RI <- list(eMLE_MO_RI,eREML_MO_RI,eMLE_intMO_RI,eREML_intMO_RI,
                   eMLE_MOcol_RI,eREML_MOcol_RI,eMLE_intMOcol_RI,eREML_intMOcol_RI,

```

```

eMLE_M1_RI,eREML_M1_RI,eMLE_intM1_RI,eREML_intM1_RI,
eMLE_M2_RI,eREML_M2_RI,eMLE_intM2_RI,eREML_intM2_RI)
RLRT_resum_RI$"p-valor RLRT" <- sapply(1:length(RLRT_list_RI),function(l1){
  if (class(RLRT_list_RI[[l1]])!="htest"){
    return("-")
  } else {
    round(RLRT_list_RI[[l1]]$p.value,4)
  }
})

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

```

Model_Lineal	p-valor RLRT
M.Base MLE	1
M.Base REML	1
M.Base Int. MLE	1
M.Base Int. REML	1
M.Corr.Col MLE	1
M.Corr.Col REML	1
M.Corr.Col. Int. MLE	1
M.Corr.Col. Int. REML	1
M.LogTrans. MLE	1
M.LogTrans. REML	1
M.LogTrans.Int. MLE	1
M.LogTrans.Int. REML	1
M.Box-Cox MLE	1
M.Box-Cox REML	1
M.Box-Cox Int. MLE	1
M.Box-Cox Int. REML	1

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

## Diagnòstic models

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

```

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 adi
    10^(round(log10(abs(mean(ML_table$AIC))/abs(mean(ML_table$Sigma))))))
  } else {
    1/(10^(round(log10(abs(mean(ML_table$Sigma))/abs(mean(ML_table$AIC))))))
  }

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

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

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

  return(list(R1,R2))
}

```

```

M_RI_listAO <- list(MO,intMO,M0col,intM0col,
                  MO_RI[1:3],intMO_RI[1:3],M0col_RI[1:3],intM0col_RI[1:3])
M_RI_listA <- ConvList(M_RI_listAO)

```

```

## Warning in if (class(List[[l1]]) == "list") {: the condition has length > 1
## and only the first element will be used

```

```

## Warning in if (class(List[[l1]]) == "list") {: the condition has length > 1
## and only the first element will be used

```

```

## Warning in if (class(List[[l1]]) == "list") {: the condition has length > 1
## and only the first element will be used

```

```

## Warning in if (class(List[[l1]]) == "list") {: the condition has length > 1
## and only the first element will be used

```

```

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

```

```

## Warning in if (class(List[[l1]]) == "list") {: the condition has length > 1

```

```

## and only the first element will be used

## Warning in if (class(List[[l1]]) == "list") {: the condition has length > 1
## and only the first element will be used

## Warning in if (class(List[[l1]]) == "list") {: the condition has length > 1
## and only the first element will be used

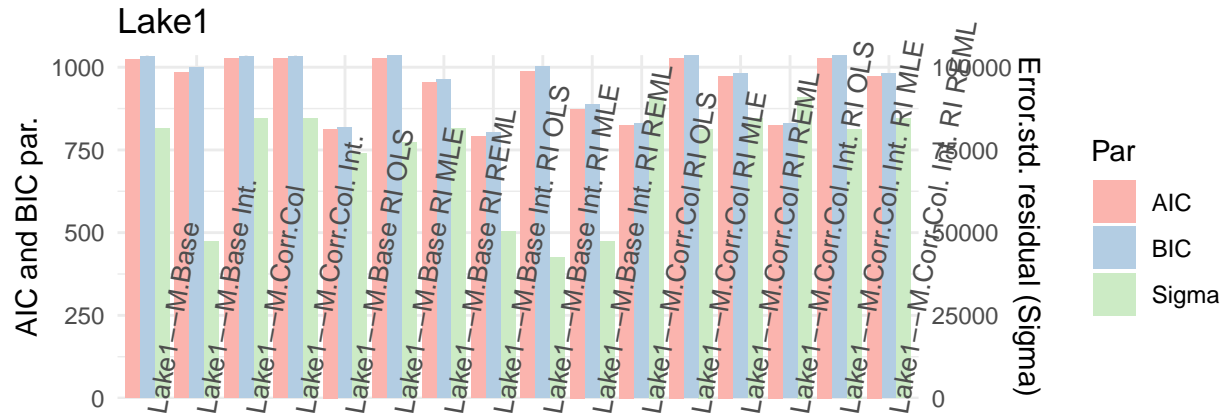
## Warning in if (class(List[[l1]]) == "list") {: the condition has length > 1
## and only the first element will be used

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

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

## [[1]]
##
##           Sigma      AIC      BIC
## Lake1--M.Base      81386.03 1023.8574 1032.3018
## Lake1--M.Base Int.  47433.69  983.9564  999.1563
## Lake1--M.Corr.Col   84579.43 1026.0323 1032.7878
## Lake1--M.Corr.Col. Int. 84579.43 1026.0323 1032.7878
## Lake1--M.Base RI OLS   73964.95  813.1882  819.0511
## Lake1--M.Base RI MLE   77209.57 1025.8574 1035.9907
## Lake1--M.Base RI REML  81386.03  954.0256  963.5267
## Lake1--M.Base Int. RI OLS 50420.37  791.9142  803.6401
## Lake1--M.Base Int. RI MLE 42425.99  985.9564 1002.8452
## Lake1--M.Base Int. RI REML 47433.69  873.8382  888.4956
## Lake1--M.Corr.Col RI OLS  91010.46  825.5455  829.9427
## Lake1--M.Corr.Col RI MLE  81345.89 1028.0323 1036.4767
## Lake1--M.Corr.Col RI REML 84579.43  973.3637  981.4183
## Lake1--M.Corr.Col. Int. RI OLS 91010.46  825.5455  829.9427
## Lake1--M.Corr.Col. Int. RI MLE 81345.89 1028.0323 1036.4767
## Lake1--M.Corr.Col. Int. RI REML 84579.43  973.3637  981.4183
##
## [[2]]

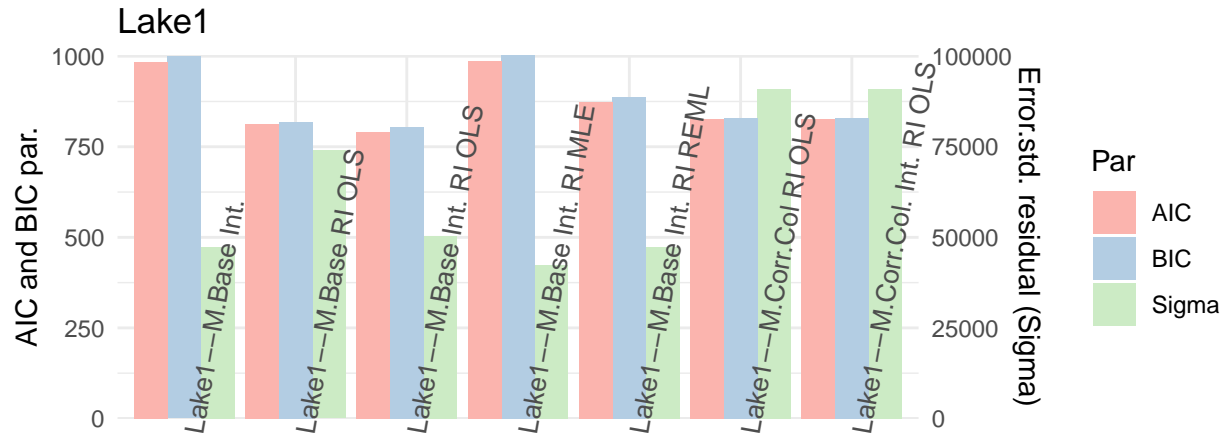
```



### Model

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

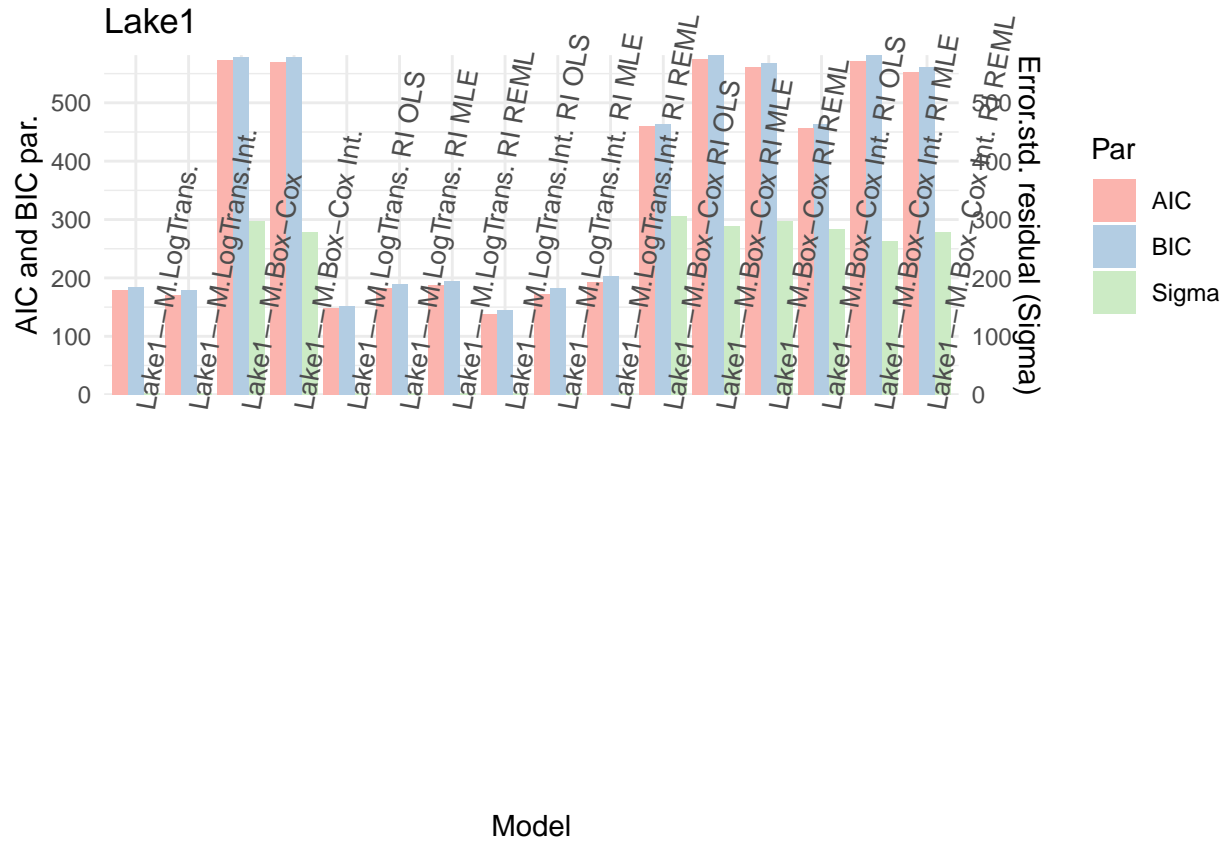
```
## [[1]]
##           Sigma      AIC      BIC
## Lake1--M.Base Int.      47433.69 983.9564 999.1563
## Lake1--M.Base RI OLS      73964.95 813.1882 819.0511
## Lake1--M.Base Int. RI OLS      50420.37 791.9142 803.6401
## Lake1--M.Base Int. RI MLE      42425.99 985.9564 1002.8452
## Lake1--M.Base Int. RI REML      47433.69 873.8382 888.4956
## Lake1--M.Corr.Col RI OLS      91010.46 825.5455 829.9427
## Lake1--M.Corr.Col. Int. RI OLS      91010.46 825.5455 829.9427
##
## [[2]]
```



### Model

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

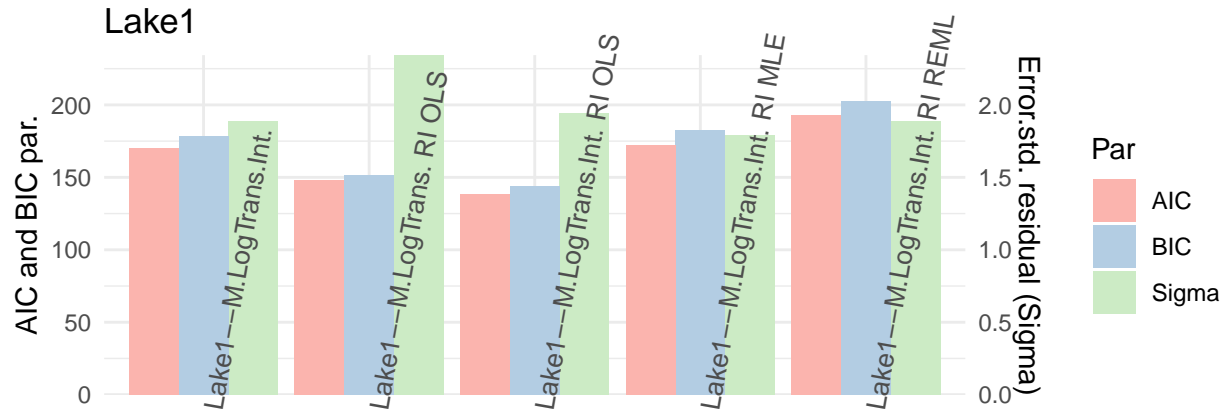
```
## [[1]]
##              Sigma      AIC      BIC
## Lake1--M.LogTrans.      2.1784 179.7516 184.8182
## Lake1--M.LogTrans.Int.      1.8898 170.2175 178.6619
## Lake1--M.Box-Cox      296.4675 572.8184 577.8850
## Lake1--M.Box-Cox Int.      277.9218 569.4879 577.9322
## Lake1--M.LogTrans. RI OLS      2.3434 148.2996 151.2311
## Lake1--M.LogTrans. RI MLE      2.1233 181.7516 188.5071
## Lake1--M.LogTrans. RI REML      2.1784 188.0539 194.6042
## Lake1--M.LogTrans.Int. RI OLS      1.9420 138.1412 144.0041
## Lake1--M.LogTrans.Int. RI MLE      1.7928 172.2175 182.3508
## Lake1--M.LogTrans.Int. RI REML      1.8898 192.9877 202.4888
## Lake1--M.Box-Cox RI OLS      305.9566 460.0965 463.0279
## Lake1--M.Box-Cox RI MLE      288.9608 574.8184 581.5739
## Lake1--M.Box-Cox RI REML      296.4675 561.4673 568.0177
## Lake1--M.Box-Cox Int. RI OLS      283.5571 457.0965 462.9594
## Lake1--M.Box-Cox Int. RI MLE      263.6598 571.4879 581.6211
## Lake1--M.Box-Cox Int. RI REML      277.9218 552.3310 561.8321
##
## [[2]]
```



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

```
## [[1]]
##           Sigma      AIC      BIC
## Lake1--M.LogTrans.Int.    1.8898 170.2175 178.6619
## Lake1--M.LogTrans. RI OLS  2.3434 148.2996 151.2311
## Lake1--M.LogTrans.Int. RI OLS 1.9420 138.1412 144.0041
## Lake1--M.LogTrans.Int. RI MLE 1.7928 172.2175 182.3508
## Lake1--M.LogTrans.Int. RI REML 1.8898 192.9877 202.4888
##
## [[2]]
```

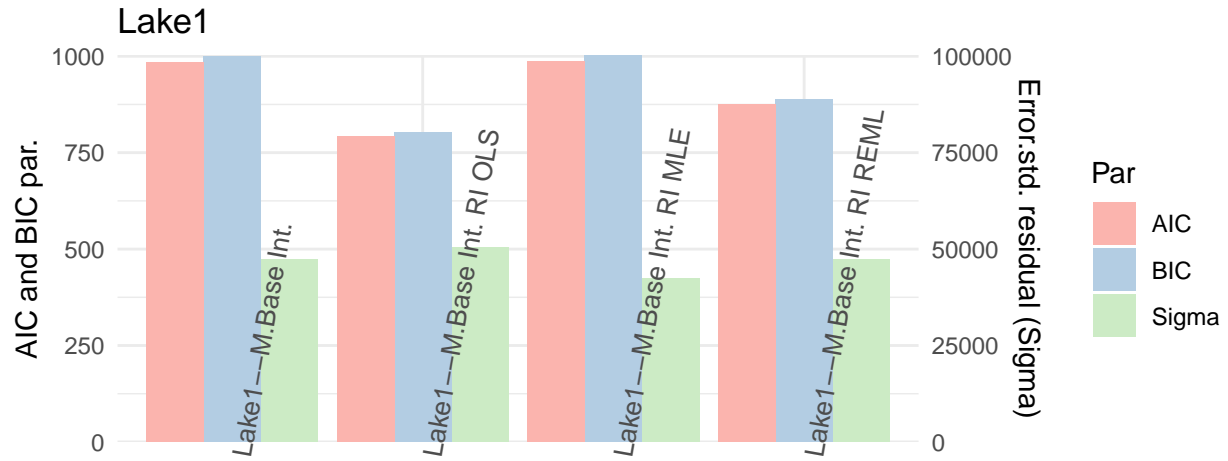




### Model

*#Deixem només de base els d'interacció i fora el de correcció de colinealitat també:*  
 (DiagAi\_RI <- ML\_Resum(M\_RI\_listA[-c(1,3:7,11:16)], "Lake1"))

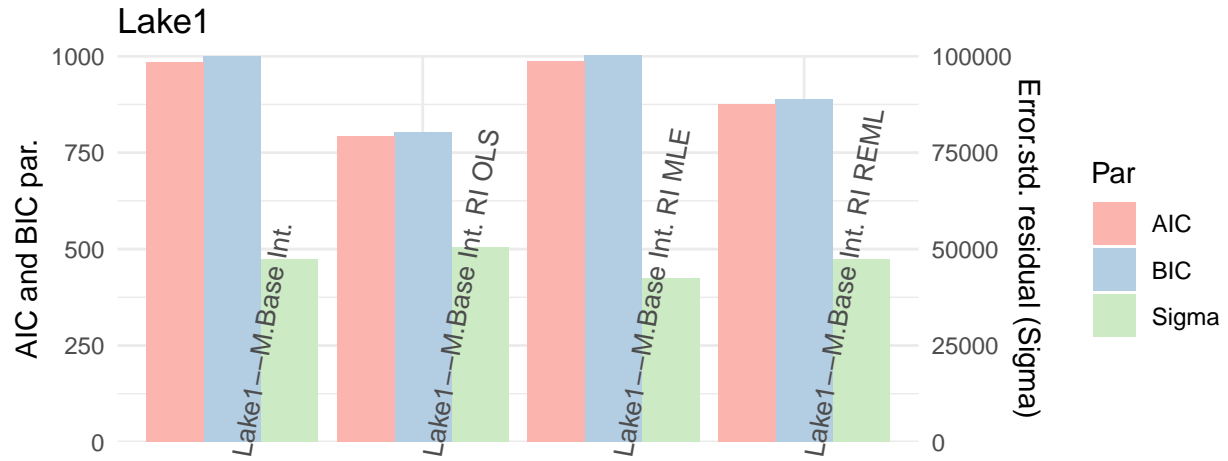
```
## [[1]]
##           Sigma      AIC      BIC
## Lake1--M.Base Int.      47433.69 983.9564 999.1563
## Lake1--M.Base Int. RI OLS 50420.37 791.9142 803.6401
## Lake1--M.Base Int. RI MLE 42425.99 985.9564 1002.8452
## Lake1--M.Base Int. RI REML 47433.69 873.8382 888.4956
##
## [[2]]
```



### Model

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

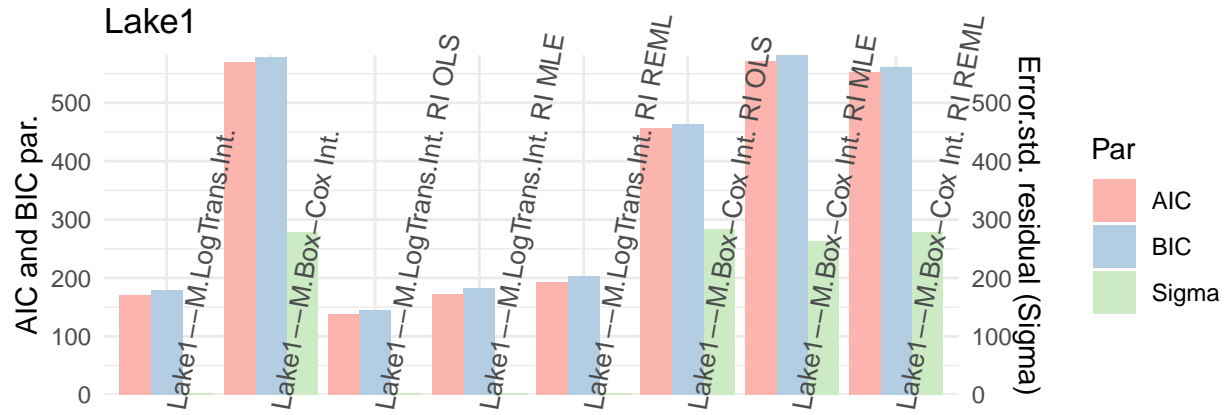
```
## [[1]]
##           Sigma      AIC      BIC
## Lake1--M.Base Int. 47433.69 983.9564 999.1563
## Lake1--M.Base Int. RI OLS 50420.37 791.9142 803.6401
## Lake1--M.Base Int. RI MLE 42425.99 985.9564 1002.8452
## Lake1--M.Base Int. RI REML 47433.69 873.8382 888.4956
##
## [[2]]
```



### Model

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

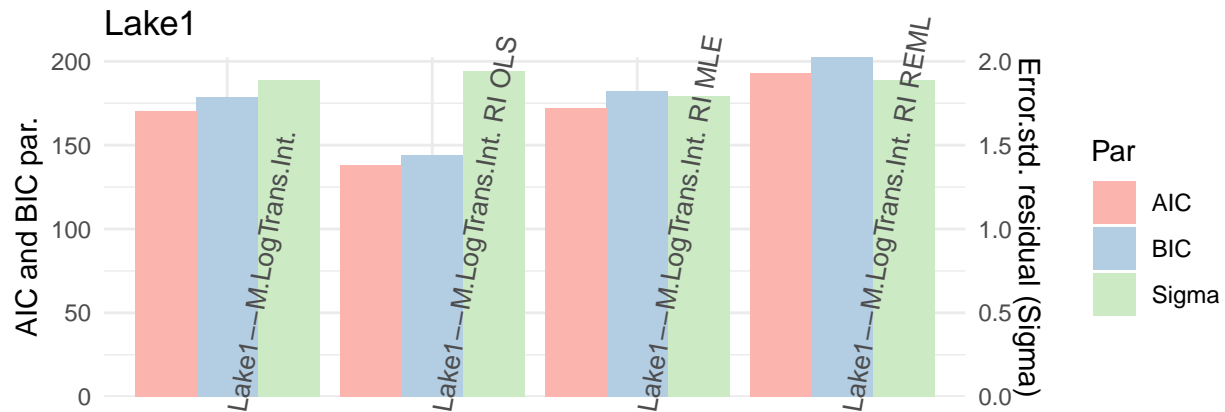
```
## [[1]]
##           Sigma      AIC      BIC
## Lake1--M.LogTrans.Int.      1.8898 170.2175 178.6619
## Lake1--M.Box-Cox Int.      277.9218 569.4879 577.9322
## Lake1--M.LogTrans.Int. RI OLS      1.9420 138.1412 144.0041
## Lake1--M.LogTrans.Int. RI MLE      1.7928 172.2175 182.3508
## Lake1--M.LogTrans.Int. RI REML      1.8898 192.9877 202.4888
## Lake1--M.Box-Cox Int. RI OLS      283.5571 457.0965 462.9594
## Lake1--M.Box-Cox Int. RI MLE      263.6598 571.4879 581.6211
## Lake1--M.Box-Cox Int. RI REML      277.9218 552.3310 561.8321
##
## [[2]]
```



### Model

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

```
## [[1]]
##              Sigma      AIC      BIC
## Lake1--M.LogTrans.Int.    1.8898 170.2175 178.6619
## Lake1--M.LogTrans.Int. RI OLS 1.9420 138.1412 144.0041
## Lake1--M.LogTrans.Int. RI MLE 1.7928 172.2175 182.3508
## Lake1--M.LogTrans.Int. RI REML 1.8898 192.9877 202.4888
##
## [[2]]
```



## Model

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

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

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

```
M_RI_listA_R2 <- list(M0,intM0,M0col,intM0col,M0_RI_MLE,M0_RI_REML,
                    intM0_RI_MLE,intM0_RI_REML,M0col_RI_MLE,M0col_RI_REML,
                    intM0col_RI_MLE,intM0col_RI_REML)

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

M_RI_dfA_R2 <- data.frame(Model = names(M_RI_listA_R2))
M_RI_dfA_R2$R2 <- sapply(1:length(M_RI_listA_R2),function(M){
  if (class(M_RI_listA_R2[[M]])[1]=="lmm"){
    summary(M_RI_listA_R2[[M]])$r.squared
  }
})
```

```

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

```

```

## Warning in paste(attr(M_RI_listA_R2[[M]]$terms, "predvars")[-c(1:2)]) == :
## longer object length is not a multiple of shorter object length

```

```

## Warning in paste(attr(M_RI_listA_R2[[M]]$terms, "predvars")[-c(1:2)]) == :
## longer object length is not a multiple of shorter object length

```

```

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

```

```

M_RI_listB_R2 <- list(M1,intM1,M2,intM2,M1_RI_MLE,
                    M1_RI_REML,intM1_RI_MLE,intM1_RI_REML,
                    M2_RI_MLE,M2_RI_REML,intM2_RI_MLE,intM2_RI_REML)

```

```

names(M_RI_listB_R2)[1:4] <- NomsModels[5:8]

```

```

names(M_RI_listB_R2)[(4+1):length(M_RI_listB_R2)]<-paste(rep(NomsModels[5:8],each=2),rep(paste("RI",c("1",

```

```

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

```

```

## Warning in paste(attr(M_RI_listB_R2[[M]]$terms, "predvars")[-c(1:2)]) == :
## longer object length is not a multiple of shorter object length

```

```

## Warning in paste(attr(M_RI_listB_R2[[M]]$terms, "predvars")[-c(1:2)]) == :
## longer object length is not a multiple of shorter object length

```

```

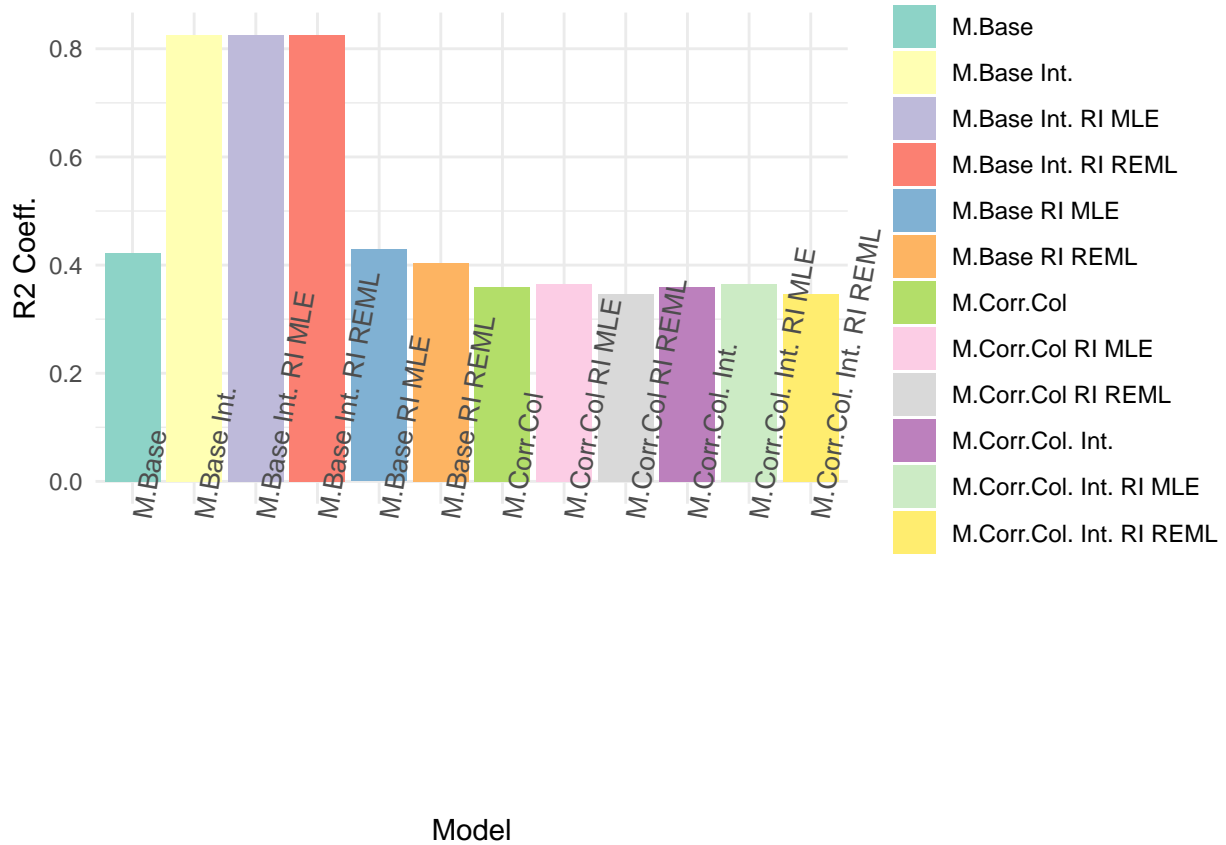
## Warning in paste(attr(M_RI_listB_R2[[M]]$terms, "predvars")[-c(1:2)]) == :
## longer object length is not a multiple of shorter object length

```

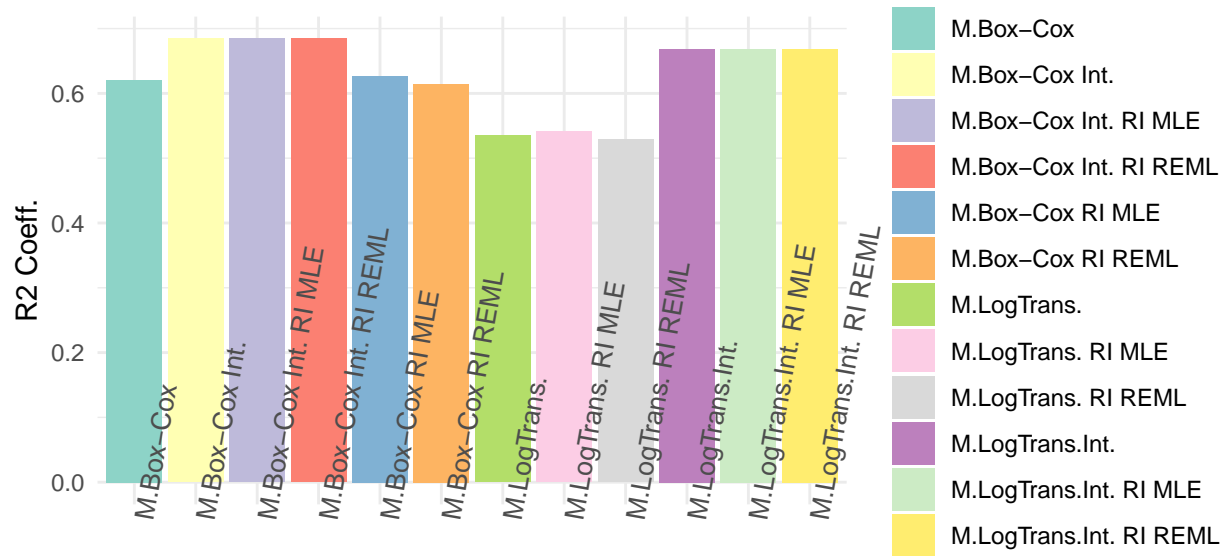
```
## Warning in paste(attr(M_RI_listB_R2[[M]]$terms, "predvars")[-c(1:2)]) == :
## longer object length is not a multiple of shorter object length
```

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

R2\_A



R2\_B



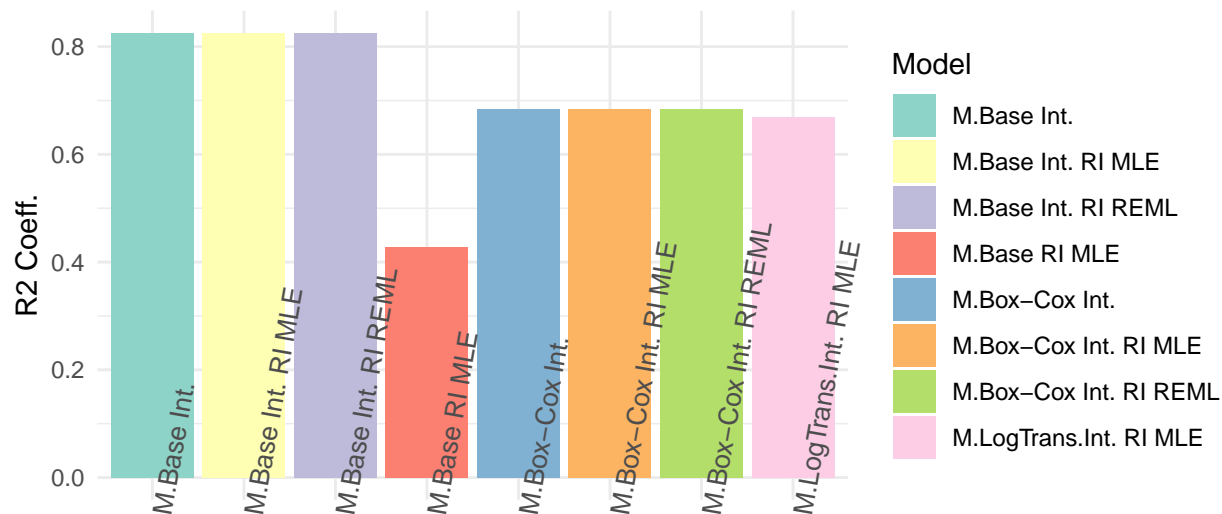
### Model

```
BestA_R2 <- order(M_RI_dfA_R2$R2,decreasing=T)[1:4]
BestB_R2 <- order(M_RI_dfB_R2$R2,decreasing=T)[1:4]
M_RI_df_sel_R2 <- rbind(M_RI_dfA_R2[BestA_R2,],M_RI_dfB_R2[BestB_R2,])

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

R2_sel
```





### Model

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

## Pendent Aleatòria

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

## Ajust models

```
Update_LM_RIS <- function(LM,RI_var,RIS_var,df) { #Objecte del tipus LM

  fmOLS <- update(LM$call$formula,paste(" ~ . + Error(",RI_var,"/",RIS_var,""))
  ML_I_RIS_sOLS <- summary(ML_I_RIS_OLS <- aov(fmOLS, data = df))

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

```

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

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

return(l)
}

```

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

*#NO ES PODEN FER ANOVAS DE RESULTATS DE FUNCIONS JA QUE NECESSITA LES FORMULES ORIGINALS, ES FA INDIVIDUALMENT*

```
MO_RIS<-Update_LM_RIS(MO,"Ident","Tiempo",LAKE1)
```

```
MO_RIS_REML<-lme(MO$call$formula, random = ~1+Tiempo|Ident, control=ctrl, data = LAKE1, method="REML")
(MO_RI_RIS_anREML <- anova(MO_RI_REML,MO_RIS_REML))
```

```
##           Model df      AIC      BIC    logLik  Test  L.Ratio p-value
## MO_RI_REML      1  6 954.0256 963.5267 -471.0128
## MO_RIS_REML     2  8 957.7893 970.4574 -470.8946 1 vs 2 0.2363341 0.8885
```

```
intMO_RIS<-Update_LM_RIS(intMO,"Ident","Tiempo",LAKE1)
```

```
intMO_RIS_REML<-lme(intMO$call$formula, random = ~1+Tiempo|Ident, control=ctrl, data = LAKE1, method="REML")
(intMO_RI_RIS_anREML <- anova(intMO_RI_REML,intMO_RIS_REML))
```

```
##           Model df      AIC      BIC    logLik  Test  L.Ratio
## intMO_RI_REML    1 10 873.8382 888.4956 -426.9191
## intMO_RIS_REML   2 12 872.5253 890.1141 -424.2626 1 vs 2 5.312956
##           p-value
## intMO_RI_REML
## intMO_RIS_REML 0.0702
```

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

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

```
##           Model df      AIC      BIC    logLik  Test  L.Ratio
## M0col_RI_REML    1  5 973.3637 981.4183 -481.6818
## M0col_RIS_REML   2  7 977.4262 988.7026 -481.7131 1 vs 2 0.06251701
##           p-value
## M0col_RI_REML
## M0col_RIS_REML 0.9692
```

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

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

```
##           Model df      AIC      BIC    logLik  Test  L.Ratio
## intM0col_RI_REML  1  5 973.3637 981.4183 -481.6818
## intM0col_RIS_REML  2  7 977.4262 988.7026 -481.7131 1 vs 2 0.06251701
##           p-value
```

```

## intM0col_RI_REML
## intM0col_RIS_REML 0.9692
M1_RIS<-Update_LM_RIS(M1,"Ident","Tiempo",LAKE1)

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

##          Model df      AIC      BIC   logLik   Test     L.Ratio
## M1_RI_REML      1  4 188.0539 194.6042 -90.02693
## M1_RIS_REML      2  6 192.0543 201.8799 -90.02717 1 vs 2 0.0004871679
##          p-value
## M1_RI_REML
## M1_RIS_REML 0.9998
intM1_RIS<-Update_LM_RIS(intM1,"Ident","Tiempo",LAKE1)

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

##          Model df      AIC      BIC   logLik   Test     L.Ratio
## intM1_RI_REML      1  6 192.9877 202.4888 -90.49385
## intM1_RIS_REML      2  8 189.3124 201.9806 -86.65620 1 vs 2 7.675298
##          p-value
## intM1_RI_REML
## intM1_RIS_REML 0.0215
M2_RIS<-Update_LM_RIS(M2,"Ident","Tiempo",LAKE1)

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

##          Model df      AIC      BIC   logLik   Test     L.Ratio
## M2_RI_REML      1  4 561.4673 568.0177 -276.7337
## M2_RIS_REML      2  6 565.4728 575.2983 -276.7364 1 vs 2 0.005464781
##          p-value
## M2_RI_REML
## M2_RIS_REML 0.9973
intM2_RIS<-Update_LM_RIS(intM2,"Ident","Tiempo",LAKE1)

intM2_RIS_REML<-lme(intM2$call$formula, random = ~1+Tiempo|Ident, control=ctrl, data = LAKE1, method="REML")
(intM2_RI_RIS_anREML <- anova(intM2_RI_REML,intM2_RIS_REML))

##          Model df      AIC      BIC   logLik   Test     L.Ratio
## intM2_RI_REML      1  6 552.3310 561.8321 -270.1655
## intM2_RIS_REML      2  8 556.0538 568.7220 -270.0269 1 vs 2 0.2771768
##          p-value
## intM2_RI_REML
## intM2_RIS_REML 0.8706
#Generació dels objectes MLE i comparativa
MO_RIS_MLE<-lme(MO$call$formula, random = ~1+Tiempo|Ident, control=ctrl, data = LAKE1, method="ML")
(MO_RI_RIS_anMLE <- anova(MO_RI_MLE,MO_RIS_MLE))

##          Model df      AIC      BIC   logLik   Test     L.Ratio p-value
## MO_RI_MLE      1  6 1025.857 1035.991 -506.9287
## MO_RIS_MLE      2  8 1029.952 1043.463 -506.9759 1 vs 2 0.0943392 0.9539

```

```
intMO_RIS<-Update_LM_RIS(intMO,"Ident","Tiempo",LAKE1)
```

```
intMO_RIS_MLE<-lme(intMO$call$formula, random = ~1+Tiempo|Ident, control=ctrl, data = LAKE1, method="ML")  
(intMO_RI_RIS_anMLE <- anova(intMO_RI_MLE,intMO_RIS_MLE))
```

```
##           Model df      AIC      BIC   logLik   Test   L.Ratio p-value  
## intMO_RI_MLE      1 10 985.9564 1002.845 -482.9782  
## intMO_RIS_MLE     2 12 984.1226 1004.389 -480.0613 1 vs 2 5.833798 0.0541
```

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

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

```
##           Model df      AIC      BIC   logLik   Test   L.Ratio  
## M0col_RI_MLE      1  5 1028.032 1036.477 -509.0162  
## M0col_RIS_MLE     2  7 1032.037 1043.859 -509.0184 1 vs 2 0.004471928  
##           p-value  
## M0col_RI_MLE  
## M0col_RIS_MLE 0.9978
```

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

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

```
##           Model df      AIC      BIC   logLik   Test   L.Ratio  
## intM0col_RI_MLE   1  5 1028.032 1036.477 -509.0162  
## intM0col_RIS_MLE  2  7 1032.037 1043.859 -509.0184 1 vs 2 0.004471928  
##           p-value  
## intM0col_RI_MLE  
## intM0col_RIS_MLE 0.9978
```

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

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

```
##           Model df      AIC      BIC   logLik   Test   L.Ratio  
## M1_RI_MLE         1  4 181.7516 188.5071 -86.87579  
## M1_RIS_MLE        2  6 185.7520 195.8853 -86.87602 1 vs 2 0.0004611924  
##           p-value  
## M1_RI_MLE  
## M1_RIS_MLE 0.9998
```

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

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

```
##           Model df      AIC      BIC   logLik   Test   L.Ratio p-value  
## intM1_RI_MLE      1  6 172.2175 182.3508 -80.10876  
## intM1_RIS_MLE     2  8 168.7252 182.2362 -76.36260 1 vs 2 7.492317 0.0236
```

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

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

```

(M2_RI_RIS_anMLE <- anova(M2_RI_MLE,M2_RIS_MLE))

##           Model df      AIC      BIC    logLik    Test      L.Ratio p-value
## M2_RI_MLE      1  4 574.8184 581.5739 -283.4092
## M2_RIS_MLE     2  6 578.8238 588.9571 -283.4119 1 vs 2 0.005454479 0.9973

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

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

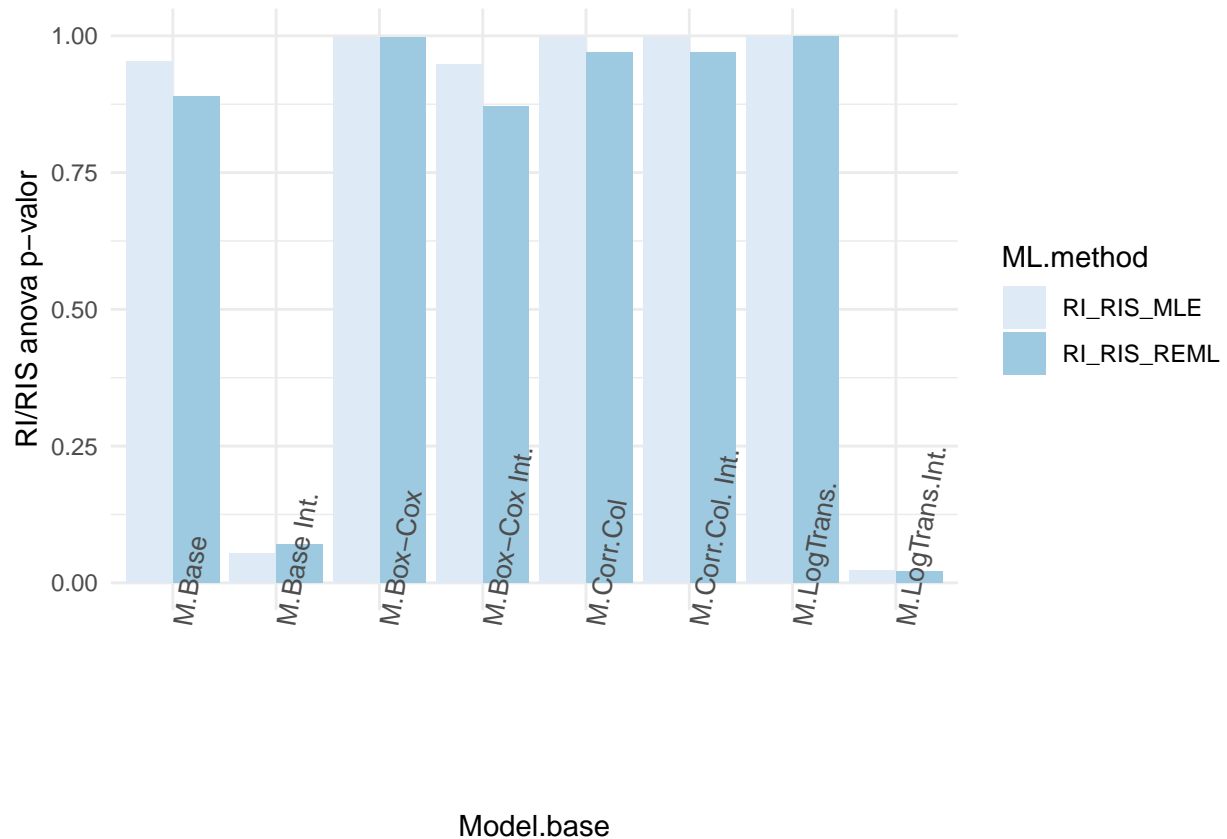
##           Model df      AIC      BIC    logLik    Test      L.Ratio
## intM2_RI_MLE    1  6 571.4879 581.6211 -279.7439
## intM2_RIS_MLE   2  8 575.5950 589.1060 -279.7975 1 vs 2 0.1071298
##                                     p-value
## intM2_RI_MLE
## intM2_RIS_MLE 0.9478

RI_RIS_an <- data.frame(Model = paste(rep(NomsModels,2),rep(c("RI_RIS_MLE", "RI_RIS_REML"),each=length(NomsModels))),
RI_RIS_an$Model.base <- rep(NomsModels,2)
RI_RIS_an$ML.method <- rep(c("RI_RIS_MLE", "RI_RIS_REML"),each=length(NomsModels))
RI_RIS_an$p.valor <- c(MO_RI_RIS_anMLE$p-value[2], intMO_RI_RIS_anMLE$p-value[2], MOcol_RI_RIS_anMLE$p-value[2],
intMOcol_RI_RIS_anMLE$p-value[2], M1_RI_RIS_anMLE$p-value[2], intM1_RI_RIS_anMLE$p-value[2],
M2_RI_RIS_anMLE$p-value[2], intM2_RI_RIS_anMLE$p-value[2],
MO_RI_RIS_anREML$p-value[2], intMO_RI_RIS_anREML$p-value[2], MOcol_RI_RIS_anREML$p-value[2],
intMOcol_RI_RIS_anREML$p-value[2], M1_RI_RIS_anREML$p-value[2], intM1_RI_RIS_anREML$p-value[2],
M2_RI_RIS_anREML$p-value[2], intM2_RI_RIS_anREML$p-value[2])

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

RI_RIS_an_plot

```



En la majoria de models segueix essent la significació baixa (p-valor alt), però es veu curiosament que hi ha alguns models on es torna o queda proper a la significància, concretament si es pren els que tenen p-valors per sota de 0.25 serien M.Base Int. RI\_RIS\_MLE, M.LogTrans.Int. RI\_RIS\_MLE, M.Base Int. RI\_RIS\_REML, M.LogTrans.Int. RI\_RIS\_REML amb p-valors de 0.0541, 0.0236, 0.0702, 0.0215 respectivament.

### Diagnòstic models

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

```
M_RIS_listA0 <- list(M0,intM0,M0col,intM0col,
                    M0_RIS[1:3],intM0_RIS[1:3],M0col_RIS[1:3],intM0col_RIS[1:3])
M_RIS_listA <- ConvList(M_RIS_listA0)
```

```
## Warning in if (class(List[[l1]]) == "list") {: the condition has length > 1
## and only the first element will be used
```

```
## Warning in if (class(List[[l1]]) == "list") {: the condition has length > 1
## and only the first element will be used
```

```
## Warning in if (class(List[[l1]]) == "list") {: the condition has length > 1
## and only the first element will be used
```

```
## Warning in if (class(List[[l1]]) == "list") {: the condition has length > 1
## and only the first element will be used
```

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

```
M_RIS_listB <- ConvList(M_RIS_listB0)
```

```
## Warning in if (class(List[[l1]]) == "list") {: the condition has length > 1  
## and only the first element will be used
```

```
## Warning in if (class(List[[l1]]) == "list") {: the condition has length > 1  
## and only the first element will be used
```

```
## Warning in if (class(List[[l1]]) == "list") {: the condition has length > 1  
## and only the first element will be used
```

```
## Warning in if (class(List[[l1]]) == "list") {: the condition has length > 1  
## and only the first element will be used
```

```
names(M_RIS_listA)[1:4] <- NomsModels[1:4]
```

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

```
names(M_RIS_listB)[1:4] <- NomsModels[5:8]
```

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

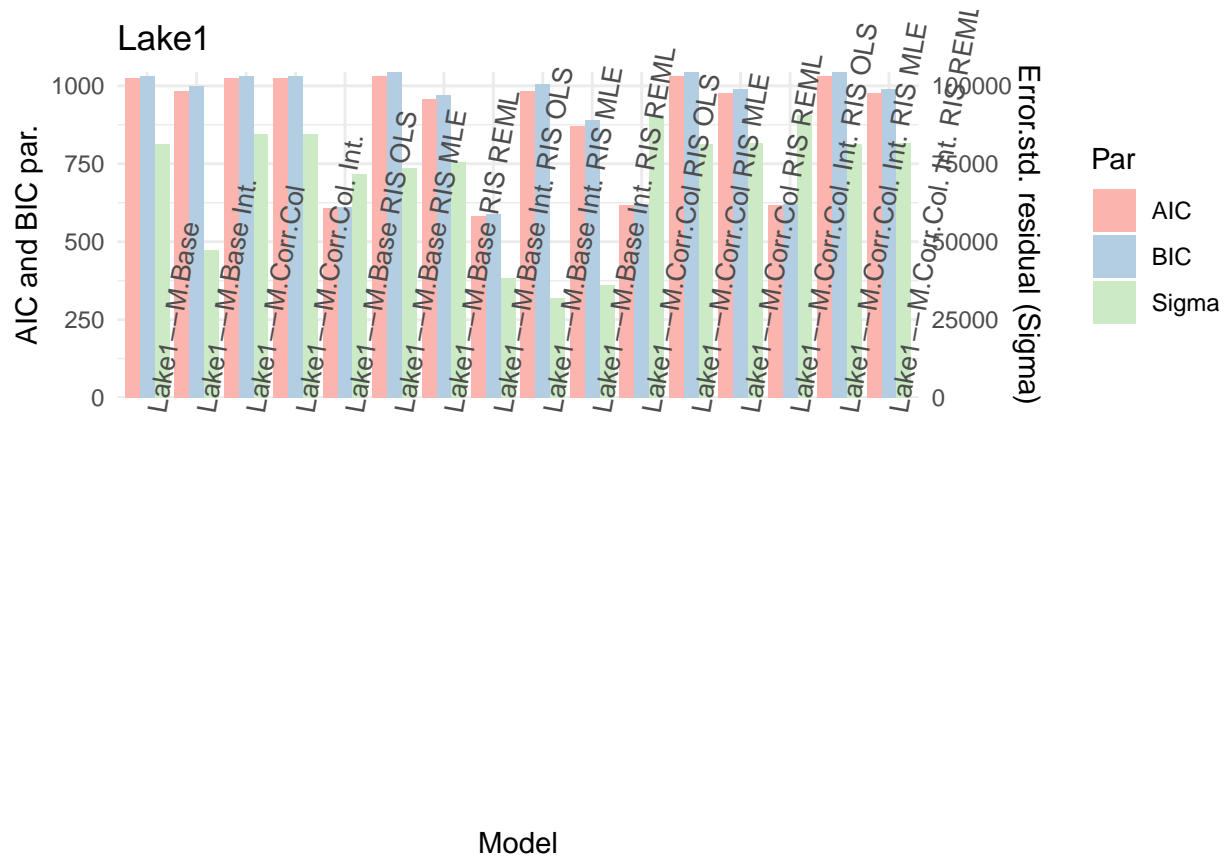
```
(DiagA_RIS <- ML_Resum(M_RIS_listA,"Lake1"))
```

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

```
##           Sigma      AIC      BIC  
## Lake1--M.Base      81386.03 1023.8574 1032.3018  
## Lake1--M.Base Int.  47433.69  983.9564  999.1563  
## Lake1--M.Corr.Col   84579.43 1026.0323 1032.7878  
## Lake1--M.Corr.Col. Int. 84579.43 1026.0323 1032.7878  
## Lake1--M.Base RIS OLS  71784.03  608.7288  612.2630  
## Lake1--M.Base RIS MLE  73628.98 1029.9517 1043.4628  
## Lake1--M.Base RIS REML 75424.20  957.7893  970.4574  
## Lake1--M.Base Int. RIS OLS 38378.48  581.8568  590.1032  
## Lake1--M.Base Int. RIS MLE 32069.02  984.1226 1004.3891  
## Lake1--M.Base Int. RIS REML 35948.69  872.5253  890.1141  
## Lake1--M.Corr.Col RIS OLS  90625.78  618.9833  621.3394  
## Lake1--M.Corr.Col RIS MLE  81343.91 1032.0368 1043.8590  
## Lake1--M.Corr.Col RIS REML 81534.59  977.4262  988.7026  
## Lake1--M.Corr.Col. Int. RIS OLS 90625.78  618.9833  621.3394  
## Lake1--M.Corr.Col. Int. RIS MLE 81343.91 1032.0368 1043.8590  
## Lake1--M.Corr.Col. Int. RIS REML 81534.59  977.4262  988.7026
```

```
##
```

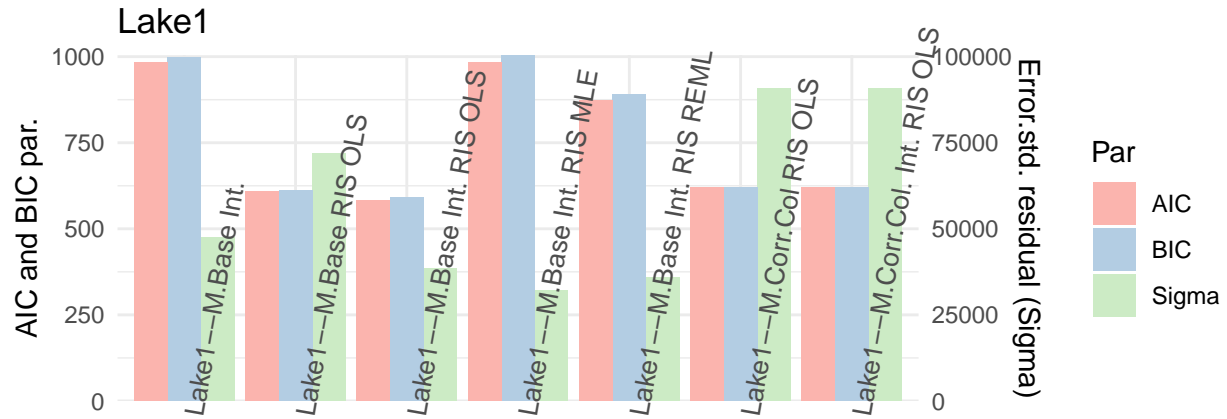
```
## [[2]]
```



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

```
## [[1]]
##           Sigma      AIC      BIC
## Lake1--M.Base Int. 47433.69 983.9564 999.1563
## Lake1--M.Base RIS OLS 71784.03 608.7288 612.2630
## Lake1--M.Base Int. RIS OLS 38378.48 581.8568 590.1032
## Lake1--M.Base Int. RIS MLE 32069.02 984.1226 1004.3891
## Lake1--M.Base Int. RIS REML 35948.69 872.5253 890.1141
## Lake1--M.Corr.Col RIS OLS 90625.78 618.9833 621.3394
## Lake1--M.Corr.Col. Int. RIS OLS 90625.78 618.9833 621.3394
##
## [[2]]
```

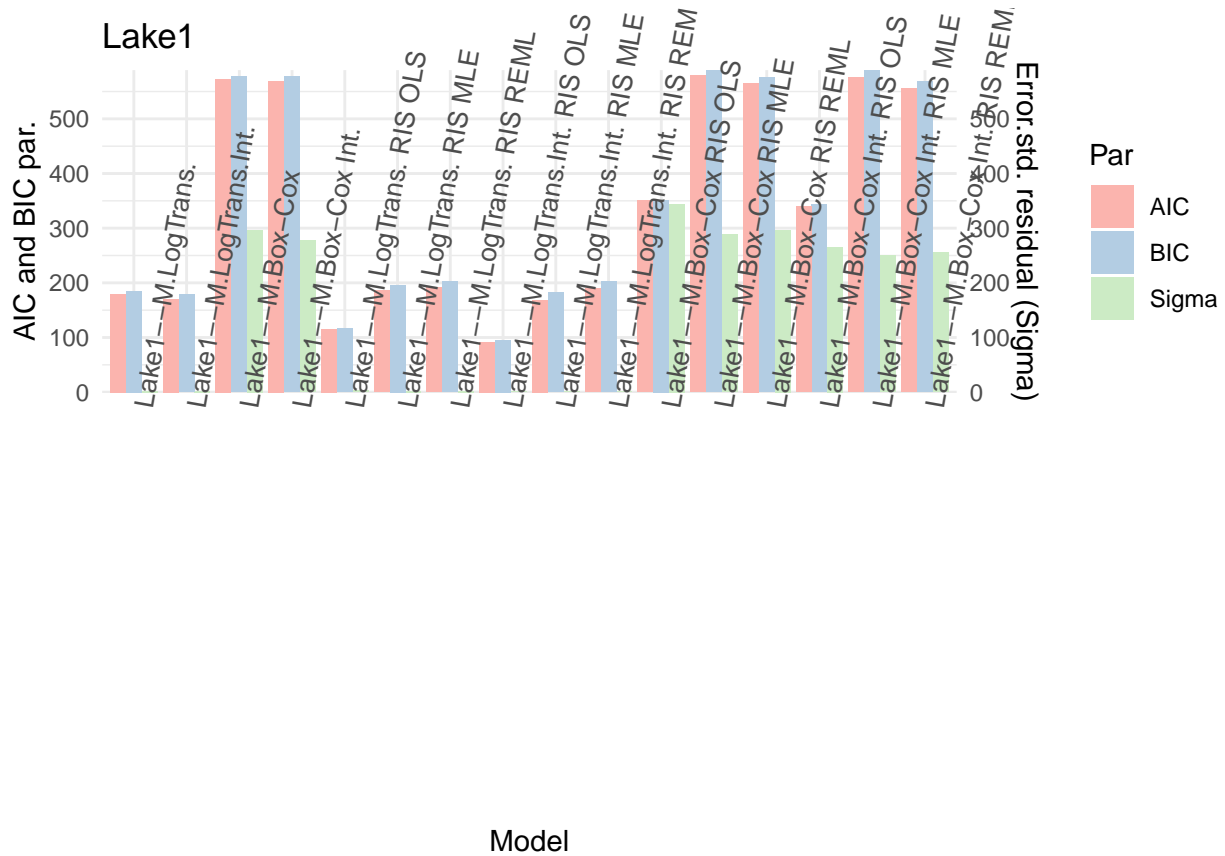




### Model

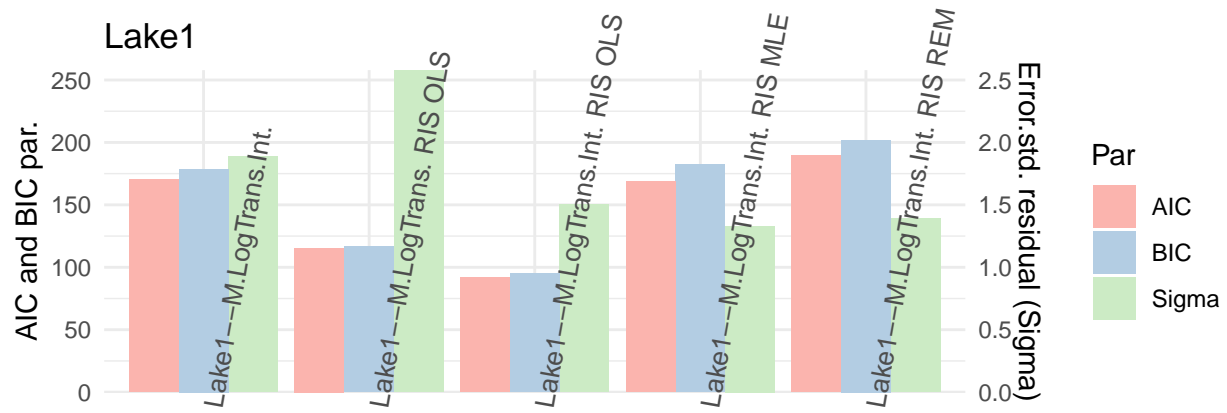
```
(DiagB_RIS<-ML_Resum(M_RIS_listB,"Lake1"))
```

```
## [[1]]
##              Sigma      AIC      BIC
## Lake1--M.LogTrans.      2.1784 179.7516 184.8182
## Lake1--M.LogTrans.Int.      1.8898 170.2175 178.6619
## Lake1--M.Box-Cox      296.4675 572.8184 577.8850
## Lake1--M.Box-Cox Int.      277.9218 569.4879 577.9322
## Lake1--M.LogTrans. RIS OLS      2.5788 115.5800 116.7580
## Lake1--M.LogTrans. RIS MLE      2.1233 185.7520 195.8853
## Lake1--M.LogTrans. RIS REML      2.1784 192.0543 201.8799
## Lake1--M.LogTrans.Int. RIS OLS      1.5054  91.6563  95.1905
## Lake1--M.LogTrans.Int. RIS MLE      1.3307 168.7252 182.2362
## Lake1--M.LogTrans.Int. RIS REML      1.3916 189.3124 201.9806
## Lake1--M.Box-Cox RIS OLS      344.1067 350.4747 351.6528
## Lake1--M.Box-Cox RIS MLE      288.9329 578.8238 588.9571
## Lake1--M.Box-Cox RIS REML      296.4283 565.4728 575.2983
## Lake1--M.Box-Cox Int. RIS OLS      265.9007 340.0107 343.5448
## Lake1--M.Box-Cox Int. RIS MLE      249.5092 575.5950 589.1060
## Lake1--M.Box-Cox Int. RIS REML      255.5007 556.0538 568.7220
##
## [[2]]
```



```
BestB2 <- unique(c(order(DiagB_RIS[[1]]$AIC,decreasing=F)[1:4],
                      order(DiagB_RIS[[1]]$Sigma,decreasing=F)[1:4]))
ML_Resum(M_RIS_listB[sort(BestB2)],"Lake1") #Comparativa més selectiva
```

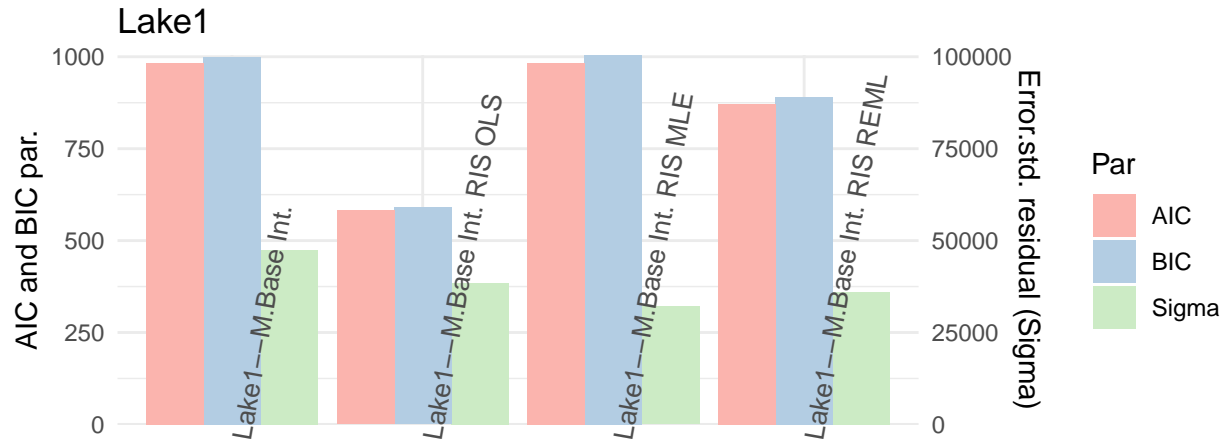
```
## [[1]]
##           Sigma      AIC      BIC
## Lake1--M.LogTrans.Int.  1.8898 170.2175 178.6619
## Lake1--M.LogTrans. RIS OLS  2.5788 115.5800 116.7580
## Lake1--M.LogTrans.Int. RIS OLS  1.5054  91.6563  95.1905
## Lake1--M.LogTrans.Int. RIS MLE  1.3307 168.7252 182.2362
## Lake1--M.LogTrans.Int. RIS REML 1.3916 189.3124 201.9806
##
## [[2]]
```



### Model

*#Deixem només de base els d'interacció i fora el de correcció de colinealitat també:*  
 (DiagAi\_RIS <- ML\_Resum(M\_RIS\_listA[-c(1,3:7,11:16)],"Lake1"))

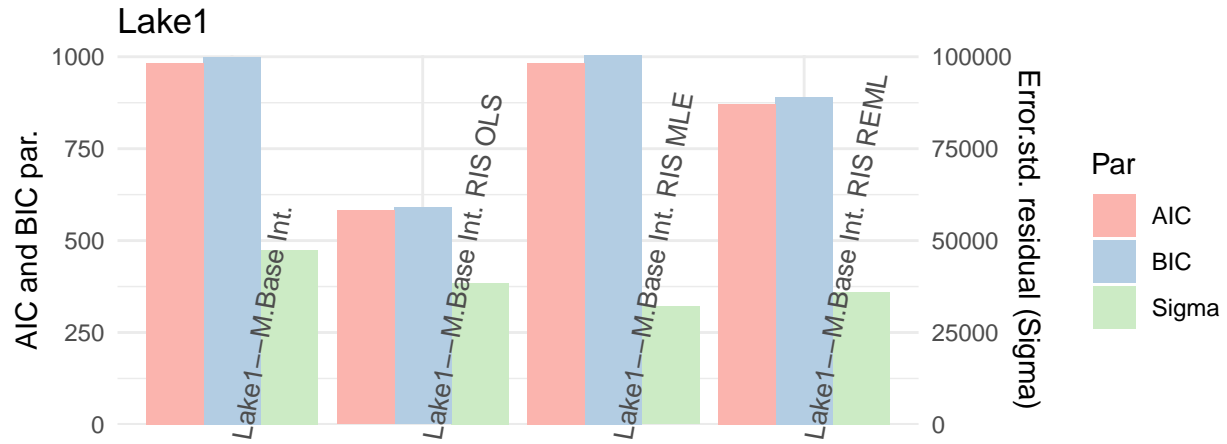
```
## [[1]]
##           Sigma      AIC      BIC
## Lake1--M.Base Int.      47433.69 983.9564 999.1563
## Lake1--M.Base Int. RIS OLS 38378.48 581.8568 590.1032
## Lake1--M.Base Int. RIS MLE 32069.02 984.1226 1004.3891
## Lake1--M.Base Int. RIS REML 35948.69 872.5253 890.1141
##
## [[2]]
```



### Model

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

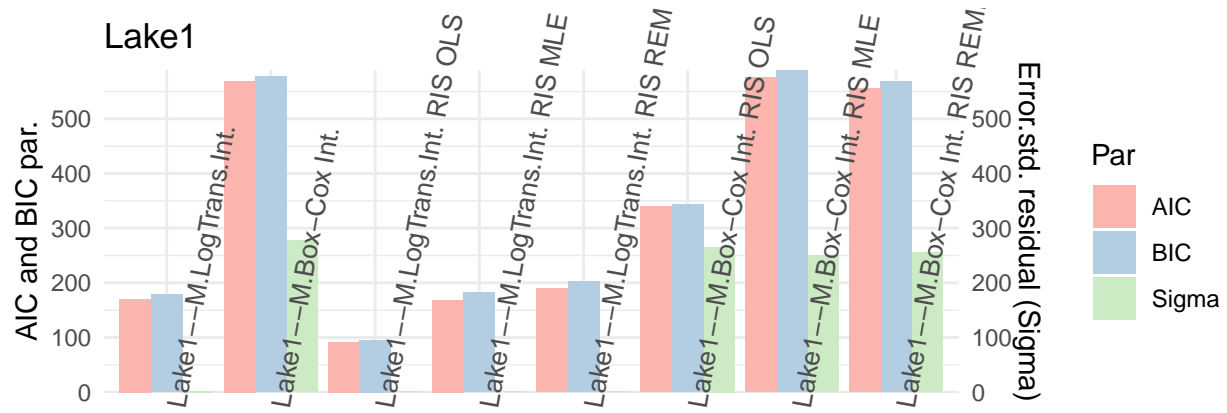
```
## [[1]]
##           Sigma      AIC      BIC
## Lake1--M.Base Int. 47433.69 983.9564 999.1563
## Lake1--M.Base Int. RIS OLS 38378.48 581.8568 590.1032
## Lake1--M.Base Int. RIS MLE 32069.02 984.1226 1004.3891
## Lake1--M.Base Int. RIS REML 35948.69 872.5253 890.1141
##
## [[2]]
```



### Model

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

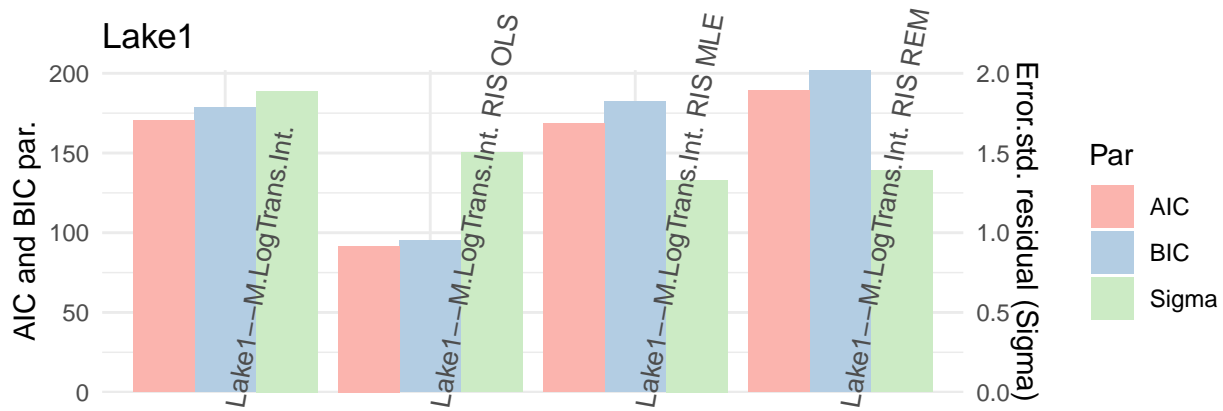
```
## [[1]]
##              Sigma      AIC      BIC
## Lake1--M.LogTrans.Int.      1.8898 170.2175 178.6619
## Lake1--M.Box-Cox Int.      277.9218 569.4879 577.9322
## Lake1--M.LogTrans.Int. RIS OLS      1.5054  91.6563  95.1905
## Lake1--M.LogTrans.Int. RIS MLE      1.3307 168.7252 182.2362
## Lake1--M.LogTrans.Int. RIS REML      1.3916 189.3124 201.9806
## Lake1--M.Box-Cox Int. RIS OLS      265.9007 340.0107 343.5448
## Lake1--M.Box-Cox Int. RIS MLE      249.5092 575.5950 589.1060
## Lake1--M.Box-Cox Int. RIS REML      255.5007 556.0538 568.7220
##
## [[2]]
```



### Model

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

```
## [[1]]
##              Sigma      AIC      BIC
## Lake1--M.LogTrans.Int.      1.8898 170.2175 178.6619
## Lake1--M.LogTrans.Int. RIS OLS 1.5054  91.6563  95.1905
## Lake1--M.LogTrans.Int. RIS MLE 1.3307 168.7252 182.2362
## Lake1--M.LogTrans.Int. RIS REML 1.3916 189.3124 201.9806
##
## [[2]]
```



## Model

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

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

```
M_RIS_listA_R2 <- list(M0,intM0,M0col,intM0col,M0_RIS_MLE,M0_RIS_REML,
                    intM0_RIS_MLE,intM0_RIS_REML,M0col_RIS_MLE,M0col_RIS_REML,
                    intM0col_RIS_MLE,intM0col_RIS_REML)

names(M_RIS_listA_R2)[1:4] <- NomsModels[1:4]
names(M_RIS_listA_R2)[(4+1):length(M_RIS_listA_R2)]<-paste(rep(NomsModels[1:4],each=2),rep(paste("RIS",
M_RIS_dfa_R2 <- data.frame(Model = names(M_RIS_listA_R2))
M_RIS_dfa_R2$R2 <- sapply(1:length(M_RIS_listA_R2),function(M){
  if (class(M_RIS_listA_R2[[M]])[1]=="lmm"){
    summary(M_RIS_listA_R2[[M]])$r.squared
  } else if (class(M_RIS_listA_R2[[M]])=="lme"){
    if(all(paste(attr(M_RIS_listA_R2[[M]]$terms,"predvars")[-c(1:2)]) ==
              paste(attr(M_RIS_listA_R2[[M]]$terms,"term.labels")))){
      rsquared(M_RIS_listA_R2[[M]])$Conditional
```

```

} else {
  summary(lm(formula(paste(paste("M_RIS_listA_R2[[" ,M, "]"$data$",
    attr(M_RIS_listA_R2[[M]]$terms,"variables")[[2]],sep=""),
    "~", "M_RIS_listA_R2[[" ,M, "]"$fitted"),sep=""))) $r.squared
}
}
})

```

```

## Warning in paste(attr(M_RIS_listA_R2[[M]]$terms, "predvars")[-c(1:2)]) == :
## longer object length is not a multiple of shorter object length

```

```

## Warning in paste(attr(M_RIS_listA_R2[[M]]$terms, "predvars")[-c(1:2)]) == :
## longer object length is not a multiple of shorter object length

```

```

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

```

```

M_RIS_listB_R2 <- list(M1,intM1,M2,intM2,M1_RIS_MLE,
  M1_RIS_REML,intM1_RIS_MLE,intM1_RIS_REML,
  M2_RIS_MLE,M2_RIS_REML,intM2_RIS_MLE,intM2_RIS_REML)

```

```

names(M_RIS_listB_R2)[1:4] <- NomsModels[5:8]

```

```

names(M_RIS_listB_R2)[(4+1):length(M_RIS_listB_R2)]<-paste(rep(NomsModels[5:8],each=2),rep(paste("RIS",

```

```

M_RIS_dfB_R2 <- data.frame(Model = names(M_RIS_listB_R2))

```

```

M_RIS_dfB_R2 <- data.frame(Model = names(M_RIS_listB_R2))

```

```

M_RIS_dfB_R2$R2 <- sapply(1:length(M_RIS_listB_R2),function(M){

```

```

  if (class(M_RIS_listB_R2[[M]])[1]=="lmm"){

```

```

    summary(M_RIS_listB_R2[[M]])$r.squared

```

```

  } else if (class(M_RIS_listB_R2[[M]])=="lme"){

```

```

    if(all(paste(attr(M_RIS_listB_R2[[M]]$terms,"predvars")[-c(1:2)]) ==

```

```

      paste(attr(M_RIS_listB_R2[[M]]$terms,"term.labels"))){

```

```

      rsquared(M_RIS_listB_R2[[M]])$Conditional

```

```

    } else {

```

```

      summary(lm(formula(paste(paste("M_RIS_listB_R2[[" ,M, "]"$data$",

```

```

        attr(M_RIS_listB_R2[[M]]$terms,"variables")[[2]],sep=""),

```

```

        "~", "M_RIS_listB_R2[[" ,M, "]"$fitted"),sep=""))) $r.squared

```

```

    }

```

```

  }

```

```

})

```

```

## Warning in paste(attr(M_RIS_listB_R2[[M]]$terms, "predvars")[-c(1:2)]) == :
## longer object length is not a multiple of shorter object length

```

```

## Warning in paste(attr(M_RIS_listB_R2[[M]]$terms, "predvars")[-c(1:2)]) == :
## longer object length is not a multiple of shorter object length

```

```

## Warning in paste(attr(M_RIS_listB_R2[[M]]$terms, "predvars")[-c(1:2)]) == :
## longer object length is not a multiple of shorter object length

```

```

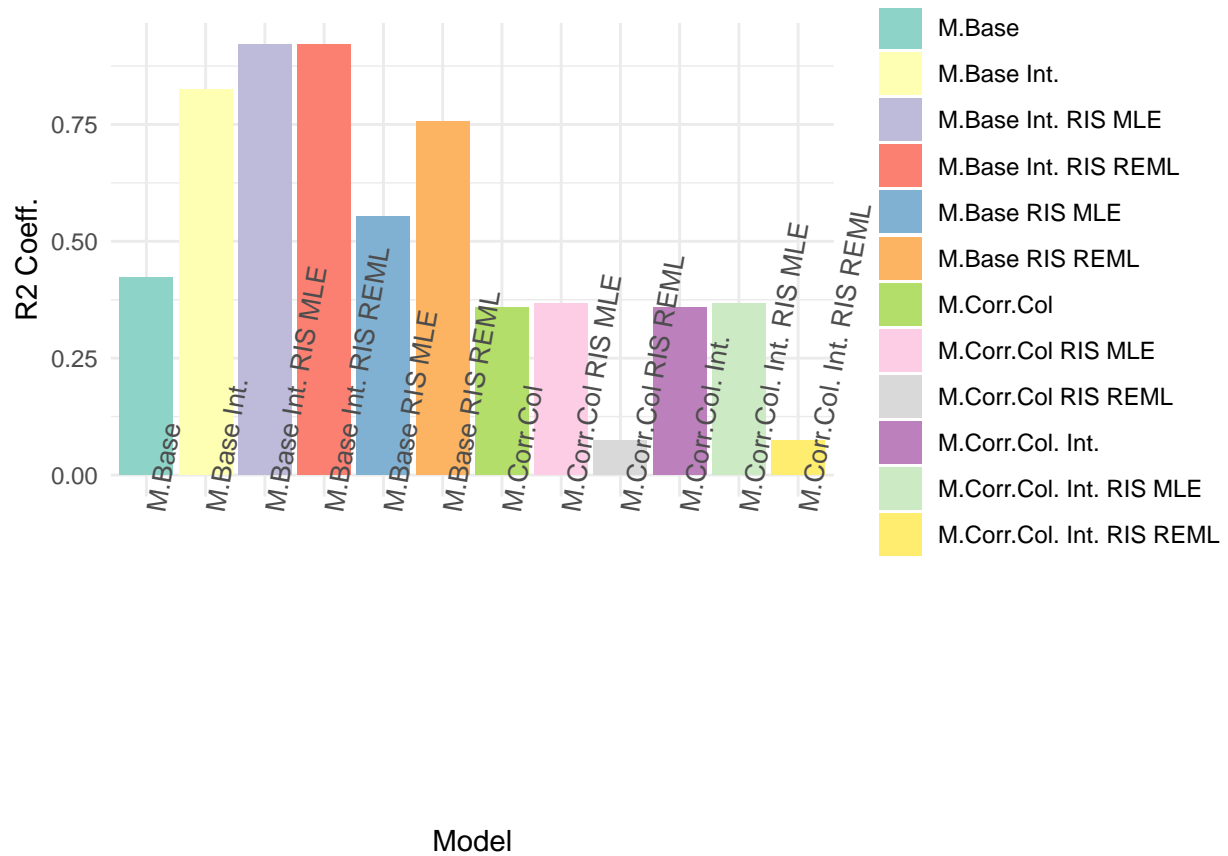
## Warning in paste(attr(M_RIS_listB_R2[[M]]$terms, "predvars")[-c(1:2)]) == :
## longer object length is not a multiple of shorter object length

```

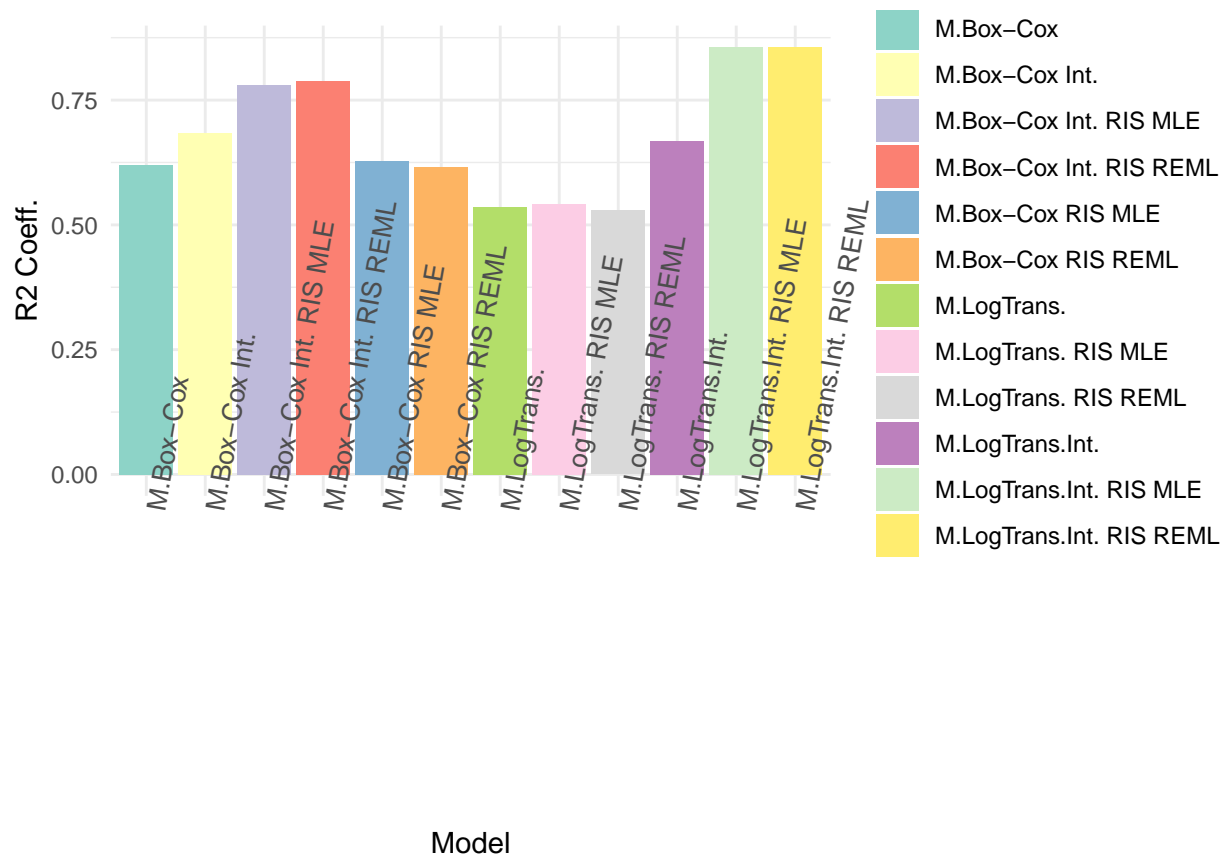


```
R2_B2 <- ggplot(data=M_RIS_dfB_R2, aes(x=Model, y=R2, fill=Model)) +
  geom_bar(stat="identity", position=position_dodge())+
  scale_fill_brewer(palette="Set3") + theme_minimal() +
  labs(y = "R2 Coeff.") +
  theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))
```

R2\_A2



R2\_B2



```

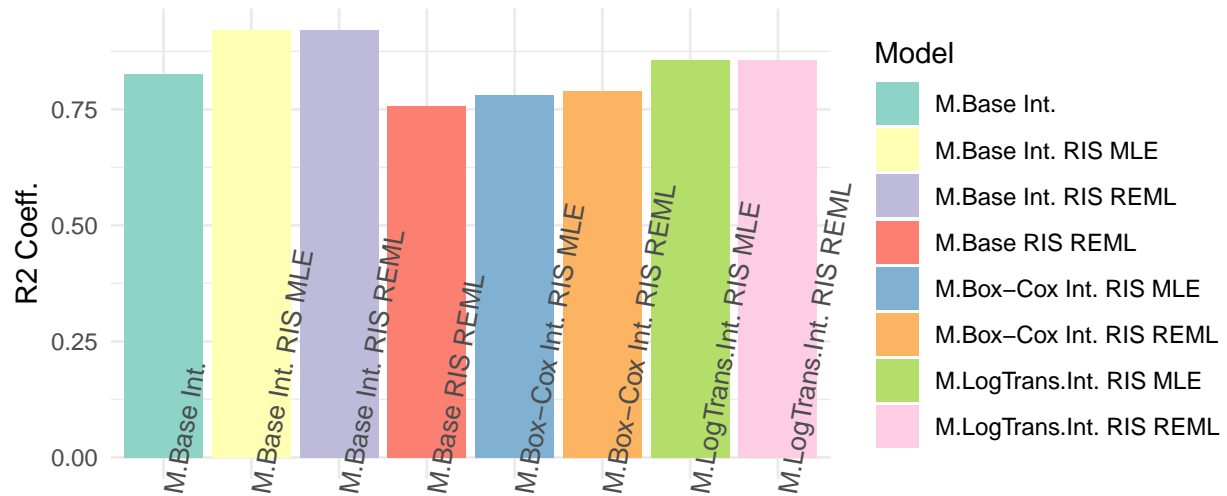
Model

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

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

R2_sel2

```

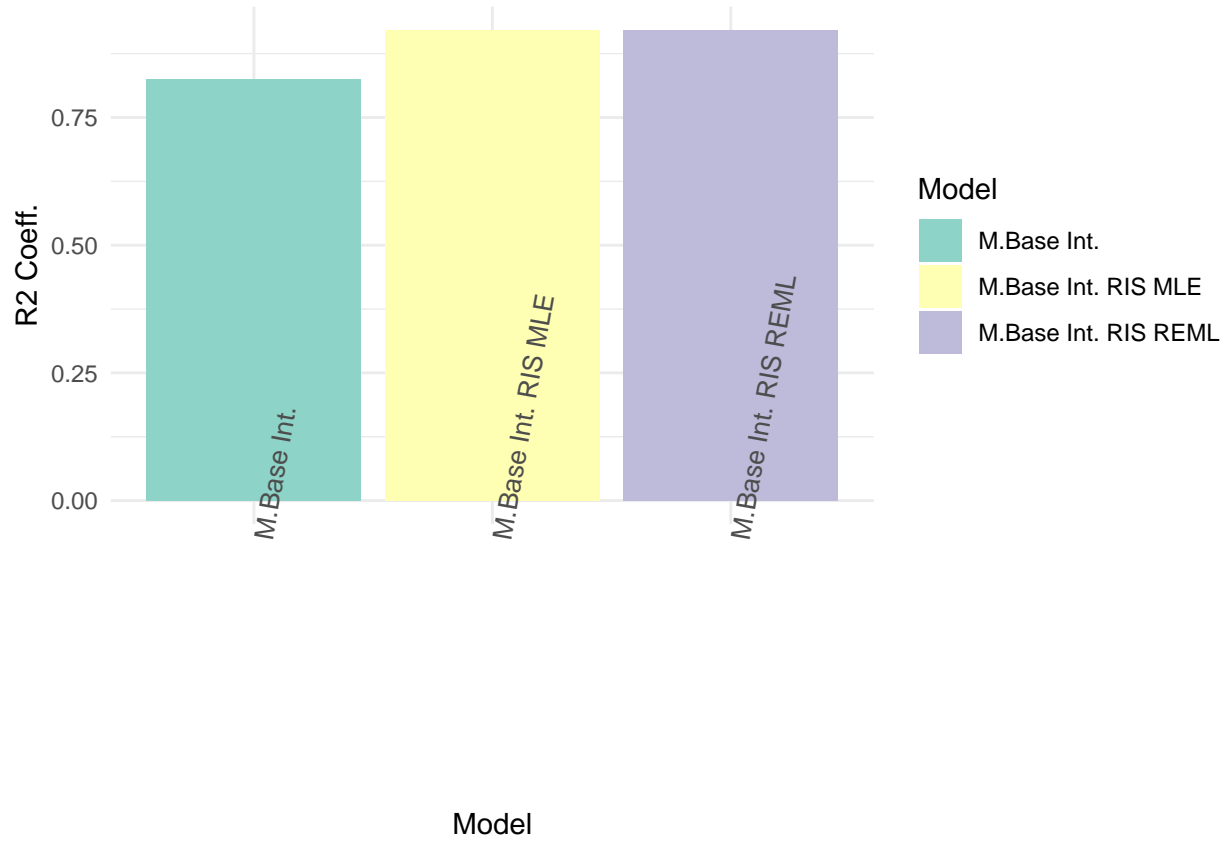


### Model

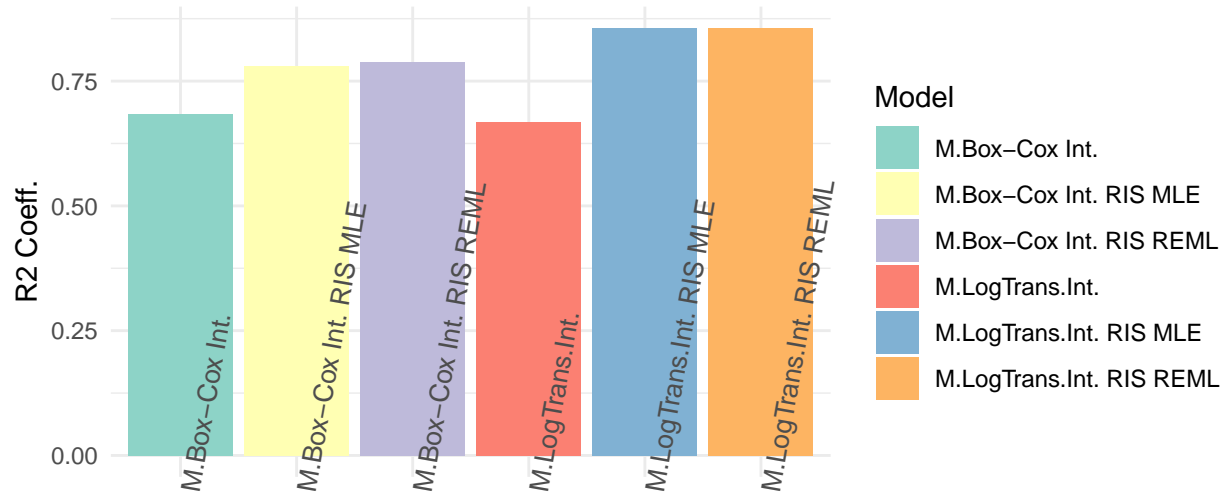
```
#Deixem només els models amb interacció
M_RIS_dFA_R2i <- M_RIS_dFA_R2[-c(1,3:6,9:12),]
R2_A2i <- ggplot(data=M_RIS_dFA_R2i, aes(x=Model, y=R2, fill=Model)) +
  geom_bar(stat="identity", position=position_dodge()) +
  scale_fill_brewer(palette="Set3") + theme_minimal() +
  labs(y = "R2 Coeff.") +
  theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))

M_RIS_dFB_R2i <- M_RIS_dFB_R2[-c(1,3,5:6,9:10),]
R2_B2i <- ggplot(data=M_RIS_dFB_R2i, aes(x=Model, y=R2, fill=Model)) +
  geom_bar(stat="identity", position=position_dodge()) +
  scale_fill_brewer(palette="Set3") + theme_minimal() +
  labs(y = "R2 Coeff.") +
  theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))

R2_A2i
```



R2\_B2i

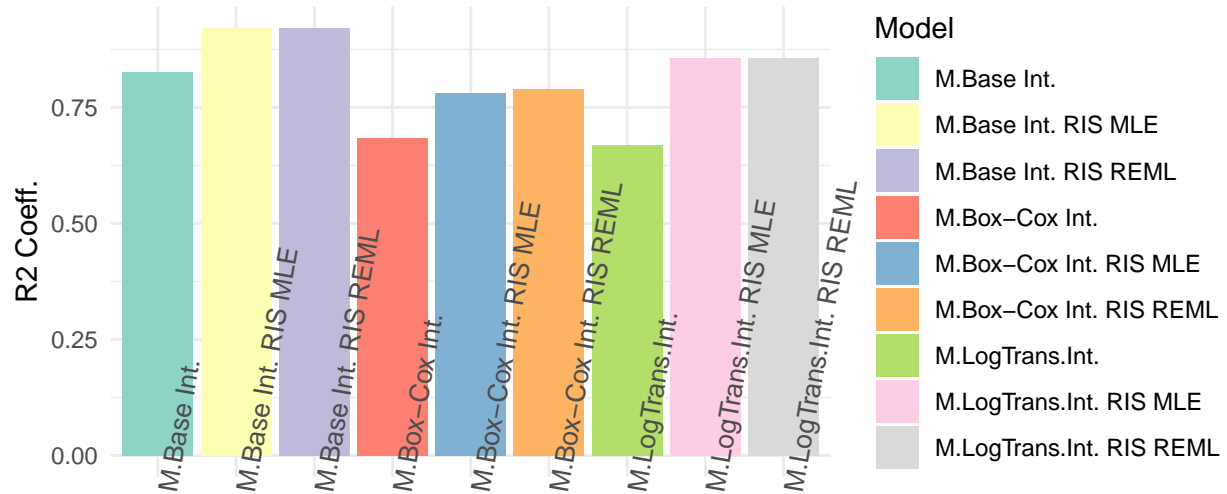


### Model

```
M_RIS_df_sel_R2i <- rbind(M_RIS_dfA_R2i,M_RIS_dfB_R2i)

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

R2_sel2i
```



## Model

Sí que s'observa aquesta vegada una millora més aparent en el coeficient  $R^2$ , concretament si es pren els 4 models amb un coeficient més alt serien M.Base Int. RIS REML, M.Base Int. RIS MLE, M.LogTrans.Int. RIS REML, M.LogTrans.Int. RIS MLE amb coeficients de 0.9211, 0.9208, 0.8564, 0.8556 respectivament. Curiosament segueixen essent els que denotaven millora amb l'efecte aleatori.

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

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

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

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

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

```
Lake1IntrIS_MLE.SMO.sdres <- residuals(intM0_RIS_MLE,type = "pearson")
Lake1IntrIS_REML.SMO.sdres <- residuals(intM0_RIS_REML,type = "pearson")
par(mfrow=c(1,3))
```

```

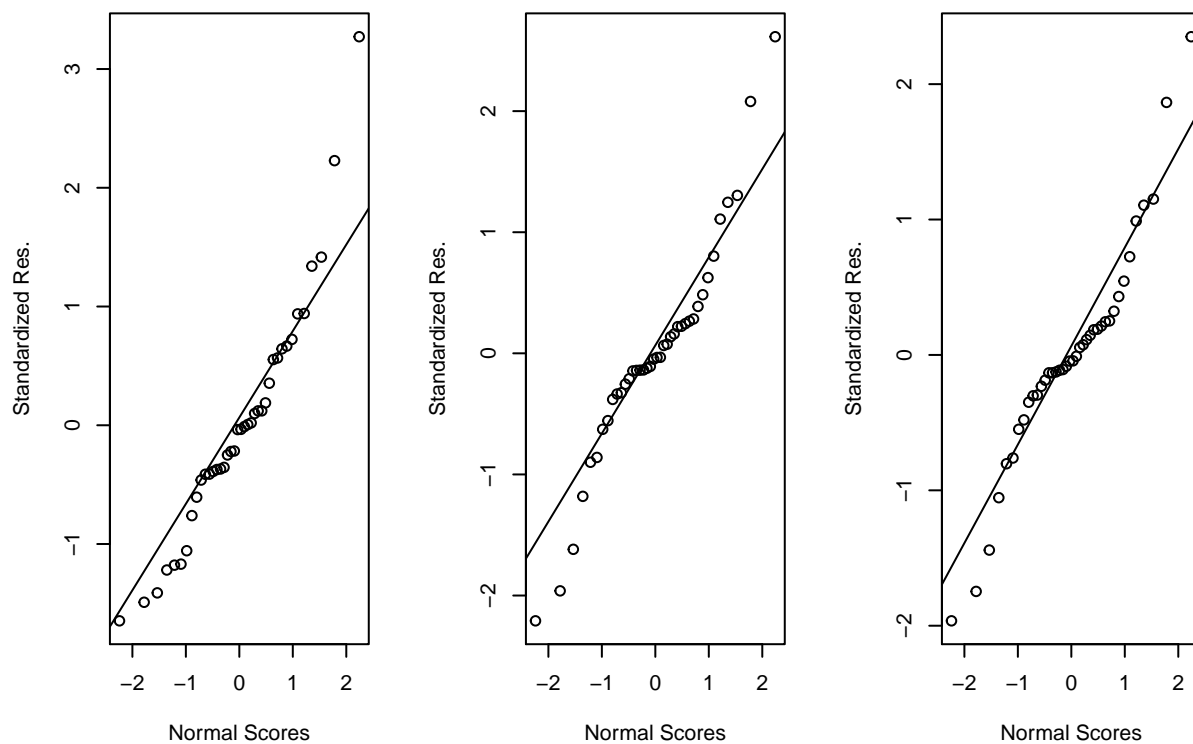
qqnorm(Lake1Int.SMO.sdres,
       ylab="Standardized Res.",
       xlab="Normal Scores",
       main="M. Original Int. - Standardized Res.")
qqline(Lake1Int.SMO.sdres)

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

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

```

**M. Original Int. – Standardized Residuals**



S0Int

```

##
## Shapiro-Wilk normality test
##
## data: intM0$residuals
## W = 0.92773, p-value = 0.01359

```

```

(SOIntrRIS_MLE <- shapiro.test(intMO_RIS_MLE$residuals))

##
## Shapiro-Wilk normality test
##
## data:  intMO_RIS_MLE$residuals
## W = 0.93961, p-value = 0.0009178
(SOIntrRIS_REML <- shapiro.test(intMO_RIS_REML$residuals))

##
## Shapiro-Wilk normality test
##
## data:  intMO_RIS_REML$residuals
## W = 0.9398, p-value = 0.0009395
Lake1IntrRIS_MLE.SM1.sdres <- residuals(intM1_RIS_MLE,type = "pearson")
Lake1IntrRIS_REML.SM1.sdres <- residuals(intM1_RIS_REML,type = "pearson")
par(mfrow=c(1,3))

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

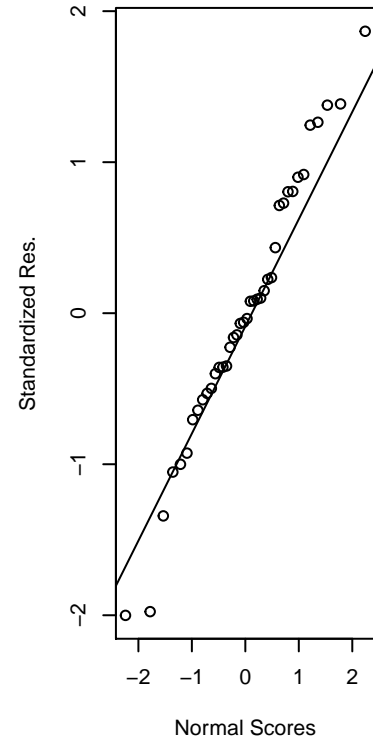
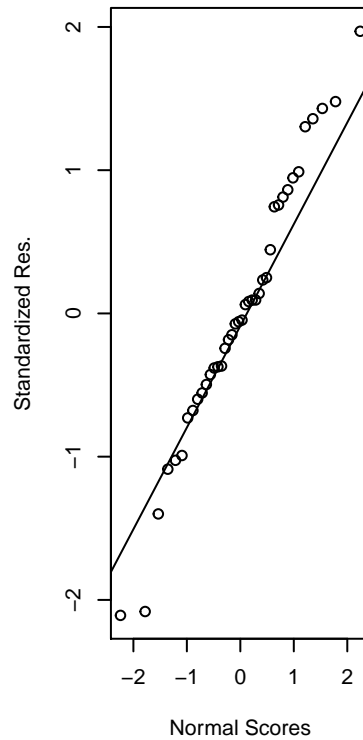
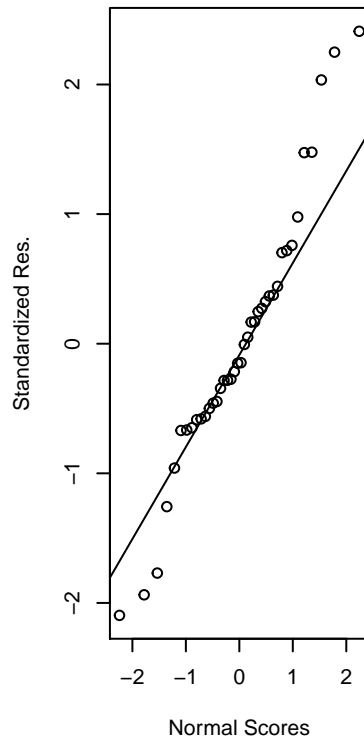
qqnorm(Lake1IntrRIS_MLE.SM1.sdres,
       ylab="Standardized Res.",
       xlab="Normal Scores",
       main="M. Logt Int RIS MLE - Standardized Res.")
qqline(Lake1Int.SM1.sdres)

qqnorm(Lake1IntrRIS_REML.SM1.sdres,
       ylab="Standardized Res.",
       xlab="Normal Scores",
       main="M. Logt Int RIS REML - Standardized Res.")
qqline(Lake1Int.SM1.sdres)

```



**M. Logt Int. – Standardized Re. Logt Int RIS MLE – Standardize Logt Int RIS REML – Standardize**



S1Int

```
##
## Shapiro-Wilk normality test
##
## data: intM1$residuals
## W = 0.9557, p-value = 0.1192
```

```
(S1IntRIS_MLE <- shapiro.test(intM1_RIS_MLE$residuals))
```

```
##
## Shapiro-Wilk normality test
##
## data: intM1_RIS_MLE$residuals
## W = 0.95018, p-value = 0.003532
```

```
(S1IntRIS_REML <- shapiro.test(intM1_RIS_REML$residuals))
```

```
##
## Shapiro-Wilk normality test
##
## data: intM1_RIS_REML$residuals
## W = 0.95009, p-value = 0.003488
```

```
Lake1IntRIS_MLE.SM2.sdres <- residuals(intM2_RIS_MLE,type = "pearson")
Lake1IntRIS_REML.SM2.sdres <- residuals(intM2_RIS_REML,type = "pearson")
par(mfrow=c(1,3))
```

```
qqnorm(Lake1Int.SM2.sdres,
```

```

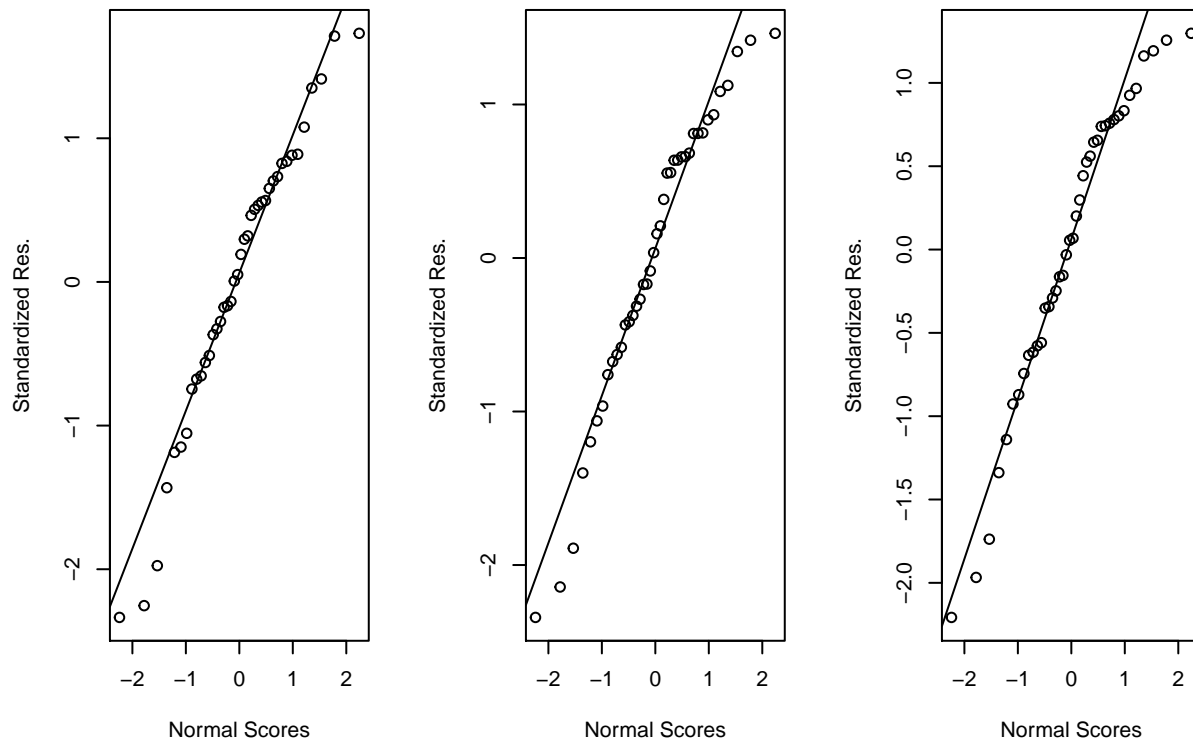
      ylab="Standardized Res.",
      xlab="Normal Scores",
      main="M. BCt Int. - Standardized Res.")
qqline(Lake1Int.SM2.sdres)

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

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

```

**M. BCt Int. – Standardized Res. BCt Int RIS MLE – Standardized BCt Int RIS REML – Standardize**



S2Int

```

##
## Shapiro-Wilk normality test
##
## data: intM2$residuals
## W = 0.96332, p-value = 0.2172

```

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

```
##  
## Shapiro-Wilk normality test  
##  
## data: intM2_RIS_MLE$residuals  
## W = 0.96907, p-value = 0.04949
```

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

```
##  
## Shapiro-Wilk normality test  
##  
## data: intM2_RIS_REML$residuals  
## W = 0.97436, p-value = 0.1078
```

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

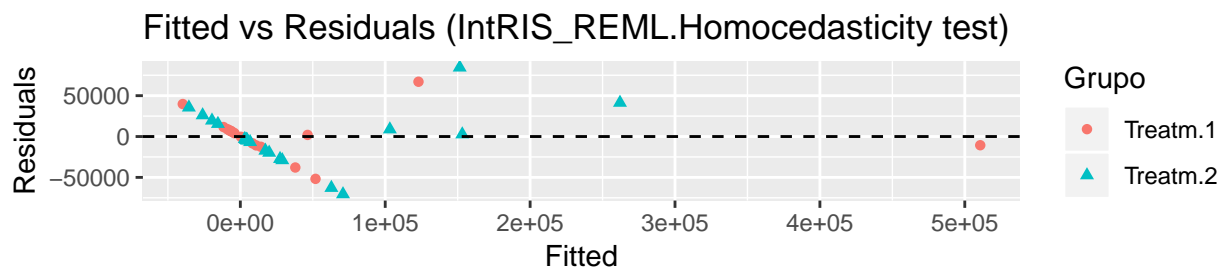
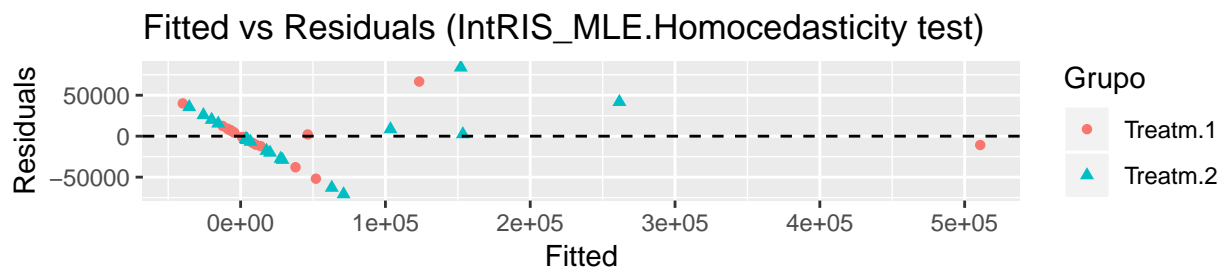
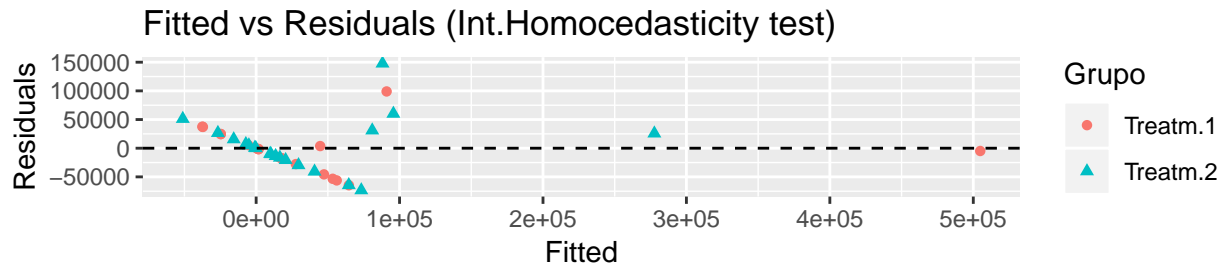
```
pHIntRIS_MLE01 <- ggplot(LAKE1, aes(x=fitted(intMO_RIS_MLE), y=residuals(intMO_RIS_MLE),  
                                shape=Grupo, color=Grupo)) +  
  geom_jitter(position=position_jitter(1), cex=1.5) +  
  geom_hline(yintercept=0, linetype="dashed", color = "black") +  
  labs(title="Fitted vs Residuals (IntRIS_MLE.Homocedasticity test)",  
        x="Fitted", y = "Residuals")
```

```
pHIntRIS_MLE02 <-ggplot(LAKE1, aes(x=fitted(intMO_RIS_MLE), y=sqrt(abs(residuals(intMO_RIS_MLE))),  
                                shape=Grupo, color=Grupo)) +  
  geom_jitter(position=position_jitter(1), cex=1.5) +  
  geom_hline(yintercept=0, linetype="dashed", color = "black") +  
  labs(title="Fitted vs Residuals (IntRIS_MLE.Homocedasticity test)",  
        x="Fitted", y = expression(sqrt("|residuals|")))
```

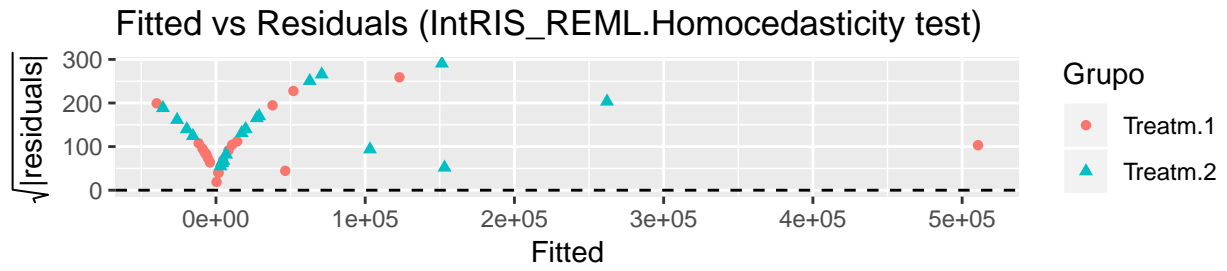
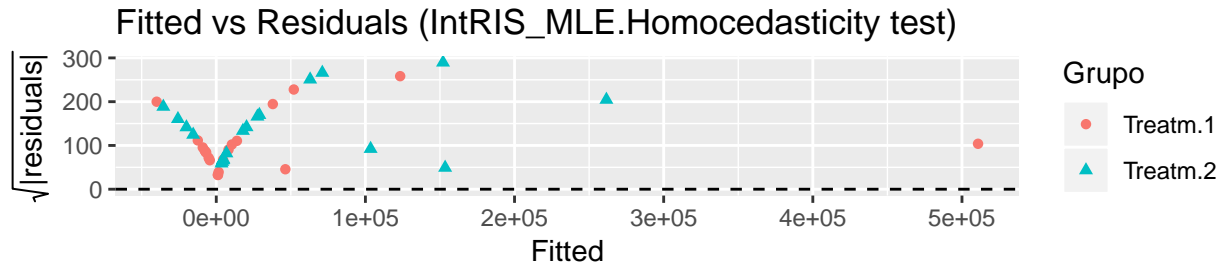
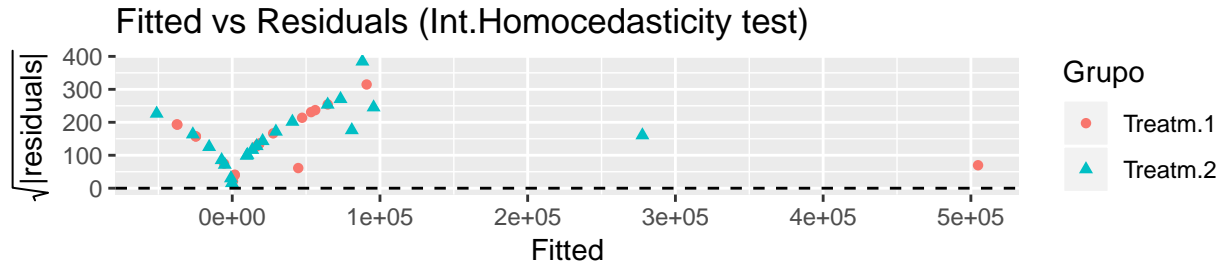
```
pHIntRIS_REML01 <- ggplot(LAKE1, aes(x=fitted(intMO_RIS_REML), y=residuals(intMO_RIS_REML),  
                                shape=Grupo, color=Grupo)) +  
  geom_jitter(position=position_jitter(1), cex=1.5) +  
  geom_hline(yintercept=0, linetype="dashed", color = "black") +  
  labs(title="Fitted vs Residuals (IntRIS_REML.Homocedasticity test)",  
        x="Fitted", y = "Residuals")
```

```
pHIntRIS_REML02 <-ggplot(LAKE1, aes(x=fitted(intMO_RIS_REML), y=sqrt(abs(residuals(intMO_RIS_REML))),  
                                shape=Grupo, color=Grupo)) +  
  geom_jitter(position=position_jitter(1), cex=1.5) +  
  geom_hline(yintercept=0, linetype="dashed", color = "black") +  
  labs(title="Fitted vs Residuals (IntRIS_REML.Homocedasticity test)",  
        x="Fitted", y = expression(sqrt("|residuals|")))
```

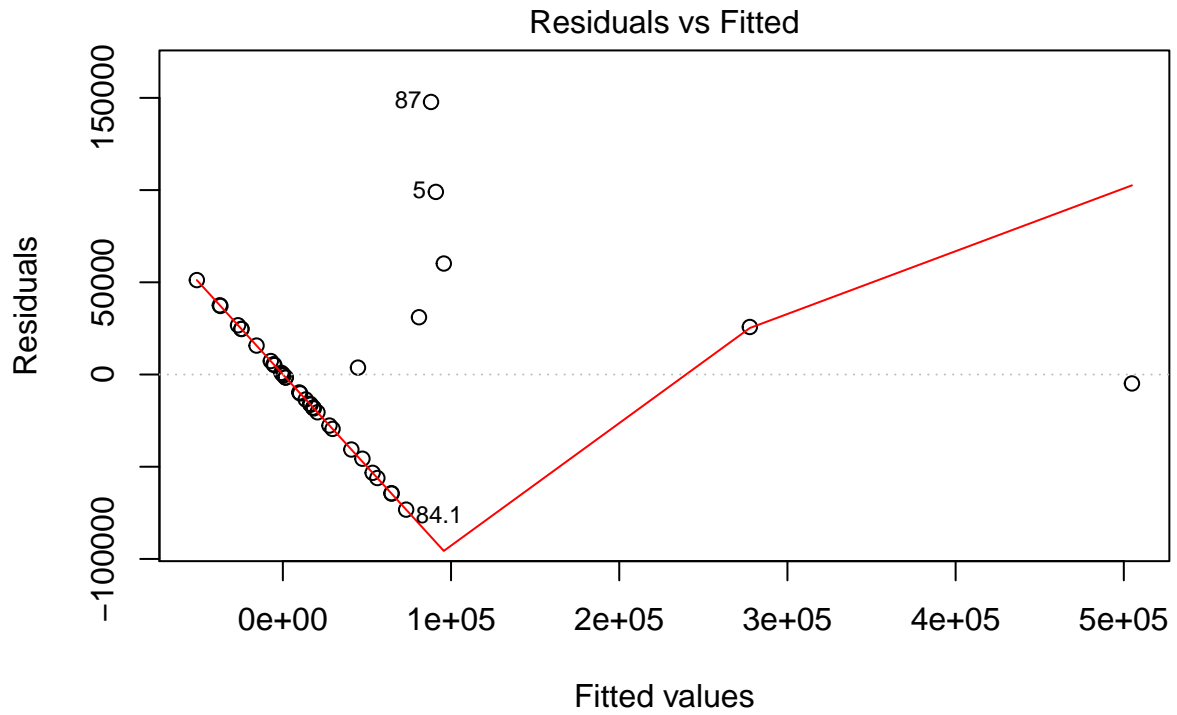
```
grid.arrange(pHInt01, pHIntRIS_MLE01,pHIntRIS_REML01,nrow=3)
```



```
grid.arrange(pHInt02, pHIntRIS_MLE02, pHIntRIS_REML02, nrow=3)
```

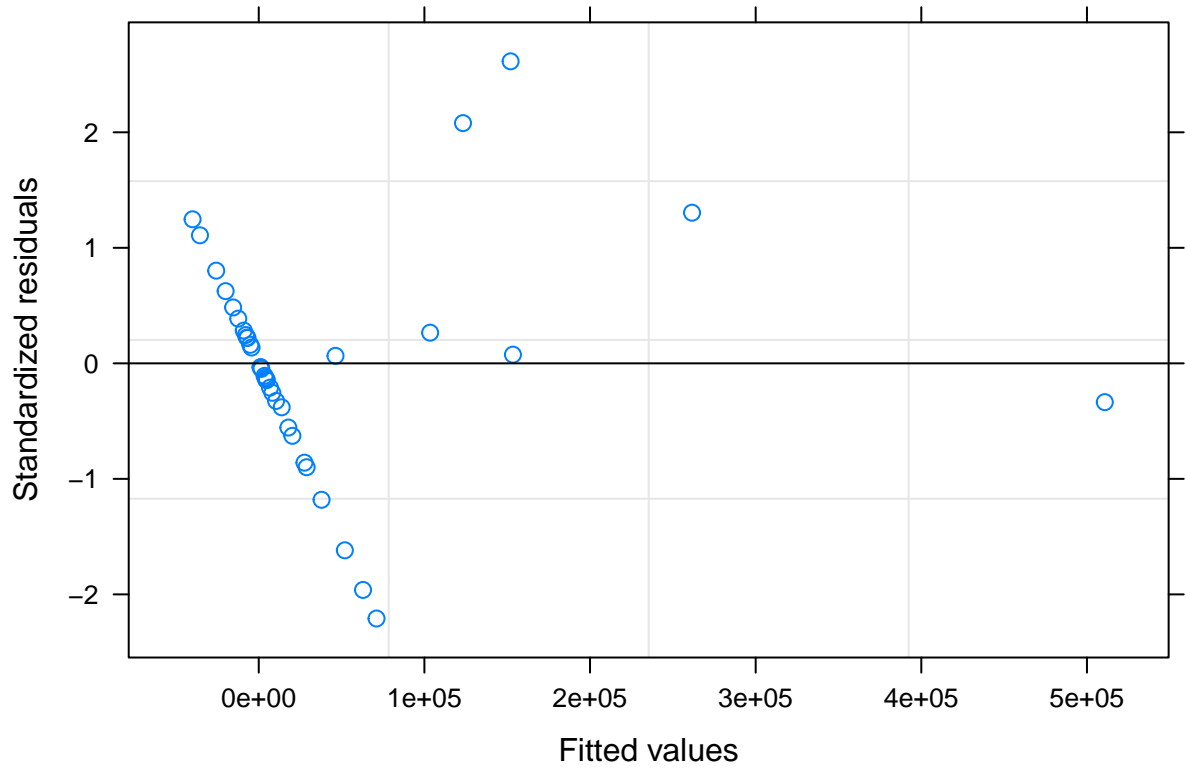


```
plot(intM0, which=1)
```

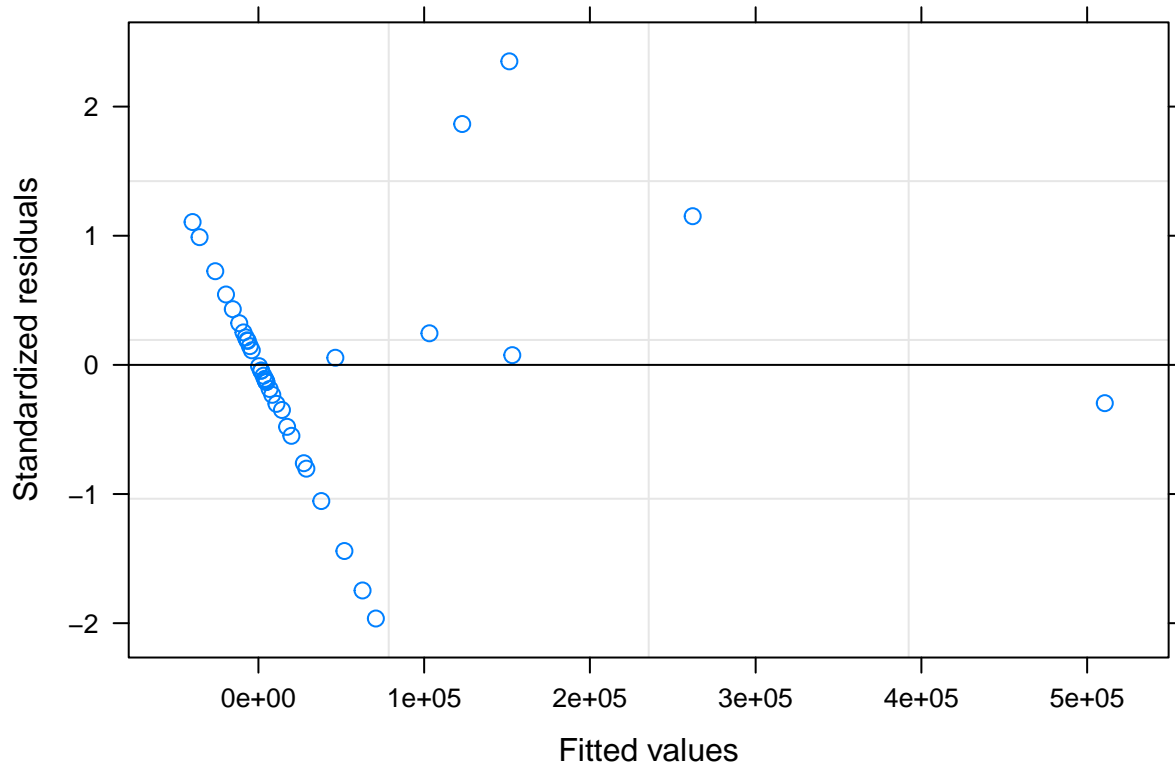


lm(CargaViral ~ Tiempo + CD4P + Cloro + Tiempo:CD4P + Tiempo:Cloro + CD4P:C

```
plot(intM0_RIS_MLE, which=1)
```



```
plot(intM0_RIS_REML,which=1)
```



```
pHIntRIS_MLE11 <- ggplot(LAKE1, aes(x=fitted(intM1_RIS_MLE), y=residuals(intM1_RIS_MLE),
                                   shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Logt.IntRIS_MLE.Homocedasticity test)",
        x="Fitted", y = "Residuals")
```

```
pHIntRIS_MLE12 <-ggplot(LAKE1, aes(x=fitted(intM1_RIS_MLE), y=sqrt(abs(residuals(intM1_RIS_MLE))),
                                   shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Logt.IntRIS_MLE.Homocedasticity test)",
        x="Fitted", y = expression(sqrt("|residuals|")))
```

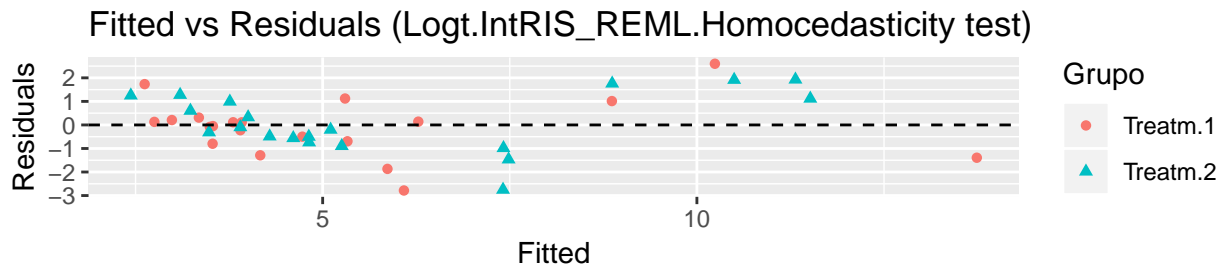
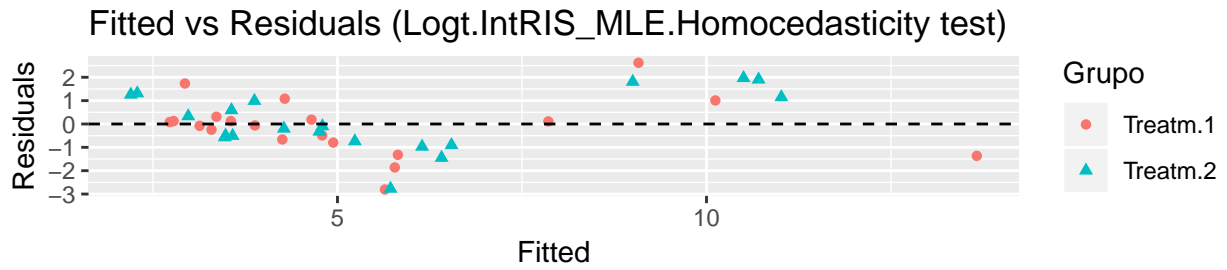
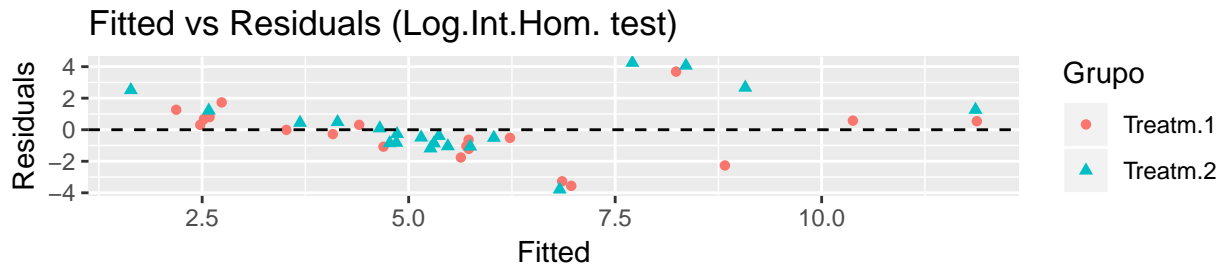
```
pHIntRIS_REML11 <- ggplot(LAKE1, aes(x=fitted(intM1_RIS_REML), y=residuals(intM1_RIS_REML),
                                   shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Logt.IntRIS_REML.Homocedasticity test)",
        x="Fitted", y = "Residuals")
```

```
pHIntRIS_REML12 <-ggplot(LAKE1, aes(x=fitted(intM1_RIS_REML), y=sqrt(abs(residuals(intM1_RIS_REML))),
                                   shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (Logt.IntRIS_REML.Homocedasticity test)",
```

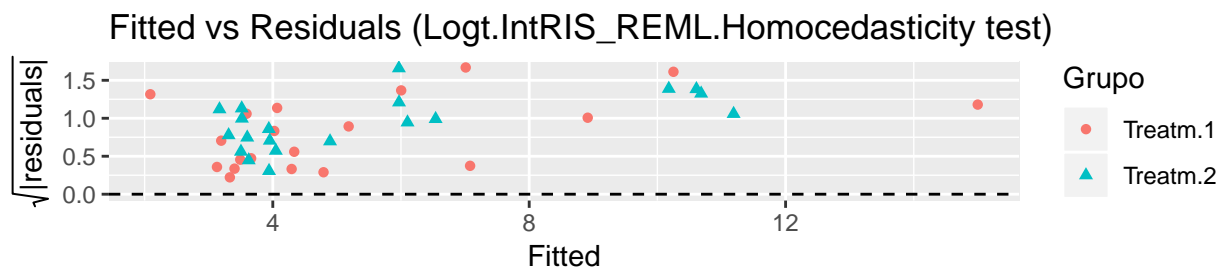
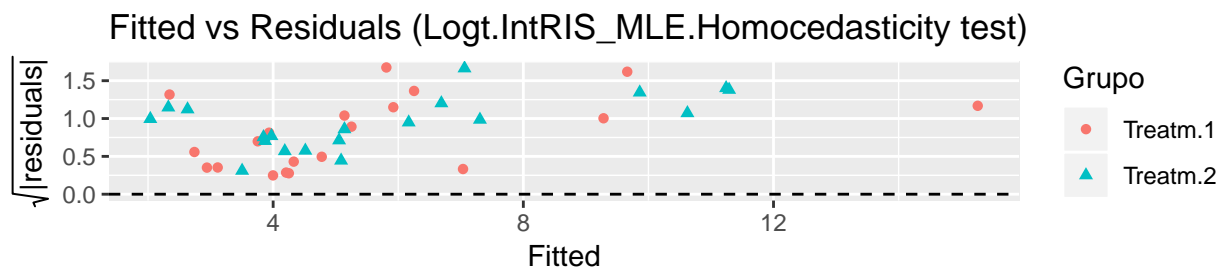
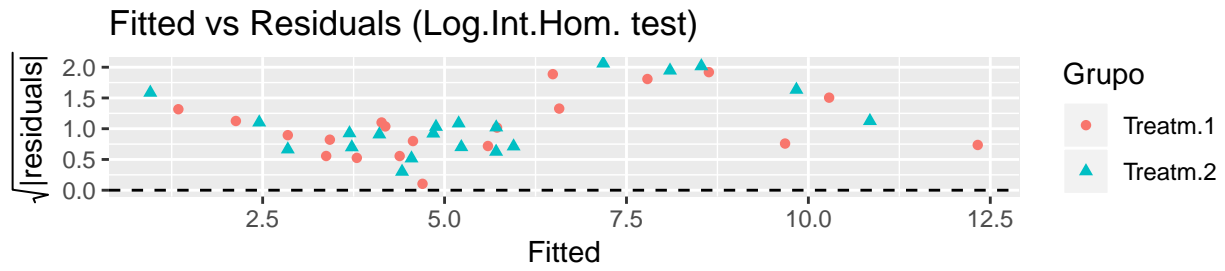


```
x="Fitted", y = expression(sqrt("|residuals|"))
```

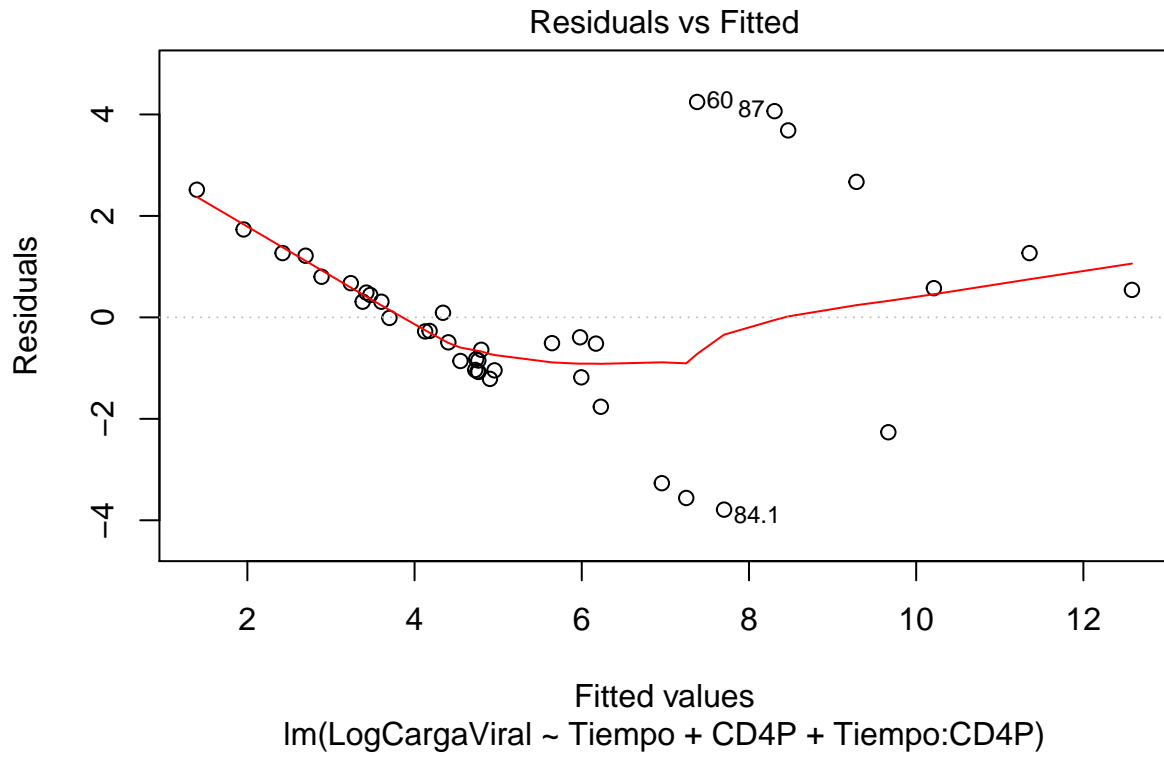
```
grid.arrange(pHInt11, pHIntRIS_MLE11,pHIntRIS_REML11,nrow=3)
```



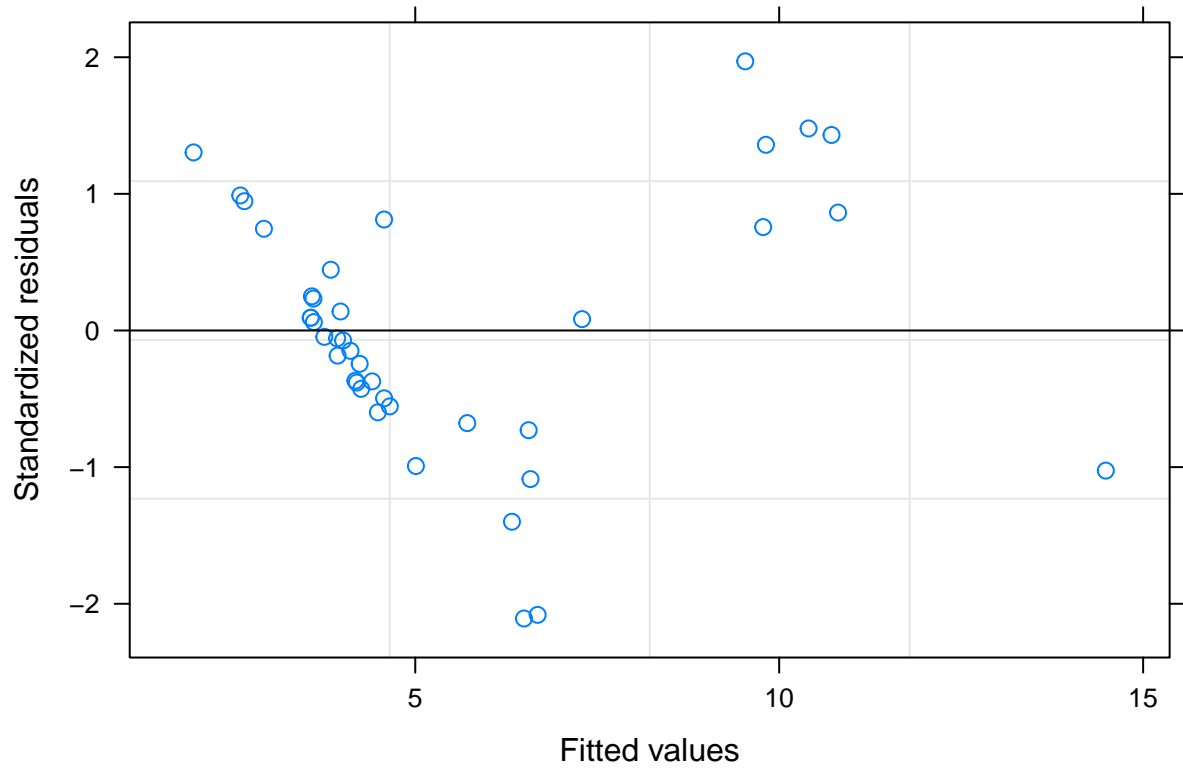
```
grid.arrange(pHInt12, pHIntRIS_MLE12,pHIntRIS_REML12,nrow=3)
```



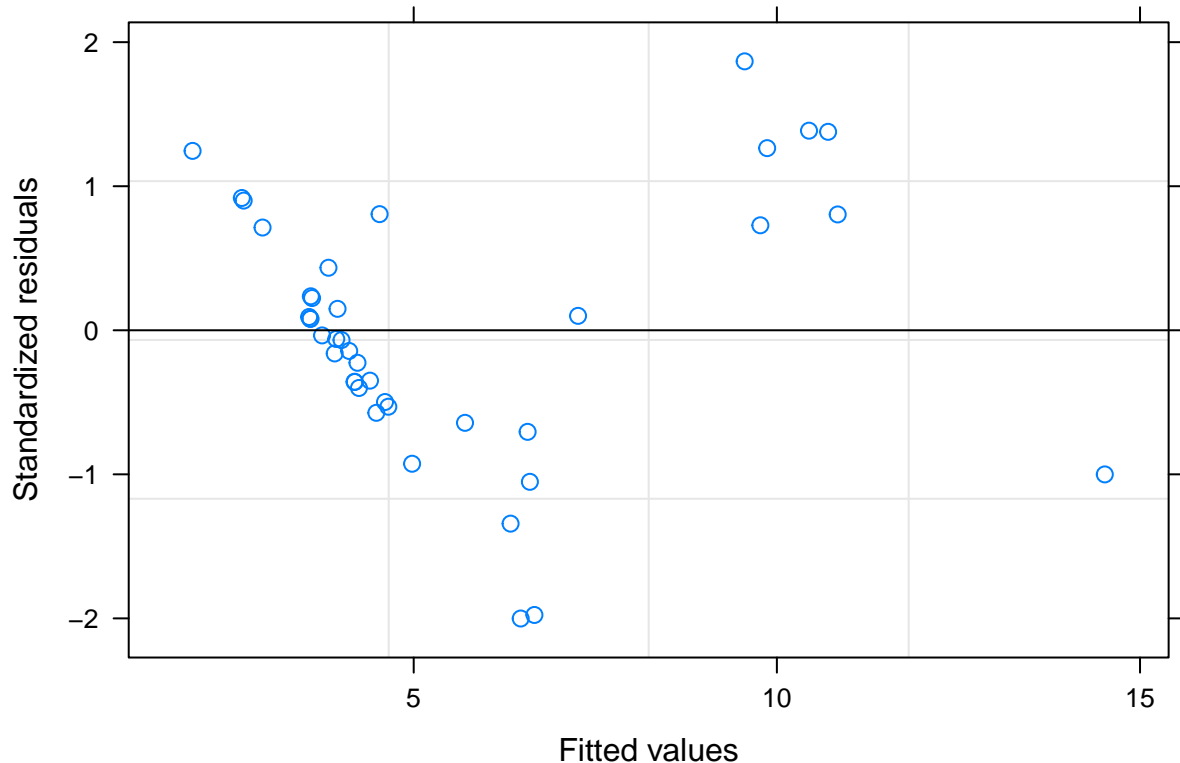
```
plot(intM1, which=1)
```



```
plot(intM1_RIS_MLE, which=1)
```



```
plot(intM1_RIS_REML, which=1)
```



```
pHIntRIS_MLE21 <- ggplot(LAKE1, aes(x=fitted(intM2_RIS_MLE), y=residuals(intM2_RIS_MLE),
                                   shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted vs Residuals (BCT.IntRIS_MLE.Homocedasticity test)",
       x="Fitted", y = "Residuals")
```

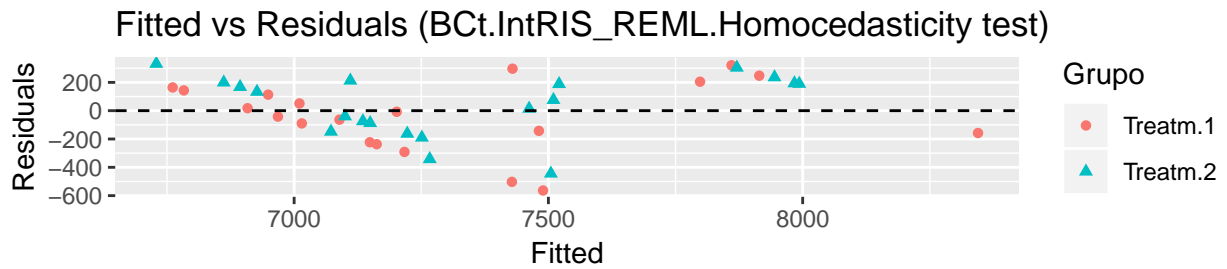
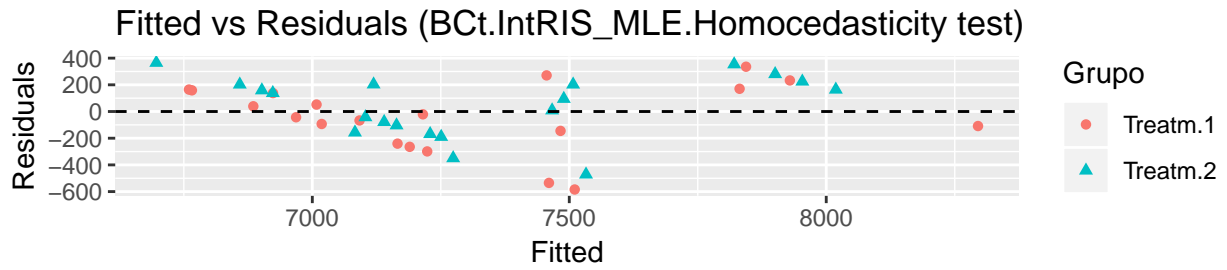
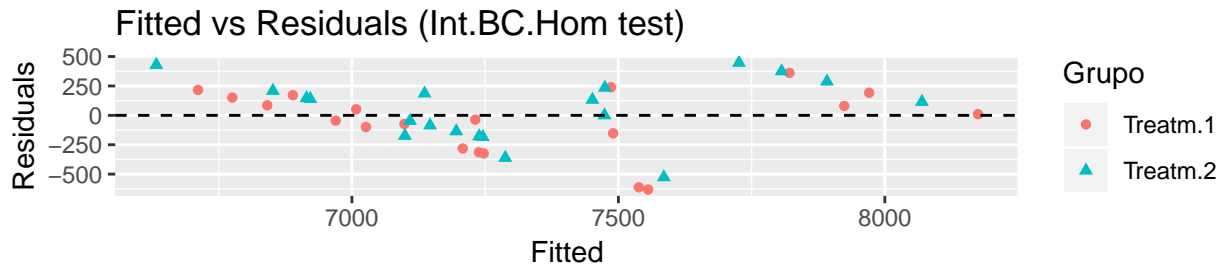
```
pHIntRIS_MLE22 <-ggplot(LAKE1, aes(x=fitted(intM2_RIS_MLE), y=sqrt(abs(residuals(intM2_RIS_MLE))),
                                   shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted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

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

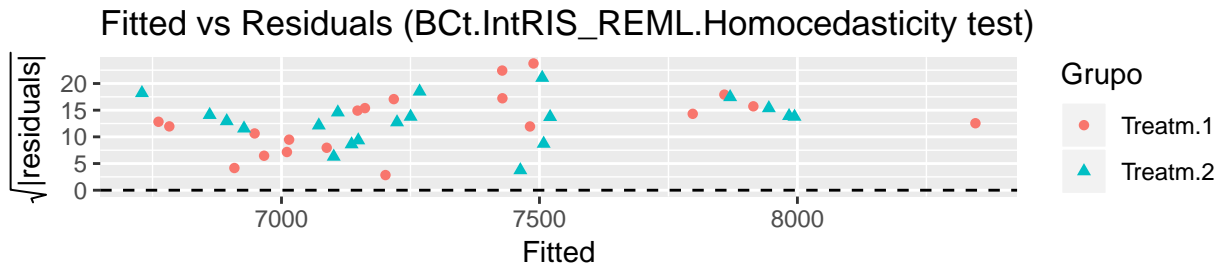
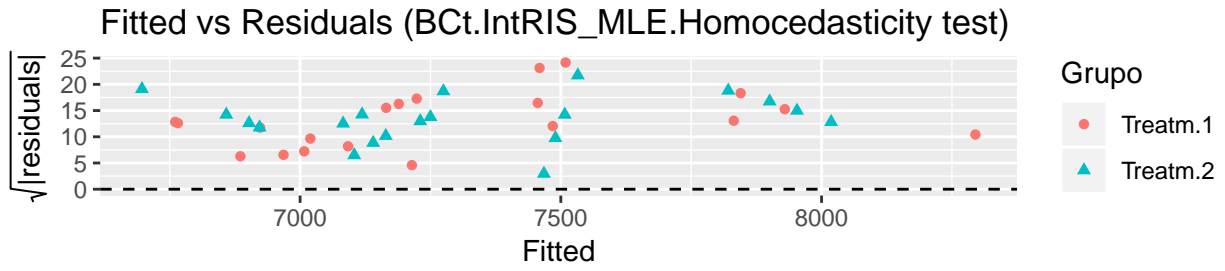
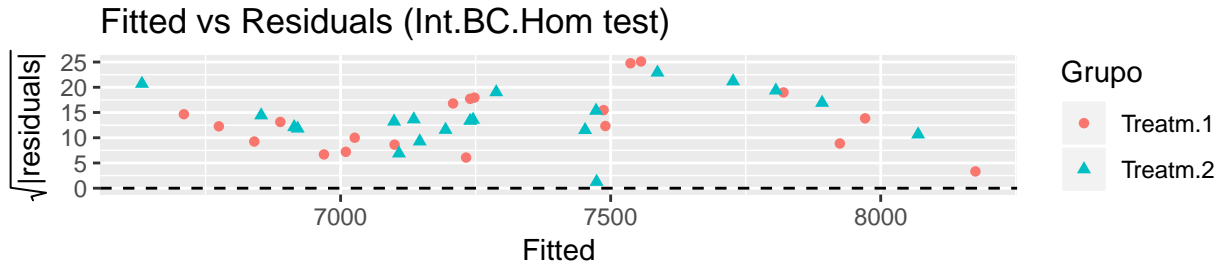
```
pHIntRIS_REML22 <-ggplot(LAKE1, aes(x=fitted(intM2_RIS_REML), y=sqrt(abs(residuals(intM2_RIS_REML))),
                                   shape=Grupo, color=Grupo)) +
  geom_jitter(position=position_jitter(1), cex=1.5) +
  geom_hline(yintercept=0, linetype="dashed", color = "black") +
  labs(title="Fitted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

```
x="Fitted", y = expression(sqrt("|residuals|"))
```

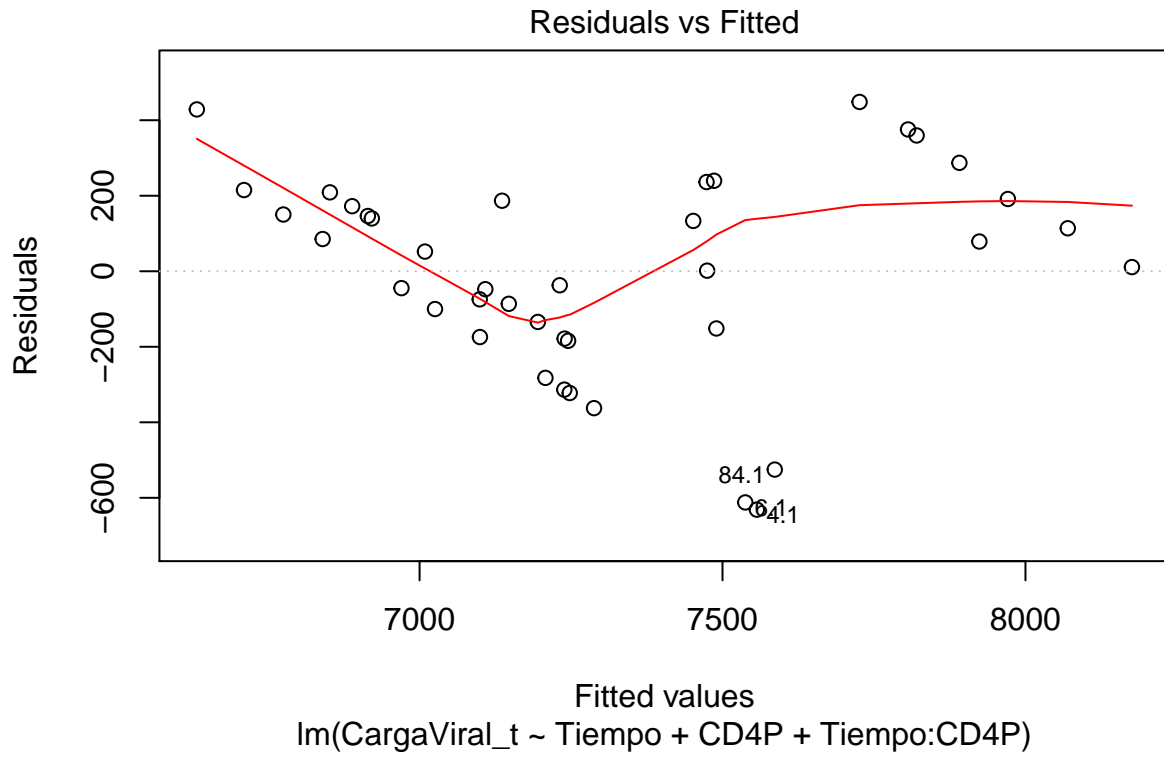
```
grid.arrange(pHInt21, pHIntRIS_MLE21,pHIntRIS_REML21,nrow=3)
```



```
grid.arrange(pHInt22, pHIntRIS_MLE22,pHIntRIS_REML22,nrow=3)
```

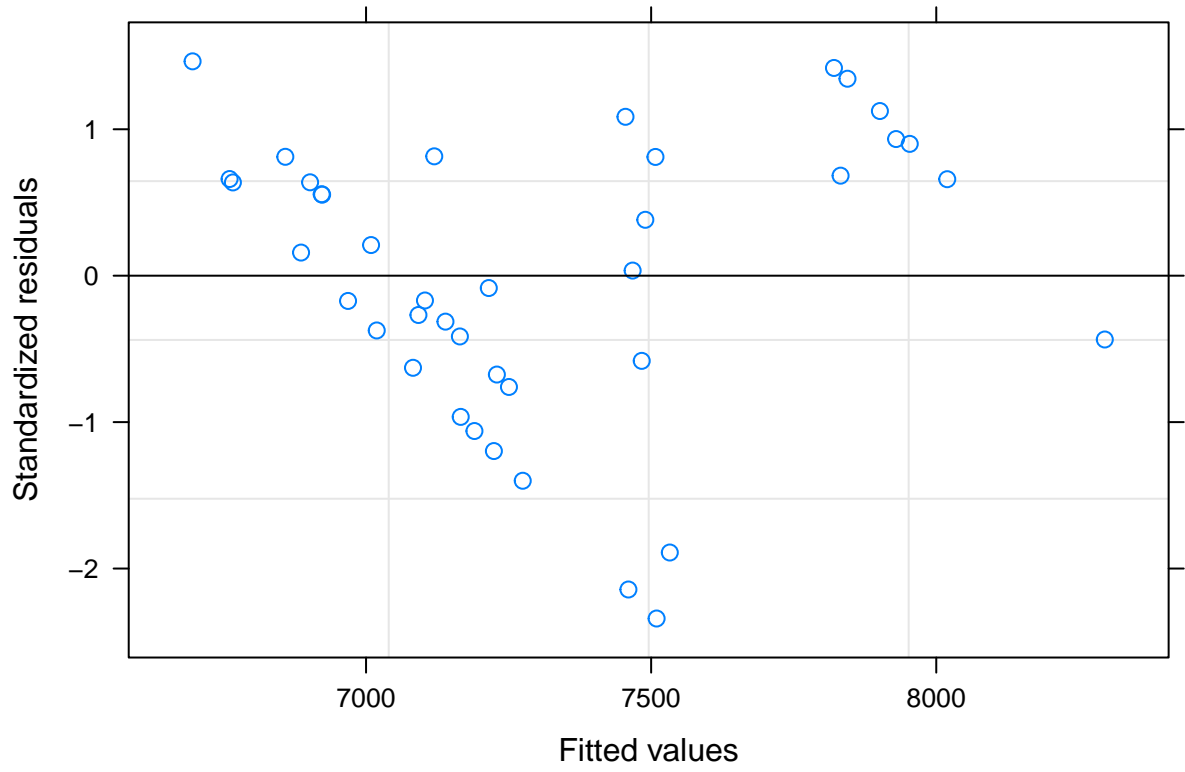


```
plot(intM2, which=1)
```

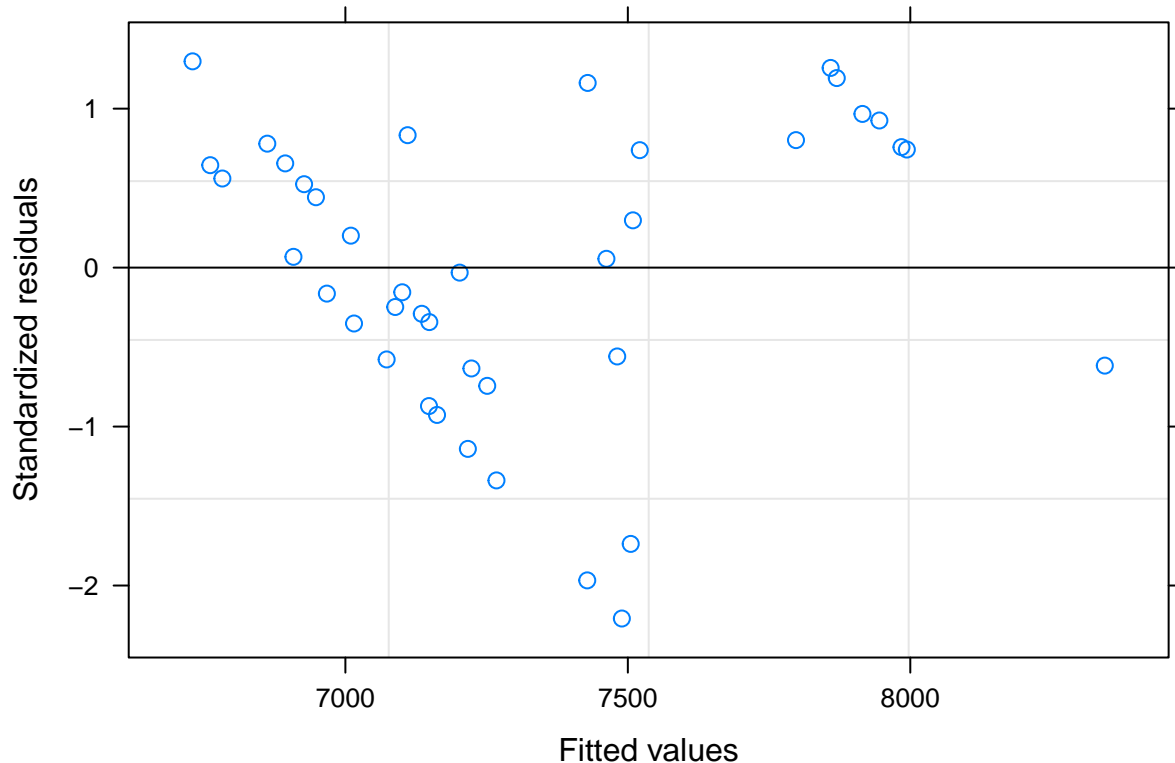


```
plot(intM2_RIS_MLE, which=1)
```





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



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

## AJUST FUNCIONS DE COVARIÀNCIA

### Ajust models

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

```
Comp_W_LM <- function(fLME,WVI,WVE,WVP) # WVI;WVE;WVP Per decidir quines funcions s'apliquen essent-> 0
{
  baseML <- update(fLME, control=ctrl, method="ML")
  baseREML <- update(fLME, control=ctrl, method="REML")
  l <- list()
  fMLk <- "IS"
  fREk <- "RIS"

  if(WVI==1 | WVI==3){
    ML_I_RI_WVI_sMLE <- summary(ML_I_RI_WVI_MLE <- update(baseML, weights = varIdent(form = ~Tiempo))
    le <- length(l)+1
    l[[le]] <- ML_I_RI_WVI_sMLE
    names(l)[le] <- paste(fMLk,"MLE ",fREk,"VarIdent")
    le <- length(l)+1
  }
}
```

```

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

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

if(WVE==2 | WVE==3){
ML_I_RI_WVE_sREML <- summary(ML_I_RI_WVE_REML <- update(baseREML, weights = varExp(form = ~Tiempo))
le <- length(l)+1
l[[le]] <- ML_I_RI_WVE_sREML
names(1)[le] <- paste(fMLk,"REML ",fREk,"VarExp")
le <- length(l)+1
l[[le]] <- ML_I_RI_WVE_REML
names(1)[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 te

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

return(l)
}

```

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

```

#Models.sel <- list(intM0_RIS_MLE,intM0_RIS_REML, intM1_RIS_MLE,intM1_RIS_REML, intM2_RIS_MLE,intM2_RIS
intM0_RIS_W <- Comp_W_LM(intM0_RIS_MLE,3,3,0)

```

```

## Warning in logLik.reStruct(object, conLin): Singular precision matrix in

```

```

## level -1, block 1

## Warning in logLik.reStruct(object, conLin): Singular precision matrix in
## level -1, block 1

## Warning in logLik.reStruct(object, conLin): Singular precision matrix in
## level -1, block 1

## Warning in logLik.reStruct(object, conLin): Singular precision matrix in
## level -1, block 1

## Warning in logLik.reStruct(object, conLin): Singular precision matrix in
## level -1, block 1

## Warning in logLik.reStruct(object, conLin): Singular precision matrix in
## level -1, block 1

## Warning in logLik.reStruct(object, conLin): Singular precision matrix in
## level -1, block 1

intM1_RIS_W <- Comp_W_LM(intM1_RIS_MLE,3,3,3)

## Warning in logLik.reStruct(object, conLin): Singular precision matrix in
## level -1, block 1

## Warning in logLik.reStruct(object, conLin): Singular precision matrix in
## level -1, block 1

## Warning in logLik.reStruct(object, conLin): Singular precision matrix in
## level -1, block 1

## Warning in logLik.reStruct(object, conLin): Singular precision matrix in
## level -1, block 1

## Warning in logLik.reStruct(object, conLin): Singular precision matrix in
## level -1, block 1

## Warning in logLik.reStruct(object, conLin): Singular precision matrix in
## level -1, block 1

## Warning in logLik.reStruct(object, conLin): Singular precision matrix in
## level -1, block 1

## Warning in logLik.reStruct(object, conLin): Singular precision matrix in
## level -1, block 1

intM2_RIS_W <- Comp_W_LM(intM2_RIS_MLE,3,3,2)

## Warning in logLik.reStruct(object, conLin): Singular precision matrix in

```

```
## level -1, block 1

## Warning in logLik.reStruct(object, conLin): Singular precision matrix in
## level -1, block 1

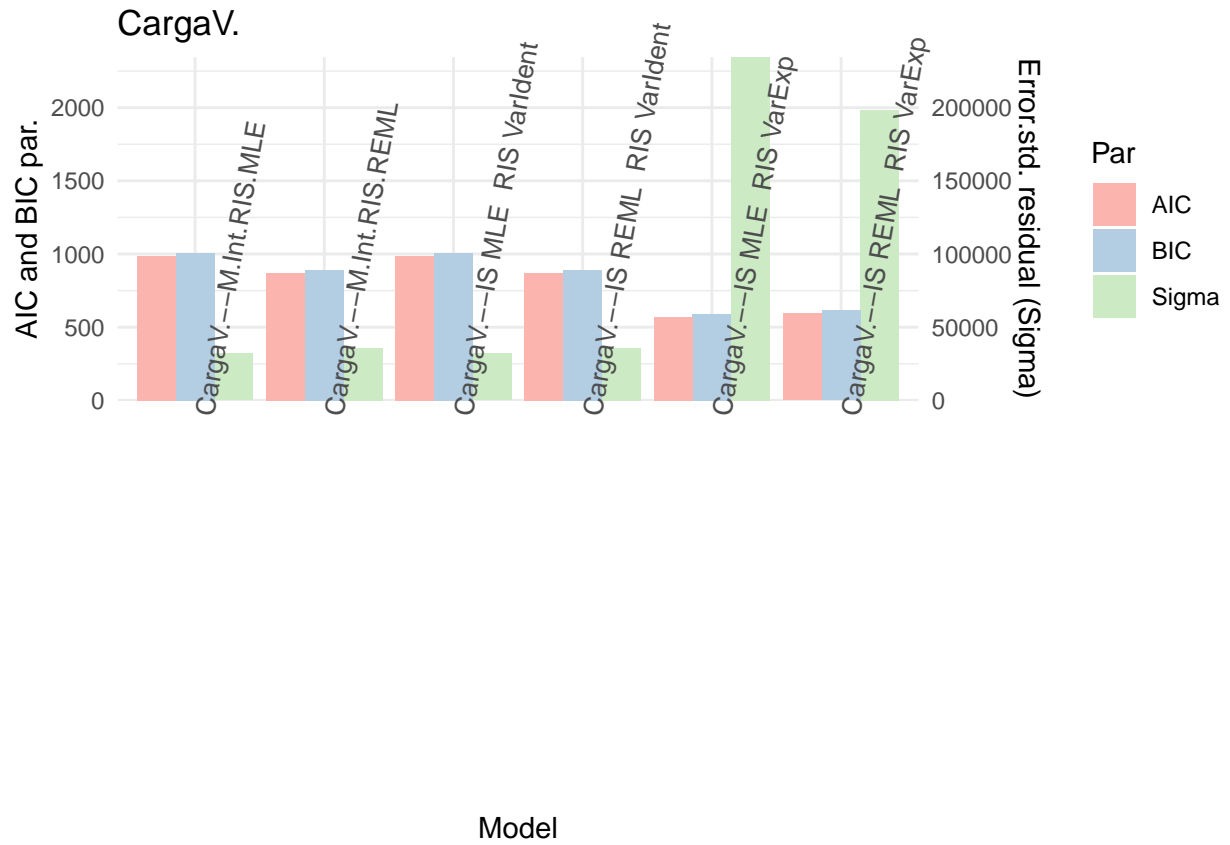
## Warning in logLik.reStruct(object, conLin): Singular precision matrix in
## level -1, block 1

## Warning in logLik.reStruct(object, conLin): Singular precision matrix in
## level -1, block 1
```

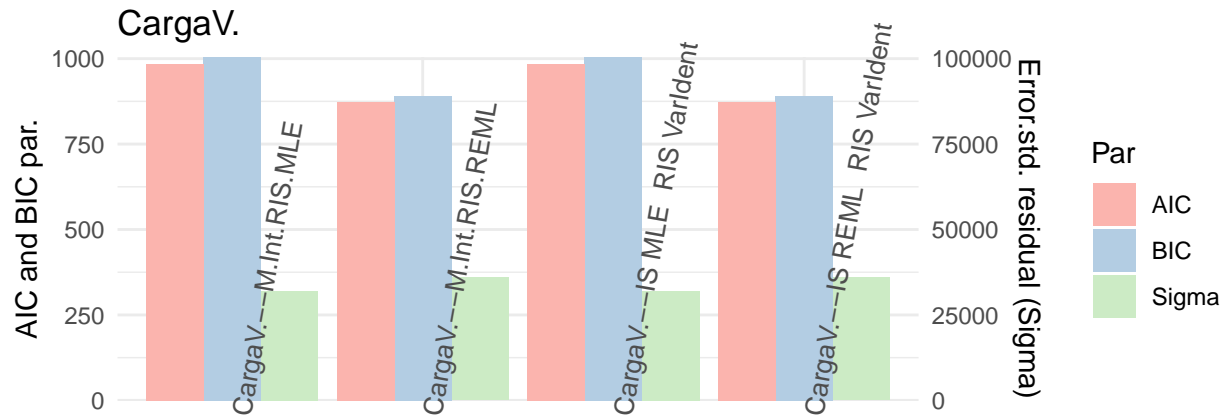
## Diagnòstic models

```
intMO_RIS_Ws0 <- list(Models.sel[c(1:2)],intMO_RIS_W[c(1,3,5,7)])
intMO_RIS_Ws <- ConvList(intMO_RIS_Ws0)
(intMO_QI<-ML_Resum(intMO_RIS_Ws,"CargaV.)) #Es guarda la taula per us posterior
```

```
## [[1]]
##           Sigma      AIC      BIC
## CargaV.--M.Int.RIS.MLE    32069.02  984.1226 1004.3891
## CargaV.--M.Int.RIS.REML    35948.69  872.5253  890.1141
## CargaV.--IS MLE  RIS VarIdent  32069.02  984.1226 1004.3891
## CargaV.--IS REML  RIS VarIdent  35948.69  872.5253  890.1141
## CargaV.--IS MLE  RIS VarExp    234497.34  567.8461  589.8015
## CargaV.--IS REML  RIS VarExp    198543.08  594.2267  613.2813
##
## [[2]]
```



```
ML_Resum(intM0_RIS_Ws[1:4], "CargaV.") [[2]]
```

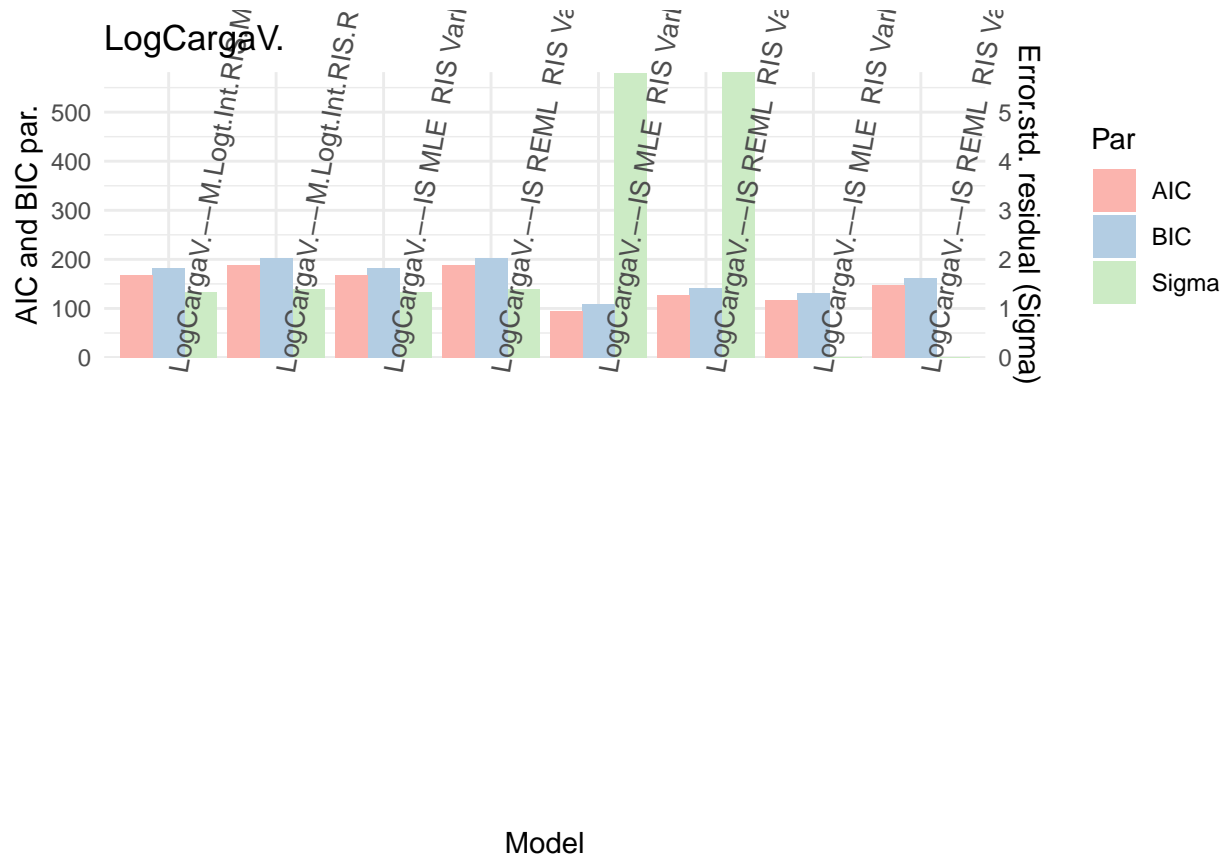


### Model

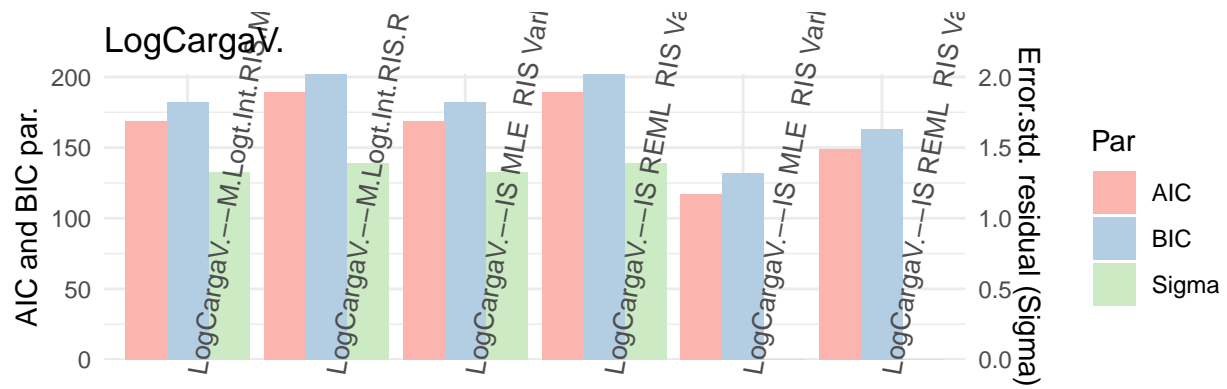
```
intM1_RIS_Ws0 <- list(Models.sel[c(3:4)],intM1_RIS_W[c(1,3,5,7,9,11)])
intM1_RIS_Ws <- ConvList(intM1_RIS_Ws0)
(intM1_QI <- ML_Resum(intM1_RIS_Ws,"LogCargaV."))
```

```
## [[1]]
##
##           Sigma      AIC      BIC
## LogCargaV.--M.Logt.Int.RIS.MLE  1.3307 168.7252 182.2362
## LogCargaV.--M.Logt.Int.RIS.REML  1.3916 189.3124 201.9806
## LogCargaV.--IS MLE RIS VarIdent  1.3307 168.7252 182.2362
## LogCargaV.--IS REML RIS VarIdent  1.3916 189.3124 201.9806
## LogCargaV.--IS MLE RIS VarExp    5.8062  93.8243 109.0242
## LogCargaV.--IS REML RIS VarExp    5.8149 127.6175 141.8692
## LogCargaV.--IS MLE RIS VarPower  0.0000 116.8606 132.0605
## LogCargaV.--IS REML RIS VarPower  0.0000 148.6902 162.9418
##
## [[2]]
```





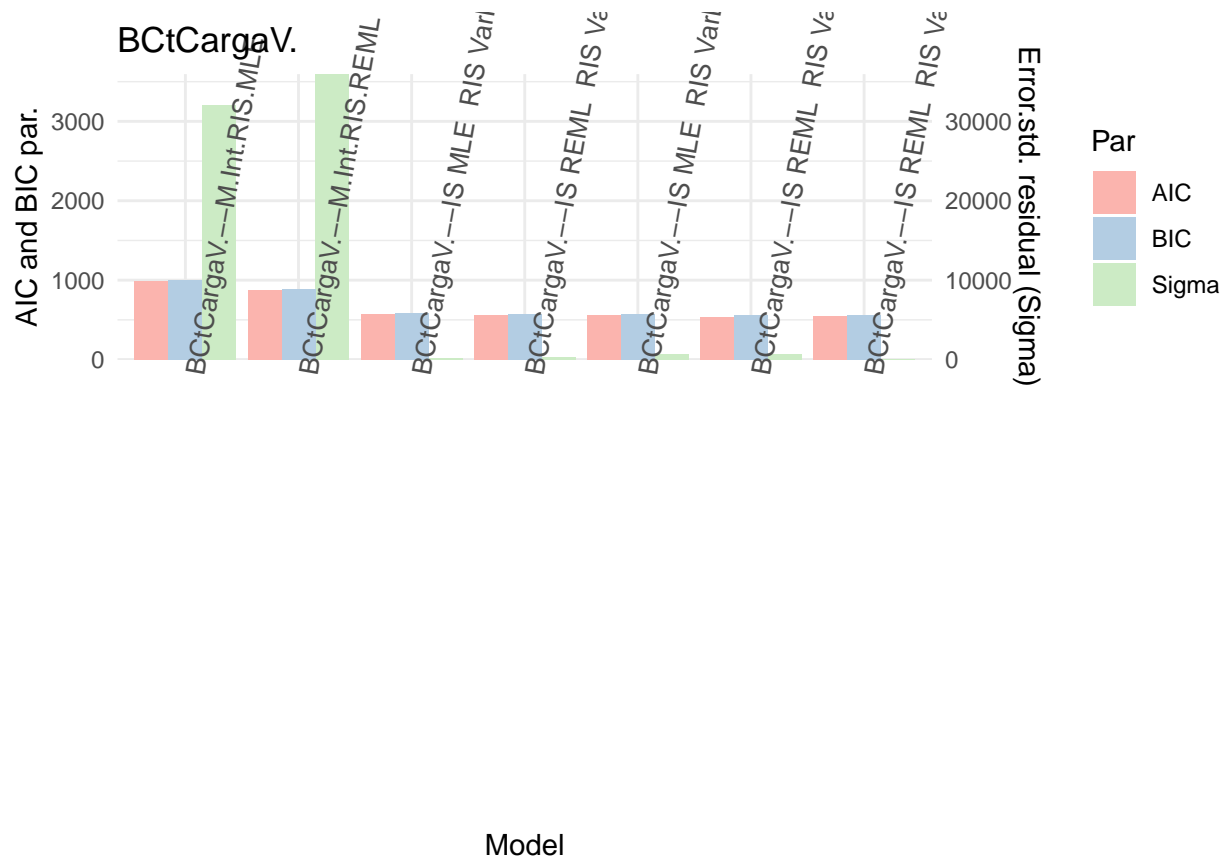
```
ML_Resum(intM1_RIS_Ws[-c(5:6)], "LogCargaV. ")[[2]]
```



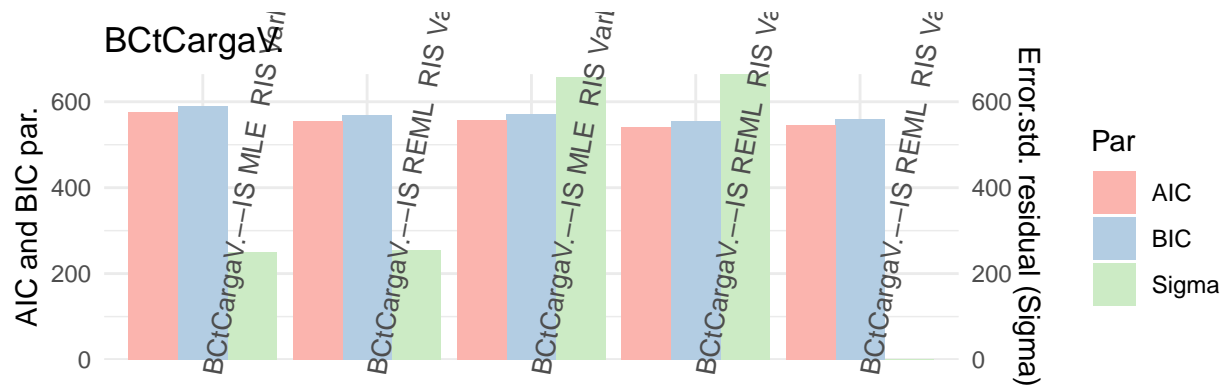
### Model

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

```
## [[1]]
##
##          Sigma      AIC      BIC
## BtCargaV.--M.Int.RIS.MLE  32069.0244  984.1226 1004.3891
## BtCargaV.--M.Int.RIS.REML  35948.6941  872.5253  890.1141
## BtCargaV.--IS MLE RIS VarIdent  249.5092  575.5950  589.1060
## BtCargaV.--IS REML RIS VarIdent  255.5007  556.0538  568.7220
## BtCargaV.--IS MLE RIS VarExp  657.1780  557.3659  572.5659
## BtCargaV.--IS REML RIS VarExp  664.2374  541.3619  555.6136
## BtCargaV.--IS REML RIS VarPower  0.0000  546.2706  560.5222
##
## [[2]]
```



```
ML_Resum(intM2_RIS_Ws[-c(1:2)], "BCtCargaV. ") [[2]]
```



## Model

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

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

```
MO_RIS_W_list_R20 <- list(Models.sel[c(1:2)],intMO_RIS_W[c(2,4,6,8)])
MO_RIS_W_list_R2 <- ConvList(MO_RIS_W_list_R20)

MO_RIS_W_dfA_R2 <- data.frame(Model = paste("M.Int",str_remove(str_remove(str_remove(names(MO_RIS_W_list_R20),
MO_RIS_W_dfA_R2$R2 <- sapply(1:length(MO_RIS_W_list_R2),function(M){
  if (class(MO_RIS_W_list_R2[[M]])[1]=="lmm"){
    summary(MO_RIS_W_list_R2[[M]])$r.squared
  } else if (class(MO_RIS_W_list_R2[[M]])=="lme"){
    if(all(paste(attr(MO_RIS_W_list_R2[[M]]$terms,"predvars")[-c(1:2)]) ==
      paste(attr(MO_RIS_W_list_R2[[M]]$terms,"term.labels")))){
      rsquared(MO_RIS_W_list_R2[[M]]$Conditional)
    } else {
      summary(lm(formula(paste(paste("MO_RIS_W_list_R2[[" ,M, "]"$data$,
        attr(MO_RIS_W_list_R2[[M]]$terms,"variables")[[2]],sep=""),
        "~", "MO_RIS_W_list_R2[[" ,M, "]"$fitted"),sep="")))$r.squared
    }
  }
})
```

```
## Warning in paste(attr(MO_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length

## Warning in paste(attr(MO_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length

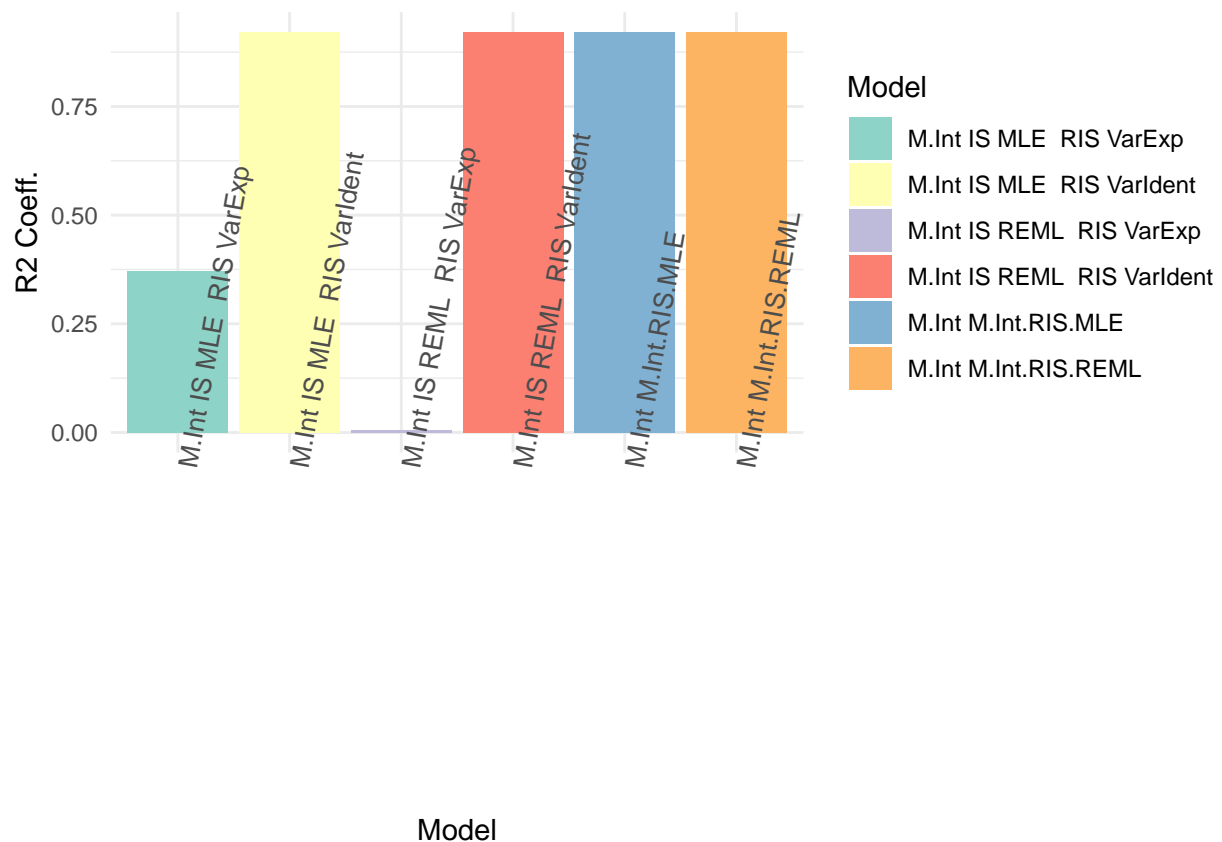
## Warning in paste(attr(MO_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length

## Warning in paste(attr(MO_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length

## Warning in paste(attr(MO_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length

## Warning in paste(attr(MO_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length
```

```
R2_MOW <- ggplot(data=MO_RIS_W_dfa_R2, aes(x=Model, y=R2, fill=Model)) +
  geom_bar(stat="identity", position=position_dodge()) +
  scale_fill_brewer(palette="Set3") + theme_minimal() +
  labs(y = "R2 Coeff.") +
  theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))
R2_MOW
```



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

```

M1_RIS_W_dfA_R2 <- data.frame(Model = paste("Log.M.Int",str_remove(str_remove(str_remove(names(M1_RIS_W_

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

```

```

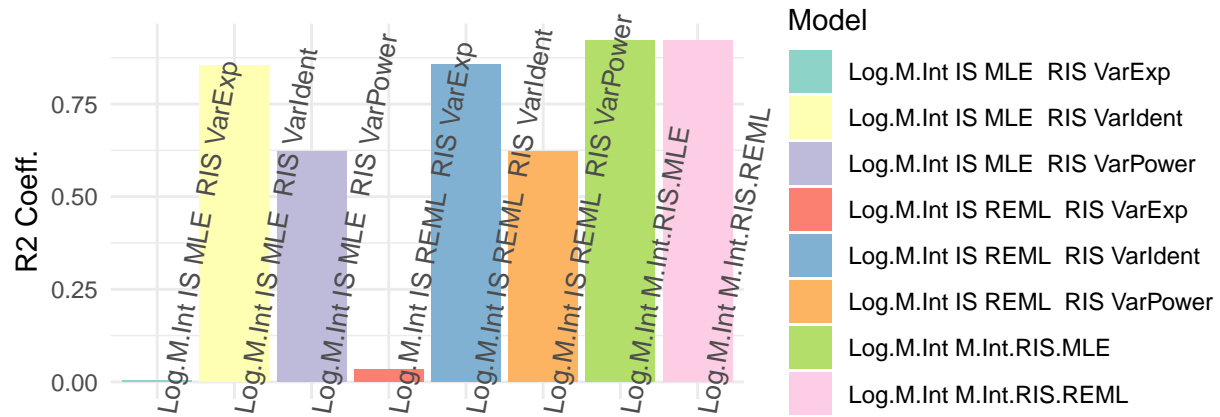
## Warning in paste(attr(M1_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length
## Warning in paste(attr(M1_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length
## Warning in paste(attr(M1_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length
## Warning in paste(attr(M1_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length
## Warning in paste(attr(M1_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length
## Warning in paste(attr(M1_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length
## Warning in paste(attr(M1_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length
## Warning in paste(attr(M1_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length

```

```

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

```



### Model

```

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

M2_RIS_W_dfA_R2 <- data.frame(Model = paste("BCT.M.Int",str_remove(str_remove(str_remove(names(M2_RIS_W_
M2_RIS_W_dfA_R2$R2 <- sapply(1:length(M2_RIS_W_list_R2),function(M){
  if (class(M2_RIS_W_list_R2[[M]])[1]=="lmm"){
    summary(M2_RIS_W_list_R2[[M]])$r.squared
  } else if (class(M2_RIS_W_list_R2[[M]])=="lme"){
    if(all(paste(attr(M2_RIS_W_list_R2[[M]]$terms,"predvars")[-c(1:2)]) ==
      paste(attr(M2_RIS_W_list_R2[[M]]$terms,"term.labels")))){
      rsquared(M2_RIS_W_list_R2[[M]])$Conditional
    } else {
      summary(lm(formula(paste(paste("M2_RIS_W_list_R2[["M,"]]"$data$",
        attr(M2_RIS_W_list_R2[[M]]$terms,"variables")[[2]],sep=""),
          "~", "M2_RIS_W_list_R2[["M,"]]"$fitted"),sep="")))$r.squared
    }
  }
})

## Warning in paste(attr(M2_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length

## Warning in paste(attr(M2_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length

## Warning in paste(attr(M2_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])

```

```
## == : longer object length is not a multiple of shorter object length

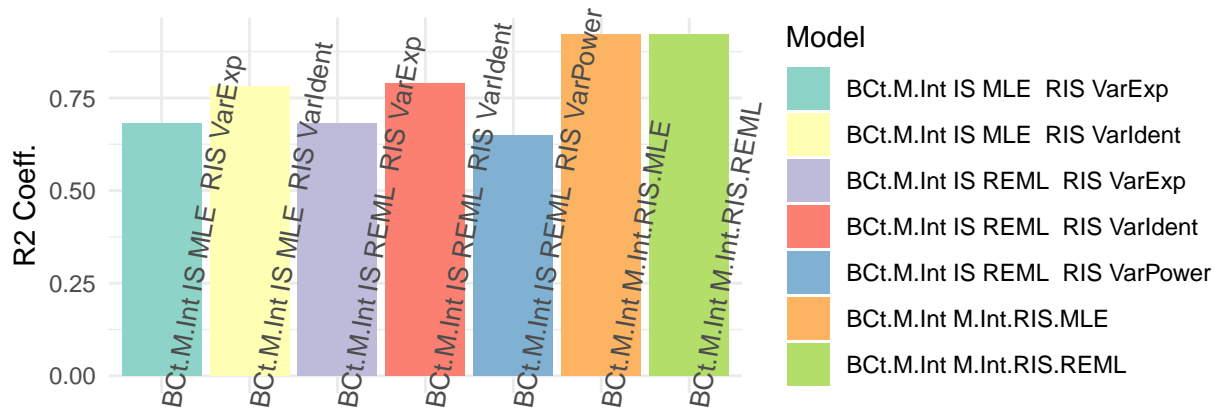
## Warning in paste(attr(M2_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length

## Warning in paste(attr(M2_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length

## Warning in paste(attr(M2_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length

## Warning in paste(attr(M2_RIS_W_list_R2[[M]]$terms, "predvars")[-c(1:2)])
## == : longer object length is not a multiple of shorter object length
```

```
R2_M2W <- ggplot(data=M2_RIS_W_dfa_R2, aes(x=Model, y=R2, fill=Model)) +
  geom_bar(stat="identity", position=position_dodge()) +
  scale_fill_brewer(palette="Set3") + theme_minimal() +
  labs(y = "R2 Coeff.") +
  theme(axis.text.x = element_text(angle=80,size = 10,hjust=0))
R2_M2W
```



### Model

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

```
M0_RIS_W_list_base <- M0_RIS_W_list_R2
```

```
names(M0_RIS_W_list_base) <- str_remove(str_remove(str_remove(names(M0_RIS_W_list_base), "Model"), "Base
#M0_RIS_W_list_R2
```

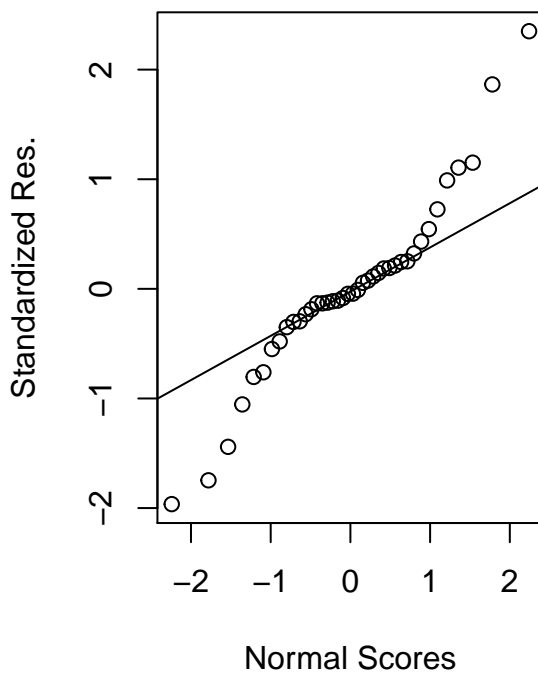
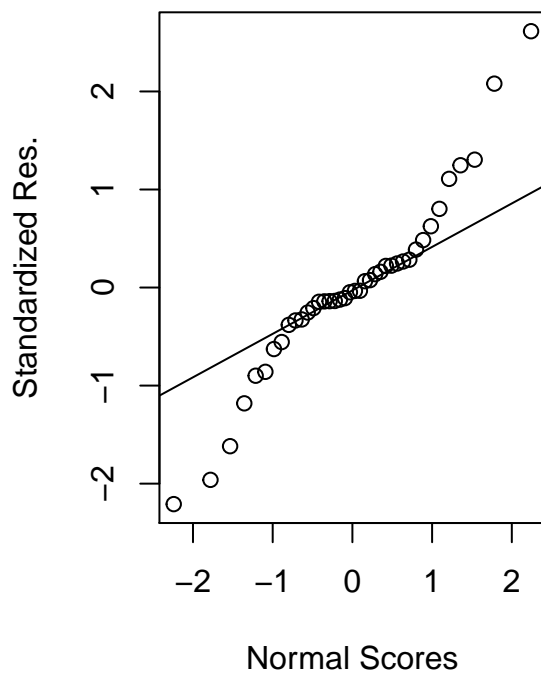


```

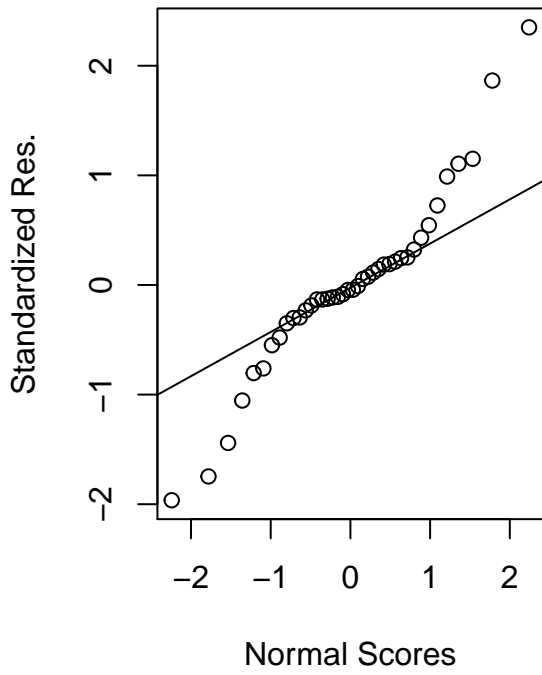
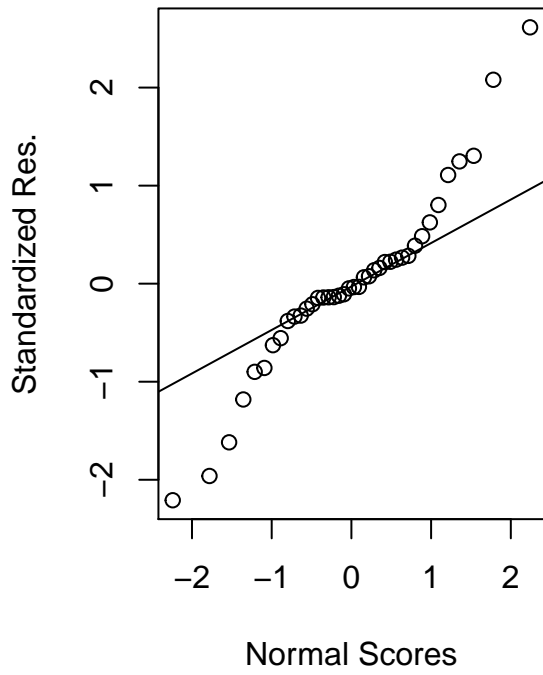
par(mfrow=c(1,2))
QQ_MO_list <- lapply(1:length(MO_RIS_W_list_base),function(QQ){
  IntrIS_MLE_W.SMO.sdres <- residuals(MO_RIS_W_list_base[[QQ]],type = "pearson")
  qqnorm(IntrIS_MLE_W.SMO.sdres,
    ylab="Standardized Res.",
    xlab="Normal Scores",
    main=paste("MO",names(MO_RIS_W_list_base[QQ]),"Standard. Res. "))
  qqline(IntrIS_MLE_W.SMO.sdres)
})

```

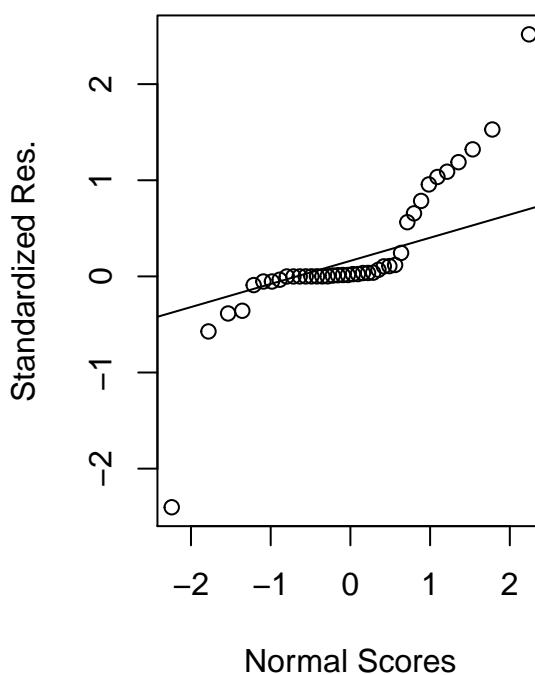
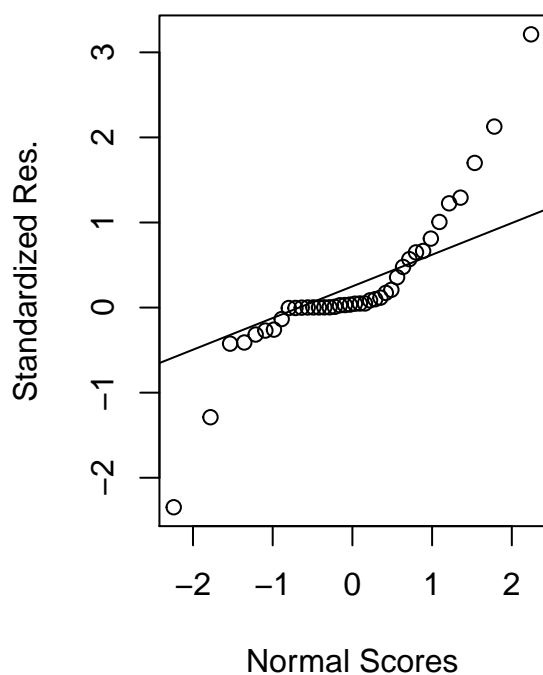
**M0 M.Int.RIS.MLE Standard. Res    M0 M.Int.RIS.REML Standard. Re**



**M0 IS MLE RIS VarIdent Standard. I0 IS REML RIS VarIdent Standard.**



## M0 IS MLE RIS VarExp Standard. M0 IS REML RIS VarExp Standard.



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

```
Shap_M0_df
```

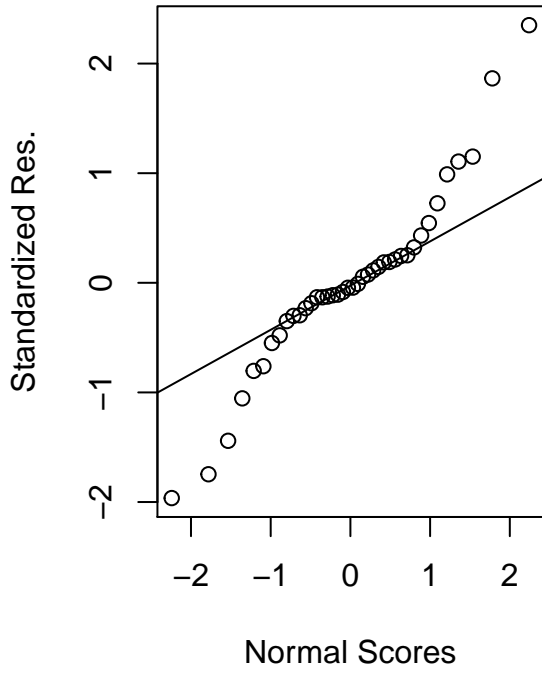
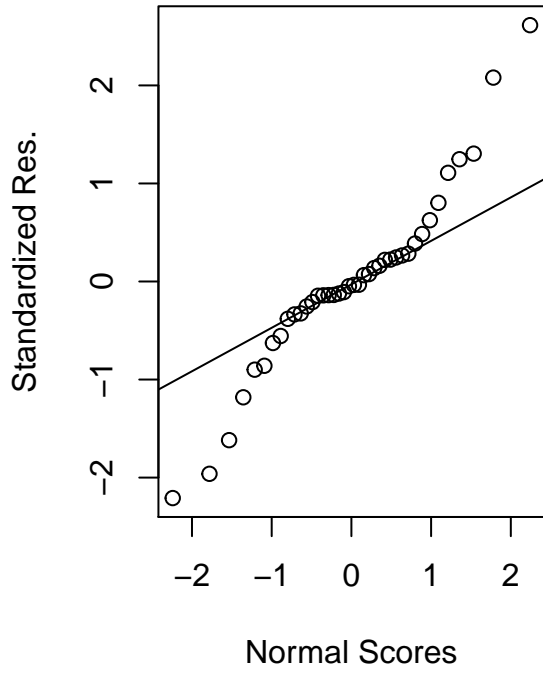
```
##           Model Shapiro_p.valor
## 1      M2 M.Int.RIS.MLE  9.177733e-04
## 2      M2 M.Int.RIS.REML  9.395222e-04
## 3 M2 IS MLE RIS VarIdent  9.177733e-04
## 4 M2 IS REML RIS VarIdent  9.395222e-04
## 5      M2 IS MLE RIS VarExp  6.938304e-16
## 6      M2 IS REML RIS VarExp  6.777748e-16
```

```
M1_RIS_W_list_base <- M1_RIS_W_list_R2
```

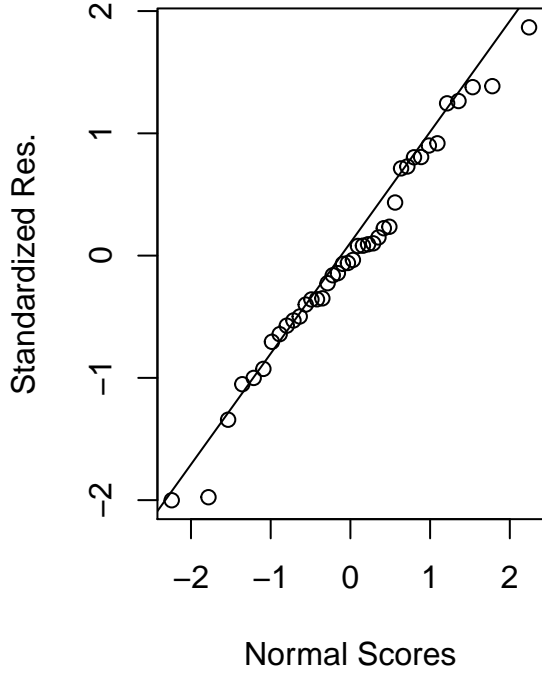
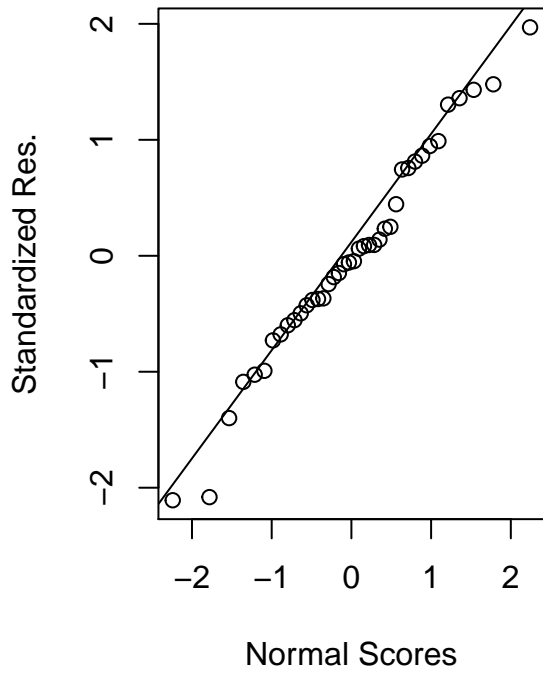
```
names(M1_RIS_W_list_base) <- str_remove(str_remove(str_remove(names(M1_RIS_W_list_base), "Model"), "Base"),
#M1_RIS_W_list_R2
par(mfrow=c(1,2))
QQ_M1_list <- lapply(1:length(M1_RIS_W_list_base), function(QQ){
  IntrIS_MLE_W.SM1.sdres <- residuals(M1_RIS_W_list_base[[QQ]], type = "pearson")
  qqnorm(IntrIS_MLE_W.SM1.sdres,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

})

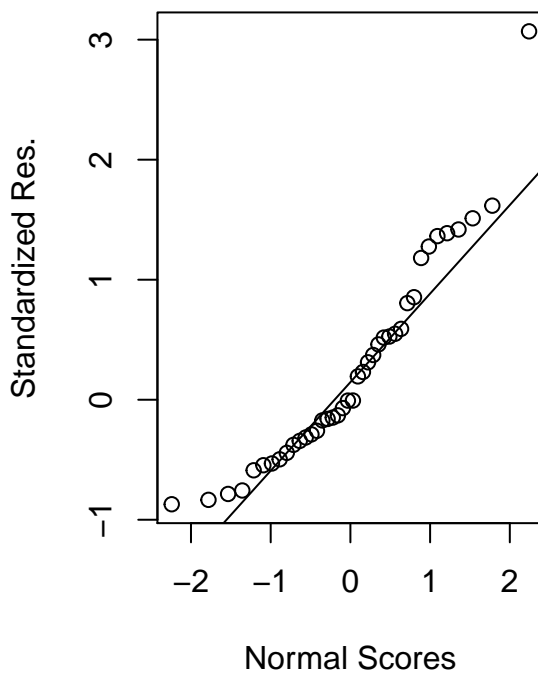
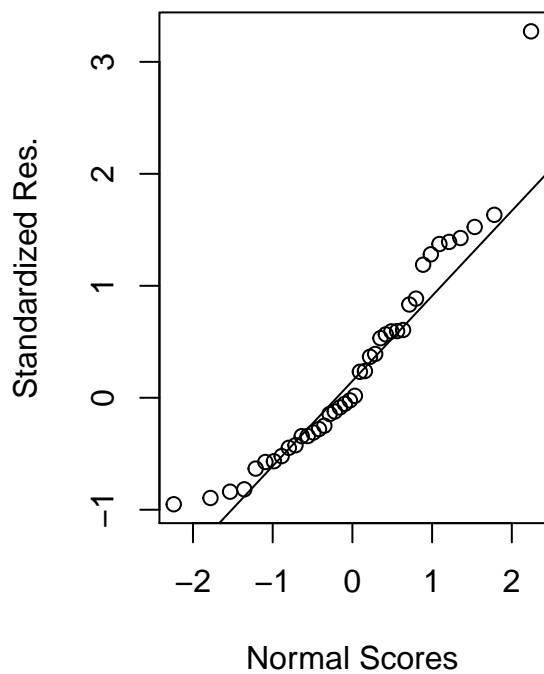
**M1 M.Int.RIS.MLE Standard. Res**    **M1 M.Int.RIS.REML Standard. Re**



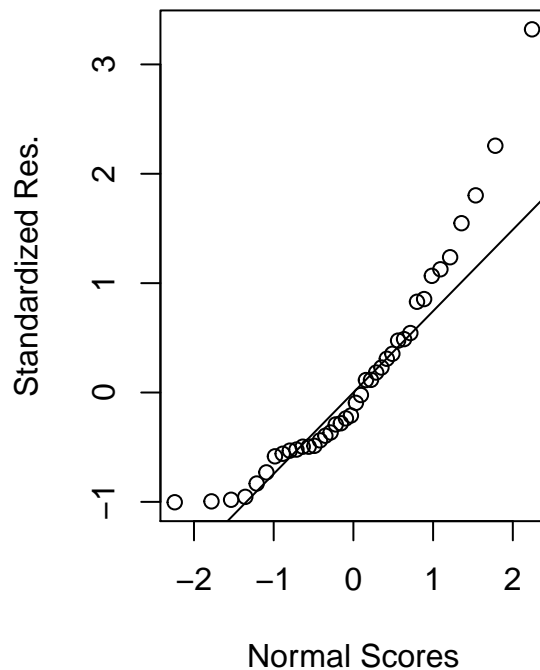
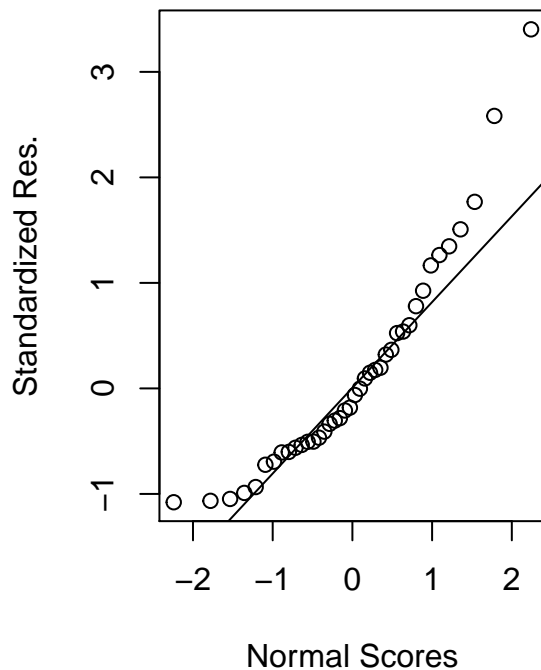
**M1 IS MLE RIS VarIdent Standard. M1 IS REML RIS VarIdent Standard.**



**M1 IS MLE RIS VarExp Standard. M1 IS REML RIS VarExp Standard.**



## M1 IS MLE RIS VarPower Standard.1 IS REML RIS VarPower Standard



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

```
Shap_M1_df
```

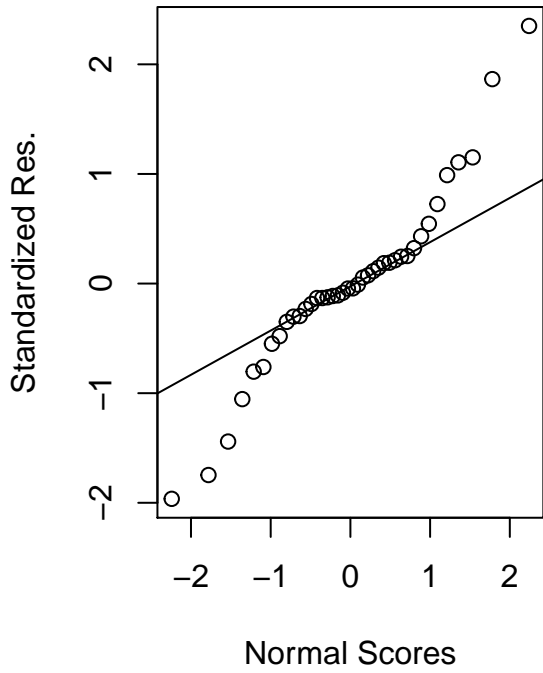
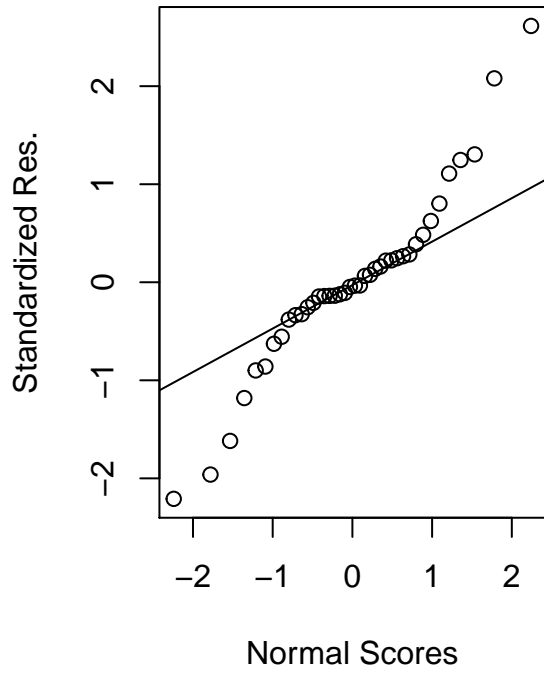
```
##           Model Shapiro_p.valor
## 1      M2 M.Int.RIS.MLE  9.177733e-04
## 2      M2 M.Int.RIS.REML  9.395222e-04
## 3 M2 IS MLE RIS VarIdent  3.531845e-03
## 4 M2 IS REML RIS VarIdent  3.487835e-03
## 5      M2 IS MLE RIS VarExp  2.781421e-13
## 6      M2 IS REML RIS VarExp  2.725970e-13
## 7      M2 IS MLE RIS VarPower  3.551689e-12
## 8      M2 IS REML RIS VarPower  4.694416e-12
```

```
M2_RIS_W_list_base <- M2_RIS_W_list_R2
```

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

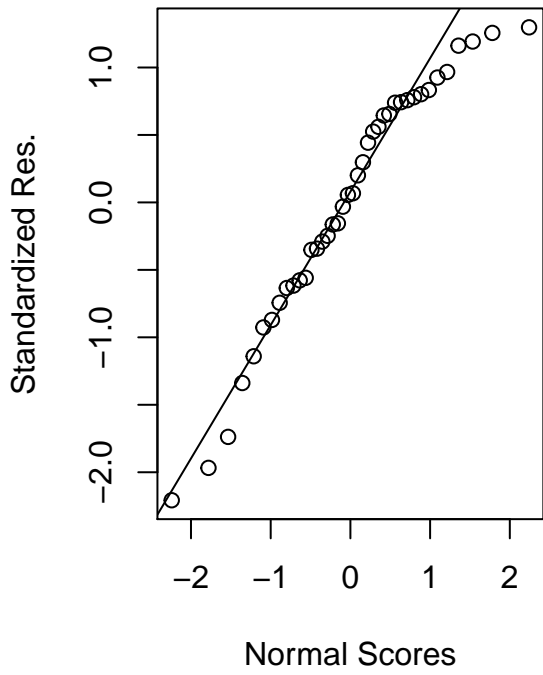
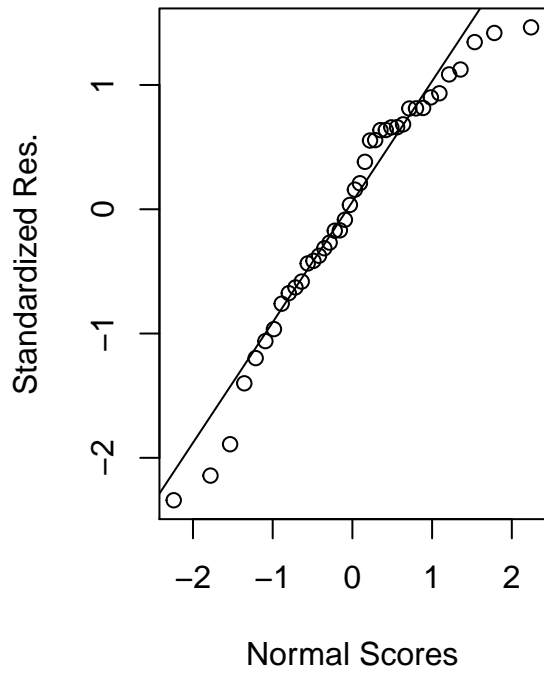
```
main=paste("M2",names(M2_RIS_W_list_base[QQ]),"Standard. Res.")
qqline(IntRIS_MLE_W.SM2.sdres)
})
```

**M2 M.Int.RIS.MLE Standard. Res**    **M2 M.Int.RIS.REML Standard. Re**

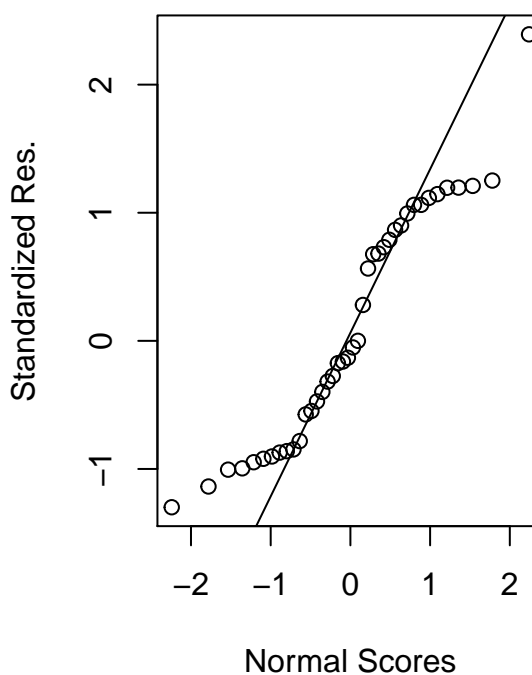
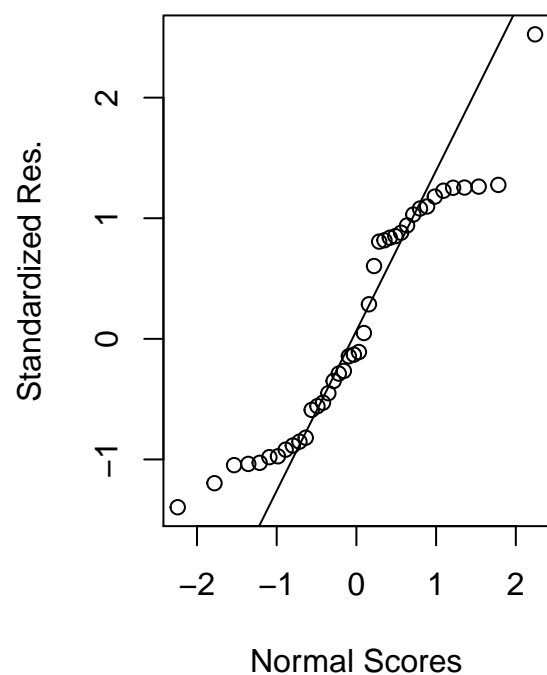




**M2 IS MLE RIS VarIdent Standard. I2 IS REML RIS VarIdent Standard.**



## M2 IS MLE RIS VarExp Standard. M2 IS REML RIS VarExp Standard.

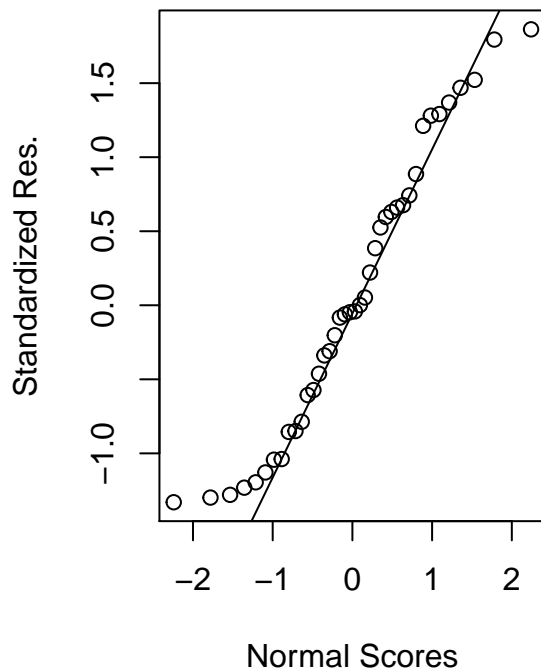


```
Shap_M2_df <- data.frame(Model = paste("M2", names(M2_RIS_W_list_base)))
Shap_M2_df$Shapiro_p.valor <- Shap_M2_v <- sapply(1:length(M2_RIS_W_list_base), function(Sh){
  shapiro.test(M2_RIS_W_list_base[[Sh]]$residuals)$p
})
```

```
Shap_M2_df
```

##	Model	Shapiro_p.valor
## 1	M2 M.Int.RIS.MLE	9.177733e-04
## 2	M2 M.Int.RIS.REML	9.395222e-04
## 3	M2 IS MLE RIS VarIdent	4.949362e-02
## 4	M2 IS REML RIS VarIdent	1.077835e-01
## 5	M2 IS MLE RIS VarExp	1.087542e-05
## 6	M2 IS REML RIS VarExp	1.699225e-05
## 7	M2 IS REML RIS VarPower	8.798470e-03

## 2 IS REML RIS VarPower Standard



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

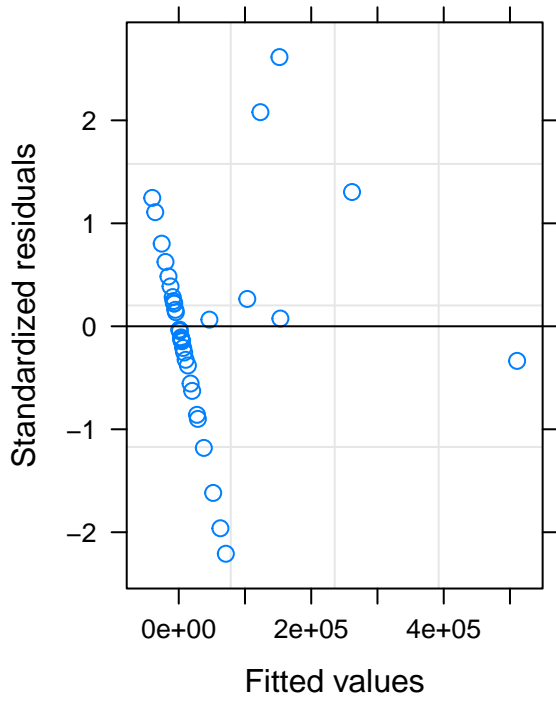
```
#Variation of "Levene's Test"
```

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

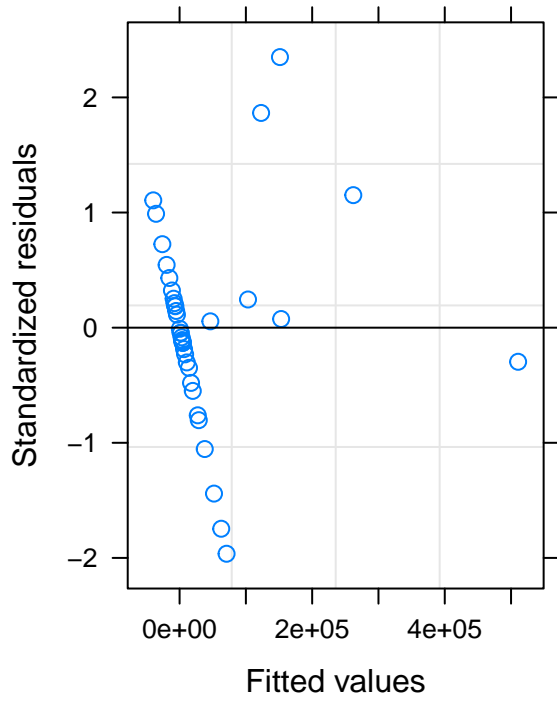
```
RP_MO_list <- lapply(1:length(MO_RIS_W_list_base),function(RP){  
  plot(MO_RIS_W_list_base[[RP]],which=1,main=paste("M0",names(MO_RIS_W_list_base)[RP]))  
})
```

```
marrangeGrob(RP_MO_list, ncol=2, nrow=1)
```

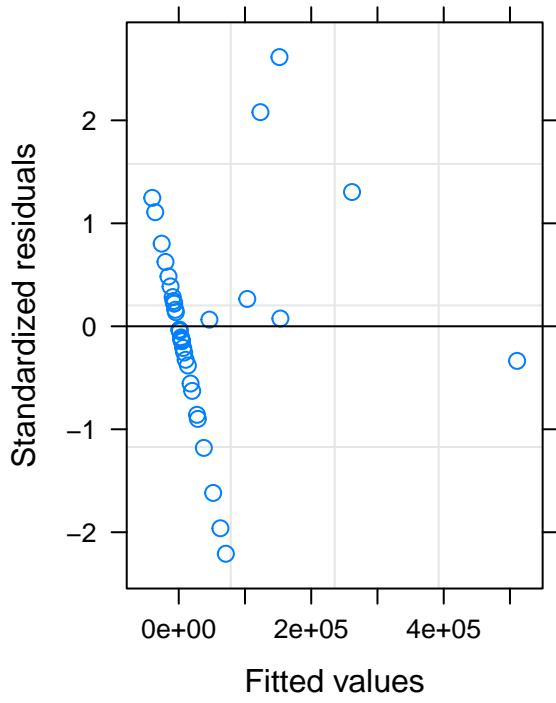
**M0 M.Int.RIS.MLE**



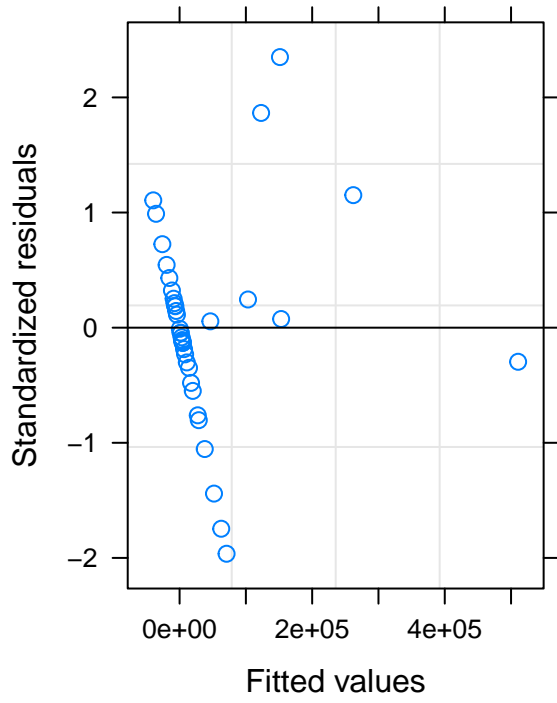
**M0 M.Int.RIS.REML**



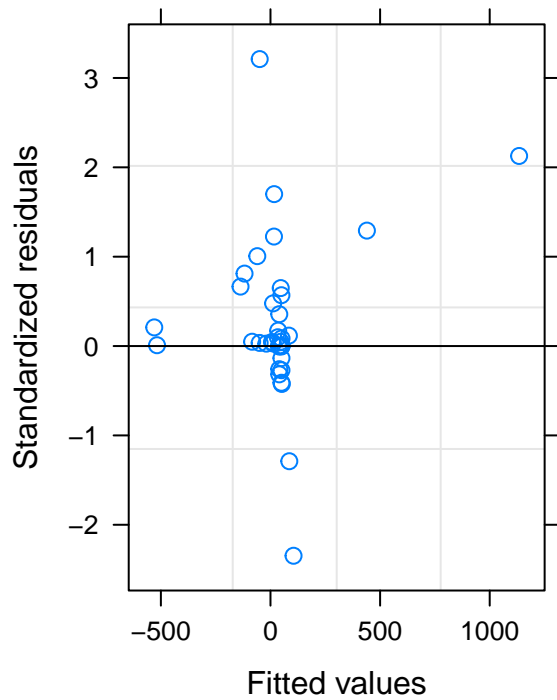
**M0 IS MLE RIS VarIdent**



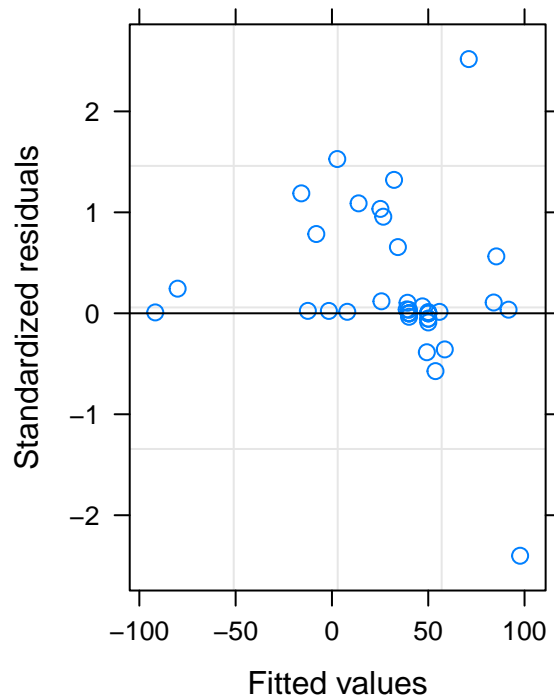
**M0 IS REML RIS VarIdent**



**M0 IS MLE RIS VarExp**



**M0 IS REML RIS VarExp**



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

```
names(LT_MO_list)<-names(MO_RIS_W_list_base)
LT_MO_list
```

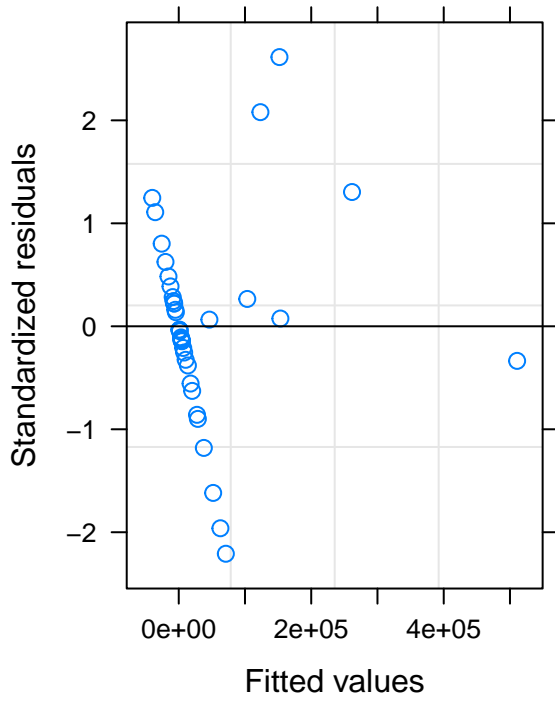
```
## $M.Int.RIS.MLE
## Analysis of Variance Table
##
## Response: Model.F.Res2
##           Df      Sum Sq   Mean Sq F value Pr(>F)
## Ident      7 3.4565e+19 4.9378e+18  2.4131 0.04203 *
## Residuals 32 6.5481e+19 2.0463e+18
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## $M.Int.RIS.REML
## Analysis of Variance Table
##
## Response: Model.F.Res2
##           Df      Sum Sq   Mean Sq F value Pr(>F)
## Ident      7 3.4961e+19 4.9944e+18  2.4079 0.0424 *
## Residuals 32 6.6374e+19 2.0742e+18
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

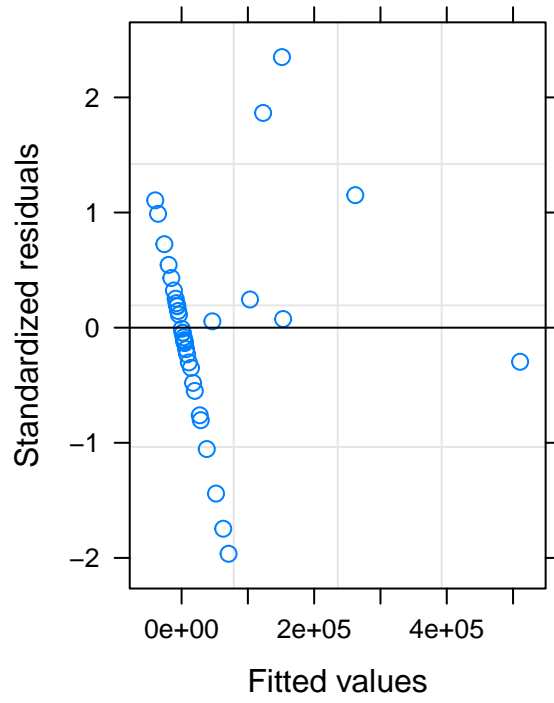
##
## $'IS MLE RIS VarIdent '
## Analysis of Variance Table
##
## Response: Model.F.Res2
##      Df      Sum Sq   Mean Sq F value Pr(>F)
## Ident    7 3.4565e+19 4.9378e+18  2.4131 0.04203 *
## Residuals 32 6.5481e+19 2.0463e+18
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## $'IS REML RIS VarIdent '
## Analysis of Variance Table
##
## Response: Model.F.Res2
##      Df      Sum Sq   Mean Sq F value Pr(>F)
## Ident    7 3.4961e+19 4.9944e+18  2.4079 0.0424 *
## Residuals 32 6.6374e+19 2.0742e+18
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## $'IS MLE RIS VarExp '
## Analysis of Variance Table
##
## Response: Model.F.Res2
##      Df      Sum Sq   Mean Sq F value Pr(>F)
## Ident    7 9.5405e+21 1.3629e+21  0.7219 0.6544
## Residuals 32 6.0417e+22 1.8880e+21
##
## $'IS REML RIS VarExp '
## Analysis of Variance Table
##
## Response: Model.F.Res2
##      Df      Sum Sq   Mean Sq F value Pr(>F)
## Ident    7 9.6273e+21 1.3753e+21  0.723 0.6535
## Residuals 32 6.0873e+22 1.9023e+21
RP_M1_list <- lapply(1:length(M1_RIS_W_list_base),function(RP){
  plot(M1_RIS_W_list_base[[RP]],which=1,main=paste("M1",names(M1_RIS_W_list_base)[RP]))
})
marrangeGrob(RP_M1_list, ncol=2, nrow=1)

```

**M1 M.Int.RIS.MLE**

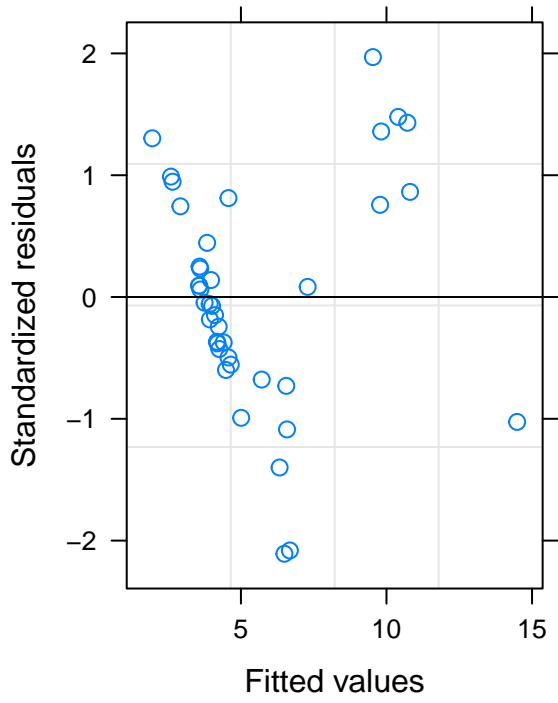


**M1 M.Int.RIS.REML**

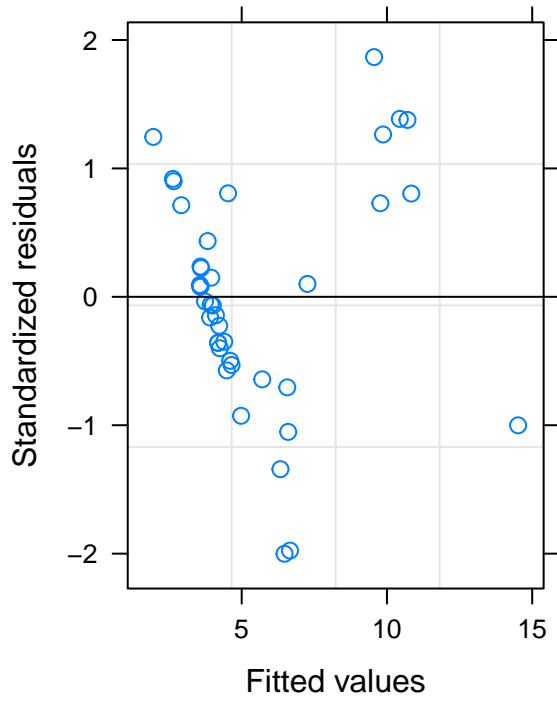




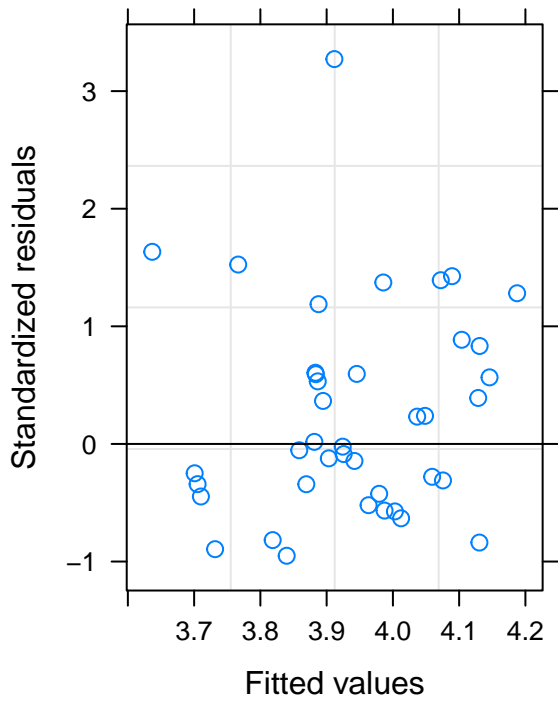
**M1 IS MLE RIS VarIdent**



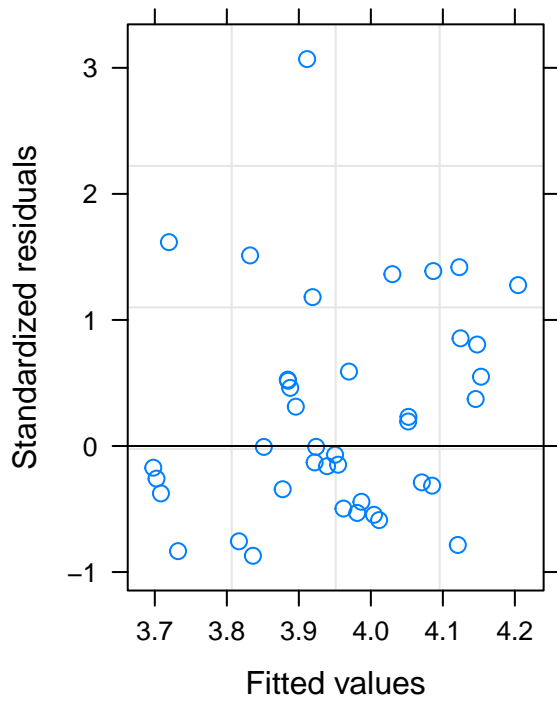
**M1 IS REML RIS VarIdent**

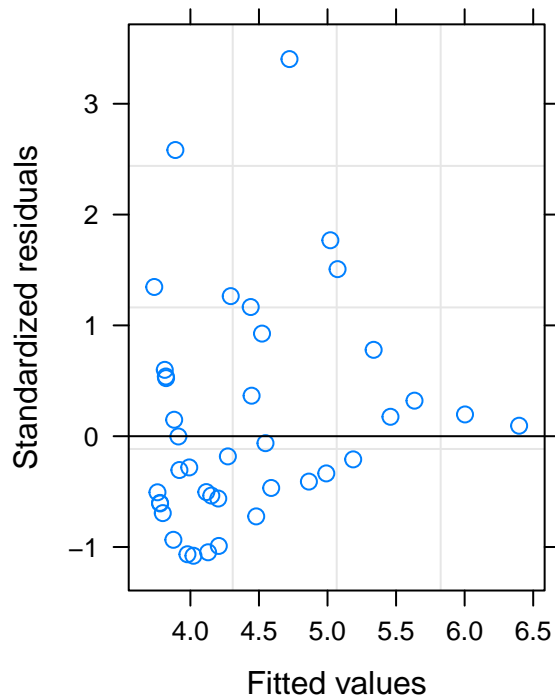
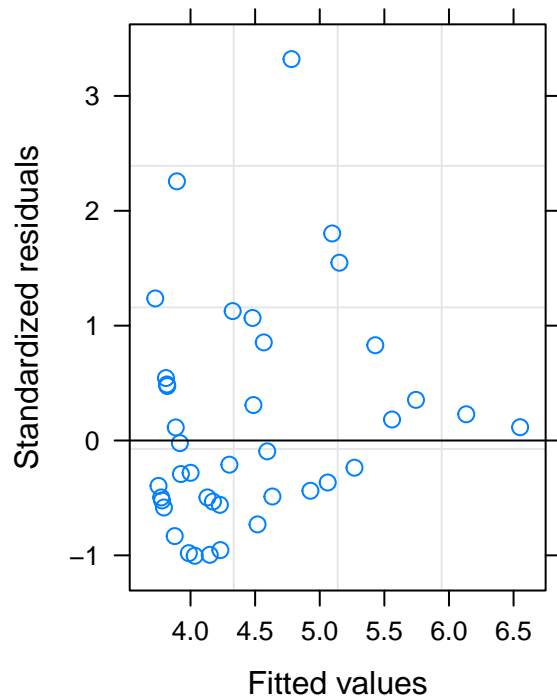


**M1 IS MLE RIS VarExp**



**M1 IS REML RIS VarExp**



**M1 IS MLE RIS VarPower****M1 IS REML RIS VarPower**

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

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

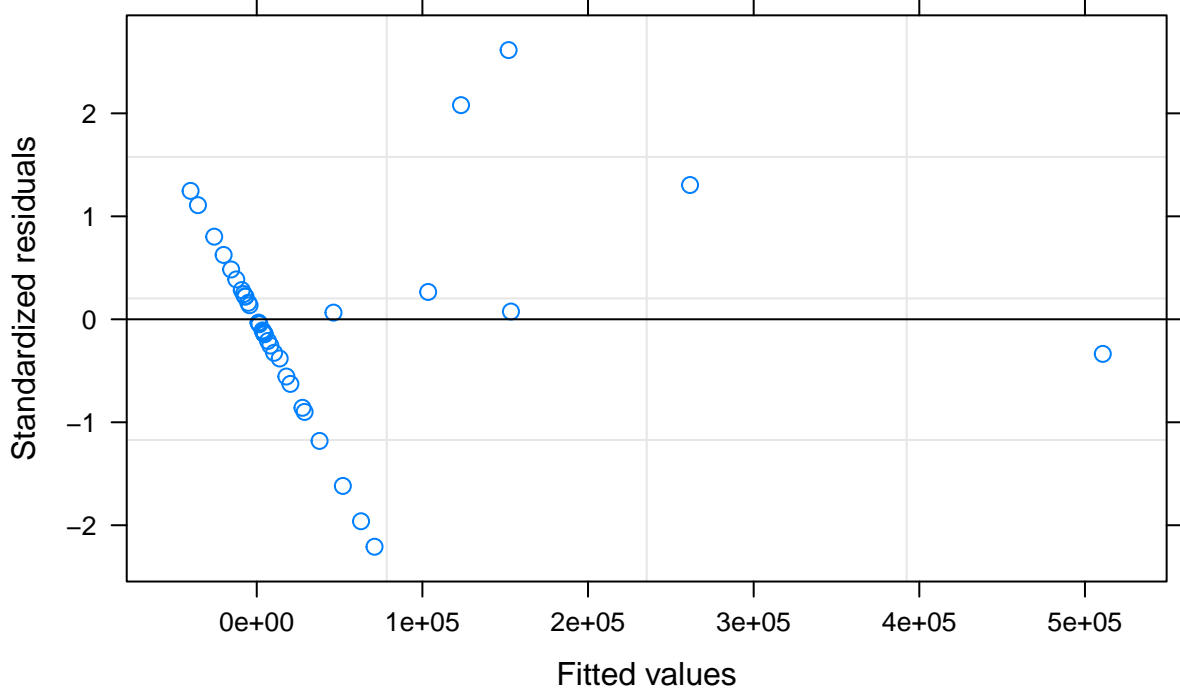
```
## $'Logt M.Int.RIS.MLE'
## Analysis of Variance Table
##
## Response: Model.F.Res2
##           Df      Sum Sq   Mean Sq F value Pr(>F)
## Ident      7 3.4565e+19 4.9378e+18  2.4131 0.04203 *
## Residuals 32 6.5481e+19 2.0463e+18
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## $'Logt M.Int.RIS.REML'
## Analysis of Variance Table
##
## Response: Model.F.Res2
##           Df      Sum Sq   Mean Sq F value Pr(>F)
## Ident      7 3.4961e+19 4.9944e+18  2.4079 0.0424 *
## Residuals 32 6.6374e+19 2.0742e+18
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

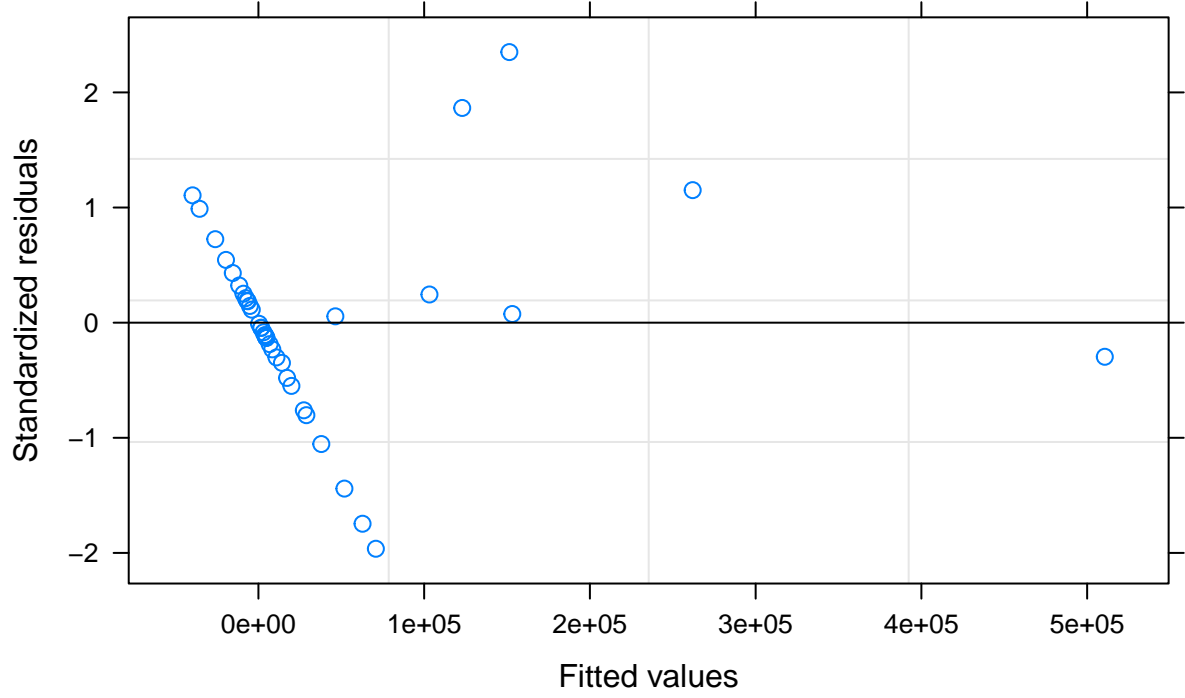
##
## $'Logt IS MLE RIS VarIdent '
## Analysis of Variance Table
##
## Response: Model.F.Res2
##           Df Sum Sq Mean Sq F value Pr(>F)
## Ident      7  34.297  4.8996  1.1832 0.3399
## Residuals 32 132.516  4.1411
##
## $'Logt IS REML RIS VarIdent '
## Analysis of Variance Table
##
## Response: Model.F.Res2
##           Df Sum Sq Mean Sq F value Pr(>F)
## Ident      7  33.593  4.7991  1.1992 0.3315
## Residuals 32 128.061  4.0019
##
## $'Logt IS MLE RIS VarExp '
## Analysis of Variance Table
##
## Response: Model.F.Res2
##           Df Sum Sq Mean Sq F value Pr(>F)
## Ident      7   800.5  114.36  0.1408 0.9941
## Residuals 32 25990.3  812.20
##
## $'Logt IS REML RIS VarExp '
## Analysis of Variance Table
##
## Response: Model.F.Res2
##           Df Sum Sq Mean Sq F value Pr(>F)
## Ident      7   774.4  110.63  0.1393 0.9943
## Residuals 32 25414.5  794.20
##
## $'Logt IS MLE RIS VarPower '
## Analysis of Variance Table
##
## Response: Model.F.Res2
##           Df Sum Sq Mean Sq F value Pr(>F)
## Ident      7   361.5   51.64  0.1461 0.9934
## Residuals 32 11312.4  353.51
##
## $'Logt IS REML RIS VarPower '
## Analysis of Variance Table
##
## Response: Model.F.Res2
##           Df Sum Sq Mean Sq F value Pr(>F)
## Ident      7   344.9   49.28  0.1477 0.9931
## Residuals 32 10674.8  333.59
RP_M2_list <- lapply(1:length(M2_RIS_W_list_base),function(RP){
  plot(M2_RIS_W_list_base[[RP]],which=1,main=paste("M2",names(M2_RIS_W_list_base)[RP]))
})
marrangeGrob(RP_M2_list, ncol=1, nrow=1)

```

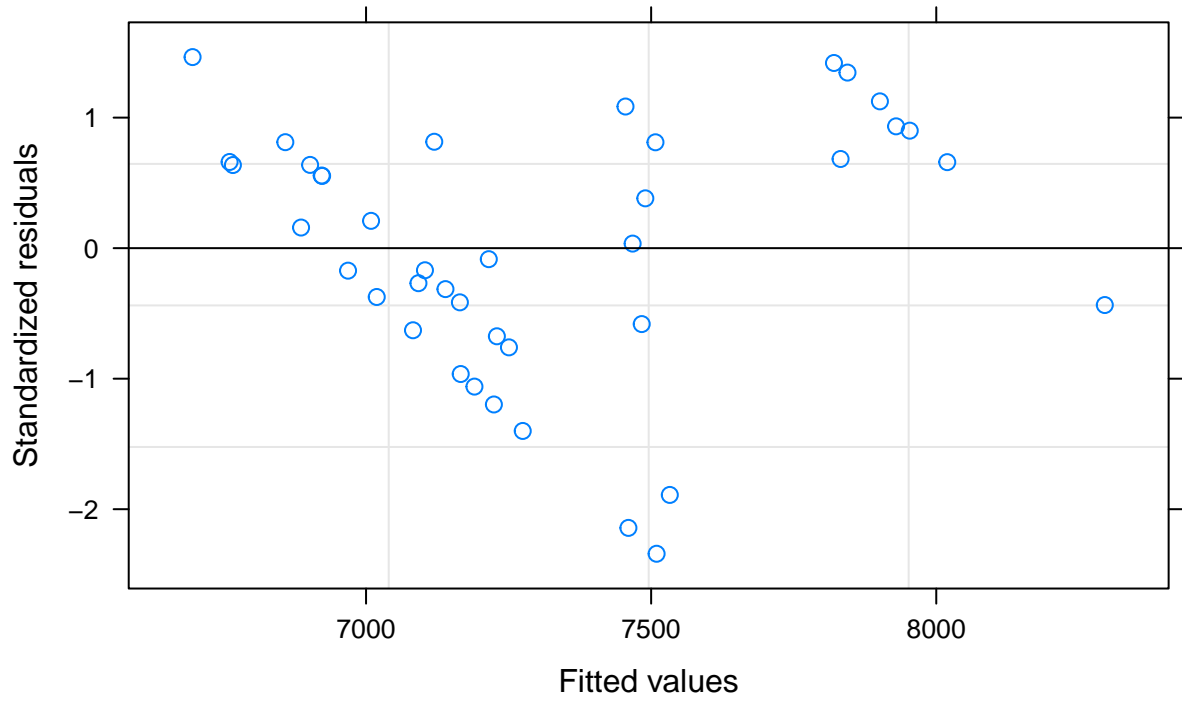
**M2 M.Int.RIS.MLE**



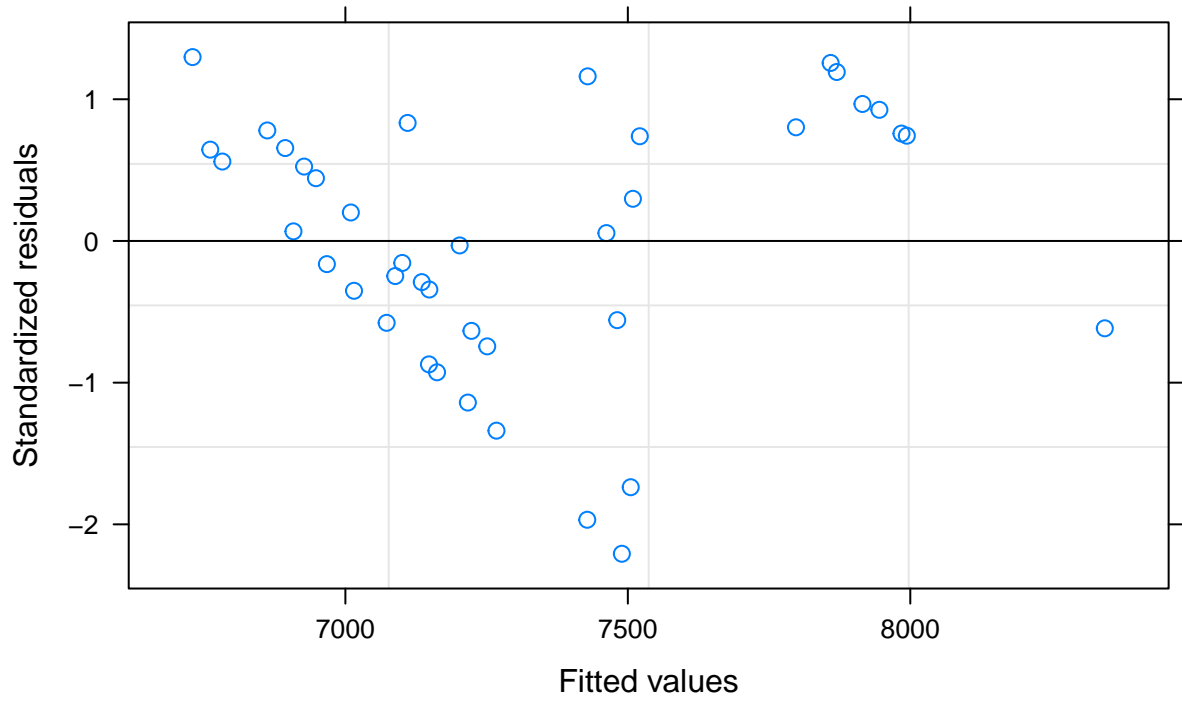
**M2 M.Int.RIS.REML**



### M2 IS MLE RIS VarIdent

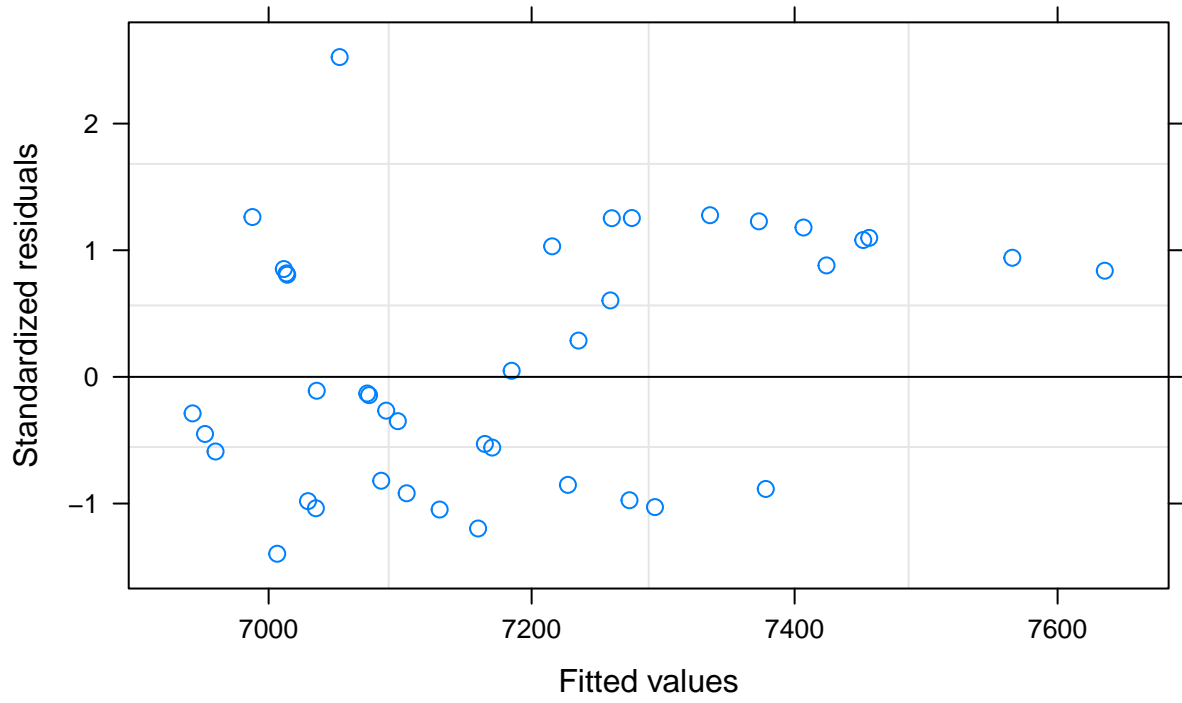


**M2 IS REML RIS VarIdent**

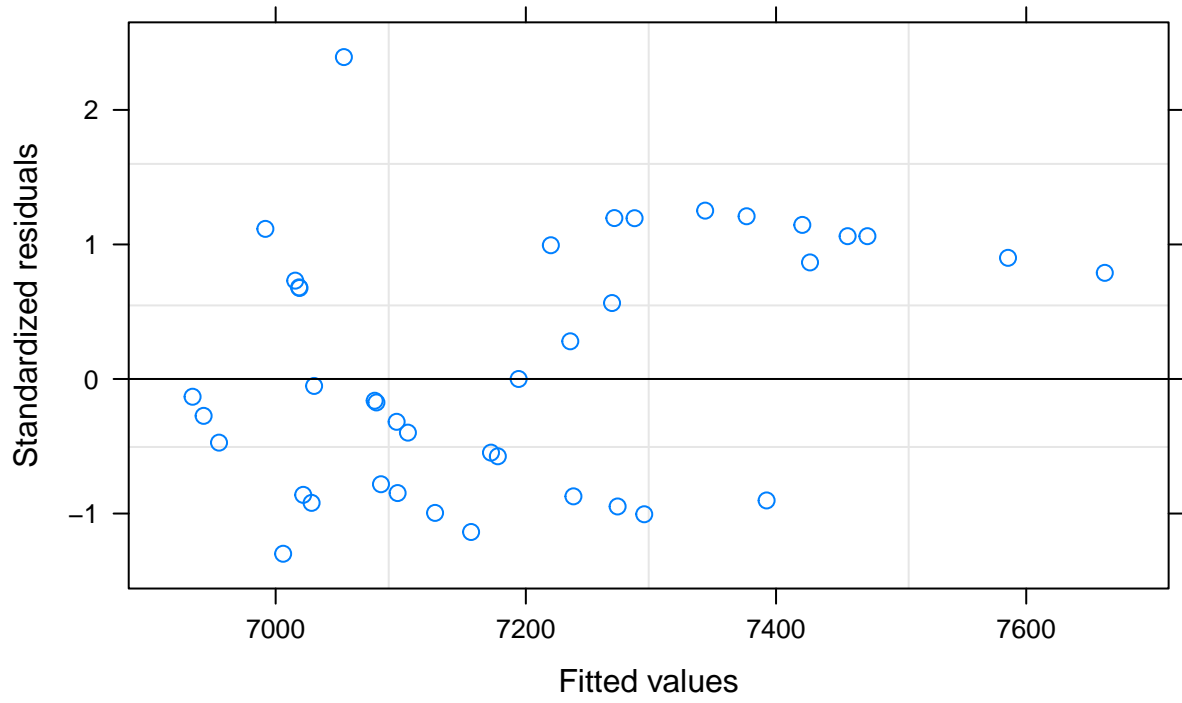




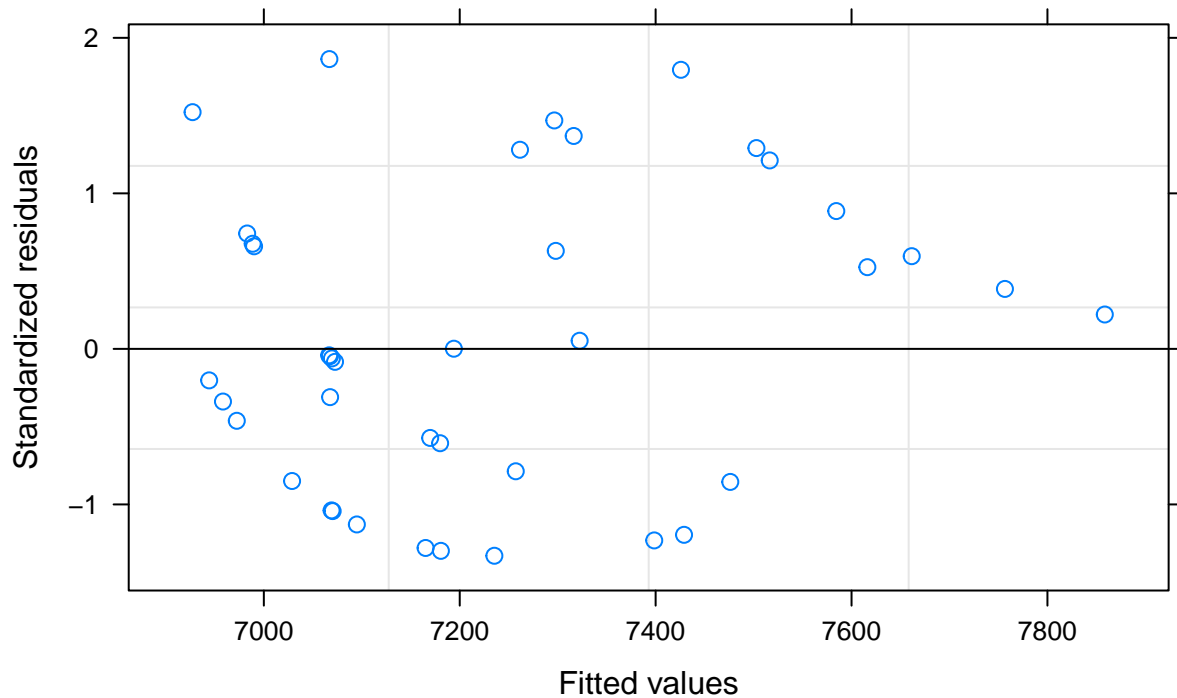
### M2 IS MLE RIS VarExp



### M2 IS REML RIS VarExp



### M2 IS REML RIS VarPower



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

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

```
## $'Bct M.Int.RIS.MLE'
## Analysis of Variance Table
##
## Response: Model.F.Res2
##      Df      Sum Sq   Mean Sq F value Pr(>F)
## Ident    7 3.4565e+19 4.9378e+18  2.4131 0.04203 *
## Residuals 32 6.5481e+19 2.0463e+18
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## $'Bct M.Int.RIS.REML'
## Analysis of Variance Table
##
## Response: Model.F.Res2
##      Df      Sum Sq   Mean Sq F value Pr(>F)
## Ident    7 3.4961e+19 4.9944e+18  2.4079 0.0424 *
## Residuals 32 6.6374e+19 2.0742e+18
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

##
## $'Bct IS MLE RIS VarIdent '
## Analysis of Variance Table
##
## Response: Model.F.Res2
##      Df      Sum Sq    Mean Sq F value Pr(>F)
## Ident    7 3.0061e+10 4294400176  0.7051 0.6677
## Residuals 32 1.9489e+11 6090431011
##
## $'Bct IS REML RIS VarIdent '
## Analysis of Variance Table
##
## Response: Model.F.Res2
##      Df      Sum Sq    Mean Sq F value Pr(>F)
## Ident    7 2.4240e+10 3462840051  0.702 0.6701
## Residuals 32 1.5784e+11 4932635905
##
## $'Bct IS MLE RIS VarExp '
## Analysis of Variance Table
##
## Response: Model.F.Res2
##      Df      Sum Sq    Mean Sq F value Pr(>F)
## Ident    7 3.5176e+10 5.0251e+09  0.1016 0.9979
## Residuals 32 1.5822e+12 4.9443e+10
##
## $'Bct IS REML RIS VarExp '
## Analysis of Variance Table
##
## Response: Model.F.Res2
##      Df      Sum Sq    Mean Sq F value Pr(>F)
## Ident    7 3.5565e+10 5.0807e+09  0.1088 0.9973
## Residuals 32 1.4945e+12 4.6704e+10
##
## $'Bct IS REML RIS VarPower '
## Analysis of Variance Table
##
## Response: Model.F.Res2
##      Df      Sum Sq    Mean Sq F value Pr(>F)
## Ident    7 3.3229e+10 4.7471e+09  0.1999 0.9832
## Residuals 32 7.5997e+11 2.3749e+10

```

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

## Optimització

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

```

NomsModelsRISW <- paste(rep(paste(rep(paste("M.int RIS",c("MLE","REML")),4),
                                rep(c("No.W","VarI","VarE","VarP"),each=2)),3),

```

```

      rep(c("No.t", "Logt", "Bct"), each=8))

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

ResumRISW <- rbind(intM0_QI[[1]],intM1_QI[[1]],intM2_QI[[1]])
rownames(ResumRISW) <- NomsModelsRISW
ResumRISW$R2 <- c(MO_RIS_W_dFA_R2$R2,M1_RIS_W_dFA_R2$R2,M2_RIS_W_dFA_R2$R2)
ResumRISW$Shapiro_p.valor <- c(Shap_MO_df$Shapiro_p.valor,Shap_M1_df$Shapiro_p.valor,Shap_M2_df$Shapiro

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

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

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

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

  ML_table_p <- lapply(1:length(p),function(pvar){
    Table <- data.frame(Model = if(length(p[[pvar]])>1){
      rep(rownames(df), ncol(df[,p[[pvar]]]))
    } else{
      rownames(df)
    })
    Table$Model <- factor(Table$Model, levels = rownames(df))
    Table$ParQ <- unlist(df[,p[[pvar]])
    Table$Par <- rep(colnames(df)[p[[pvar]]],each=nrow(df))
    return(Table)
  })

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

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

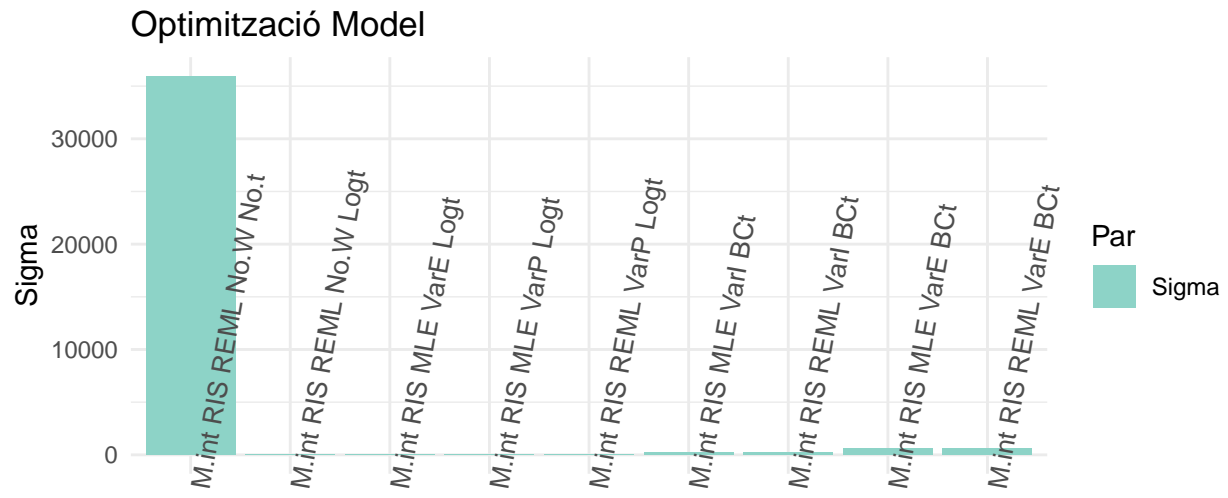
BestFi <- sort(unique(as.vector(sapply(1:ncol(ResumRISW),function(v){

```

```
order(ResumRISW[,v],decreasing=optorder[v])[1:2]
)))))
```

```
Convdf_plot(ResumRISW[BestFi,])
```

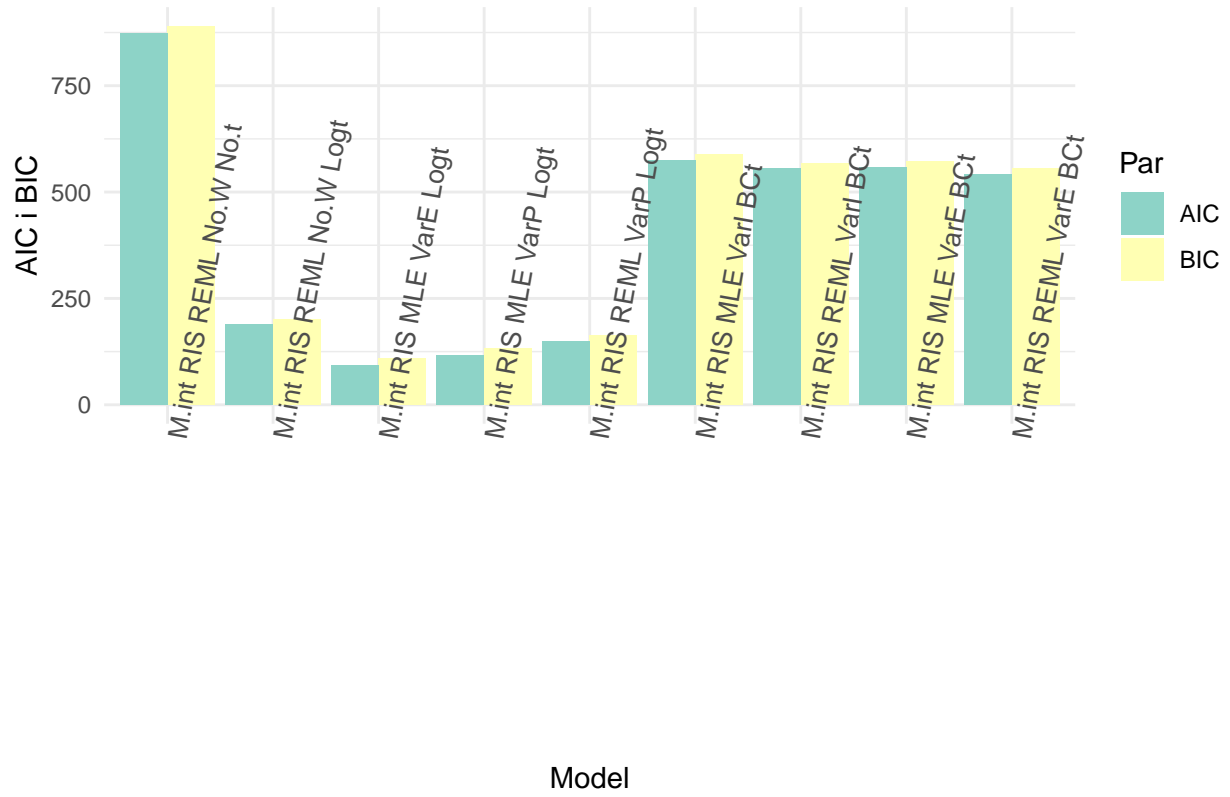
```
## [[1]]
```



Model

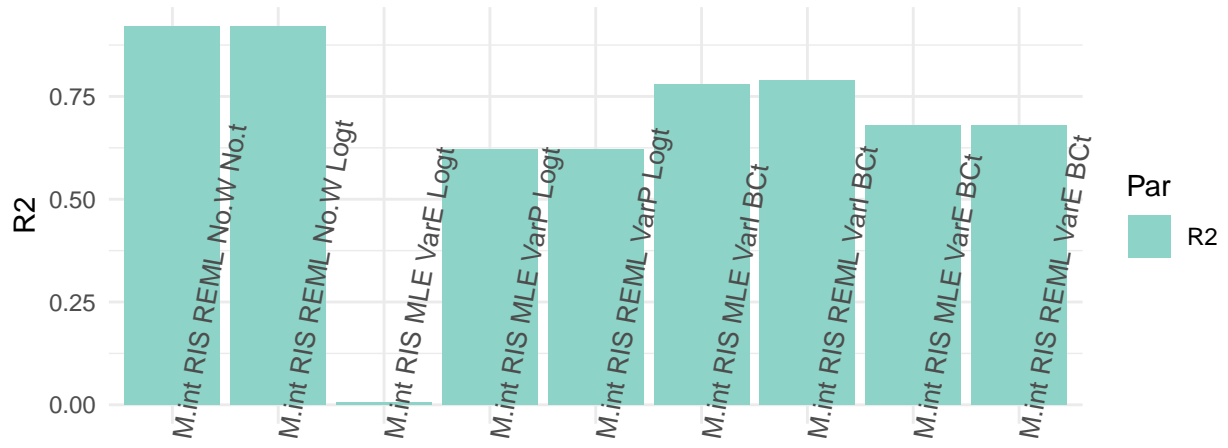
```
##  
## [[2]]
```

## Optimització Model



##  
## [[3]]

# Optimització Model

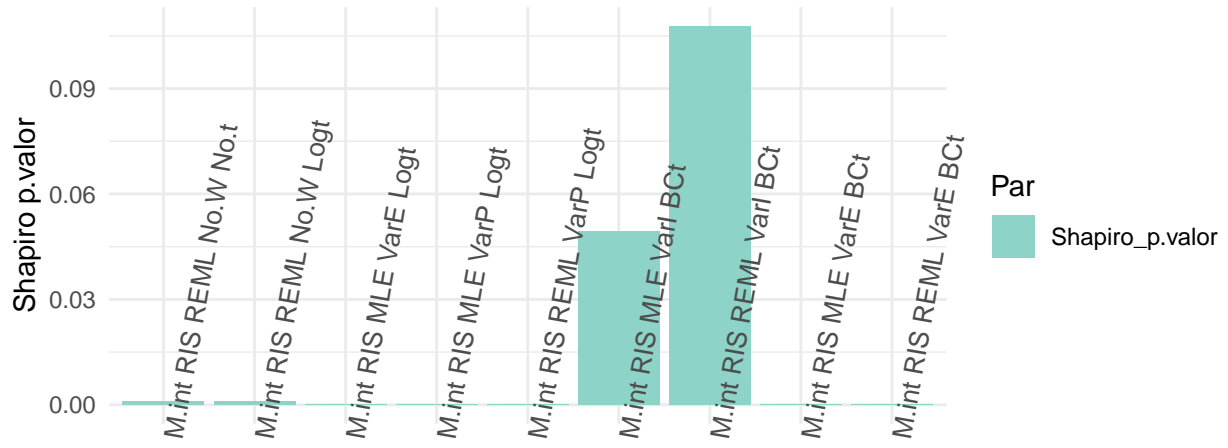


##  
## [[4]]

Model

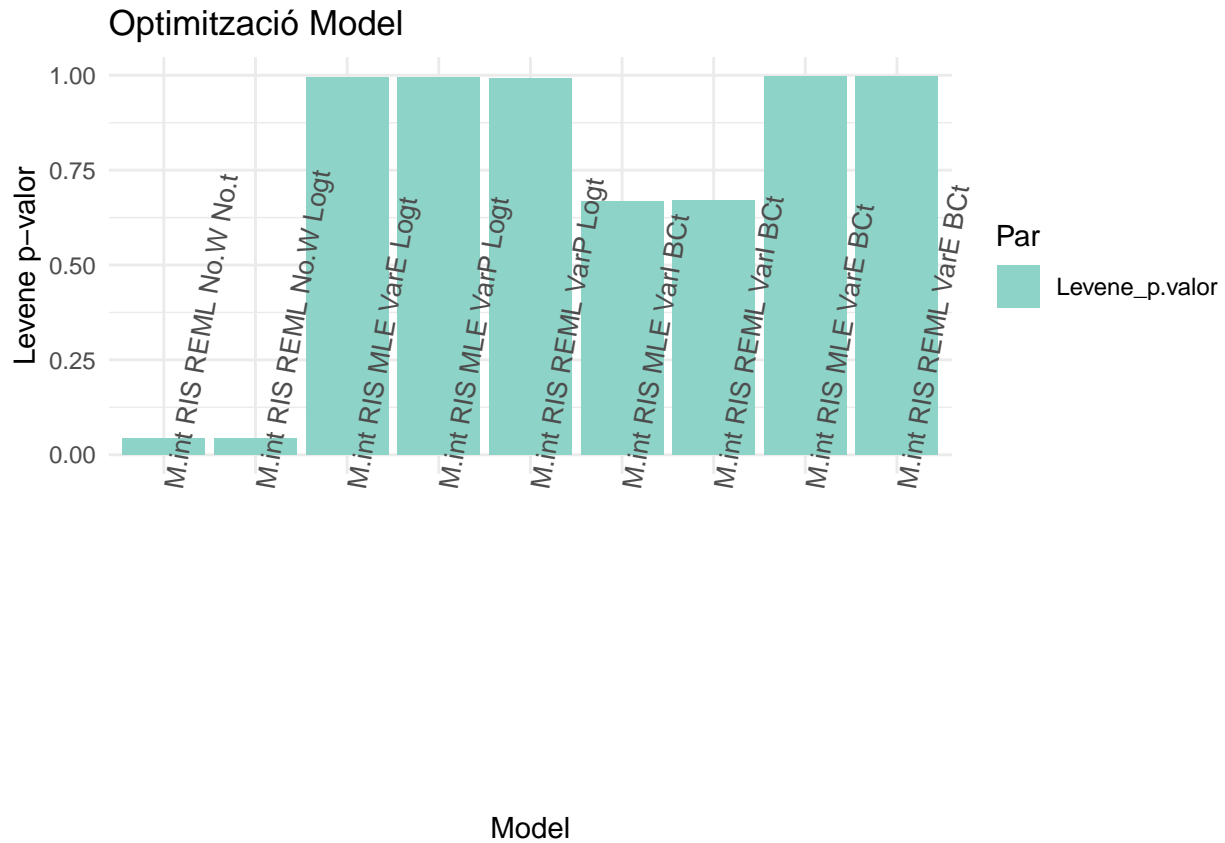


## Optimització Model



Model

```
##  
## [[5]]
```



```

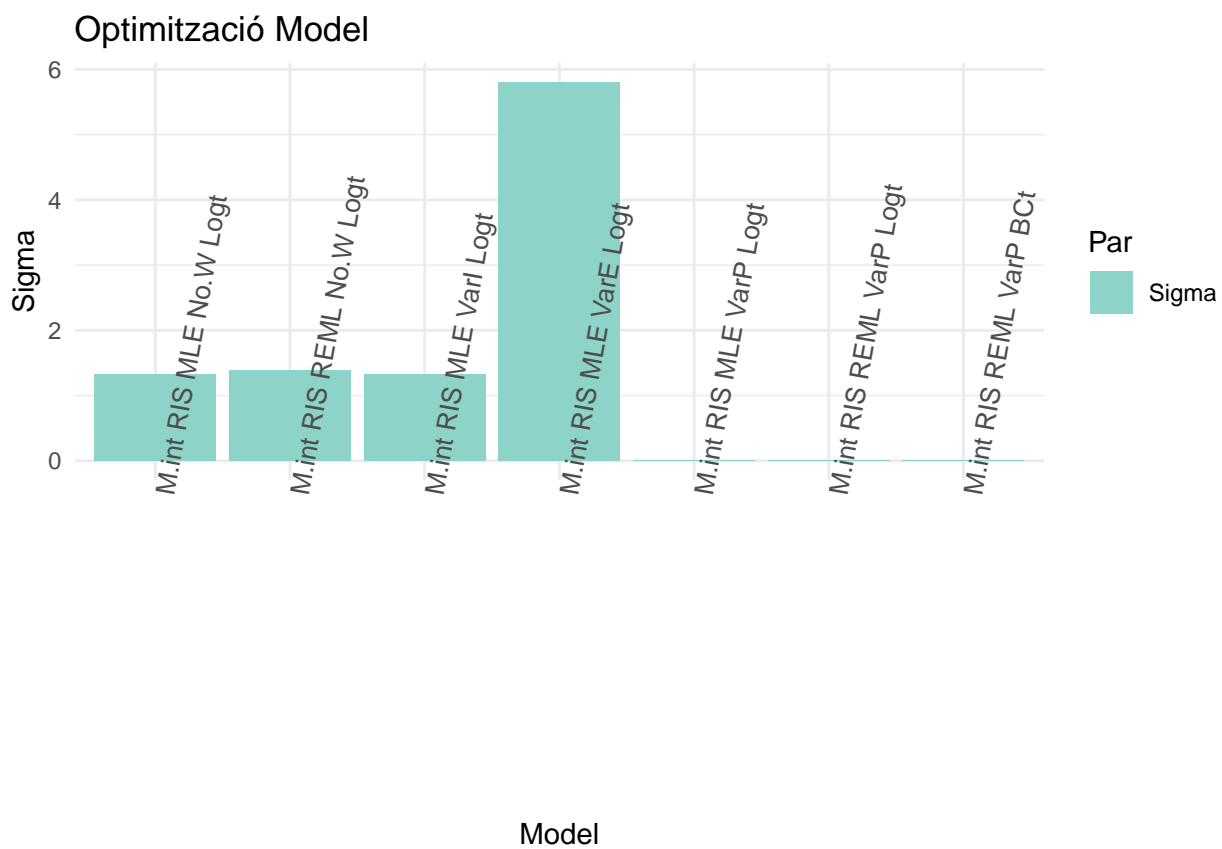
#No es considera el shapiro ni els models amb Sigmes massa grans
ResumRISW2 <- ResumRISW[~which(ResumRISW$Sigma>100),]

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

Convdf_plot(ResumRISW2[BestFi2,])

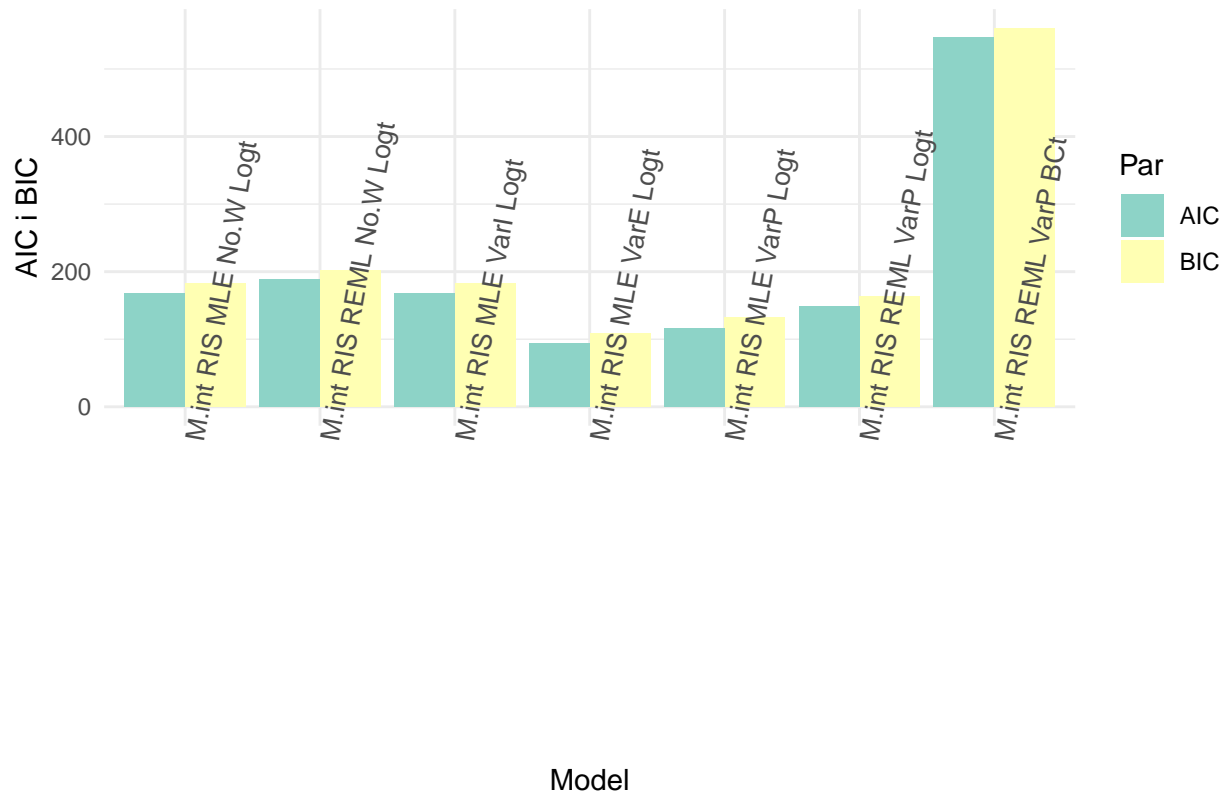
```

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



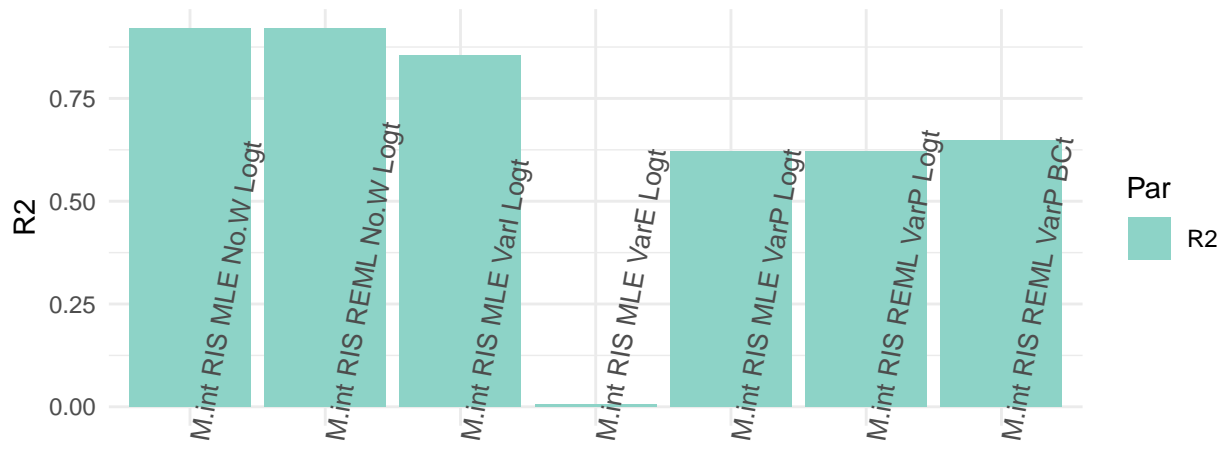
##  
## [[2]]

## Optimització Model



##  
## [[3]]

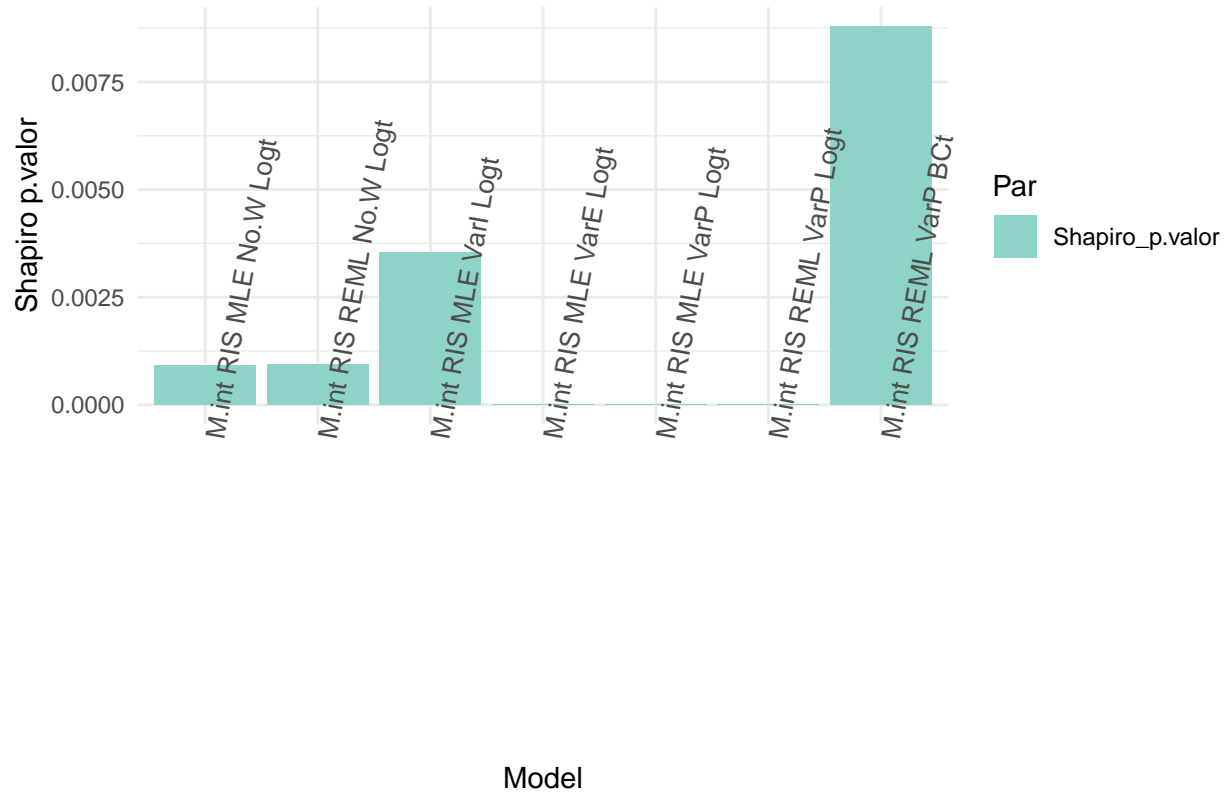
## Optimització Model



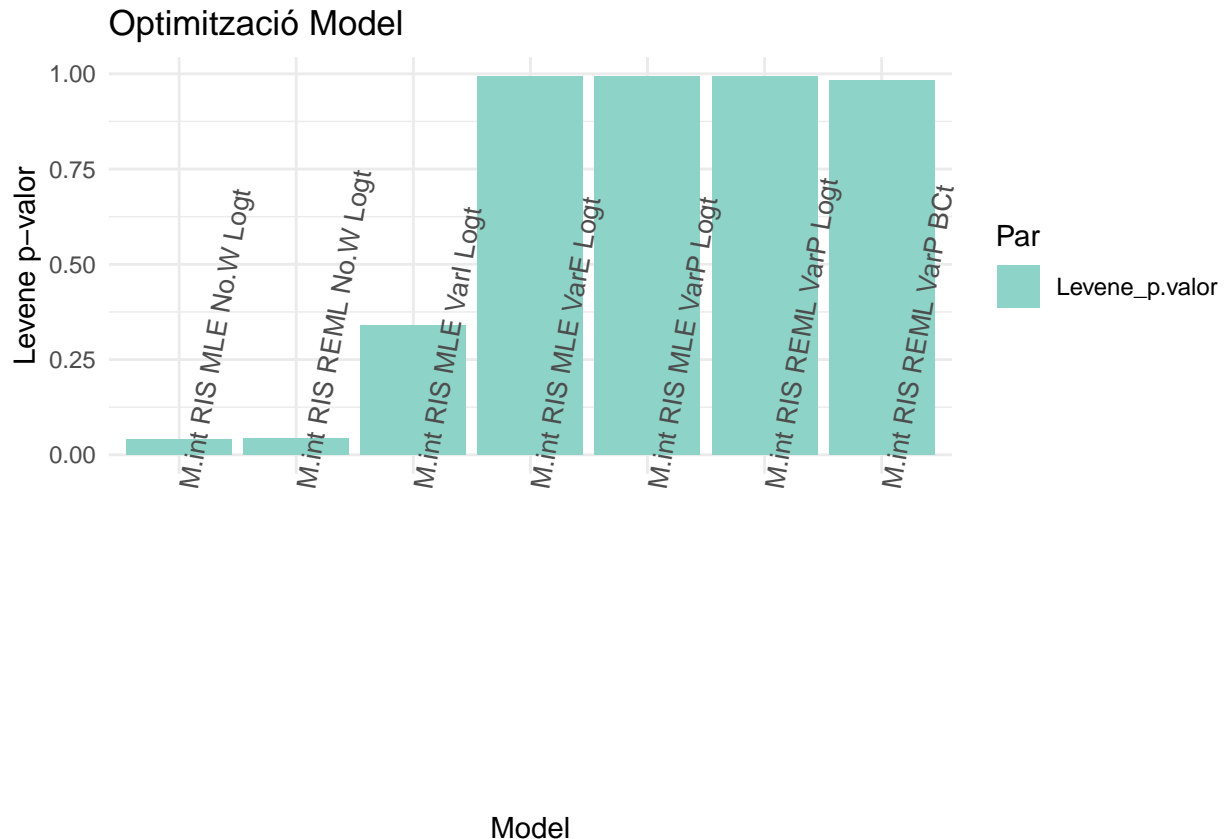
##  
## [[4]]

Model

## Optimització Model



```
##
## [[5]]
```



De manera resumida:

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

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

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

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

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

## [[1]]
## Ident = pdLogChol(1 + Tiempo)
##      Variance      StdDev      Corr
## (Intercept) 2.008409e-19 4.481527e-10 (Intr)
## Tiempo      5.694890e-15 7.546449e-08 -1
```

```
## Residual    2.047354e-14 1.430858e-07
##
## [[2]]
## Ident = pdLogChol(1 + Tiempo)
##      Variance      StdDev      Corr
## (Intercept) 4.485131e-169 6.697112e-85 (Intr)
## Tiempo      3.587467e-172 1.894061e-86 -0.982
## Residual    5.331961e-169 7.302028e-85
```

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

```
## [[1]]
## Ident aocampo 002
## Conditional variance covariance matrix
##      1      2      3      4      5
## 1 4.2487 0.00000 0.000000 0.000000 0.000000
## 2 0.0000 0.51613 0.000000 0.000000 0.000000
## 3 0.0000 0.00000 0.029884 0.000000 0.000000
## 4 0.0000 0.00000 0.000000 0.056424 0.000000
## 5 0.0000 0.00000 0.000000 0.000000 0.051378
## Standard Deviations: 2.0612 0.71842 0.17287 0.23754 0.22667
##
```

```
## [[2]]
## Ident aocampo 002
## Conditional variance covariance matrix
##      1      2      3      4      5
## 1 174080      0      0      0      0
## 2      0 63839      0      0      0
## 3      0      0 22559      0      0
## 4      0      0      0 18890      0
## 5      0      0      0      0 14827
## Standard Deviations: 417.23 252.66 150.2 137.44 121.77
```

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

```
## [[1]]
## [[1]][[1]]
## Ident aocampo 002
## Marginal variance covariance matrix
##      1      2      3      4      5
## 1 4.2487e+00 -4.0562e-16 -8.1144e-16 -1.2173e-15 -1.6231e-15
## 2 -4.0562e-16 5.1613e-01 1.6389e-12 2.4586e-12 3.2782e-12
## 3 -8.1144e-16 1.6389e-12 2.9884e-02 4.9184e-12 6.5581e-12
## 4 -1.2173e-15 2.4586e-12 4.9184e-12 5.6424e-02 9.8379e-12
## 5 -1.6231e-15 3.2782e-12 6.5581e-12 9.8379e-12 5.1378e-02
## Standard Deviations: 2.0612 0.71842 0.17287 0.23754 0.22667
##
## [[1]][[2]]
##      1      2      3      4      5
## 1 1.000000e+00 -2.739135e-16 -2.277243e-15 -2.486133e-15 -3.473972e-15
```



```

## 2 -2.739135e-16  1.000000e+00  1.319637e-11  1.440687e-11  2.013128e-11
## 3 -2.277243e-15  1.319637e-11  1.000000e+00  1.197748e-10  1.673661e-10
## 4 -2.486133e-15  1.440687e-11  1.197748e-10  1.000000e+00  1.827184e-10
## 5 -3.473972e-15  2.013128e-11  1.673661e-10  1.827184e-10  1.000000e+00
##
##
## [[2]]
## [[2]][[1]]
## Ident aocampo 002
## Marginal variance covariance matrix
##           1           2           3           4           5
## 1  1.7408e+05  2.9903e-169  1.4954e-169  5.1212e-173  -1.4944e-169
## 2  2.9903e-169  6.3839e+04  1.0337e-169  5.5425e-171  -9.2285e-170
## 3  1.4954e-169  1.0337e-169  2.2559e+04  1.1034e-170  -3.5134e-170
## 4  5.1212e-173  5.5425e-171  1.1034e-170  1.8890e+04  2.2016e-170
## 5 -1.4944e-169 -9.2285e-170 -3.5134e-170  2.2016e-170  1.4827e+04
## Standard Deviations: 417.23 252.66 150.2 137.44 121.77
##
## [[2]][[2]]
##           1           2           3           4
## 1  1.000000e+00  2.836525e-174  2.386256e-174  8.930590e-178
## 2  2.836525e-174  1.000000e+00  2.723912e-174  1.596060e-175
## 3  2.386256e-174  2.723912e-174  1.000000e+00  5.345081e-175
## 4  8.930590e-178  1.596060e-175  5.345081e-175  1.000000e+00
## 5 -2.941374e-174 -2.999580e-174 -1.921086e-174  1.315546e-174
##           5
## 1 -2.941374e-174
## 2 -2.999580e-174
## 3 -1.921086e-174
## 4  1.315546e-174
## 5  1.000000e+00

```

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

- En els primers resultats s'observen a les matrius D donant poca informació, simplement es veu la baixa escala obtinguda per la variància residual.
- La segona sèria de matrius són les matrius R que reflexen la matriu de variàncies-covariàncies residuals dels diferents temps que tenen tots els lots. S'està en el cas on es considera la variància residual heterogènia i per tant es veu com la funció addicional recalcula de manera que a mesura que s'avança en els punts de temps augmenta la variància pròpia del temps (i si el conjunt és més dispers també més variància es calcula). En els 2 casos sembla que el sistema calcula que a mesura que avança el temps es redueix la variància.
- La tercera sèria de matrius mostra:
  - La matriu primera de cada punt de la sèria és de tipus marginal on es suma la variància residual i la variància degut a l'efecte aleatori en la diagonal, mentre que la variància degut a l'efecte aleatori forma part de la covariància en aquest cas. Això és degut a que en aquesta matriu es reflexa l'efecte de la variància total dins un dels individus (equivalent per tots els altres) que té la variància de l'efecte aleatori sumat a la variància residual al centrar-nos en un punt, i es relaciona amb la resta de punts corresponents als altres temps amb el factor calculat aleatori de la variància dins el mateix lot. En els dos casos hi ha un recàlcul a partir de la funció de la variància que determina com es relacionen les diferents repeticions. NO sembla haver-hi un patró senzill en funció de la distància ja que el sistema ho calcula amb les dades de què disposa.
  - Es reflecteix també a la segona matriu on es plasma la correlació de variàncies i on es veu que s'ha calculat una correlació variant segons quin temps es relaciona amb quin, però baixes en general (és a dir que per aquest cas no s'ha aconseguit trobar un model que expliqui bé aquesta correlació, o

simplement al haver una dispersió en funció del temps baixa la correlació).

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

Els models finals són:

```
lapply(Opt,summary)
```

```
## $'M.Logt.RIS.W-VarP.'  
## Linear mixed-effects model fit by REML  
## Data: LAKE1  
##      AIC      BIC    logLik  
## 148.6902 162.9418 -65.34508  
##  
## Random effects:  
## Formula: ~1 + Tiempo | Ident  
## Structure: General positive-definite, Log-Cholesky parametrization  
##      StdDev      Corr  
## (Intercept) 4.481527e-10 (Intr)  
## Tiempo      7.546449e-08 -1  
## Residual    1.430858e-07  
##  
## Variance function:  
## Structure: Power of variance covariate  
## Formula: ~fitted(.)  
## Parameter estimates:  
##      power  
## 10.53387  
## Fixed effects: intM1$call$formula  
##      Value Std.Error DF   t-value p-value  
## (Intercept) 6.911785 1.2209466 29  5.661005 0.0000  
## Tiempo      -0.070609 0.0293028 29 -2.409634 0.0225  
## CD4P        -0.092617 0.0422670 29 -2.191237 0.0366  
## Tiempo:CD4P 0.002147 0.0010477 29  2.049207 0.0496  
## Correlation:  
##      (Intr) Tiempo CD4P  
## Tiempo      -0.968  
## CD4P        -0.981  0.964  
## Tiempo:CD4P 0.930 -0.981 -0.964  
##  
## Standardized Within-Group Residuals:  
##      Min      Q1      Med      Q3      Max  
## -1.0033954 -0.5022150 -0.1524636  0.5019610  3.3206677  
##  
## Number of Observations: 40  
## Number of Groups: 8  
##  
## $'M.BCt.RIS.W-V'  
## Linear mixed-effects model fit by REML  
## Data: LAKE1  
##      AIC      BIC    logLik  
## 546.2706 560.5222 -264.1353  
##  
## Random effects:
```

```

## Formula: ~1 + Tiempo | Ident
## Structure: General positive-definite, Log-Cholesky parametrization
##           StdDev      Corr
## (Intercept) 6.697112e-85 (Intr)
## Tiempo      1.894061e-86 -0.982
## Residual    7.302028e-85
##
## Variance function:
## Structure: Power of variance covariate
## Formula: ~fitted(.)
## Parameter estimates:
##   power
## 22.41349
## Fixed effects: intM2$call$formula
##           Value Std.Error DF   t-value p-value
## (Intercept) 7946.899  392.2934 29 20.257541 0.0000
## Tiempo      -24.123   10.5765 29 -2.280817 0.0301
## CD4P        -22.654   16.5012 29 -1.372849 0.1803
## Tiempo:CD4P  0.619     0.4382 29  1.412335 0.1685
## Correlation:
##           (Intr) Tiempo CD4P
## Tiempo    -0.936
## CD4P       -0.973  0.912
## Tiempo:CD4P 0.921 -0.977 -0.945
##
## Standardized Within-Group Residuals:
##           Min      Q1      Med      Q3      Max
## -1.33010638 -0.80255701 -0.04421368  0.69241594  1.86325484
##
## Number of Observations: 40
## Number of Groups: 8

```