Trabalho final de Data Mining – Coronavírus no Rio Grande do Sul

Gabriel Vinícius Heisler[1]

[1]Universidade Federal de Santa Maria (UFSM)
Santa Maria – RS – Brasil

gvheisler@inf.ufsm.br

Abstract. *In a environment with lots of data we need effective ways to discover useful information without wasting too much time. One of the main ways to do this is by using data mining and analysis. This is the final work of the Data Mining course and aims to explore the data from the COVID-19 pandemic in Rio Grande do Sul in order to find useful information.*

Resumo. *Em um ambiente com muitos dados precisamos de maneiras eficazes de descobrir informações úteis sem perder muito tempo. Uma das principais maneiras para isso é utilizando a análise e a mineração de dados. Este é o trabalho final da disciplina de Data Mining e tem como objetivo explorar os dados da pandemia da COVID-19 no Rio Grande do Sul, a fim de encontrar informações úteis.*

1. Introdução

O processo de descoberta de conhecimento em bases de dados vem se tornando cada vez mais frequente, adentrando diversas áreas da ciência. Não é diferente na área da saúde. A descoberta de padrões em sintomas de uma doença pode facilitar a sua detecção, a previsão da quantidade de casos de uma determinada epidemia e os seus picos pode ajudar os órgãos responsáveis a se prepararem, entre muitos outros usos. Neste trabalho, analisaremos os dados da pandemia da COVID-19 (dados desde o início da pandemia, em 2020, até o momento, no ano de 2023) no estado do Rio Grande do Sul, utilizando a linguagem R para mostrar fatores interessantes e utilizar técnicas de mineração de dados para realizar associações e previsões. Para este trabalho, serão adotadas as etapas do processo denominado *KDD – Knowledge Discovery in Databases* (ou, em português, descoberta de conhecimento em bases de dados) propostas por [Fayyad et al. 1996]

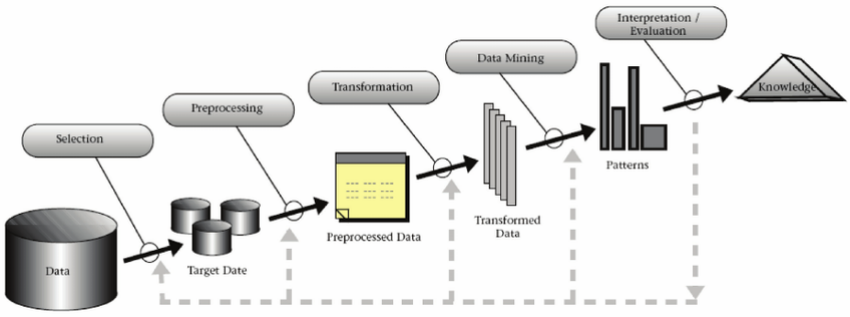

Figura 1. Etapas do processo de KDD. Figura retirada de [Fayyad et al. 1996]

2. SARS-CoV-2

Em dezembro de 2019, um surto de pneumonia de origem desconhecida foi relatado em Wuhan, província de Hubei, na China. Segundo [Ciotti et al. 2020], a investigação de amostras respiratórias em células das vias aéreas humanas levou ao isolamento de um novo vírus respiratório, que foi denominado SARS coronavírus 2 (SARS-CoV-2).

2.1. SARS, ou SRAG

SARS, como define a Organização Mundial de Saúde [WorldHealthOrganization], significa, em inglês, *Severe Acute Respiratory Syndrome*. Em português, utilizamos o termo *SRAG*, Síndrome Respiratória Aguda Grave.

2.2. A pandemia da COVID-19

No dia 11 de março de 2020, a Organização Mundial da Saúde declarou a COVID-19, a doença causada pelo vírus SARS-CoV-2, uma pandemia mundial [Cucinotta and Vanelli 2020]. Isto resultou em muitas consequências, as principais sendo os *lockdowns* e a corrida pelas vacinas. A pandemia não afetou somente a saúde física do povo: com o fechamento das cidades e empresas, para a tentativa de controle da pandemia, foram gerados e agravados diversos problemas socio-econômicos [Nicola et al. 2020], sociais e psicológicos [Osofsky et al. 2020]. Isso, é claro, sem mencionar as mais de seis milhões de vítimas fatais desta doença.

3. Extração e observação dos dados

Como já mencionado, os dados que iremos analisar[1] são a respeito dos casos confirmados de COVID-19 no Rio Grande do Sul, desde o começo da pandemia (em fevereiro de 2020) até o começo de 2023. Esses dados são disponibilizados publicamente na web pelo governo do estado do Rio Grande do Sul[2]. Em R[3], podemos fazer a leitura do conjunto de dados com a função read.csv2 (2 pois nosso conjunto de dados é separado pelo caracter ';'). Como parâmetro, podemos utilizar a *url* de download direto do conjunto de dados no formato CSV[4]

```
read.csv2(url("https://ti.saude.rs.gov.br/covid19/download"))
```

Após a leitura dos dados, podemos observar que nosso conjunto de dados conta com aproximadamente 3 milhões de linhas (em cada linha temos um caso confirmado de COVID-19 no RS) e 30 colunas (veja na figura 2 uma parte do conjunto).
Como podemos consultar em ti.saude.rs.gov.br/covid19/api, nas colunas do nosso conjunto de dados (o qual agora, após a leitura, é da classe data.frame), temos os seguintes dados:

[1] Vídeo de apresentação disponível no Drive
[2] Dados disponíveis em ti.saude.rs.gov.br/covid19/api
[3] Todos os códigos disponíveis no github
[4] Link para download direto dos dados completos

COD_IBGE	MUNICIPIO	COD_REGIAO_COVID	REGIAO_COVID	SEXO	FAIXAETARIA	CRITERIO	DATA_CONFIRMACAO	DATA_SINTOMAS	DATA_INCLUSAO
430003	ACEGUÁ	16	BAGE - R22	Feminino	20 a 29	TESTE RÁPIDO	26/01/2022	25/01/2022	29/01/2022
430003	ACEGUÁ	16	BAGE - R22	Masculino	20 a 29	TESTE RÁPIDO	27/01/2022	24/01/2022	29/01/2022
430003	ACEGUÁ	16	BAGE - R22	Feminino	20 a 29	TESTE RÁPIDO	27/01/2022	27/01/2022	29/01/2022
430003	ACEGUÁ	16	BAGE - R22	Feminino	40 a 49	TESTE RÁPIDO	27/01/2022	25/01/2022	29/01/2022
430003	ACEGUÁ	16	BAGE - R22	Feminino	50 a 59	TESTE RÁPIDO	28/01/2022	26/01/2022	29/01/2022
430003	ACEGUÁ	16	BAGE - R22	Masculino	30 a 39	TESTE RÁPIDO	21/01/2022	21/01/2022	29/01/2022
430003	ACEGUÁ	16	BAGE - R22	Feminino	01 a 04	TESTE RÁPIDO	27/01/2022	25/01/2022	29/01/2022
430003	ACEGUÁ	16	BAGE - R22	Masculino	40 a 49	TESTE RÁPIDO	27/01/2022	25/01/2022	29/01/2022
430003	ACEGUÁ	16	BAGE - R22	Masculino	01 a 04	TESTE RÁPIDO	27/01/2022	25/01/2022	29/01/2022
430003	ACEGUÁ	16	BAGE - R22	Masculino	60 a 69	TESTE RÁPIDO	27/01/2022	25/01/2022	29/01/2022
430003	ACEGUÁ	16	BAGE - R22	Feminino	20 a 29	TESTE RÁPIDO	27/01/2022	25/01/2022	29/01/2022

Figura 2. Parte do conjunto de dados

COD_IBGE	Código IBGE do Município
MUNICIPIO	Nome do município
COD_REGIAO_COVID	Código da região de saúde COVID
REGIAO_COVID	Nome da região de saúde COVID
SEXO	Sexo
FAIXAETARIA	Faixa Etária
CRITERIO	Tipo de teste
DATA_CONFIRMACAO	Data de confirmação
DATA_SINTOMAS	Data de início dos sintomas
DATA_INCLUSAO	Data de inclusão no dashboard do RS
DATA_EVOLUCAO	Data da evolução
EVOLUCAO	Descrição da evolução
HOSPITALIZADO	Paciente foi hospitalizado
FEBRE	Sintomas de febre
TOSSE	Sintomas de tosse
GARGANTA	Sintomas de dor de garganta
DISPNEIA	Sintomas de dor de dispnéia/falta de ar
OUTROS	Outros sintomas
CONDICOES	Alguma condição de saúde que necessite atenção
GESTANTE	Paciente é gestante
DATA_INCLUSAO_OBITO	Data de inclusão da informação de óbito
DATA_EVOLUCAO_ESTIMADA	Data da evolução estimada (casos não hospitalizados)
RACA_COR	Raça/Cor
ETNIA_INDIGENA	Etnia indígena
PROFISSIONAL_SAUDE	Profissional de saúde
BAIRRO	Bairro (apenas municípios com mais de 100.000 hab)
SRAG	Paciente apresentou síndrome respiratória aguda grave
FONTE_INFORMACAO	Fonte da informação
PAIS_NASCIMENTO	Pais de nascimento
PES_PRIV_LIBERDADE	Pessoa privada de liberdade

Figura 3. Dicionário de dados

4. Geração inicial de gráficos

Como podemos perceber, temos uma quantia considerável de dados. Para realizar as análises, teremos que filtrar estes dados de acordo com nossas necessidades. Como primeira análise, vamos gerar um gráfico contendo a quantidade de casos confirmados por dia, desde o começo da pandemia:
Primeiramente, devemos carregar o dataset.

```
ds <- read.csv2("caminho\\ou\\link\\do\\dataset.csv")
```

Após, selecionamos as colunas que iremos usar para gerar o gráfico desejado. Baseado no dicionário de dados, queremos as colunas 8 e 12 (Data de confirmação e Evolução, respectivamente).

```
ds <- ds[,c(8,12)]
```

Como queremos gerar um gráfico usando dias como índice, devemos transformar nossa coluna de Data de confirmação no formato Date (a coluna originalmente está no formato character).

```
ds$DATA_CONFIRMACAO <- as.Date(ds$DATA_CONFIRMACAO,
                               format = "%d/%m/%Y")
```

Agora, criamos um vetor com todas as datas entre o primeiro caso e o último caso registrados no nosso conjunto, e criamos um novo data.frame no qual salvamos a exata quantidade de casos registrados em cada dia.

```
dias <- seq(as.Date(min(ds$DATA_CONFIRMACAO)),
    as.Date(max(ds$DATA_CONFIRMACAO)), by = 'days')

casosDia <- data.frame(matrix(nrow = length(dias), ncol = 2))

colnames(casosDia) <- c('dia', 'casos')

casosDia$dia <- dias

for (i in 1:nrow(casosDia)) {
  dfAux <- ds[which(ds$DATA_CONFIRMACAO==casosDia[i,1]),]
  casosDia[i,2] <- nrow(dfAux)
}
```

Agora, no nosso dataframe "casosDia"temos em uma coluna as datas e na outra os casos confirmados na data.

dia	casos
2020-02-25	3
2020-02-26	7
2020-02-27	5
2020-02-28	5
2020-02-29	1

Figura 4. Começo do dataframe

Tendo estes dados, podemos facilmente plotar o gráfico de casos de COVID-19 confirmados por dia em todo o histórico. (veja na figura 5)

```
plot(x = casosDia$dia, y = casosDia$casos, type = 'l',
            xlab = 'Data', ylab = 'Quantidade de casos')
```

Neste gráfico podemos ver algumas coisas interessantes, como o fato de que existiram alguns picos da doença espalhados pelos anos. A "olho nu"podemos analisar que as principais subidas no número de casos aconteceram próximas trocas de ano. Alguns fatores

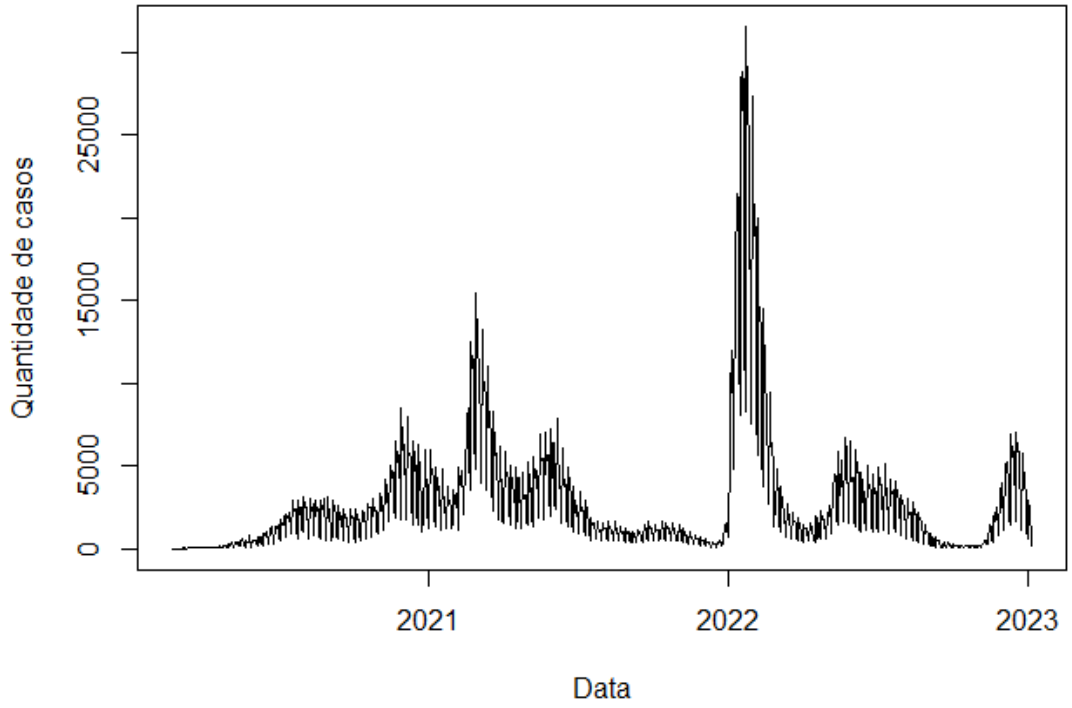

Figura 5. Gráfico de casos de COVID-19 confirmados por dia

podem ter determinado isso: a proximidade entre as pessoas nas festas de final de ano, novas variantes da doença, entre outros. Falaremos disto mais adiante. Para gerarmos um gráfico com os óbitos por COVID-19 utilizamos basicamente o mesmo processo feito para os casos, porém usamos a coluna DATA_EVOLUÇÃO e selecionamos apenas os óbitos: (podemos ver o gráfico na figura 6)

```
ds <- ds[,c(11,12)]
ds <- ds[-which(ds$EVOLUCAO!='OBITO'),]
[...]
plot(x = obitosDia$dia, y = obitosDia$dia$obitos, type = 'l',
     xlab = 'Data', ylab = 'Quantidade de óbitos', col = 'red')
```

5. Associação

A associação é uma das principais técnicas para a mineração de dados. Primeiramente, iremos aplicar o algoritmo Apriori nos sintomas para descobrir quais são os mais frequentes e quais os sintomas mais comuns em casos de óbito. Mas, antes de aplicarmos o algoritmo em si, devemos realizar o processo de KDD, como já explicado.

5.1. Aplicação do algoritmo Apriori nos sintomas dos pacientes

O algoritmo Apriori é um dos mais conhecidos e utilizados. Ele trabalha com podas baseadas em suporte.
Primeiramente selecionamos os dados:

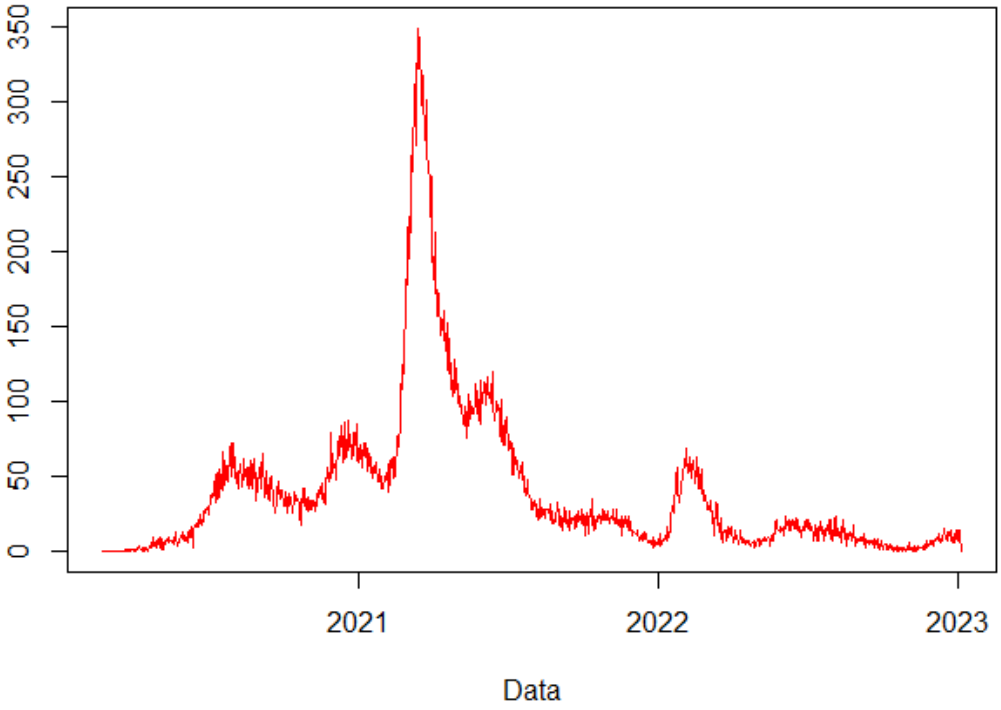

Figura 6. Gráfico de óbitos por COVID-19 por dia

```
ds <- read.csv2("caminho\\ou\\link\\do\\dataset.csv")
```

Essa é, segundo [Assunção 2021], a etapa mais trabalhosa do processo de KDD. Porém, ao verificarmos o nosso conjunto de dados, percebemos que em geral temos dados completos (sem dados essenciais faltantes). Nosso dataframe tem quase 3 milhões de linhas e 30 colunas, porém não necessitamos de todas as colunas. Iremos utilizar, para primeira análise, a coluna de evolução do caso de COVID-19 e as colunas de sintomas.

```
ds <- ds[,c(12, 14:18)]
```

Agora, nosso dataframe contém apenas 6 colunas (1 para a evolução do paciente e 5 para os sintomas). Para aplicarmos o algoritmo Apriori, precisamos que os nossos dados estejam no formato Cesta de compras, ou*one-hot-encoding*. Por "sorte", nosso dataframe atual está basicamente neste formato. Cada coluna tem apenas dois valores possíveis ("OBITO"ou "RECUPERADO"na coluna "EVOLUCAO"e "SIM"e "NAO"nas colunas dos sintomas). Então, o que precisamos fazer antes de aplicar o algoritmo é apenas transformar todos os elementos em Factor.

```
for (i in 1:ncol(ds)) {
    ds[,i] <- as.factor(ds[,i])
}
```

Agora podemos aplicar o algoritmo Apriori. Este algoritmo é da biblioteca "Arules".

```
library(arules)
```

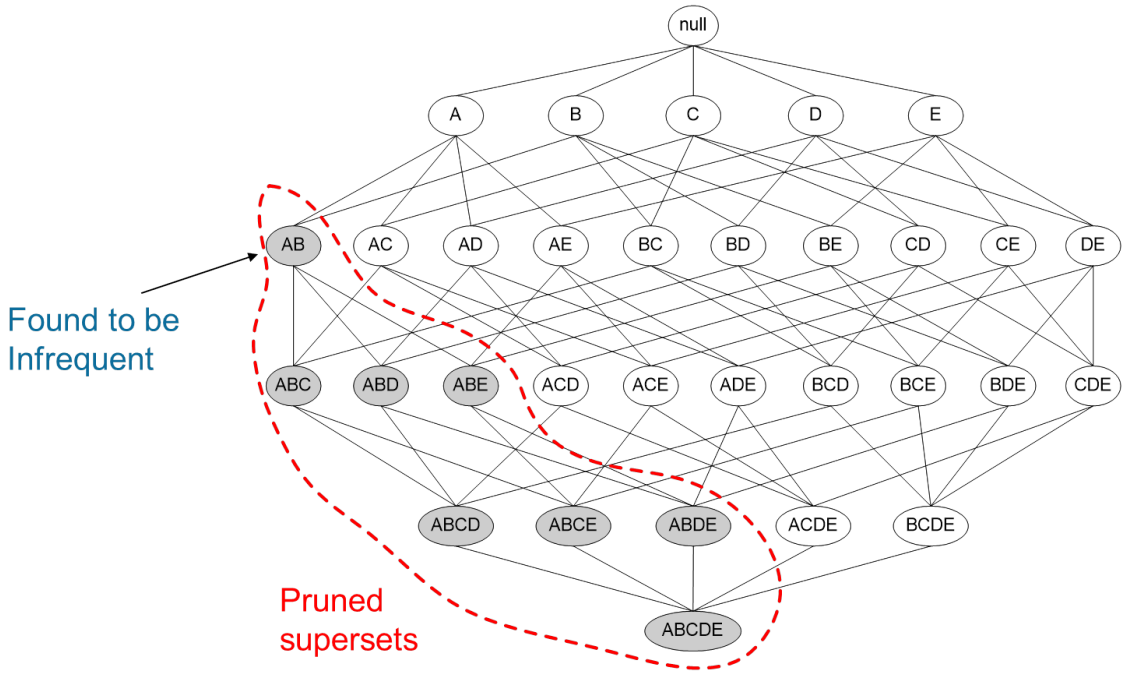

Figura 7. Ilustração de poda baseada em suporte, do livro de [Tan et al. 2016]

Agora geramos as regras de associação. Primeiramente, para teste usaremos confiança e suporte em 0.5.

```
regras <- apriori(data = ds, parameter =
    list(conf = 0.5, supp = 0.5), target = 'rules',
                                        minlen = 2)
```

Se inspecionarmos (após ordenar por maior confiança) iremos receber 23 regras e associação. as 5 principais regras são:

lhs	=>	rhs
{FEBRE=NAO, DISPNEIA=NAO}	=>	{EVOLUCAO=RECUPERADO}
{GARGANTA=NAO, DISPNEIA=NAO}	=>	{EVOLUCAO=RECUPERADO}
{DISPNEIA=NAO}	=>	{EVOLUCAO=RECUPERADO}
{DISPNEIA=NAO, OUTROS=NAO}	=>	{EVOLUCAO=RECUPERADO}
{FEBRE=NAO}	=>	{EVOLUCAO=RECUPERADO}

Porém, nessa geração apenas apareceram regras nas quais a evolução do paciente é "RECUPERADO". Caso o desejado seja ver os principais sintomas dos pacientes que vieram a óbito encontramos um problema: Se gerarmos as regras da mesma maneira mas com o suporte extremamente baixo, ainda assim não iremos gerar regras nas quais o rhs é "EVOLUCAO=OBITO". Isso acontece devido ao fato de em nossos dados termos uma quantia extremamente maior de casos nos quais a evolução do paciente foi recuperado do que óbito. Sendo assim, com tão "poucos"casos de óbito, o suporte fica baixo demais para gerarmos regras. Mas caso a gente deseje gerar essas regras mesmo assim, podemos evitar esse problema (abrindo mão de alguns "medidores"): Podemos simplesmente ignorar todas as linhas nas quais a evolução do paciente é a recuperação e considerar apenas aquelas nas quais a evolução foi óbito.

EVOLUCAO	FEBRE	TOSSE	GARGANTA	DISPNEIA	OUTROS
RECUPERADO	SIM	NAO	NAO	NAO	SIM
RECUPERADO	NAO	SIM	SIM	NAO	NAO
RECUPERADO	NAO	NAO	NAO	NAO	NAO
RECUPERADO	NAO	NAO	SIM	NAO	NAO
RECUPERADO	SIM	NAO	NAO	NAO	NAO
RECUPERADO	NAO	NAO	NAO	NAO	NAO
RECUPERADO	NAO	NAO	SIM	NAO	NAO
RECUPERADO	SIM	NAO	NAO	NAO	SIM

Figura 8. Dataframe com a evolução dos pacientes e os sintomas

```
ds <- ds[-which(ds$EVOLUCAO!='OBITO'),]
```

Assim, se gerarmos as regras, iremos conseguir regras nas quais o rhs é "EVOLUCAO=OBITO". Podemos gerar um subset das regras para filtrar o que desejamos,

```
regras <- apriori(data = ds, parameter =
        list(conf = 0.5, supp = 0.5),
        target = 'rules', minlen = 2)

regras <- subset(regras, rhs %in% 'EVOLUCAO=OBITO')
```

Porém, o problema que encontraremos é o seguinte: Como teremos apenas um tipo de evolução do paciente, a confiança da regra do apriori sempre será 1. Então, para vermos as regras mais "fortes"podemos ordenar por maior suporte.

```
regras <- sort(regras, by = 'support', decreasing = TRUE)
inspect(regras)
```

Nessas regras, podemos ver que, por algum motivo, 85% das pessoas que foram a óbito não tiveram dor de garganta, 83% tiveram dispneia, entre outras. Mais regras geradas estão disponíveis no github.

lhs	=>	rhs	support
{GARGANTA=NAO}	=>	{EVOLUCAO=OBITO}	0.8576544
{DISPNEIA=SIM}	=>	{EVOLUCAO=OBITO}	0.8325162
{OUTROS=NAO}	=>	{EVOLUCAO=OBITO}	0.7254746
{GARGANTA=NAO, DISPNEIA=SIM}	=>	{EVOLUCAO=OBITO}	0.7105263
{GARGANTA=NAO, OUTROS=NAO}	=>	{EVOLUCAO=OBITO}	0.6243451

As regras podem ser geradas de diversas maneiras diferentes, e podemos analisar as saídas e descobrir alguns fatores interessantes, como por exemplo que o principal sintoma das pessoas que foram a óbito foi dispneia, depois tosse, entre outros.

6. Árvore de decisão

Podemos também gerar uma árvore de decisão para classificar os principais fatores de risco para os pacientes. [da Silva et al. 2017] define que as árvores de decisão consistem em uma coleção de nós internos e nós folha, organizados em um modelo hierárquico. Esta é uma das técnicas mais populares na mineração de dados. Nesta tarefa, teremos o mesmo problema da tarefa anterior: a quantidade de casos que resultaram em recpueração é muito superior à quantidade de óbitos. Por isso, para a utilização do algoritmo de "Recursive Partitioning and Regression Trees", ou RPART [Therneau et al. 1997], iremos "balancear"os dados. Separaremos todos os óbitos e a mesma quantidade de recuperações e juntaremos em um dataframe, tendo exatamenteo 50% de cada classe. Também selecionaremos apenas as colunas que desejamos levar em consideração na nossa árvore de decisão e transformamos todas as colunas em Factor.

```
ds <- ds[,-c(1:11, 13, 19:26, 27:30)]
ds <- ds[-which(ds$EVOLUCAO!='OBITO'&ds$EVOLUCAO!='RECUPERADO'),]

dsObitos <- ds[which(ds$EVOLUCAO=='OBITO'),]

dsRecuperados <- ds[which(ds$EVOLUCAO=='RECUPERADO'),]
dsRecuperados <- dsRecuperados[sample(nrow(dsObitos)),]

ds <- rbind(dsRecuperados, dsObitos)

for (i in 1:ncol(ds)) {
  ds[,i] <- as.factor(ds[,i])
}
```

Agora podemos gerar a nossa árvore de decisão com a função *rpart* e mostrá-la com a função *rpart.plot*.

```
tree <- rpart(formula = EVOLUCAO ~ .,data = ds,
                    method = "class", cp = 0.1)
rpart.plot(tree, type = 3, clip.right.labs = FALSE,
                                      under = FALSE)
```

Na fórmula da árvore escolhemos EVOLUCAO como classe alvo. A árvore gerada, porém, não nos mostra muito (veja na figura 9). Isto acontece pois o sintoma de dispneia é um sintoma muito decisivo (o que também foi provado utilizando a associação) e a árvore é gerada baseada apenas nesse sintoma. Agora, para melhor demonstração, tiramos a coluna "DISPNEIA"(e mudamos o parâmetro de complexidade da árvore) e geramos a árvore novamente (veja na figura 10).

```
tree <- rpart(formula = EVOLUCAO ~ .,data = ds,
                    method = "class", cp = 0.001)
```

Perceba que agora temos uma árvore mais completa, pois os outos sintomas (que não dispneia) não são tão "decisivos".

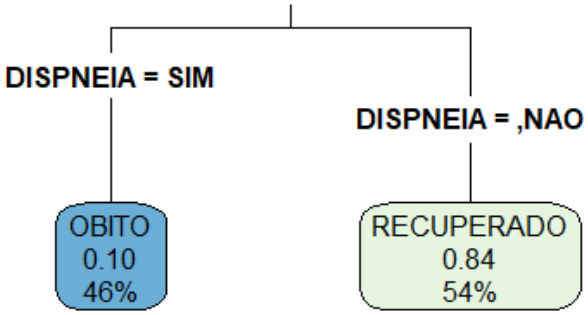

Figura 9. Árvore gerada

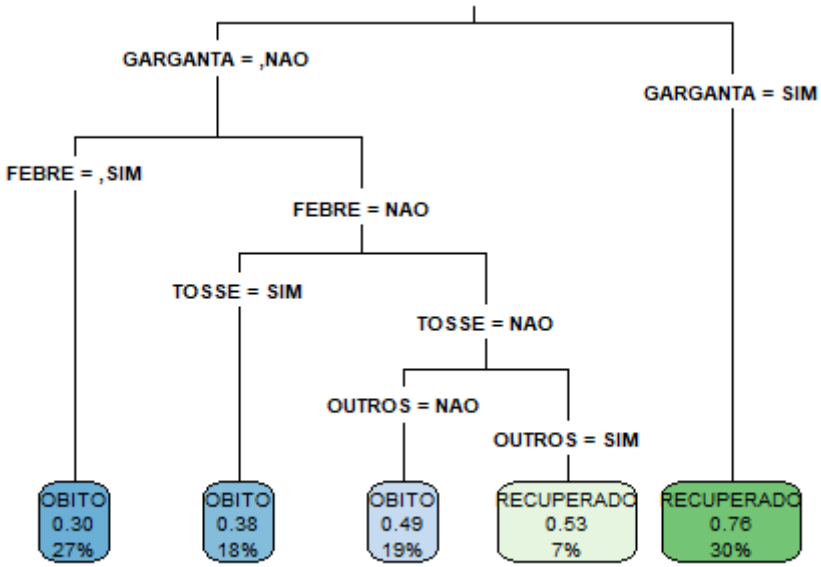

Figura 10. Nova árvore de decisão gerada baseada nos sintomas

7. Outras características e curiosidades

7.1. Picos de casos e óbitos (e o efeito das vacinas)

Se analisarmos alguns gráficos (como por exemplo o da figura 11) podemos ver que, logicamente, normalmente o aumento no número de óbitos está relacionado ao aumento no número de casos. Porém, podemos notar também que houve uma mudança entre a primeira e a segunda metade dos gráficos. Os óbitos não acompanharam a "onda"que aconteceu no começo de 2022 nos casos (até acompanharam, mas de maneira mais "calma"). Na figura 13 podemos ver um gráfico de casos e óbitos normalizado utilizando a normalização min-max. Como descrito em [Han et al. 2022], a normalização min-max

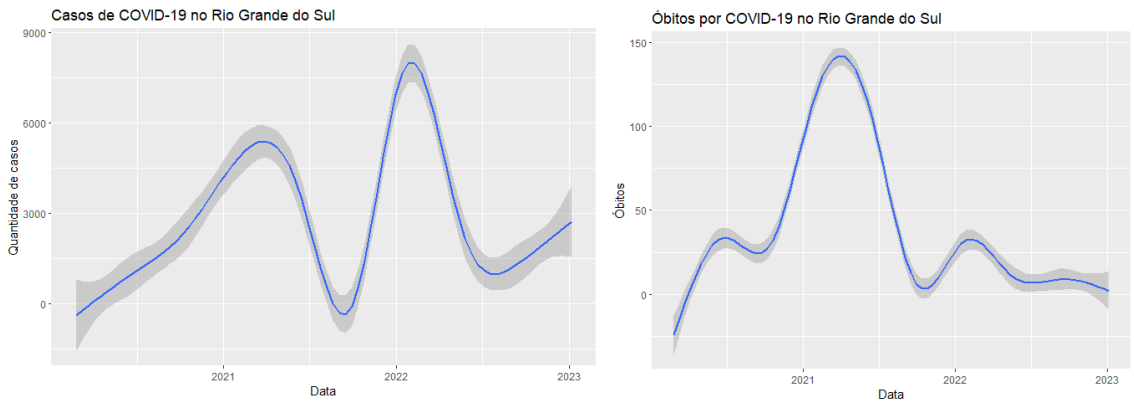

Figura 11. Gráficos de casos e óbitos suavizados

preserva a relação entre os valores originais dos dados. Neste caso, normalizamos a coluna dos casos e a coluna dos óbitos separadamente, e deixamos ambas entre 0 e 1.

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

Figura 12. Normalização min-max, retirada do livro de [Han et al. 2022]

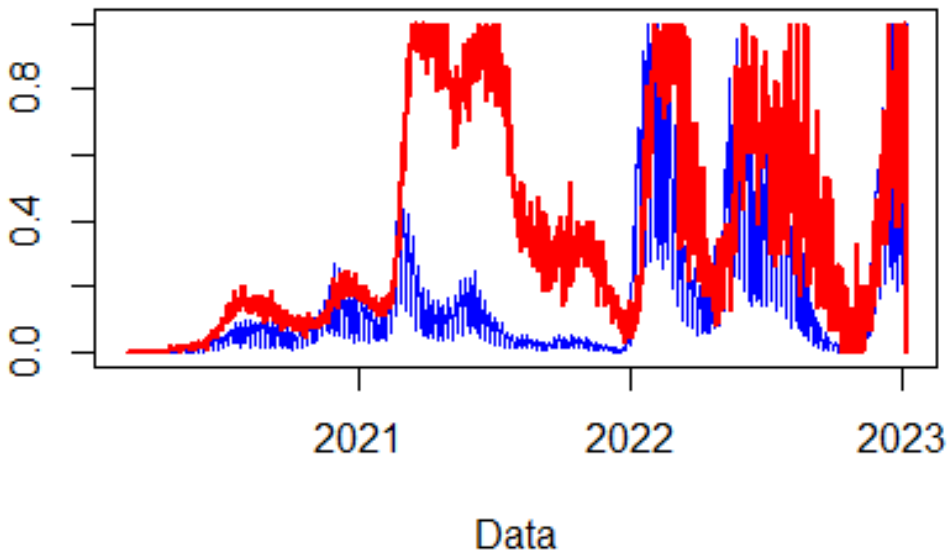

Figura 13. Gráfico normalizado entre 0 e 1

Neste gráfico percebemos que os máximos de casos e óbitos não estão relacionados. Possivelmente isso se deve ao fato da vacinação ter avançado bastante no Rio Grande do Sul durante o ano de 2021, diminuindo assim a quantidade de óbitos.

7.2. Sexo

Gerando alguns gráficos podemos verificar que homens tem mais probabilidade de irem a óbito por COVID-19

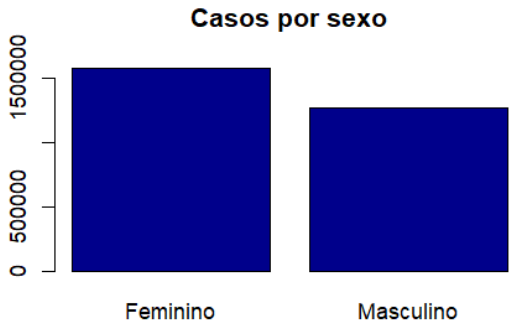

Figura 14. Casos por sexo

Figura 15. Óbitos por sexo

7.3. Faixa etária

A faixa etária está diretamente ligada aos óbitos. Isto pode ser visto utilizando mineração de dados ou simplesmente utilizando gráficos e vendo, pois é visível "a olho nu". A seguir vemos alguns gráficos que nos mostram isto. Vemos que a quantidade de casos está distribuída entre as faixas etárias, porém a quantidade de óbitos por faixa etária mostra que pessoas mais idosas são mais vulneráveis. Também é mostrado um gráfico com a porcentagem de óbitos por faixa etária (figura 18).

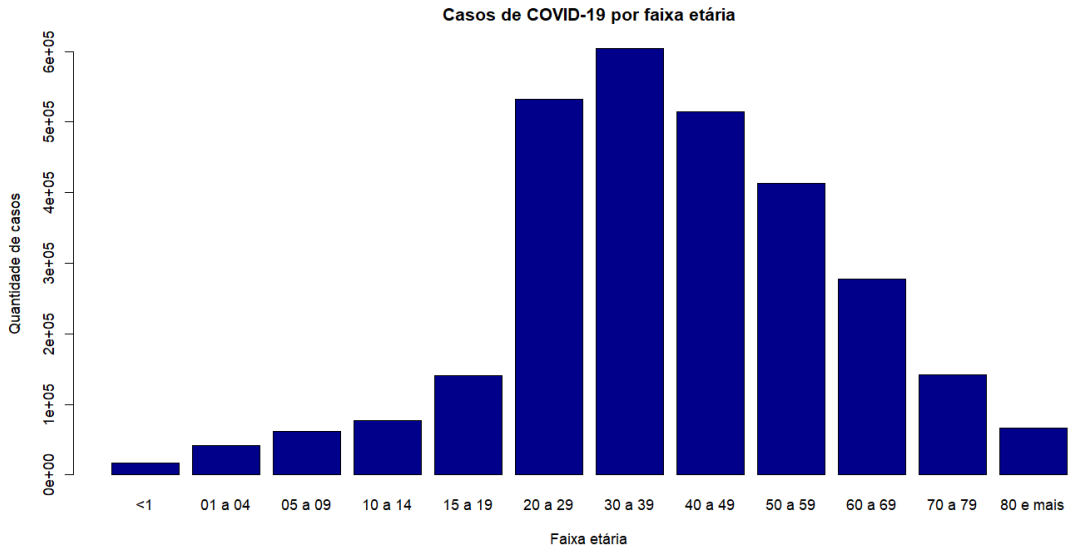

Figura 16. Casos por faixa etária

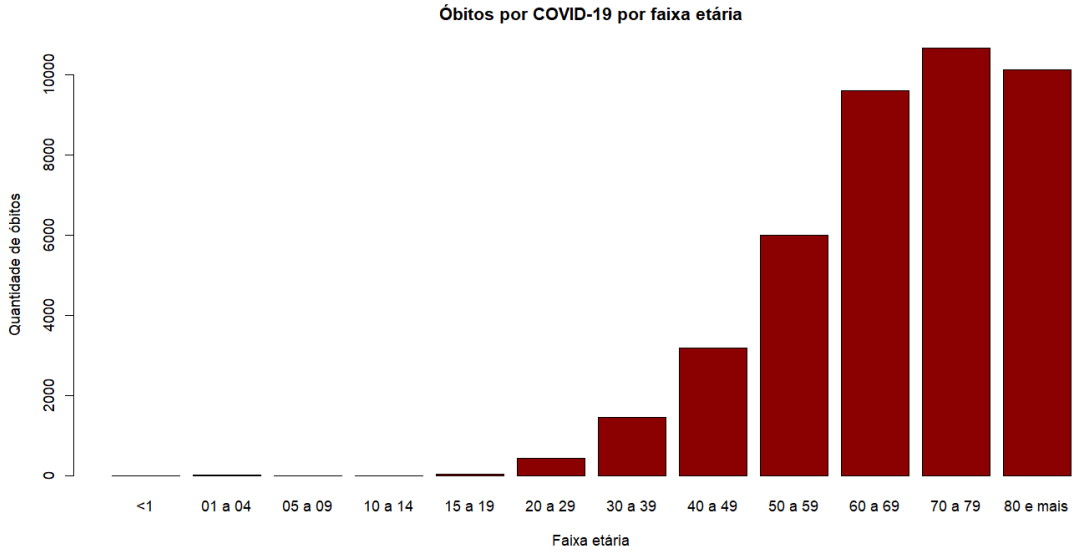

Figura 17. Óbitos por faixa etária

7.4. Semanas

Podemos notar que no gráfico geral de casos confirmados por dia temos muitas altas e baixas. Isso se deve ao fato de a média de casos confirmados em sábados e domingos ser abaixo dos outros dias da semana. A quantidade de óbitos não depende do dia da semana.

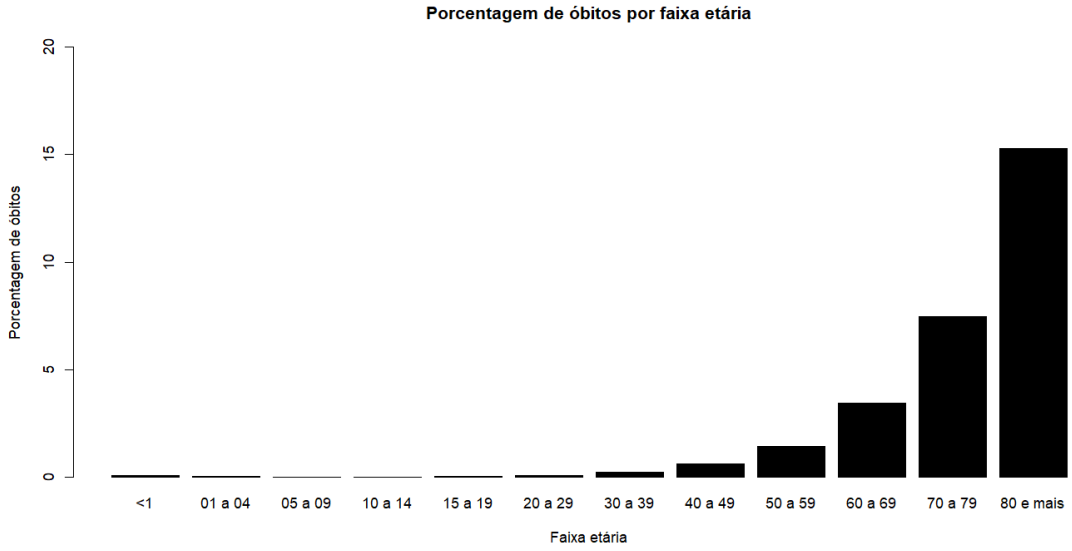

Figura 18. Porcentagem de óbitos por faixa etária

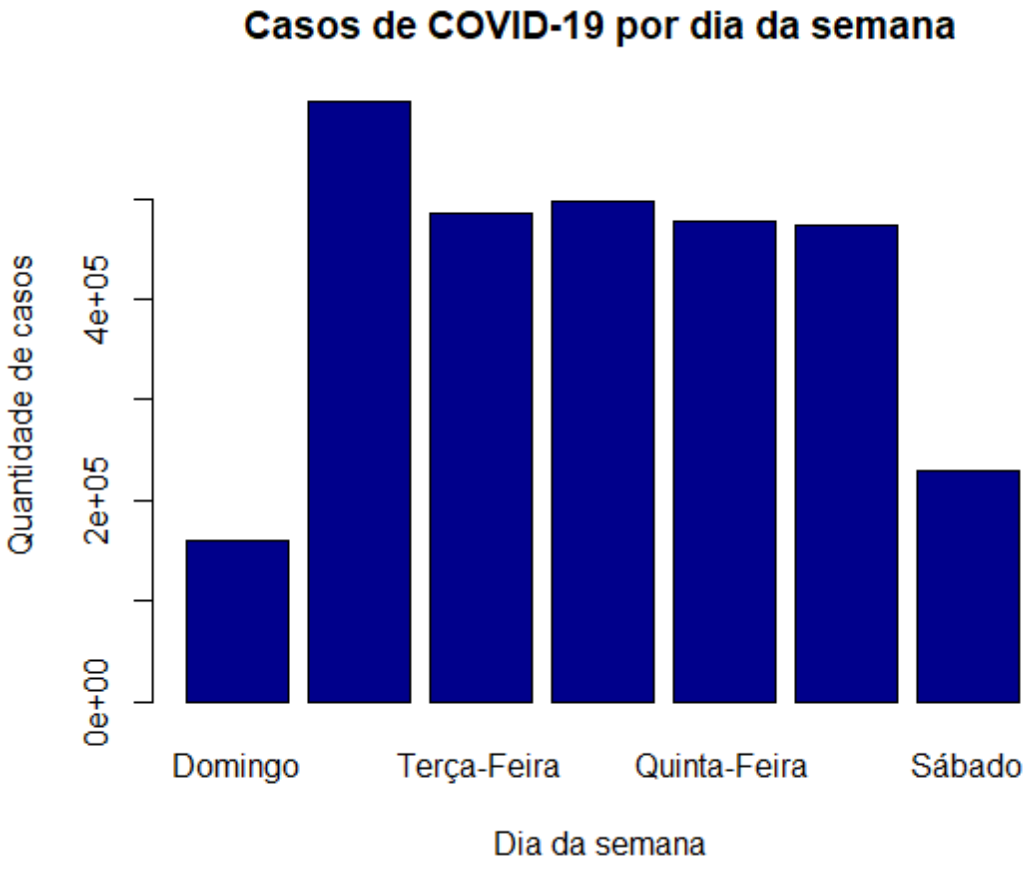

Figura 19. Casos por dia da semana

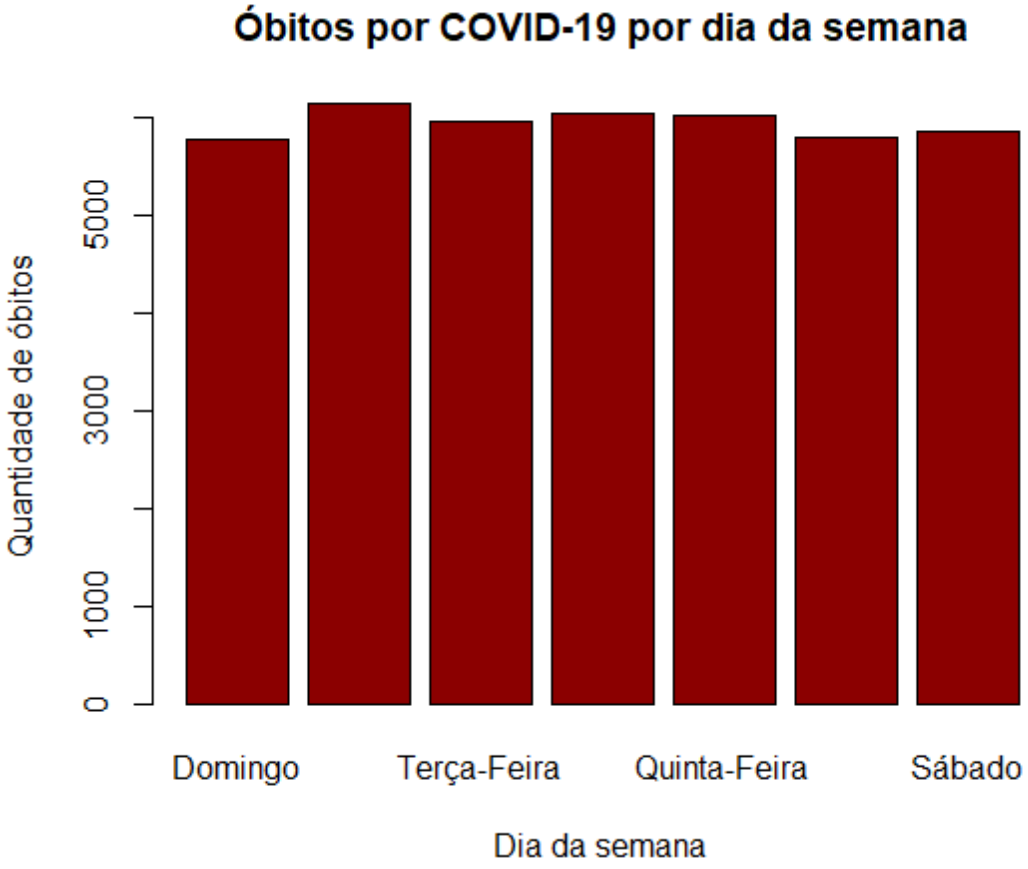

Figura 20. Óbitos por dia da semana

7.5. Casos por cidade

O RS tem 497 municípios. Ao gerar o gráfico de casos por cidade podemos perceber que poucas cidades tiveram a maior parte do número de casos. As 50 cidades com mais caso (10% das cidades) obtiveram 67% dos casos. (veja na figura 21)

8. Conclusão

Existem diversas maneiras de tratarmos e utilizarmos os dados. Neste trabalho a intenção foi utilizar algoritmos simples para analisar os dados da pandemia da COVID-19 no RS. Mais algoritmos poderiam ter sido usados, como redes neurais (para classificação), algoritmos de agrupamento (em uma análise geográfica das cidades e seus casos de COVID) entre outros. O foco neste trabalho em específico foi tentar explicar de maneira simples alguns algoritmos muito úteis. Mesmo com estes algoritmos que foram usados mais análises podem ser feitas. Mas estes outros algoritmos e outras análises ficam para trabalhos futuros.

Referências

[Assunção 2021] Assunção, J. V. C. (2021). *Uma breve introdução à Mineração de Dados. Bases para a ciência de dados, com exemplos em R.*, volume 1.

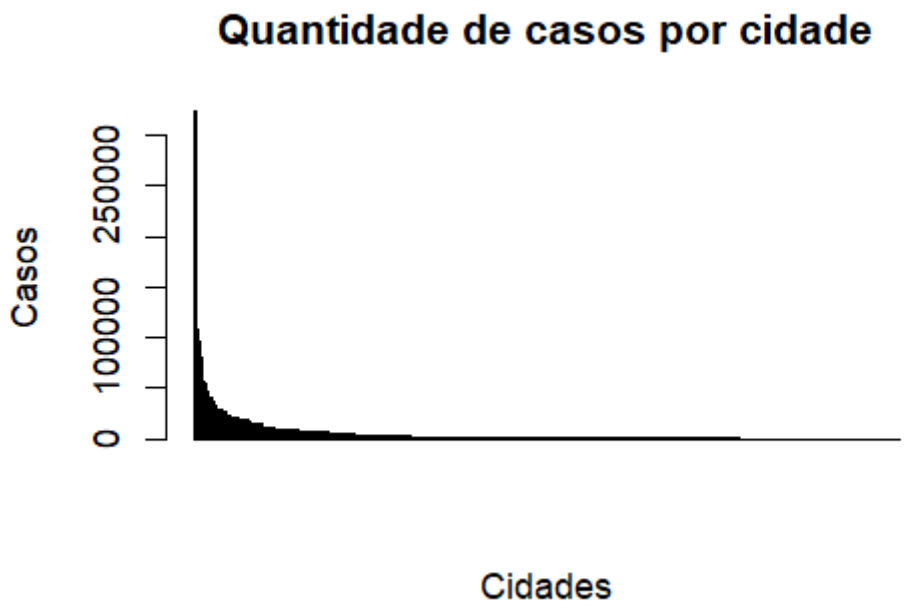

Figura 21. Quantidade de casos por cidade

[Ciotti et al. 2020] Ciotti, M., Ciccozzi, M., Terrinoni, A., Jiang, W.-C., Wang, C.-B., and Bernardini, S. (2020). The covid-19 pandemic. *Critical reviews in clinical laboratory sciences*, 57(6):365–388.

[Cucinotta and Vanelli 2020] Cucinotta, D. and Vanelli, M. (2020). Who declares covid-19 a pandemic. *Acta Bio Medica: Atenei Parmensis*, 91(1):157.

[da Silva et al. 2017] da Silva, L. A., Peres, S. M., and Boscarioli, C. (2017). *Introdução à mineração de dados: com aplicações em R*. Elsevier Brasil.

[Fayyad et al. 1996] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37.

[Han et al. 2022] Han, J., Pei, J., and Tong, H. (2022). *Data mining: concepts and techniques*. Morgan kaufmann.

[Nicola et al. 2020] Nicola, M., Alsafi, Z., Sohrabi, C., Kerwan, A., Al-Jabir, A., Iosifidis, C., Agha, M., and Agha, R. (2020). The socio-economic implications of the coronavirus pandemic (covid-19): A review. *International journal of surgery*, 78:185–193.

[Osofsky et al. 2020] Osofsky, J. D., Osofsky, H. J., and Mamon, L. Y. (2020). Psychological and social impact of covid-19. *Psychological Trauma: Theory, Research, Practice, and Policy*, 12(5):468.

[Tan et al. 2016] Tan, P.-N., Steinbach, M., and Kumar, V. (2016). *Introduction to data mining*. Pearson Education India.

[Therneau et al. 1997] Therneau, T. M., Atkinson, E. J., et al. (1997). An introduction to recursive partitioning using the rpart routines. Technical report, Technical report Mayo Foundation.

[WorldHealthOrganization] WorldHealthOrganization. Severe acute respiratory syndrome (sars). Acessado em 16 de janeiro de 2023.