Trabalho computacional ME

importando libs

```r
library(lme4)
```

```
## Warning: pacote 'lme4' foi compilado no R versão 4.4.2
```

```
## Carregando pacotes exigidos: Matrix
```

```
## Warning: pacote 'Matrix' foi compilado no R versão 4.4.2
```

```r
library(data.table)
library(zoo)
```

```
##
## Anexando pacote: 'zoo'
```

```
## Os seguintes objetos são mascarados por 'package:data.table':
##
##     yearmon, yearqtr
```

```
## Os seguintes objetos são mascarados por 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(ggplot2)
library(GGally)
```

```
## Warning: pacote 'GGally' foi compilado no R versão 4.4.2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
library(zoo)
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(data.table)
library(ggplot2)
library(gridExtra)
```

```
## Warning: pacote 'gridExtra' foi compilado no R versão 4.4.2
```

```r
library(reshape2)
```

```
## Warning: pacote 'reshape2' foi compilado no R versão 4.4.2
```

```
##
## Anexando pacote: 'reshape2'
```

```
## Os seguintes objetos são mascarados por 'package:data.table':
##
##     dcast, melt
```

```r
library(glmnet)
```

```
## Warning: pacote 'glmnet' foi compilado no R versão 4.4.2
```

```
## Loaded glmnet 4.1-8
```

AirQuality

```r
path = "C:\\Users\\mateu\\Documents\\MEGA\\Matérias UFC CD\\Modelagem Estatistica\\trabalhoPratico\\tra
dataAir = fread(path)
```

```r
head(dataAir, n = 4)
```

```
##    rownames Ozone Solar.R  Wind  Temp Month   Day
##       <int> <int>   <int> <num> <int> <int> <int>
## 1:        1    41     190   7.4    67     5     1
## 2:        2    36     118   8.0    72     5     2
## 3:        3    12     149  12.6    74     5     3
## 4:        4    18     313  11.5    62     5     4
```

Analise exploratória dos dados

Quantidade de instancias

```r
length(dataAir$rownames)
```

```
## [1] 153
```

Verificando quantidade de valores únicos

```
dataAir[, lapply( .SD, uniqueN)]
```

```
##    rownames Ozone Solar.R  Wind  Temp Month   Day
##       <int> <int>   <int> <int> <int> <int> <int>
## 1:      153    68     118    31    40     5    31
```

Observe que *rownames* é PK onde a quantidade de valores únicos é igual ao número de linhas do conj. de dados. Será removido do conj. de dados

```
dataAir[,rownames:=NULL]
```

Verificando valores nulos no conj. de dados

```
dataAir[, lapply(.SD,
                 FUN = function(x){
                     return(sum(as.numeric(is.na(x))))
                     })]
```

```
##    Ozone Solar.R  Wind  Temp Month   Day
##    <num>   <num> <num> <num> <num> <num>
## 1:    37       7     0     0     0     0
```

Ozone tem valores nulos

```
dados.null = dataAir[is.na(dataAir$Ozone)==TRUE,.SD]
dados.null
```

```
##     Ozone Solar.R  Wind  Temp Month   Day
##     <int>   <int> <num> <int> <int> <int>
##  1:    NA      NA  14.3    56     5     5
##  2:    NA     194   8.6    69     5    10
##  3:    NA      66  16.6    57     5    25
##  4:    NA     266  14.9    58     5    26
##  5:    NA      NA   8.0    57     5    27
##  6:    NA     286   8.6    78     6     1
##  7:    NA     287   9.7    74     6     2
##  8:    NA     242  16.1    67     6     3
##  9:    NA     186   9.2    84     6     4
## 10:    NA     220   8.6    85     6     5
## 11:    NA     264  14.3    79     6     6
## 12:    NA     273   6.9    87     6     8
## 13:    NA     259  10.9    93     6    11
## 14:    NA     250   9.2    92     6    12
## 15:    NA     332  13.8    80     6    14
## 16:    NA     322  11.5    79     6    15
## 17:    NA     150   6.3    77     6    21
## 18:    NA      59   1.7    76     6    22
## 19:    NA      91   4.6    76     6    23
## 20:    NA     250   6.3    76     6    24
## 21:    NA     135   8.0    75     6    25
## 22:    NA     127   8.0    78     6    26
```

```
## 23:    NA      47  10.3    73     6    27
## 24:    NA      98  11.5    80     6    28
## 25:    NA      31  14.9    77     6    29
## 26:    NA     138   8.0    83     6    30
## 27:    NA     101  10.9    84     7     4
## 28:    NA     139   8.6    82     7    11
## 29:    NA     291  14.9    91     7    14
## 30:    NA     258   9.7    81     7    22
## 31:    NA     295  11.5    82     7    23
## 32:    NA     222   8.6    92     8    10
## 33:    NA     137  11.5    86     8    11
## 34:    NA      64  11.5    79     8    15
## 35:    NA     255  12.6    75     8    23
## 36:    NA     153   5.7    88     8    27
## 37:    NA     145  13.2    77     9    27
##     Ozone Solar.R  Wind  Temp Month   Day
```

> Como a maioria dos dados nulos são a target, vou usar as futuras features para predição dessas target

Removendo os valores nulos

```r
dataAir = na.omit(dataAir, 'Ozone')
dataAir = na.omit(dataAir, 'Solar.R')
```

```r
dataAir[, lapply(.SD,
                 FUN = function(x){
                     return(sum(as.numeric(is.na(x))))
                     })]
```

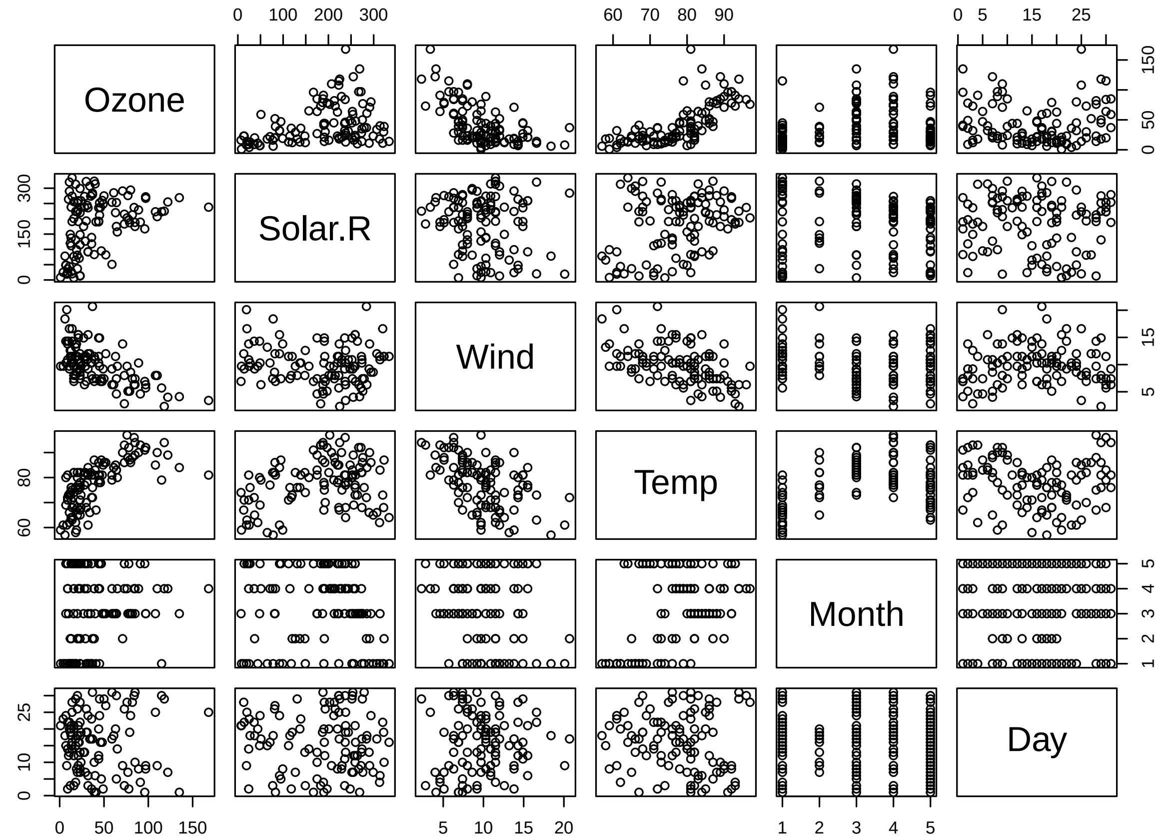
```
##    Ozone Solar.R  Wind  Temp Month   Day
##    <num>   <num> <num> <num> <num> <num>
## 1:     0       0     0     0     0     0
```

> Os dados nulos foram removidos do conjunto de dados

```r
dataAir[, Month:=as.factor(Month)]
```

Fazendo o pairplot dos atributos para verificar o tipo de relação entre eles se é linear ou não

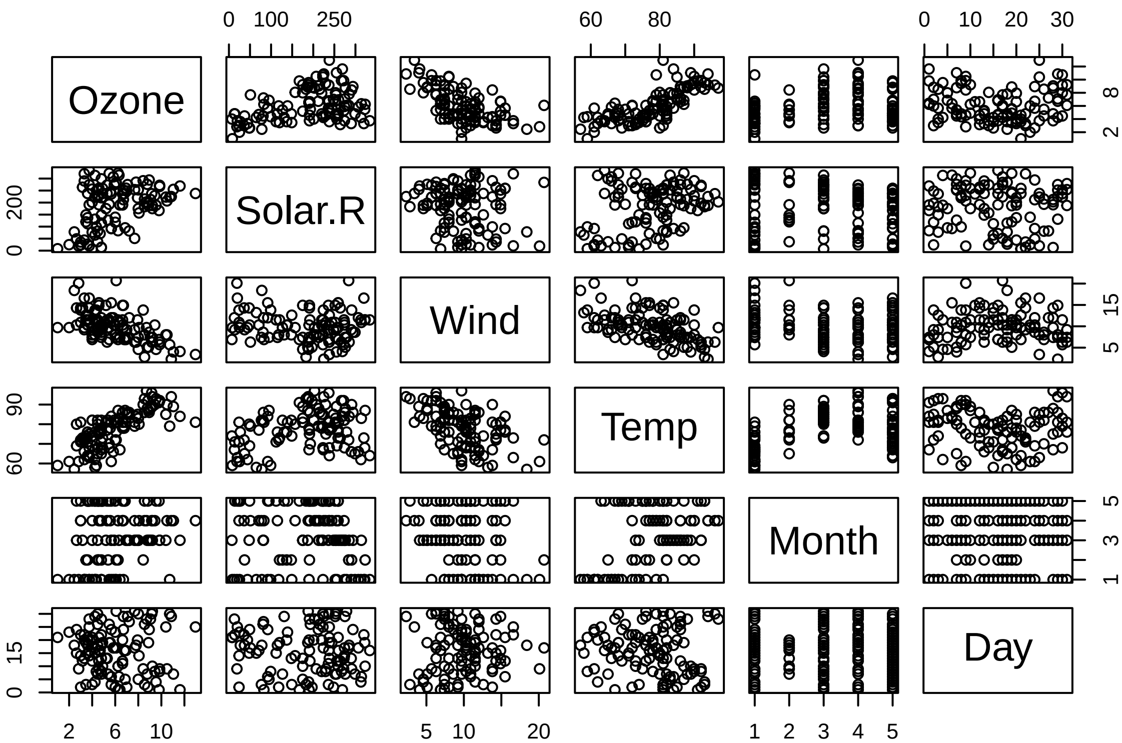
```r
pairs(dataAir)
```




```r
air = copy(dataAir)
```

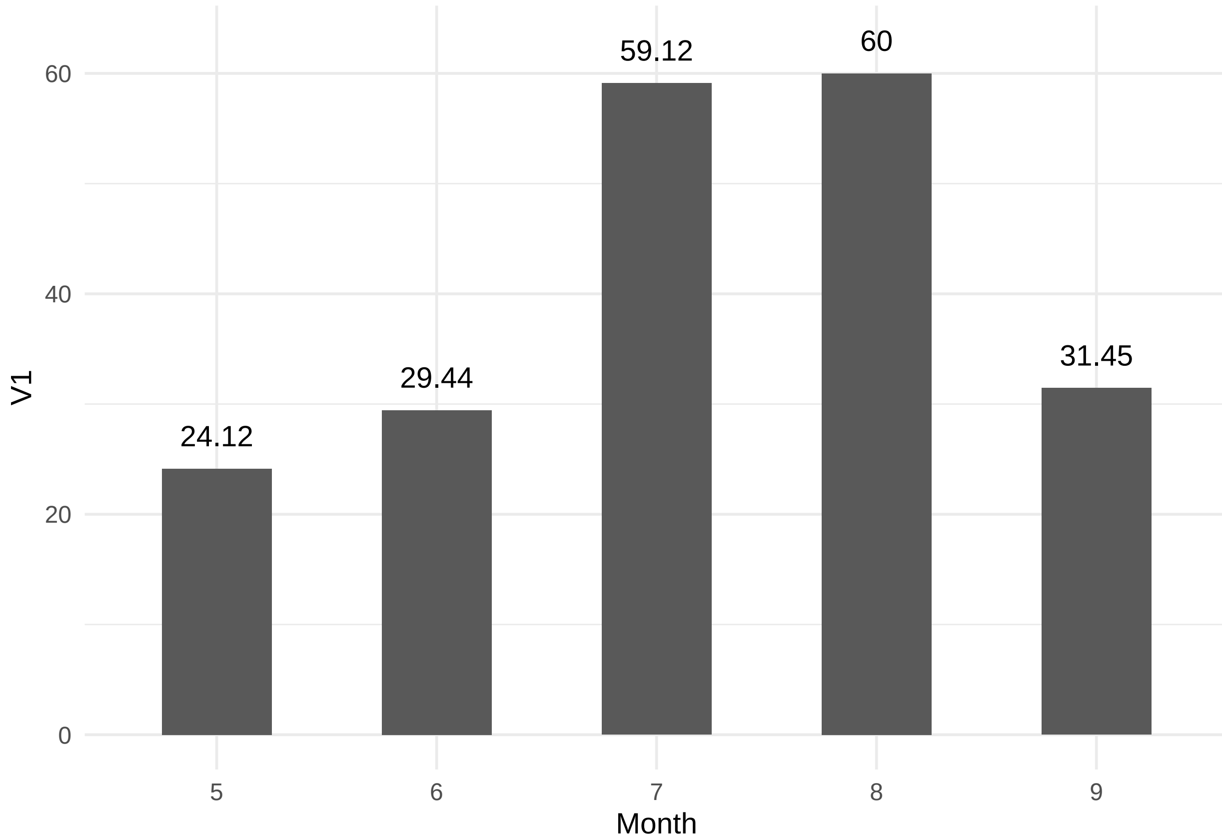
Fazendo a transformação da raiz quadrada para *Ozone*

```r
air[,Ozone:=sqrt(Ozone)]
pairs(air)
```



Verificando a ozonio com base nos meses e dias

```
subset1_ = dataAir[, mean(Ozone), by=Month]

ggplot(subset1_, aes(x=Month, y=V1)) +
  geom_bar(stat = "identity", width = 0.5) +
  geom_text(aes(label = round(V1, 2)), nudge_y = 3)+
  theme_minimal()
```



Em média o Ozonio tem maior numeros no mes 8, 7

Verificando verificando o dia do mes com maior media de ozonio

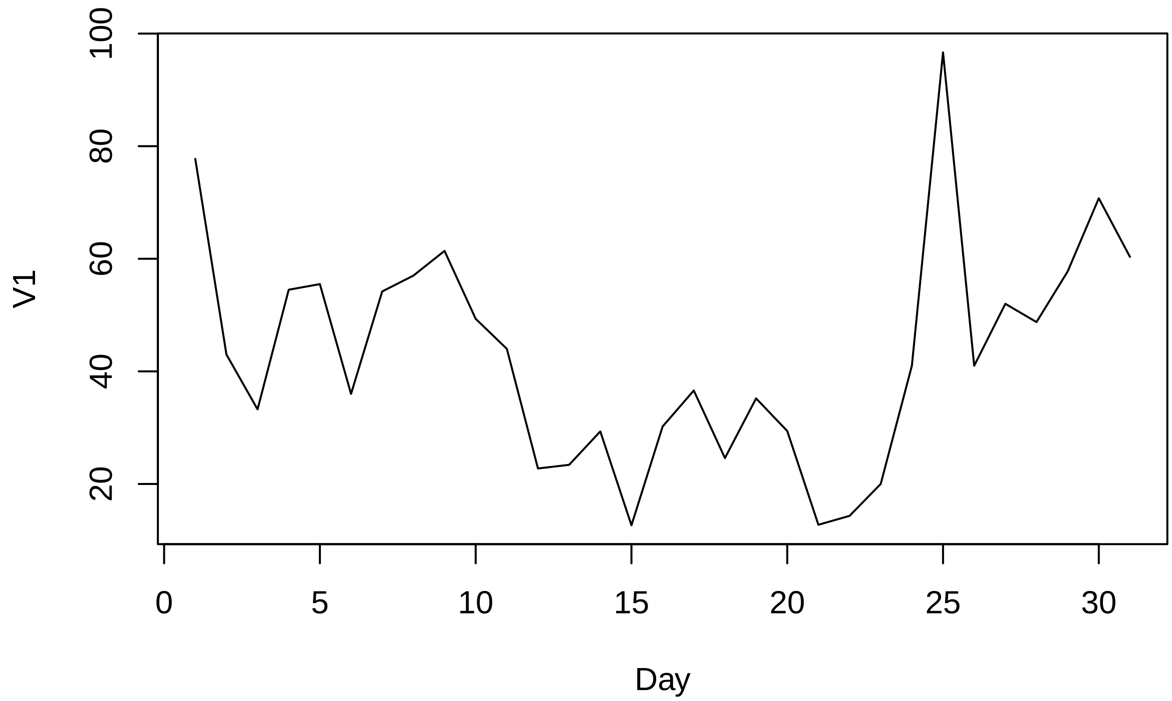
```
subset = dataAir
subset[, mes.dia := paste(dataAir$Month, dataAir$Day, sep='-')]
subset = subset[, mean(Ozone), by=mes.dia][order(V1, decreasing = TRUE)][1]
subset
```

```
##    mes.dia    V1
##     <char> <num>
## 1:    8-25   168
```

No dia 25 do mes 8 é o dia de com media de maior ozonio

Verificando a media de ozonio em relação aos dias

```
subset2_ = dataAir[order(Day), mean(Ozone), by=Day]

plot(subset2_, type='l')
```



o dia 25

Verificando a distribuição de cada atributo

```
num_cols <- names(dataAir)[sapply(dataAir, is.numeric)]
num_cols
```

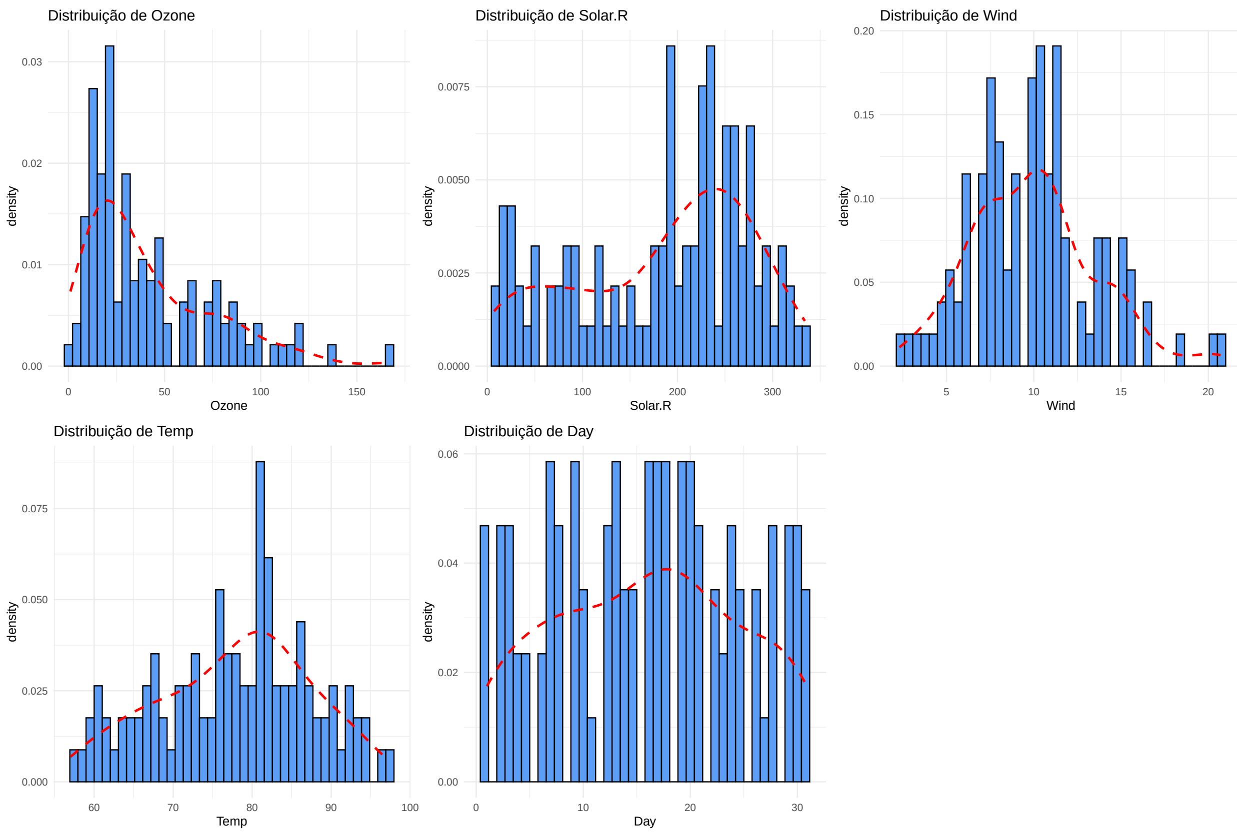
```
## [1] "Ozone"   "Solar.R" "Wind"    "Temp"    "Day"
```

```
plots = lapply(num_cols,
               function(col) {
                 ggplot(dataAir, aes_string(x = col)) +
                   geom_histogram(aes(y = ..density..),bins = 40, fill = "#5c9ef6", color = "black") +
                   geom_density(color = "#FF0000", linewidth = 1, linetype = "dashed", adjust = 1) +
                   ggtitle(paste("Distribuição de", col))  +
                  theme_minimal()
})
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
# Exibindo o primeiro gráfico como exemplo
combined_plot <- do.call(grid.arrange, c(plots, ncol = 3))
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Vemos que os dados não seguem uma distribuição normal

Verificando a correlação dos atributos

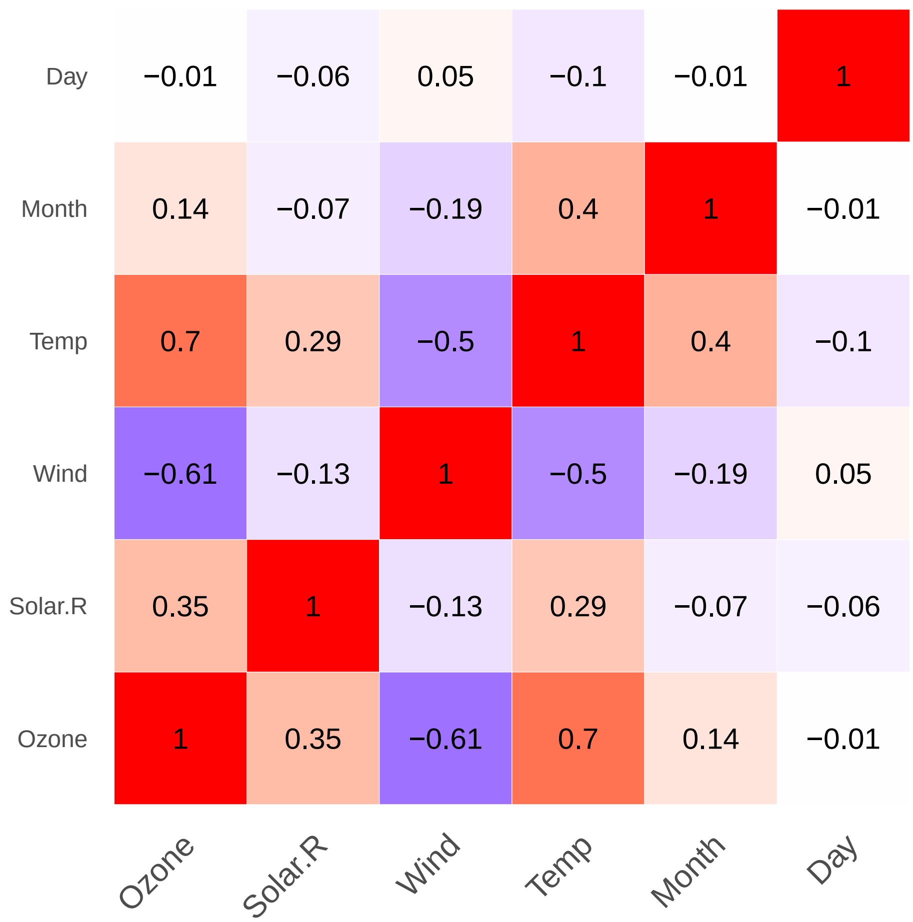
```r
dataAir[, Month:=as.numeric(Month)]
dataAir[, mes.dia:=NULL]
```

```r
dtCor = round(cor(dataAir[,.SD], method="pearson"), 2)
dtCor = melt(dtCor)
```

```r
ggheatmap <- ggplot(dtCor, aes(Var2, Var1, fill = value))+
 geom_tile(color = "white")+
 scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
    name="Pearson\nCorrelation") +
```

```r
  theme_minimal()+ # minimal theme
 theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 12, hjust = 1))+
 coord_fixed()

ggheatmap +
geom_text(aes(Var2, Var1, label = value), color = "black", size = 4) +
theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  panel.grid.major = element_blank(),
  panel.border = element_blank(),
  panel.background = element_blank(),
  axis.ticks = element_blank(),
  legend.justification = c(1, 0),
  legend.position = c(1, 1),
  legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                title.position = "top", title.hjust = 0.5))
```

```
## Warning: A numeric `legend.position` argument in `theme()` was deprecated in ggplot2
## 3.5.0.
## i Please use the `legend.position.inside` argument of `theme()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Criando o modelo de regressão linear

Lendo o arquivo novamente

```
# lendo o arquivo
path = "C:\\Users\\mateu\\Documents\\MEGA\\Matérias UFC CD\\Modelagem Estatistica\\trabalhoPratico\\tra
dataAir = fread(path)

# Removendo Rownames
dataAir[, rownames:=NULL]
dataAir[, Day:=NULL]
# removendo valores nulos
dataAir = na.omit(dataAir, 'Ozone')
dataAir = na.omit(dataAir, 'Solar.R')

# passando para as factor
#dataAir[, Month:=as.factor(Month)]


# Transformação logaritmica
dataAir[, Ozone:=log(Ozone)]
dataAir[, Wind:=log(Wind)]
```

```r
dataAir[, Solar.R:=log(Solar.R)]
dataAir[, Temp:=log(Temp)]

head(dataAir, 3)
```

```
##       Ozone  Solar.R     Wind     Temp Month
##       <num>    <num>    <num>    <num> <int>
## 1: 3.713572 5.247024 2.001480 4.204693     5
## 2: 3.583519 4.770685 2.079442 4.276666     5
## 3: 2.484907 5.003946 2.533697 4.304065     5
```

```r
model = lm(Ozone ~. ,data = dataAir)
model
```

```
##
## Call:
## lm(formula = Ozone ~ ., data = dataAir)
##
## Coefficients:
## (Intercept)      Solar.R         Wind         Temp        Month
##   -11.34466      0.29190     -0.65866      3.47693     -0.04688
```

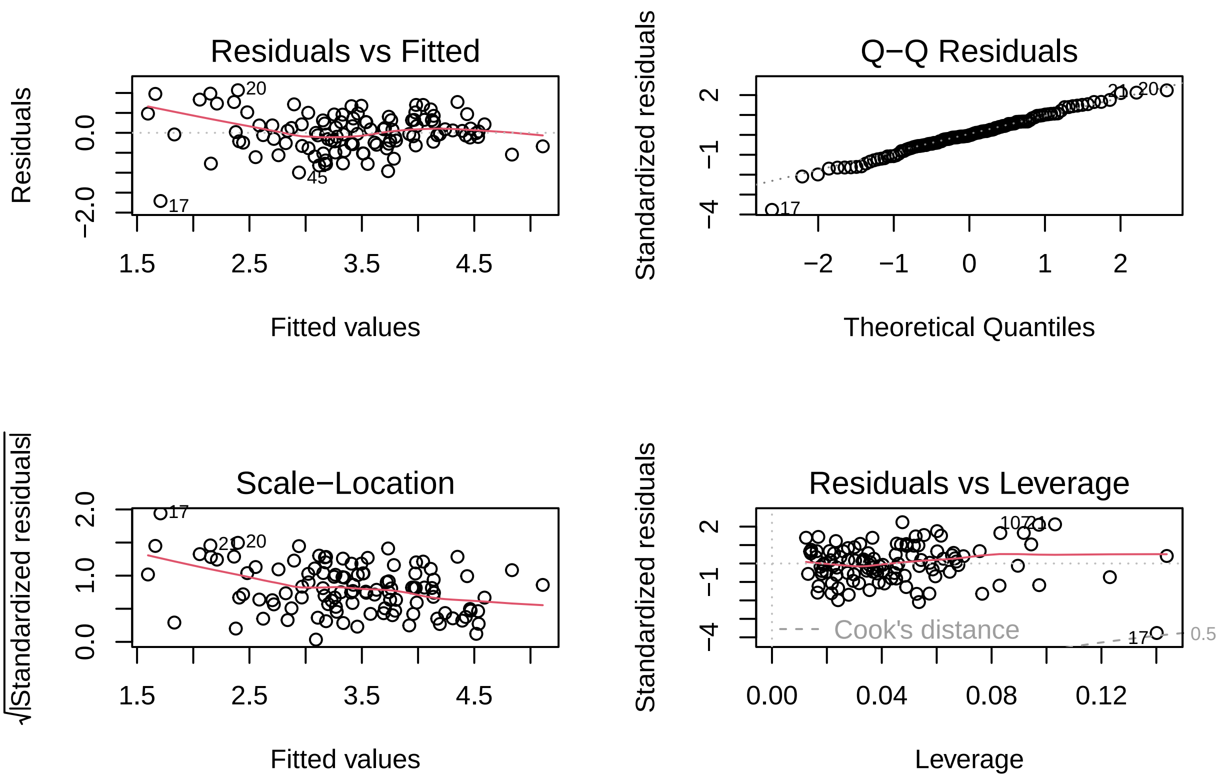
```r
summary(model)
```

```
##
## Call:
## lm(formula = Ozone ~ ., data = dataAir)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.70869 -0.27153 -0.00702  0.31616  1.06837
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11.34466    2.16494  -5.240 8.22e-07 ***
## Solar.R       0.29190    0.05933   4.920 3.19e-06 ***
## Wind         -0.65866    0.13730  -4.797 5.29e-06 ***
## Temp          3.47693    0.50246   6.920 3.57e-10 ***
## Month        -0.04688    0.03537  -1.326    0.188
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4891 on 106 degrees of freedom
## Multiple R-squared:  0.6925, Adjusted R-squared:  0.6809
## F-statistic: 59.67 on 4 and 106 DF,  p-value: < 2.2e-16
```

- Intercepto: Representa o nível estimado de Ozone quando todas as variáveis preditoras são zero. No contexto deste modelo, essa interpretação pode não fazer sentido físico, pois valores como Temp = 0 ou Solar.R = 0 podem ser irreais.
- Solar.R: A cada aumento de 1 unidade em Solar.R (radiação solar), espera-se que o nível de Ozone aumente em 0.29190 unidades, mantendo as outras variáveis constantes. Como o p-valor é muito pequeno, esse efeito é estatisticamente significativo.

- Wind : Cada aumento de 1 unidade na velocidade do vento (Wind) reduz o nível de Ozone em 0.65866 unidades. Esse coeficiente negativo sugere que ventos mais fortes dispersam o ozônio, diminuindo sua concentração.
- Temp : Cada aumento de 1 unidade na temperatura (Temp) aumenta o nível de Ozone em 3.47693 unidades. Isso sugere que temperaturas mais altas favorecem a formação de ozônio.
- Month (não significativo): O coeficiente indica uma leve redução no Ozone à medida que os meses passam, mas o p-valor alto sugere que esse efeito não é estatisticamente significativo. Mas ficara no modelo

```
par(mfrow=c(2, 2))
plot(model)
```



Como a maioria dos dados nulos são a target , então as features serão utilizadas para predição

```
x = dados.null[, .SD, .SDcols = names(dados.null)[2:6]]
x = na.omit(x, 'Solar.R')
x = x[, Day:=NULL]
x = x[, Wind:=log(Wind)]
x = x[, Solar.R:=log(Solar.R)]
x = x[, Temp:=log(Temp)]

expm1(predict.lm(model, newdata = x, interval = 'prediction'))
```

```
##          fit        lwr        upr
```

```
## 1  25.129118  8.728642  69.17740
## 2   5.365360  1.339468  16.31924
## 3   9.906599  2.974792  28.92707
## 4  41.766205 15.047973 112.96756
## 5  31.932140 11.353129  86.79361
## 6  14.885625  4.908320  41.71148
## 7  45.684219 16.460948 123.81661
## 8  52.409019 18.966315 141.86679
## 9  30.241492 10.630072  82.92302
## 10 70.301133 25.587631 190.21115
## 11 64.513617 23.029921 177.61208
## 12 68.827521 24.787289 188.08085
## 13 34.724075 12.281338  95.09043
## 14 37.215863 13.291904 101.18738
## 15 40.574551 14.552306 110.13743
## 16 70.698077 23.160795 211.76677
## 17 41.237055 14.561831 113.63746
## 18 45.115457 16.215659 122.52914
## 19 30.433689 10.814020  82.63595
## 20 34.389542 12.292182  93.22228
## 21 16.804730  5.646598  46.69484
## 22 27.211932  9.534205  74.55512
## 23 13.883843  4.442049  39.70687
## 24 44.001307 15.842843 119.23609
## 25 32.334807 11.465185  88.14504
## 26 38.336003 13.818177 103.42048
## 27 47.808204 16.871523 132.29813
## 28 40.709772 14.701695 109.79728
## 29 39.463423 14.169151 106.93542
## 30 63.201006 22.993459 170.78720
## 31 35.424165 12.615670  96.44066
## 32 20.712267  7.127403  57.00408
## 33 24.548199  8.563983  67.24672
## 34 63.696824 23.224814 171.78477
## 35 20.971880  7.192200  57.92966
```

Acima são os valores preditos e seu intervalo de confiança

Sleep Study

```
path = "C:\\Users\\mateu\\Documents\\MEGA\\Matérias UFC CD\\Modelagem Estatistica\\trabalhoPratico\\tral

sleep = fread(path)

head(sleep, 2)
```

```
##       V1 Reaction  Days Subject
##    <int>    <num> <int>   <int>
## 1:     1 249.5600     0     308
## 2:     2 258.7047     1     308
```

Verificando únicos

```
sleep[, lapply(.SD, uniqueN)]
```

```
##        V1 Reaction  Days Subject
##     <int>    <int> <int>   <int>
## 1:    180      180    10      18
```

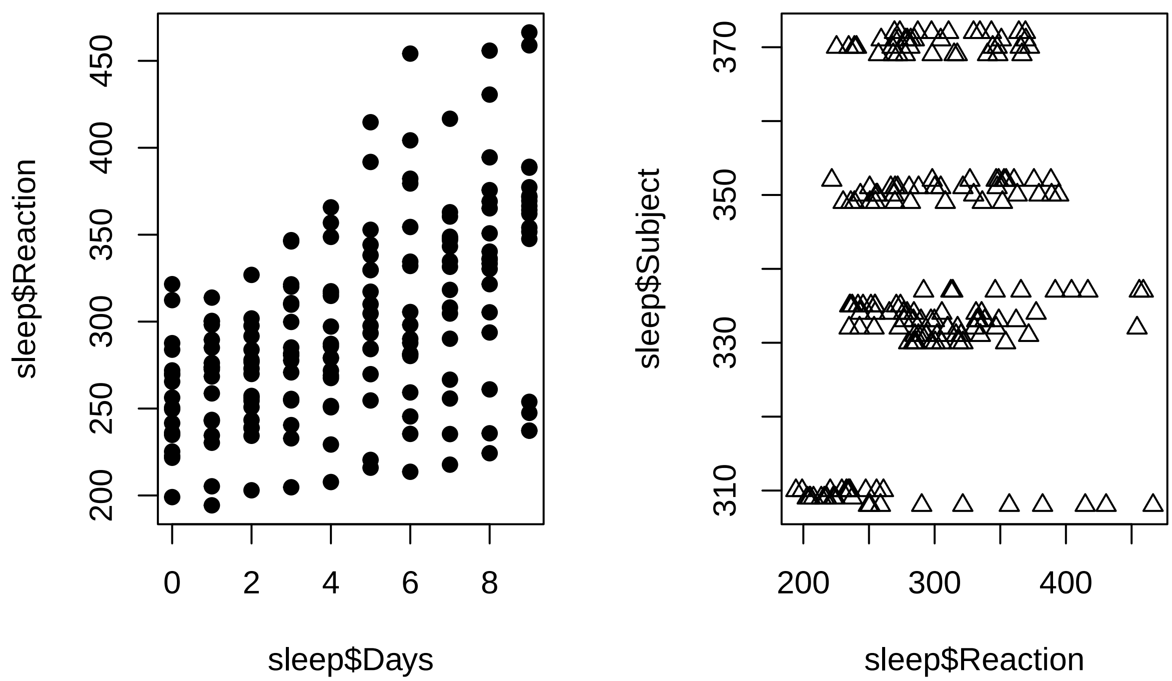
V1 é o numero de intasncia do conj. dados

```
sleep[, V1:=NULL]
```

Analise Exploratória

```
#sleep[, Days:=as.factor(Days)]
```

```
par(mfrow=c(1, 2))
plot(sleep$Days, sleep$Reaction, pch=19)
plot(sleep$Reaction, sleep$Subject, pch=2)
```



Criando modelo

```
model.lmer = lmer(Reaction~Days + (1|Subject), data = sleep)
model.lmer
```

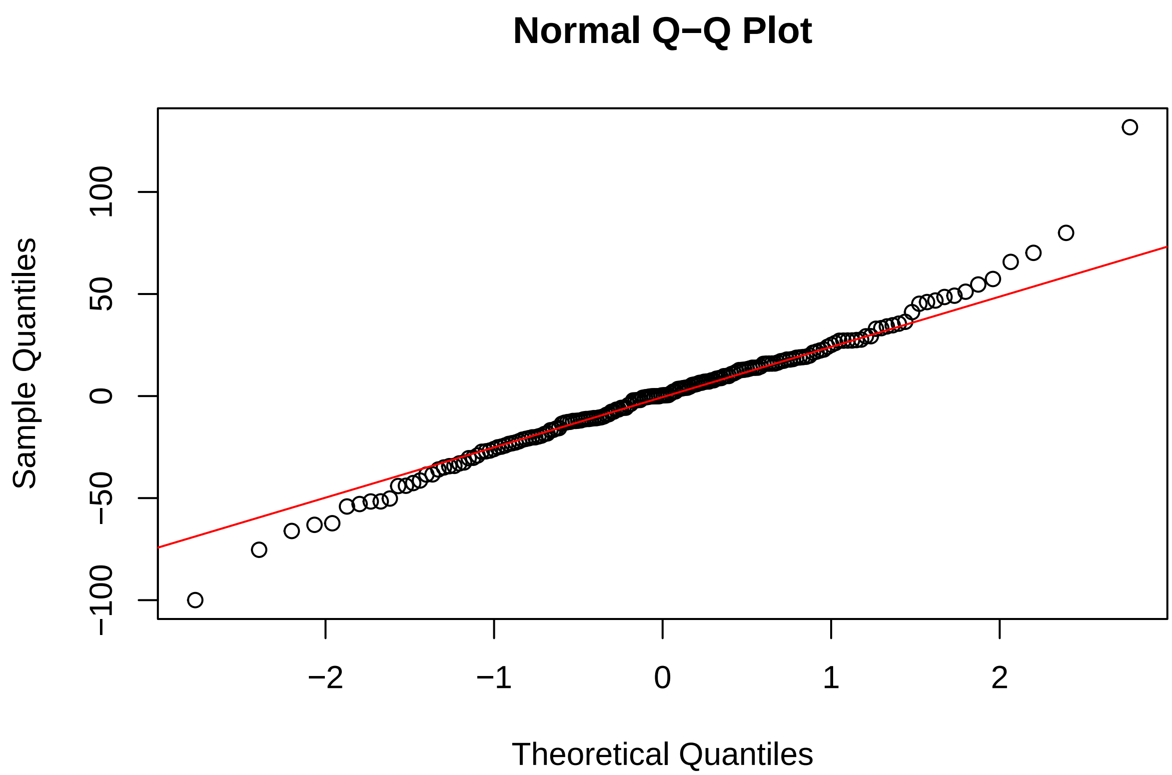
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Reaction ~ Days + (1 | Subject)
##    Data: sleep
## REML criterion at convergence: 1786.465
## Random effects:
##  Groups   Name        Std.Dev.
##  Subject  (Intercept) 37.12
##  Residual             30.99
## Number of obs: 180, groups:  Subject, 18
## Fixed Effects:
## (Intercept)         Days
##      251.41        10.47
```

```
summary(model.lmer)
```

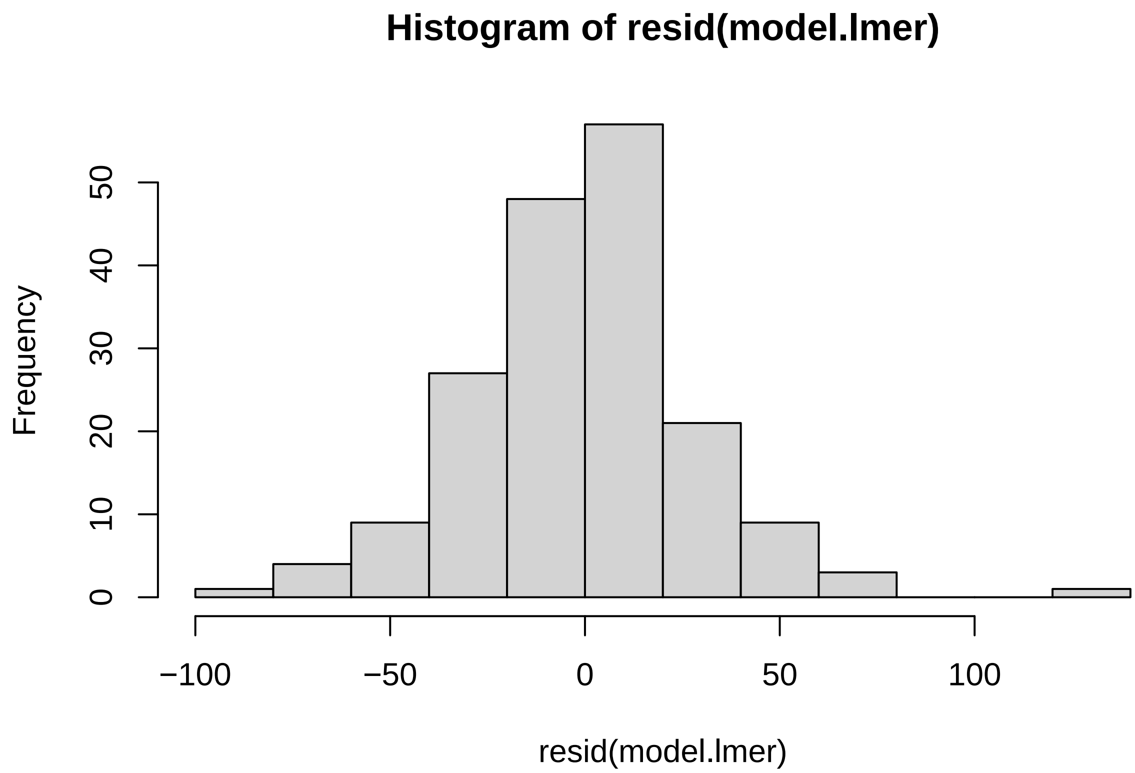
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Reaction ~ Days + (1 | Subject)
##    Data: sleep
##
## REML criterion at convergence: 1786.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2257 -0.5529  0.0109  0.5188  4.2506
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 1378.2   37.12
##  Residual              960.5   30.99
## Number of obs: 180, groups:  Subject, 18
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 251.4051     9.7467   25.79
## Days         10.4673     0.8042   13.02
##
## Correlation of Fixed Effects:
##      (Intr)
## Days -0.371
```

- O efeito fixo de Days é positivo e significativo, sugerindo um crescimento ao longo do tempo.
- Existe variação entre indivíduos, pois o desvio padrão dos efeitos aleatórios é alto (37.12).
- O modelo ajusta tanto variação sistemática (efeitos fixos) quanto diferenças individuais (efeitos aleatórios).

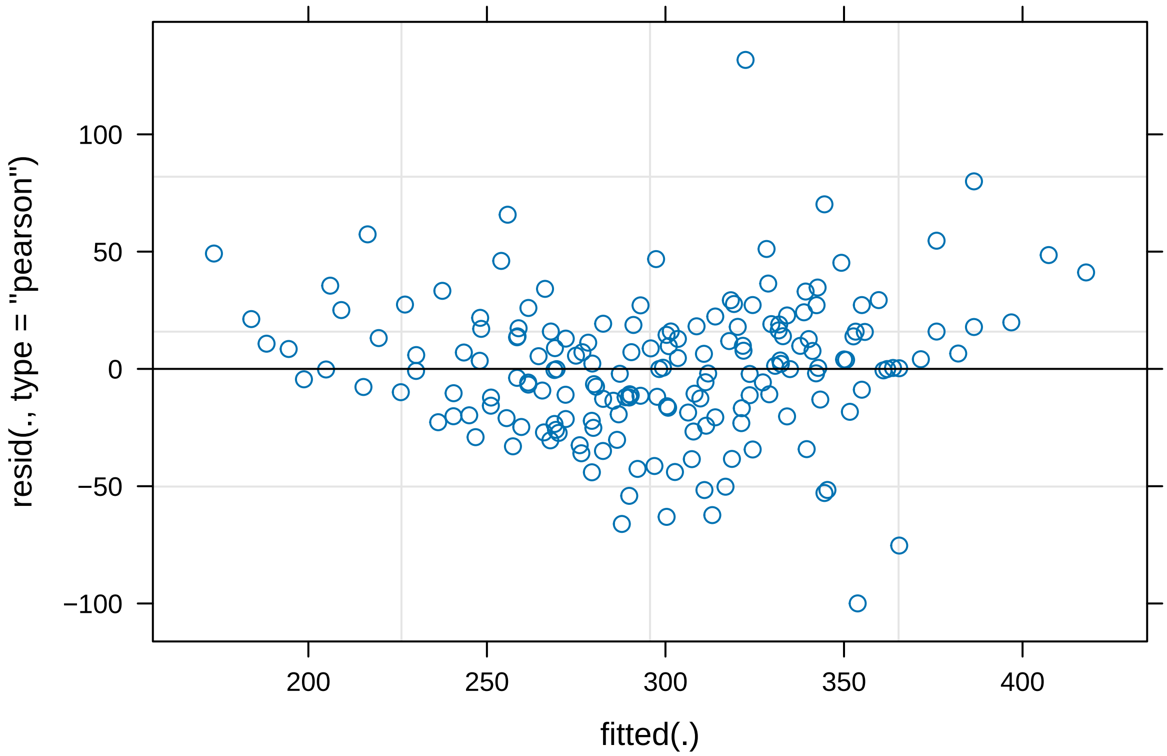
```r
# QQ plot dos resíduos
qqnorm(resid(model.lmer))
qqline(resid(model.lmer), col = "red")
```




```r
hist(resid(model.lmer), 10)
```




```r
plot(model.lmer)
```

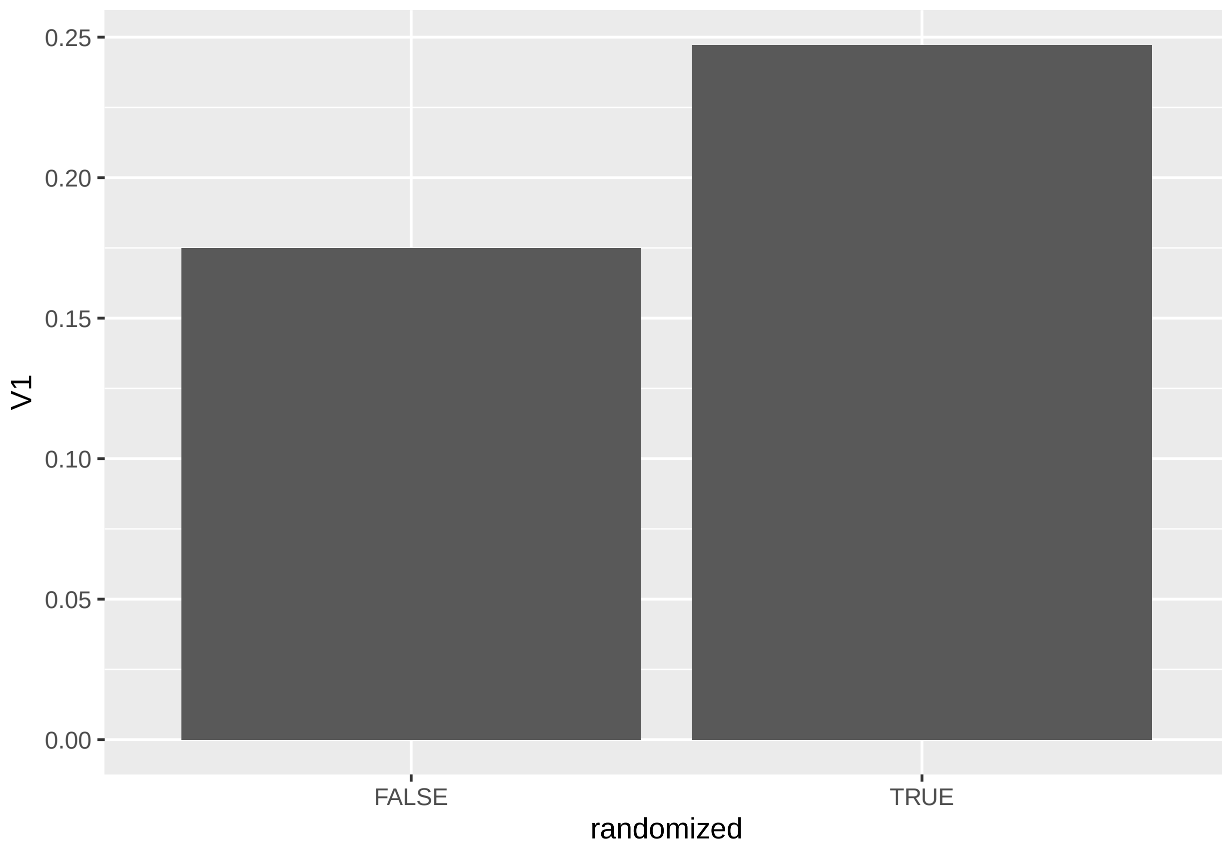


Metanalise

```
path = "C:\\Users\\mateu\\Documents\\MEGA\\Matérias UFC CD\\Modelagem Estatistica\\trabalhoPratico\\tral
meta = fread(path)

head(meta, n = 3)
```

```
##        V1           yi          vi    cluster randomized
##     <int>        <num>       <num>     <char>     <lgcl>
## 1:      1 -0.388300800 0.113865100 ACE 2013a      FALSE
## 2:      2  0.001426734 0.008177298 ACE 2013a      FALSE
## 3:      3 -0.077824834 0.015627721 ACE 2013b      FALSE
```

```
subset = meta[, mean(yi), by=randomized]


ggplot(subset, aes(x = randomized, y=V1)) +
  geom_bar(stat = "identity")
```



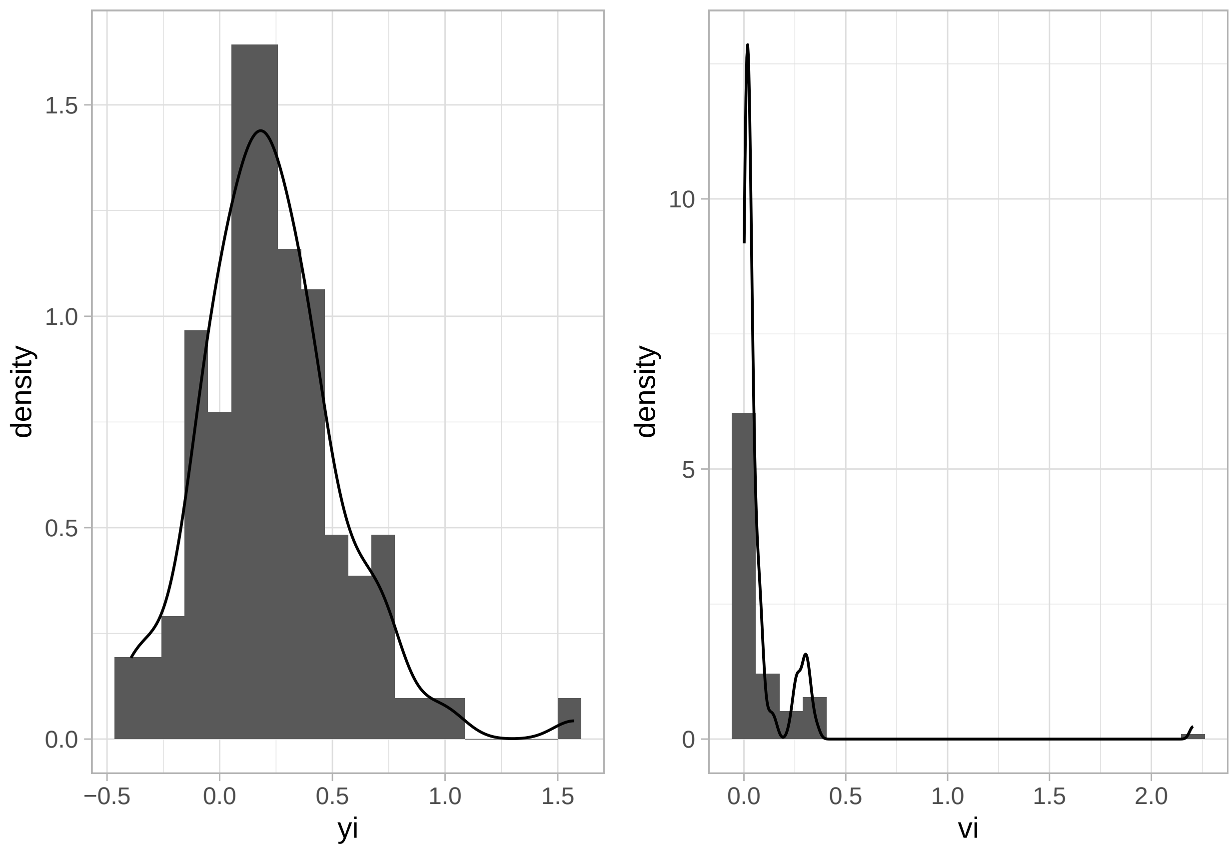

```r
# Pegando o nome do autor
meta[, cluster_name:=unlist(lapply(meta$cluster, function(x){
                                                            return(c(strsplit(x, ' ',fixed = TRUE))[[1]][1])
                                                        }))
]
```

```r
# Pegando o ano

lista_num = unlist(lapply(
  meta$cluster, function(x){
    x = c(strsplit(x, ' ', fixed = TRUE))
    for(i in x[[1]]){

      string = i
      string = sub('a', '', string)
      string = sub('b', '', string)
      if (!anyNA(suppressWarnings(as.numeric(string)))){
        return(as.numeric(string))
    }
  }})
)

meta[, cluster_year := lista_num]
```

Removendo a coluna cluste e V1

```
meta[, cluster:=NULL]
meta[, V1:=NULL]
```

```
plots = lapply(c("yi", "vi"),
               function(x){
                 ggplot(data = meta, mapping = aes_string(x = x)) +
                   geom_histogram(aes(y = ..density..) , bins = 20) +
                   geom_density() +
                   theme_light()
               })

combined_plot <- do.call(grid.arrange, c(plots, ncol = 2))
```




```
shapiro.test(meta$vi)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  meta$vi
## W = 0.32394, p-value < 2.2e-16
```

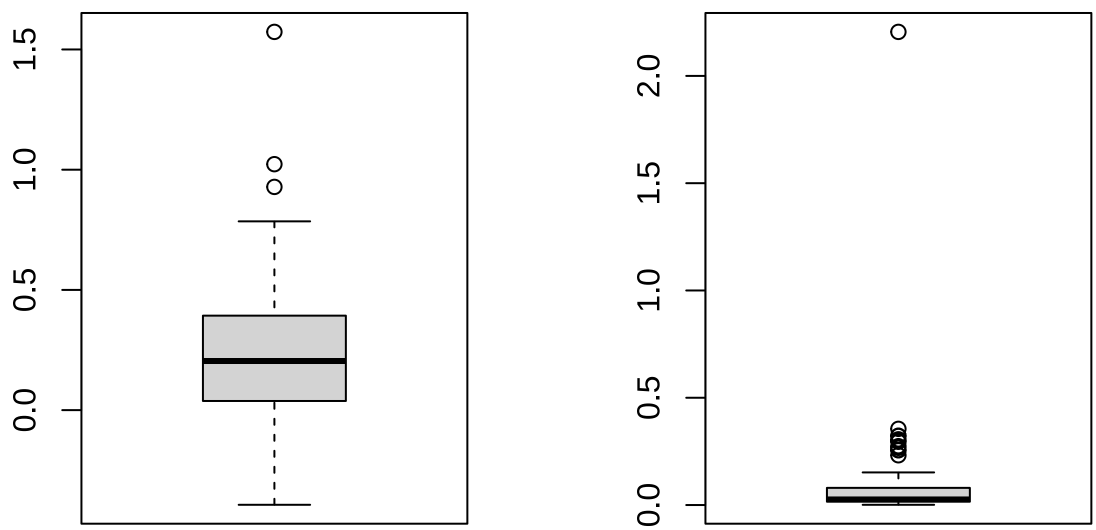
```
shapiro.test(meta$yi)
```

```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  meta$yi
## W = 0.95498, p-value = 0.001792
```

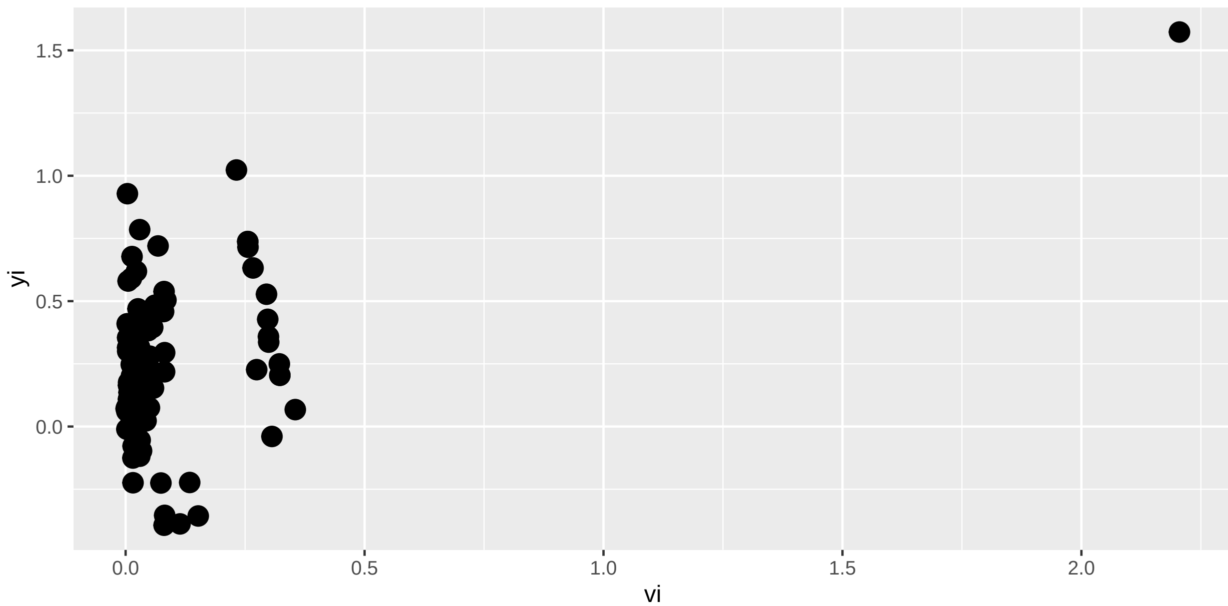
Verificando a presença de outliers nos atributos log da razão de chances e erro padrão

```r
par(mfrow=c(1,2))
boxplot(meta$yi)
boxplot(meta$vi)
```



Vemos a presença de outliers em ambos atributos

```r
ggplot(data = meta, mapping = aes(x = vi, y =yi)) +
  geom_point(size=4) +
  theme(legend.position = "none")
```



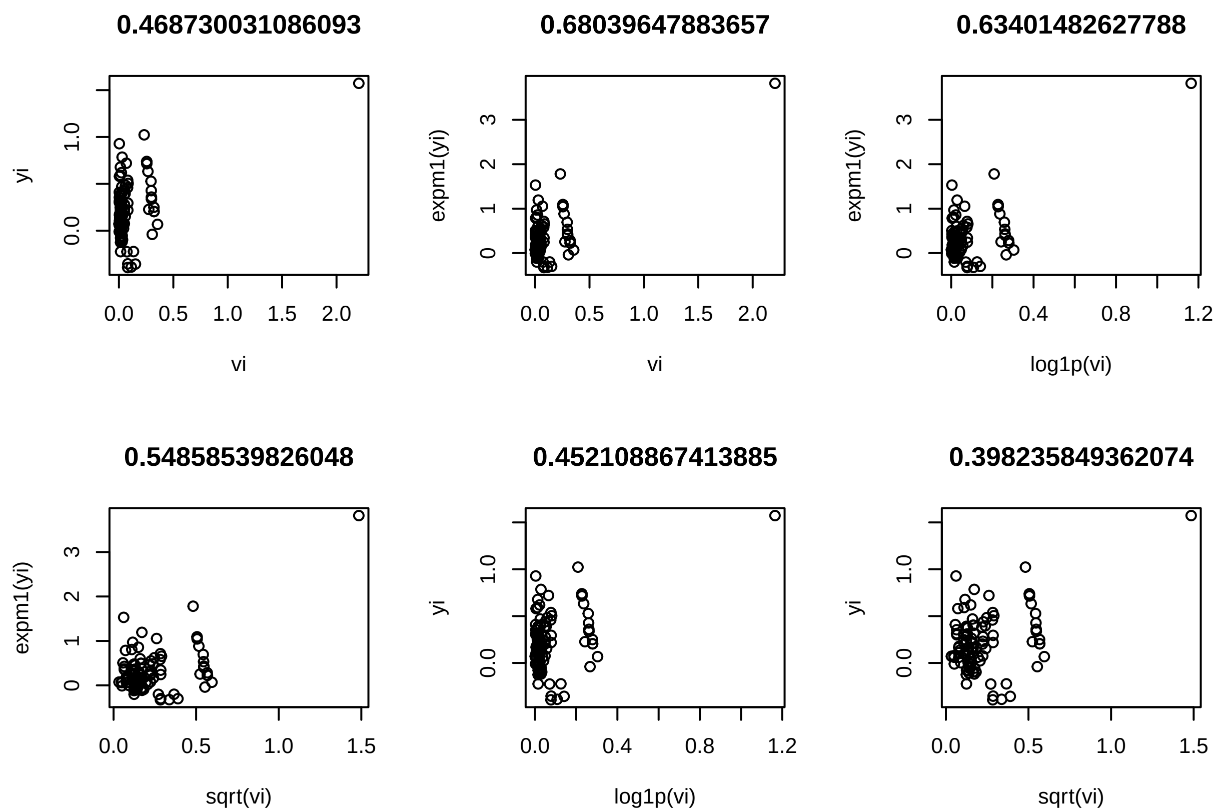
Vemos que o outliers bem distante da concentração dos dados

```
cor(meta$vi, meta$yi)
```

```
## [1] 0.46873
```

Para não remover o outliers vamos ver se a transformação log ou sqrt atenua esse ploblema, além a inversa da log

```
par(mfrow=c(2, 3))

yi = meta$yi
vi = meta$vi
plot(vi, yi)
title(cor(vi, yi))

plot(vi, expm1(yi))
title(cor(vi, expm1(yi)))

plot(log1p(vi), expm1(yi))
title(cor(log1p(vi), expm1(yi)))

plot(sqrt(vi), expm1(yi))
title(cor(sqrt(vi), expm1(yi)))

plot(log1p(vi), yi)
title(cor(log1p(vi), yi))

plot(sqrt(vi), yi)
title(cor(sqrt(vi), yi))
```



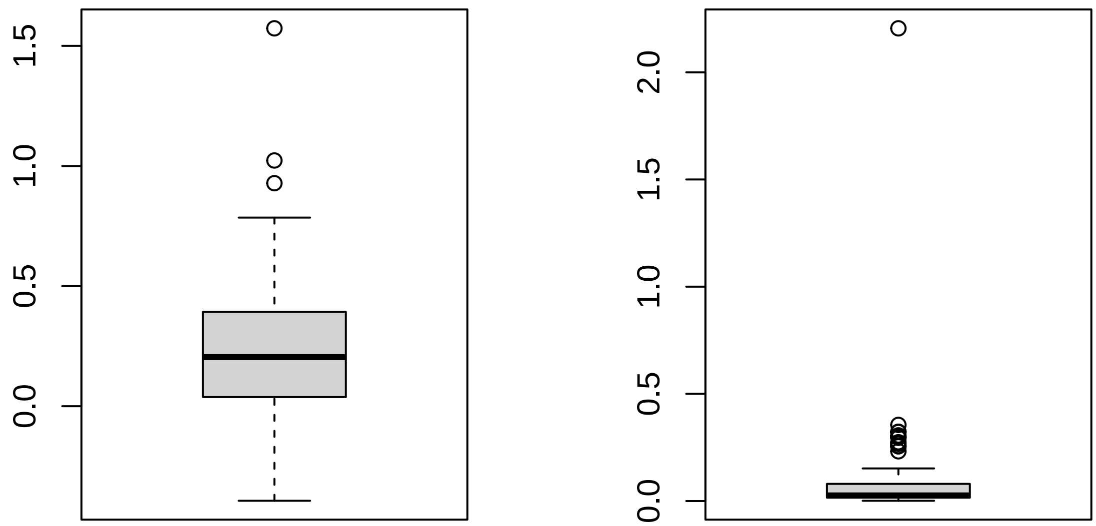
Passando randomized para int

```
meta[, randomized:=as.integer(randomized)]
```

Removendo a instancia com outliers

```
meta.sem.outlier = meta
```

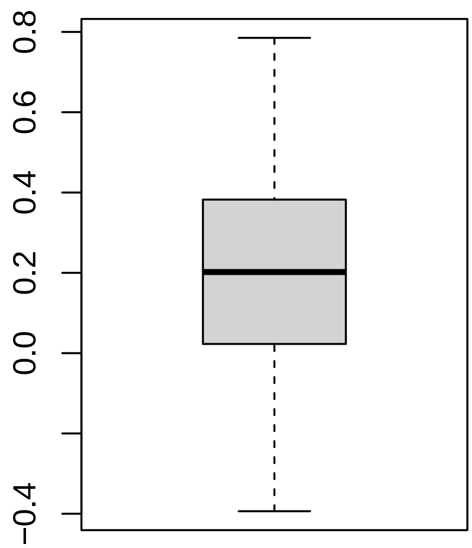
```
par(mfrow=c(1, 2))
boxplot(meta.sem.outlier$yi)
boxplot(meta.sem.outlier$vi)
```

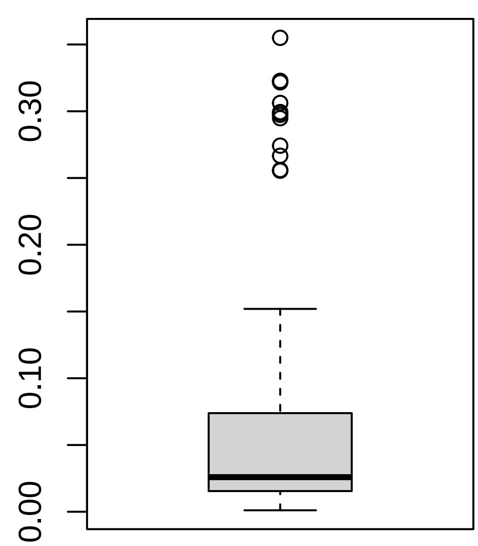


Remover os outliers tendo como base *yi*

```
remove_outliers_iqr <- function(data, column) {
  Q1 <- quantile(data[[column]], 0.25, na.rm = TRUE)
  Q3 <- quantile(data[[column]], 0.75, na.rm = TRUE)
  IQR <- Q3 - Q1

  lower_bound <- Q1 - 1.5 * IQR
  upper_bound <- Q3 + 1.5 * IQR

  data_clean <- data[data[[column]] >= lower_bound & data[[column]] <= upper_bound, ]
  return(data_clean)
}


meta.sem.outlier = remove_outliers_iqr(meta.sem.outlier, 'yi')
```

```
par(mfrow=c(1, 2))
boxplot(meta.sem.outlier$yi)
boxplot(meta.sem.outlier$vi)
```





Criando Modelo

```
# Criar modelo inicial
model.meta <- lm(expm1(yi) ~ ., data = meta)

# Stepwise selection usando AIC
model.step <- step(model.meta, direction = "both", trace = TRUE)
```

```
## Start:  AIC=-202.14
## expm1(yi) ~ vi + randomized + cluster_name + cluster_year
##
##                Df Sum of Sq     RSS     AIC
## - cluster_year  1    0.0175  7.2873 -203.90
## <none>                       7.2698 -202.15
## - randomized    1    0.2007  7.4704 -201.42
## - cluster_name 26    6.7490 14.0187 -188.48
## - vi            1    5.3741 12.6439 -148.80
##
## Step:  AIC=-203.9
## expm1(yi) ~ vi + randomized + cluster_name
##
##                Df Sum of Sq     RSS     AIC
## <none>                       7.2873 -203.90
## - randomized    1    0.1831  7.4704 -203.42
```

```
## + cluster_year  1    0.0175  7.2698 -202.15
## - cluster_name 26    7.2985 14.5858 -186.51
## - vi            1    5.3859 12.6731 -150.57
```

```r
# Ver resultado final do modelo reduzido
summary(model.step)
```

```
##
## Call:
## lm(formula = expm1(yi) ~ vi + randomized + cluster_name, data = meta)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58729 -0.13885  0.00000  0.08348  1.32726
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -0.19001    0.18514  -1.026 0.308233
## vi                             1.26967    0.17527   7.244 4.18e-10 ***
## randomized                    -0.27687    0.20728  -1.336 0.185904
## cluster_nameAmiot              0.87946    0.42405   2.074 0.041711 *
## cluster_nameAnderson           0.58861    0.33384   1.763 0.082179 .
## cluster_nameBertolaso          0.71713    0.32073   2.236 0.028502 *
## cluster_nameByrd-Bredbenner    0.87257    0.42405   2.058 0.043289 *
## cluster_nameCaldwell           0.90937    0.31258   2.909 0.004835 **
## cluster_nameCooney             0.17675    0.21487   0.823 0.413484
## cluster_nameCordts             1.62219    0.42406   3.825 0.000278 ***
## cluster_nameDoebel             0.83448    0.32073   2.602 0.011279 *
## cluster_nameEarle              1.33856    0.35852   3.734 0.000378 ***
## cluster_nameFeltz              0.05518    0.36994   0.149 0.881852
## cluster_nameFIAPO              1.20657    0.28830   4.185 8.05e-05 ***
## cluster_nameFlens              0.11109    0.36996   0.300 0.764850
## cluster_nameHennessy           0.28790    0.35851   0.803 0.424635
## cluster_nameKunst              0.79425    0.30710   2.586 0.011753 *
## cluster_nameLackner            0.06488    0.42409   0.153 0.878837
## cluster_nameMacDonald          0.53544    0.35855   1.493 0.139778
## cluster_nameNorris             0.49961    0.22309   2.239 0.028260 *
## cluster_nameNovotna            0.78478    0.35847   2.189 0.031867 *
## cluster_namePalomo-Velez       0.64258    0.33379   1.925 0.058220 .
## cluster_nameReese              0.62654    0.29097   2.153 0.034696 *
## cluster_nameRouk               0.87810    0.30707   2.860 0.005566 **
## cluster_nameSchnabelrauch      0.64787    0.30706   2.110 0.038390 *
## cluster_nameSchwitzgebel       0.32659    0.37000   0.883 0.380389
## cluster_nameSilva              0.63961    0.35849   1.784 0.078664 .
## cluster_nameSpanikova          0.03529    0.35885   0.098 0.921932
## cluster_nameTian               0.53436    0.32071   1.666 0.100079
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3204 on 71 degrees of freedom
## Multiple R-squared:  0.7324, Adjusted R-squared:  0.6269
## F-statistic: 6.941 on 28 and 71 DF,  p-value: 2.131e-11
```

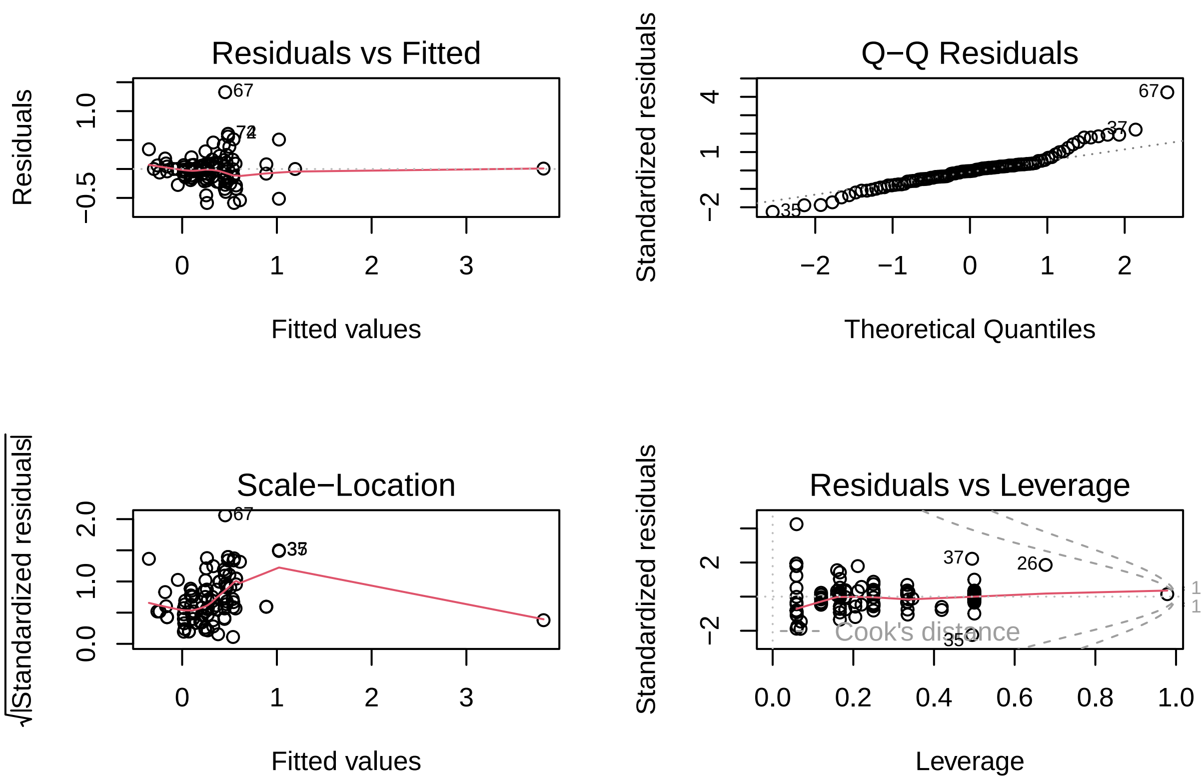
```r
summary(model.meta)
```

```
##
## Call:
## lm(formula = expm1(yi) ~ ., data = meta)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58751 -0.13942  0.00000  0.08258  1.32722
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   46.10729  112.68120   0.409 0.683654
## vi                             1.26846    0.17633   7.194 5.54e-10 ***
## randomized                    -0.30307    0.21804  -1.390 0.168936
## cluster_nameAmiot              1.02067    0.54779   1.863 0.066621 .
## cluster_nameAnderson           0.69909    0.43021   1.625 0.108656
## cluster_nameBertolaso          0.78929    0.36734   2.149 0.035121 *
## cluster_nameByrd-Bredbenner    0.82978    0.43908   1.890 0.062924 .
## cluster_nameCaldwell           1.01836    0.41138   2.475 0.015730 *
## cluster_nameCooney             0.20774    0.22892   0.907 0.367260
## cluster_nameCordts             1.67138    0.44304   3.773 0.000335 ***
## cluster_nameDoebel             0.90673    0.36743   2.468 0.016044 *
## cluster_nameEarle              1.50272    0.53823   2.792 0.006749 **
## cluster_nameFeltz              0.19316    0.50125   0.385 0.701144
## cluster_nameFIAPO              1.29940    0.36763   3.535 0.000729 ***
## cluster_nameFlens              0.22605    0.46560   0.486 0.628834
## cluster_nameHennessy           0.38306    0.42860   0.894 0.374520
## cluster_nameKunst              0.90475    0.40959   2.209 0.030454 *
## cluster_nameLackner            0.22913    0.58462   0.392 0.696304
## cluster_nameMacDonald          0.63059    0.42862   1.471 0.145712
## cluster_nameNorris             0.58086    0.29911   1.942 0.056170 .
## cluster_nameNovotna            0.94900    0.53830   1.763 0.082273 .
## cluster_namePalomo-Velez       0.78375    0.48040   1.631 0.107287
## cluster_nameReese              0.69901    0.34173   2.046 0.044564 *
## cluster_nameRouk               0.99628    0.42207   2.360 0.021045 *
## cluster_nameSchnabelrauch      0.74307    0.38613   1.924 0.058366 .
## cluster_nameSchwitzgebel       0.46453    0.50124   0.927 0.357232
## cluster_nameSilva              0.73479    0.42860   1.714 0.090880 .
## cluster_nameSpanikova          0.10761    0.40160   0.268 0.789515
## cluster_nameTian               0.62954    0.39716   1.585 0.117446
## cluster_year                  -0.02300    0.05598  -0.411 0.682423
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3223 on 70 degrees of freedom
## Multiple R-squared:  0.7331, Adjusted R-squared:  0.6225
## F-statistic: 6.629 on 29 and 70 DF,  p-value: 5.325e-11
```

```r
par(mfrow=c(2, 2))
plot(model.meta)
```

```
## Warning: not plotting observations with leverage one:
```

```
##   4, 12, 27, 34, 38, 47, 92

## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produzidos
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produzidos
```



Criando modelo sem outliers

```r
model.meta.sem.outlier = lm(yi ~. , data = meta.sem.outlier)

model.meta.sem.outlier
```

```
##
## Call:
## lm(formula = yi ~ ., data = meta.sem.outlier)
##
## Coefficients:
##                (Intercept)                          vi
##                  15.558718                    1.560326
##                 randomized           cluster_nameAmiot
##                  -0.246188                    0.818552
##        cluster_nameAnderson        cluster_nameBertolaso
##                   0.613433                    0.700317
## cluster_nameByrd-Bredbenner          cluster_nameCaldwell
```

```
##                  0.757623                    0.817319
##          cluster_nameCooney          cluster_nameCordts
##                  0.211427                    1.219737
##          cluster_nameDoebel           cluster_nameEarle
##                  0.739634                    1.134314
##           cluster_nameFeltz           cluster_nameFIAPO
##                  0.124467                    0.662594
##           cluster_nameFlens        cluster_nameHennessy
##                  0.179503                    0.297296
##           cluster_nameKunst          cluster_nameLackner
##                  0.767768                    0.038062
##       cluster_nameMacDonald          cluster_nameNorris
##                  0.561590                    0.500644
##         cluster_nameNovotna    cluster_namePalomo-Velez
##                  0.744951                    0.662469
##           cluster_nameReese            cluster_nameRouk
##                  0.368800                    0.811731
##   cluster_nameSchnabelrauch    cluster_nameSchwitzgebel
##                  0.628096                    0.398374
##           cluster_nameSilva       cluster_nameSpanikova
##                  0.642874                   -0.024692
##            cluster_nameTian                cluster_year
##                  0.543172                   -0.007842
```

```
summary(model.meta.sem.outlier)
```

```
##
## Call:
## lm(formula = yi ~ ., data = meta.sem.outlier)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39969 -0.09538  0.00000  0.07201  0.45846
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 15.558718  69.715454   0.223  0.82408
## vi                           1.560326   0.606038   2.575  0.01225 *
## randomized                  -0.246188   0.134353  -1.832  0.07134 .
## cluster_nameAmiot            0.818552   0.338378   2.419  0.01828 *
## cluster_nameAnderson         0.613433   0.265276   2.312  0.02383 *
## cluster_nameBertolaso        0.700317   0.226508   3.092  0.00290 **
## cluster_nameByrd-Bredbenner  0.757623   0.270591   2.800  0.00667 **
## cluster_nameCaldwell         0.817319   0.253602   3.223  0.00196 **
## cluster_nameCooney           0.211427   0.141499   1.494  0.13982
## cluster_nameCordts           1.219737   0.272994   4.468 3.12e-05 ***
## cluster_nameDoebel           0.739634   0.228269   3.240  0.00186 **
## cluster_nameEarle            1.134314   0.331607   3.421  0.00107 **
## cluster_nameFeltz            0.124467   0.309034   0.403  0.68841
## cluster_nameFIAPO            0.662594   0.267871   2.474  0.01592 *
## cluster_nameFlens            0.179503   0.286857   0.626  0.53360
## cluster_nameHennessy         0.297296   0.264142   1.126  0.26439
## cluster_nameKunst            0.767768   0.252356   3.042  0.00335 **
## cluster_nameLackner          0.038062   0.362682   0.105  0.91673
```

```
## cluster_nameMacDonald        0.561590   0.264455   2.124  0.03740 *
## cluster_nameNorris           0.500644   0.184630   2.712  0.00850 **
## cluster_nameNovotna          0.744951   0.332949   2.237  0.02859 *
## cluster_namePalomo-Velez     0.662469   0.295989   2.238  0.02854 *
## cluster_nameReese            0.368800   0.253533   1.455  0.15044
## cluster_nameRouk             0.811731   0.260088   3.121  0.00266 **
## cluster_nameSchnabelrauch    0.628096   0.238278   2.636  0.01042 *
## cluster_nameSchwitzgebel     0.398374   0.308843   1.290  0.20152
## cluster_nameSilva            0.642874   0.264060   2.435  0.01758 *
## cluster_nameSpanikova       -0.024692   0.256190  -0.096  0.92351
## cluster_nameTian             0.543172   0.244688   2.220  0.02982 *
## cluster_year                -0.007842   0.034631  -0.226  0.82155
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1985 on 67 degrees of freedom
## Multiple R-squared:  0.6052, Adjusted R-squared:  0.4342
## F-statistic: 3.541 on 29 and 67 DF,  p-value: 1.012e-05
```

```r
par(mfrow=c(2, 2))
plot(model.meta.sem.outlier)
```

```
## Warning: not plotting observations with leverage one:
##   4, 12, 27, 34, 35, 36, 45, 89
```