

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

Anderson Portilho Lopes

**COMPARAÇÃO ENTRE MODELOS DE PREVISÃO DA EVOLUÇÃO DOS
PACIENTES COM SÍNDROME RESPIRATÓRIA AGUDA GRAVE (SRAG) POR
COVID-19.**

Belo Horizonte

2021

Anderson Portilho Lopes

**COMPARAÇÃO ENTRE MODELOS DE PREVISÃO DA EVOLUÇÃO DOS
PACIENTES COM SINDROME RESPIRATÓRIA AGUDA GRAVE (SRAG) POR
COVID-19.**

Trabalho de Conclusão de Curso apresentado
ao Curso de Especialização em Ciência de
Dados e Big Data como requisito parcial à
obtenção do título de especialista.

Belo Horizonte

2021

SUMÁRIO

1. Introdução	4
1.1. Contextualização	5
1.2. O problema proposto	5
1.3. Tecnologia Utilizada	6
2. Coleta de Dados.....	8
3. Processamento/Tratamento de Dados.....	15
3.1. Engenharia de Atributos	24
4. Análise e Exploração dos Dados	27
4.1. Balanceamento do dataset.	27
4.2. Variáveis demográficas (relacionadas ao paciente).....	28
4.3. Variáveis relacionadas a sintomas e sinais.	33
4.4. Variáveis relacionadas a doenças pré-existentes.	41
4.5. Variáveis relacionadas ao tratamento.	50
5. Criação de Modelos de Machine Learning	55
5.1. Padronização estatística dos valores numéricos.....	55
5.2. Seleção das variáveis de maior relevância aos modelos	56
5.3. Divisão dos dados e variáveis para treino e teste.....	59
5.4. Modelo Linear Generalizado - GLM.....	60
5.5. Modelo Naives Bayes.....	62
6. Apresentação dos Resultados	65
6.1. Modelo Linear Generalizado - GLM.....	67
6.2. Modelo Naive Bayes.....	71
6.3. Considerações finais dos resultados	74
7. Links	76

1. Introdução

O ano de 2020 foi inesperado e atípico em todos os continentes do planeta Terra, devido a epidemia de COVID-19.

O primeiro caso confirmado da doença causada pelo vírus SARS-Cov-2, foi identificado e registrado na cidade Wuhan, na China, no dia 31 de dezembro de 2019.

A partir de então, o vírus começou a se espalhar rapidamente pelo mundo: primeiramente pelo continente Asiático e depois para o Ocidente (o que inclui a maior parte da Europa, além de muitos países de origem colonial europeia nas Américas e na Oceania).

Em fevereiro de 2019, a transmissão da Covid-19, nome dado à doença infecciosa causada pelo SARS-Cov-2, no Irã e na Itália, deixou o mundo em alerta, visto que o número de pessoas infectadas cresceu rapidamente, lotando hospitais.

Ainda em fevereiro de 2019, o primeiro caso da doença foi registrado no Brasil, na cidade de São Paulo (SP). Já em março de 2019, a OMS (Organização Mundial de Saúde) declara estado de PANDEMIA, dado o aumento exponencial de infectados pelo mundo, com elevado número de mortos por complicações de saúde causadas pelo SARS-Cov-2.

Após um ano de Pandemia por COVID-19 declarada, o que se nota é que a doença afeta as pessoas de diferentes maneiras. A maioria dos infectados apresentará sintomas leves a moderados da doença e não precisarão ser hospitalizadas. Porém, um número menor de pessoas sofrerá consequências graves pela contaminação do vírus, em função das doenças respiratórias que inicialmente ele é capaz de gerar.

Infelizmente, no dia 11 de março de 2021, de acordo com Conselho Nacional de Secretários de Saúde, nosso país registrou recorde no número de mortes por complicações geradas pela Covid-19, chegando a 2.286 óbitos em apenas 24 horas. Nosso país registrou mais do que o dobro de mortes diárias pelo novo coronavírus do que a Ásia, o continente mais populoso do mundo e primeiro epicentro da doença.

Em números absolutos, atualmente, o Brasil mantém-se como o segundo país com mais mortes por covid-19 no mundo, atrás apenas dos Estados Unidos da América. Foram 270.656 desde o início da pandemia, de acordo com boletim do Conselho Nacional de Secretários da Saúde (Conass).

1.1.Contextualização

Tipicamente, define-se por coronavírus a “família” de vírus que causam, a princípio, doenças respiratórias.

A COVID-19 apresenta um espectro clínico variando de infecções assintomáticas a quadros graves. De acordo com a Organização Mundial de Saúde, a maioria (cerca de 80%) dos pacientes com COVID-19 podem ser assintomáticos ou ter poucos sintomas e, aproximadamente 20% dos casos detectados requer atendimento hospitalar por apresentarem dificuldade respiratória, dos quais aproximadamente 5% podem necessitar de suporte ventilatório.

Nestes casos, a doença apresenta complicações à saúde, tais como a doença grave do trato respiratório inferior sem causa clara, como é o caso de pacientes que se apresentem em Síndrome Respiratória Aguda Grave (SRAG).

Nesta síndrome, o indivíduo apresenta-se em franca dispneia/desconforto respiratório/dificuldade para respirar com saturação de oxigênio (O₂) menor do que 95% em ar ambiente ou coloração azulada dos lábios ou rosto (cianose) ou queixa de pressão persistente no tórax.

1.2.O problema proposto

Acompanhar os números diariamente divulgados na mídia, tais como a evolução número de casos, óbitos média móvel, entre outros, são de suma importância, pois tais informações auxiliam na tomada de decisões de política pública a fim de combater a disseminação da doença. Nesse sentido, muitos órgãos realizam uma análise da evolução da doença ao longo do tempo (análise descritiva e série temporal).

Julga-se ser de extrema importância enriquecer os estudos relativos a infecção por SARS-CoV-2, sob muitas óticas, produzindo conhecimentos que possam ser úteis a outros profissionais que, por ventura, não saibam produzir este conhecimento com dados, mas que possam se beneficiar destes.

Este estudo pretende, com os resultados de modelos preditivos, prever a evolução dos casos em cura ou óbito, dos pacientes com SRAG por Covid 19.

Acredita-se que o diferencial do nosso estudo está no fato de que os resultados dos nossos modelos preditivos serão com base na combinação 05 grupos com diversas informações, diretamente relacionada a doença, conforme segue:

1. **Demográfica (relacionadas ao paciente):** idade, sexo, escolaridade etc;
2. **Sintomas e sinais:** febre, vômitos, diarreia etc. Aqui utilizamos os dois conceitos pois algumas queixas do paciente (sintomas), podem ter sido percebidas pelos profissionais da saúde (sinais), como por exemplo a febre;
3. **Doenças pré-existentes - Grupo de risco:** diabetes, hipertensão, asma, gestantes, puérperas etc.
4. **Relacionadas ao tratamento da doença:** fatores como a ventilação não invasiva, necessidade de suporte respiratório via ventilação mecânica etc.
5. **Evolução dos casos:** dos casos identificados como de SRAG por covid 19, será analisado de que forma o paciente evoluiu (cura ou óbito).

1.3. Tecnologia Utilizada

Para o processamento dos dados, bem como para todo o desenvolvimento dos algoritmos preditivos, optamos por utilizar a linguagem R, que é gratuita e pode ser obtida em: <https://www.r-project.org/>.

A linguagem R não é considerada uma linguagem de programação por muitos profissionais da área de tecnologia. Talvez isso ocorra devido à grande especialização em estatística que o R possui. Entretanto, este fato, além de nos auxiliar a trabalhar com os dados, disponibiliza diversas bibliotecas que complementam suas qualidades, tornando ao nosso ver e conhecimento, a ferramenta adequada para o desenvolvimento dos nossos estudos.

Para uma melhor experiência e detalhamento deste estudo, vamos utilizar um ambiente de desenvolvimento integrado (integrated development environment-IDE) para a linguagem R, o programa RStudio, que poderá ser obtido em: <https://www.rstudio.com/products/rstudio/download/>.

A vantagem de utilizar uma IDE é que, além de tornar todo estudo mais visual, a IDE possui ferramentas para plotagem de gráficos, histórico dos resultados, área de console e de edição de código, permitindo na mesma tela verificar diversas informações.

Existem algumas variações do programa RStudio, edições comerciais que são pagas e a versão de código aberto que utilizaremos, a Desktop RStudio.

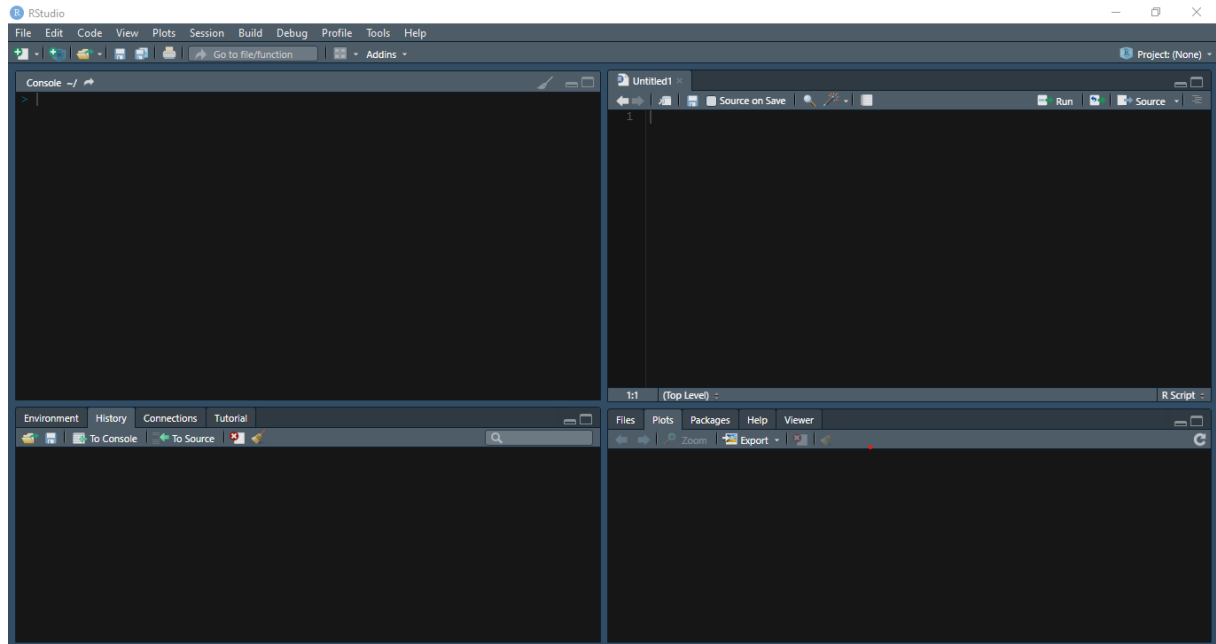


Figura 01 – Tela inicial do RStudio.

2. Coleta de Dados

O presente estudo se inicia com a obtenção do dataset relacionado a Síndrome Respiratória Aguda Grave (SRAG). Dentre as síndromes respiratórias tabuladas no conjunto de dados, existem diversos registros, o de nosso interesse: Covid-19.

Além do dataset, existem diversas outras informações relacionadas aos dados da SRAG, que são disponibilizadas pelo Governo Federal, no endereço eletrônico: <https://opendatasus.saude.gov.br/dataset?tags=SRAG>.

O dataset pode ser obtido no formato CSV, diretamente no link: <https://s3-sa-east-1.amazonaws.com/ckan.saude.gov.br/SRAG/2020/INFLUD-01-03-2021.csv>.

Devido ao tamanho de 614 MB, optamos por fazer o download do arquivo e iniciar nosso trabalho a partir deste, embora seja possível carregar os dados no ambiente do RStudio diretamente do link acima.

```
#-----
# Carregando o Dataset
#-----
existe_arquivo <- if(file.exists("INFLUD-01-03-2021.csv")) {
  "INFLUD-01-03-2021.csv"
} else {
  "https://s3-sa-east-1.amazonaws.com/ckan.saude.gov.br/SRAG/2020/INFLUD-01-03-2021.csv"
}
srag <- fread(existe_arquivo) # carrega os dados no formato data.table
```

Figura 02 – Abrindo / download do dataset e carregando suas informações para a variável srag.

Nas imagens que ilustram o código, as linhas que iniciam com “#”, tratam-se de linhas comentadas, identificando o comando, função e outras informações da linha imediatamente abaixo.

Trata-se de um arquivo com grande número de informações. Nele existem 154 colunas que a partir deste ponto, passamos a chamar de variáveis e 1.183.767 linhas (registros).

```
> # Verificar as dimensões do dataset
> dim(srag)
[1] 1183767    154
```

Figura 03 – Verificação da dimensão do dataset.

Na tabela abaixo, descrevemos o nome da variável, a descrição do que se trata a informação presente e o tipo de dado que as variáveis descrevem.

	Nome variável	Descrição	Tipo
1	DT_NOTIFIC	Data de preenchimento da ficha de notificação.	Date (DD/MM/AAAA)

2	SEM_NOT	Semana Epidemiológica do preenchimento da ficha de notificação	Varchar (12)
3	DT_SIN_PRI	Data de 1º sintomas do caso.	Date (DD/MM/AAAA)
4	SEM_PRI	Semana Epidemiológica do início dos sintomas.	Varchar (6)
5	SG_UF_NOT	Unidade Federativa onde está localizada a Unidade Sentinela que realizou a notificação.	Varchar (2)
6	ID_REGIONA	Regional de Saúde onde está localizado o Município realizou a notificação.	Varchar (6)
7	CO_REGIONA	Regional de Saúde onde está localizado o Município realizou a notificação.	Varchar (6)
8	ID_MUNICIP	Município onde está localizada a Unidade Sentinela que realizou a notificação.	Varchar (6)
9	CO_MUN_NOT	Município onde está localizada a Unidade Sentinela que realizou a notificação.	Varchar (6)
10	ID_UNIDADE	Unidade Sentinela que realizou o atendimento, coleta de amostra e registro do caso.	Varchar (7)
11	CO_UNI_NOT	Unidade Sentinela que realizou o atendimento, coleta de amostra e registro do caso.	Varchar (7)
12	CS_SEXO	Sexo do paciente.	Varchar (1)
13	DT_NASC	Data de nascimento do paciente.	Date (DD/MM/AAAA)
14	NU_IDADE_N	Idade informada pelo paciente quando não se sabe a data de nascimento. Na falta desse dado é registrada a idade aparente	Varchar (3)
15	TP_IDADE	Detalha o valor informado no campo NU_IDADE_N, sendo 1 – Dia, 2 – Mês e 3 - Ano	Varchar (1)
16	COD_IDADE	Detalha o valor informado no campo NU_IDADE_N, sendo 1 – Dia, 2 – Mês e 3 – Ano	Varchar (1)
17	CS_GESTANT	Idade gestacional da paciente.	Varchar (1)
18	CS_RACA	Cor ou raça declarada pelo paciente: Branca; Preta; Amarela; Parda (Pessoa que se declarou mulata, cabocla, cafuza, mameluca ou mestiça de preto com pessoa de outra cor ou raça); e, Indígena.	Varchar (2)
19	CS_ETINIA	Nome e código da etnia do paciente, quando indígena.	Varchar (4)
20	CS_ESCOL_N	Nível de escolaridade do paciente. Para os níveis fundamental e médio deve ser considerada a última série ou ano concluído.	Varchar (1)
21	ID_PAIS	País de residência do paciente.	Varchar (3)
22	CO_PAIS	País de residência do paciente.	Varchar (3)
23	SG_UF	Unidade Federativa de residência do paciente.	Varchar (2)
24	ID_RG_RESI	Regional de Saúde onde está localizado o Município de residência do paciente.	Varchar (6)
25	CO_RG_RESI	Regional de Saúde onde está localizado o Município de residência do paciente.	Varchar (6)
26	ID_MN_RESI	Município de residência do paciente.	Varchar (6)
27	CO_MUN_RES	Município de residência do paciente.	Varchar (6)
28	CS_ZONA	Zona geográfica do endereço de residência do paciente.	Varchar (1)
29	SURTO_SG	Caso é proveniente de surto de SG?	Varchar (1)
30	NOSOCOMIAL	Caso de SRAG com infecção adquirida após internação.	Varchar (1)
31	AVE_SUINO	Caso com contato direto com aves ou suínos.	Varchar (1)

32	FEBRE	Paciente apresentou febre?	Varchar (1)
33	TOSSE	Paciente apresentou tosse?	Varchar (1)
34	GARGANTA	Paciente apresentou dor de garganta?	Varchar (1)
35	DISPNEIA	Paciente apresentou dispneia?	Varchar (1)
36	DESC_RESP	Paciente apresentou desconforto respiratório?	Varchar (1)
37	SATURACAO	Paciente apresentou saturação O2 < 95%?	Varchar (1)
38	DIARREIA	Paciente apresentou diarreia?	Varchar (1)
39	VOMITO	Paciente apresentou vômito?	Varchar (1)
40	OUTRO_SIN	Paciente apresentou outro (s) sintoma (s)?	Varchar (1)
41	OUTRO_DES	Listar outros sinais e sintomas apresentados pelo paciente.	Varchar (30)
42	PUERPERA	Paciente é puérpera ou parturiente (mulher que pariu recentemente – até 45 dias do parto)?	Varchar (1)
43	FATOR_RISC	Paciente apresenta algum fator de risco	Varchar (1)
44	CARDIOPATI	Paciente possui Doença Cardiovascular Crônica?	Varchar (1)
45	HEMATOLOGI	Paciente possui Doença Hematológica Crônica?	Varchar (1)
46	SIND_DOWN	Paciente possui Síndrome de Down?	Varchar (1)
47	HEPATICA	Paciente possui Doença Hepática Crônica?	Varchar (1)
48	ASMA	Paciente possui Asma?	Varchar (1)
49	DIABETES	Paciente possui Diabetes <i>mellitus</i> ?	Varchar (1)
50	NEUROLOGIC	Paciente possui Doença Neurológica?	Varchar (1)
51	PNEUMOPATI	Paciente possui outra pneumopatia crônica?	Varchar (1)
52	IMUNODEPRE	Paciente possui Imunodeficiência ou Imunodepressão (diminuição da função do sistema imunológico)?	Varchar (1)
53	RENAL	Paciente possui Doença Renal Crônica?	Varchar (1)
54	OBESIDADE	Paciente possui obesidade?	Varchar (1)
55	OBES_IMC	Valor do IMC (Índice de Massa Corporal) do paciente calculado pelo profissional de saúde.	Varchar (3)
56	OUT_MORBI	Paciente possui outro (s) fator (es) de risco?	Varchar (1)
57	MORB_DESC	Listar outro (s) fator (es) de risco do paciente.	Varchar (30)
58	VACINA	Informar se o paciente foi vacinado contra gripe na última campanha, após verificar a documentação / caderneta. Caso o paciente não tenha a caderneta, direcionar a pergunta para ele ou responsável e preencher o campo com o código correspondente a resposta.	Varchar (1)
59	DT_UT_DOSE	Data da última dose de vacina contra gripe que o paciente tomou.	Date (DD/MM/AAAA)
60	MAE_VAC	Se paciente < 6 meses, a mãe recebeu vacina?	Varchar (1)
61	DT_VAC_MAE	Se a mãe recebeu vacina, qual a data?	Date (DD/MM/AAAA)
62	M_AMAMENTA	Se paciente < 6 meses, a mãe amamenta a criança?	Varchar (1)
63	DT_DOSEUNI	Se >= 6 meses e <= 8 anos, data da dose única para crianças vacinadas em campanhas de anos anteriores	Date (DD/MM/AAAA)
64	DT_1_DOSE	Se >= 6 meses e <= 8 anos, data da 1ª dose para crianças vacinadas pela primeira vez	Date (DD/MM/AAAA)
65	DT_2_DOSE	Se >= 6 meses e <= 8 anos data da 2ª dose para crianças vacinadas pela primeira vez	Date (DD/MM/AAAA)

66	ANTIVIRAL	Fez uso de antiviral para tratamento da doença?	Varchar (1)
67	TP_ANTIVIR	Qual antiviral utilizado?	Varchar (1)
68	OUT_ANTIV	Se o antiviral utilizado não foi Oseltamivir ou Zanamivir, informar qual antiviral foi utilizado.	Varchar (30)
69	DT_ANTIVIR	Data em que foi iniciado o tratamento com o antiviral.	Date (DD/MM/AAAA)
70	HOSPITAL	O paciente foi internado?	Varchar (1)
71	DT_INTERNA	Data em que o paciente foi hospitalizado.	Date (DD/MM/AAAA)
72	SG_UF_INTE	Unidade Federativa de internação do paciente.	Varchar (2)
73	ID_RG_INTE	Regional de Saúde onde está localizado o Município de internação do paciente.	Varchar (6)
74	CO_RG_INTE	Regional de Saúde onde está localizado o Município de internação do paciente.	Varchar (6)
75	ID_MN_INTE	Município onde está localizado a Unidade de Saúde onde o paciente internou.	Varchar (20)
76	CO_MU_INTE	Município onde está localizado a Unidade de Saúde onde o paciente internou.	Varchar (20)
77	UTI	O paciente foi internado em UTI?	Varchar (1)
78	DT_ENTUTI	Data de entrada do paciente na unidade de Terapia intensiva (UTI).	Date (DD/MM/AAAA)
79	DT_SAIDUTI	Data em que o paciente saiu da Unidade de Terapia intensiva (UTI).	Date (DD/MM/AAAA)
80	SUPPORT_VEN	O paciente fez uso de suporte ventilatório?	Varchar (1)
81	RAIOX_RES	Informar resultado de Raio X de Tórax.	Varchar (1)
82	RAIOX_OUT	Informar o resultado do RX de tórax se selecionado a opção 5-Outro.	Varchar (30)
83	DT_RAIOX	Se realizou RX de Tórax, especificar a data do exame.	Date (DD/MM/AAAA)
84	AMOSTRA	Foi realizado coleta de amostra para realização de teste diagnóstico?	Varchar (1)
85	DT_COLETA	Data da coleta da amostra para realização do teste diagnóstico.	Date (DD/MM/AAAA)
86	TP_AMOSTRA	Tipo da amostra clínica coletada para o teste diagnóstico.	Varchar (30)
87	OUT_AMOST	Descrição do tipo da amostra clínica, caso diferente das listadas nas categorias do campo.	Varchar (30)
88	PCR_RESUL	Resultado do teste de RT-PCR/outro método por Biologia Molecular.	Varchar (1)
89	DT_PCR	Data do Resultado RT-PCR/outro método por Biologia Molecular	Date (DD/MM/AAAA)
90	POS_PCRFLU	Resultado da RTPCR foi positivo para Influenza	Varchar (1)
91	TP_FLU_PCR	Resultado diagnóstico do RTPCR para o tipo de Influenza.	Varchar (1)
92	PCR_FLUASU	Subtipo para Influenza A.	Varchar (1)
93	FLUASU_OUT	Outro subtipo para Influenza A.	Varchar (30)
94	PCR_FLUBLI	Linhagem para Influenza B.	Varchar (1)
95	FLUBLI_OUT	Outra linhagem para Influenza B.	Varchar (30)
96	POS_PCROUT	Resultado da RTPCR foi positivo para outro vírus respiratório	Varchar (1)
97	PCR_VSR	Resultado diagnóstico do RTPCR para (VSR).	Varchar (1)
98	PCR_PARA1	Resultado diagnóstico do RTPCR para Parainfluenza 1.	Varchar (1)

99	PCR_PARA2	Resultado diagnóstico do RT-PCR para Parainfluenza 2.	Varchar (1)
100	PCR_PARA3	Resultado diagnóstico do RT-PCR para Parainfluenza 3.	Varchar (1)
101	PCR_PARA4	Resultado diagnóstico do RT-PCR para Parainfluenza 4.	Varchar (1)
102	PCR_ADENO	Resultado diagnóstico do RT-PCR para Adenovírus.	Varchar (1)
103	PCR_METAP	Resultado diagnóstico do RT-PCR para Metapneumovírus.	Varchar (1)
104	PCR_BOCA	Resultado diagnóstico do RTPCR para Bocavírus.	Varchar (1)
105	PCR_RINO	Resultado diagnóstico do RTPCR para Rinovírus.	Varchar (1)
106	PCR_OUTRO	Resultado diagnóstico do RTPCR para Outro vírus respiratório.	Varchar (1)
107	DS_PCR_OUT	Nome do outro vírus respiratório identificado pelo RT-PCR.	Varchar (30)
108	CLASSI_FIN	Diagnóstico final do caso. Se tiver resultados divergentes entre as metodologias laboratoriais, priorizar o resultado do RTPCR.	Varchar (1)
109	CLASSI_OUT	Descrição de qual outro agente etiológico foi identificado	Varchar (30)
110	CRITERIO	Indicar qual o critério de confirmação.	Varchar (1)
111	EVOLUCAO	Evolução do caso	Varchar (1)
112	DT_EVOLUCA	Data da alta ou óbito	Date (DD/MM/AAAA)
113	DT_ENCERRA	Data do encerramento do caso.	Date (DD/MM/AAAA)
114	DT_DIGITA	Data da digitação das informações	Date (DD/MM/AAAA)
115	HISTO_VGM	Histórico de viagem internacional até 14 dias antes do início dos sintomas	Varchar (1)
116	PAIS_VGM	País onde foi realizada a viagem	Varchar (3)
117	CO_PS_VGM	País onde foi realizada a viagem	Varchar (3)
118	LO_PS_VGM	Local (cidade, estado, província e outros) onde foi realizada a viagem	Varchar (30)
119	DT_VGM	Data em que foi realizada a viagem	Date (DD/MM/AAAA)
120	DT_RT_VGM	Data em que retornou de viagem	Date (DD/MM/AAAA)
121	PCR_SARS2	Resultado diagnóstico do RTPCR para (SARS-CoV-2).	Varchar (1)
122	PAC_COCBO	Código CBO da ocupação do paciente	Varchar (6)
123	PAC_DSCBO	Descrição da ocupação profissional do paciente	Varchar (6)
124	OUT_ANIM	Informar o animal que o paciente teve contato se selecionado a opção 3.	Varchar (60)
125	DOR_ABD	Paciente apresentou dor abdominal?	Varchar (1)
126	FADIGA	Paciente apresentou fadiga?	Varchar (1)
127	PERD_OLFT	Paciente apresentou perda do olfato?	Varchar (1)
128	PERD_PALA	Paciente apresentou perda do paladar?	Varchar (1)
129	TOMO_RES	Informar o resultado da tomografia.	Number(3)
130	TOMO_OUT	Informar o resultado da tomografia se selecionado a opção 5-Outro	Varchar (100)
131	DT_TOMO	Se realizou tomografia, especificar a data do exame.	Date (DD/MM/AAAA)
132	TP_TES_AN	Tipo do teste antigênico que foi realizado.	Number(3)
133	DT_RES_AN	Data do resultado do teste antigênico.	Date (DD/MM/AAAA)
134	RES_AN	Resultado do Teste Antigênico	Varchar (1)
135	POS_AN_FLU	Resultado do Teste Antigênico que foi positivo para Influenza	Varchar (1)

136	TP_FLU_AN	Resultado do Teste Antigênico, para o tipo de Influenza.	Varchar (1)
137	POS_AN_OUT	Resultado do Teste Antigênico, que foi positivo para outro vírus respiratório.	Varchar (1)
138	AN_SARS2	Resultado do Teste Antigênico, para SARS-CoV-2.	Varchar (1)
139	AN_VSR	Resultado do Teste Antigênico, para VSR.	Varchar (1)
140	AN_PARA1	Resultado do Teste Antigênico, Parainfluenza 1.	Varchar (1)
141	AN_PARA2	Resultado do Teste Antigênico. Parainfluenza 2.	Varchar (1)
142	AN_PARA3	Resultado do Teste Antigênico. Parainfluenza 3.	Varchar (1)
143	AN_ADENO	Resultado do Teste Antigênico. Adenovírus.	Varchar (1)
144	AN_OUTRO	Resultado do Teste Antigênico. Outro vírus respiratório.	Varchar (1)
145	DS_AN_OUT	Nome do outro vírus respiratório identificado pelo Teste Antigênico.	Varchar (30)
146	TP_AM_SOR	Tipo de amostra sorológica que foi coletada.	Number(3)
147	SOR_OUT	Descrição do tipo da amostra clínica, caso diferente das listadas na categoria um (1) do campo.	Varchar (30)
148	DT_CO_SOR	Data da coleta do material para diagnóstico por Sorologia.	Data (DD/MM/AAAA)
149	TP_SOR	Tipo do Teste Sorológico que foi realizado	Number(3)
150	OUT_SOR	Descrição do tipo de Teste Sorológico.	Varchar 2(100)
151	DT_RES	Data do Resultado do Teste Sorológico	Date (DD/MM/AAAA)
152	RES_IGG	Resultado da Sorologia para SARS-CoV-2	Varchar (1)
153	RES_IGM	Resultado da Sorologia para SARS-CoV-2	Varchar (1)
154	RES_IGA	Resultado da Sorologia para SARS-CoV-2	Varchar (1)

Além do dataset, na mesma página onde ele foi obtido, está disponível um arquivo chamado “dicionario-de-dados-srag-hospitalizado-27.07.2020-final.pdf”. Este arquivo nos auxiliou a entender os valores presentes em cada variável, permitindo o adequado tratamento dos valores em nossos códigos. Abaixo segue o link do dicionário de dados: <https://opendatasus.saude.gov.br/dataset/ae90fa8f-3e94-467e-a33f-94adbb66edf8/resource/8f571374-c555-4ec0-8e44-00b1e8b11c25/download/dicionario-de-dados-srag-hospitalizado-27.07.2020-final.pdf>.

Complementa a identificação, origem e qualificação dos dados que utilizaremos em nossos estudos as informações a seguir:

Campo	Valor
Autor	Datasus
Última Atualização	3 de Março de 2021, 19:00 (UTC-03:00)
Criado	22 de Julho de 2020, 11:54 (UTC-03:00)
Cobertura Geográfica	Nacional

Cobertura Temporal	A partir da Semana Epidemiológica 01 de cada ano.
Contato	dadosabertos@saude.gov.br
Frequência Atualização	Semanal
Granularidade Geográfica	Município
Granularidade temporal	Dia
Referências	Classificação Brasileira de Ocupações (MTE), Tabela de Código de Municípios (IBGE), Cadastro Nacional de Estabelecimentos de Saúde (CNES).
Área Responsável	Coordenação-Geral do Programa Nacional de Imunizações (CGPNI/DEIDT/SVS/MS)

Como nosso objetivo é prever a evolução da doença, a partir de informações tais como de sintomas, comorbidade e características do tratamento, este foi o único dataset compatível com este tipo de informação, evidenciando a riqueza dos dados que estudaremos. Desta forma, não foi possível utilizar outro dataset.

Além disso, o dataset é estruturado pelo autor e nos fornece, de forma ampla e diversificada, todas as informações necessárias para esta pesquisa. A abundância das variáveis disponíveis somadas à quantidade acima de 1 milhão de registros, nos proporciona infinitas possibilidades de análises para a realização de treinos e testes, neste presente estudo.

3. Processamento/Tratamento de Dados

Para o correto processamento e tratamento dos dados, visualizamos as informações pertinentes das variáveis, tais como os tipos atribuídos pela linguagem R no carregamento, além de uma amostra dos seus valores.

```
> # visualização dos tipos atribuídos as variáveis pelo R no carregamento do dataset
> str(srag)
Classes 'data.table' and 'data.frame': 1183767 obs. of 154 variables:
 $ DT_NOTIFIC: chr "24/01/2020" "28/02/2020" "15/03/2020" "29/02/2020" ...
 $ SEM_NOT : int 4 9 12 9 12 12 12 12 13 13 ...
 $ DT_SIN_PRI: chr "22/01/2020" "25/02/2020" "11/03/2020" "22/02/2020" ...
 $ SEM_PRI : int 4 9 11 8 12 12 12 12 12 13 ...
 $ SG_UF_NOT : chr "AM" "AM" "ES" "MA" ...
 $ ID_REGIONA: chr "ENTORNO DE MANAUS E RIO NEGRO" "ENTORNO DE MANAUS E RIO NEGRO" "COLATINA" "REGIÃO
L DE SAUDE METROPOLITANA" ...
 $ CO_REGIONA: int 5584 5584 1509 1430 1331 1331 1331 1331 1331 1570 ...
 $ ID_MUNICIP: chr "MANAUS" "MANAUS" "COLATINA" "SAO LUIS" ...
 $ CO_MUN_NOT: int 130260 130260 320150 211130 355030 355030 355030 355030 355030 421360 ...
 $ ID_UNIDADE: chr "HOSPITAL E P S DA CRIANÇA DA ZONA OESTE" "HOSPITAL E PRONTO SOCORRO DA ZONA NORTE"
```

Figura 04 – Verificação das 10 primeiras variáveis, seus tipos e amostra dos valores encontrados.

Na sequência, visualizamos a quantidade de valores ausentes, também conhecidos como “missing” ou “NA”.

```
> # verificando valores missing (NA) nas variáveis do dataset
> sapply(srag, function(x)sum(is.na(x)))
DT_NOTIFIC      SEM_NOT DT_SIN_PRI      SEM_PRI SG_UF_NOT ID_REGIONA CO_REGIONA ID_MUNICIP CO_MUN_NOT
0              0          0          0          0          0      155790          0          0
ID_UNIDADE CO_UNI_NOT CS_SEXO DT_NASC NU_IDADE_N TP_IDADE COD_IDADE CS_GESTANT CS_RACA
0              0          0          0          0          0          0          0      54621
CS_ETINIA CS_ESCOL_N ID_PAIS CO_PAIS SG_UF ID_RG_RESI CO_RG_RESI ID_MN_RESI CO_MUN_RES
0      358687          0          0          0          0      153494          0          113
CS_ZONA SURTO_SG NOSOCOMIAL AVE_SUINO FEBRE TOSSE GARGANTA DISPNEIA DESC_RESP
128812 192743 252228 214844 156432 131871 293115 133022 194540
SATURACAO DIARREIA VOMITO OUTRO_SIN OUTRO_DES PUERPERA FATOR_RISC CARDIOPATI HEMATOLOGI
188347 305755 316878 308143 0 708116 0 571488 704011
SIND_DOWN HEPATICA ASMA DIABETES NEUROLOGIC PNEUMOPATI IMUNODEPRE RENAL OBESIDADE
706090 705152 694938 611753 690809 690422 698671 694649 693467
OBES_IMC OUT_MORBI MORB_DESC VACINA DT_UT_DOSE MAE_VAC DT_VAC_MAE M_AMAMENTA DT_DOSEUNI
0 620492 0 250650 0 1175048 0 1177290 0
DT_1_DOSE DT_2_DOSE ANTIVIRAL TP_ANTIVIR OUT_ANTIV DT_ANTIVIR HOSPITAL DT_INTERNA SG_UF_INTE
0 0 165875 1012260 0 0 30915 0 0
ID_RG_INTE CO_RG_INTE ID_MN_INTE CO_MU_INTE UTI DT_ENTUTI DT_SAIDUTI SUPORT_VEN RAIOS_RES
0 282895 0 75857 170147 0 0 162153 403912
RAIOS_OUT DT_RAIOX AMOSTRA DT_COLETA TP_AMOSTRA OUT_AMOST PCR_RESUL DT_PCR POS_PCRFLU
0 0 37017 0 106626 0 92756 0 892272
TP_FLU_PCR PCR_FLUASU FLUASU_OUT PCR_FLUBLI FLUBLI_OUT POS_PCROUT PCR_VSR PCR_PARA1 PCR_PARA2
1181769 1182417 0 1183400 0 691676 1182592 1183683 1183744
PCR_PARA3 PCR_PARA4 PCR_ADENO PCR_METAP PCR_BOCA PCR_RINO PCR_OUTRO DS_PCR_OUT CLASSI_FIN
1183658 1183744 1183331 1183337 1183721 1181417 1180296 0 82616
CLASSI_OUT CRITERIO EVOLUCAO DT_EVOLUCA DT_ENCERRA DT_DIGITA HISTO_VGM PAIS_VGM CO_PS_VGM
0 115314 161837 0 0 0 0 0 1182436
```

Figura 05 – Verificação dos valores ausentes (missing ou NA).

Assim, após uma extensa análise dos dados, definição dos objetivos gerais deste estudo e análise das variáveis e seus respectivos valores, encontrados ou ausentes, passamos ao tratamento dos dados, modelando-os da melhor forma para atingir nosso objetivo, com os melhores resultados.

Iniciamos definindo que nossa variável target, que será utilizada para medir os acertos dos nossos modelos de predição é a EVOLUÇÃO. De acordo com o

dicionário de dados, os valores de nosso interesse para esta variável são: 1. Se refere aos casos onde os pacientes evoluíram para cura e 2. Se refere aos casos onde, infelizmente, o paciente evoluiu para óbito.

Definimos também alguns pressupostos que delimitam nosso universo do estudo, conforme:

1. Com relação a dimensão geográfica, definimos que serão analisados somente os dados referenciados ao estado de São Paulo (é um estado representativo em números e diversidade do povo brasileiro). Então, aplicamos um filtro na variável SG_UF_NOT == "SP".
2. Com relação ao tipo de SRAG que vamos estudar, conforme já definido, serão somente aqueles relacionados a COVID 19. Então, aplicamos um filtro na variável CLASSI_FIN == "5".

```
# Criação de arrays com aplicação de filtros (pressupostos)
srag_sp <- subset(srag, SG_UF_NOT == "SP")
srag_sp_covid <- subset(srag_sp, CLASSI_FIN == "5")
srag_sp_covid_v1 <- filter(srag_sp_covid, EVOLUCAO %in% c("1", "2"))
```

Figura 06 – Tratamento da variável target e dos pressupostos.

Na sequência, excluimos 110 variáveis que julgamos não possuir relação direta com nosso objetivo ou, também, por foram utilizadas na filtragem, por exemplo: a variável SU_UF_NOT, utilizada no filtro, demonstrou que somente os dados do estado de SP estão presentes no array, então pôde ser descartada por já ter contribuído com o objetivo. As Variáveis excluídas foram: 1-COD_IDADE, 2-DT_NOTIFIC, 3-SEM_NOT, 4-SEM_PRI, 5-ID_MUNICIP, 6-SG_UF_NOT, 7-CO_MUN_NOT, 8-ID_REGIONA, 9-CO_REGIONA, 10-ID_UNIDADE, 11-CO_UNI_NOT, 12-NU_IDADE_N, 13-TP_IDADE, 14-CS_GESTANT, 15-CS_ETINIA, 16-ID_PAIS, 17-CO_PAIS, 18-SG_UF, 19-ID_RG_RESI, 20-CO_RG_RESI, 21-ID_MN_RESI, 22-CO_MUN_RES, 23-CS_ZONA, 24-HISTO_VGM, 25-PAIS_VGM, 26-CO_PS_VGM, 27-LO_PS_VGM, 28-DT_VGM, 29-DT_RT_VGM, 30-SURTO_SG, 31-NOSOCOMIAL, 32-AVE_SUINO, 33-OUT_ANIM, 34-OUTRO_DES, 35-OBES_IMC, 36-MORB_DESC, 37-VACINA, 38-DT_UT_DOSE, 39-MAE_VAC, 40-DT_VAC_MAE, 41-M_AMAMENTA, 42-DT_DOSEUNI, 43-DT_1_DOSE, 44-DT_2_DOSE, 45-TP_ANTIVIR, 46-OUT_ANTIV, 47-DT_ANTIVIR, 48-SG_UF_INTE, 49-ID_RG_INTE, 50-CO_RG_INTE, 51-ID_MN_INTE, 52-CO_MU_INTE, 53-RAIOX_RES, 54-RAIOX_OUT, 55-DT_RAIOX, 56-TOMO_OUT, 57-DT_TOMO, 58-

AMOSTRA, 59-DT_COLETA, 60-TP_AMOSTRA, 61-OUT_AMOST, 62-TP_TES_AN, 63-DT_RES_AN, 64-POS_AN_FLU, 65-TP_FLU_AN, 66-POS_AN_OUT, 67-AN_SARS2, 68-AN_VSR, 69-AN_PARA1, 70-AN_PARA2, 71-AN_PARA3, 72-AN_ADENO, 73-AN_OUTRO, 74-DS_AN_OUT, 75-PCR_RESUL, 76-DT_PCR, 77-POS_PCRFLU, 78-TP_FLU_PCR, 79-PCR_FLUASU, 80-FLUASU_OUT, 81-PCR_FLUBLI, 82-FLUBLI_OUT, 83-POS_PCROUT, 84-PCR_VSR, 85-PCR_PARA1, 86-PCR_PARA2, 87-PCR_PARA3, 88-PCR_PARA4, 89-PCR_ADENO, 90-PCR_METAP, 91-PCR_BOCA, 92-PCR_RINO, 93-PCR_OUTRO, 94-DS_PCR_OUT, 95-TP_AM_SOR, 96-DT_CO_SOR, 97-TP_SOR, 98-OUT_SOR, 99-SOR_OUT, 100-RES_IGG, 101-RES_IGM, 102-RES_IGA, 103-DT_RES, 104-CLASSI_FIN, 105-CLASSI_OUT, 106-CRITERIO, 107-DT_ENCERRA, 108-DT_DIGITA, 109-PAC_COCBO, 110-PAC_DSCBO.

```
# Excluindo variáveis que não possuem relação com objetivo do problema.
srag_sp_covid_v1$COD_IDADE = NULL
srag_sp_covid_v1$DT_NOTIFIC = NULL
srag_sp_covid_v1$SEM_NOT = NULL
srag_sp_covid_v1$SEM_PRI = NULL
srag_sp_covid_v1$ID_MUNICIP = NULL
srag_sp_covid_v1$SG_UF_NOT = NULL
srag_sp_covid_v1$CO_MUN_NOT = NULL
srag_sp_covid_v1$ID_REGIONA = NULL
srag_sp_covid_v1$CO_REGIONA = NULL
srag_sp_covid_v1$ID_UNIDADE = NULL
srag_sp_covid_v1$CO_UNI_NOT = NULL
srag_sp_covid_v1$NU_IDADE_N = NULL
```

Figura 07 – Exclusão das variáveis que não possuem relação direta com nosso objetivo.

Na sequência, gravamos as informações tratadas em um novo arquivo, preservando assim o dataset original. Carregamos o novo arquivo, eliminamos a variável criada pela linguagem R quando da criação do novo arquivo e conferimos suas novas dimensões.

```
# Criação do novo dataset com as variáveis e valores selecionados
write.csv(srag_sp_covid_v1, "srag_sp_covid_v1.csv")

# Carregando o novo dataset
srag_sp_v1 <- fread("srag_sp_covid_v1.csv")

# eliminando variável criada pelo R ao salvar novo arquivo
srag_sp_v1$v1 <- NULL
```

Figura 08 – Carregamento do arquivo com os dados relacionados ao objetivo do estudo.

Verificamos que a dimensão do nosso novo dataset é de 44 variáveis com 187.085 registros.

```
> # Verificando a nova dimensão e nome das variáveis
    relacionadas ao objetivo do estudo
> dim(srag_sp_v1)
[1] 187085    44
```

Figura 09 – Dimensão do novo dataset.

A seguir, realizou-se o tratamento dos dados, tanto dos valores ausentes quanto os que se apresentaram fora dos padrões. Identificamos que existem três grupos que precisam de tratamentos diferenciados, de acordo com os tipos de informações que recebem como valores.

O primeiro grupo é aquele que recebe valores categóricos, onde cada valor tem um significado previamente definido por seu autor. Consultando o dicionário de dados, identificamos todas as variáveis que recebem os valores conforme:

- 1 – Sim;
- 2 – Não;
- 9 – Ignorado;

Assim, decidimos que para este grupo, faremos os tratamentos dos valores ausentes preenchendo-os com o valor 9. Desta forma, manteremos o maior número de registros para análise, sem ocasionar distorções nos resultados finais dos nossos modelos.

Serão tratadas desta forma as variáveis: CS_RACA, CS_ESCOL_N, FEBRE, TOSSE, GARGANTA, DISPNEIA, DESC_RESP, SATURACAO, DIARREIA, VOMITO, OUTRO_SIN, PUERPERA, CARDIOPATI, HEMATOLOGI, SIND_DOWN, HEPATICA, ASMA, DIABETES, NEUROLOGIC, PNEUMOPATI, IMUNODEPRE, RENAL, OBESIDADE, OUT_MORBI, ANTIVIRAL, HOSPITAL, UTI, SUPORT_VEN, PCR_SARS2, DOR_ABD, FADIGA, PERD_OLFT, PERD_PALA, TOMO_RES, RES_AN

```
srag_sp_v1$CS_RACA <- coalesce(srag_sp_v1$CS_RACA,9)
srag_sp_v1$CS_ESCOL_N <- coalesce(srag_sp_v1$CS_ESCOL_N,9)
srag_sp_v1$FEBRE <- coalesce(srag_sp_v1$FEBRE,9)
srag_sp_v1$TOSSE <- coalesce(srag_sp_v1$TOSSE,9)
srag_sp_v1$GARGANTA <- coalesce(srag_sp_v1$GARGANTA,9)
srag_sp_v1$DISPNEIA <- coalesce(srag_sp_v1$DISPNEIA,9)
srag_sp_v1$DESC_RESP <- coalesce(srag_sp_v1$DESC_RESP,9)
srag_sp_v1$SATURACAO <- coalesce(srag_sp_v1$SATURACAO,9)
srag_sp_v1$DIARREIA <- coalesce(srag_sp_v1$DIARREIA,9)
srag_sp_v1$VOMITO <- coalesce(srag_sp_v1$VOMITO,9)
srag_sp_v1$OUTRO_SIN <- coalesce(srag_sp_v1$OUTRO_SIN,9)
srag_sp_v1$PUERPERA <- coalesce(srag_sp_v1$PUERPERA,9)
srag_sp_v1$CARDIOPATI <- coalesce(srag_sp_v1$CARDIOPATI,9)
srag_sp_v1$HEMATOLOGI <- coalesce(srag_sp_v1$HEMATOLOGI,9)
```

Figura 10 – Tratando valores missing numéricos

As variáveis do segundo grupo são aquelas receberam algum caractere de texto em seus valores. Para este grupo, não identificamos valores ausentes, portanto, caberá somente sua padronização. As duas variáveis com esta característica, são: CS_SEXO e FATOR_RISC.

```
> str(srag_sp_v1$CS_SEXO)
chr [1:187085] "M" "F" "F" "F" "M" "F" "M" "F" "F" "M" ...
> str(srag_sp_v1$FATOR_RISC)
chr [1:187085] "S" "S" "S" "N" "S" "N" "S" "N" "S" "N" ...
```

Figura 11 – Variáveis que receberam texto em seus valores.

Embora o dicionário de dados indique que estas variáveis recebam os mesmos valores padrão, assim como as tratadas anteriormente, com ligeira variação somente na interpretação da variável CS_SEXO, verificamos que de fato, os valores contidos são de texto. Então, procedemos com o tratamento conforme comandos:

```
# atribuindo valores na variável sexo conforme equivalente no dicionário
srag_sp_v1$CS_SEXO[srag_sp_v1$CS_SEXO == "M"] <- 1
srag_sp_v1$CS_SEXO[srag_sp_v1$CS_SEXO == "F"] <- 2
srag_sp_v1$CS_SEXO[srag_sp_v1$CS_SEXO == "I"] <- 9

# atribuindo valores na variável fator de risco conforme equivalente no dicionário
srag_sp_v1$FATOR_RISC[srag_sp_v1$FATOR_RISC == "S"] <- 1
srag_sp_v1$FATOR_RISC[srag_sp_v1$FATOR_RISC == "N"] <- 2
```

Figura 12 – Atribuindo valores nas variáveis que receberam texto, conforme equivalente no dicionário.

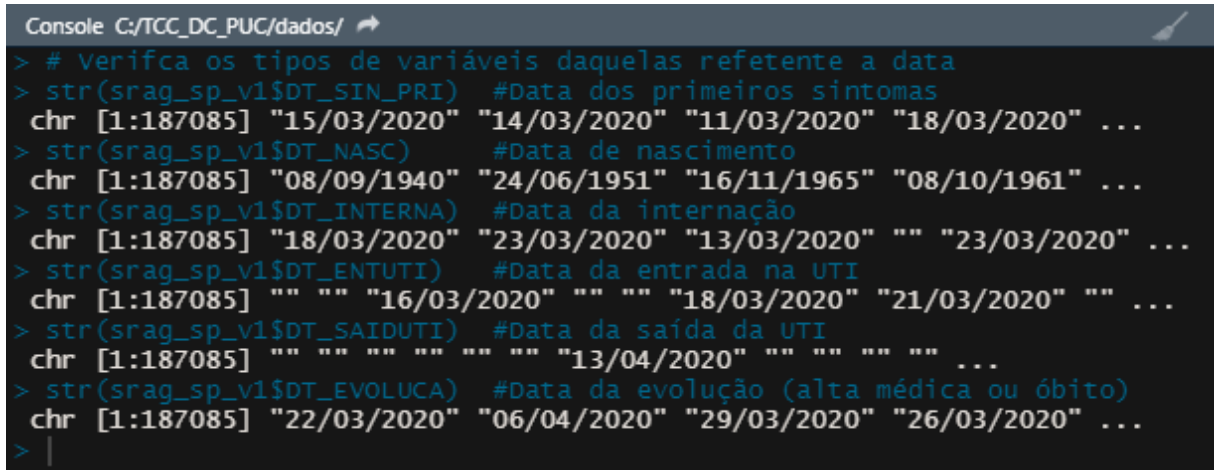
Os dois grupos tratados anteriormente, são compostos por variáveis categóricas pois, apresentam um número limitado dos tipos de respostas em seus valores. Em termos estatísticos e também na linguagem R, existe um recurso chamado fator (factor). Neste recurso armazenamos as variáveis categóricas, facilitando sua exploração.

Então, para o melhor entendimento do comportamento destas variáveis, vamos em cada uma dela substituir os valores numéricos por seus significados e converte-las para fator.

```
srag_sp_v1$HEPATICA = factor(srag_sp_v1$HEPATICA, labels = c("Sim", "Não", "Ignorado"))
srag_sp_v1$ASMA = factor(srag_sp_v1$ASMA, labels = c("Sim", "Não", "Ignorado"))
srag_sp_v1$DIABETES = factor(srag_sp_v1$DIABETES, labels = c("Sim", "Não", "Ignorado"))
srag_sp_v1$NEUROLOGIC = factor(srag_sp_v1$NEUROLOGIC, labels = c("Sim", "Não", "Ignorado"))
srag_sp_v1$PNEUMOPATI = factor(srag_sp_v1$PNEUMOPATI, labels = c("Sim", "Não", "Ignorado"))
srag_sp_v1$IMUNODEPRE = factor(srag_sp_v1$IMUNODEPRE, labels = c("Sim", "Não", "Ignorado"))
srag_sp_v1$RENAL = factor(srag_sp_v1$RENAL, labels = c("Sim", "Não", "Ignorado"))
srag_sp_v1$OBESIDADE = factor(srag_sp_v1$OBESIDADE, labels = c("Sim", "Não", "Ignorado"))
srag_sp_v1$OUT_MORBI = factor(srag_sp_v1$OUT_MORBI, labels = c("Sim", "Não", "Ignorado"))
srag_sp_v1$ANTIVIRAL = factor(srag_sp_v1$ANTIVIRAL, labels = c("Sim", "Não", "Ignorado"))
srag_sp_v1$HOSPITAL = factor(srag_sp_v1$HOSPITAL, labels = c("Sim", "Não", "Ignorado"))
srag_sp_v1$UTI = factor(srag_sp_v1$UTI, labels = c("Sim", "Não", "Ignorado"))
srag_sp_v1$SUPPORT_VEN = factor(srag_sp_v1$SUPPORT_VEN, labels = c("Sim, invasivo", "Sim, não invasivo"))
srag_sp_v1$EVOLUCAO = factor(srag_sp_v1$EVOLUCAO, labels = c("Cura", "Óbito"))
srag_sp_v1$PCR_SARS2 = factor(srag_sp_v1$PCR_SARS2, labels = c("marcado pelo usuário", "Não marcado"))
srag_sp_v1$TOMO_RES = factor(srag_sp_v1$TOMO_RES, labels = c("Típico COVID-19", "Indeterminado COVID-19"))
srag_sp_v1$RES_AN = factor(srag_sp_v1$RES_AN, labels = c("positivo", "Negativo", "Inconclusivo", "Não"))
```

Figura 13 – Substituindo os valores por seus respectivos significados e transformando para fator.

O terceiro grupo de dados que identificamos são aqueles que recebem um valor referente a uma data. Estes campos encontram-se com seus valores no formato Brasileiro e, devido a este fato, a linguagem R quando carregou o dataset, reconheceu os valores como texto. Isso impossibilita de criamos as variáveis numéricas que necessitamos aos nossos estudos.



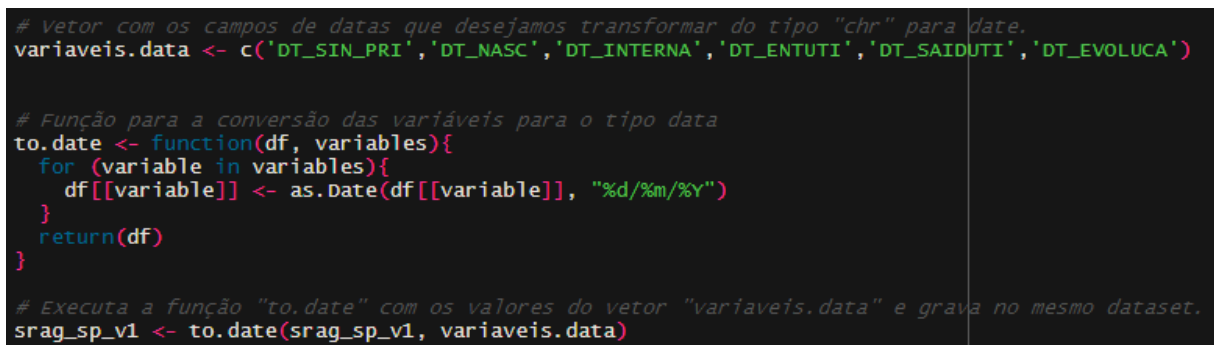
```

Console C:/TCC_DC_PUC/dados/ ➔
> # Verifica os tipos de variáveis daquelas referentes a data
> str(srag_sp_v1$DT_SIN_PRI) #Data dos primeiros sintomas
chr [1:187085] "15/03/2020" "14/03/2020" "11/03/2020" "18/03/2020" ...
> str(srag_sp_v1$DT_NASC) #Data de nascimento
chr [1:187085] "08/09/1940" "24/06/1951" "16/11/1965" "08/10/1961" ...
> str(srag_sp_v1$DT_INTERNA) #Data da internação
chr [1:187085] "18/03/2020" "23/03/2020" "13/03/2020" "" "23/03/2020" ...
> str(srag_sp_v1$DT_ENTUTI) #Data da entrada na UTI
chr [1:187085] "" "" "16/03/2020" "" "" "18/03/2020" "21/03/2020" "" ...
> str(srag_sp_v1$DT_SAIDUTI) #Data da saída da UTI
chr [1:187085] "" "" "" "" "" "" "13/04/2020" "" "" "" "" ...
> str(srag_sp_v1$DT_EVOLUCA) #Data da evolução (alta médica ou óbito)
chr [1:187085] "22/03/2020" "06/04/2020" "29/03/2020" "26/03/2020" ...
> |

```

Figura 14 – Verificação dos tipos de variáveis dos campos que se referem a uma data.

Criamos um Array com o nome das variáveis que desejamos ajustar seu tipo para uma data. Na sequência, utilizamos este Array em uma função, que percorrerá cada variável e definirá seu tipo para date. Por fim, executamos a função e colocamos os tipos ajustados sobrescrevendo as informações anteriores. A figura a seguir ilustra os três passos descritos.



```

# Vetor com os campos de datas que desejamos transformar do tipo "chr" para date.
variaveis.data <- c('DT_SIN_PRI', 'DT_NASC', 'DT_INTERNA', 'DT_ENTUTI', 'DT_SAIDUTI', 'DT_EVOLUCA')

# Função para a conversão das variáveis para o tipo data
to.date <- function(df, variables){
  for (variable in variables){
    df[[variable]] <- as.Date(df[[variable]], "%d/%m/%Y")
  }
  return(df)
}

# Executa a função "to.date" com os valores do vetor "variaveis.data" e grava no mesmo dataset.
srag_sp_v1 <- to.date(srag_sp_v1, variaveis.data)

```

Figura 15 – Função para o tratamento dos campos que se referem a uma data.

Após a execução da função verificamos novamente as variáveis de data. Notamos que a função foi executada com sucesso e também que existem valores ausentes nas datas.

```

> str(srag_sp_v1$DT_SIN_PRI) #Data dos primeiros sintomas
Date[1:187085], format: "2020-03-15" "2020-03-14" "2020-03-11" "2020-03-18"
> str(srag_sp_v1$DT_NASC) #Data de nascimento
Date[1:187085], format: "1940-09-08" "1951-06-24" "1965-11-16" "1961-10-08"
> str(srag_sp_v1$DT_INTERNA) #Data da internação
Date[1:187085], format: "2020-03-18" "2020-03-23" "2020-03-13" NA "2020-03-2
> str(srag_sp_v1$DT_ENTUTI) #Data da entrada na UTI
Date[1:187085], format: NA NA "2020-03-16" NA NA "2020-03-18" "2020-03-21" NA
> str(srag_sp_v1$DT_SAIDUTI) #Data da saída da UTI
Date[1:187085], format: NA NA NA NA NA NA "2020-04-13" NA NA NA NA "2020-04-
> str(srag_sp_v1$DT_EVOLUCA) #Data da evolução (alta médica ou óbito)
Date[1:187085], format: "2020-03-22" "2020-04-06" "2020-03-29" "2020-03-26"

```

Figura 16 – Verificação dos tipos de variáveis dos campos que se referem a uma data pós conversão.

Antes tais variáveis de data não apresentavam valores ausentes pois, estavam classificadas como caracteres, onde um espaço em branco, também é caractere. Isso nos mostra a importância do tratamento dos dados antes de efetivamente fazer uso deste nos modelos. Vejamos os valores ausentes.

```

> sapply(srag_sp_v1, function(x)sum(is.na(x)))
DT_SIN_PRI      CS_SEXO      DT_NASC      CS_RACA      CS_ESCOL_N      FEBRE
0              0          100           0           0              0
TOSSE          GARGANTA      DISPNEIA      DESC_RESP      SATURACAO      DIARREIA
0              0           0           0           0              0
VOMITO        OUTRO_SIN      PUERPERA      FATOR_RISC      CARDIOPATI      HEMATOLOGI
0              0           0           0           0              0
SIND_DOWN      HEPATICA          ASMA          DIABETES      NEUROLOGIC      PNEUMOPATI
0              0           0           0           0              0
IMUNODEPRE      RENAL      OBESIDADE      OUT_MORBI      ANTIVIRAL      HOSPITAL
0              0           0           0           0              0
DT_INTERNA      UTI      DT_ENTUTI      DT_SAIDUTI      SUPORT_VEN      EVOLUCAO
6228            0          127138          151717          0              0
DT_EVOLUCA      PCR_SARS2      DOR_ABD          FADIGA      PERD_OLFT      PERD_PALA
7973            0           0           0           0              0
TOMO_RES        RES_AN
0              0

```

Figura 17 – Verificação dos valores ausentes das variáveis de data.

As ausências de valores em algumas destas variáveis são esperadas pois, nem todos os pacientes, por exemplo, foram transferidos e tratados na UTI. Entretanto, valores ausentes em variáveis tais como, data de nascimento ou data da evolução (cura ou óbito), podem representar dados incompletos, cabendo alguma decisão por parte do cientista de dados. Vamos a análise detalhada.

DT_SIN_PRI: Na variável que se refere a data dos primeiros sintomas, não encontramos valores ausentes, portanto, faremos uso desta variável.

```
> # DT_SIN_PRI
> dim(filter(srag_sp_v1, is.na(DT_SIN_PRI) & EVOLUCAO == "cura"))
[1] 0 44
> dim(filter(srag_sp_v1, is.na(DT_SIN_PRI) & EVOLUCAO == "óbito"))
[1] 0 44
```

Figura 18 – Valores ausentes na variável DT_SIN_PRI relacionados a evolução cura ou óbito.

DT_NASC: Na variável que se refere a data de nascimento dos pacientes, encontramos 100 registros ausentes. Destes registros, 64 evoluíram para a cura e os outros 36 para o óbito dos pacientes.

```
> # DT_NASC
> dim(filter(srag_sp_v1, is.na(DT_NASC) & EVOLUCAO == "cura"))
[1] 64 44
> dim(filter(srag_sp_v1, is.na(DT_NASC) & EVOLUCAO == "óbito"))
[1] 36 44
```

Figura 19 – Valores ausentes na variável DT_NASC relacionados a evolução cura ou óbito.

Considerando que o total de registros ausentes nesta variável representam 0,053% do dataset, optamos pela exclusão dos registros ausentes e pela utilização da variável. Considerando que a maioria dos registros excluídos estava relacionados aos casos de cura, a exclusão também contribuiu para o balanceamento do dataset.

```
# DT_NASC - eliminando valores NA que impossibilitam o cálculo da idade
# Deletado 100 registros
srag_sp_v1 <- subset(srag_sp_v1, !is.na(DT_NASC))
```

Figura 20 – Eliminando os 100 registros com DT_NASC ausentes.

DT_INTERNA: Na variável que se refere a data em que o paciente foi hospitalizado, encontramos 6228 registros ausentes. Destes 6228 registros, 3846 evoluíram para a cura enquanto que 2382 para o óbito do paciente. Considerando que o total de registros ausentes nesta variável representam 3,33% do dataset, poderíamos fazer a exclusão e utilizar a variável, porém, optamos pelo seu descarte. O motivo do descarte foi para não comprometer o número de registros que pretendemos utilizar em nossos modelos e que, outras datas que poderiam ser utilizadas para encontrar o intervalo de tempo entre os fatos, também apresentam elevados números de registros ausentes.

```
> # DT_INTERNA
> dim(filter(srag_sp_v1, is.na(DT_INTERNA) & EVOLUCAO == "cura"))
[1] 3846 44
> dim(filter(srag_sp_v1, is.na(DT_INTERNA) & EVOLUCAO == "óbito"))
[1] 2382 44
```

Figura 21 – Valores ausentes na variável DT_INTERNA relacionados a evolução cura ou óbito.

DT_ENTUTI: Na variável que se refere a data em que o paciente entrou na UTI, encontramos 127.138 registros ausentes. Destes 103.330 evoluíram para a cura enquanto que 23.808 para o óbito do paciente. Considerando que o total de registros ausentes nesta variável representam 68% do dataset, optamos pelo seu descarte, assim preservando os registros para análise, porém sem interferências nos resultados dos modelos.

```
> # DT_ENTUTI
> dim(filter(srag_sp_v1, is.na(DT_ENTUTI) & EVOLUCAO == "Cura"))
[1] 103330    44
> dim(filter(srag_sp_v1, is.na(DT_ENTUTI) & EVOLUCAO == "Óbito"))
[1] 23808     44
```

Figura 22 – Valores ausentes na variável DT_INTERNA relacionados a evolução cura ou óbito.

DT_SAIDUTI: Na variável que se refere a data em que o paciente saiu da UTI, encontramos 151.717 registros ausentes. Destes 115.339 evoluíram para a cura enquanto que 36.378 para o óbito do paciente. Considerando que o total de registros ausentes nesta variável representam 81% do dataset, optamos pelo seu descarte, assim preservando os registros para análise, porém sem interferências nos resultados dos modelos.

```
> # DT_SAIDUTI
> dim(filter(srag_sp_v1, is.na(DT_SAIDUTI) & EVOLUCAO == "Cura"))
[1] 115339    44
> dim(filter(srag_sp_v1, is.na(DT_SAIDUTI) & EVOLUCAO == "Óbito"))
[1] 36378     44
```

Figura 23 – Valores ausentes na variável DT_INTERNA relacionados a evolução cura ou óbito.

Foram, então, eliminadas as variáveis DT_INTERNA, DT_ENTUTI e DT_SAIDUTI sem que fossem utilizadas em nossos modelos.

```
# Eliminando as variáveis de data cujo optamos por não utilizar em nossos modelos
srag_sp_v1$DT_INTERNA <- NULL
srag_sp_v1$DT_ENTUTI <- NULL
srag_sp_v1$DT_SAIDUTI <- NULL
```

Figura 24 – Eliminando as variáveis descartadas.

DT_EVOLUCA: Na variável que se refere a data da alta médica ou óbito do paciente, que é diretamente relacionada a nossa variável target, encontramos 7973 registros ausentes. Destes registros, 7.933 evoluíram para a cura e os outros 40 para o óbito dos pacientes.

```
> # DT_EVOLUCA
> dim(filter(srag_sp_v1, is.na(DT_EVOLUCA) & EVOLUCAO == "Cura"))
[1] 7933 44
> dim(filter(srag_sp_v1, is.na(DT_EVOLUCA) & EVOLUCAO == "Óbito"))
[1] 40 44
```

Figura 25 – Valores ausentes na variável DT_EVOLUCA relacionados a evolução cura ou óbito.

Considerando que o total de registros ausentes nesta variável representam 4,26% do dataset e pela sua relevância aos objetivos do nosso estudo, optamos pela exclusão dos registros ausentes e pela utilização da variável. Considerando que a grande maioria dos registros excluídos estava relacionados aos casos de cura, a exclusão também contribuiu para o balanceamento do dataset.

```
# DT_EVOLUCA - Eliminando valores NA que impossibilitam o cálculo do período patogênico
# Deletado 7973 registros
srag_sp_v1 <- subset(srag_sp_v1, !is.na(DT_EVOLUCA))
```

Figura 26 – Eliminando os 7973 com DT_EVOLUCA ausentes.

Considerando que 09 registros estavam nos dois comandos de exclusão dos registros ausentes (sem DT_NASC e sem DT_EVOLUCA), a nova dimensão do dataset é de: 179.021 registros com 41 variáveis.

3.1. Engenharia de Atributos

A engenharia de atributos consiste em um processo onde os dados são transformados para que sejam melhor utilizados nos modelos de aprendizado de máquinas. Normalmente esta tarefa está relacionada com nível de conhecimento do cientista de dados e o domínio do problema.

Neste estudo já utilizamos a engenharia de atributos de forma prática, quando transformamos os dados categóricos numéricos em categóricos com os significados numéricos. Neste exemplo, a engenharia de atributos foi de transformação pois, não houve alteração semântica dos atributos, apenas sua alteração para o melhor entendimento a aplicação aos métodos.

Decidimos mencionar este processo pois, com as variáveis de data que acabamos de tratar, vamos criar nova variáveis. Isso se deve ao fato de que, no nosso estudo, a data absoluta não representa um valor significativo ao modelo. Para fornecer atributos com maior relevância ao modelo, vamos criar novas variáveis que serão um número inteiro, representando a diferença em dias entre os eventos marcados pelas datas.

Ao primeiro destes novos atributos demos o nome de *período patogênico*. Este atributo será obtido da diferença em dias entre a data de evolução e data dos primeiros sintomas. Em outras palavras, fornecerá ao nosso modelo um número inteiro que representa o tempo em que o paciente esteve doente.

Ao segundo novo atributo demos o nome de *idade em dias*. Embora o dataset original tivesse esta informação, poderia não ser precisa ao modelo pois, dependia de outra variável para definir se o número inteiro se referia aos dias de vida, aos meses de vida ou anos. Ambas variáveis poderiam ter valores ausentes, comprometendo os números de registros para treino e teste. Esta variável será obtida pela diferença em dias entre a data dos primeiros sintomas e a data do nascimento. Portanto, fornecerá ao modelo a idade que o paciente tinha quando apresentou os primeiros sintomas.

```
# Criando variável de período patogênico.
# Período em dias da data dos primeiros sintomas até a cura ou óbito
# calculado pela diferença em dias das variáveis DT_SIN_PRI e DT_EVOLUCA
srag_sp_v1$PERIODO_PATOGENICO <- difftime(srag_sp_v1$DT_EVOLUCA,srag_sp_v1$DT_SIN_PRI, units = c("days"))

# Criando variável de IDADE_EM_DIAS.
# Período em dias referente a idade que o paciente tinha na data dos primeiros sintomas.
# calculado pela diferença em dias das variáveis DT_SIN_PRI e DT_NASC
srag_sp_v1$IDADE_EM_DIAS <- difftime(srag_sp_v1$DT_SIN_PRI,srag_sp_v1$DT_NASC, units = c("days"))

# Transformando a variável período patogênico para o tipo numérica inteiro (int)
srag_sp_v1$PERIODO_PATOGENICO <- as.numeric(srag_sp_v1$PERIODO_PATOGENICO, units="days")

# Transformando a variável idade em dias para o tipo numérica inteiro (int)
srag_sp_v1$IDADE_EM_DIAS <- as.numeric(srag_sp_v1$IDADE_EM_DIAS, units="days")
```

Figura 27 – Criando novas variáveis Período patogênico e idade em dias.

Após a criação das variáveis, transformamos seus tipos para inteiro e assim, podemos deletar as variáveis contendo as descrições das datas do nosso dataset.

Gravamos todos estes tratamentos em um novo arquivo, assim finalizando o tratamento das variáveis. Assim, o presente estudo prosseguiu para o aprofundamento e a exploração dos dados.

```

# Transformando a variável período patogênico para o tipo numérica inteiro (int)
srag_sp_v1$PERIODO_PATOGENICO <- as.numeric(srag_sp_v1$PERIODO_PATOGENICO, units="days")

# Transformando a variável idade em dias para o tipo numérica inteiro (int)
srag_sp_v1$IDADE_EM_DIAS <- as.numeric(srag_sp_v1$IDADE_EM_DIAS, units="days")

# Eliminando as variáveis de data que foram utilizadas acima e não serão mais úteis
srag_sp_v1$DT_SIN_PRI <- NULL
srag_sp_v1$DT_EVOLUCA <- NULL
srag_sp_v1$DT_NASC <- NULL

#=====
#== FIM - Engenharia de atributos
#=====

# Salvando as alterações em novo arquivo
write.csv(srag_sp_v1, "srag_sp_covid_v2.csv")

```

Figura 28 – Transformando os valores das novas variáveis para numéricos e excluindo as variáveis de data utilizadas para criar as novas variáveis.

4. Análise e Exploração dos Dados

Devido ao grande número de variáveis existentes, este capítulo será extenso, pois, faremos a análise de cada variável de forma detalhada, bem como os ajustes, quando necessários.

Neste momento cabe esclarecer um termo que poderá ser frequentemente utilizado daqui em diante: “*outliers*”. *Outliers* são dados que se diferenciam drasticamente de todos os outros, são pontos considerados “fora da curva normal”. Em outras palavras, um *outlier* é um valor que se diferencia da normalidade, que não será representativo ao modelo e que provavelmente irá causar anomalias nos resultados obtidos por meio de algoritmos e sistemas de análise. Por isso será importante eliminar estes dados para que o modelo possa ser o mais preciso possível.

4.1. Balanceamento do dataset.

Após todos os tratamentos das variáveis anteriormente descritas, iniciamos nosso dataset para exploração com 179.021 registros e 40 variáveis.

A partir deste momento, em que se iniciou a exploração das variáveis, também se tornou importante analisar o balanceamento do dataset. A variável target deve estar balanceada, para que os modelos preditivos possam fornecer o melhor resultado, do contrário, os modelos fornecerão uma visão distorcida dos dados.

Observamos que todos os registros da variável target estão divididos em 70,37% dos pacientes que evoluíram para a cura o restante, 39,63%, dos pacientes que evoluíram para óbito.

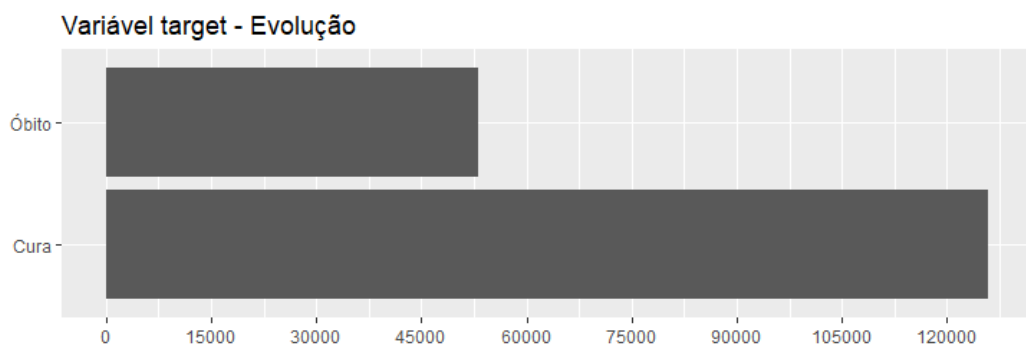


Figura 29 – Período patológico – marcações estatísticas.

Para balancear nossa base de dados, optamos pela técnica da reamostragem (resampling). Neste estudo, considerando que existem 70% dos registros correspondente aos casos de cura, faremos uma sub-amostragem, removendo de forma aleatória alguns registros desta classe até que sua amostragem seja igual à do óbito.

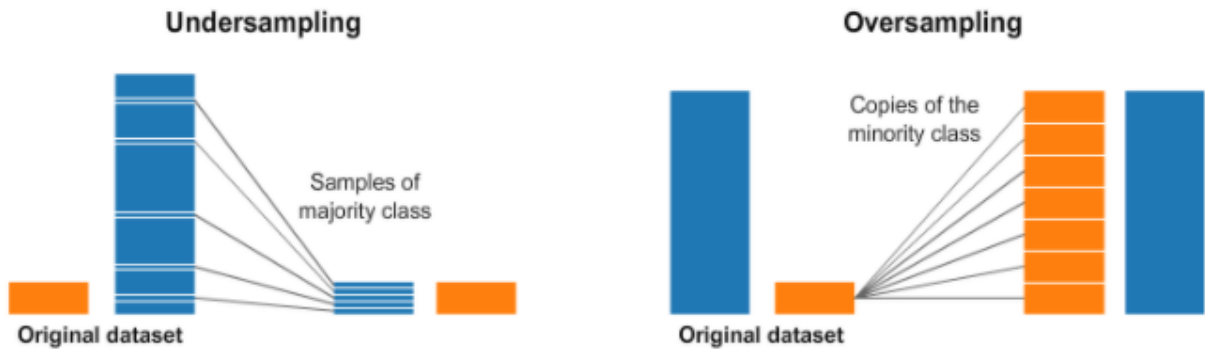


Figura 30 – Técnica de balanceamento por reamostragem.
Disponível em: <https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets>

Abaixo demonstramos como estavam os valores da variável target antes e após o balanceamento:

```
> summary(srag_sp_v1$EVOLUCAO)
Cura Óbito
125971 53050
```

a

```
> summary(srag_sp_v1$EVOLUCAO)
Cura Óbito
53050 53050
```

b

Figura 31 – Dataset desbalanceado (a) / Balanceado (b)

4.2. Variáveis demográficas (relacionadas ao paciente)

4.2.1. Variável PERÍODO_PATOGENICO

Observamos as seguintes métricas estatísticas para a variável:

```
> summary(srag_sp_v1$PERÍODO_PATOGENICO)
Min. 1st Qu. Median Mean 3rd Qu. Max.
 0.00  10.00  15.00 18.43  22.00 317.00
```

Figura 32 – Período patológico – marcações estatísticas.

Conforme a visualização das métricas estatísticas, 75% dos infectados pela COVID 19, permaneceram doentes por cerca de 22 dias. Observamos que a curva do histograma está deslocada a esquerda, sugerindo a presença de outliers.



Figura 33 – Período patogênico – histograma indicando a presença de outliers.

O boxplot abaixo, confirma a presença de outliers. Para ajustar a normalidade e evitar o overfitting no modelo vamos excluir os registros com período patogênico acima de 40 dias.

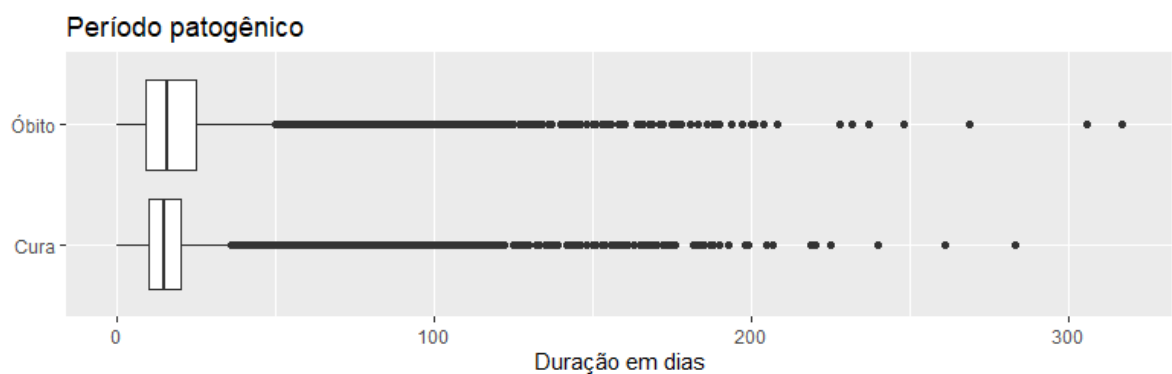


Figura 34 – Período patogênico – Boxplot indicando a presença de outliers.

No total foram excluídos 6582 dados outliers, sendo 2540 referentes a cura e 4042 referente aos óbitos. Abaixo, seguem os histogramas e boxplot após a exclusão dos registros.

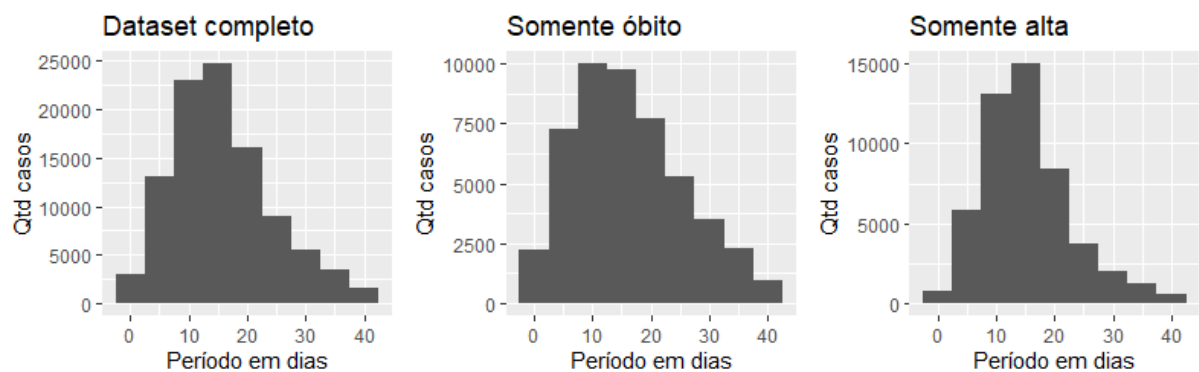


Figura 35 – Período patogênico – Histograma após exclusão de parte dos outliers

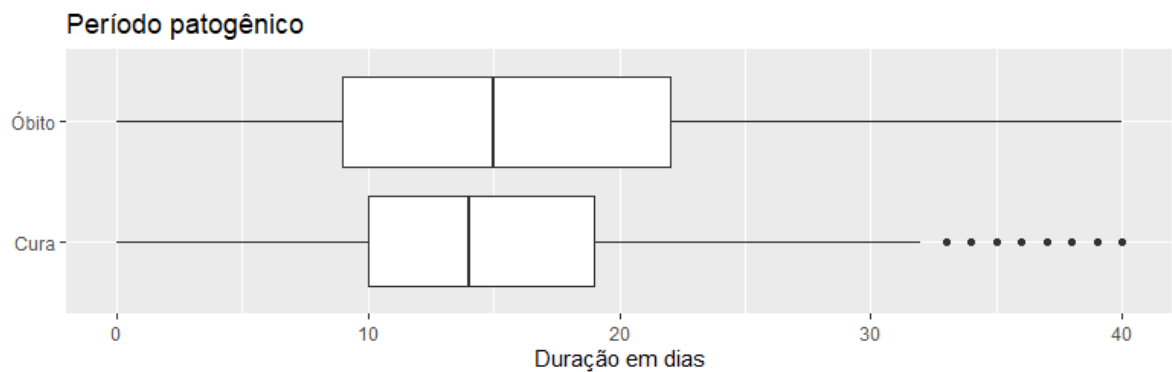


Figura 36 – Período patogênico – Boxplot após exclusão de parte dos outliers.

4.2.2. Variável IDADE_EM_DIAS

Observamos as seguintes métricas estatísticas para esta variável:

```
> summary(srag_sp_v1$IDADE_EM_DIAS)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0   18453   23421   22836   27725   40312
```

Figura 37 – Idade em dias – marcações estatísticas.

Conforme a visualização das métricas estatísticas, 75% dos casos de doentes se concentram em idades até 27725 dias, equivalente a aproximadamente 76 anos quando os pacientes sentiram os primeiros sintomas. A média e a mediana equivalem respectivamente ao equivalente a 64 anos e a mediana a 62 anos e seis meses. De forma geral, os gráficos da variável idade em dias, seguem uma distribuição normal. As idades acima de 60 anos ficam evidenciadas pelas curvas dos histogramas deslocadas para o lado direito, com maior destaque aos casos de óbitos.

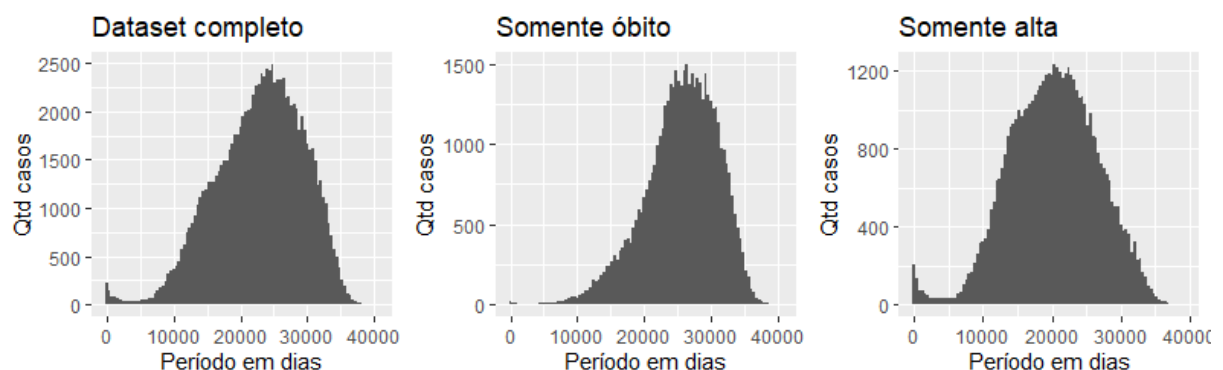


Figura 38 – Idade em dias – Curvas com distribuição normal e leve deslocamento a direita.

O boxplot abaixo, indica a presença de outliers. Para ajustar a normalidade e evitar o overfitting no modelo vamos excluir os registros com idade menor que 10.000 dias, aproximadamente 27 anos.

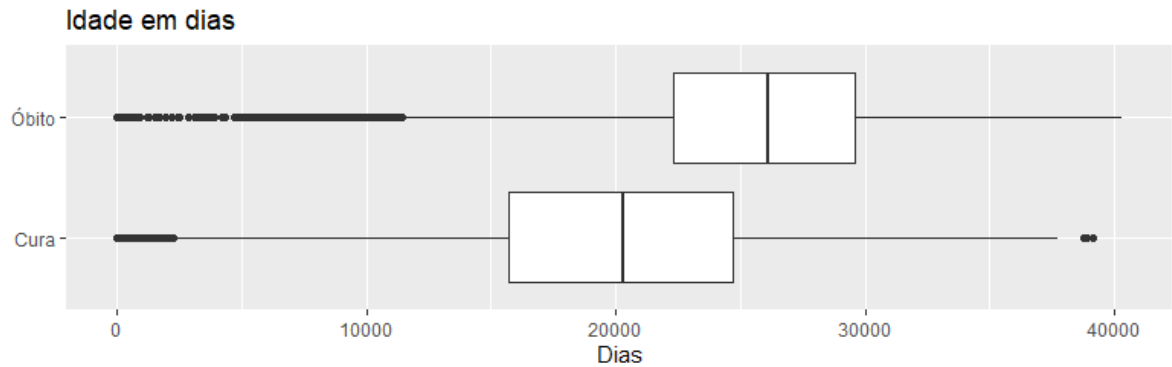


Figura 39 – Idade em dias – Boxplot indicando a presença de outliers.

No total foram excluídos 2889 dados outliers, sendo 2521 referentes a cura e 368 referente aos óbitos. Abaixo, seguem os histogramas e boxplot após a exclusão dos registros.

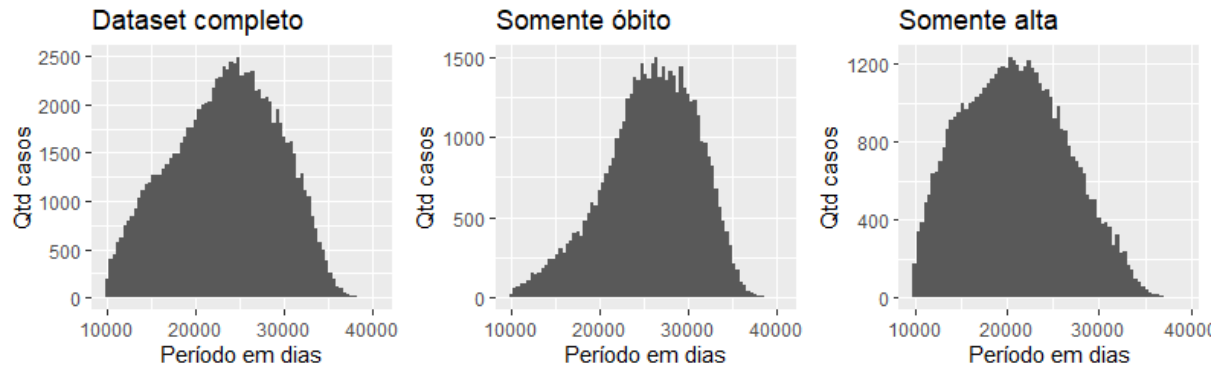


Figura 40 – Período patogênico – boxplot.

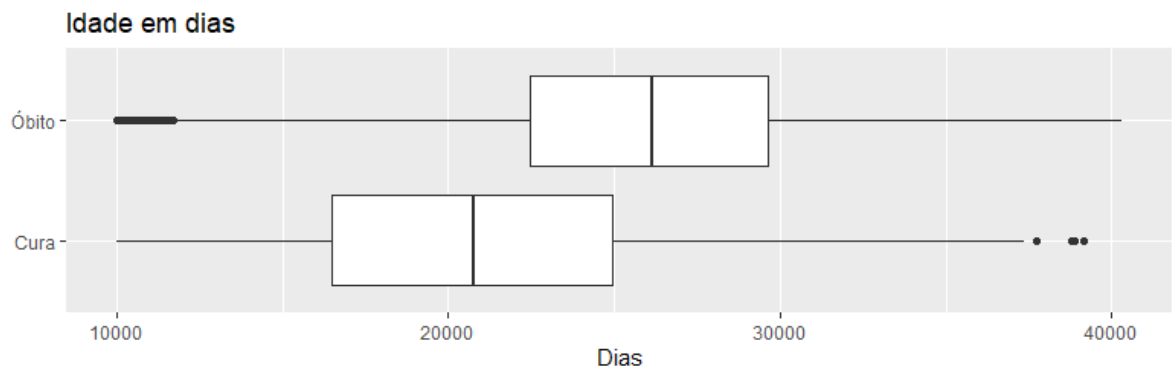


Figura 41 – Período patogênico – Boxplot após exclusão de parte dos outliers

4.2.3. Variável CS_RACA

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$CS_RACA)
Amarela   Branca Ignorado Indígena   Parda   Preta
  1324    52046   19609      73    18601   4976
```

Figura 42 – CS_RACA - Categorias e quantidades.

Para todas as categorias, observamos que a quantidade de óbito é maior que a de cura, destacando-se os pacientes de raça branca. Somente nos registros onde as raças foram ignoradas, os números de óbito foram menores.

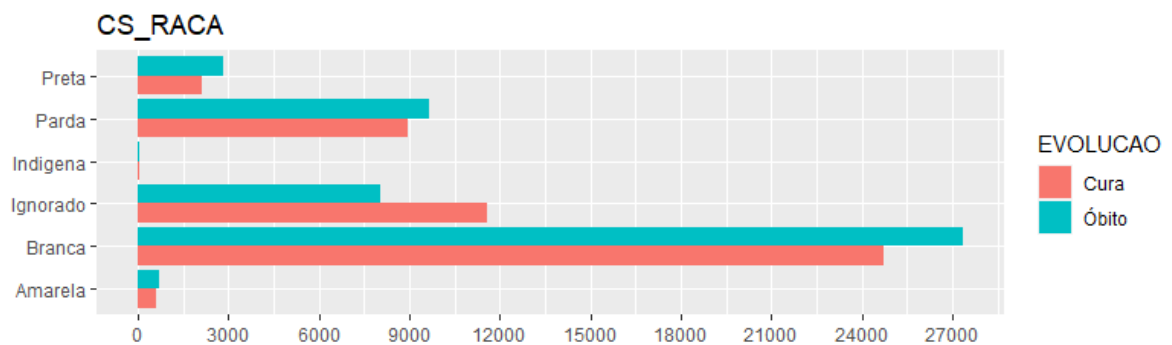


Figura 43 – Relação das variáveis CS_RACA e EVOLUCAO.

4.2.4. Variável CS_SEXO

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$CS_SEXO)
Feminino Ignorado Masculino
  41962      5      54662
```

Figura 44 – CS_SEXO - Categorias e quantidades.

Observamos que indivíduos do sexo masculino foram mais acometidos pela doença do que os do sexo feminino. Além disso, a porcentagem de óbitos também é maior para os pacientes do sexo masculino.

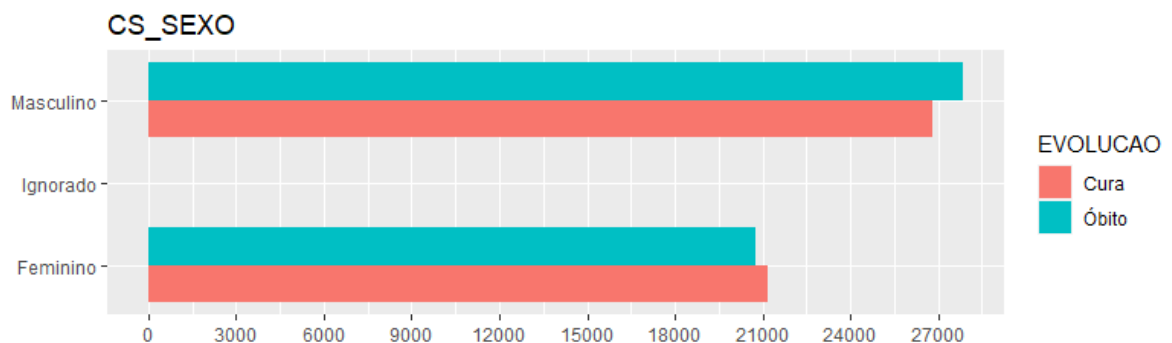


Figura 45 – Relação das variáveis CS_SEXO e EVOLUCAO.

4.2.5. Variável CS_ESCOL_N

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$CS_ESCOL_N)
```

Fundamental 1	Fundamental 2	Ignorado	Médio	Sem escolaridade	superior
11058	6952	59378	11448	2019	5774

Figura 46 – CS_ESCOL_N- Categorias e quantidades.

Observamos que quanto maior o nível de escolaridade do paciente, a relação de cura da doença é maior. Algumas hipóteses podem servir para explicar este comportamento: pacientes com maior escolaridade podem ter melhor esclarecimento sobre os fatos, melhor condição para praticar hábitos preventivos e normalmente, com mais possibilidades de praticar suas atividades profissionais de forma remota.

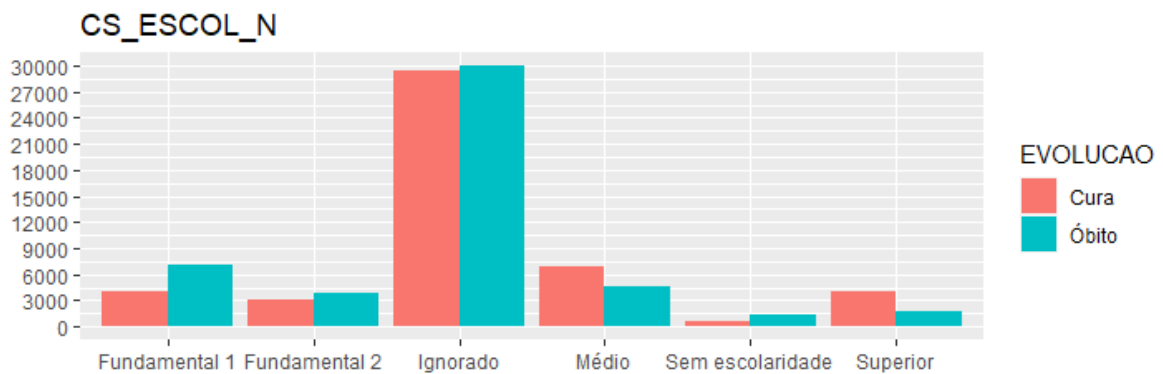


Figura 47 – Relação das variáveis CS_ESCOL_N e EVOLUCAO.

4.3. Variáveis relacionadas a sintomas e sinais.

4.3.1. Variável FEBRE

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$FEBRE)
```

Ignorado	Não	Sim
12465	27926	56238

Figura 48 – FEBRE- Categorias e quantidades.

Os dados válidos sobre febre combinados com a variável target, nos demonstram uma informação interessante: grande parte dos pacientes que apresentam febre durante o trânsito da doença, se curaram, enquanto que aqueles que não apresentaram febre como sintoma, vieram a óbito, em sua maioria.

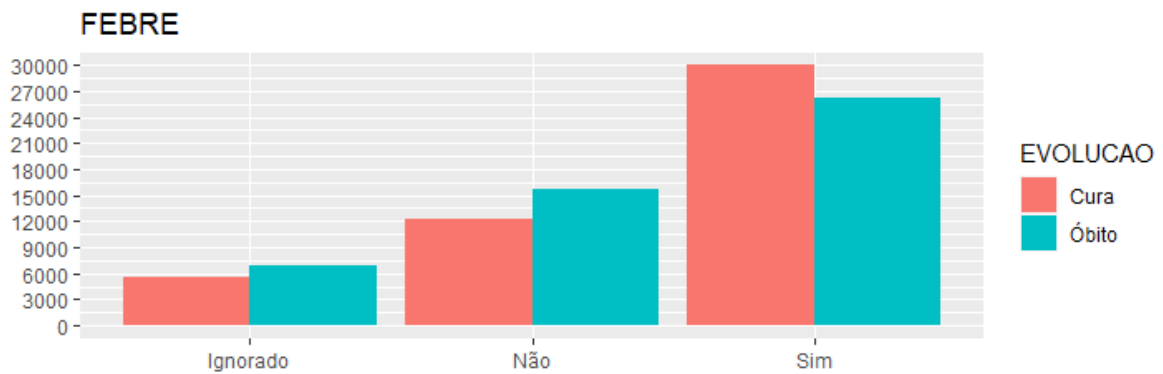


Figura 49 – Relação das variáveis FEBRE e EVOLUCAO.

4.3.2. Variável TOSSE

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$TOSSE)
Ignorado   Não   Sim
  9888   18944  67797
```

Figura 50 – TOSSE- Categorias e quantidades.

Os dados válidos sobre tosse combinados com a variável target, nos demonstram uma outra informação interessante: A maioria dos pacientes que apresentam tosse tiveram maior quantidade de cura, enquanto que aqueles que não apresentaram tosse tiveram maior quantidade de óbito.

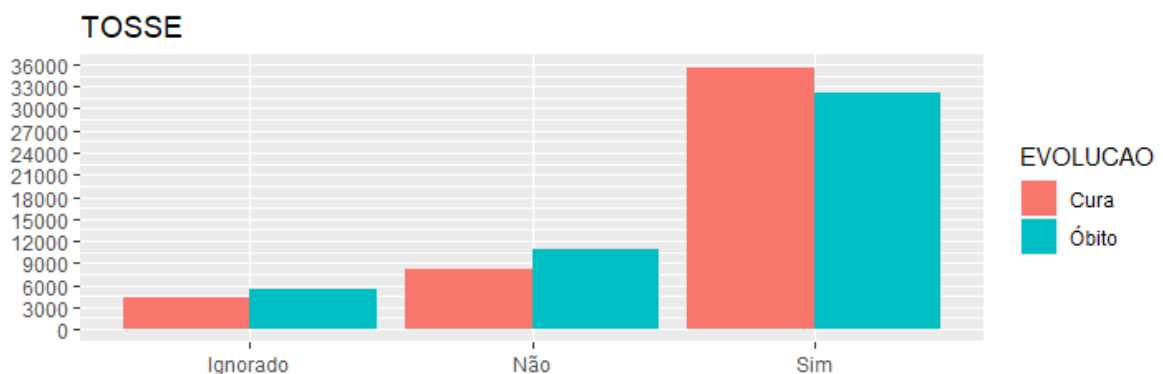


Figura 51 – Relação das variáveis TOSSE e EVOLUCAO.

4.3.3. Variável GARGANTA

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$GARGANTA)
Ignorado   Não   Sim
 24063   57080  15486
```

Figura 52 – GARGANTA- Categorias e quantidades.

Os dados válidos sobre dor de garganta combinados com a variável target, nos demonstram que a minoria dos pacientes apresentou dor de garganta, entretanto, nesta situação o número de cura foi maior, enquanto que aqueles que não apresentaram dor de garganta, que foram a maioria dos pacientes, tiveram maior número de óbitos entre seus casos.

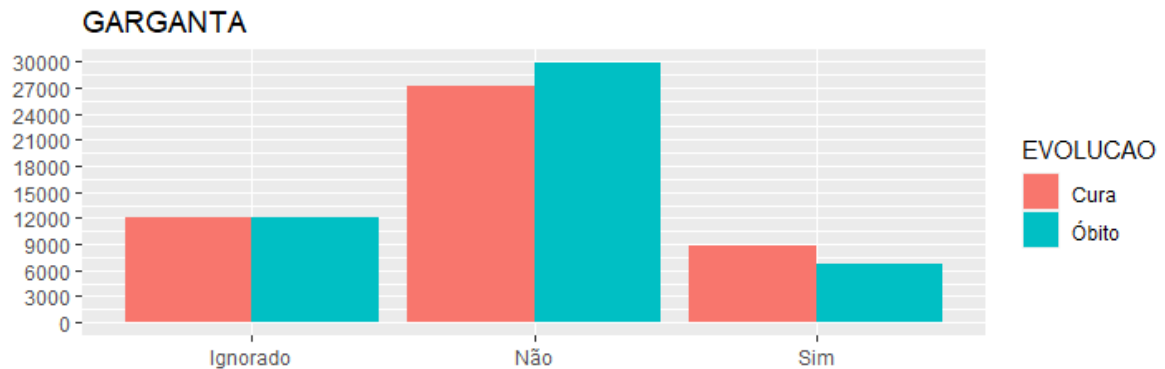


Figura 53 – Relação das variáveis GARGANTA e EVOLUCAO.

4.3.4. Variável DISPNEIA

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$DISPNEIA)
Ignorado   Não     Sim
  10026   17984  68619
```

Figura 54 – DISPNEIA- Categorias e quantidades.

Os dados válidos sobre dispneia, demonstram que a maioria dos pacientes apresentaram este sintoma e que, dentre estes casos, houve maior número de óbitos. Dentre os pacientes que não apresentaram dispneia, observamos maior número de cura.

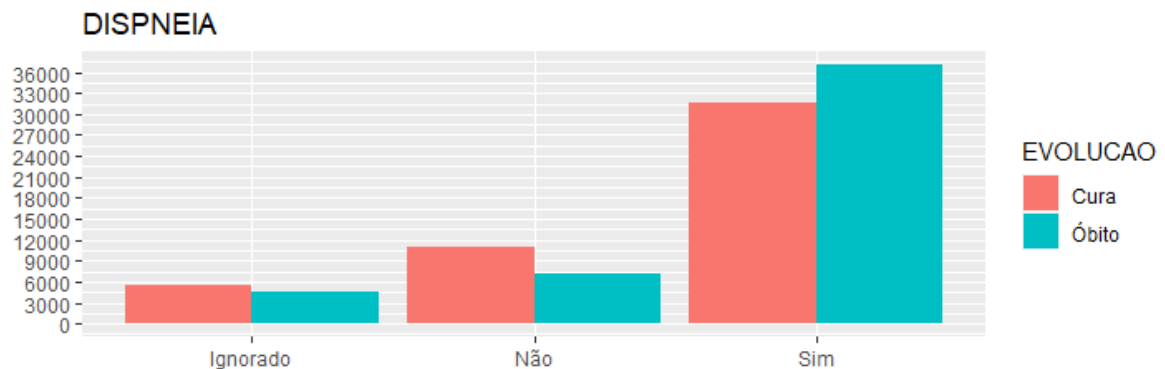


Figura 55 – Relação das variáveis DISPNEIA e EVOLUCAO.

4.3.5. Variável DESC_RESP

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$DESC_RESP)
Ignorado   Não     Sim
  15173    23578   57878
```

Figura 56 – DESC_RESP- Categorias e quantidades.

Os dados válidos sobre desconforto respiratório, demonstram que a maioria dos pacientes apresentaram este sintoma e que, dentre estes casos houve maior número de óbitos. Dentre os pacientes que não apresentaram desconforto respiratório, observamos maior número de cura da doença.

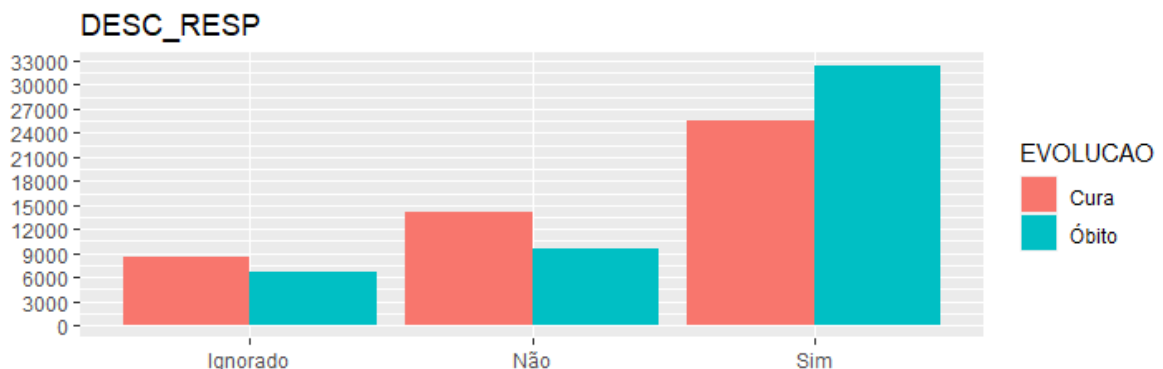


Figura 57 – Relação das variáveis DESC_RESP e EVOLUCAO.

4.3.6. Variável SATURAÇÃO

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$SATURACAO)
Ignorado   Não     Sim
  13518    21143   61968
```

Figura 58 – SATURACAO - Categorias e quantidades.

Os dados válidos sobre saturação, demonstram que os pacientes que apresentaram $O_2 < 95\%$, representam a maioria dos casos, dentre estes casos houve maior número de óbitos. Dentre os pacientes que não apresentaram queda na saturação, observamos maior número de cura.

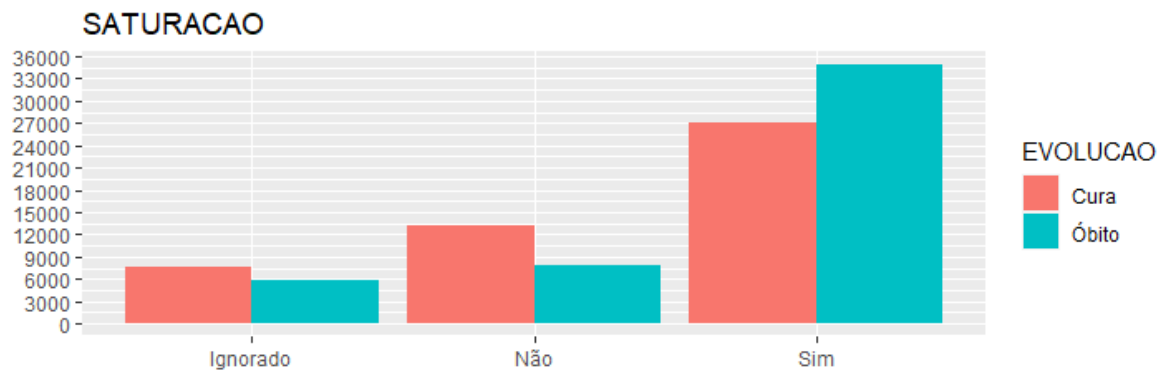


Figura 59 – Relação das variáveis SATURACAO e EVOLUCAO.

4.3.7. Variável DIARREIA

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$DIARREIA)
Ignorado   Não     Sim
 25104    59095   12430
```

Figura 60 – DIARREIA- Categorias e quantidades.

Os dados válidos sobre dos pacientes que apresentaram diarreia, nos demonstram que este não é um sintoma apresentado na maioria dos casos, entretanto, entre os que apresentaram a relação de cura foi maior.

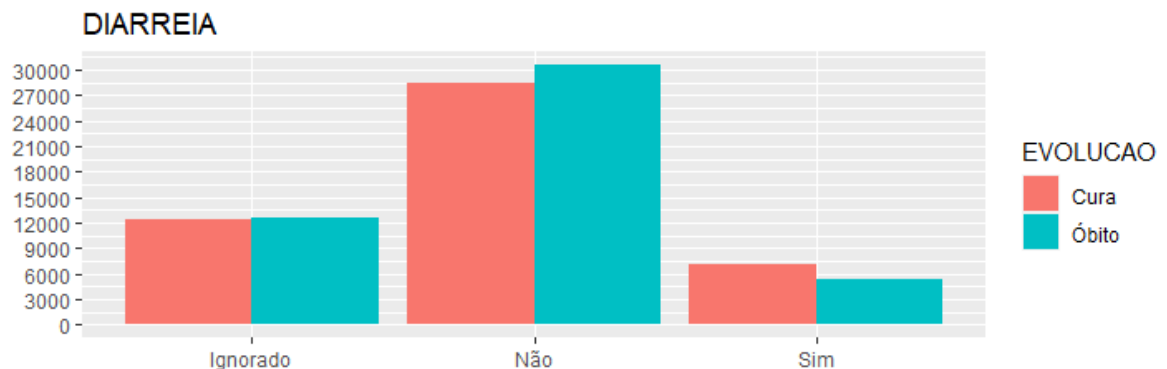


Figura 61 – Relação das variáveis DIARREIA e EVOLUCAO.

4.3.8. Variável VÔMITO

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$VOMITO)
Ignorado   Não     Sim
 26547    63277   6805
```

Figura 62 – VÔMITO- Categorias e quantidades.

Os dados válidos sobre vômito, nos demonstram que este não é um sintoma apresentado na maioria dos casos. Dentre os que apresentam ou não, a relação de cura e óbito estão quase equilibradas, entretanto, existe uma ligeira alta de cura entre os pacientes que apresentaram este sinal.

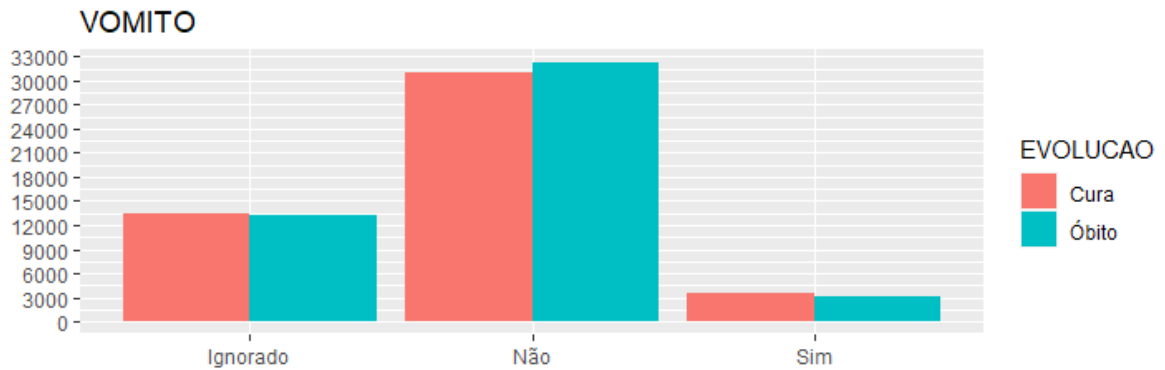


Figura 63 – Relação das variáveis VOMITO e EVOLUCAO.

4.3.9. Variável DOR_ABD

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$DOR_ABD)
Ignorado   Não     Sim
  58329   35679   2621
```

Figura 64 – DOR_ABD- Categorias e quantidades.

Observamos que a maioria das informações relativas a dores abdominais nos pacientes foram ignoradas. Além disso, foram poucos os pacientes que relataram ou apresentaram este sintoma e dentre eles, a quantidade de cura e óbito estão equilibradas. Acreditamos que esta variável não apresentará relevância quando o modelo de seleção das principais variáveis for aplicado.

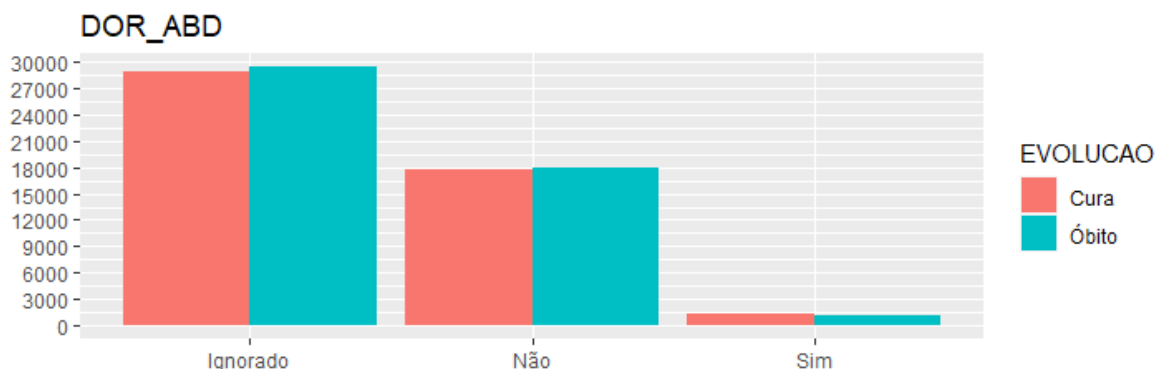


Figura 65 – Relação das variáveis DOR_ABD e EVOLUCAO.

4.3.10. Variável FADIGA

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$FADIGA)
Ignorado   Não     Sim
  56860   27610  12159
```

Figura 66 – FADIGA - Categorias e quantidades.

Observamos que a maioria das informações relativas a fadigas nos pacientes também foram ignoradas. A menor parte dos pacientes que apresentaram este sintoma apresentaram a cura da doença sem maiores complicações.

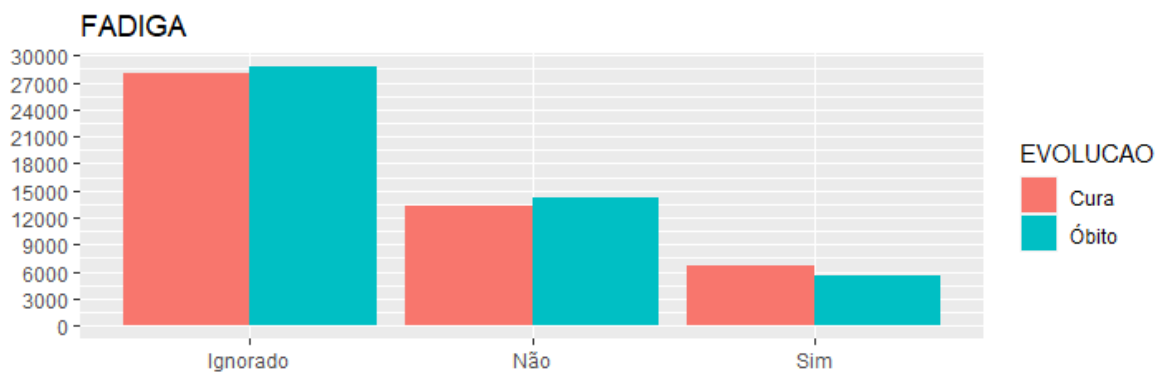


Figura 67 – Relação das variáveis FADIGA e EVOLUCAO.

4.3.11. Variável PERD_OLFT

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$PERD_OLFT)
Ignorado   Não     Sim
  58291   33208   5130
```

Figura 68 – PERD_OLFT - Categorias e quantidades.

Observamos que a maioria das informações relativas as perdas do olfato foram ignoradas. Apenas uma pequena fração dos pacientes relataram sinal, e dentre eles, aproximadamente 66% se curou. Uma maior quantidade não relatou este sinal e, dentre estes, 60% foram a óbito.

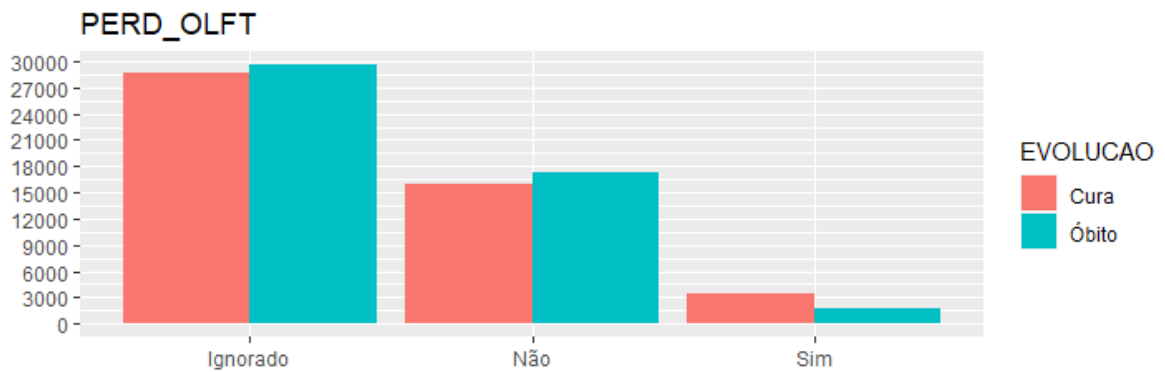


Figura 69 – Relação das variáveis PERD_OLFT e EVOLUCAO.

4.3.12. Variável PERD_PALA

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$PERD_PALA)
Ignorado    Não      Sim
  60173    33795    5550
```

Figura 70 – PERD_PALA - Categorias e quantidades.

Observamos que a maioria das informações sobre os pacientes que tiveram perda do paladar foram ignoradas, seguidas pelos pacientes que não tiveram perda de paladar.

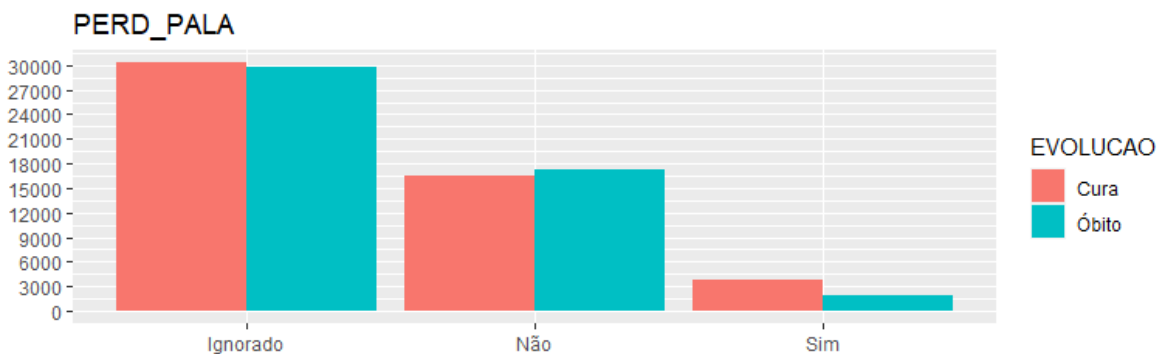


Figura 71 – Relação das variáveis PERD_PALA e EVOLUCAO.

4.3.13. Variável OUTRO_SIN

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$OUTRO_SIN)
Ignorado    Não      Sim
  27171    38676    30782
```

Figura 72 – OUTRO_SIN - Categorias e quantidades.

Observamos que a maioria dos pacientes não relataram outros sintomas além daqueles anteriormente quantificados, entretanto, neste grupo a quantidade de óbitos foi maior do que o grupo dos pacientes que relataram outros sintomas.

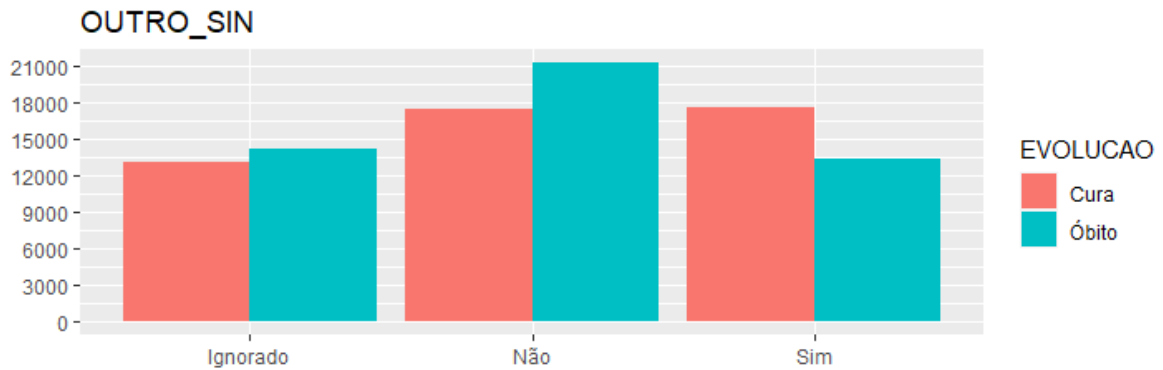


Figura 73 – Relação das variáveis OUTRO_SIN e EVOLUCAO.

4.4. Variáveis relacionadas a doenças pré-existentes.

4.4.1. Variável FATOR_RISC

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$FATOR_RISC)
Não   Sim
28926 67703
```

Figura 74 – FATOR_RISC - Categorias e quantidades.

Fator de risco é o nome dado ao grupo de doenças pré-existentes que os especialistas em saúde verificaram alguma relação com a covid-19. Os dados evidenciam que está é uma característica que se destaca pois, observamos que do total, 70% dos pacientes tinham algum fator de risco e destes, 60% evoluíram para óbito. Ao contrário, os outros 30% dos pacientes que não tinham nenhum fator de risco, 66% evoluíram para a cura.

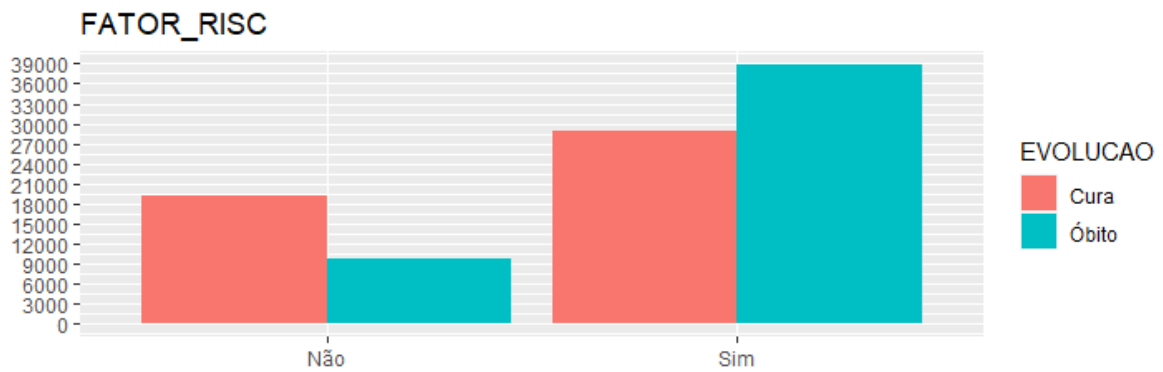


Figura 75 – Relação das variáveis FATOR_RISC e EVOLUCAO.

4.4.2. Variável PUERPERA

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$PUERPERA)
Ignorado   Não     Sim
  54927    41578    124
```

Figura 76 – PUERPERA - Categorias e quantidades.

Aproximadamente 56,84% dos dados desta variável foram classificados como ignorados e outros 43,02% classificados como não. Apenas 0,13% dos pacientes era puérpera ou parturiente, portanto, estes dados são referentes aos pacientes do sexo feminino.

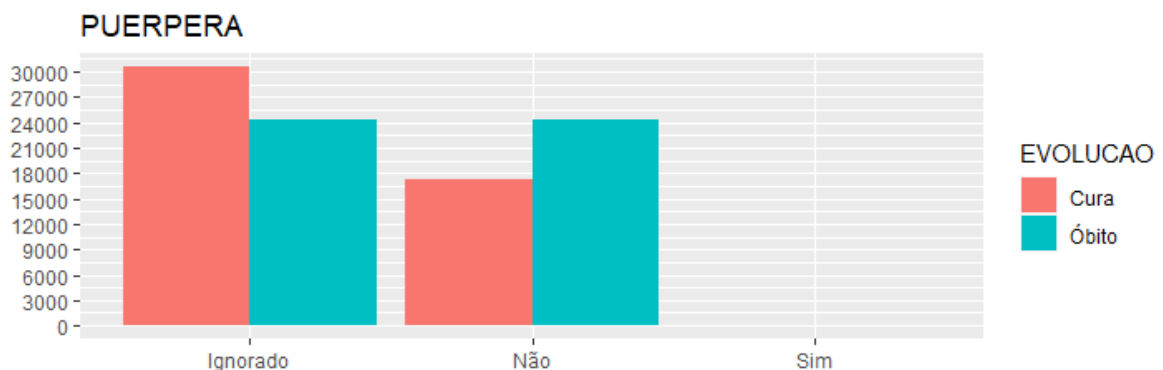


Figura 77 – Relação das variáveis PUERPERA e EVOLUCAO.

4.4.3. Variável CARDIOPATIA

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$CARDIOPATI)
Ignorado   Não     Sim
  40180    17487    38962
```

Figura 78 – CARDIOPATI - Categorias e quantidades.

Observamos significativa relevância do número de casos dos pacientes com Doença Vascular Crônica, acometidos pela COVID 19 que evoluíram para óbito.

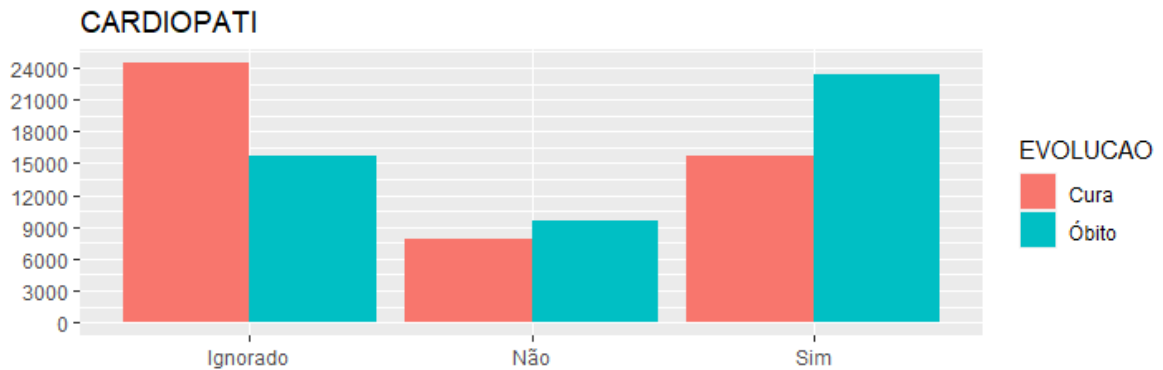


Figura 79 – Relação das variáveis CARDIOPATI e EVOLUCAO.

4.4.4. Variável HEMATOLOGICA

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$HEMATOLOGI)
Ignorado      Não      Sim
  54236      41484      909
```

Figura 80 – HEMATOLOGI - Categorias e quantidades.

Para esta variável, 56% dos dados foram ignorados, 43% dos pacientes não possuíam Doença Hematológica Crônica e somente cerca de 1% dos pacientes possuíam alguma doença hematológica preexistente.

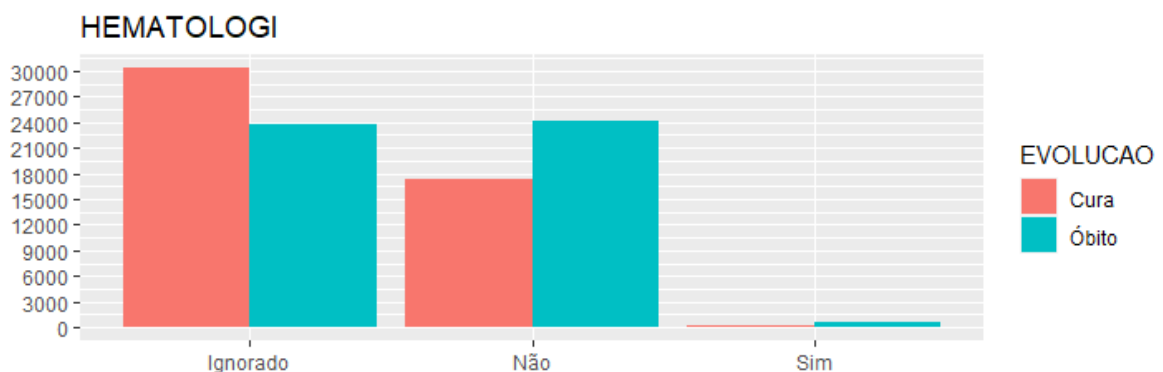


Figura 81 – Relação das variáveis HEMATOLOGI e EVOLUCAO.

4.4.5. Variável SIND_DOWN

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$SIND_DOWN)
Ignorado   Não     Sim
  54446    41947    236
```

Figura 82 – SIND_DOWN- Categorias e quantidades.

Para a Síndrome de Down, somados os dados da categoria ignorado e não somam aproximadamente 99,5%. Menos de 0,5% dos pacientes possuíam Síndrome de Down.

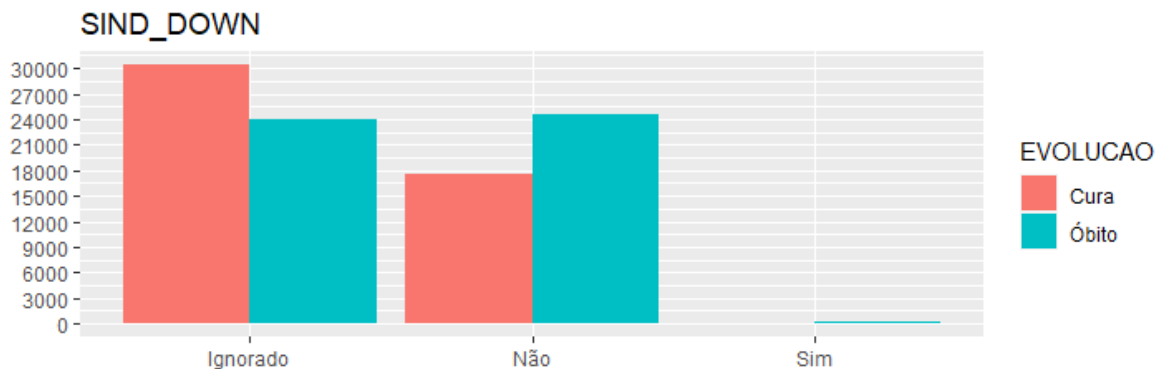


Figura 83 – Relação das variáveis SIND_DOWN e EVOLUCAO.

4.4.6. Variável HEPÁTICA

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$HEPATICA)
Ignorado   Não     Sim
  54353    41242    1034
```

Figura 84 – HEPATICA- Categorias e quantidades.

Aproximadamente 1% dos pacientes possuíam Doença Hepática Crônica. Dentre estes poucos casos, o número de óbitos foi relativamente maior também.

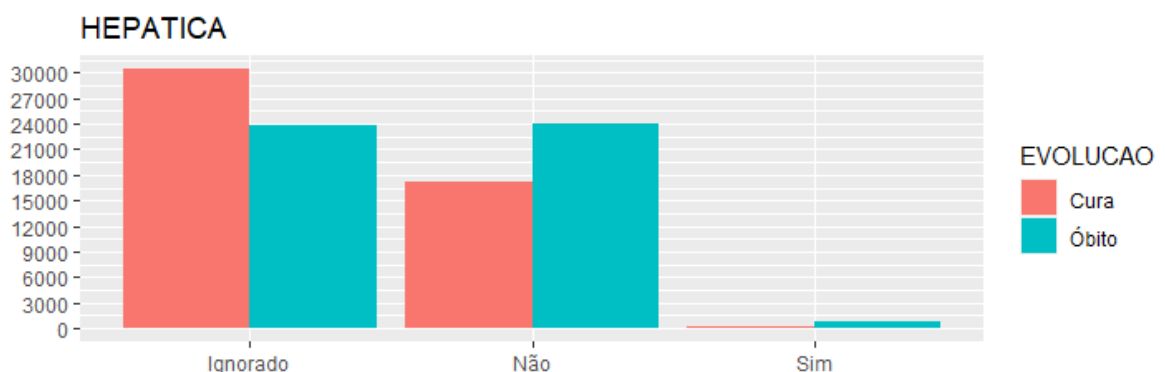


Figura 85 – Relação das variáveis HEPATICA e EVOLUCAO.

4.4.7. Variável ASMA

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$ASMA)
Ignorado   Não   Sim
  53807   40260  2562
```

Figura 86 – ASMA- Categorias e quantidades.

Observamos que o número dos pacientes que possuíam Asma foi relativamente baixo, comparado ao total de casos. Dentre estes, há um equilíbrio no número de óbitos.

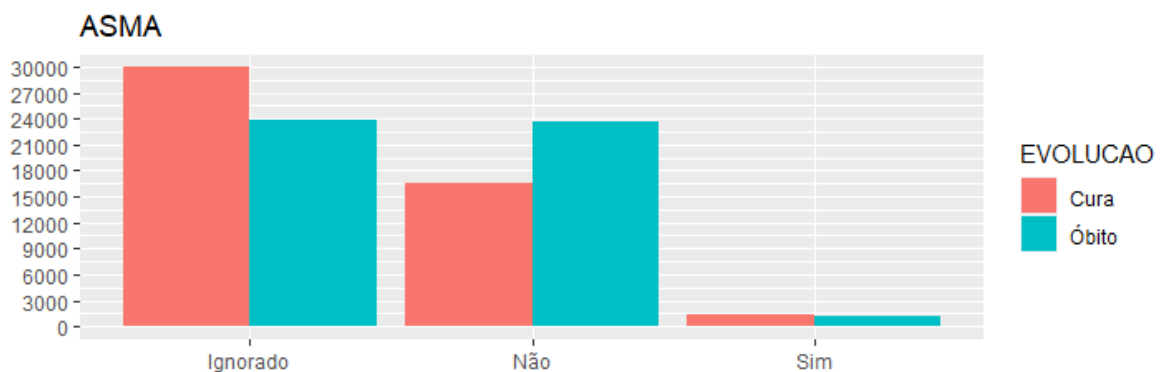


Figura 87 – Relação das variáveis ASMA e EVOLUCAO.

4.4.8. Variável DIABETES

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$DIABETES)
Ignorado   Não   Sim
  44115   24460  28054
```

Figura 88 – DIABETES- Categorias e quantidades.

O gráfico mostra que, dentre os pacientes com ou sem Diabetes, o número de casos e de óbitos é ligeiramente maior dos pacientes que possuíam diabetes.

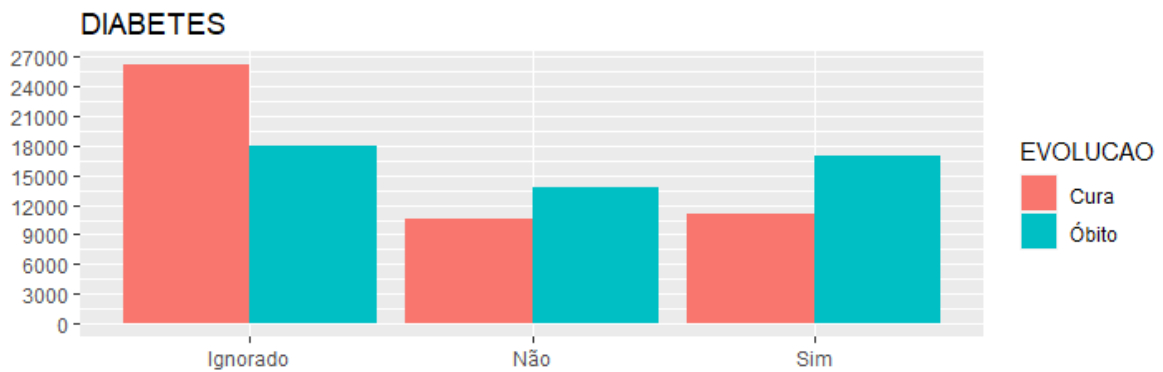


Figura 89 – Relação das variáveis DIABETES e EVOLUCAO.

4.4.9. Variável NEUROLÓGICA

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$NEUROLOGIC)
Ignorado   Não     Sim
 52766    38525    5338
```

Figura 90 – NEUROLOGIC - Categorias e quantidades.

A grande maioria dos pacientes não possuíam Doença Neurológica preexistente. Os números de óbitos são maiores com relação ao número de casos, tanto dos que possuem como dos que não.

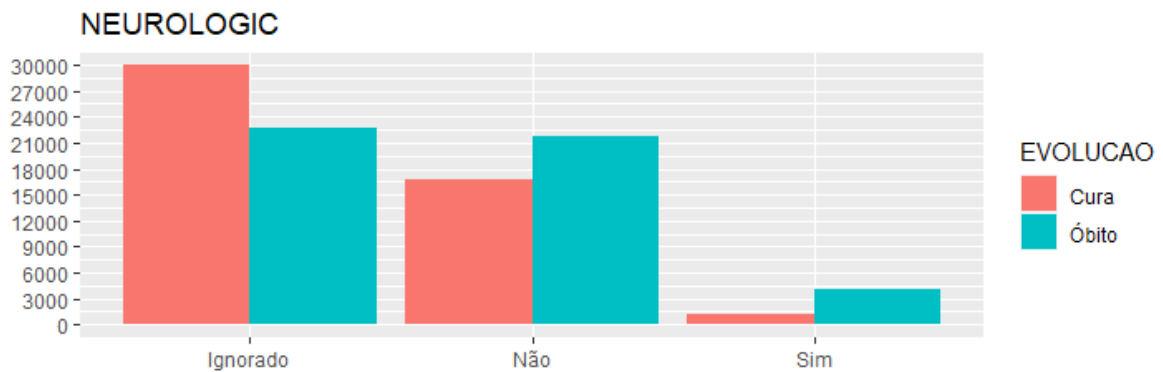


Figura 91 – Relação das variáveis NEUROLOGIC e EVOLUCAO.

4.4.10. Variável PNEUMOPATIA

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$PNEUMOPATI)
Ignorado   Não     Sim
 53024    39091    4514
```

Figura 92 – PNEUMOPATI - Categorias e quantidades.

A grande maioria dos pacientes não possuíam Pneumopatia Crônica. Os números de óbitos são maiores com relação ao número de casos, tantos dos que possuem como dos que não.

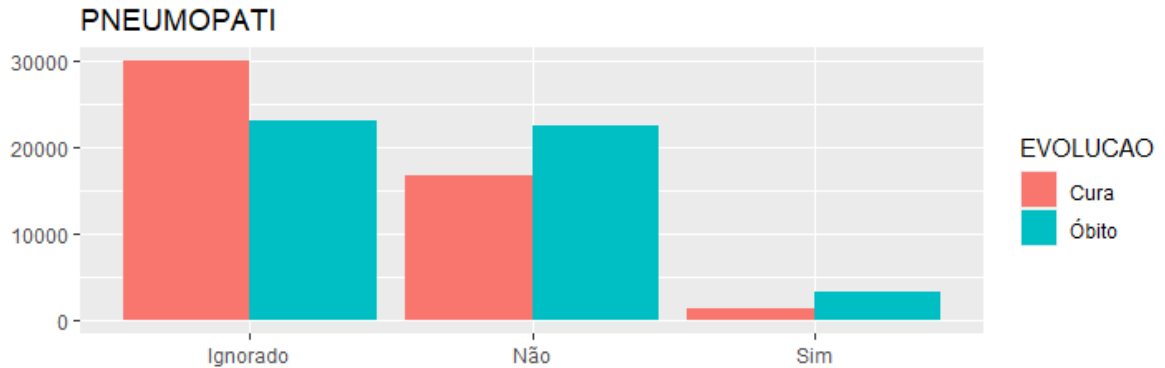


Figura 93 – Relação das variáveis PNEUMOPATI e EVOLUCAO.

4.4.11. Variável IMUNODEPRESSÃO

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$IMUNODEPRE)
Ignorado   Não     Sim
 53777   39912   2940
```

Figura 94 – IMUNODEPRE - Categorias e quantidades.

A grande maioria dos pacientes não possuíam outra Imunodepressão (diminuição da função do sistema imunológico) conhecida. Os números de óbitos são maiores com relação ao número de casos, tantos dos que possuíam como dos que não.

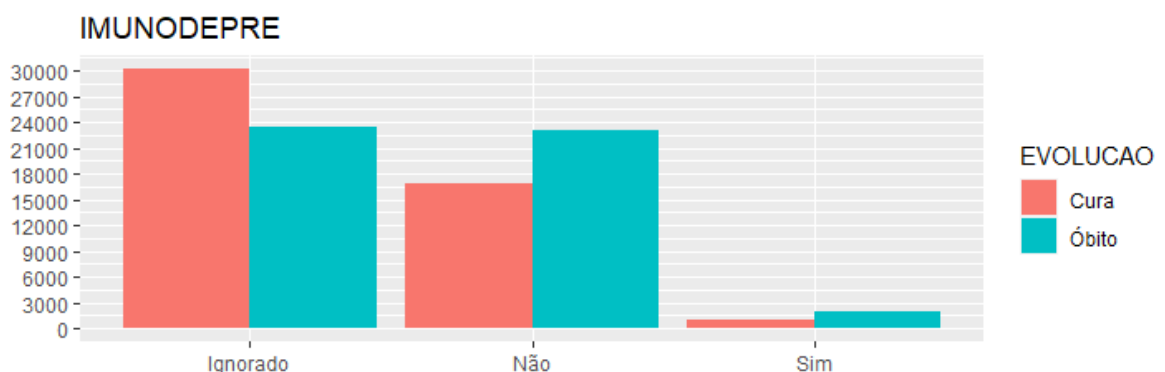


Figura 95 – Relação das variáveis IMUNODEPRE e EVOLUCAO.

4.4.12. Variável RENAL

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$RENAL)
Ignorado   Não     Sim
  53163    38788    4678
```

Figura 96 – RENAL - Categorias e quantidades.

A grande maioria dos pacientes não possuíam Doença Renal Crônica. Os números de óbitos são maiores com relação ao número de casos, tantos dos que possuem como dos que não.

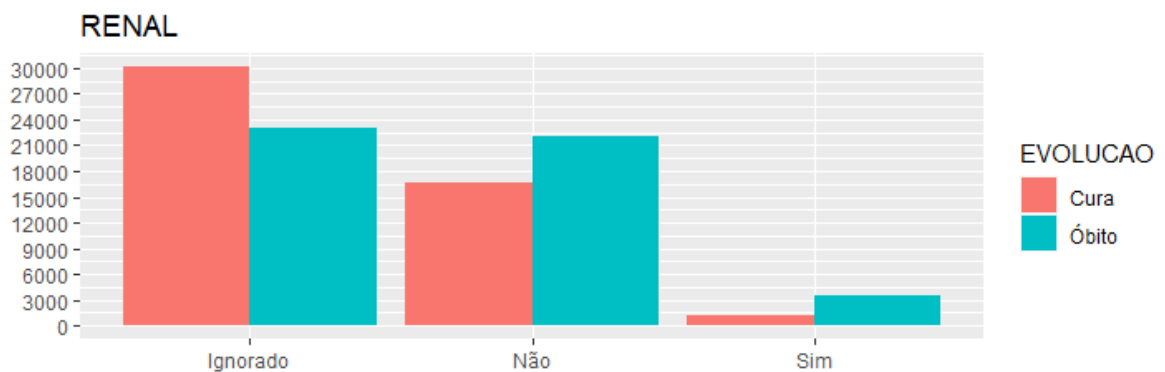


Figura 97 – Relação das variáveis RENAL e EVOLUCAO.

4.4.13. Variável OBESIDADE

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$OBESIDADE)
Ignorado   Não     Sim
  53697    36526    6406
```

Figura 98 – OBESIDADE - Categorias e quantidades.

Observamos que do total de óbitos, a menor parte foram em pacientes obesos. Entretanto o número de óbitos é maior com relação entre número de casos nos pacientes não obesos.

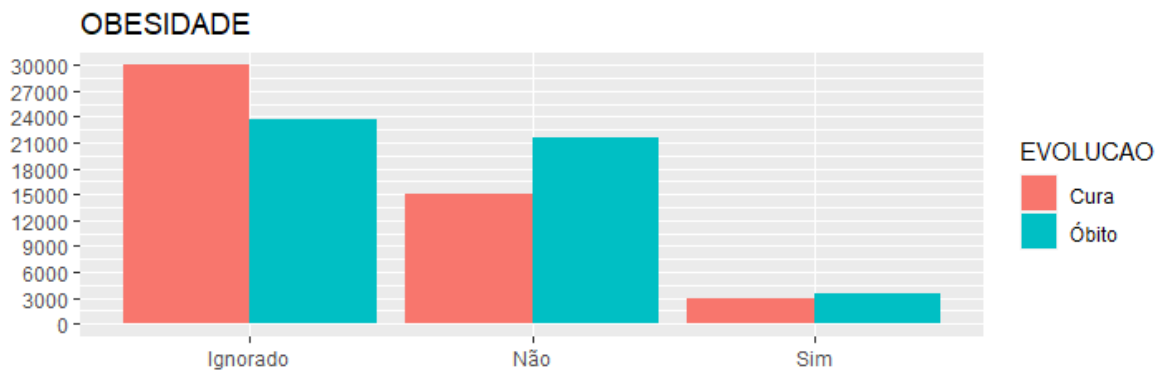


Figura 99 – Relação das variáveis OBESIDADE e EVOLUCAO.

4.4.14. Variável OUT_MORBI

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$OUT_MORBI)
Ignorado    Não      Sim 
  46361    22223    28045
```

Figura 100 – OUT_MORBI - Categorias e quantidades.

Observamos que o número de casos e também de óbitos foram maiores em pacientes com outras morbididades não relatadas anteriormente.

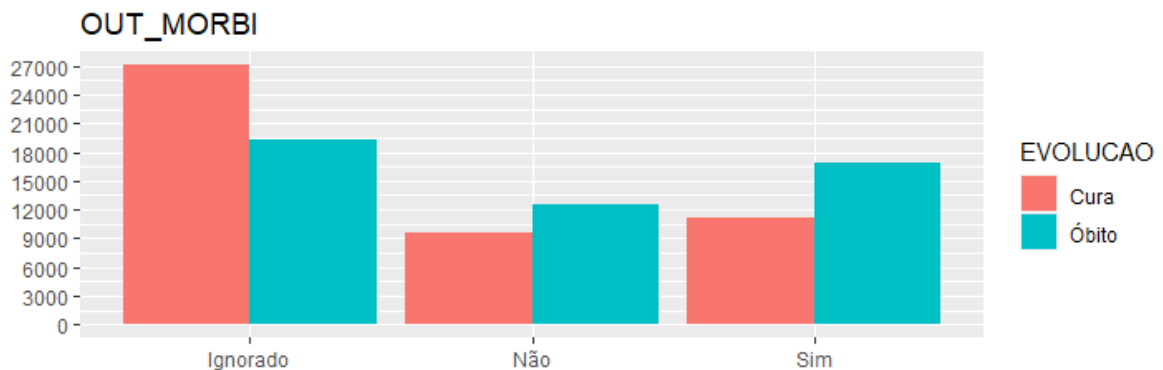


Figura 101 – Relação das variáveis OUT_MORBI e EVOLUCAO.

Neste estudo, procuramos avaliar os dados da forma mais numérica possível, mantendo-nos no foco da ciência de dados.

Com relação as variáveis relacionadas a doenças pré-existentes, também chamado de grupo de risco, a variável FATOR_RISC é a de maior relevância pois, concentra dados de todas as demais variáveis deste grupo. Algumas variáveis deste grupo isoladamente podem parecer irrelevante, entretanto, para um modelo

computacional estatísticos podem fazer alguma diferença, sendo preditoras ou contribuindo para o maior acerto do modelo.

4.5. Variáveis relacionadas ao tratamento.

4.5.1. Variável ANTIVIRAL

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$ANTIVIRAL)
Ignorado   Não     Sim
  19899   55365  21365
```

Figura 102 – ANTIVIRAL - Categorias e quantidades.

Estes dados fazem referência os pacientes que utilizaram algum medicamento antiviral. Observamos o menor número de pacientes utilizaram medicamento antiviral e neste grupo o número de óbitos é ligeiramente maior.

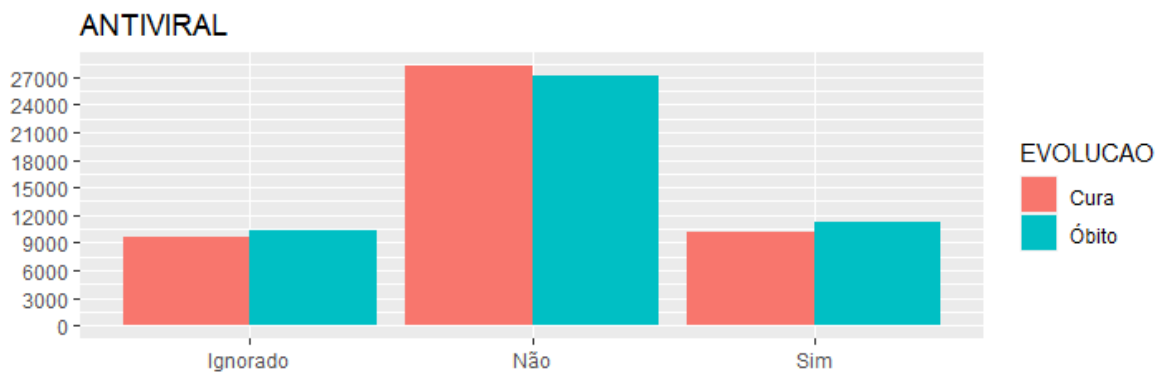


Figura 103 – Relação das variáveis ANTIVIRAL e EVOLUCAO.

4.5.2. Variável HOSPITAL

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$HOSPITAL)
Ignorado   Não     Sim
    967   1948  93714
```

Figura 104 – HOSPITAL - Categorias e quantidades.

Os dados demonstram claramente que 97% dos pacientes foram internados, sendo que o número de óbitos e de cura estão equilibrados.

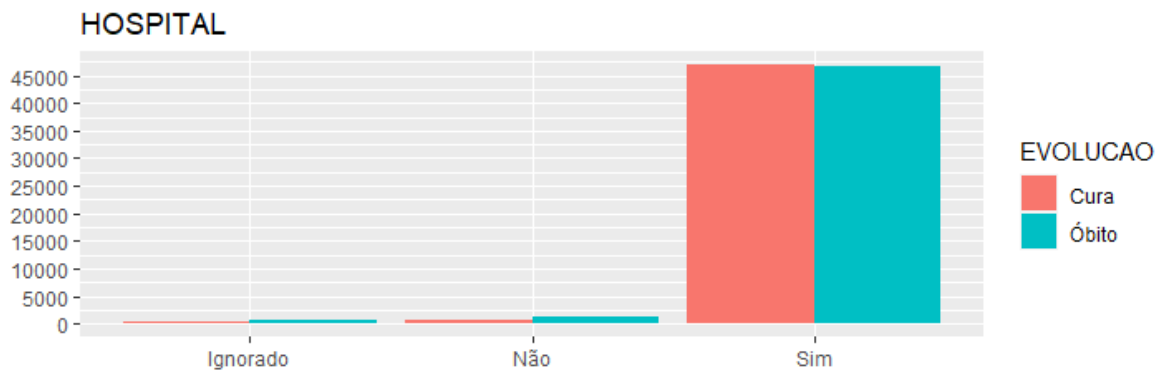


Figura 105 – Relação das variáveis HOSPITAL e EVOLUCAO.

4.5.3. Variável UTI

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$UTI)
Ignorado    Não    Sim
  9225    50038   37366
```

Figura 106 – UTI - Categorias e quantidades.

Dos pacientes que foram internados conforme informações da variável anterior, 39% necessitaram de tratamento na unidade de terapia intensiva – UTI e deste grupo a maioria evoluiu para óbito, enquanto que os pacientes que não necessitaram deste tipo de cuidado, a metade deles apresentou cura.

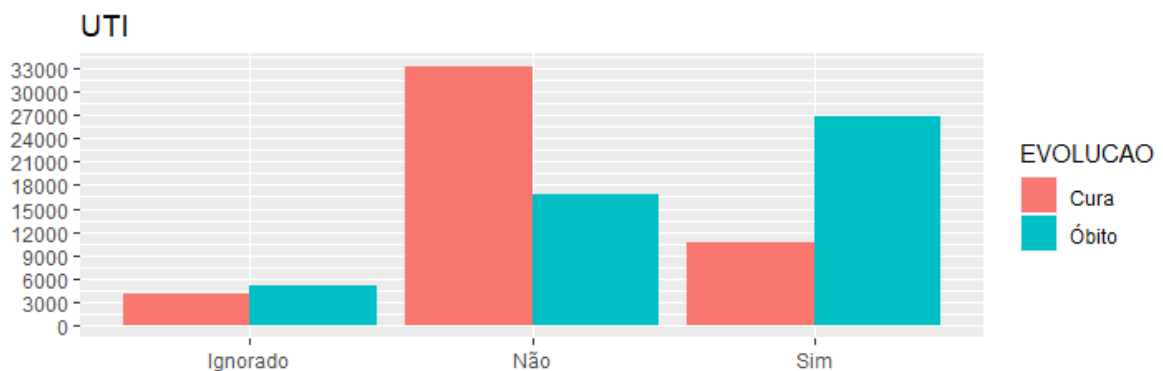


Figura 107 – Relação das variáveis UTI e EVOLUCAO.

4.5.4. Variável SUPORT_VEN

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$SUPORT_VEN)
Ignorado    Não    Sim, invasivo Sim, não invasivo
  11131    19456    20587    45455
```

Figura 108 – SUPORT_VEN- Categorias e quantidades.

Observamos nesta variável que os pacientes que necessitaram de suporte ventilatório invasivo foram os que mais evoluíram para óbito. Dentre os pacientes que não necessitaram de suporte ventilatório bem como aqueles que necessitaram, de forma não invasiva, evoluíram para a cura.

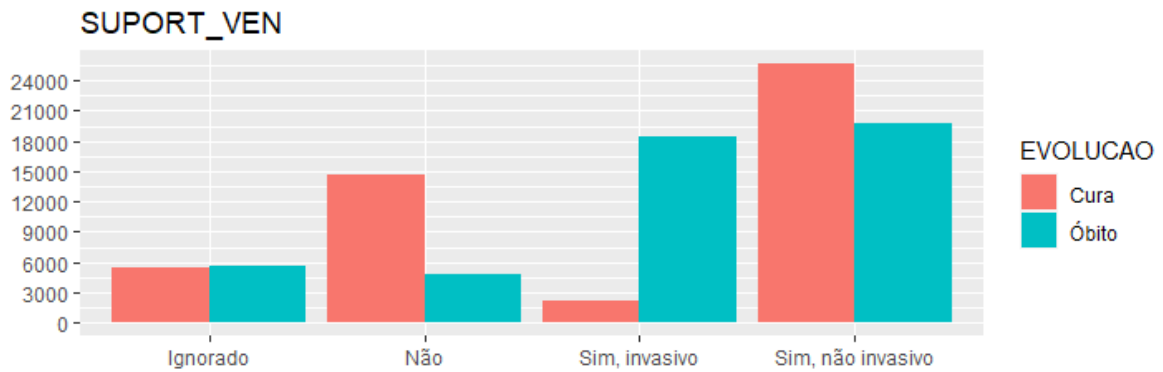


Figura 109 – Relação das variáveis SUPOORT_VEN e EVOLUCAO.

4.5.5. Variável PCR_SARS2

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$PCR_SARS2)
marcado pelo usuário   Não marcado
            82446             14183
```

Figura 110 – PCR_SARS2 - Categorias e quantidades.

Observamos que as informações sobre o resultado do diagnóstico do RT-PCR para SARS-CoV-2 era marcado pelo usuário e que a maioria havia realizado o teste. Entretanto, não há significativa diferença de cura e óbito entre os que fizeram o teste e marcaram e entre os que não marcaram.

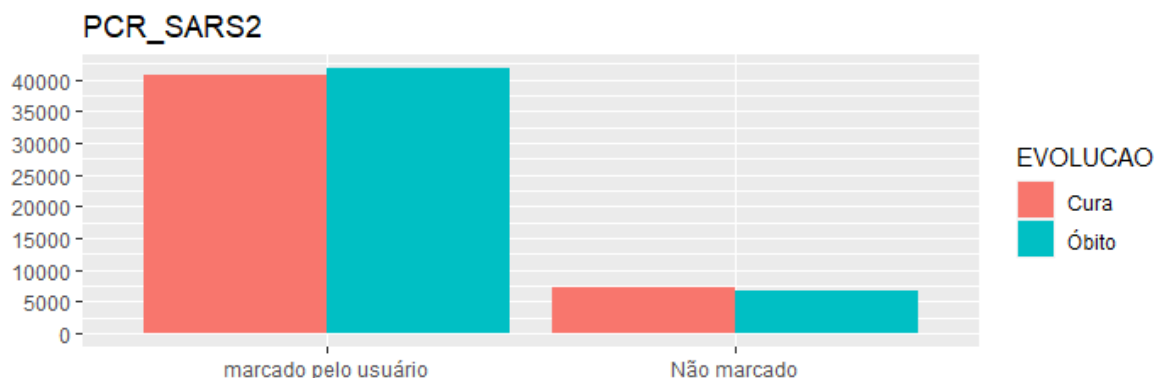


Figura 111 – Relação das variáveis PCR_SARS2 e EVOLUCAO.

4.5.6. Variável TOMO_RES

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$TOMO_RES)
```

Atípico COVID-19	Ignorado	Indeterminado COVID-19	Não realizado
802	61336	1421	5935
Negativo para Pneumonia	Outro	Tipico COVID-19	
177	2044	24914	

Figura 112 – TOMO_RES - Categorias e quantidades.

Analisando os dados sobre o resultado da tomografia, observamos um maior número de cura quando o resultado foi classificado como típico COVID-19.

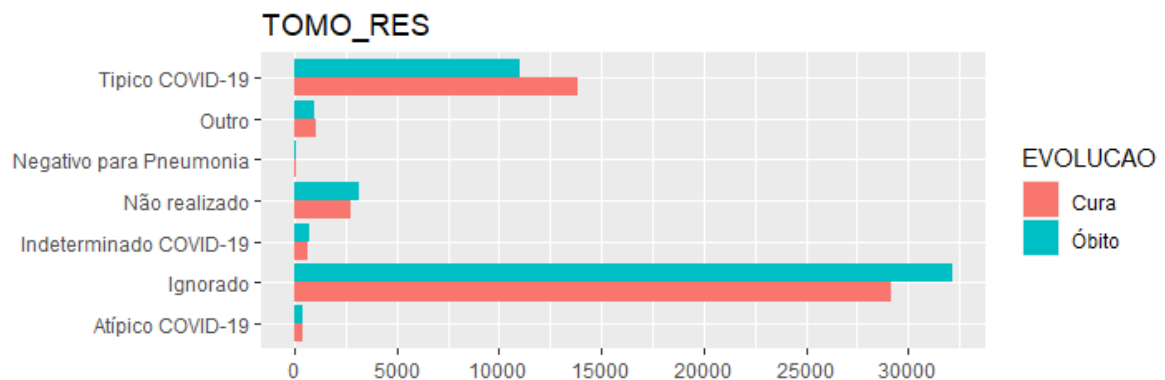


Figura 113 – Relação das variáveis TOMO_RES e EVOLUCAO.

4.5.7. Variável RES_AN

Observamos as seguintes categorias e quantidades:

```
> summary(srag_sp_v1$RES_AN)
```

Aguardando resultado	Ignorado	Inconclusivo
39442	10521	6
Não realizado	Negativo	positivo
41394	1761	3505

Figura 114 – RES_AN - Categorias e quantidades.

Esta variável traz os resultados dos testes antigênicos. Entretanto, não agregou valor para nossa análise.

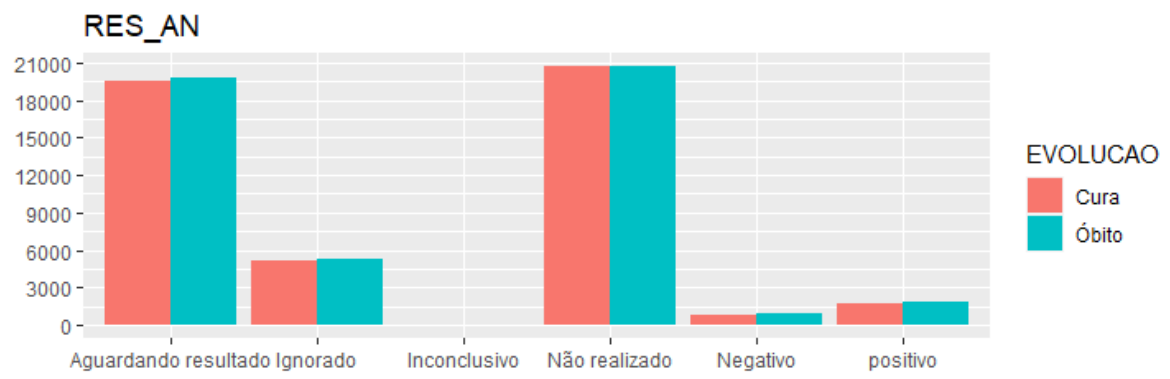


Figura 115 – Relação das variáveis RES_AN e EVOLUCAO.

5. Criação de Modelos de Machine Learning

Devido as características do nosso dataset, dos valores contidos em nossas variáveis e do resultado que pretendemos prever, dentre os diversos tipos de aprendizagem de máquina, utilizaremos a aprendizagem supervisionada.

Neste tipo de aprendizagem, o modelo aprende com os dados pré-definidos, no nosso caso, das 40 variáveis incluindo a variável target, assim referenciando ao modelo, nosso objetivo de prever se os pacientes vão evoluir seu tratamento para cura ou para óbito, conforme a combinação das informações de todas as variáveis.

O aprendizado supervisionado possui duas principais subcategorias: regressão e classificação. De forma geral, a diferença entre elas é basicamente que, a regressão prevê uma quantidade e a classificação, um rótulo. Portanto, como desejamos saber se a evolução do paciente será cura ou óbito, utilizaremos a subcategoria de classificação.

5.1. Padronização estatística dos valores numéricos

Após a eliminação de parte dos registros outliers, optamos por realizar a padronização dos valores das variáveis numéricas: PERIODO_PATOGENICO e IDADES_EM_DIAS. Com isso, pretendemos deixar todos os valores em uma mesma escala, porém sem distorcer as diferenças no intervalo de valores. Na fórmula a seguir, a média dos dados é centralizada assumindo o valor zero e os demais dados são escalados em desvio padrão.

$$X_i = \frac{X_i - \mu}{\sigma}$$

Onde:

μ = Média

σ = Desvio padrão

Figura 116 – Fórmula da padronização

Desta forma, evitaremos que nosso modelo faça previsões enviesado pelos valores das variáveis com maior ordem de grandeza.

```
# Padronização a variável PERIODO_PATOGENICO
srag_sp_v1$PERIODO_PATOGENICO <- scale(srag_sp_v1$PERIODO_PATOGENICO, center=T, scale=T)
# padronização dos valores da variável IDADES_EM_DIAS
srag_sp_v1$IDADE_EM_DIAS <- scale(srag_sp_v1$IDADE_EM_DIAS, center=T, scale=T)
```

Figura 117 – Comando para a padronização das variáveis numéricas.

Abaixo apresentamos as informações estatísticas básicas da variável PERIODO_PATOGENICO, antes e após a padronização

```
summary(srag_sp_v1$PERIODO_PATOGENICO)
Min. 1st Qu. Median Mean 3rd Qu. Max.
0.00 10.00 15.00 18.43 22.00 317.00
```

a

```
summary(srag_sp_v1$PERIODO_PATOGENICO)
v1
Min. :-1.86360
1st Qu. :-0.68549
Median :-0.09643
Mean : 0.00000
3rd Qu. : 0.61044
Max. : 2.84885
```

b

Figura 118 – Período patogênico – Antes da padronização (a) / Após a padronização (b)

Na sequência, estão as informações estatísticas básicas da variável IDADES_EM_DIAS, antes e após a padronização

```
summary(srag_sp_v1$IDADE_EM_DIAS)
Min. 1st Qu. Median Mean 3rd Qu. Max.
0 18560 23471 22870 27694 40312
```

a

```
summary(srag_sp_v1$IDADE_EM_DIAS)
v1
Min. :-2.24561
1st Qu. :-0.73506
Median : 0.05821
Mean : 0.00000
3rd Qu. : 0.76353
Max. : 2.86145
```

b

Figura 119 – Idade em dias – Antes da padronização (a) / Após a padronização (b)

5.2. Seleção das variáveis de maior relevância aos modelos

O primeiro algoritmo que vamos utilizar trata-se do recursive feature elimination – RFE (Eliminação recursiva de recursos). Considerando que temos 40 variáveis, precisamos identificar aquelas de maior relevância para utilizarmos em nossos modelos de previsão.

Analisando cada um dos gráficos do capítulo anterior, poderíamos escolher as variáveis que verificamos as informações de maior relevância. Entretanto, preferimos além de nossas análises, submeter as informações a uma função que analisa os

dados de cada variável, atribuindo a cada uma delas um valor de importância, classificando-as da maior relevância a menos relevante. Desta forma, podemos fornecer aos modelos de previsão um conjunto das melhores variáveis preditoras, contribuindo para a precisão e acertos.

Definimos que a função fará 20 interações (validações cruzadas) entre todas as variáveis, exceto da variável evolução, que será o parâmetro de acerto da função. Para este modelo, selecionamos 10% dos dados de nosso dataset.

```
#####
#### Amostra de 10% dos dados para a seleção das variáveis mais relevantes
#####
#verificando informações da amostragem de 10% do dataset para a seleção de variáveis.
iamostra <- sample(1:nrow(srag_sp_v1), size = 0.1 * nrow(srag_sp_v1))
amostra_selecao_variaveis <- srag_sp_v1[iamostra,]
```

Figura 120 – Amostragem para aplicação do modelo RFE, de seleção de variáveis.

Definindo a amostra, aplicamos o modelo conforme segue.

```
#####
#== Feature Selection - Seleção de Variáveis
#####
#?rfe
#Função para seleção de variáveis
run.feature.selection <- function(num.iters=20, feature.vars, class.var){
  set.seed(10)
  variable.sizes <- 1:39
  control <- rfeControl(functions = rfFuncs, method = "cv",
                        verbose = FALSE, returnResamp = "all",
                        number = num.iters)
  results.rfe <- rfe(x = feature.vars, y = class.var,
                    sizes = variable.sizes,
                    rfeControl = control)
  return(results.rfe)
}

train.data[, -40]

# Executando a função
rfe.results <- run.feature.selection(feature.vars = amostra_selecao_variaveis[, -40],
                                   class.var = amostra_selecao_variaveis[, 40])

#verificando os resultados da seleção de variáveis
rfe.results

# Salvando os resultado da seleção das variáveis
saveRDS(rfe.results, "rfe_results.rds")

#####
#== FIM - Feature Selection - Seleção de Variáveis
#####
```

Figura 121 – Função para seleção de variáveis.

O resultado do desempenho obtido pela reamostragem de 20 validações cruzadas sobre todas as variáveis foram:

variables	Accuracy	Kappa	AccuracySD	KappaSD
1	0.5918	0.1836	0.01204	0.02409
2	0.7424	0.4848	0.01821	0.03640
3	0.7632	0.5264	0.01991	0.03981
4	0.7744	0.5488	0.01295	0.02589
5	0.7719	0.5438	0.01279	0.02558
6	0.7684	0.5368	0.01339	0.02677
7	0.7678	0.5357	0.01629	0.03257
8	0.7715	0.5430	0.01472	0.02945
9	0.7685	0.5370	0.01354	0.02707
10	0.7735	0.5471	0.01263	0.02526

29	0.7740	0.5479	0.01420	0.02839
30	0.7752	0.5504	0.01310	0.02619
31	0.7756	0.5512	0.01445	0.02889
32	0.7744	0.5488	0.01479	0.02958
33	0.7783	0.5566	0.01348	0.02696
34	0.7793	0.5587	0.01356	0.02712
35	0.7787	0.5574	0.01339	0.02677
36	0.7753	0.5506	0.01323	0.02645
37	0.7774	0.5548	0.01468	0.02935
38	0.7760	0.5521	0.01379	0.02757
39	0.7777	0.5554	0.01660	0.03320

Figura 122 – Classificação ordenada das 10 primeiras e 10 últimas variáveis.

A classificação por variáveis apresenta a seguinte sequência:

<code>> varImp((rfe.results))</code>		PERD_PALA	7.884890
	overall	OUTRO_SIN	7.717963
IDADE_EM_DIAS	84.504236	PERD_OLFT	7.596662
SUPORT_VEN	79.942860	ASMA	7.556680
UTI	46.170712	FEBRE	7.479871
PERIODO_PATOGENICO	28.898173	CARDIOPATI	7.039692
SATURACAO	14.985960	OBESIDADE	6.640277
FATOR_RISC	12.089064	FADIGA	6.516558
CS_RACA	11.400354	DOR_ABD	6.350192
TOMO_RES	11.364261	DISPNEIA	6.219324
OUT_MORBI	10.591318	HEMATOLOGI	5.760173
DESC_RESP	10.523979	PNEUMOPATI	5.750993
CS_ESCOL_N	10.062409	ANTIVIRAL	5.561881
GARGANTA	9.789818	SIND_DOWN	5.545054
HOSPITAL	9.514815	IMUNODEPRE	5.419638
VOMITO	9.319552	NEUROLOGIC	5.321956
DIARREIA	8.999966	RENAL	5.268660
TOSSE	8.465544	PUERPERA	5.205503
DIABETES	8.331524		

a

b

Figura 123 – Classificação das variáveis conforme função RFE.
(a) Variáveis da 1ª a 17ª / (b) Variáveis da 18ª a 35ª.

As variáveis CS_SEXO, HEPATICA, PCR_SARS2 e RES_AN não obtiveram pontuação na classificação das variáveis preditoras e por isso não aparecem na listagem acima.

5.3.Divisão dos dados e variáveis para treino e teste.

Com relação aos dados, os modelos farão uso de 100% dos dados disponíveis, sendo utilizados 60% para treino, os outros 40% restantes para teste. Para isso, de forma aleatória com uso da função `sample`, colocamos 60% dos registros em uma variável chamada `indexes`. Então atribuímos estes dados a variável de treino e, os dados que ficaram de fora a variável de teste.

```
#####
#== Definindo a Amostragem - Dados de treino e teste
#####

##### Dados e variáveis de todo dataset
#####
# Dividindo os dados em treino e teste - 60:40 ratio (cross validate)
indexes <- sample(1:nrow(srag_sp_v1), size = 0.6 * nrow(srag_sp_v1))

# Preparando dados de treino e teste de todo dataset
dados_treino_full <- srag_sp_v1[indexes,] # 60% dos dados para treinar o modelo
dados_teste_full <- srag_sp_v1[-indexes,] # 40% dos dados para testar o modelo
```

Figura 124 – Padronização das variáveis numéricas.

Desta forma, nosso modelo terá as seguintes quantidades de treino e teste.

```
> dim(dados_treino_full)
[1] 57977 40
```

a

```
> dim(dados_teste_full)
[1] 38652 40
```

b

Figura 125 – quantidade de registros – Para treino (a) / Para teste (b).

Para que todos os modelos recebam os mesmos parâmetros de testes das hipóteses, deixaremos todos os dados e variáveis padronizados, para as aplicações.

Pretendemos explorar duas hipóteses de treino e teste, com cada um dos modelos que vamos utilizar, sendo um a primeira com a disponibilização de todas as variáveis e a segunda disponibilizando apenas as variáveis de maior relevância, segundo a classificação RFE.

Com a aplicação das duas hipóteses, poderemos verificar o comportamento de cada modelo, buscando o maior número de acerto no próprio, observar seu comportamento quando recebe diferentes grupos de variáveis, bem como comparar o comportamento individual com os demais modelos de cada modelo.

Assim definimos os parâmetros que serão aplicados aos modelos quando da hipótese de receberem todas as variáveis. Os valores da variável `target`, anteriormente 1 para cura e 2 para óbito, foram respectivamente substituídos para 0 e 1, compatibilizando os dados com o modelo, conforme abaixo.

```

# Preparando variáveis de teste
var_teste_full <- dados_teste_full[,-40]
#pegando as 39 variáveis, menos a última que é a de evolução

# Separando a variável de treino
var_teste_full_target <- dados_teste_full[,40]
#pegando somente a variável target

# Transformando os dados para zero e um, compatibilizando com os dados do modelo.
var_teste_full_target = factor(var_teste_full_target, labels = c(0, 1))

```

Figura 126 – Definição de todas as variáveis de treino e de teste.

Com relação as variáveis de maior relevância segundo modelo RFE, embora o modelo tenha deixado somente 04 variáveis fora da classificação, optamos por utilizar somente aquelas que obtiveram overall maior que 10. Com esta decisão, pretendemos somente as de maior relevância aos modelos podendo melhorar seus acertos, além de que, ficará mais evidente a diferença dos resultados quando um modelo receber todas as variáveis ou somente as de maior relevância.

```

##### Dados e somente as variáveis mais relevantes
#####
# Preparar o dataset com as 11 mais relevantes + variável preditora
list_var_maior_relevancia <- c('IDADE_EM_DIAS', 'SUPOORT_VEN', 'UTI', 'PERIODO_PATOGENICO',
'SATURACAO', 'FATOR_RISC', 'CS_RACA', 'TOMO_RES', 'OUT_MORBI', 'DESC_RESP', 'CS_ESCOL_N', 'EVOLU')

# Preparando dados de treino e teste
dados_treino_maior_relevancia <- dados_treino_full[,list_var_maior_relevancia]
dados_teste_maior_relevancia <- dados_teste_full[,list_var_maior_relevancia]

# Preparando variáveis de teste
var_teste_maior_relevancia <- dados_teste_maior_relevancia[,-12] #todas as variáveis menos a EVOLU
var_teste_maior_relevancia_target <- dados_teste_maior_relevancia[,12] #Pega somente a variável EVOLU

#transformando os dados para zero e um, compatibilizando com os dados do modelo.
var_teste_maior_relevancia_target = factor(var_teste_maior_relevancia_target, labels = c(0, 1))
#####
#== FIM - Dados e somente as variáveis mais relevantes
#####

```

Figura 127 – Definição de todas as variáveis de maior relevância segundo modelo RFE.

5.4. Modelo Linear Generalizado - GLM.

5.4.1. Todas as variáveis.

O primeiro modelo vamos utilizar é o GLM: trata-se de um algoritmo de regressão logística utilizado na forma binomial.

Nesta etapa, estamos criando o modelo de previsão e nesta hipótese, fornecemos para o seu treinamento, todas as variáveis de teste, isso significa, que o modelo está recebendo 40 variáveis com 57977 registros. Abaixo demostramos a parte do código onde o modelo é criado e treinado.

```

#=====
# modelo_glm_full_var - Construindo um modelo de regressão logística utilizando todas as
# variáveis do dataset (39 variáveis) (exceto a variável target)
#=====
# ?glm

# Montando o modelo de regressão logística, (GLM) modelo linear generalizado da família binomial
target_formula <- "EVOLU ~ ."
target_formula <- as.formula(target_formula)
modelo_glm_full_var <- glm(formula = target_formula, data = dados_treino_full,
                           family = "binomial")

# Salvando o modelo utilizado no TCC
saveRDS(modelo_glm_full_var, "result_regressao_glm_full_var.rds")

# Carregando resultado modelo GLM com todas as variáveis
modelo_glm_full_var <- readRDS("result_regressao_glm_full_var.rds", refhook = NULL)

# Visualizando o modelo
summary(modelo_glm_full_var)

```

Figura 128 – Modelo de regressão logística utilizando todas as variáveis para treino.

Com o comando “summary”, podemos observar um extenso relatório dos resultados obtidos pelo modelo, tais como ele considerou a relevância de cada variável, o erro padrão, valor Z etc.

Para testar este modelo utilizaremos a função “predict”. Esta função calcula a probabilidade de sequências categóricas, com base nos valores de entrada. Abaixo ilustramos o uso da função “predict”, recebendo como parâmetro o modelo criado e treinado anteriormente, recebendo também 40 variáveis como 38652 dados de teste.

```

# Testando o modelo com os dados de teste
modelo_glm_full_predicao <- predict(modelo_glm_full_var, dados_teste_full, type="response")
modelo_glm_full_predicao <- round(modelo_glm_full_predicao)

```

Figura 129 – Modelo de regressão logística – Testando o modelo.

Para avaliar resultado do modelo, utilizamos a matriz de confusão, que recebe como parâmetros os resultados da “predict”, bem como a variável target. Faremos a apresentação dos resultados e comparações no próximo capítulo.

```

# Avaliando o modelo com os valores da variável target
var_matriz_confusao_glm_full_var <- confusionMatrix(table(data = modelo_glm_full_predicao,
                                                           reference = var_teste_full_target), positive = '1')

```

Figura 130 – Modelo de regressão logística – Avaliando o modelo.

5.4.2. Com 11 variáveis de maior relevância.

Nesta hipótese, utilizaremos o mesmo modelo de regressão logística da família binomial, porém, ele receberá as informações somente das 11 variáveis de maior relevância de acordo com o modelo RFE, juntamente com a variável target.

A sequência é a mesma do modelo anterior, mudando apenas os valores que são recebidos como parâmetros das variáveis de treino e teste. Portanto, nesta

hipótese, o modelo receberá 11 variáveis de maior relevância mais a target, com 57977 registros para treino.

```
#=====
#== modelo_glm_relevante_var - Construindo um modelo de regressão logística utilizando variáveis
# que tiveram pontuação geral maior que 10 no algoritmo recursive feature elimination - RFE
#=====

# Montando o modelo de regressão logística, (GLM) modelo linear generalizado da família binomial
target_formula <- "EVOLU ~ ."
target_formula <- as.formula(target_formula)
modelo_glm_relevante_var <- glm(formula = target_formula, data = dados_treino_maior_relevancia,
                                family = "binomial")

# Salvando o modelo utilizado no TCC
saveRDS(modelo_glm_relevante_var, "result_regressao_glm_relevante_var.rds")

# Carregando resultado modelo GLM com as principais variáveis.
modelo_glm_relevante_var <- readRDS("result_regressao_glm_relevante_var.rds", refhook = NULL)

# Visualizando o modelo
summary(modelo_glm_relevante_var)
```

Figura 131 – Modelo de regressão logística utilizando 11 variáveis de maior relevância segundo RFE.

O teste também será efetuado com a função “predict”, porém recebendo 11 variáveis de maior relevância juntamente com a variável target e 38652 dados de teste.

```
# Testando o modelo com os dados de teste
modelo_glm_relevantes_predicao <- predict(modelo_glm_relevante_var,
                                           dados_teste_maior_relevancia, type="response")
modelo_glm_relevantes_predicao <- round(modelo_glm_relevantes_predicao)
```

Figura 132 – Modelo de regressão logística – Testando o modelo com 12 variáveis.

Abaixo a matriz de confusão com os parâmetros desta hipótese. Os resultados e comparações serão explicitados no próximo capítulo.

```
# Avaliando o modelo com os valores da variável target
var_matriz_confusao_glm_relevante_var <- confusionMatrix(table(data = modelo_glm_relevantes_predicao,
                                                                reference = var_teste_maior_relevancia_target), positive = '1')
```

Figura 133 – Modelo de regressão logística – Avaliando o modelo treinado com 12 variáveis.

5.5. Modelo Naives Bayes.

5.5.1. Todas as variáveis.

Este modelo, é muito popular e utilizado na aprendizagem de máquina. Possui grande simplicidade para seu uso e desempenho relativamente maior que outros classificadores, com respeitável precisão. Para este modelo cada variável é única e independente, devido a esta característica, tem um tempo menor para realizar a classificação. Nesta hipótese, vamos treinar o modelo com todas as

variáveis e com os 57977 registros. Abaixo demostramos a parte do código onde o modelo é criado e treinado.

```
#=====
# modelo_naive_full_var - Construindo um modelo de regressão logística utilizando
# todas as variáveis do dataset (39 variáveis) (exceto a variável target)
#=====

# Montando o modelo classificador probabilístico
target_formula <- "EVOLU ~ ."
target_formula <- as.formula(target_formula)
modelo_naive_full_var <- naiveBayes(formula = target_formula, data = dados_treino_full)

# Salvando o modelo utilizado no TCC
saveRDS(modelo_naive_full_var, "result_regressao_naives_full_var.rds")

# Carregando resultado modelo NaivesBayes com todas as variáveis
modelo_naive_full_var <- readRDS("result_regressao_naives_full_var.rds", refhook = NULL)
```

Figura 134 – Modelo de Naive Bayes - testando o modelo.

A função “predict” também será utilizada para testar as previsões do modelo. Como parâmetros fornecemos os como 38652 dados de teste das 40 variáveis. Na sequência, transformamos o resultado para fator, convertendo os valores de cura o óbito para zero e um, compatibilizando os resultados da previsão com o modelo.

```
# Testando o modelo com os dados de teste
modelo_naive_full_predicao <- predict(modelo_naive_full_var, dados_teste_full)

# Transformando os dados para zero e um, compatibilizando com os dados do modelo.
modelo_naive_full_predicao = factor(modelo_naive_full_predicao, labels = c(0, 1))
```

Figura 135 – Modelo de Naive Bayes - Testando o modelo.

Os resultados que avaliaremos no próximo capítulo serão gerados pela matriz de confusão.

```
# Avaliando o modelo com os valores da variável target
var_matriz_confusao_naive_full_var <- confusionMatrix(table(data = modelo_naive_full_predicao,
                                                             reference = var_teste_full_target), positive = '1')
```

Figura 136 – Modelo de Naive Bayes - Avaliando o modelo.

5.5.2. Com 11 variáveis de maior relevância.

Nesta hipótese, utilizamos o modelo Naives Bayes, porém somente fornecendo como parâmetros os dados das variáveis de maior relevância.

```
#=====
#== modelo_naive_relevante_var - Construindo um modelo de regressão logística utilizando variáveis
# que tiveram pontuação geral maior que 10 no algoritmo recursive feature elimination - RFE
#=====

# Montando o modelo classificador probabilístico
target_formula <- "EVOLU ~ ."
target_formula <- as.formula(target_formula)
modelo_naive_relevante_var <- naiveBayes(formula = target_formula, data = dados_treino_maior_relevancia)

# Salvando o modelo utilizado no TCC
saveRDS(modelo_naive_relevante_var, "result_regressao_naives_relevante_var.rds")

# Carregando resultado modelo NaivesBayes com as principais variáveis.
modelo_naive_relevante_var <- readRDS("result_regressao_naives_relevante_var.rds", refhook = NULL)
```

Figura 137 – Modelo de Naive Bayes utilizando 11 variáveis de maior relevância segundo RFE.

A função “predict” também será utilizada para testar o modelo, assim como no anterior. Na sequência, transformamos o resultado para fator, compatibilizando-os com o modelo

```
# Testando o modelo com os dados de teste
modelo_naive_relevante_predicao <- predict(modelo_naive_relevante_var, dados_teste_maior_relevancia)

# Transformando os dados para zero e um, compatibilizando com os dados do modelo.
modelo_naive_relevante_predicao = factor(modelo_naive_relevante_predicao, labels = c(0, 1))
```

Figura 138 – Modelo de Naive Bayes - Testando o modelo com 12 variáveis.

Os resultados que avaliaremos no próximo capítulo serão gerados pela matriz de confusão.

```
# Avaliando o modelo com os valores da variável target
var_matriz_confusao_naive_relevante_var <- confusionMatrix(table(data = modelo_naive_relevante_predicao,
                                                                reference = var_teste_maior_relevancia_target), positive = '1')
```

Figura 139 – Modelo de Naive Bayes - – Avaliando o modelo treinado com 12 variáveis

6. Apresentação dos Resultados

Para verificar o desempenho dos modelos, utilizaremos uma tabela chamada Matriz de Confusão. Nesta matriz, são expressos de forma clara e objetiva, os valores reais da variável target que fornecemos para o teste, bem como os valores previstos pelo modelo. A tabela abaixo ilustra as respectivas posições dos resultados com seus significados na matriz de confusão.

Dados Previsão	Dados teste	
	Cura	Óbito
Cura	Cura Acerto Verdadeiro Positivo (TP)	Óbito Erro Falso Negativo (FN)
Óbito	Cura Erro Falso Positivo (FP)	Óbito Acerto Verdadeiro Negativo (TN)

- **Cura acerto (true positive - TP):** Os valores reais da classe que estamos buscando foi prevista corretamente. Então os valores que fornecemos no teste coincidem os previstos pelo modelo, ou seja, os pacientes evoluíram para cura, e o modelo previu o a cura destes pacientes.
- **Cura erro (false positive - FP):** Os valores reais da classe que estamos buscando prever foi previsto incorretamente. Neste caso, os valores que fornecemos no teste são diferentes dos previstos pelo modelo, ou seja, os pacientes evoluíram para cura, e o modelo previu o óbito destes pacientes.
- **Óbito acerto (true negative - TN):** Os valores reais da classe que estamos buscando foi prevista corretamente. Então os valores que fornecemos no teste coincidem os previstos pelo modelo, ou seja, os pacientes evoluíram para óbito, e o modelo previu o óbito destes pacientes.
- **Óbito erro (false negative - FN):** Os valores reais da classe que estamos buscando prever foi previsto incorretamente. Neste caso, os valores que fornecemos no teste são diferentes dos previstos pelo modelo, ou seja, os pacientes evoluíram para óbito, mas o modelo previu a cura destes pacientes.

Complementando a matriz de confusão, vamos ilustrar o desempenho dos modelos com o gráfico da curva Receiver Operating Characteristic - ROC e Area Under the Curve - AUC.

Com a curva ROC, podemos verificar o quanto preciso ficou o modelo em distinguir duas coisas, no nosso estudo entre cura e óbito. A Curva ROC é um gráfico simples, mas robusto, que permite representar a relação normalmente antagônica entre a sensibilidade e especificidade.

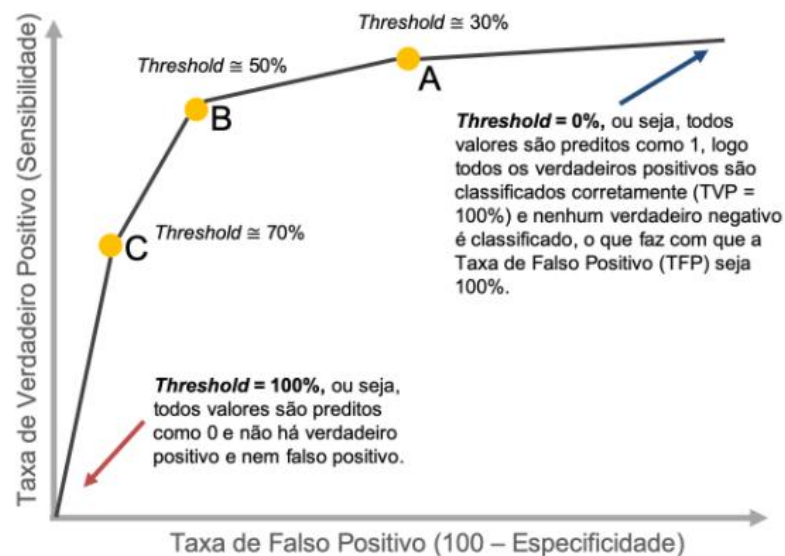


Figura 140 – Exemplo da curva ROC – Sensibilidade x Especificidade.

Fonte: <https://cienciaenegocios.com/curva-roc-e-auc-em-machine-learning/> acesso em 15/04/2021.

Para um melhor entendimento da curva ROC, abaixo demostramos três curvas de exemplo didáticas, na sequência o comportamento de cada uma delas em uma curva ROC.

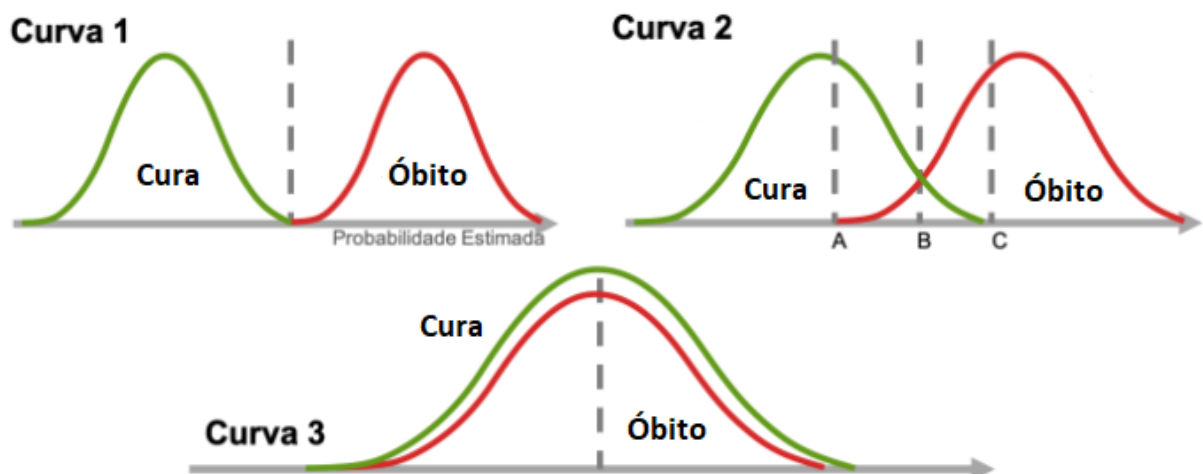


Figura 141 – Exemplo da curva ROC – Sensibilidade x Especificidade.

Fonte: Adaptado de <https://cienciaenegocios.com/curva-roc-e-auc-em-machine-learning/> acesso em 15/04/2021.

As três curvas didáticas acima ilustradas, são representadas na curva ROC conforme abaixo:

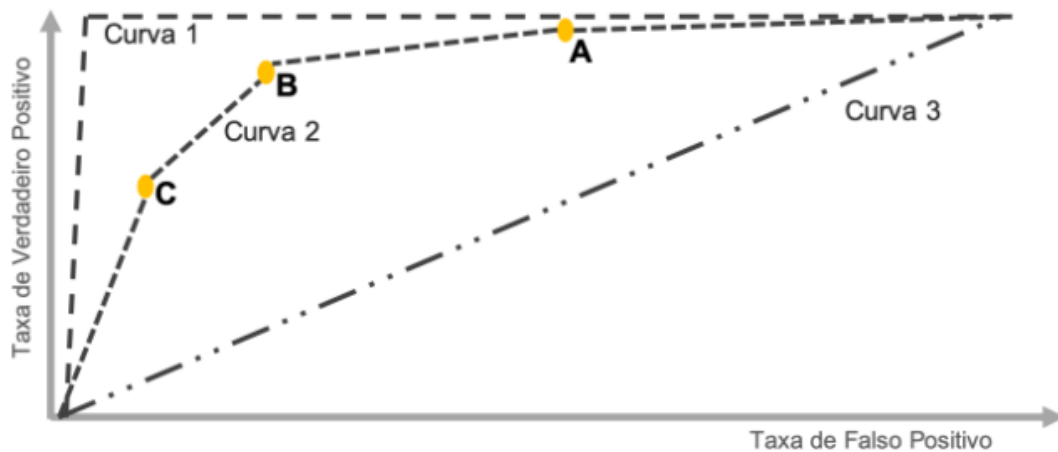


Figura 142 – Exemplo da curva ROC – Sensibilidade x Especificidade.

Fonte: <https://cienciaenegocios.com/curva-roc-e-auc-em-machine-learning/> acesso em 15/04/2021.

No caso da curva 1, um modelo teria acertado 100% das curas e dos óbitos. Na curva 2, que muito se assemelha a maioria dos resultados, o modelo tem grande parte de acertos, mas existem falsas curas e falsos óbitos. Na curva 3, o modelo não agregou nenhum valor, ficando tudo em 50%. Outras variações podem ocorrer, inclusive com resultados abaixo de 50%, demonstrando o fraco desempenho do modelo ou os dados que ele recebeu em seu treinamento.

Com relação a AUC, trata-se de um número que varia de 0% a 100%, representado pela área do gráfico abaixo da curva. Este valor pode auxiliar quando pela curva ROC, se torna difícil pela semelhança das curvas de modelos distintos. Portanto, quanto maior o valor de AUC, melhor foi o resultado do modelo.

6.1. Modelo Linear Generalizado - GLM.

6.1.1. Todas as variáveis

O resultado do modelo GLM nesta hipótese, atingiu acurácia 77,76%, quando fornecemos para o seu treinamento todas as 40 variáveis existentes no dataset. Para esta hipótese, a matriz de confusão apresentou-se da seguinte forma:

		Dados teste	
Dados Previsão		Cura	Óbito
	Cura	15200	4471
	Óbito	4127	14854

Abaixo verificamos outros valores da matriz de confusão na aferição dos valores:

```
> arq_matriz_confusao_glm_full_var
Confusion Matrix and Statistics

      reference
data    0      1
 0 15200  4471
 1  4127 14854

      Accuracy : 0.7776
      95% CI   : (0.7734, 0.7817)
  No Information Rate : 0.5
  P-Value [Acc > NIR] : < 2.2e-16

      Kappa : 0.5551

  Mcnemar's Test P-Value : 0.0002164

      Sensitivity : 0.7686
      Specificity : 0.7865
   Pos Pred Value : 0.7826
   Neg Pred Value : 0.7727
      Prevalence : 0.5000
   Detection Rate : 0.3843
   Detection Prevalence : 0.4911
   Balanced Accuracy : 0.7776

      'Positive' class : 1
```

Figura 143 – Modelo GLM com 40 variáveis de treino.

Conforme indica o valor AUC, o modelo GLM recebendo todas as variáveis para treino atingiu 86% de acerto. A curva ROC do modelo GLM recebendo todas as variáveis ficou da seguinte forma:

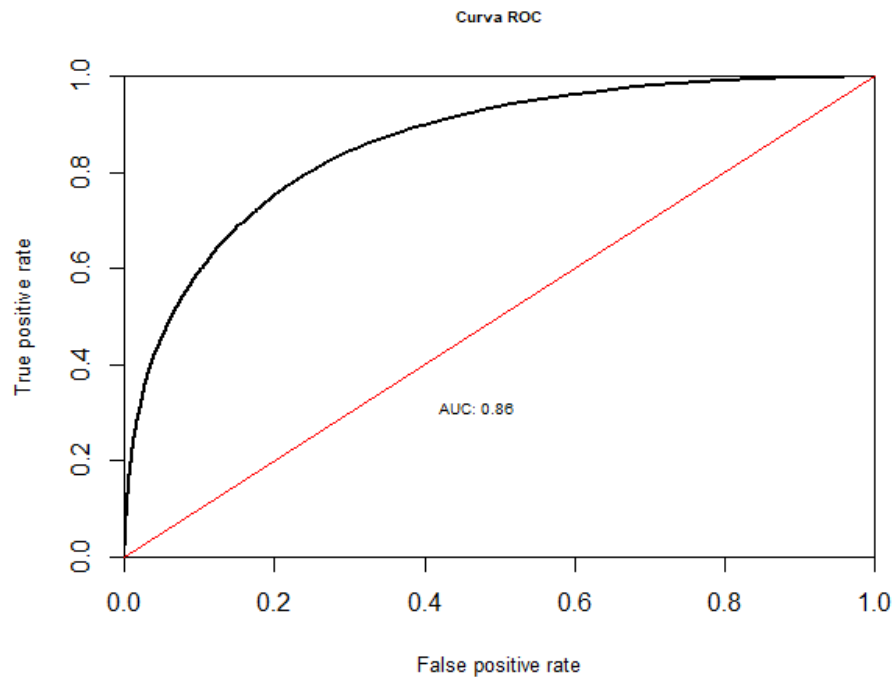


Figura 144 – Curva ROC - modelo GLM com 40 variáveis de treino.

6.1.2. Com 11 variáveis de maior relevância

O resultado do modelo nesta hipótese foi uma acurácia de 76,72% quando fornecemos para o seu treinamento somente as 11 variáveis da maior relevância segundo o modelo RFE. Para esta hipótese, a matriz de confusão ficou da seguinte forma:

		Dados teste	
		Cura	Óbito
Dados Previsão	Cura	14877	4721
	Óbito	4278	14776

Abaixo verificamos outros valores considerados pela matriz de confusão na aferição dos valores:

```

> var_matriz_confusao_glm_relevante_var
Confusion Matrix and Statistics

      reference
data    0      1
 0 14877  4721
 1  4278 14776

      Accuracy : 0.7672
      95% CI   : (0.7629, 0.7714)
  No Information Rate : 0.5044
  P-Value [Acc > NIR] : < 2.2e-16

      Kappa : 0.5344

  Mcnemar's Test P-Value : 3.172e-06

      Sensitivity : 0.7579
      Specificity : 0.7767
   Pos Pred Value : 0.7755
   Neg Pred Value : 0.7591
      Prevalence : 0.5044
   Detection Rate : 0.3823
  Detection Prevalence : 0.4930
   Balanced Accuracy : 0.7673

      'Positive' class : 1

```

Figura 145 – Modelo GLM com 11 variáveis de treino.

Conforme indica o valor AUC, o modelo GLM recebendo somente as 11 variáveis de maior relevância segundo o algoritmo RFE para treino, atingiu 85% de acerto. A curva ROC do modelo GLM recebendo 11 variáveis apresentou-se da seguinte forma:

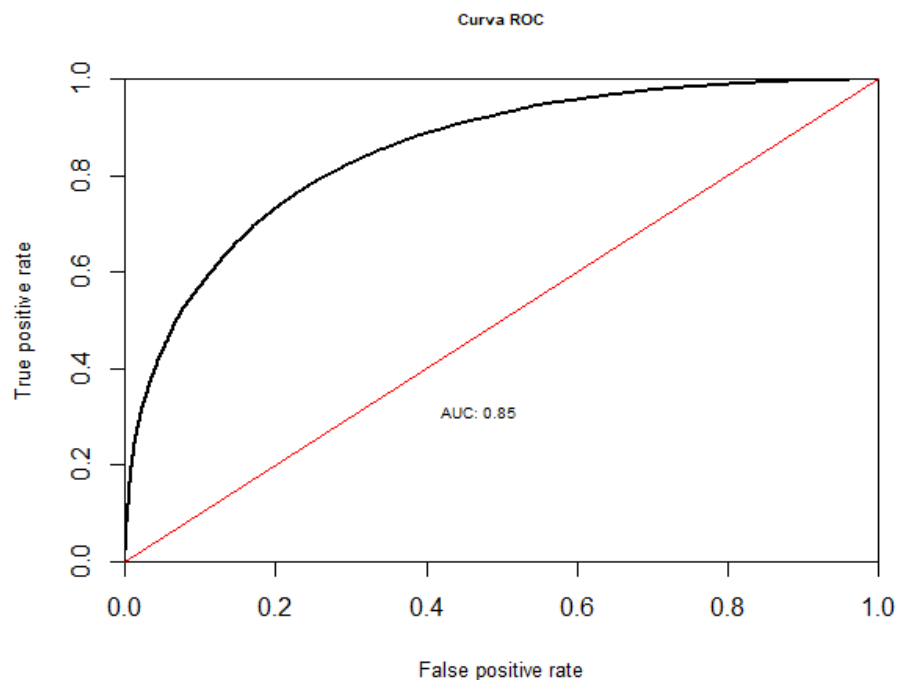


Figura 146 – Curva ROC - modelo GLM com 11 variáveis de treino.

6.2. Modelo Naive Bayes.

6.2.1. Todas as variáveis

O resultado do modelo Naive Bayes teve acurácia geral de 64,30% quando fornecemos para o seu treinamento todas as 40 variáveis existentes no dataset. Para esta hipótese, a matriz de confusão ficou da seguinte forma:

		Dados teste	
Dados Previsão		Cura	Óbito
	Cura	12252	6896
	Óbito	6903	12601

Abaixo verificamos outros valores da matriz de confusão na aferição dos valores:

```
> arq_matriz_confusao_naive_full_var
Confusion Matrix and Statistics

      reference
data    0      1
 0 12252  6896
 1  6903 12601

      Accuracy : 0.643
      95% CI   : (0.6382, 0.6478)
  No Information Rate : 0.5044
  P-Value [Acc > NIR] : <2e-16

      Kappa : 0.2859

  McNemar's Test P-Value : 0.9593

      Sensitivity : 0.6463
      Specificity : 0.6396
   Pos Pred Value : 0.6461
   Neg Pred Value : 0.6399
      Prevalence : 0.5044
  Detection Rate : 0.3260
  Detection Prevalence : 0.5046
  Balanced Accuracy : 0.6430

      'Positive' Class : 1
```

Figura 147 – Modelo Naive Bayes com 40 variáveis de treino.

Conforme indica o valor AUC, o modelo Naive Bayes recebendo todas as variáveis para treino atingiu 64,00% de acerto. A curva ROC do modelo recebendo todas as variáveis ficou da seguinte forma:

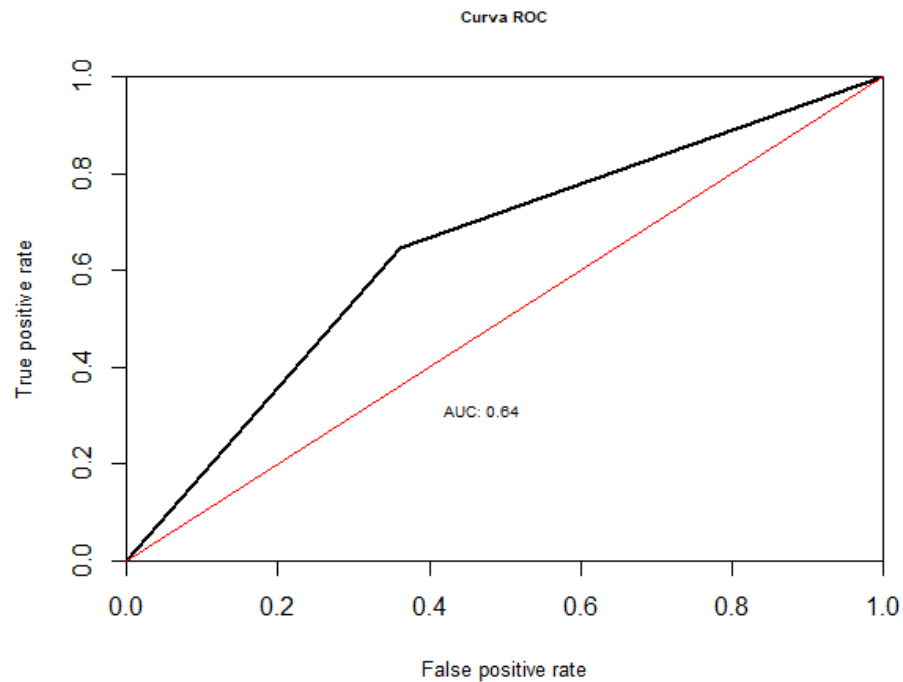


Figura 148 – Curva ROC - modelo Naive Bayes com 40 variáveis de treino.

6.2.2. Com 11 variáveis de maior relevância

O resultado do modelo Naive Bayes teve acurácia de 75,48% quando fornecemos para o seu treinamento somente as 11 variáveis de maior relevância. Para esta hipótese, a matriz de confusão apresenta-se:

		Dados teste	
		Cura	Óbito
Dados Previsão	Cura	14325	4647
	Óbito	4830	14850

Abaixo verificamos outros valores da matriz de confusão na aferição dos valores:


```

> arq_matriz_confusao_naive_relevante_var
Confusion Matrix and Statistics

      reference
data    0      1
 0 14325  4647
 1  4830 14850

      Accuracy : 0.7548
      95% CI   : (0.7505, 0.7591)
  No Information Rate : 0.5044
  P-Value [Acc > NIR] : < 2e-16

      Kappa : 0.5095

  Mcnemar's Test P-Value : 0.06155

      Sensitivity : 0.7617
      Specificity : 0.7478
   Pos Pred Value : 0.7546
   Neg Pred Value : 0.7551
      Prevalence : 0.5044
  Detection Rate : 0.3842
  Detection Prevalence : 0.5092
   Balanced Accuracy : 0.7548

   'Positive' Class : 1

```

Figura 149 – Modelo Naive Bayes com 11 variáveis de treino.

Conforme indica o valor AUC, o modelo Naive Bayes recebendo para treino somente as variáveis de maior relevância, de acordo com o algoritmo RFE, atingiu 75,00% de acerto. A curva ROC deste modelo ficou da seguinte forma:

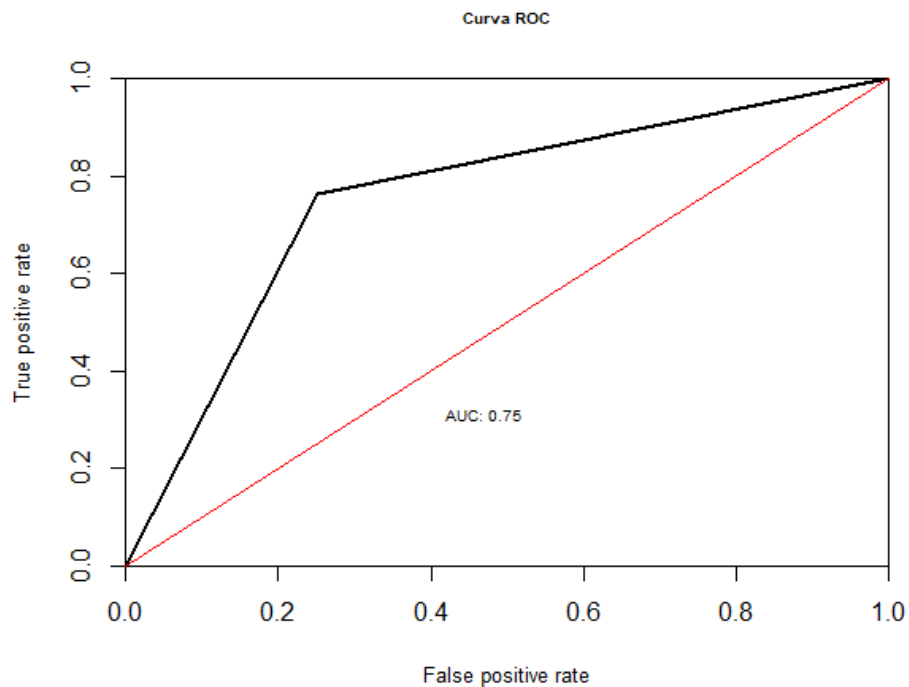


Figura 150 – Curva ROC - modelo Naive Bayes com 11 variáveis de treino.

6.3. Considerações finais dos resultados

Inicialmente, verificamos a importância de testar diversos modelos antes de optar pela aplicação de um específico, pois, mesmo recebendo exatamente os mesmos dados e variáveis, podem ocorrer algumas variações dos resultados, contribuindo para indicar qual foi o melhor para a situação em estudo.

Lembramos que antes de aplicar os modelos, o dataset passou por diversos tratamentos, dentre os quais destacamos:

1. Eliminação de variáveis não diretamente relacionada ao problema;
2. Tratamento dos valores missing;
3. Criação de novas variáveis;
4. Eliminação de variáveis não relevantes ou que foram utilizadas na criação de outras variáveis;
5. Balanceamento do dataset pela variável target;
6. Padronização dos valores das variáveis numéricas.

Portanto, em nosso estudo, utilizamos as variáveis selecionadas e tratadas, após criteriosa análise das variáveis e seus valores.

A algoritmo RFE que utilizamos para selecionar as variáveis de maior relevância foi muito assertivo, sendo seu resultado muito similar ao deste estudante na análise exploratória. Cabe esclarecer que para esta classificação, utilizamos somente 10% dos dados do dataset e, esta amostra levou aproximadamente 5 horas para sua conclusão (Referência: Sistema operacional Windows, Processador I7 de terceira geração, 8Gb de RAM e Disco SSD).

Nas hipóteses dos modelos GLM, observamos que obtive o melhor resultado a hipótese que recebeu todas as variáveis disponíveis no dataset para treino e teste. A hipótese que recebeu somente as 11 variáveis de maior relevância teve o desempenho aproximadamente 1% menor, embora com ligeiro ganho de velocidade na entrega dos resultados.

Nas hipóteses dos modelos Naives Bayes, observamos que obtive o melhor resultado a hipótese que recebeu somente as 11 variáveis de maior relevância, sendo seu desempenho aproximadamente 11% maior, com relação ao teste que recebeu todas as variáveis.

Abaixo demostramos as curvas ROC de todas as hipóteses.

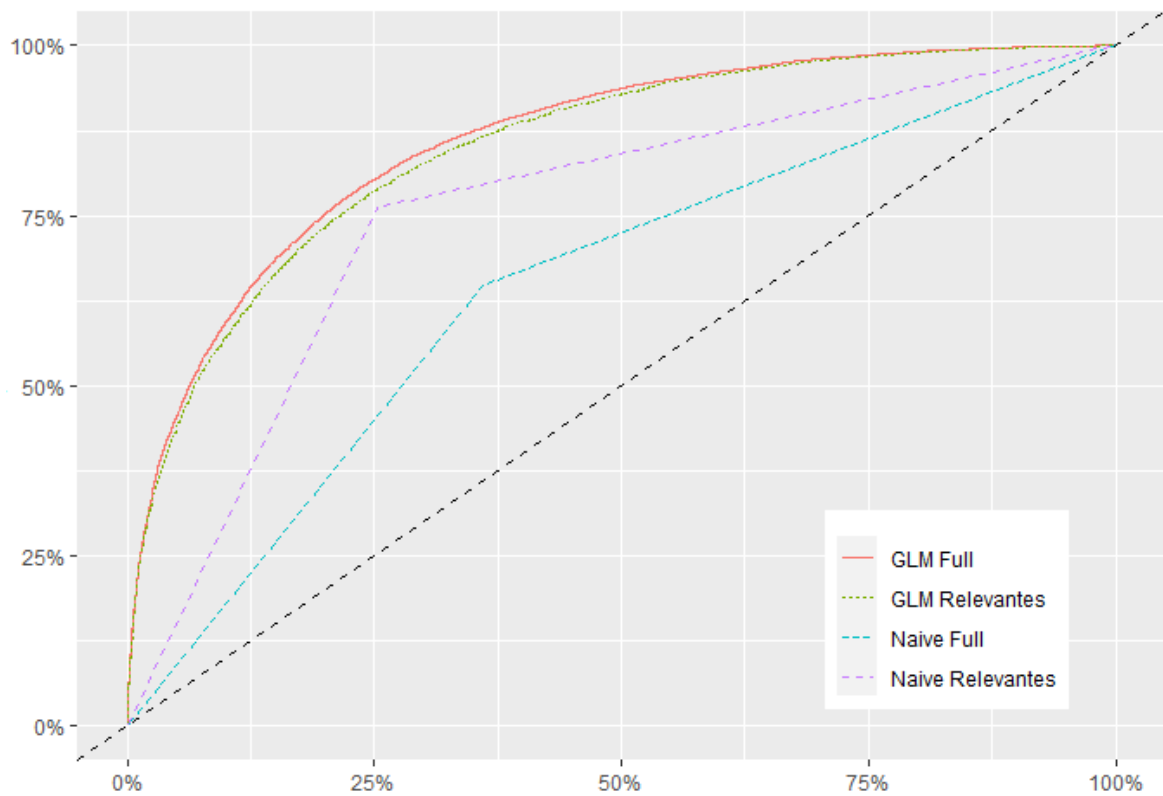


Figura 151 – Curva ROC – Comparativo dentre todas as hipóteses estudadas.

Como resultado deste estudo, concluímos que o modelo GLM conseguiu ser mais generalista quando recebeu as 40 variáveis para treino, apresentando o melhor desempenho dentre todas as hipóteses neste estudo.

7. Links

Abaixo estão descritos os links com o endereço do vídeo de apresentação do estudo e dos repositórios com todos os dados utilizados, sejam obtidos, modificados e resultados parciais salvos.

Link para o vídeo: <https://youtu.be/FzzD5-WD8AM>

Link para o repositório GitHub: https://github.com/lopesmk/TCC_DC.git

Link para o repositório Google Drive:

<https://drive.google.com/drive/folders/1706zWmlrNAg3CbIBOLIJwO7EeR0miP1?usp=sharing>