

ANOVA - Aplicação Real de Índice de Qualidade Percebida de Telecomunicações

Estudo de Caso da ANATEL

Esse arquivo apresenta uma aplicação real de um projeto realizado na ANATEL.

Análise de Variância do IQP das Operadoras Brasileiras

Importando as Bibliotecas Necessárias

```
suppressMessages(library(car))
suppressMessages(library(agricolae))
suppressMessages(library(dplyr))
suppressMessages(library(ggplot2))
```

Fazemos em primeiro um passo a criação de funções que irão nos ajudar a manipular os dados como queremos

Funções

```
impute_na <- function(data, columns){
  # data: data.frame, columns: character

  demais_cols <- base::setdiff(names(data), columns)
  data0 <- data[demais_cols]
  data_na <- data[columns]

  replace_na <- function(x){
    x[is.na(x)] <- as.integer(mean(x, na.rm = TRUE))
    return(x)
  }

  data_na <- apply(
    X = data_na, MARGIN = 2, FUN = replace_na
  )

  data <- cbind(data0, data_na)

  return(data)
}

wilcox_teste <- function(f., groups, values, data, verbose = FALSE){
  # f.: formula, groups: character(1), values: character(2)
  # results: htest

  index <- (data[[groups]] %in% values)
  data_sub <- data[index,]
```

```

test <- wilcox.test(f., data = data_sub, exact = FALSE)

if (verbose) {
  cat("=====\n")
  msg <- paste0(
    values, collapse = ' VS '
  )
  cat("Testando diferenças para: ", msg, '\n')
  print(test)
  cat("-----\n")
  return(invisible(test))
}

return(test)
}

plot_media_indice <- function(data, servico){
  # data: data.frame, servico: character(1)

  title. <- paste0('Média do índice para ', servico, ' - CI 95%')
  #par(las = 1)
  gplots::plotmeans(
    indice ~ operadora, xlab = "Operadora", ylab = "Índice",
    main = title., data = data
  )
}

```

Depois importarmos os dados dos IQPS de cada serviço de telecomunicações.

Carregando os dados de IQP

```

dados <- list(
  pospre = read.csv2("indices/indice-af-pospre.csv", stringsAsFactors = TRUE),
  bl = read.csv2("indices/indice-af-bl.csv", stringsAsFactors = TRUE),
  tv = read.csv2("indices/indice-af-tv.csv", stringsAsFactors = TRUE),
  tf = read.csv2("indices/indice-af-tf.csv", stringsAsFactors = TRUE)
)

cols_replace_na <- base::setdiff(names(dados$bl), c("response_id", "operadora"))

dados <- lapply(
  X = dados, FUN = impute_na, columns = cols_replace_na
)

```

Teste de Homogeneidade da variância do Erro

Como vimos na aula Teórica existe uma necessidade de se estimar um modelo homecedástico, ou seja , que os erros possuem variância comum.

```

levene_res <- lapply(
  X = dados, FUN = function(data){with(data, leveneTest(indice, operadora))}
)

```

```

bartlett_res <- lapply(
  X = dados, FUN = function(data){bartlett.test(indice ~ operadora, data = data)}
)

levene_resumo <- data.frame(
  servico = names(levene_res),
  estatistica_F = sapply(levene_res, function(x){x[['F value']][1]}),
  p_valor = sapply(levene_res, function(x){x[['Pr(>F)']][1]}),
  row.names = NULL
)

bartlett_resumo <- data.frame(
  servico = names(bartlett_res),
  K_squared = sapply(bartlett_res, function(x){x[['statistic']][1]}),
  p_valor = sapply(bartlett_res, function(x){x[['p.value']][1]}),
  row.names = NULL
)

print(list(levene = levene_resumo, bartlett = bartlett_resumo))

```

```

## $levene
##   servico estatistica_F      p_valor
## 1  pospre      0.7884381 0.5004946543
## 2     bl      6.2871894 0.0003622947
## 3     tv      0.5566365 0.6943282299
## 4     tf      0.5083372 0.6768924794
##
## $bartlett
##   servico K_squared      p_valor
## 1  pospre  2.561017 0.46436478
## 2     bl   9.804342 0.02030465
## 3     tv   2.188735 0.70109211
## 4     tf   2.140881 0.54368665

```

```

pos_pre = data.frame(dados["pospre"])
head(pos_pre)

```

```

##   pospre.response_id pospre.operadora pospre.J1 pospre.J2 pospre.J3 pospre.J4
## 1                401             CLARO         7         6         6         7
## 2                406             CLARO        10         9         9         9
## 3                415             CLARO         5         5         5         4
## 4                421             CLARO         7        10        10        10
## 5                427             CLARO         5         8         5         5
## 6                433             CLARO         8         9         8         9
##   pospre.J5 pospre.indice
## 1         6      6.412921
## 2         9      9.202964
## 3         4      4.595887
## 4        10      9.391108
## 5         5      5.516528
## 6         9      8.576289

```

```

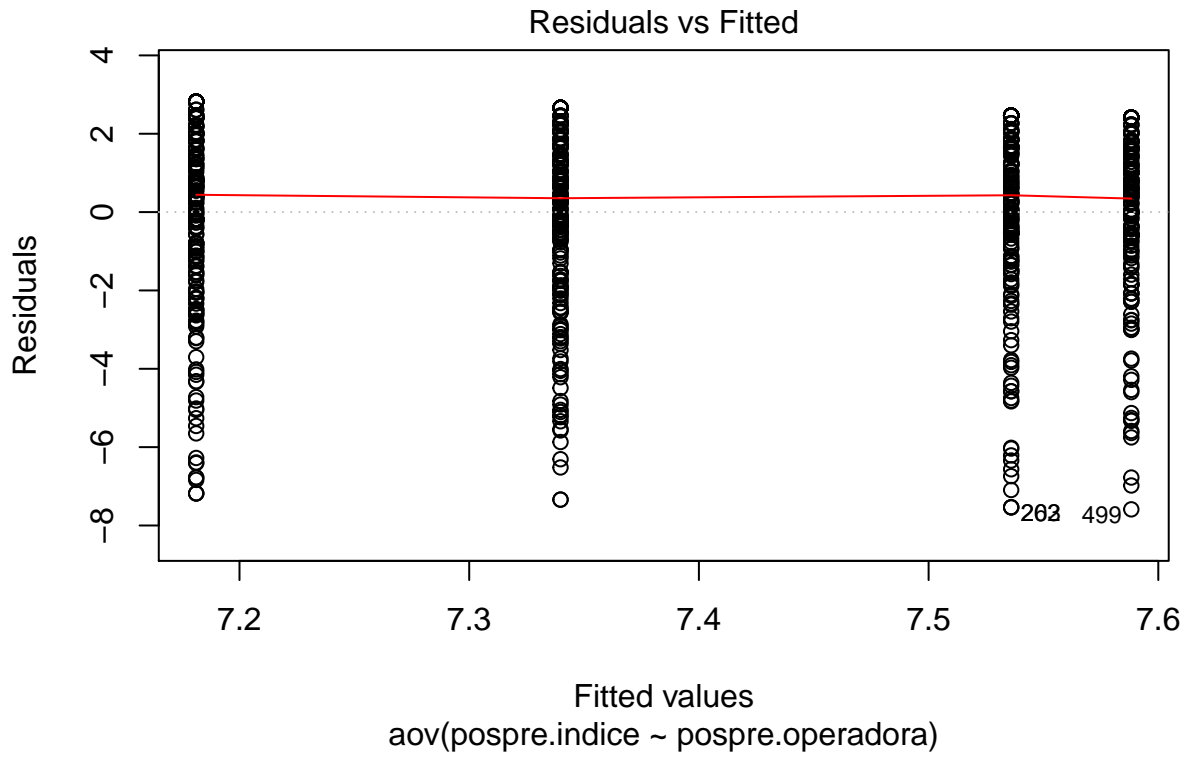
anova_pos_pre <- aov(pospre.indice ~ pospre.operadora, data = pos_pre)
resumo_pos_pre <- summary(anova_pos_pre);resumo_pos_pre

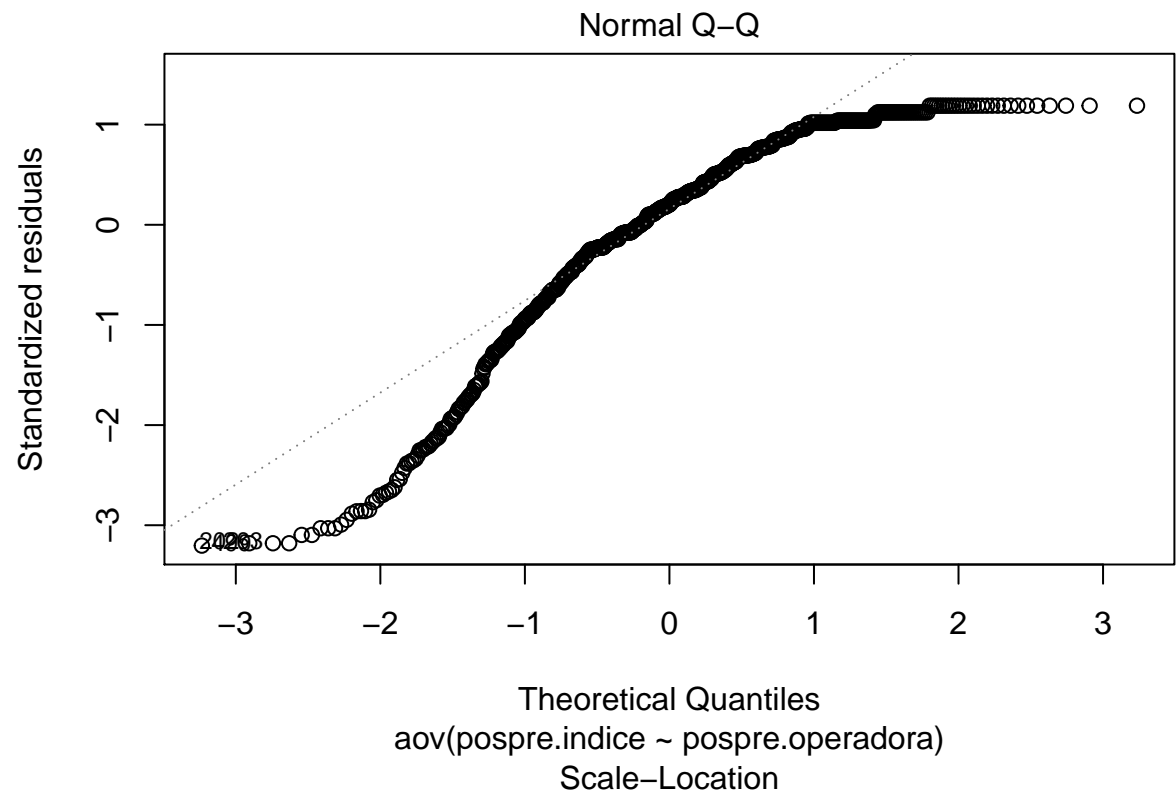
```

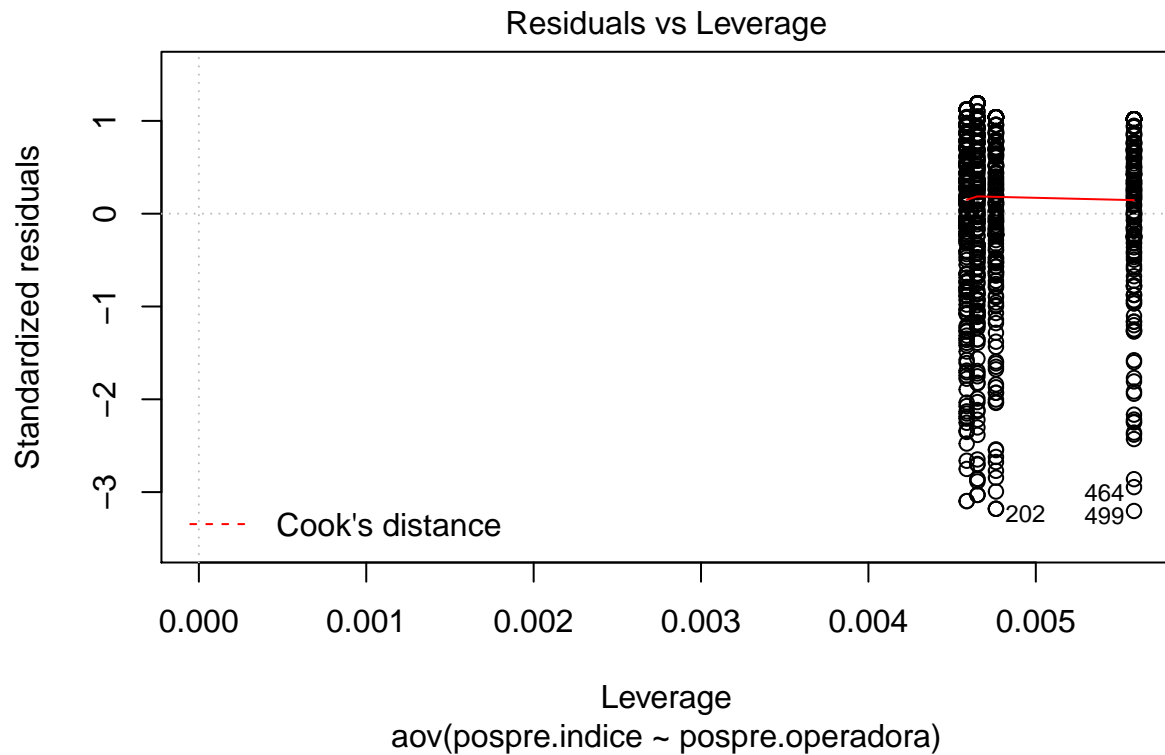
##		Df	Sum Sq	Mean Sq	F value	Pr(>F)
##	pospre.operadora	3	21	7.103	1.259	0.287
##	Residuals	818	4616	5.643		

Gráfico de Avaliação de Normalidade dos Resíduos

```
plot(anova_pos_pre)
```







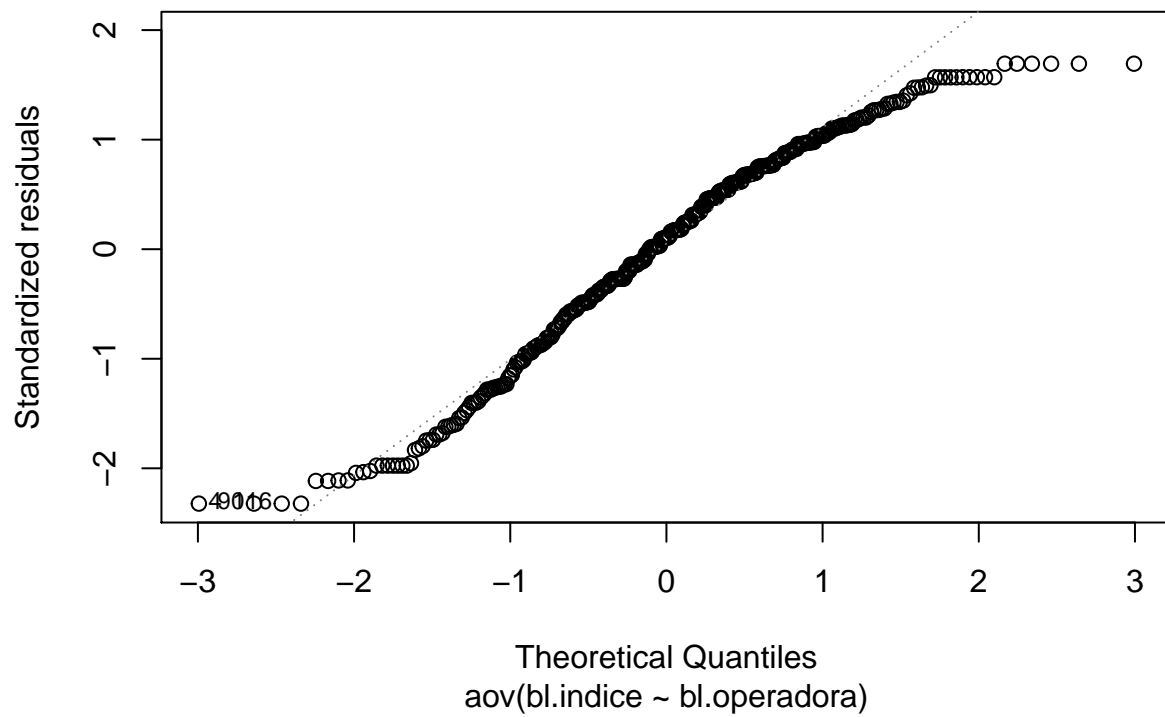
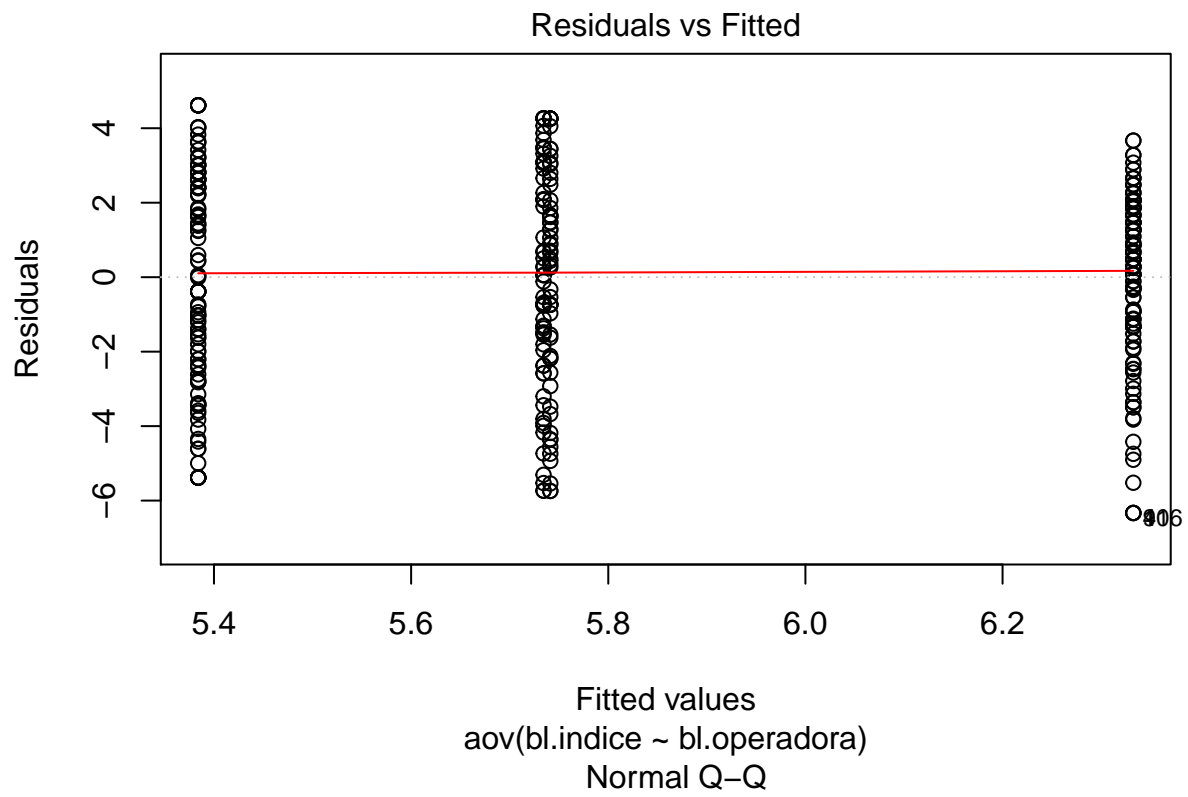
```
bl = data.frame(dados["bl"])
head(bl)
```

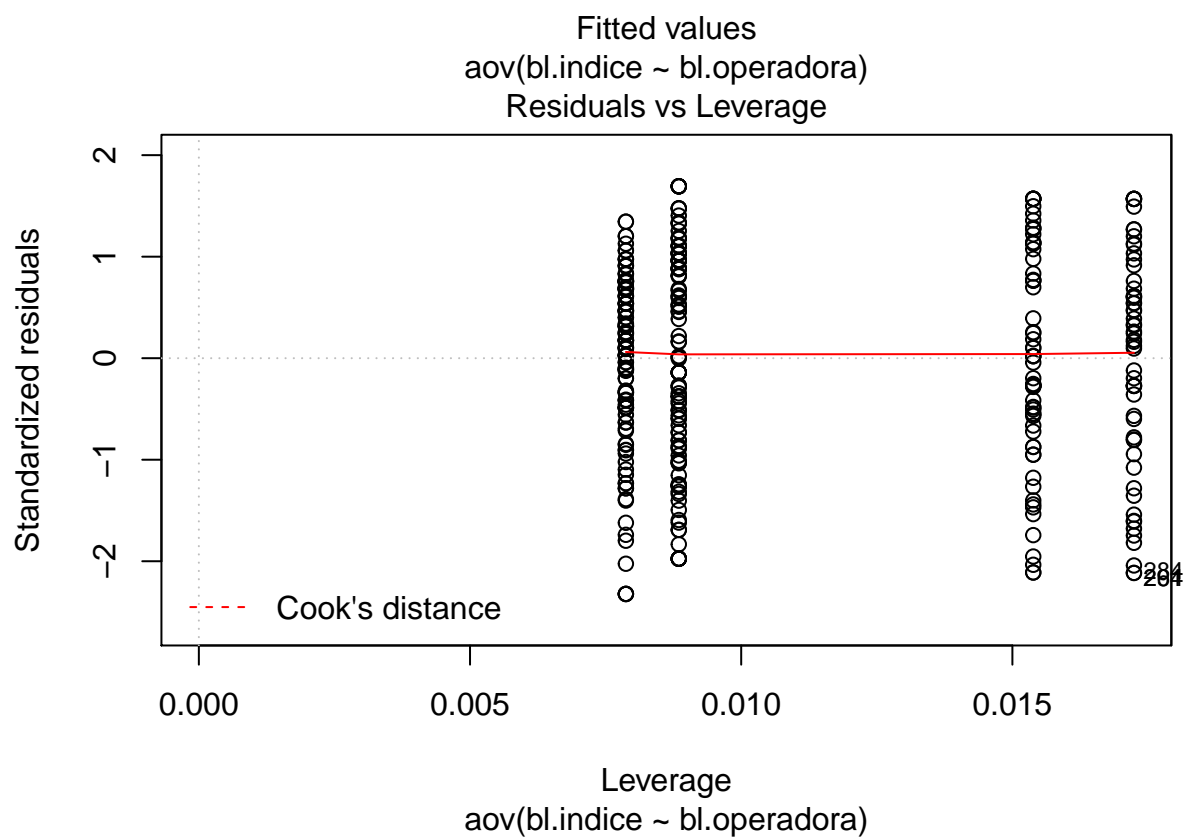
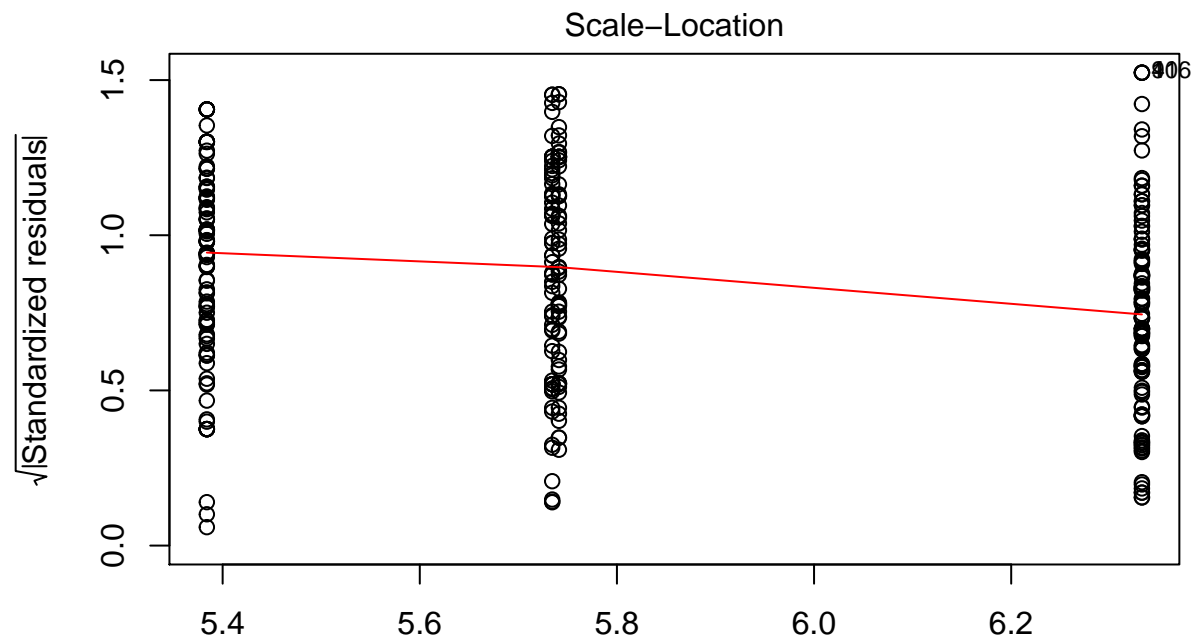
```
##   bl.response_id bl.operadora bl.J1 bl.J2 bl.J3 bl.J4 bl.J5 bl.indice
## 1             93   CLARO/NET    8    8    8    8    7  7.805963
## 2             95   CLARO/NET    4    3    4    4    5  4.006854
## 3             96   CLARO/NET    7    6    7    7    7  6.812817
## 4            102   CLARO/NET    0    0    0    0    0  0.000000
## 5            105   CLARO/NET    6    6   10    0    5  5.425101
## 6            106   CLARO/NET    5    5    6    6    5  5.419391
```

```
anova_bl <- aov(bl.indice ~ bl.operadora, data = bl)
resumo_bl <- summary(anova_bl);resumo_bl
```

```
##           Df Sum Sq Mean Sq F value Pr(>F)
## bl.operadora  3   55.6  18.536   2.473 0.0615 .
## Residuals  359 2691.1   7.496
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(anova_bl)
```





Teste ANOVA

```
anovas <- lapply(
  X = dados, FUN = function(data){aov(indice ~ operadora, data = data)}
)
```



```

anovas <- anovas[base::setdiff(names(anovas), 'bl')]
anovas_summary <- lapply(
  X = anovas, FUN = summary
)

# Teste robusto
reg_mean <- lm(indice ~ operadora, data = dados$bl)
anova_bl <- Anova(reg_mean, white.adjust = TRUE)

## Coefficient covariances computed by hccm()
extrair_estat <- function(summary_aov, estat){
  # summary_aov: summary.aov, estat: character(1)
  summary_aov[[1]][[estat]][1]
}

anova_resumo <- data.frame(
  servico = c(names(anovas_summary), 'bl'),
  estat_F = c(
    sapply(anovas_summary, extrair_estat, estat = 'F value'),
    anova_bl[1, 'F']
  ),
  p_valor = c(
    sapply(anovas_summary, extrair_estat, estat = 'Pr(>F)'),
    anova_bl[1, 'Pr(>F)']
  ),
  variancia_homo = c('SIM', 'SIM', 'SIM', 'NÃO'),
  row.names = NULL
)

print(list(anova_resumo))

```

```

## [[1]]
##   servico  estat_F    p_valor variancia_homo
## 1  pospre 1.2588355 0.28738126             SIM
## 2     tv 1.9194499 0.10713925             SIM
## 3     tf 0.5457894 0.65143508             SIM
## 4     bl 2.7265489 0.04400772             NÃO

```

Teste Tukey

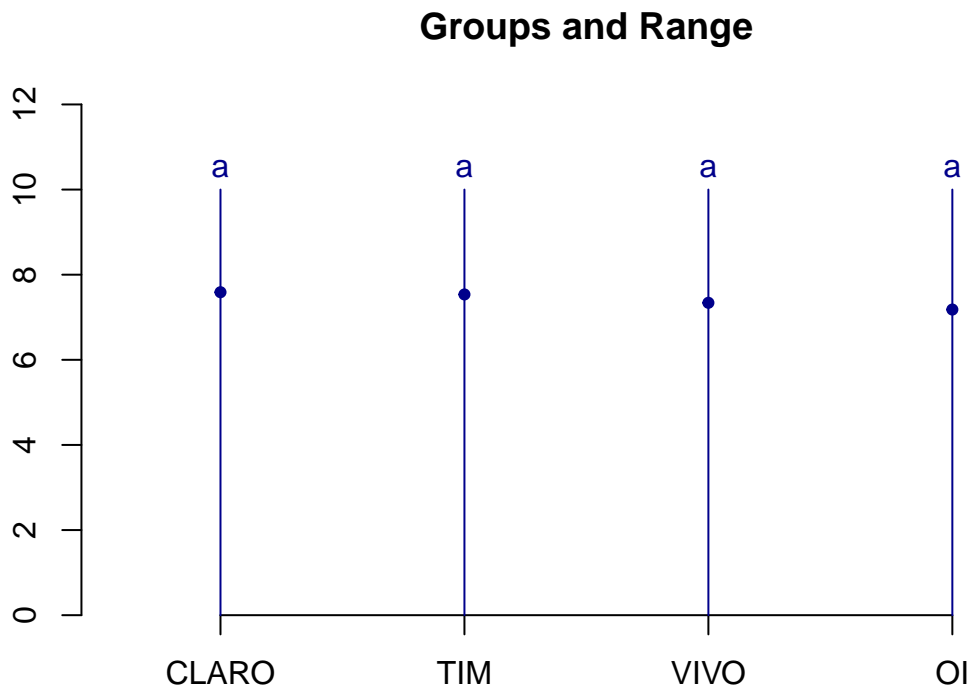
```

# POS e PRE
tuk_pospre <- HSD.test(
  anovas$pospre, "operadora", group = T, alpha = 0.05, console = T
)

##
## Study: anovas$pospre ~ "operadora"
##
## HSD Test for indice
##
## Mean Square Error: 5.642676

```

```
##
## operadora, means
##
##      indice      std   r Min Max
## CLARO 7.588255 2.239637 179    0 10
## OI     7.181139 2.509922 215    0 10
## TIM    7.535949 2.358149 210    0 10
## VIVO   7.339752 2.363808 218    0 10
##
## Alpha: 0.05 ; DF Error: 818
## Critical Value of Studentized Range: 3.640638
##
## Groups according to probability of means differences and alpha level( 0.05 )
##
## Treatments with the same letter are not significantly different.
##
##      indice groups
## CLARO 7.588255    a
## TIM    7.535949    a
## VIVO   7.339752    a
## OI     7.181139    a
plot(tuk_pospre)
```



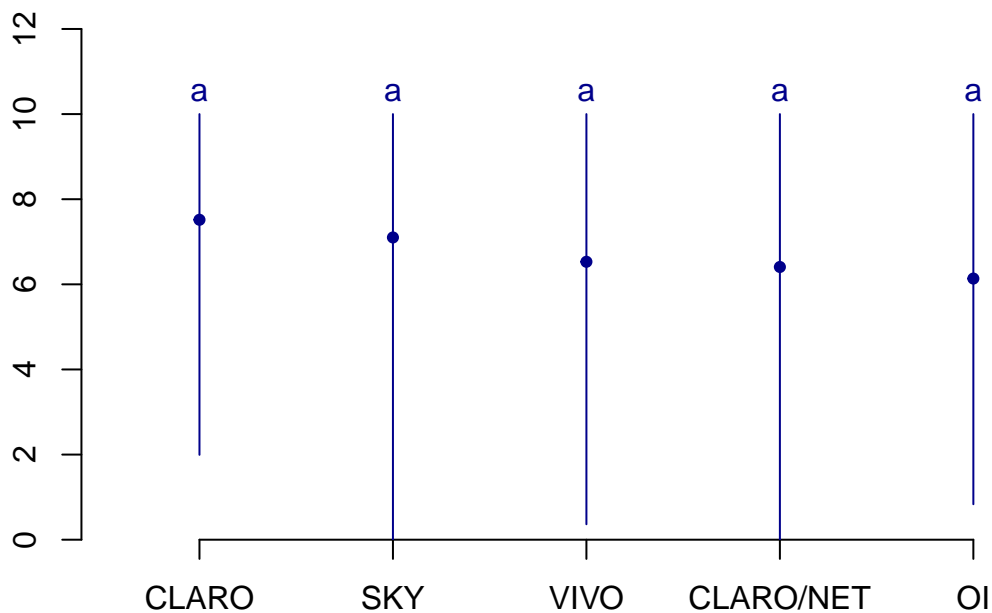
```
# TV
tuk_tv <- HSD.test(
  anovas$tv, "operadora", group = T, alpha = 0.05, console = T
)
```

```
##
## Study: anovas$tv ~ "operadora"
##
## HSD Test for indice
```

```
##
## Mean Square Error: 6.510203
##
## operadora, means
##
##      indice      std    r      Min Max
## CLARO      7.517112 2.156280 28 1.9935552 10
## CLARO/NET 6.407584 2.545175 144 0.0000000 10
## OI         6.135902 2.531356 13 0.8333357 10
## SKY        7.101252 2.567359 87 0.0000000 10
## VIVO       6.529988 2.864186 30 0.3629215 10
##
## Alpha: 0.05 ; DF Error: 297
## Critical Value of Studentized Range: 3.881465
##
## Groups according to probability of means differences and alpha level( 0.05 )
##
## Treatments with the same letter are not significantly different.
##
##      indice groups
## CLARO      7.517112      a
## SKY        7.101252      a
## VIVO       6.529988      a
## CLARO/NET 6.407584      a
## OI         6.135902      a
```

```
plot(tuk_tv)
```

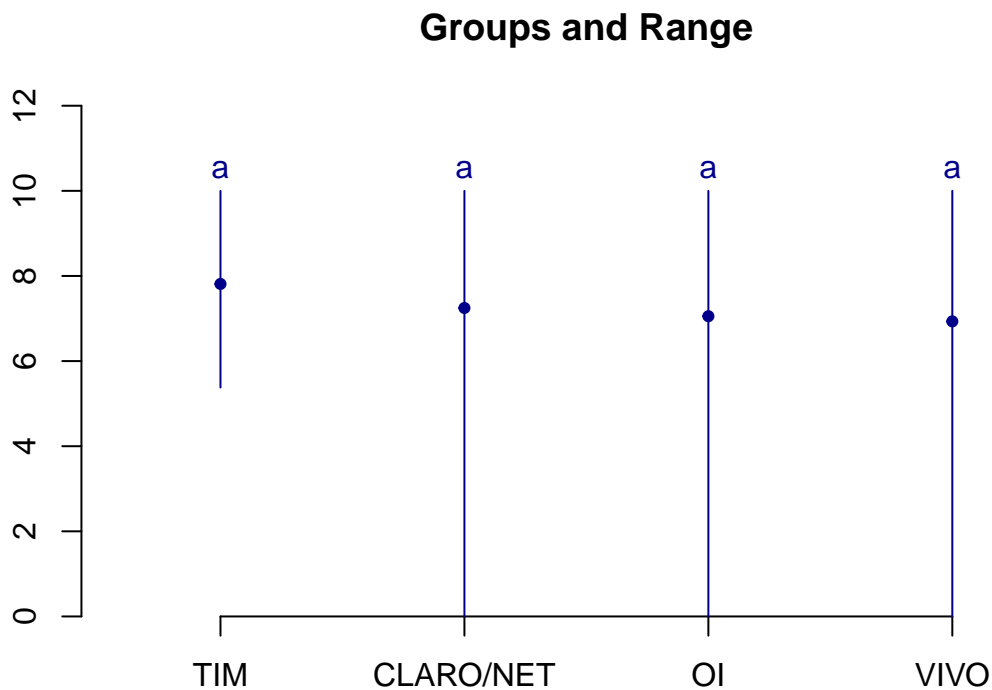
Groups and Range



```
# TF
tuk_tf <- HSD.test(
  anovas$tf, "operadora", group = T, alpha = 0.05, console = T
)
```

```
##
## Study: anovas$tf ~ "operadora"
##
## HSD Test for indice
##
## Mean Square Error: 5.62389
##
## operadora, means
##
##      indice      std  r      Min Max
## CLARO/NET 7.247677 2.410059 84 0.000000 10
## OI        7.054390 2.454974 86 0.000000 10
## TIM       7.812645 1.701315 11 5.378155 10
## VIVO      6.933335 2.294477 63 0.000000 10
##
## Alpha: 0.05 ; DF Error: 240
## Critical Value of Studentized Range: 3.658742
##
## Groups according to probability of means differences and alpha level( 0.05 )
##
## Treatments with the same letter are not significantly different.
##
##      indice groups
## TIM       7.812645      a
## CLARO/NET 7.247677      a
## OI        7.054390      a
## VIVO      6.933335      a
```

```
plot(tuk_tf)
```



Teste Não Paramétrico de Mann-Whitney

```
data_bl <- dados$bl
data_bl <- data_bl %>%
  mutate(operadora = as.character(operadora))

resumo_bl <- data_bl %>%
  group_by(operadora) %>%
  summarise(
    N = n(),
    iqp_medio = mean(indice, na.rm = TRUE),
    std = sd(indice, na.rm = TRUE),
    .groups = 'drop'
  ) %>%
  arrange(desc(iqp_medio)) %>%
  as.data.frame()
print(resumo_bl)

##   operadora   N iqp_medio      std
## 1 CLARO/NET 127  6.332651 2.292885
## 2      SKY   58  5.741163 2.945023
## 3     VIVO   65  5.734353 2.911336
## 4      OI  113  5.383967 2.975917

# Gerando pares de operadoras para teste
operadoras <- unique(data_bl$operadora)
operadoras <- combn(
  x = operadoras, m = 2, simplify = FALSE
)
names(operadoras) <- sapply(
  operadoras, function(x){paste(x, collapse = '-')})
)

# Testando pares de operadoras
testes_wilcox <- lapply(
  X = operadoras, FUN = wilcox_teste, f. = indice ~ operadora,
  groups = 'operadora', data = data_bl
)

resumo_wilcox <- data.frame(
  comparacao = names(testes_wilcox),
  statistic_W = sapply(testes_wilcox, `[`, i = 'statistic'),
  p_valor = sapply(testes_wilcox, `[`, i = 'p.value'),
  row.names = NULL
)

print(list(resumo_wilcox))

## [[1]]
##      comparacao statistic_W      p_valor
## 1  CLARO/NET-OI      8402.0 0.02237521
## 2  CLARO/NET-SKY      4071.5 0.25081532
## 3 CLARO/NET-VIVO      4613.0 0.18313091
## 4      OI-SKY       3069.5 0.49930586
```

```
## 5      OI-VIVO      3390.0 0.39407453
## 6      SKY-VIVO      1903.5 0.92730525
```

```
# Média do índice por operadora
```

```
medias_bl <- data_bl %>%
  group_by(operadora) %>%
  summarise(indice_medio = mean(indice), .groups = 'drop')
medias_nulo <- medias_bl %>%
  mutate(indice_medio = 0)
medias <- rbind(medias_bl, medias_nulo)

p1 <- medias_bl %>%
  ggplot(aes(x = operadora, y = indice_medio)) +
  geom_point(color = '#808000', size = 2.5) +
  geom_line(data = medias, aes(x = operadora, y = indice_medio), color = '#808000') +
  theme_light() +
  labs(
    x = 'Operadora', y = 'Índice',
    title = 'Índice médio por operadora - BL'
  ) +
  theme(
    plot.title = element_text(hjust = .5)
  )
```

```
print(p1)
```

