

Processo Seletivo: 4intelligence

Luiz Paulo Tavares Gonçalves

25/01/2022

Bibliotecas utilizadas

```
rm(list = ls()) # Limpando a memória

# Packages
library(renv)
init()
library(readxl) # Importação dos Datasets
library(tidyverse) # Manipulação
library(DataExplorer) # Análise Exploratória
library(Amelia) # Tratamento de NA's
library(lubridate) # Datetime
library(tidyr) # Pivot e higienização do dataset
library(cowplot) # Junção de plotes
library(TSstudio) # Plotes - Time Series
library(forecast) # Time Series
library(knitr) # Tables
library(rmarkdown) # Rmarkdown
library(cluster) # Clusterização
library(factoextra) # Dendrograma - cluster
library(GGally) # Matriz de Correlação
library(urca) # Teste de Raiz unitária
snapshot()
```

```
## * Lockfile written to '~/Área de Trabalho/Ciência de Dados: Machine Learning & Deep Learning/Processos
```

PROVA TEÓRICA

Questão 4

Seguindo os pressupostos de modelagem, em primeiro, inicia-se fazendo uma análise exploratória dos dados. Assim, verificando o comportamento dos dados. No presente caso, em modelagem com séries temporais, buscaria verificar a presença de sazonalidade, autocorrelação, estacionariedade etc. Em segundo, buscaria aplicar testes formais de estacionariedade e cointegração, pois, no caso de séries temporais, a presença de cointegração ou de raiz unitária pode influenciar na escolha do modelo, por exemplo, entre um modelo VAR ou VEC. Em terceiro, caso necessário, passaria para o pré-processamento dos dados, isto é, a diferenciação das séries. Em quarto, com os resultados dos testes e da análise exploratória, passaria finalmente para

a modelagem. No caso univariado, considerando apenas a venda de computadores, poderia selecionar até mesmo modelos de machine learning para séries temporais como, por exemplo, séries temporais com redes neurais multicamadas ou redes neurais com base radial. No caso de modelos multivariados, poderia-se modelar, como no presente caso de venda de computadores (oferta), um modelo com equações simultâneas (até mesmo utilizando variáveis instrumentais): considerando n variáveis explicativas como manda a lei de demanda e oferta (preço de venda, renda dos consumidores, preço de um bem substituto etc).

PROVA DE PROGRAMAÇÃO - Parte A

1. Importe o arquivo "pmc.xlsx" pulando as três primeiras linhas

```
data.pmc <- read_excel("pmc.xlsx", skip = 3)
```

2. Renomeie as colunas e as deixe na seguinte ordem: "data", "var", "setor" e colunas de localidades. Para essas últimas, utilize siglas (BR e siglas dos estados).

```
data.pmc <- data.pmc %>% rename("data" = "...1", "var" = "...2",
                                "setor" = "...3", "BR" = "Brasil",
                                "CE" = "Ceará", "PE" = "Pernambuco",
                                "BA" = "Bahia", "MG" = "Minas Gerais",
                                "ES" = "Espírito Santo", "RJ" = "Rio de Janeiro",
                                "SP" = "São Paulo", "PA" = "Paraná",
                                "SC" = "Santa Catarina", "RS" = "Rio Grande do Sul",
                                "GO" = "Goiás", "DF" = "Distrito Federal")
```

3. Converta os valores das colunas de Brasil e UFs para que fiquem na classe numérica. Em seguida como passo 4. Substitua os textos na coluna "var". Faça com que assumam apenas as palavras "volume" ou "receita", a depender de cada linha.

```
data.pmc <- data.pmc %>% mutate(BR = as.numeric(BR), CE = as.numeric(CE),
                                PE = as.numeric(PE), BA = as.numeric(BA),
                                MG = as.numeric(MG), ES = as.numeric(ES),
                                RJ = as.numeric(RJ), SP = as.numeric(SP),
                                PA = as.numeric(PA), SC = as.numeric(SC),
                                RS = as.numeric(RS), GO = as.numeric(GO),
                                DF = as.numeric(DF))
```

```
data.pmc <- data.pmc[1:4208, ] # Elimando a ultima linha, a linha da fonte
variavel <- rep(c("Volume", "Receita"), 263, each = 8) # repetir a cada 8 linhas volume, receita
data.base <- data.frame(variavel, data.pmc) # Unir em data.frame

# Finalizado a organização de var

data.base <- data.base %>%
  select(!(var)) %>%
  rename("var" = "variavel")
```

5. Converta a coluna “data” para que ela assuma a classe própria para data. O padrão deve ser YYYY-MM-DD. (Utilize 01 quando não houver especificado o dia).

```
# Criar um vetor de datas

data <- seq.Date(from = as_date("2000-01-01"),
                 to = as_date("2021-11-01"),
                 by = "month")

data = rep(c(data), each = 16)

data.base <- data.frame(data, data.base)

data.base <- data.base %>%
  select(!(data.1))
```

6. Para as colunas “data” e “var”, preencha os espaços contendo “NA” com a informação que não seja “NA” localizada imediatamente acima.

Isso já foi feito em conjunto com os passos anteriores!

7. Exclua todas as linhas em que a coluna “var” seja igual à “receita”.

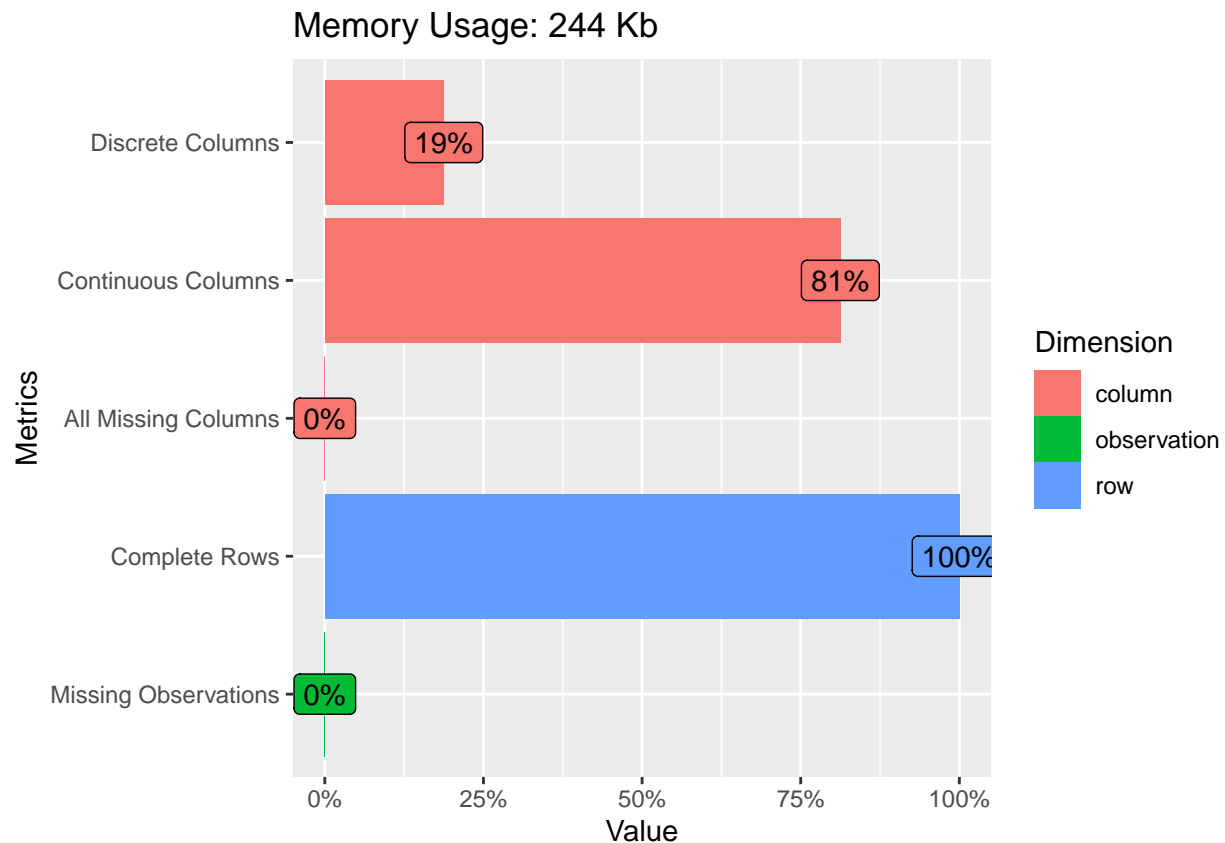
```
data.base <- data.base %>%
  filter(var %in% c("Volume"))

data.base <- na.omit(data.base)

# Visualizando o dataset limpo
```

Como pode ser visualizado, o dataset finalmente está limpo:

```
plot_intro(data.base)
```



Parte B

1. Utilize visualizações e estatísticas para comparar o setor de Móveis e Eletrodomésticos com os outros setores. Como a sede da empresa é em São Paulo, verifique o comportamento neste estado. (Dê maior destaque para o período da pandemia).

```
# Organização do dataset

data.base[, 3] <- str_sub(data.base$setor, start = 1, end = 7) # Abreviar os rótulos

data.sp <- data.base %>%
  filter(data >= "2020-01-01") %>%
  select(data, var, setor, SP)
```

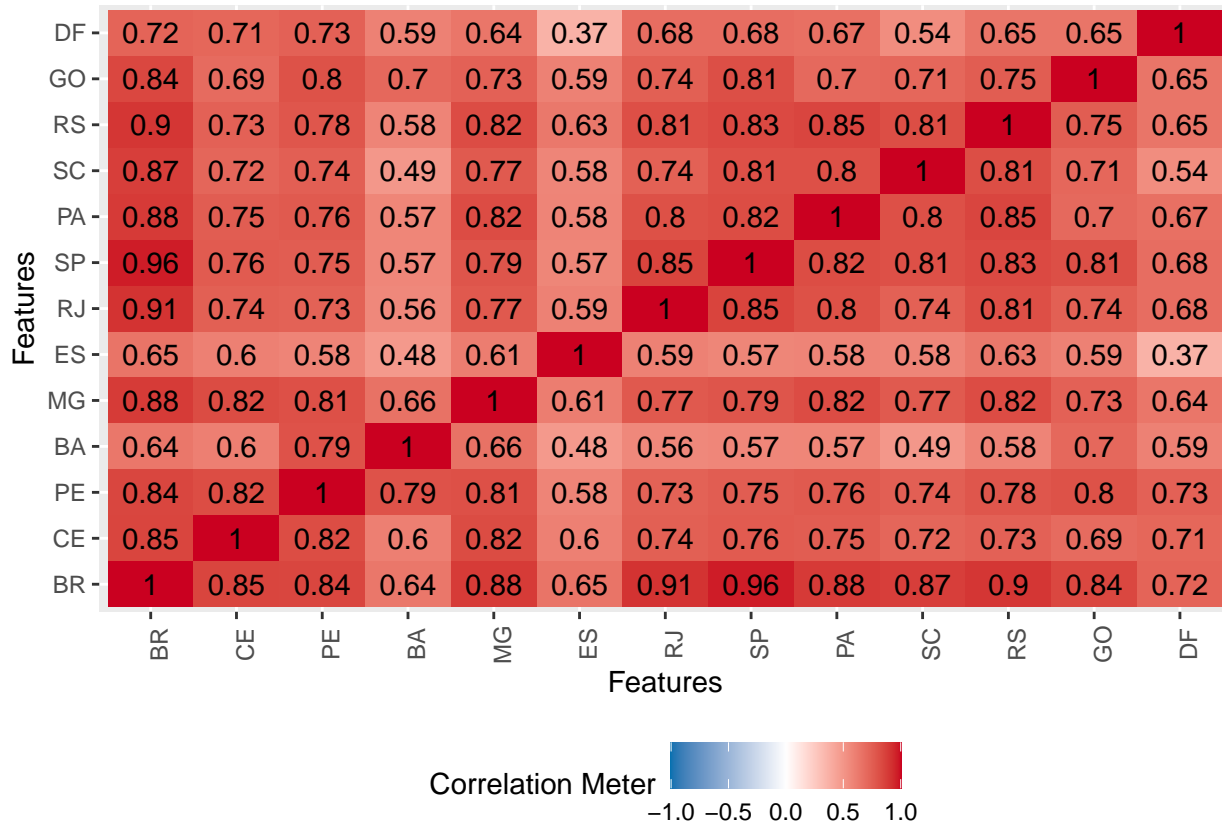
Análise Exploratória

Após filtrado apenas os dados referente ao Estado de São Paulo, para o período de Janeiro de 2020 a Novembro de 2021, inicia-se a análise exploratória buscando mapear e comparar os setores produtivos. Assim, inicia-se plotando uma correlação de Pearson dos setores entre os Estados e com o Brasil ao longo do período. A correlação de Pearson pode ser dada como segue, no qual \bar{x} e \bar{y} representam a média de x e y como de praxe:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$$

Como pode ser observado, em todos os Estados registrou-se um relação de associação linear entre os índices positiva, isto é, uma correlação positiva. Oscilando, em grande parte, entre correlações moderadas e fortes. Não obstante, como a correlação de Pearson é vulnerável a outliers e em casos da rejeição da hipótese de normalidade, vamos observar de forma mais minuciosa tais relações. A seguir pode-se visualizar boxplots para os setores:

```
plot_correlation(data.base[, 4:16])
```



Nota-se, nos boxplots, que os Artigos farmacêuticos, médicos, ortopédicos, de perfumaria e cosméticos representaram a maior mediana dos índices de volume de vendas. Seguido por Outros artigos de uso pessoal e doméstico, Hipermercados, supermercados, produtos alimentícios, bebidas e fumo. E finalmente, Móveis e eletrodomésticos. Observe que todos os setores, com exceção de Combustíveis e lubrificantes, retornaram a presença de outliers (pontos discrepantes). Vamos visualizar a estatística descritiva dos setores mais de perto a seguir.

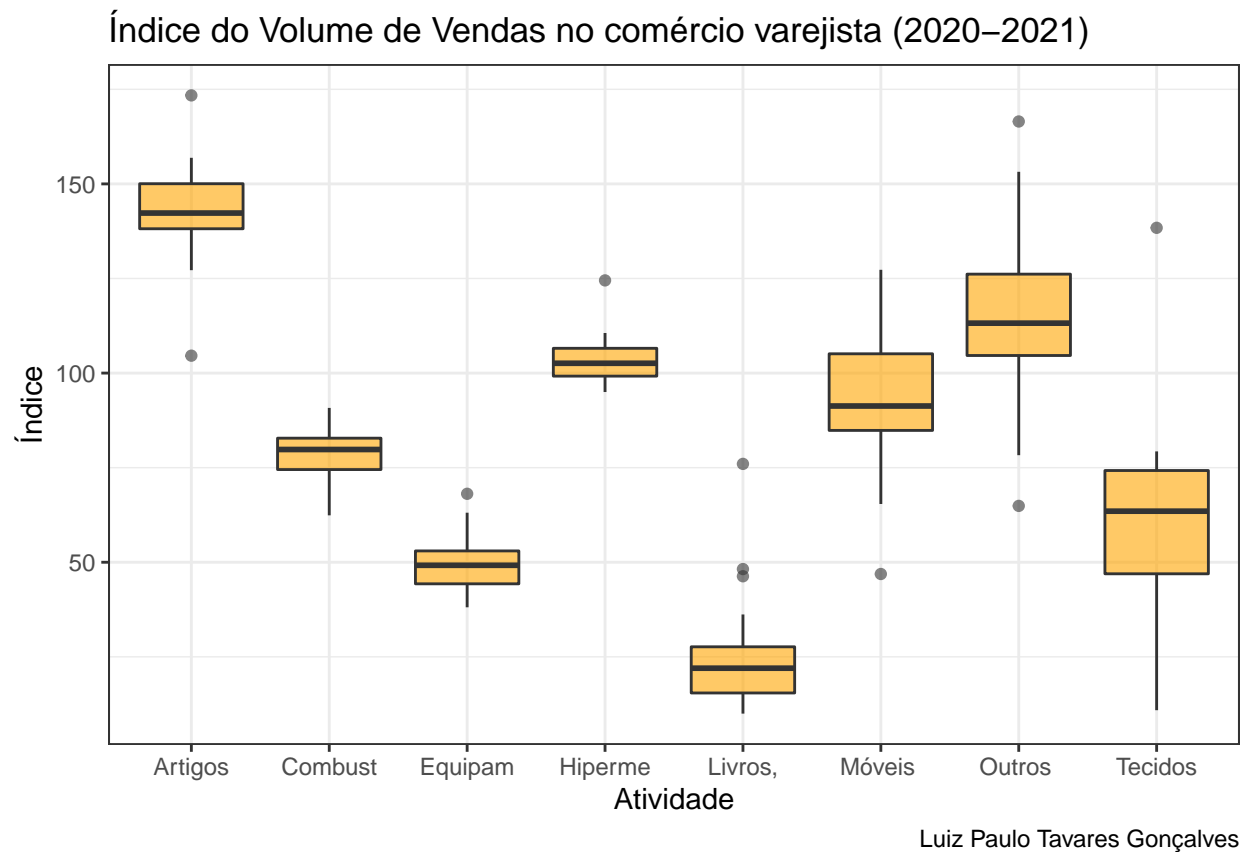
```
# Iniciando a análise exploratória
# Boxplot

ggplot2::ggplot(data.sp, aes(
  y = SP, x = setor)) +
  geom_boxplot(pch = 19, fill = "orange", alpha = 0.6) +
  labs(y = "Índice", x = "Atividade",
```

```

title = "Índice do Volume de Vendas no comércio varejista (2020-2021)",
caption = "Luiz Paulo Tavares Gonçalves")+
theme_bw()

```



Visualizando as estatísticas descritivas

```

data.table <- data.sp %>% group_by(setor) %>%
  summarise(Média = mean(SP),
            Mediana = median(SP),
            Desvio = sd(SP),
            CV = (mean(SP)/sd(SP)))

kable(data.table, caption = "Estatística Descritiva dos Setores (2020-2021)")

```

Table 1: Estatística Descritiva dos Setores (2020-2021)

setor	Média	Mediana	Desvio	CV
Artigos	143.61304	142.3	13.235637	10.850482
Combust	78.07391	79.8	7.426196	10.513312
Equipam	50.08696	49.2	7.090666	7.063787
Hiperme	103.47826	102.6	6.363815	16.260413
Livros,	25.11739	22.0	15.193948	1.653118
Móveis	92.31739	91.3	18.790052	4.913099

setor	Média	Mediana	Desvio	CV
Outros	115.20000	113.2	23.454579	4.911621
Tecidos	60.88261	63.5	25.866856	2.353692

2. Compare visualmente o efeito da pandemia neste setor entre os estados. Observe diferenças na intensidade e no tempo para recuperação.

Para uma comparação entre os Estados com relação ao setor de Móveis e eletrodomésticos, optou-se por filtrar um recorte temporal de janeiro de 2019 a novembro de 2021. Assim, obtendo uma amostra mais robusta e uma base de comparação pré-pandemia. Como pode ser visualizado nos plots a seguir, o mês de abril de 2020 representou uma queda abrupta no volume de vendas no setor de móveis e eletrodomésticos - um visível outlier. Com uma tendência de recuperação em “V” que vai de junho até encontrar uma leve queda em setembro e outubro e, posteriormente, com um aumento nos dois últimos meses do ano.

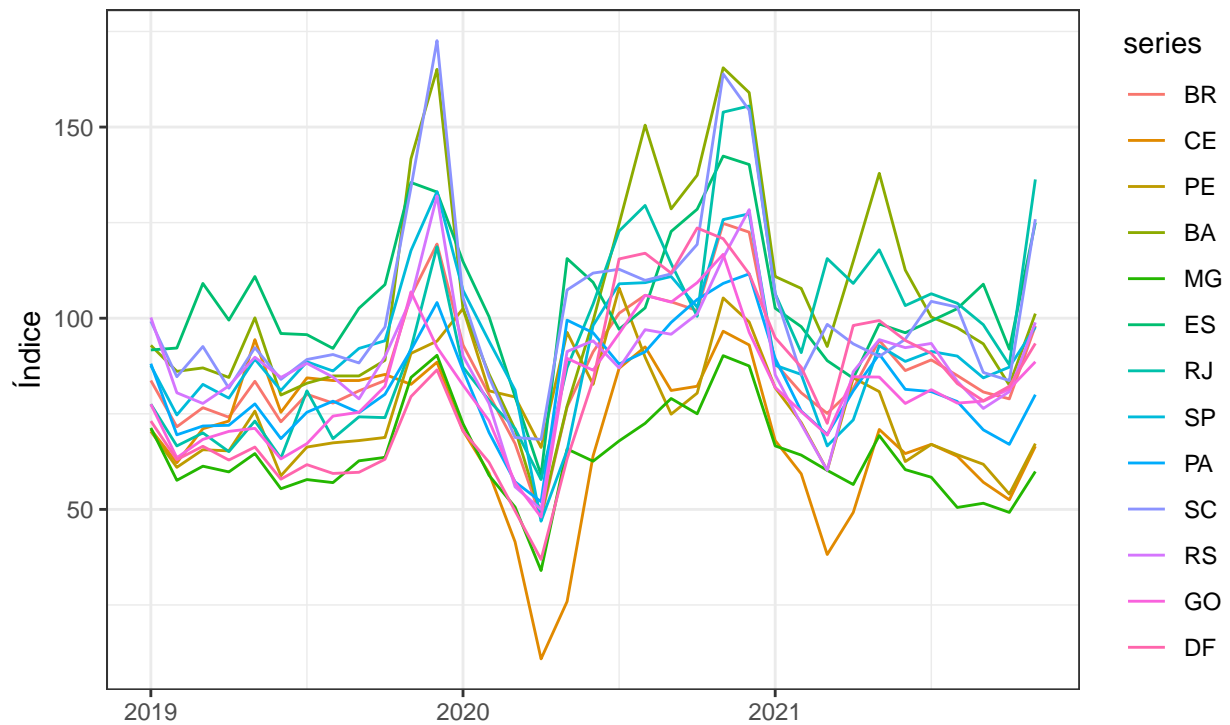
```
data.moveis <- data.base %>%
  filter(setor %in% c("Móveis ") & data >= "2019-01-01")

# Passando para série temporal
moveis.time <- ts(data.moveis[, 4:16], start = c(2019, 1),
                  end = c(2021, 11), frequency = 12)

# Visualizando as séries temporais

autoplot(moveis.time)+
  labs(title = "Índice do Volume de Vendas no comércio varejista (2020-2021)",
       y = "Índice", x = "", caption = "Luiz Paulo Tavares Gonçalves")+
  theme_bw()
```

Índice do Volume de Vendas no comércio varejista (2020–2021)



Luiz Paulo Tavares Gonçalves

```
# Polar mapeando pontos sazonais
```

```
ts_polar(moveis.time, title = "Polar - Mapeando possíveis pontos sazonais")
```

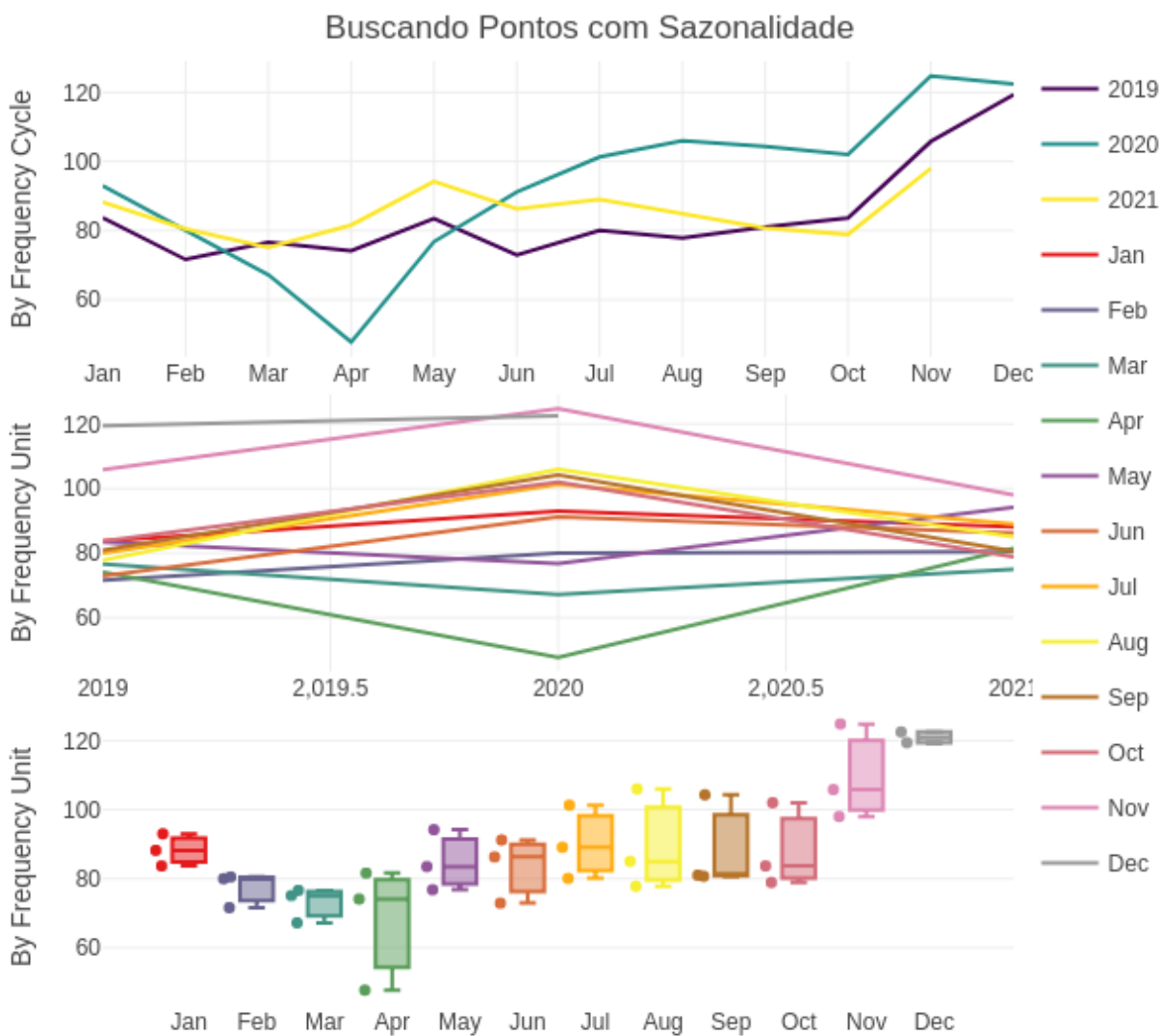
```
# Sazonalidade
```

```
ts_seasonal(moveis.time, type = 'all', palette_normal = "viridis",
             Ygrid = T, Xgrid = T, title = "Buscando Pontos com Sazonalidade") # Sazonalidade
```

```
ts_seasonal(moveis.time, type = "box",
             title = "Boxplot para cada mês: Índice") # Boxplot
```

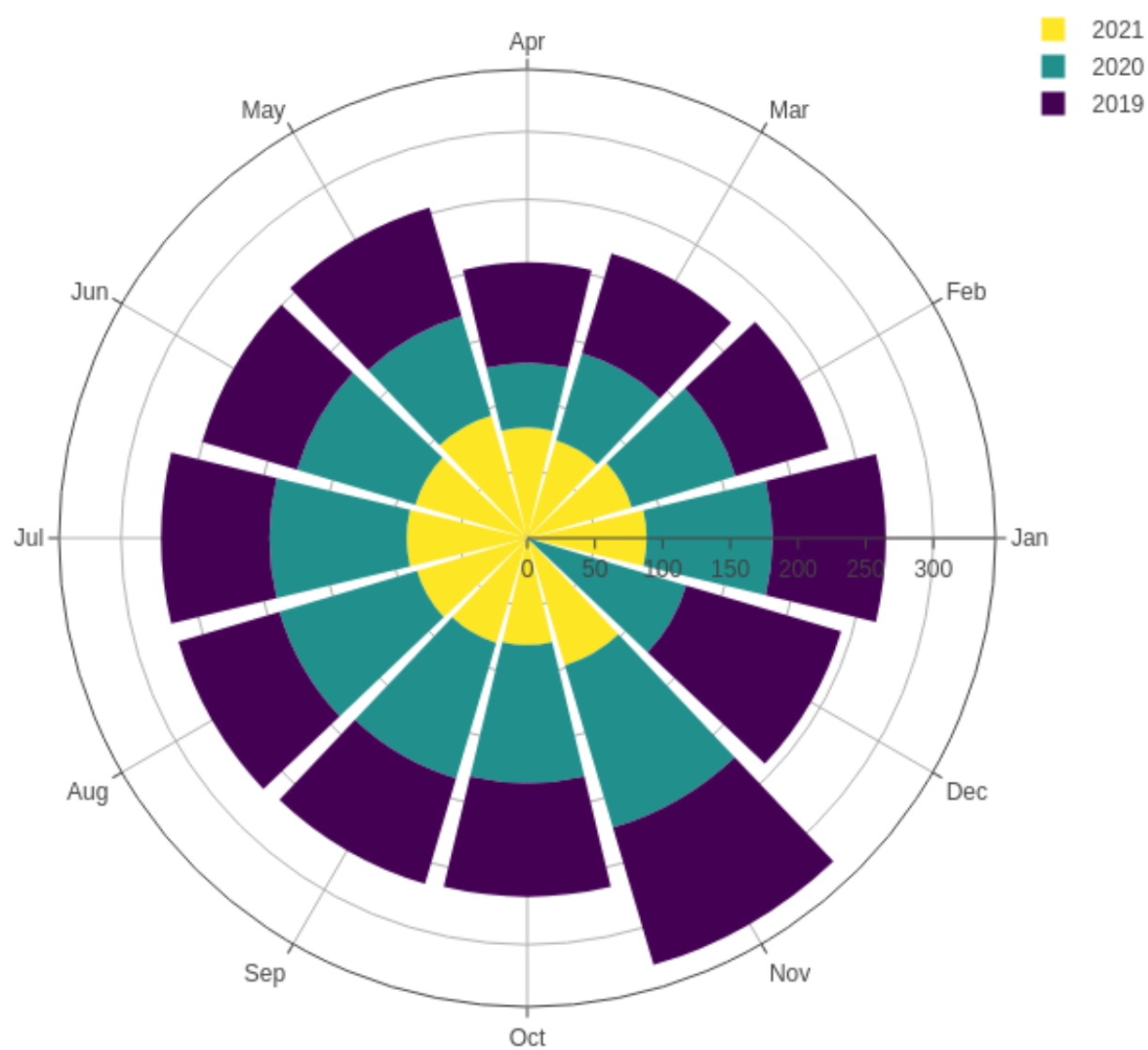
```
ts_heatmap(moveis.time, padding = T,
            title = "Heatmap: mapeamento os pontos sazonais") # Sazonalidade
```

```
ts_surface(moveis.time) # Plote 3D
```

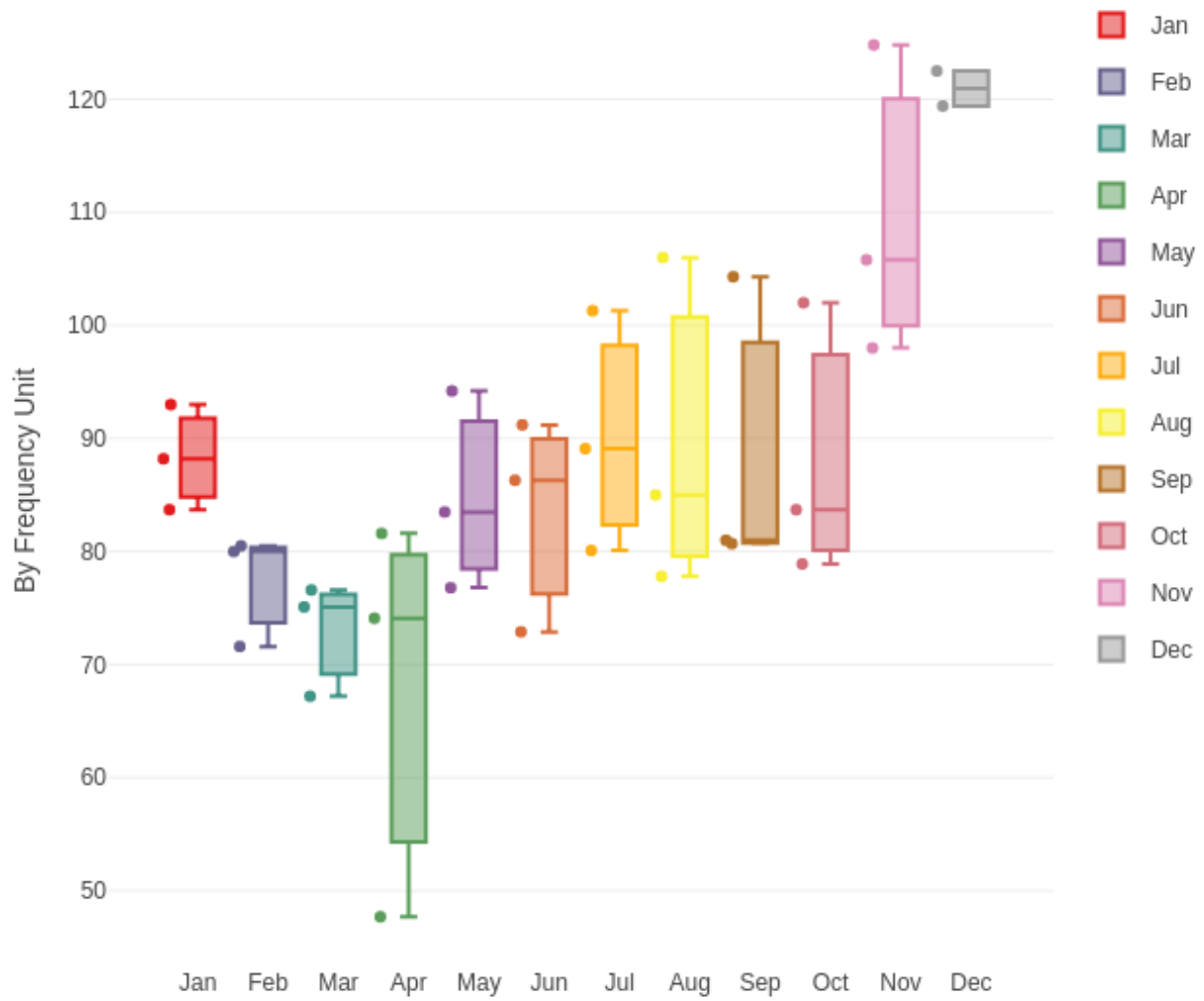



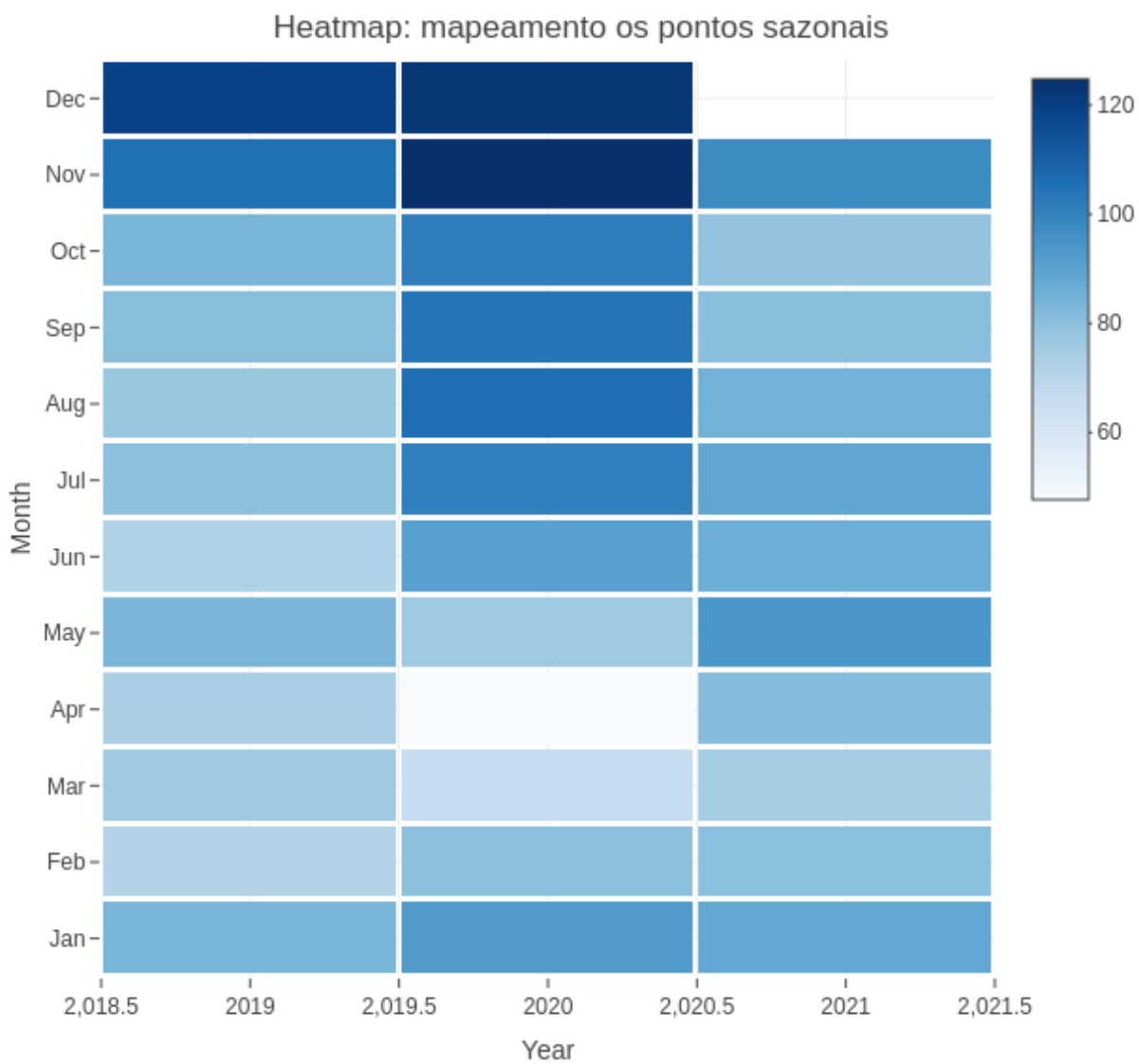
Note que nos plots a seguir, pode-se explorar pontos de sazonalidade dos três anos selecionados. Retornando possíveis pontos sazonais no final do ano (como pode ser observado no plote heatmap). Em 2021, por sua vez, não ocorreu queda tão abrupta como em 2020 - apenas repetindo o padrão de aumento do índice do volume de vendas no final do ano.

Polar - Mapeando possíveis pontos sazonais

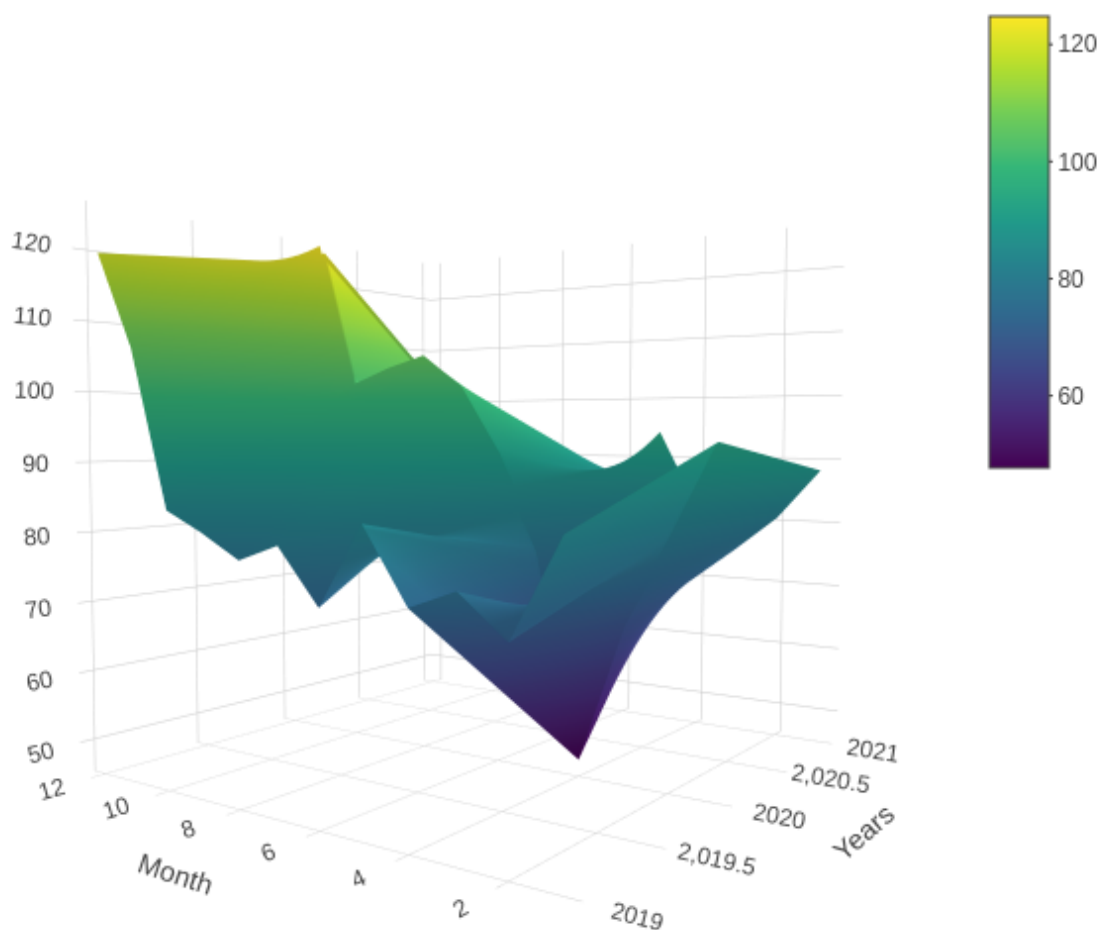


Boxplot para cada mês: Índice





Surface Plot - moveis.time



Por fim, buscando mapear a diferença entre os Estados, pode-se classificar os Estados em grupos de acordo com o índice de volume de vendas para os anos selecionados. Bem, para isso aplicou-se um algoritmo não supervisionado de Machine Learning bem simples: o Dendrograma. A ideia é básica, calcula-se a distância (similiaridade) entre os Estados formando grupos (clusters). Na metodologia adotada, pegou a mediana para cada Estado (dado os anos de 2019 a 2021) e, posteriormente, aplicou o dendrograma classificando os Estados em grupos. Assim, padroniza os dados e depois aplica a distância Euclidiana:

$$Z = \frac{x - \mu}{\sigma}$$

$$D_E(p, q) = \sqrt{(p_1 - q_1)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

```
data.cluster <- data.moveis %>%
  pivot_longer(cols = BR:DF,
               names_to = "Estados")
```

```
data.cluster <- data.cluster %>% group_by(Estados) %>%
  summarise(indice = median(value))

dendrograma <- data.frame(scale(data.cluster[,2]))
row.names(dendrograma) <- data.cluster$Estados

# Agrupando - Euclidean & Ward.D2

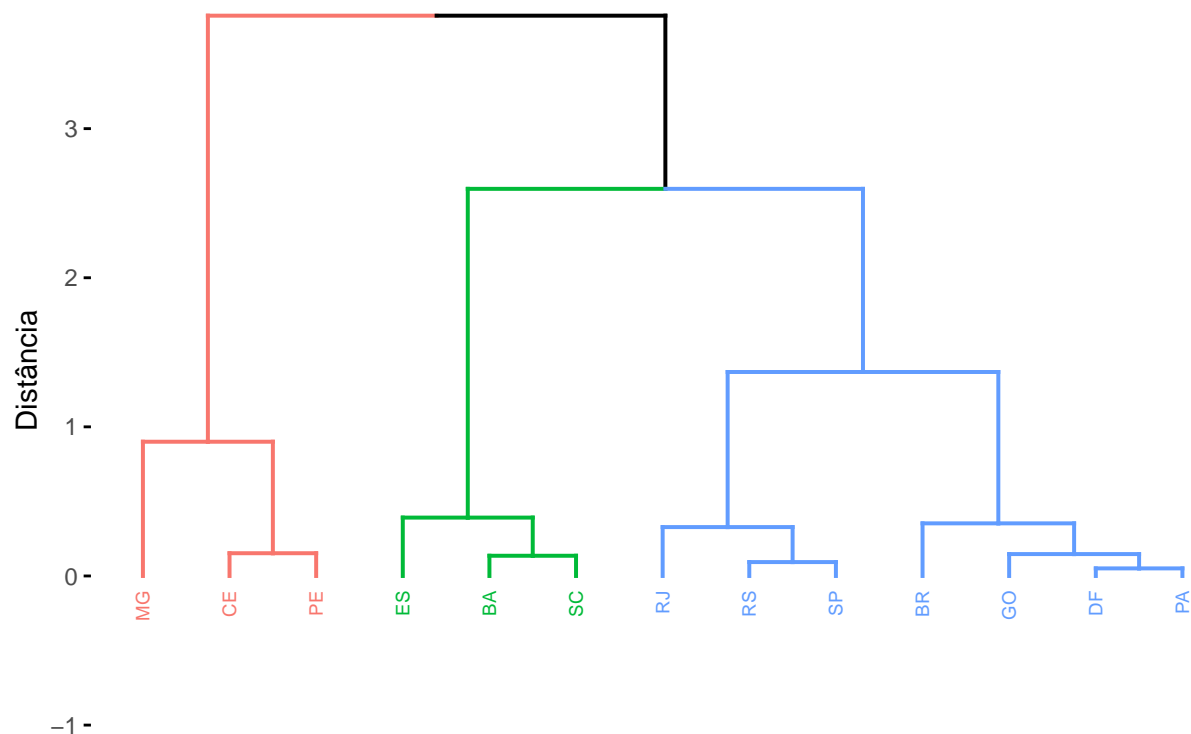
distancia<- dist(dendrograma, method = "euclidean")
clusterizado<- hclust(distancia, method = "ward.D2")
```

Em seguida pode-se observar os agrupamentos. Nota-se que o grupo 3 (cor vermelha) retorna o grupo com os Estados com o menor índice; por sua vez, o grupo 1 (cor verde) retorna o grupo com os Estados com o maior índice. E o grupo 2 (cor azul), pode-se considerar os Estados com o índice moderado.

```
# Dendrograma para visualizar os agrupamentos

fviz_dend(clusterizado, k=3, labels_track_height = 1, ylab = "Distância",
  main = "Agrupamento por semelhança (distância) entre a mediana dos Estados",
  cex = 0.5)
```

Agrupamento por semelhança (distância) entre a mediana dos Estados



```
data.cluster$agrupamentos <- cutree(clusterizado, k=3)
grupo.1 <- filter(data.cluster, agrupamentos == 1)
grupo.2 <- filter(data.cluster, agrupamentos == 2)
grupo.3 <- filter(data.cluster, agrupamentos == 3)
```

```
media.1 <- mean(grupo.1$indice)
media.2 <- mean(grupo.2$indice)
media.3 <- mean(grupo.3$indice)

Medias <- data.frame(media.1, media.2, media.3)
```

Concluindo, pode-se avaliar que o grupo 1 com BA, ES e SC representam os Estados que obtiveram os maiores índices de volume de venda e, por outro lado, o grupo 3 com CE, MG e PE representam os Estados que obtiveram os menores índices de volume de venda. E o grupo 2 (BR, DF, GO, PA, RJ, RS, SP) representam os Estados moderados no volume de vendas. A média do índice dos grupos pode ser visualizado a seguir:

```
kable(Medias, caption = "Média dos agrupamentos",
      col.names = c("Grupo 1", "Grupo 2", "Grupo 3"))
```

Table 2: Média dos agrupamentos

Grupo 1	Grupo 2	Grupo 3
99.83333	84.88571	68.73333

3. Calcule a variação percentual na fase inicial da pandemia, entre abril de 2019 e abril de 2020, para todos os estados, dentro do setor de interesse. Mostre graficamente as diferenças.

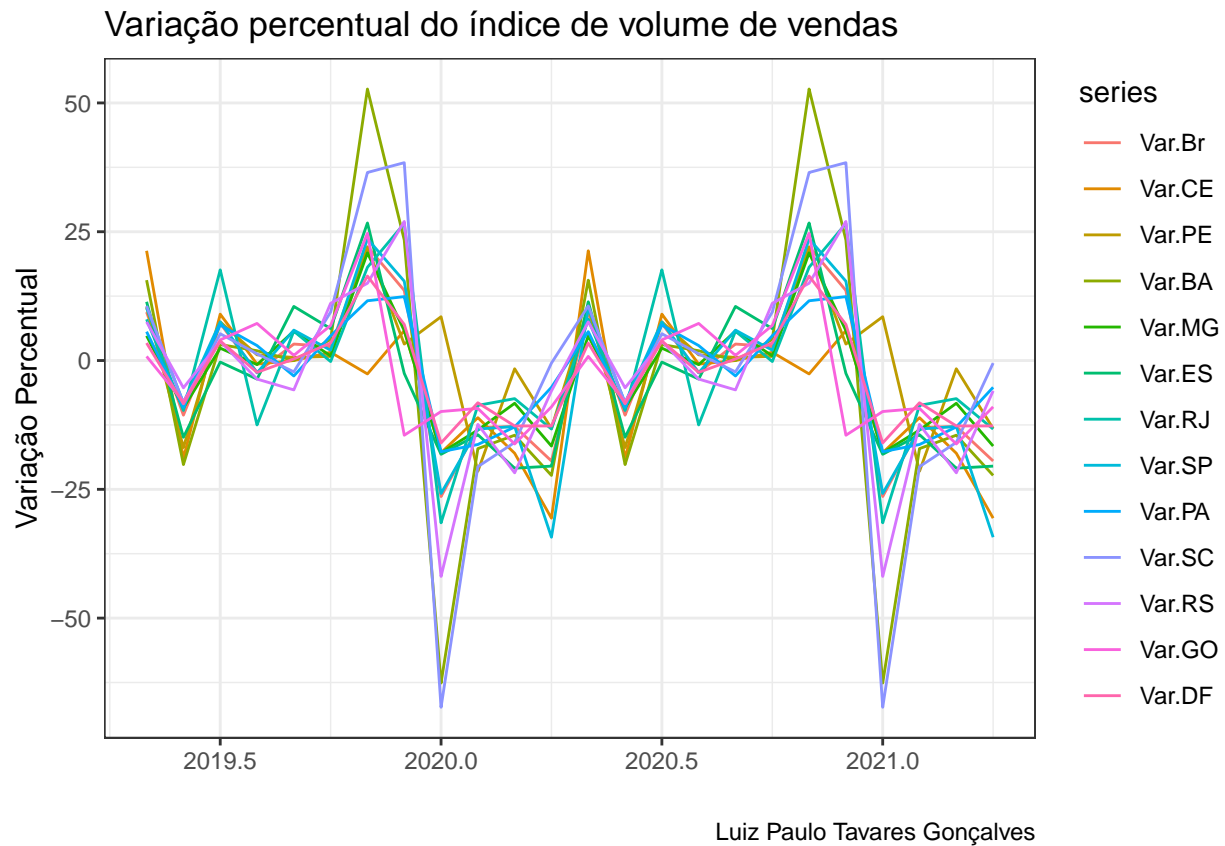
```
var.data <- data.base %>%
  filter(setor %in% c("Móveis ") & data >= "2019-04-01") %>%
  filter(data <= "2020-04-01") %>%
  mutate(Var.Br = BR - lag(BR),
         Var.CE = CE - lag(CE),
         Var.PE = PE - lag(PE),
         Var.BA = BA - lag(BA),
         Var.MG = MG - lag(MG),
         Var.ES = ES - lag(ES),
         Var.RJ = RJ - lag(RJ),
         Var.SP = SP - lag(SP),
         Var.PA = PA - lag(PA),
         Var.SC = SC - lag(SC),
         Var.RS = RS - lag(RS),
         Var.GO = GO - lag(GO),
         Var.DF = DF - lag(DF))

# Organizando

var.data <- data.frame(var.data[c(1:3, 17:29)])
var.data <- data.frame(var.data[2:13, ])

var.time <- ts(var.data[, 4:16], start = c(2019, 5),
              end = c(2021, 4), frequency = 12)
```

```
autoplot(var.time)+
  labs(title = "Variação percentual do índice de volume de vendas",
       y = "Variação Percentual", x = "", caption = "Luiz Paulo Tavares Gonçalves")+
  theme_bw()
```



PARTE C

1. Importe a base “rendimento_efetivo_real.csv”.

```
data.rendimento <- read_delim("rendimento_efetivo_real.csv", delim = ";")
data.rendimento$...3 = NULL
```

2. Renomeie as colunas para que tenhamos “data” e “renda”.

```
data.rendimento <- data.rendimento %>%
  rename("data" = "Data",
        "renda" = "Rendimento real médio de todos os trabalhos efetivos mensais - R$ Penúltimo mês da s")
```


3. Ajuste a coluna data para o formato YYYY-MM-DD. (Utilize 01 quando não houver especificado o dia)

```
data <- seq.Date(from = as_date("2012-03-01"),
                to = as_date("2021-10-01"),
                by = "month")

data.rendimento <- data.frame(data, data.rendimento)
data.rendimento$data.1 = NULL
```

4. Junte com a base anterior já organizada.

```
data.base <- data.base %>%
  filter(data >= "2012-03-01" &
         data <= "2021-10-01")
# Junção das bases

data.base <- data.frame(data.base, data.rendimento)

# Limpando

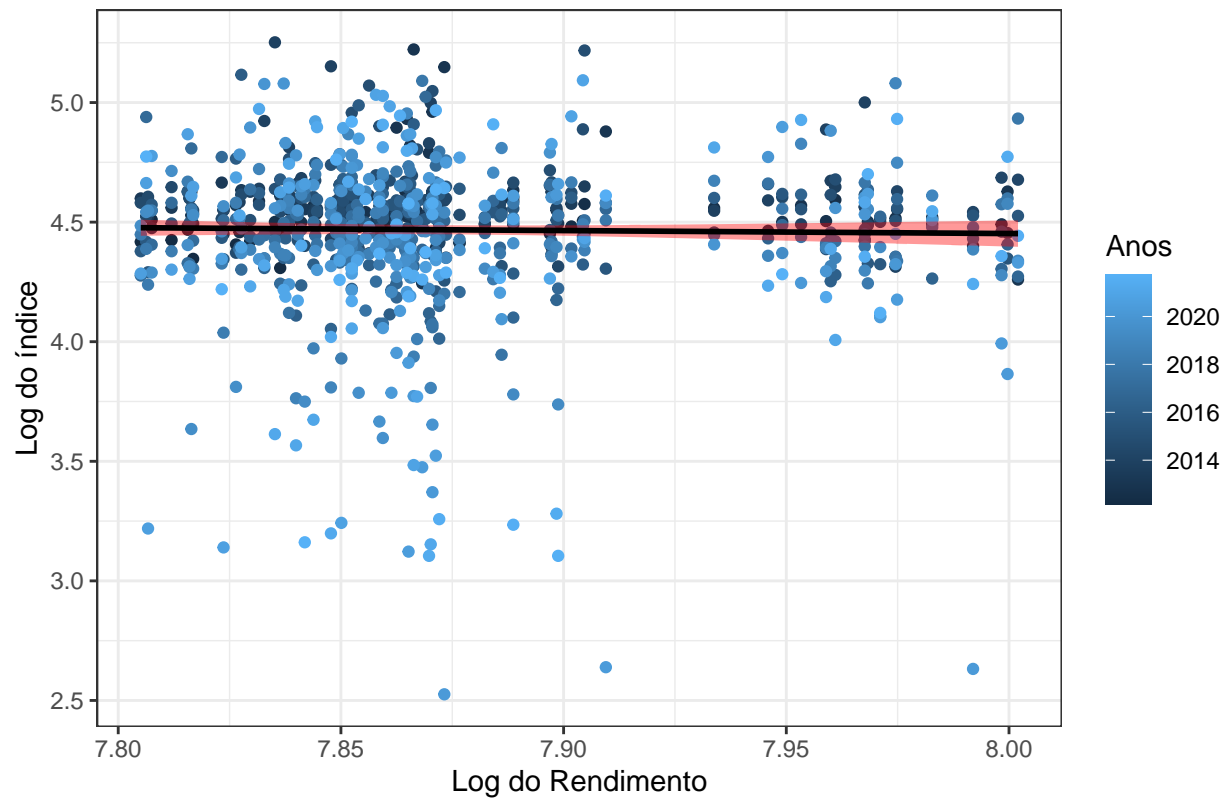
data.base <- data.base %>%
  select(!c(var, setor, data.1)) %>%
  rename("Anos" = "data")
```

5. Gere uma visualização com as duas variáveis de interesse (note que é apenas para Brasil).

Antes da plotagem da relação entre as variáveis rendimento e índice do volume de vendas, optou-se por transformar ambas variáveis por logaritmo natural, assim, buscando uma melhor estabilização da variância

```
ggplot(data = data.base, aes(
  y = log(BR), x = log(renda), colour = Anos))+
  geom_point(pch = 19)+
  geom_smooth(method = "lm", col = "black", fill = "red")+
  labs(y = "Log do índice", x = "Log do Rendimento",
       title = "Índice vs Rendimento - Ajustamento Linear Via MQO")+
  theme_bw()
```

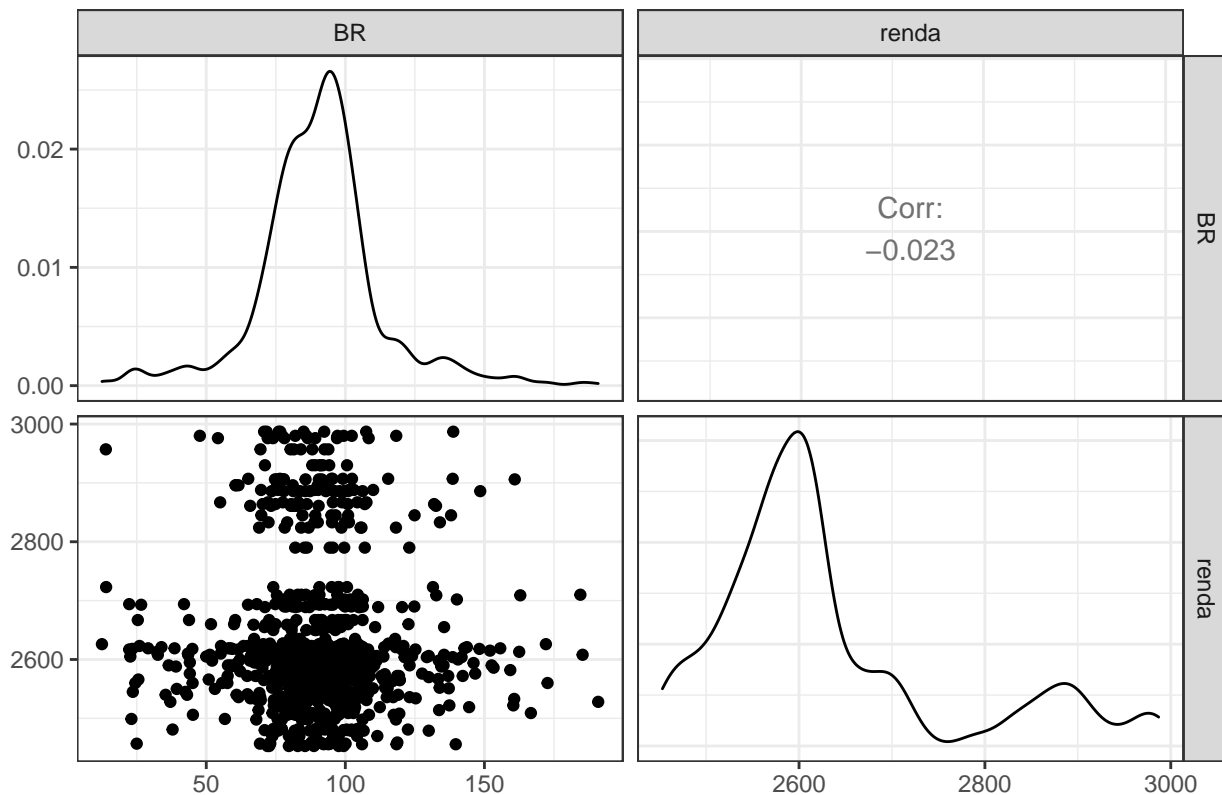
Índice vs Rendimento – Ajustamento Linear Via MQO



Como observado, as variáveis não têm um bom ajustamento linear. Vamos ver a correlação entre as variáveis e a distribuição das variáveis

```
ggpairs(data.base[c(2, 15)],  
        lower = list(contínuos = "smooth"))+  
ggtitle("Matriz de Correlação de Pearson")+  
theme_bw()
```

Matriz de Correlação de Pearson



Nota-se uma fraquíssima correlação negativa e, para piorar, não significativa

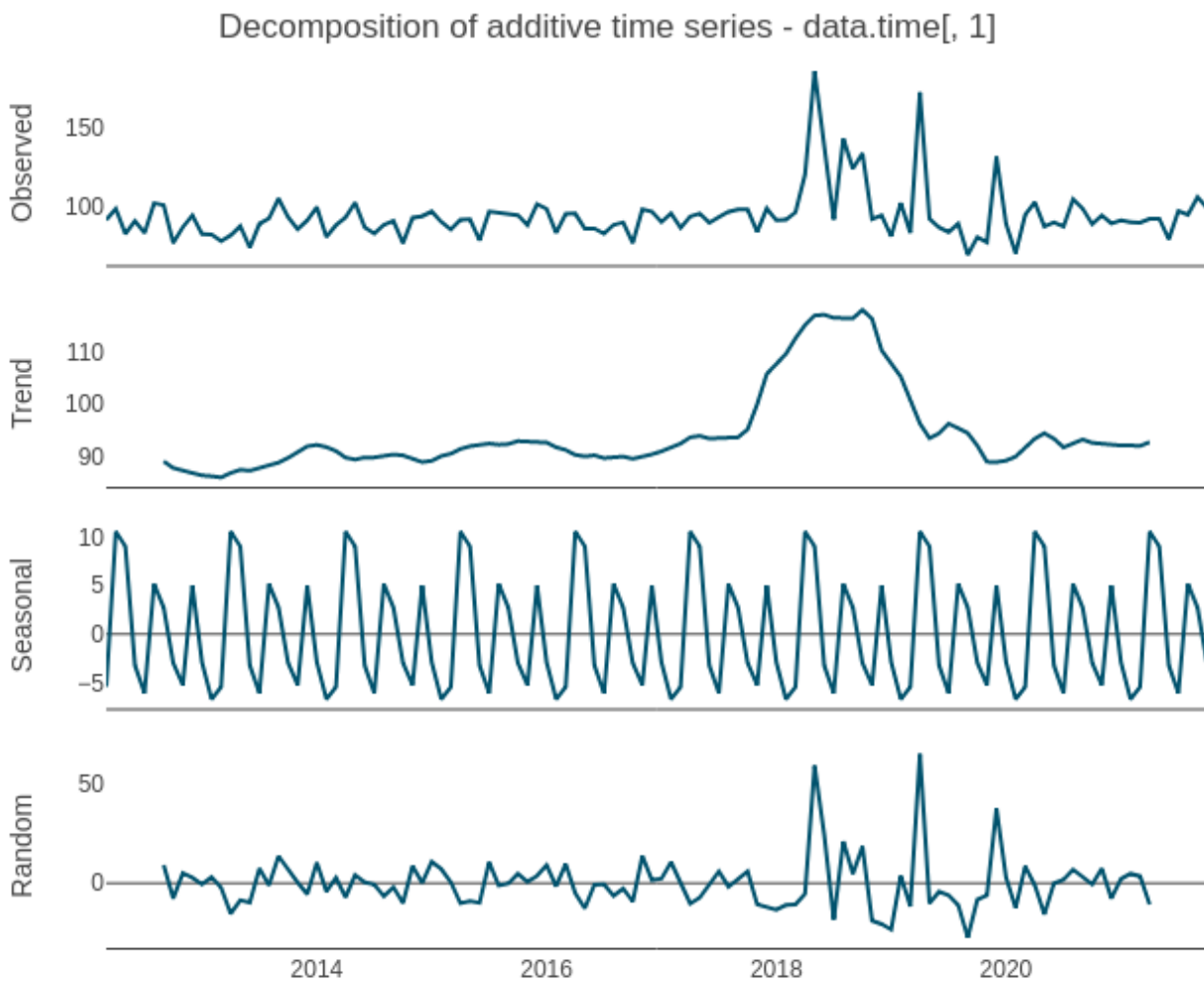
6. Crie um modelo estatístico que estime a relação entre volume de vendas de móveis e eletrodomésticos com o rendimento real médio efetivo.

Como indica os dois plotes anteriores, aparentemente uma modelagem via MQO pressupondo linearidade não é adequado. Não obstante, tratando-se de uma série temporal cabe alguns testes antes da escolha da modelagem adequada. Primeiro, vamos observar a decomposição da variável dependente (índice de venda) e variável explicativa (renda):

```
data.time <- ts(data = data.base[c(2, 15)],
               start = c(2012, 3), end = c(2021, 10), frequency = 12)
```

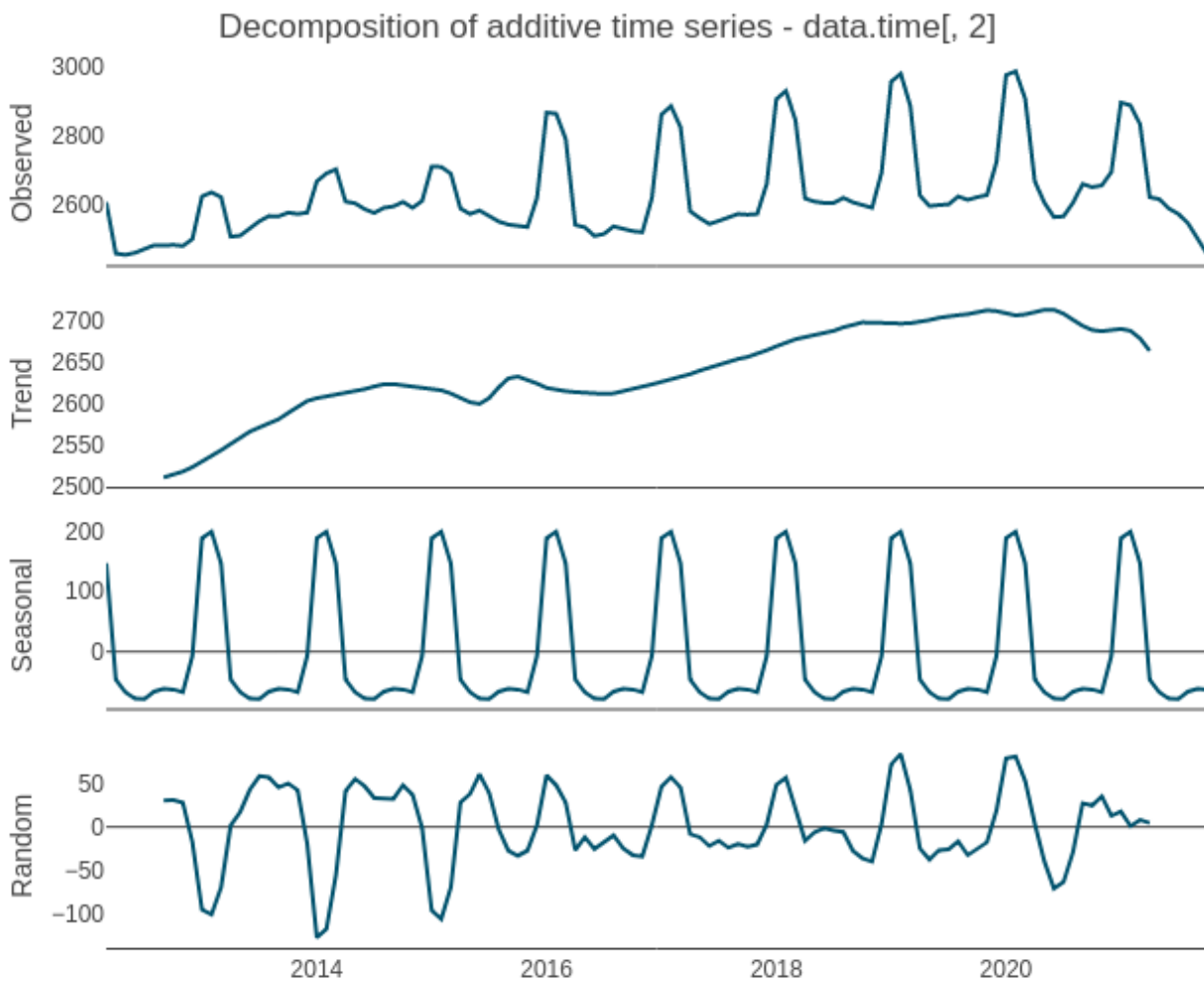
A decomposição da série temporal do índice como pode ser visualizado a seguir indica sazonalidade

```
ts_decompose(data.time[, 1], type = "additive")
```



Assim como, a renda também apresenta sazonalidade

```
ts_decompose(data.time[, 2], type = "additive")
```



Testando estacionariedade

Outro teste fundamental dado o problema de modelagem com séries temporais e testar a presença de raiz unitária nas séries Para isso optou-se em usar o teste de Phillips-Perron, o qual segue o teste de hipótese:

H_0 = estacionário: $p > 0.05$

H_a = não estacionário: $p \leq 0.05$

```
pp.indice <- ur.pp(data.time[,1])
summary(pp.indice)
```

```
##
## #####
## # Phillips-Perron Unit Root Test #
## #####
##
## Test regression with intercept
##
##
```

```
## Call:
## lm(formula = y ~ y.l1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.577  -8.113  -1.591   2.853  83.407
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  66.86176    8.60938   7.766   4e-12 ***
## y.l1         0.29120    0.09003   3.235   0.0016 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.71 on 113 degrees of freedom
## Multiple R-squared:  0.08474, Adjusted R-squared:  0.07664
## F-statistic: 10.46 on 1 and 113 DF, p-value: 0.001598
##
##
## Value of test-statistic, type: Z-alpha is: -89.2992
##
##      aux. Z statistics
## Z-tau-mu      7.9465
```

```
pp.renda <- ur.pp(data.time[, 2])
summary(pp.renda)
```

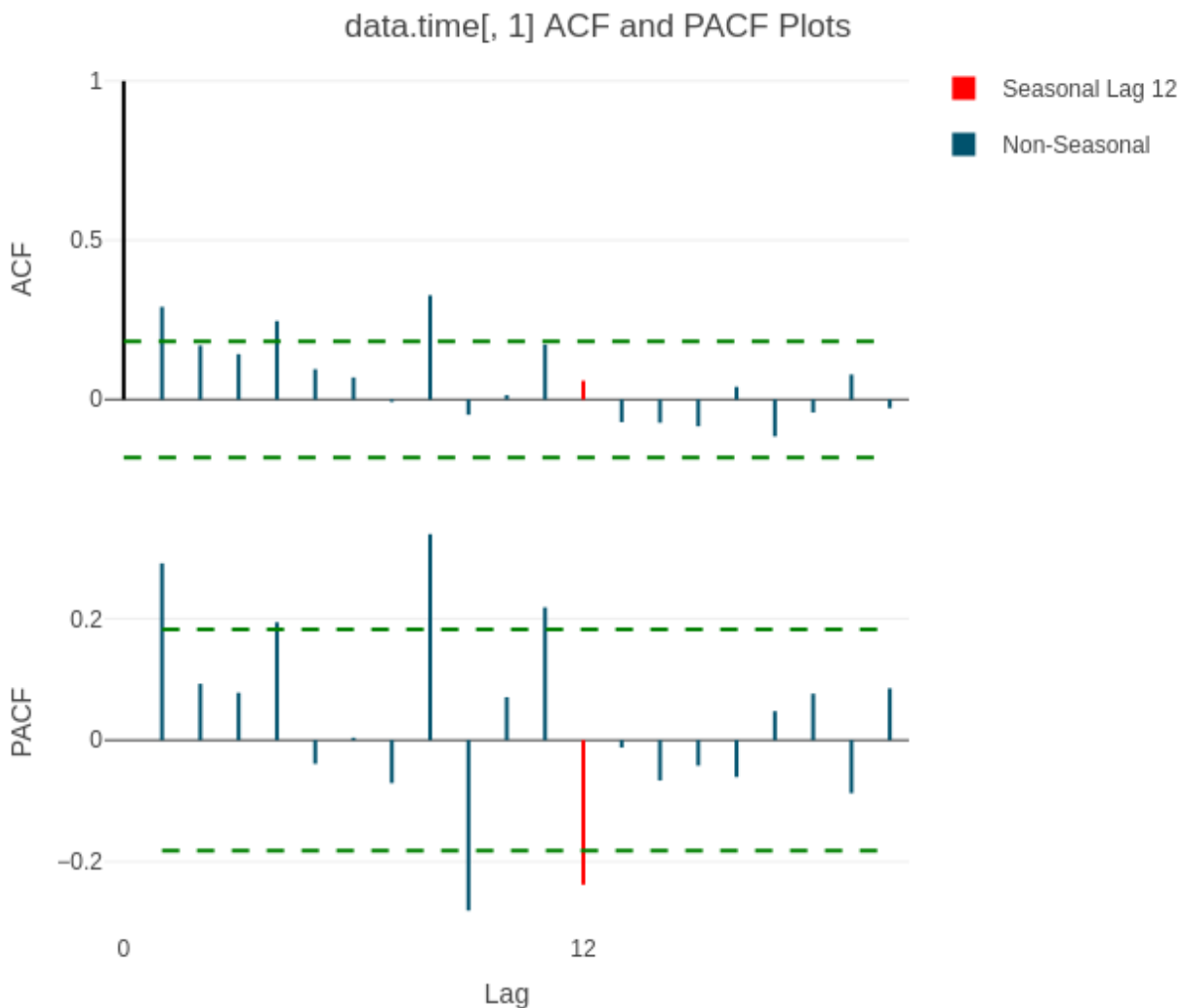
```
##
## #####
## # Phillips-Perron Unit Root Test #
## #####
##
## Test regression with intercept
##
##
## Call:
## lm(formula = y ~ y.l1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -211.95  -27.50  -10.73   10.90  279.52
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  609.88395  161.89772   3.767 0.000264 ***
## y.l1         0.76776    0.06144  12.495 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 83.95 on 113 degrees of freedom
## Multiple R-squared:  0.5801, Adjusted R-squared:  0.5764
## F-statistic: 156.1 on 1 and 113 DF, p-value: < 2.2e-16
##
##
```

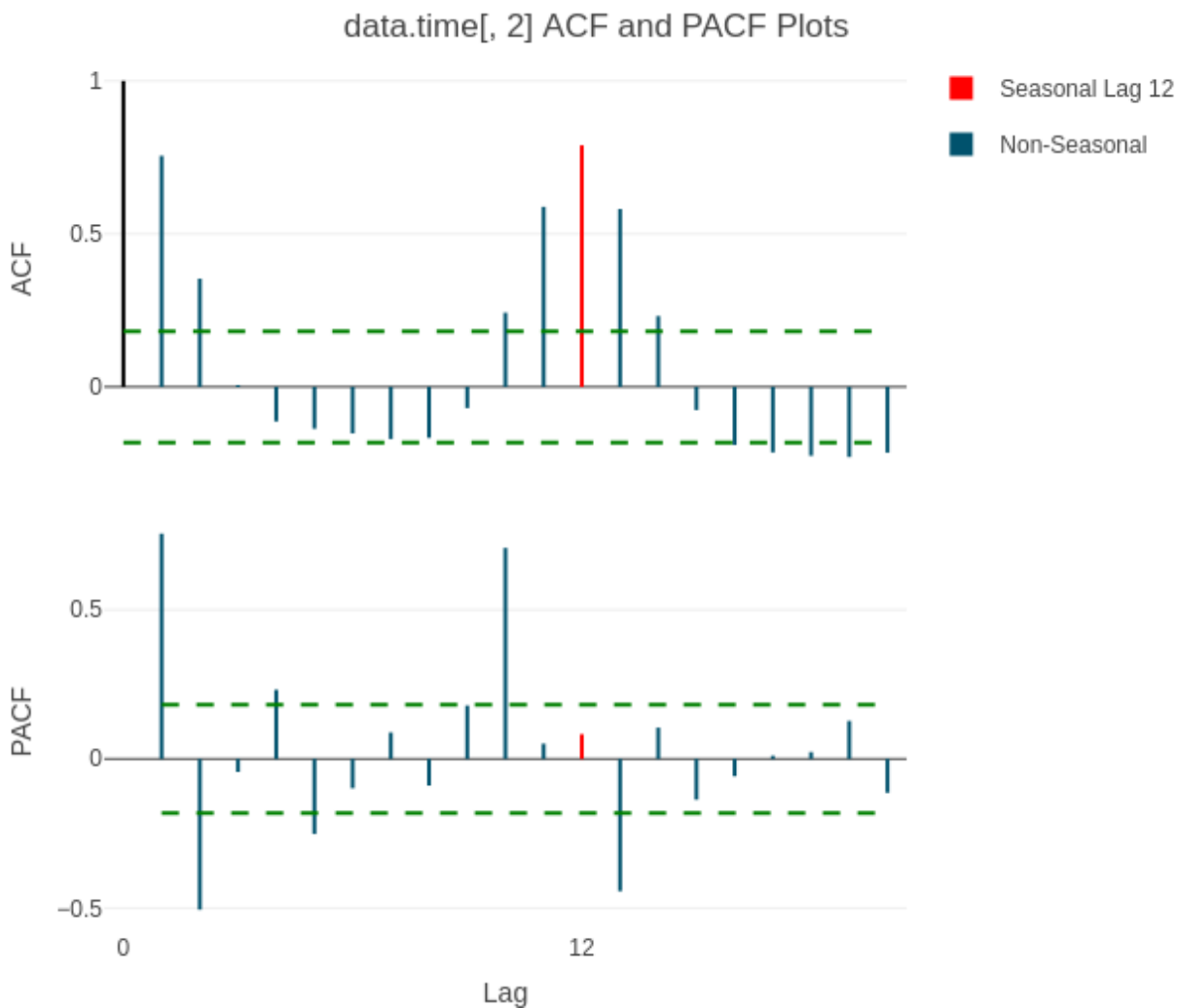
```
## Value of test-statistic, type: Z-alpha is: -30.8746
##
##      aux. Z statistics
## Z-tau-mu      4.032
```

Como pode ser observado, a série temporal do índice de venda não é estacionário. Assim como, a renda também não é estacionária. Com séries não estacionárias a modelagem torna-se mais complexa. Precisando, assim, testar a cointegração para posteriormente a escolha da modelagem que, dependendo do resultado da cointegração, precisará passar por diferenciação das séries. Não obstante, antes vamos testar a presença de autocorrelação nas séries como segue com ACF e PACF:

```
ts_cor(data.time[, 1], seasonal = T, ci = 0.95)
```

```
ts_cor(data.time[, 2], seasonal = T, ci = 0.95)
```





Como pode ser observado, ambas apresentam autocorrelação

Teste de Cointegração

Continuando na modelagem, a seguir aplica-se o teste de cointegração. Para o teste de cointegração optou-se em escolher o teste de Johansen. Primeiro, aplica-se o critério VAR de Seleção para encontrar o número ótimo de lags:

```
library(vars) # Critério Var e Teste de Cointegração

lagselect <- VARselect(data.time, lag.max = 24, type = "trend", season = 12)

select.var <- data.frame(lagselect$selection)
select.criterio <- data.frame(lagselect$criteria)
lagselect$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      12      4      4      12
```



```
lagselect$criteria
```

```
##           1           2           3           4           5
## AIC(n)    12.86480    12.66665    12.62481    12.38918    12.45010
## HQ(n)     13.17456    13.02068    13.02309    12.83171    12.93688
## SC(n)     13.63230    13.54380    13.61160    13.48561    13.65618
## FPE(n) 388310.29027 319271.40526 307132.05583 243599.37926 260143.37065
##           6           7           8           9          10
## AIC(n)    12.50015    12.52235    12.48495    12.44218    12.35454
## HQ(n)     13.03119    13.09764    13.10449    13.10597    13.06258
## SC(n)     13.81587    13.94771    14.01995    14.08682    14.10883
## FPE(n) 275086.37637 283219.98917 275068.23335 266094.07660 246497.98654
##          11          12          13          14          15
## AIC(n)    12.42122    12.28108    12.36474    12.43844    12.49832
## HQ(n)     13.17352    13.07763    13.20554    13.32349    13.42763
## SC(n)     14.28515    14.25465    14.44796    14.63130    14.80083
## FPE(n) 266902.50303 235442.09675 260319.70170 285621.49916 309858.14788
##          16          17          18          19          20
## AIC(n)    12.51902    12.34493    12.39627    12.44784    12.47823
## HQ(n)     13.49258    13.36275    13.45834    13.55416    13.62881
## SC(n)     14.93117    14.86672    15.02770    15.18892    15.32895
## FPE(n) 324125.67681 279891.74961 303828.22929 331123.20042 354772.72020
##          21          22          23          24
## AIC(n)    12.53399    12.52671    12.58531    12.52615
## HQ(n)     13.72882    13.76579    13.86864    13.85373
## SC(n)     15.49435    15.59671    15.76496    15.81544
## FPE(n) 391683.27408 408117.91180 456854.82363 457595.87906
```

Optante em selecionar com o critério AIC, posteriormente, aplica-se o teste de cointegração:

```
ctest1t <- ca.jo(data.time, type = "trace", ecdet = "trend", K = 12)
summary(ctest1t)
```

```
##
## #####
## # Johansen-Procedure #
## #####
##
## Test type: trace statistic , with linear trend in cointegration
##
## Eigenvalues (lambda):
## [1] 1.332491e-01 5.318599e-02 -3.039766e-18
##
## Values of teststatistic and critical values of test:
##
##          test 10pct  5pct  1pct
## r <= 1 |  5.68 10.49 12.25 16.26
## r = 0  | 20.56 22.76 25.32 30.45
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
```

```
##          BR.112  renda.112  trend.112
## BR.112      1.00000000  1.0000000  1.0000000
## renda.112 -0.07461765  0.3199220 -0.9548187
## trend.112 -0.13894530 -0.1107467  1.6750909
##
## Weights W:
## (This is the loading matrix)
##
##          BR.112  renda.112  trend.112
## BR.d      -0.4093251 -0.1069745 -1.485238e-12
## renda.d   0.7540772 -0.2229359 -2.457955e-11
```

```
ctest1e <- ca.jo(data.time, type = "eigen", ecdet = "trend", K = 12)
summary(ctest1e)
```

```
##
## #####
## # Johansen-Procedure #
## #####
##
## Test type: maximal eigenvalue statistic (lambda max) , with linear trend in cointegration
##
## Eigenvalues (lambda):
## [1] 1.332491e-01 5.318599e-02 -3.039766e-18
##
## Values of teststatistic and critical values of test:
##
##          test 10pct  5pct  1pct
## r <= 1 |  5.68 10.49 12.25 16.26
## r = 0  | 14.87 16.85 18.96 23.65
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##          BR.112  renda.112  trend.112
## BR.112      1.00000000  1.0000000  1.0000000
## renda.112 -0.07461765  0.3199220 -0.9548187
## trend.112 -0.13894530 -0.1107467  1.6750909
##
## Weights W:
## (This is the loading matrix)
##
##          BR.112  renda.112  trend.112
## BR.d      -0.4093251 -0.1069745 -1.485238e-12
## renda.d   0.7540772 -0.2229359 -2.457955e-11
```

Como pode ser observado, as séries não são cointegradas nem mesmo considerando 10%.

Assim, encontra-se de acordo com a literatura em séries temporais um dos modelos possíveis para modelar dado os resultados encontrados. Com séries não estacionárias e não cointegradas (como os resultados indicam), um dos caminhos possíveis é diferenciar os dados até encontrar estacionariedade e estimar um VAR com as séries diferenciadas.